Wireless Personal Communications (2006) 38: 167–185
DOI: 10.1007/s11277-005-9000-y 

# Enabling End-to-End QoS over Hybrid Wired-Wireless Networks

LIQIANG ZHANG[1] and SHERALI ZEADALLY[2]

[1]*Department of Computer and Information Sciences, Indiana University South Bend, South Bend, Indiana 46634, USA*
*E-mail: liqiang@cs.iusb.edu*
[2]*Department of Computer Science, Wayne State University, Detroit, Michigan 48202, USA*

**Abstract.** Providing end-to-end parameterized QoS is desirable for many network applications and has received a lot of attention in recent years. However, it remains a challenge, especially over hybrid networks involving both wired networks and wireless access segments (such as IEEE 802.11 Wireless Local Area Networks (WLANs)). The difficulty in achieving such QoS arises mainly because wireless segments often constitute "gaps" in terms of resource guarantee, due to the lack of efficient resource scheduling and management ability over shared wireless media, as well as the lack of an appropriate QoS signaling interface to seamlessly embed these wireless segments into an end-to-end QoS signaling system. In this paper, we consider the scenario where an IEEE 802.11 wireless node wishes to make an end-to-end resource reservation to a remote wired Internet node and vice versa. We propose Wireless Subnet Bandwidth Manager (Wireless SBM), an extension of SBM protocol to WLANs, to provide seamless end-to-end resource reservations. Wireless SBM utilizes the enhanced resource management ability provided by Hybrid Coordination Function (introduced in the upcoming IEEE 802.11e standard) to provide parameterized resource reservation and admission control.

**Keywords:** quality of service, end-to-end, hybrid networks, wireless local area networks, signaling protocols, performance evaluation

## 1. Introduction

We have witnessed a tremendous growth of the Internet over the last few years. This rapid growth is demonstrated not only by the exponential increase in node numbers, but also by increasing multimedia traffic such as audio/video streams being delivered between end-users. These time-sensitive applications are expected to be transmitted with strict quality requirements such as bandwidth, delay and jitter. Current IP-based Internet cannot satisfy diverse quality requirements and different user expectations, due to its built-in best-effort mechanism. As a result, there is an urgent need for provisioning of Quality of Service (QoS) support with the evolution of the Internet. Several approaches have emerged as solutions to this demand: IntServ [7, 17], DiffServ [6], MPLS [22], and Generalized MPLS [22] etc. A very important component for the QoS support is the signaling protocol, which allows the end-hosts (and the intermediate nodes) of a given QoS session to communicate the desired QoS level and the corresponding resource amount. The most well-known QoS signaling protocol is the resource ReSerVation Protocol (RSVP) [8], which can be used in IntServ, IntServ over DiffServ [1, 5, 23] as well as MPLS and Generalized MPLS [2, 4]. To extend RSVP support to Local Area Networks (LANs), IETF (Internet Engineering Task Force) has proposed Subnet Bandwidth Manager (SBM) in [24], to address the RSVP-based admission control problem in shared-media IEEE 802 style LANs.

Today's Internet is also growing toward a heterogeneous IP network, where various kind of terminals as well as different transmission technologies are integrated. One example is the prevailing of the IEEE 802.11 Wireless Local Area Networks (WLANs) at home, offices and hot spot locations. In most situations, these WLANs play a role as a wireless extension to the Internet. Such extensions raise challenges in terms of providing end-to-end QoS, due to their characteristics of low capacity, time-varying channel conditions as well as the lack of appropriate signaling support to Internet QoS architectures. The currently deployed IEEE 802.11 WLANs operate with a Distributed Coordination Function (DCF) at MAC layer (the optional Point Coordination Function (PCF) is rarely supported by WLAN vendors) [11], which is based on CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). DCF does not provide any QoS guarantee since all the applications on each wireless station face the same probabilistic opportunity to access the shared wireless media. If we wish to make an end-to-end resource reservation between an IEEE 802.11 WLAN node and a remote wired station in Internet, the wireless cell could constitute a "gap". To resolve this problem, we need the following support: (1) an effective resource scheduling and management mechanism should be provided in WLANs; (2) we also need an appropriate QoS signaling, which is compatible with the QoS signaling protocol for the Internet, and transparent to applications.

In this paper, we address the problem of enabling parameterized QoS over heterogeneous IP networks from the previously discussed two aspects. First, we give a brief review of RSVP and SBM in the following section. Then in Section 3, first we will discuss the Hybrid Coordination Function (HCF) introduced in the upcoming IEEE 802.11e standard. We focus on the HCF Controlled Channel Access (HCCA) mode to investigate its ability on resource scheduling and management. Then we present how to extend SBM to IEEE 802.11e network, various design issues of Wireless SBM will be explored. Some simulation results and ongoing experiments will be discussed in Section 4. Finally, we make some concluding remarks in Section 5.

## 2. Resource ReSerVation Protocol (RSVP) and Subnet Bandwidth Manager (SBM)

Before discussing the issues of extending SBM protocol to IEEE 802.11e WLANs, we present a short review on the RSVP and SBM protocols.

### 2.1. RSVP

Resource ReSerVation Protocol (RSVP) [8] is a signaling protocol that provides resource reservation setup and control to enable integrated services [17]. It is used by a host, on behalf of an application data stream, to request a specific QoS from the network for particular data streams or flows. RSVP is also used by routers to deliver QoS control requests to all nodes along the path(s) of the flows and to establish and maintain states to provide the requested service. RSVP requests will generally, result in resources being reserved in each node along the data path. RSVP is the most complex of all the QoS technologies, for applications (hosts) and for network elements (routers and switches). As a result, it also represents the biggest departure from standard "best-effort" IP service and provides the highest level of QoS in terms of service guarantees, granularity of resource allocation and detail of feedback to QoS-enabled applications and users. RSVP enables two types of services, *namely Guaranteed service* and *Controlled Load service.* The *Guaranteed service* provides firm bounds on end-to-end queuing delays, in addition to ensuring bandwidth according to specified QoS parameters. However, the *Controlled Load service* cannot provide the strictly bounded service that Guaranteed service promises, it is equivalent to "best effort service under unloaded conditions".

The signaling of RSVP utilizes two main category messages – PATH and RESV. Whenever a sender wants to set up a QoS enabled connection to a receiver, it first sends a PATH message with SENDER_TSPEC (traffic specification) object and an optional ADSPEC to the destination. The SENDER_TSPEC object defines the QoS requirement of the ongoing data flow using a token-bucket model, in terms of Token rate, Peak rate, and Maximum packet size etc.; and the ADSPEC is used to collect information in terms of available services, delay and bandwidth estimates. When each RSVP-enabled router along the path receives the PATH message, it creates a PATH state recording the previous hop (PHOP) address (in RSVP_HOP object), and updates the ADSPEC object, then forwards the PATH message downstream to the next hop, until it reaches the receiver. RSVP is a receiver-oriented protocol as it is the receiver that decides if a reservation is to be made. If a receiver decides to make a resource reservation, it will send a RESV message upstream to the sender. The RESV message includes a flow-descriptor, which consists of FLOWSPEC and FILTER_SPEC objects. The receiver chooses reasonable service parameter values for the TSPEC and RSPEC (reservation specification) in the FLOWSPEC object, based on the SENDER_SPEC and ADSPEC objects. Here the TSPEC contains the same kind of parameters as SENDER_TSPEC although values may be different, and RSPEC specifies the bandwidth and the slack term. The FILTER_SPEC is used to identify the packets that should receive the desired QoS (e.g. the transport protocol and port number). When a RSVP-enabled router along the upstream path receives the RESV message, it uses the admission control process to authenticate the request and allocate the necessary resources. If the request cannot be satisfied (due to lack of resources or authorization failure), the router returns an error back to the receiver. If accepted, the router sends the RESV upstream to the next router. When the last router receives the RESV and accepts the request, it sends a confirmation message back to the receiver and also delivers the RESV message to the sender. Once the sender receives the RESV, it starts sending the data packets. A simplified model for above process is shown in Figure 1.

