MMMP: A MAC Protocol to Ensure QoS for Multimedia Traffic over Multi-hop Ad Hoc Networks

Sunil Kumar*, Mahasweta Sarkar*, Supraja Gurajala**, and John D. Matyjas***

Abstract: In this paper, we discuss a novel reservation-based, asynchronous MAC protocol called 'Multi-rate Multi-hop MAC Protocol' (MMMP) for multi-hop ad hoc networks that provides QoS guarantees for multimedia traffic. MMMP achieves this by providing service differentiation for multi-rate real-time traffic (both constant and variable bit rate traffic) and guaranteeing a bounded end-to-end delay for the same while still catering to the throughput requirements of non real time traffic. In addition, it administers bandwidth preservation via a feature called 'Smart Drop' and implements efficient bandwidth usage through a mechanism called 'Release Bandwidth'. Simulation results on the QualNet simulator indicate that MMMP outperforms IEEE 802.11 on all performance metrics and can efficiently handle a large range of traffic intensity. It also outperforms other similar state-of-the-art MAC protocols.

Keywords: *Medium Access Control, MAC, Ad Hoc Networks, Multi Hop, QoS, Multimedia*

1. Introduction

The rapid development in wireless technology has seen the advent of many wireless devices emerging over the past several years. Along with the phenomenal growth in technology, more and more functionalities have been squeezed into these devices. Thus a wireless node today has the capability of acting as a transmitter, a receiver and a relay. A self-organizing collection of such wireless devices, without the support of a central controller (or base station), is referred to as an *Ad Hoc Network*. The ad hoc network should automatically detect and seamlessly induct any new nodes, and automatically reconfigure if any node moves out of the network. The absence of a central controller requires that the nodes should communicate with each other in a peer-to-peer mode. The network consisting of mobile nodes is termed as a MANET, which stands for a mobile ad hoc network. All nodes in an ad hoc network have a transceiver, an antenna, and a power source. The features of these nodes can vary widely in terms of size, processing power, transmission range, and battery power [1-3]. Wireless, being a tightly controlled medium, has '*limited channel bandwidth*'. Furthermore, the factors such as multiple access, fading signals, noise and interference can significantly lower the *effective throughput* [1-3].

MAC layer, sometimes also referred to as a sub-layer of the 'Data Link' layer, involves the tasks and actions that are necessary for data transfer between different nodes in the network. Since the MAC layer has a direct bearing on how reliably and efficiently data can be transmitted between two nodes along the routing path in the network, it affects the Quality of Service (QoS) of the network. Some of the QoS-related parameters that may be quantified are: end-to-end delay, available bandwidth, probability of packet loss, etc. [1, 4-7].

Real-time consumer applications such as streaming audio/video require a reserved share of the channel capacity over relatively long durations so that QoS requirements are met [3, 14-16, 19]. However, the changing network topology, severely power constrained environment and intrinsically unreliable transmission medium in ad hoc networks poses several problems that hinder provisioning for a high QoS [1, 3]. In fact, maintaining QoS guarantees for delay sensitive traffic is quite difficult because obtaining a consistent network-wide distributed snapshot of the state of the queues and the channel at individual nodes at any given instant is an intractable problem [1, 3]. However, stringent delivery guarantees, particularly on short time scales, need not always be fulfilled for such applications. Therefore, these applications can be satisfied by soft or dynamic QoS provisioning [22-23].

The absence of a centralized coordinator in these networks rules out the use of IEEE 802.11 Point Coordination Function (PCF) mode of operation [8, 20-21,

Corresponding Author: Sunil Kumar

 * Electrical and Computer Engineering, San Diego State University, San Diego, CA 92182 USA({skumar, msarkar2}@mail.sdsu.edu)

 ** Computer Science, Clarkson University, Potsdam, NY 13699 USA (sgurajala@yahoo.com)

*** Air Force Research Laboratory, Rome, NY 13441 USA (john.matyjas@rl.af.mil)

23-25]. Existing asynchronous MAC layer support for multimedia traffic includes protocols like Enhanced Distributed Coordination Function (EDCF) [25], Black Burst [7], Multiple Access Collision Avoidance with Piggyback Reservation (MMACA/PR) [9] and its QoS enabled improvement, called as Modified MACA/PR (or MMACA/PR) in this paper [10]. Some other MAC protocols for multimedia QoS are discussed in [15-19].