The implementation of RSVP relies on a RSVP daemon residing on each host and RSVP-enabled router, which is shown in Figure 2. To make a resource reservation at a node, a RSVP reservation request is passed to two decision modules – admission control and policy control. The admission control module determines whether the node has sufficient available resource to support the request, and policy control module determines whether the user has sufficient privilege to make this reservation. If both decisions are positive, the corresponding parameters

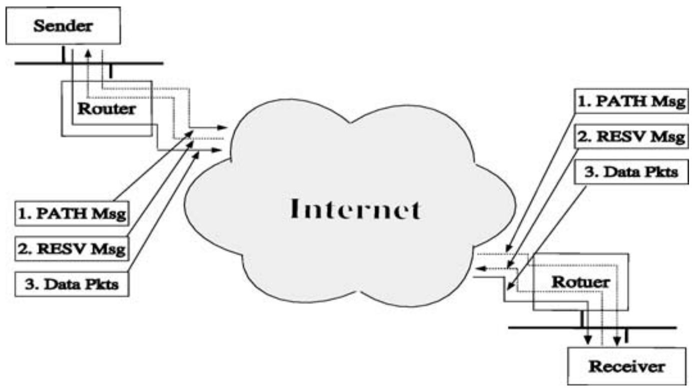



*Figure 1.* A simplified RSVP signaling process.

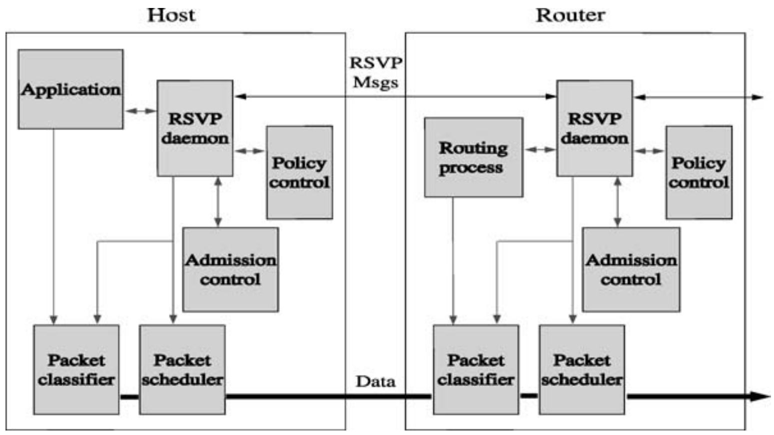



*Figure 2.* RSVP daemon in hosts and routers.

are set in packet classifier and packet scheduler modules to achieve the desired QoS. The packet classifier determines the QoS class for each packet, and the scheduler orders packet transmissions to achieve the promised QoS for each session.

### 2.2. SBM

Standard RSVP-based admission control does not work in the case of shared segments (LANs etc.) because of two reasons: (1) multiple networking devices share the capacity of the link; (2) a LAN, which is normally composed of layer-2/layer-1 networking devices (such as switches, bridges and hubs), is taken as one hop to a layer-3 router.

SBM [24] is proposed to resolve the above problem. SBM is a signaling protocol for RSVP-based admission control over IEEE 802-style networks. It describes the operation of RSVP-enabled hosts/routers and link layer devices (switches, bridges) to support reservation of LAN resources for RSVP-enabled data flow. Efficient admission control generally needs traffic control or priority queuing mechanisms at link layer. However, current LAN technologies are still a mix of legacy shared/switched LAN segments and newer switched segments based on IEEE 802. 1p specification [14], and such situations are not expected to have a big change in near future. Thus, SBM is designed to support both 802. 1p based LAN segments and legacy segments. In the former case, higher layer QoS specifications are mapped to MAC layer into 8 priority queues, LAN resources can then be managed to differentiate different flows that require different QoS services, providing an approximate emulation of RSVP-based resource reservation. In the latter case, where resources cannot be separated according to the requirement of data flows, SBM is still useful. For legacy LAN segments, the admission control mechanism only limits the total amount of traffic load imposed by RSVP-enabled flows on a shared LAN. It is expected that such a combination of admission control and the inherent rate adaptation mechanism of TCP/IP based best-effort traffic should avoid persistent traffic congestion.

In a LAN segment, the switches, bridges or routers may play the role of SBMs, which are capable of managing resources on a segment. However, only one Designated SBM (DSBM) manages the resources for a segment. The DSBM can be selected through an election algorithm from multiple SBMs. Whenever a DSBM client (a host or a router) wants to send a PATH/RESV message through a LAN segment, it will send the PATH/RESV message to the segment's

DSBM, instead of the RSVP session destination (for PATH message) or source (for RESV message) as it is done in a conventional RSVP processing. The DSBM uses a special MAC and IP address (a reserved multicast address) to listen to incoming requests from the DSBM clients. If the RSVP message is PATH, the DSBM builds and maintains a PATH state for the session and informs the previous layer 2 or layer 3 (we use L2 or L3 in the rest of this paper) hop who sent the PATH message, then forwards it to the next hop (another DSBM or a RSVP-enabled router). If the RSVP message is RESV, the DSBM does the admission decision and resource reservation (if capable) based on the bandwidth available. If sufficient resources are available and the reservation request is granted, the DSBM forwards the RESV message toward the previous hop recorded in its PATH state for the session. However, there is a problem with the above process. When the PATH message is sent to a L2 DSBM, the DSBM may not have the L3 routing information necessary to select the egress router (if multiple routers exist in that LAN segment) before forwarding the PATH message. To ensure correct operation and routing of RSVP message, additional forwarding information must be provided to DSBMs. For this purpose, a new RSVP object called LAN_NHOP is defined to keep track of the next L3 hop as the PATH message traverses an L2 domain between two L3 entities (RSVP-enabled hosts or routers). The LAN_NHOP object includes the IP address (LAN_NHOP_L3 address) as well as the corresponding MAC address (LAN_NHOP_L2 address) for the next L3 hop, in case the L2 DSBM does not have an ARP capability to determine the MAC address. In the example shown in Figure 3, there are three LAN segments in which the above process should be enforced. LAN segments A and B correspond to the access LAN segments and C is a LAN segment between two L3 RSVP-enabled routes.

The IEEE802. 1p defines a way for switches/bridges to differentiating among several "*user priority*" values encoded in packets representing different traffic classes [14]. The user priority values can be encoded either in native LAN packets (e.g., in IEEE 802.5's FC octet [16]) or by using and encapsulation above the MAC layer (e.g., the VLAN tag defined in 802.1q [15]). To utilize this mechanism for better resource management, SBM introduces a TCLASS (traffic class) object. Once a sender sends a PATH message, downstream DSBMs will insert a new TCLASS object in the PATH message that travels to the next L3 device (to some extent). The L3 device that receives the PATH message must remove and store the TCLASS object as part of its PATH state for the session. Later, when the same L3 device needs to forward a RESV

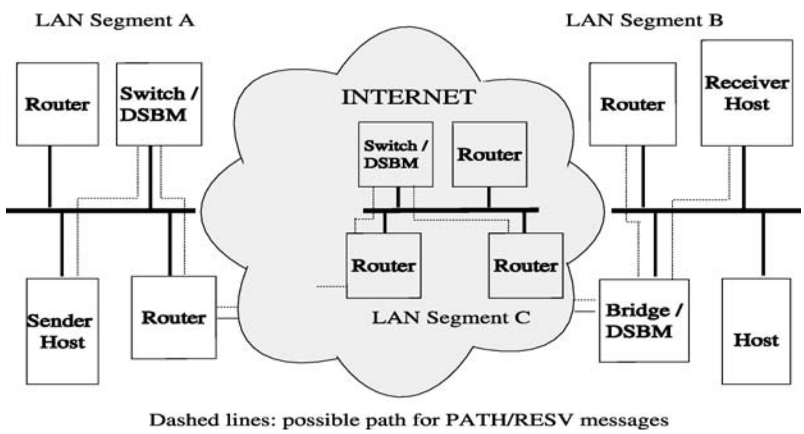



*Figure 3*. SBM protocol in an example.

message to-wards the original sender (the DSBMs), it must include the TCLASS object in the RESV message. When the RESV message arrives at the original sender, the sender must use the user-priority value from the TCLASS object to override its selection for the traffic class marked in outgoing packets.

## 3. Extending SBM to IEEE 802.11e WLANs – the Wireless SBM

In Section 1, we have discussed that in a heterogeneous IP network, which involves both wired links and wireless access segments, the wireless segments could introduce "gaps" in terms of resource reservation when we try to make an end-to-end RSVP-based QoS connection. To remove the wireless "gaps", we need: (1) an effective resource scheduling and management mechanism should be provided in the wireless segments; (2) we also need an appropriate QoS signaling, which is compatible with the RSVP/SBM QoS signaling, and transparent to the applications. We address the first challenge by investigating the resource scheduling and management ability provided by HCF Controlled Channel Access (HCCA) mode in the upcoming IEEE 802.11e standard [12], we discuss the ability of HCCA with a simple scheduler to show how to make parameterized QoS connections in WLANs. To solve the second problem, we need to address the issue of how to synchronize 802.11e signaling with RSVP/SBM protocol as well as the issue of QoS mapping between RSVP parameters and 802.11e QoS parameters. An end-to-end QoS network architecture containing Wireless SBM components is shown in Figure 4. In this figure, the Access Point plays the role as a DSBM for the WLAN segment. We extend the above discussions by focusing on this QoS architecture in the rest of this paper.

### 3.1. IEEE 802.11e MAC LAYER QoS SUPPORT

The IEEE 802.11 WLAN operates with two modes at MAC layer – a mandatory Distributed Coordination Function (DCF) and an optional Point Coordination Function (PCF). DCF is a contention-based protocol. In DCF, each wireless station has to sense the channel before

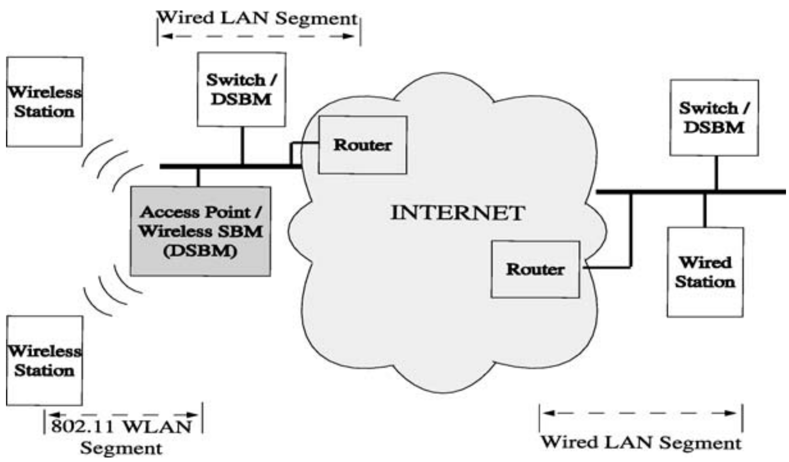



*Figure 4.* An end-to-end QoS architecture with wireless SBM.

sending its packet. If the channel has been idle for a DCF Inter Frame Space (DIFS), the wireless station waits for a back-off interval, which is a random number between 0 and Contention Window (CW), then transmits a packet. If a collision occurs during transmission, retransmission occurs repeating the above process. However the CW will be doubled. The CW is limited by a minimum value (CWMin) and a maximum value (CWMax). Each station working on DCF has only one queue, packets from different applications are served at MAC layer in FCFS (first come first serve) mode. Since all the stations share the-same MAC layer channel access parameters (DIFS, CWMin, CWMax), applications running on these stations have the same opportunities to access the channel, even though they may have different QoS requirements. Therefore DCF does not provide any QoS guarantee. PCF is a polling-based protocol, in which, a Point Coordinator (PC, generally residing on a AP) periodically polls the stations and allocates TXOPs (transmit opportunities) to each of them in a round robin fashion. Such a centralized mode is expected to utilize the channel more efficiently and provide QoS guarantee for time-sensitive applications. However, evaluations have shown its inefficiency and uncertainty [19, 20]. It is worthwhile noting that if the optional PCF is enabled, the period after each beacon transmission is divided into two sections, namely Contention Free Period (CFP) and Contention Period (CP). PCF only works in CFP.

To enhance the QoS support at the MAC layer, IEEE 802.11e introduces a Hybrid Coordination Function (HCF), which provides both contention and controll-based channel access mechanisms. The contention-based mechanism, called Enhanced Distributed Channel Access (EDCA),[1] is based on DCF; while the HCF Controlled Channel Access (HCCA) is based on PCF. EDCA differentiates Traffic Categories (TCs, defined to categorize the packets with different QoS Requirements) by providing multiple priority queues in each station. These queues are assigned different channel access parameters to achieve "prioritized" services. Packets in a higher priority queue wait for a shorter time to access the channel than packets in a lower priority queue. Such "prioritized" QoS provides good support for deploying DiffServ architecture. As with PCF, HCCA is also based on a poll-and-response protocol. However, to overcome inefficiency and uncertainty problems of PCF, HCCA has made some modifications to the frame definition and exchanging schemes. We discuss the details of HCCA in the following subsection.

#### 3.1.1. *The HCF Controlled Channel Access (HCCA) Mode and its Signaling*

To support QoS for time-sensitive applications, HCCA introduces a new concept called **Traffic Stream** (TS) to describe a QoS data flow (comparing to the TC defined in EDCA). The QoS requirements of a TS is described by TSPEC (traffic specification) parameters. A Traffic Identifier (TID, 4 bits) of the QoS Control field in each QoS data frame can be used to specify which TS the packet belongs to. Within a non-AP station, a TS can be uniquely identified by the TID; however, to identify a TS within the AP, we need to know not only the TID, but also the direction (from or to HC) and the address of the QoS station (QSTA). It is worthwhile noting that the TID values for HCCA mode does not tell anything about the QoS level. As previously mentioned, the QoS of a TS is specified by TSPEC.

In HCCA mode, whenever a QSTA needs to setup a TS through the Hybrid Coordinator (HC, which generally collocates within the AP), it will send a request with TSPEC information to the HC. If the traffic can be scheduled in the HCCA mode, the QSTA receives a downlink frame notifying the acceptance of the traffic, the scheduling information as well as the agreed

[1] In earlier versions of IEEE 802.11e draft, it is called EDCF (Enhanced DCF) [13].

TSPEC will also be included in that frame. A TS is then setup. The QSTA will be polled by the HC periodically through the CF Poll frames. In response to the CF_Poll frames, the QSTA sends several packets according to the TXOP limit information indicated in the CF-Poll frame. The TSPEC and the scheduling information are defined in TSPEC Element and Schedule Element respectively, and encapsulated in corresponding Management Action frames. Besides the TID field, the previously mentioned QoS Control field also provides some auxiliary information for the TS scheduling. For example, the HC may attach TXOP limit information to the QoS Control field in its CF_Poll frame, and QSTAs may inform their queue sizes and expected TXOP for the next poll to the HC through the QoS Control field in the QoS data frames. More information on the above frames used can be found in [12].

In contrast to PCF where the polling-based access happens only in Contention-Free Period (CFP), HCCA allows polling in both CFP and Contention Period (CP). This flexibility also helps HCCA to mitigate the inefficiency and uncertainty problems of PCF mode in supporting time-sensitive applications.

To support the HCCA mode operations, a corresponding signaling process is defined in 802.11e. The signaling process works relying on *primitives* inside each QSTA and *management QoS Action frames* between QSTAs (including the HC/AP). Some primitives may trigger the sending of appropriate action frames, while other primitives are triggered by receiving specific action frames. Figure 5 shows the signaling process of adding a TS in a HCCA -enabled BSS (Basic Service Set). It is always the responsibility of the non-AP QSTA to initiate the creation of a TS regardless of its direction. As demonstrated in Figure 5, when a non-AP QSTA needs to set up a TS between itself and the HC, the Station Management Entity (SME) of this QSTA internally sends a MLME-ADDTS, request primitive, which includes the TSPEC information to the MAC Layer Management Entity (MLME). The MLME then triggers the HCCA to send an ADDTS request QoS Action frame to the HC. As previously mentioned, a TSPEC element will be included in this frame. Receiving this frame at the HC triggers its MLME to send a MLME-ADDTS.indication primitive internally to its SME. Based on the TSPEC information contained in this primitive and available channel resources, the Admission Control Unit (ACU) of the HC will decide if this request should be accepted. If the decision is positive, it calculates a scheduling scheme based on current TSs and the newly added TS. The SME then

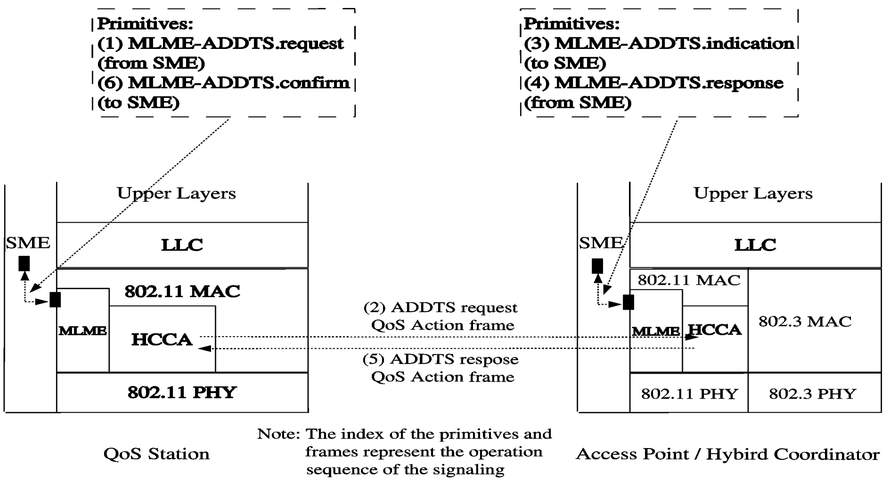



*Figure 5.* Signaling for adding a TS in 802.11e HCCA mode.

locally sends a MLME-ADDTS.response primitive, which contains the scheduling and agreed TSPEC information, to its MLME. This will trigger the HCCA of the HC to send an ADDTS response QoS Action frame to the original QSTA, the MLME of which will consequently send a MLME-ADDTS.confirm primitive to its SME. A TS will be set up, and the QSTA may start transmitting its packets once it is polled by the HC.

#### 3.1.2. *A Simple Scheduler for HCCA*

The resource scheduling and admission control process play a very important role in supporting "parameterized" QoS in HCCA mode. In this section, we present a simple scheduler proposed in 802.11e draft. As previously discussed, the HC does the admission control and scheduling based on the available channel resources and the TSPEC information coming with the TS request. To help understanding the scheduling process, we list the parameters of TSPEC Element as well as Schedule Element in Table 1. Since the admission control is based on the scheduling generation process, we introduce the latter first.

The schedule for a TS is calculated in two steps. The first step is the calculation of the scheduled Service Interval (SI), which is the interval between two successive TXOPs allocated to the same station. In the second step, the TXOP duration for a given SI is calculated for the stream. Figure 6 shows the structure of the superframe (combination of CFP and CP) used by the scheduler. The calculation of the SI is done as follows: first the scheduler calculates the minimum of all MSIs (Maximum Service Intervals) for all the stations, let this minimum

*Table 1.* TSPEC and parameters of a schedule element

| Parameters | Explanation |
|---|---|
| | TSPEC element parameters |
| Mean Data Rate ($\rho$) | Average bit rate for transfer of the packets. |
| Delay Bound (*D*) | Maximum delay allowed to transport a packet across the wireless segment. |
| Nominal MSDU SIze (*L*) | Nominal size of the packets. |
| Maximum MSDU Size (*M*) | Maximum size of the packets. |
| Maximum Burst Size (*MBS*) | Maximum size of the data burst size that can be transmitted at the peak rate. |
| Minimum PHY Rate (*R*) | Physical bit rate assumed by the scheduler for transmit time and admission control calculations. |
| Peak Data Rate (*PR*) | Maximum bit rate allowed for transfer of the packets. |
| | Schedule element parameters |
| Minimum TXOP Duration (*mTD*) | Minimum TXOP allocated to the QSTA. The *mTD* is equal to the maximum packet transmission time for any of the QSTA's TSPECs. The maximum packet transmission time of a TSPEC is the time required to send a packet of size M at the Minimum *PHY* Rate. |
| Maximum TXOP Duration (*MTD*) | Maximum TXOP duration allocated to the QSTA. The *MTD* is bounded by the transmission time of the aggregate maximum burst size. |
| Minimum Service Interval (*mSI*) | Minimum time between the start of successive TXOPs allocated to the QSTA. Given a service interval for each TSPEC (calcualted as $L/\rho$), the *mSI* contained in the service schedule is equal to the smallest service interval for any TSPEC that belongs to that QSTA. |
| Maximum Service Interval (*MSI*) | Maximum time allowed between the start of successive TXOPs allocated to the QSTA. It is related to the lowest delay bound (*D*) among the QSTA's TSPECs. |

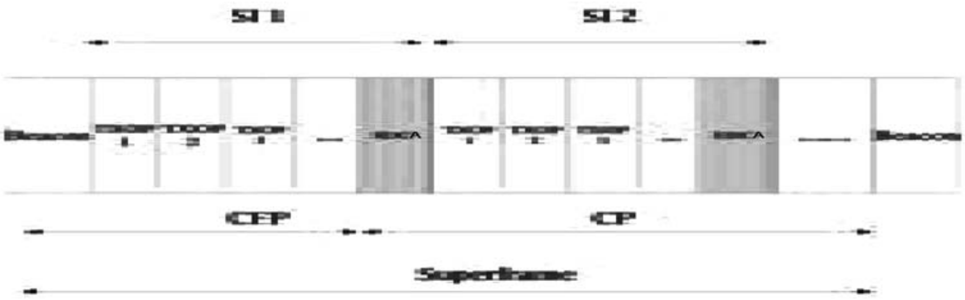



*Figure 6.* A simple scheduler for HCCA mode.

be “m”; second, the scheduler chooses a number lower than “m” that is a submultiple of the beacon interval. The SI is the same for all QSTAs, which should be recalculated every time a new TS is accepted. The TXOP for stream $i$ is calculated by the following formula:

$$\mathrm{TXOP}_i = \max\left(\frac{N_i \times L_i}{R_i} + O, \frac{M}{R_i} + O\right) \tag{1}$$

where $O$ is the overhead due to PHY and MAC headers, IFSs (Inter Frame Spaces), acknowledgement frames, and poll frames, $L_i$ and $R_i$ are the Nominal MSDU Size and Minimum PHY Rate for stream $i$ respectively (see Table 1), and $N_i$ is the number of MSDUs that arrived at the Mean Data Rate ($\rho$) during the SI:

$$N_i = \left\lceil \frac{\mathrm{SI} \times \rho_i}{L_i} \right\rceil \tag{2}$$

The Admission Control Unit (ACU) corresponding to the above scheduler is quite simple, it determines if a newly arriving stream can be admitted by checking if the following inequality is satisfied:

$$\frac{\mathrm{TXOP}_{k+1}}{\mathrm{SI}} + \sum_{i=1}^{k} \frac{\mathrm{TXOP}_i}{\mathrm{SI}} \leq \frac{T - T_{\mathrm{EDCA}}}{T} \tag{3}$$

where $k$ is the number of existing streams and $k + 1$ is used as index for the newly arriving stream, $T$ is the beacon interval and $T_{\mathrm{EDCA}}$ is the duration used by EDCA traffic.

A point worth noting is that the HC polls QSTAs instead of individual TSs. The HC adds up the TXOPs of all the TSs for one QSTA, and polls the QSTA according to this value. It is the responsibility of the scheduler residing on each QSTA to redistribute the aggregate TXOP to each TS in that QSTA. It is worth noting that the nature of wireless communications may preclude absolute guarantees to satisfy QoS requirements. However, in a controlled environment (e.g. no interference), the behavior of the scheduler can be observed and verified to be compliant with the service schedule. We present a performance evaluation of the above scheduler in Section 4.

### 3.2. Extend SBM to IEEE 802.11e WLANs-the Sesign Issues

In this section, we address the design issues of the Wireless SBM, focusing on the scenario illustrated in Figure 4, where a wireless station makes a parameterized QoS connection with a wired host crossing a 802.11e WLAN segment, several LAN segments, and the Internet. We assume the bandwidth of wired LAN segments is high enough that it is not likely to be a bottleneck for an end-to-end QoS resource reservation, so that we can focus on the WLAN segments. To integrate the signaling processes of RSVP/SBM and the 802.11e MAC signaling, we need to address at least two issues: (1) synchronization of lower layer 802.11e signaling with upper layer RSVP/SBM signaling process. (2) mapping and translation between RSVP TSPEC and 802.11e TSPEC. We discuss the first issue in Section 3.2.1, and the second issue in Section 3.2.2.

#### 3.2.1. *Synchronization of 802.11e MAC Signaling with RSVP/SBM*

As previously discussed, RSVP/SBM mainly utilizes the PATH/RES messages to complete a QoS reservation process, where RSVP/SBM daemons conduct the whole process of generating/reacting these messages. On the other hand, 802.11e WLAN relies on QoS related primitives and Action frames to make channel resource allocations, where SME initiates a TS request and makes final decisions on admission/scheduling for this request. Thus, to synchronize the 802.11e MAC signaling with RSVP/SBM, the first issue we should address is the communication between RSVP/SBM daemon and SME. We assume this communication can be implemented by some newly added primitives. Furthermore, the synchronization makes requirements related to timing, which should be reflected in the modifications needed for 802.11e signaling. We explain these new primitives and modifications along with the signaling process for two cases: QSTA as a sender and QSTA as a receiver.

**Case 1: QSTA as a sender.** In this case, a QSTA inside the 802.11 WLAN initiates a parameterized QoS reservation. The end-to-end signaling process should be as follows (Figure 7 shows the primitives added to support Wireless SBM for this case):

1. The RSVP daemon at the QSTA sends a PATH message to the DSBM residing on the AP/HC along with its Sender_TSPEC. At the same time, it also invokes a SME-ADDTS.request

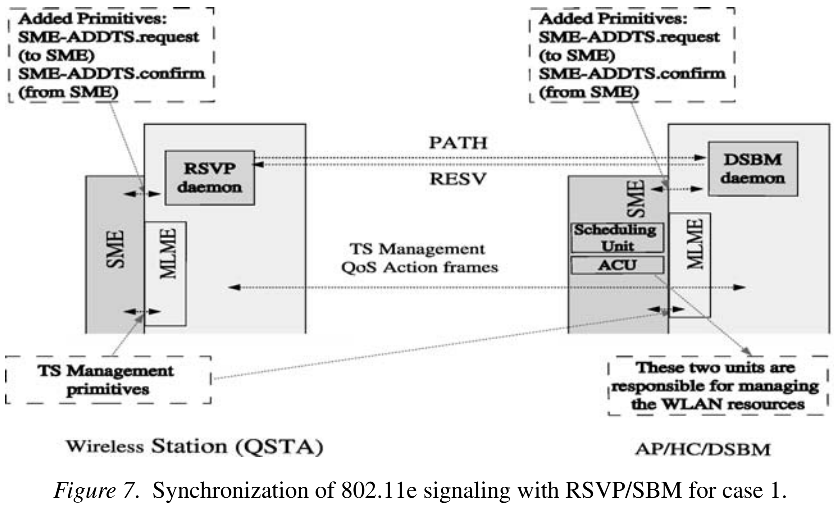



*Figure 7.* Synchronization of 802.11e signaling with RSVP/SBM for case 1.

primitive to its local SME. The primitive includes the Sender_TSPEC information, which will be translated by the SME into 802.11e TSPEC parameters. The SME then sends a MLME-ADDTS.request primitive including the 802.11e TSPEC information to the QSTA's MLME, and subsequently invokes an ADDTS request frame to be sent to the AP/HC/DSBM.
2. The AP/HC/DSBM gets the PATH message and forwards it to the next DSBM or router in the wired network following the rules discussed in Section 2. In parallel, the AP/HC/DSBM gets the ADDTS request frame, which consequently invokes its local MLME to send a MLME-ADDTS, indication primitive to its SME. In contrast to the standard 802.11e signaling, after receiving this primitive, the SME will store the 802.11e TSPEC information (here we refer to it as TSPEC-1) coming with the MLME-ADDTS.indication primitive and then waits there for appropriate primitives (to be explained in step 4) from the DSBM daemon, instead of instantly performing admission and scheduling.
3. The PATH message will finally be delivered to the receiver (a wired node in this case). If the receiver decides to accept this reservation request, it sends an RESV message along with the agreed TSPEC information to the DSBM of which LAN segment it resides on. If the DSBM decides to admit this request, it makes resource reservation and forwards the RESV message to the next upstream DSBM or router, which performs similar functions.
4. When the AP/HC/DSBM's daemon gets the RESV message, it sends a SME-ADDTS.request primitive to its SME who is waiting for this primitive. The SME translates the TSPEC information contained in the primitive into 802.11e TSPEC parameters (TSPEC-2), and then makes the admission decision based on available resources and requested resources, which is equal to min(TSPEC $-$ 1, TSPEC $-$ 2). It sends a SME-ADDTS.confirmation primitive that includes its decision to the DSBM daemon. If the decision is positive, the DSBM forwards the RESV message to the original sender (the QSTA). In parallel, the SME also sends a MIME-ADDTS.response primitive, containing the admission decision and scheduling information, to the MLME, which further triggers the MAC entity to send an ADDTS.response frame to the QSTA.
5. The QSTA's RSVP daemon finally gets the RESV message, and also receives a SME-ADDTS-confirmation primitive sent by the local SME. This primitive is triggered by receiving the MLME-ADDTS-confirmation primitive sent from the MLME. So far, a parameterized end-to-end QoS reservation is set up.

It is worthwhile noting that according to the standard 802.11e signaling, a timer is set after a QSTA sends an ADDTS request frame. The timer will expire if an ADDTS response frame is not received after a timeout value. To support the above signaling interaction, we need to set a large timeout value for the timer, because the SME of AP/HC/DSBM has to wait the RESV message before sending back the ADDTS response frame to the QSTA.

**Case 2: QSTA as a receiver.** In this case, it is the wired host that initiates a reservation process.

1. When the AP/HC/DSBM's daemon gets the PATH message initiated by the wired host, it sends a SME-ADDTS.request primitive to its local SME. The SME then waits there for a MLME-ADDTS.indication primitive from the MLME. The DSBM daemon forwards the RESV message to the QSTA.
2. After the QSTA's RSVP daemon gets the PATH message forwarded by the AP/HC/DSBM, it sends a SME-ADDTS.request primitive to its local SME. Following the standard 802.11e signaling, an ADDTS request QoS Action frame will be sent to AP/HC/DSBM.

3. The AP/HC/DSBM gets the frame, triggering the MLME to send a MLME-ADDTS. indication primitive to the SME. The SME executes the admission and scheduling process based on available resources and the requested resources, which is the minimum of TSPEC-1 (defined in the SME-ADDTS.request primitive) and TSPEC-2 (defined in the MLME-ADDTS.indication primitive). The SME then sends the MLME-ADDTS.response primitive back to MLME, which triggers the MAC entity to send an ADDTS response frame to the QSTA. The SME of the AP/HC/DSBM also sends a SME-ADDTS.confirm primitive to the DSBM daemon.
4. After receiving the ADDTS response frame, the QSTA's SME sends a SME-ADDTS. confirm primitive to its RSVP daemon. If the admission decision contained in this primitive is positive, it sends a RESV message to the AP/HC/DSBM, which then further forwards the RESV message to the next DSBM or router.

We only address the process of connection-setup (as shown above), the signaling for tearing down a connection is similar to the setup process, except that some DELTS primitives need to be used.

3.2.2. *QoS mapping*

The RSVP TSPEC describes the characteristics of data flows using a token-bucket model from the view of the IP layer, compared to 802.11e which specifies the resource requirements of a Traffic Stream from the view of the MAC layer. As shown previously, a QoS mapping process is essential for the interaction of RSVP/SBM protocol with 802.11e signaling. We defined such mapping by deriving the 802.11e TSPEC parameters from those for RSVP/SBM. Some parameters of 802.11e TSPEC can be directly mapped from RSVP/SBM TSPEC, while others have to be given by SME, based on the MAC layer parameters and status, as well as the information directly from the applications (note that SME is the logical management entity which can communicate to all the layers). We list the RSVP/SBM and 802.11e TSPEC parameters as well as their mapping relations in Table 2. The explanation of 802.11e TSPEC parameters has already been given in Table 1.

It is worthwhile noting that it is hard to define an accurate one-to-one mapping without losing or skewing any information, considering those parameters are from different layers and defined using different models. We do not intend to give a complete one-to-one mapping for all the RSVP and IEEE 802.11e HCCA parameters. Instead, we present the mapping in the Table 2 as an example.

## 4. Performance Evaluations

To evaluate our proposal, two experiments were conducted: (1) we simulated the 802.11e HCCA mode to evaluate its ability to provide parameterized QoS; (2) we are currently establishing an experimental testbed to investigate the signaling process discussed previously for extending RSVP/SBM to 802.11e WLANs. The testbed mainly includes a Linux-based AP and a RSVP-enabled router as well as several wireless and wired stations. We modified the driver of the DCF-based wireless cards (HCCA-based cards are out of market at this time) to emulate the signaling presented in this work. In following text, we only present the result of the evaluation on the ability of 802.11e HCCA mode to provide parameterized QoS.

We used Berkeley NS V2.17b [21] to implement 802.11e HCCA mode. We modified the Stanford's EDCA+HCCA code [10] to support the scheduler discussed in Section 3.1.2 (the

*Table 2.* The mapping of RSVP TSPEC to 802.11e TSPEC

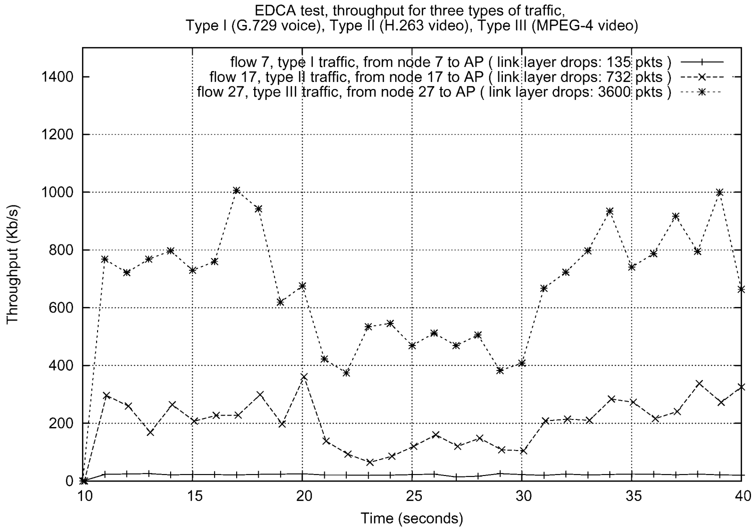
| RSVP TSPEC parameters and their definitions | 802.11e TSPEC Parameters and their mappings from RSVP TSPEC Parameters |
|---|---|
| Token Rate (r): the continually sustainable bandwidth requirements for a flow, in bytes/s. | Mean Data Rate ($\rho$): $= r * \frac{L}{L-\text{Heads\_Size}} * 8$, in bits/s, where Heads_Size refers the overhead introduced by TCP/UDP/IP protocols. |
| Peak Rate ($p$): the maximum send rate a flow, in bytes/s. | Peak Data Rate (PR): $= p * \frac{L}{L-\text{Heads\_Size}} * 8$, in bits/s. |
| Token-bucket Depth ($b$): the extent to which the data rate can exceed the sustainable average for short periods of time, in bytes, more precisely, the amount of data sent cannot exceed $r * T + b$ (where $T$ is any time period). | Maximum Burst Size (MBS): $= \frac{b}{p-r} * p$, in bytes. |
| Maximum Packet Size (MPS): the size of the biggest packet, in bytes. | Maximum MSDU Size (M): $=$ min(MPS + Heads_Size, 2346, Frag_Threshold), in bytes, where Frag_Threshold is the fragmentation threshold set at 802.11 MAC layer. |
| Minimum Policed Size (maps): the size of the smallest packet generated by the sending application, in bytes. | Not applicable. |
| Not applicable | Nominal MSDU Size (L): cannot be mapped from RSVP TSPEC. SME can give this parameter based on the information from applications. |
| Not applicable | Delay Bound (D): cannot be mapped from RSVP TSPEC. SME can give this parameter based on the information from applications. |

original scheduler implemented in Stanford's code is quite different from this one). Furthermore, the Stanford's code does not separate the EDCA transmission and HCCA transmission, the packets of a data flow are concurrently transmitted both with EDCA mode and HCCA mode. To evaluate the performance of HCCA, we separate the HCCA transmission from the EDCA transmission. Once one TS is accepted by HC, the packet of the TS will be transmitted only in HCCA mode.

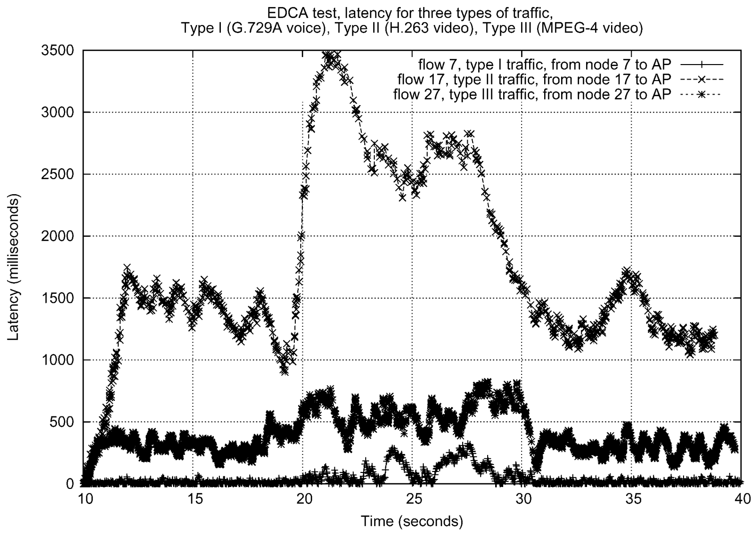
The scenario we considered includes 36 non-AP QSTAs and one AP/HC. All the traffic is sent from non-AP QSTAs to the AP/HC. 10 QSTAs send VolP stream, 10 QSTAs send H.263 video stream, and 8 QSTAs send MPEG-4 video traffic. All the above traffic sources start at 10.0 s and ends at 40.0 s. We also let another 8 QSTAs send MPEG-4 traffic sources, starting at 20.0 s and ending 30.0 s, to create varying loads on the WLAN. The VolP traffic is modeled using ITU-T G.729A codec [18], which generates 60-byte messages (40 bytes overhead is introduced by RTP/UDP/IP assuming header compression is used) periodically with an interval of 20 ms yielding a bit rate of 24 Kbits/s. For the H.263 video traffic, we used a trace of H.263 video stream encoded for the movie – Jurassic Park I [9], which produces VBR traffic with average bit rate of 390 Kbits/s. The MPEG-4 video traffic is modeled as CBR traffic, which sends a 1200-byte message every 8 ms yielding a bit rate of 1200 Kbits/s. We conducted simulations in EDCA mode with the same scenario described above. With EDCA, VolP, H.263 video, MPEG-4 video are transmitted with high, medium, and low priorities respectively. The channel access parameters of EDCA and the main TSPEC parameters used for HCCA scheduler are listed in Table 3. For both EDCA and HCCA simulations, we assume

*Table 3.* EDCA parameters and HCCA TSPEC parameters

| Traffic type | Channel access parameters for EDCA tests | | | TSPEC parameters used in HCCA tests | | |
|---|---|---|---|---|---|---|
| | Cumin | CWMax | AIFS | Max. SI (ms) | Nominal MSDU size (bytes) | Mean date rate (bits/s) |
| G.729A Audio | 7 | 31 | 43 $\mu$s | 50 | 60 | 24000 |
| H.263 Video | 31 | 63 | 43 $\mu$s | 50 | 1200 | 390000 |
| MPEG-4 Video | 63 | 1023 | 52 $\mu$s | 100 | 1200 | 1200000 |




(a) Throughput

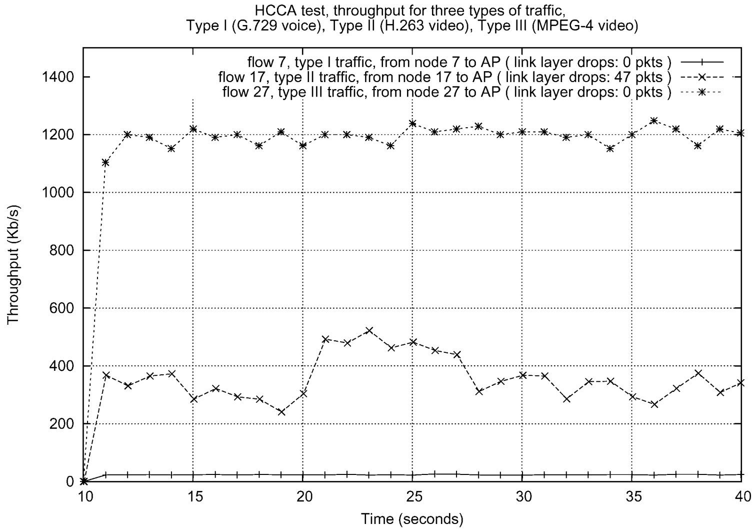



(b) Latency

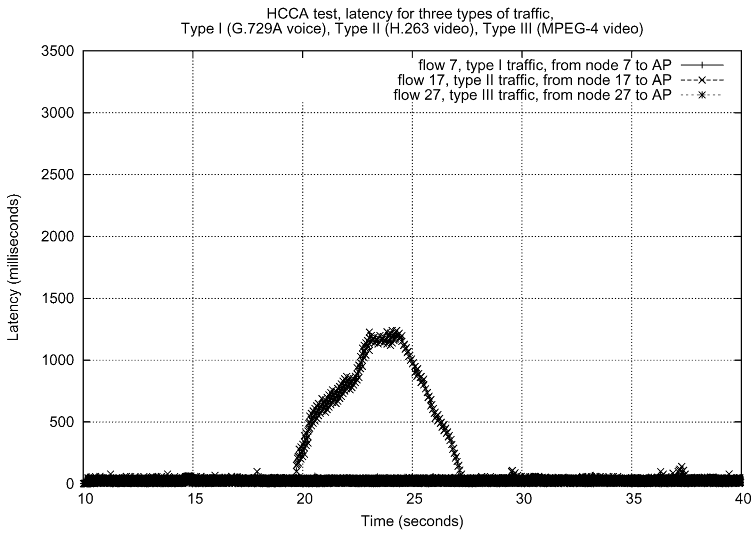
*Figure 8.* Throughput and latency for EDCA tests.

that all the nodes (including the AP) operate at IEEE 802.11a PHY mode-6 with bandwidth of 36 Mbits/s and we use the same MAC/PHY parameters as listed in [25].

The throughput and latency results for EDCA and HCCA tests are illustrated in Figures 8 and 9 (data flows with same traffic type have similar results, we selected data flows sending from node 7, 17, and 27 to be presented in the Figures). For EDCA test, there are many packets dropped, especially during 20.0 s–30.0 s, where the load of the network is higher than other times. The latency of H.263 and MPEG-4 video are quite high throughout the whole simulation period. Although the G.729A VoIP traffic has a low latency during 10.0 s–20.0 s and 30.0 s–40.0 s, it increases to unacceptable values (≥150 ms) during 20.0 s–30.0 s. While for the HCCA test, there are no dropped packets and the latency is quite low (around 50 ms) for all the data flows except the H.263 traffic, which has 47 packets dropped and sometimes




(a) Throughput




(b) Latency

*Figure 9.* Throughput and latency for HCCA tests.

has a high latency. This result shows that HCCA with the scheduler discussed in Section 3.1.2 performs well for CBR traffic, but not optimal for VBR traffic. The reason of its inefficiency for VBR traffic is that the TXOP calculated by the scheduler for the VBR traffic is a fixed value based on the Mean Data Rate, while VBR has inherent bursty characteristics which cause a number of packets remaining in the queue after polled by the HC. One way to solve this problem could be by using the Peak Rate instead of the Mean Data Rate to calculate the TXOP for VBR traffic. However, this may cause wasted resources. To avoid the wasting of resources, a good scheduler for VBR flows should be able to adapt to the bursty traffic. Efficient adaptions may require the scheduler to have an accurate model of VBR flows or be able to predict the traffic to some degree. Designing a scheduler that is efficient both for CBR and VBR traffic will be our future work.

## 5. Conclusions

In this work, we propose Wireless SBM to extend the RSVP/SBM to IEEE 802.11 WLANs, which enables end-to-end parameterized QoS over hybrid wired-wireless networks. We present our proposal from two aspects: (1) first, we investigate and evaluate the ability of IEEE 802.11e HCCA mode in supporting parameterized QoS; (2) second, we address the design issues on the synchronization of IEEE 802.11e HCCA signaling with RSVP/SBM signaling. The QoS mapping between RSVP and IEEE 802.11e HCCA TSPEC parameters is also discussed. We have used simulations to evaluate the HCCA mode of IEEE 802.11 WLANs on its ability to provide parameterized QoS. Our simulation reveals that, for CBR traffic, parameterized QoS can be achieved in an IEEE 802.11 WLAN with a simple HCCA scheduler. However, it remains a challenge to design a HCCA scheduler which is efficient for both CBR and VBR traffic so that parameterized QoS can be provided while simultaneously achieving optimal resource management. Our ongoing research efforts also include further investigating the proposed signaling process of Wireless SBM on experimental testbeds.

## References


1. W. Almesberger et al., “A Prototype Implementation for the IntServ Operation Over DiffServ Networks”, in *Proceedings of IEEE Globecom 2000*, San Francisco, December 2000.
2. D. Awduche et al., “RSVP-TE: Extensions to RSVP for LSP Tunnels”, *RFC* 3209, Dec. 2001.
3. A. Banerjee et al., *Generalized Multiprotocol Label Switching: An overview of Routing and Management Enhancements, IEEE Communications Magazine*, Vol. 39, No. 1, pp. 144–150, Jan. 2001.
4. L. Berger, ED., “Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource Reservation Protocol – Traffic Engineering (RSVP-TE) Extensions”, *RFC* 3473, Jan. 2003.
5. Y. Bernet et al., “A Framework of Integrated Services Operation Over DiffServ Networks”, IETF RFC 2998, November 2000.
6. S. Blake et al., “An Architecture for Differentiated Services”, *RFC* 2475, Dec. 1998.
7. R. Braden et al., “Integrated Services in the Internet Architecture: An Overview”, *IETF RFC* 1633, June 1994.
8. R. Braden et al., “Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification”, *IETF RFC 2205*, September 1997.
9. F. Fitzek et al., “MPEG-4 and H.263 Video Traces for Network Performance Evaluation”, *IEEE Network*, Vol. 15, No. 6, pp. 40–53, November/December 2001.
10. P. Garg et al., “Using IEEE 802.11e MAC for QoS over Wireless”, in *Proceedings of IEEE IPCC’03*, Phoenix, USA, April 2003.

11. IEEE 802.11 WG. "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", *IEEE 802.11 Standard*, 1999.
12. IEEE 802.11 WG, "Draft Supplement to Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY)-specifications: MAC Enhancements for Quality of Service (QoS)", *IEEE 802.11e/Draft* 8.0, Feb. 2004.
13. IEEE 802.11 WG, "Draft Supplement to Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: MAC Enhancements for Quality of Service (QoS)", *IEEE 802.11e/Draft 4.2*, July 2003.
14. IEEE 802.1 WG, "IEEE 802.1D-1998, Part3: Media Access Control (MAC) Bridges (incorporating IEEE 802.1p: Traffic Class Expediting and Dynamic Multicast Filtering)", *IEEE Standard 802. ID*, 1998.
15. IEEE 802.1 WG, "IEEE 802.1Q-1998, Virtual Bridge Local Area Networks", *IEEE Standard 802.IQ*, 1998.
16. IEEE 802.5 WG, "IEEE 802.5-1998, Part 5: Token Ring Access Method and Physical Layer Specification", *IEEE Standard 802.5*, 1998.
17. IETF Integrated Services Working Group, *http://www.ietf.org/html.charters/intserv-charter.html.*
18. ITU-T, "G.729 Annex A: Reduced complexity 8 Kbits/s CS. ACELP speech codec", *ITU-T Recommendation G. 729 Annex A*, November 1996.
19. S. Mangold et al., "IEEE 802.11e Wireless LAN for Quality of Service, in *Proceedings of European Wireless*, February 2002.
20. Q.Ni et al., "QoS Issues and Enhancements for IEEE 802.11 Wireless LAN", *INRIA Research Report*, No. 4612, November 2002.
21. The Network Simulator – ns-2, *http://www.isi.edu/nsnam/ns.*
22. E.C. Rosen et al., "Multiprotocol Label Switching Architecture", *RFC 3031*, Jan. 2001.
23. S. Salsano et al., "Supporting RSVP in Differentiated Service Domain: An Architecture Framework and a Salability Analysis", in *Proceedings of IEEE ICC'99*, Vancouver, Canada, June 1999.
24. R. Yavatkar et al., "SBM (Subnet Bandwidth Manager): A Protocol for RSVP-based Admission Control over IEEE 802-style networks", *IETF RFC 2814*, May 2000.
25. L. Zhang and S. Zeadally, "A Framework for Efficient Resource Management in IEEE 802.11 WLANs", *Wireless Communications and Mobile Computing* (WCMC) Journal, John Wiley & Sons, Special Issue on Emerging WLAN Applications and Technologies, Vol. 4, No. 8, December 2004.

**Liqiang Zhang** is currently an Assistant Professor in the Department of Computer and Information Sciences at Indiana University South Bend. He received his Ph.D in Computer Science (2005), MS degree in Electrical Engineering (1996), and BS degree in Computer Science (1993), respectively, from Wayne State University, China Academy of Telecommunications Technology, and China University of Mining and Technology. During 1996–2000, He worked as a research engineer and project leader in Datang Telecommunication Technology & Industry Group.

Dr. Zhang's research interests include Wireless Networks and Mobile Computing, Quality of Service and Real-time Computing, High-speed networking, Wireless Sensor Networks, and Network Security. He is a member of IEEE Communication Society, and ACM SIGCOMM.

**Sherali Zeadally** received the B.A. and M.A. degrees in computer science from University of Cambridge, Cambridge, U.K., and the Ph.D. degree in computer science from University of Buckingham, Buckingham, U.K., in 1996.

In March 1996, he joined the Department of Electrical Engineering at the University of Southern California (USC), Los Angeles, where he was an Assistant Professor until August 1999. While at USC, he was also one of the Principal Investigators of the Integrated Media Systems Center (IMSC), an NSF Engineering Research Center, where he led the networking research group from 1996 to 1999. He joined the Department of Computer Science at Wayne State University, Detroit, MI, as an Assistant Professor in August 1999. He is the Founder and Director of the High-Speed Networking Laboratory and the computer networking research group at Wayne State University. His research in computer networking has been funded by NSF and several companies, including HP, Microsoft, Compaq, Altheon, IBM, Sun Microsystems, and others. He currently serves on the editorial boards of several international journals. His research interests include high-speed networks, quality of service, operating systems, wireless and personal are networks, middleware, and performance analysis of network systems.

Dr. Zeadally is the recipient of the Outstanding Teaching Awards (in 2001 and 2003) from the College of Science at Wayne State University, the highest award given by the College of Science for excellence in teaching. He is a Fellow of the British Computer Society.