In this paper we propose MMMP (Multi-rate Multi-hop MAC Protocol) - a reservation based asynchronous MAC protocol, especially for multimedia traffic, over a multi hop Ad Hoc network. Our protocol provides QoS for real time traffic flows and delivers bounded end-to-end delay for such traffic without starving the non real time datagram packets. Our protocol employs the RTS-CTS (Request-To-Send – Clear-To-Send) mechanism to reserve transmission slots. The RTS-CTS mechanism allows neighboring nodes who hear these transmissions to update their "Reservation Table" so that they refrain from transmitting when another transmission is ongoing. In addition, we ensure flow priority (and thus Service Differentiation) by allowing the RTS packets of a real time flow with higher priority to attain the channel with higher probability than another competing lower priority real time flow. We use a Distributed Priority Scheduling (DPS) scheme [11-12] with suitable modifications to achieve this. We also ensure bandwidth conservation by preemptively "dropping" certain packets that are "useless" to the network (e.g., due to expiry of their time-to-live or header packets) via our "Smart Drop" feature. We incorporate efficient bandwidth usage by "borrowing" unused available bandwidth from an existing flow and "giving" it to a new flow, which is in need of bandwidth. We refer to this mechanism as our "Release Bandwidth" feature. We discussed a preliminary version of this protocol in [26].

The rest of the paper is organized as follows: We describe our protocol in Section 2, our simulation model in Section 3, the results and analysis of the simulations in Section 4, and finally conclude the paper in Section 5.

2. Proposed Scheme – Overview

Our protocol comprises of five modules: a DCF-like asynchronous MAC scheme, a scheduling scheme, a resource reservation scheme, a 'Smart Drop' feature and a 'Release Bandwidth' feature.

2.1 Basic MAC in MMMP

As shown in Figure 1, the time is divided into discrete units called *slots*. A fixed duration of time comprising of several slots constitutes a *cycle*. A *session* comprises of data and associated control packets exchange for a given flow between two nodes. In other words, a session is the time duration over which RTS/CTS/DATA/ACK or DATA/ACK transmissions occur between two nodes. There can be several sessions in a cycle. The scheme is as follows: The first packet in a *real time flow* uses a RTS/CTS/DATA/ACK dialogue exchange between the transmitting and receiving nodes. This exchange reserves bandwidth required for the subsequent packets of the flow. As a result, subsequent packets of the flow use the DATA/ACK exchange in the reserved time slots. On the contrary, every *non-real time datagram* packet uses the RTS/CTS/DATA/ACK dialogue as in the IEEE 802.11 DCF protocol [8].
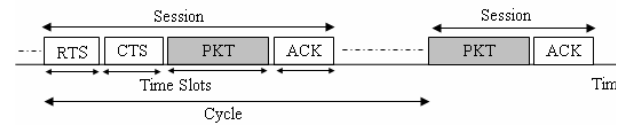
Fig. 1. Real-time and Non-real time flow transmission [3, 7]

The RTS frame of a real-time flow contains information like the transmission duration of CTS, DATA and ACK, source and destination address, flow ID and priority, the minimum and maximum bandwidth requirement of the flow and a session number. Note that every flow can have either one or more transmissions (i.e., sessions) in one cycle. The CTS frame also contains all the above information with the only exception of the transmission duration, which comprises of the time required to transmit the DATA and ACK frames. The ACK frame contains all the above information in addition to the information on available bandwidth and the priority of the next flow that the source node will be transmitting next. The DATA frame header consists of all the information that an ACK packet carries along with a sequence number for the DATA packet and the reserved bandwidth information that the current flow occupies.

Unlike MMACA/PR [10], our scheme realizes that different flows with varying packet generation rates require different amounts of bandwidth. Bandwidth is calculated in terms of the number of sessions a flow requires in one cycle. We use a Resource Reservation (RR) scheme to enable flows to have multiple sessions in one cycle, based on their bandwidth requirement. Every node in the network maintains a reservation table with the following two fields:

Receive Reservation Field (RRF) — keeps track of the sessions in which the neighboring nodes are scheduled to receive.

Transmit Reservation Field (TRF) — keeps track of the sessions in which the neighboring nodes are scheduled to transmit.

Before transmitting an RTS packet, the sender checks its reservation table for an empty chunk of time big enough to transmit RTS/CTS/DATA/ACK packets. If that is available, it sends an RTS packet. On receiving the RTS packet, the receiver checks its reservation table and transmits CTS if it has an empty time chunk that can accommodate CTS/DATA/ACK transmissions. If the source does not receive CTS it will back off for a while and retransmit RTS.

For a real-time flow, we use a simple binary back-off algorithm, where the back-off duration is static (unlike in 802.11 where it is doubled on every collision). Non-real time traffic uses binary exponential backoff (BEB) scheme as in IEEE 802.11 [8], where backoff duration is doubled on every collision based on the flow priority. This mechanism allows the network to prioritize real time multimedia traffic over delay tolerant data traffic.

If the RTS/CTS handshake is successful, the DATA packet is transmitted by the source, which also contains information for the next data packet (using piggyback). On receiving the DATA packet, the receiver and nodes that overhear DATA, calculate the next scheduled transmission time of the ensuing data packet of this flow and record it in the RRF of their reservation table. The next scheduled transmission window for that neighboring node is calculated as $[(t(n) + t(ct) - t(pkt) - PIFS), (t(n) + t(ct) + SIFS)]$, where $t(n)$ is the instant of time when the packet is heard, $t(ct)$ is the "cycle" interval, $t(pkt)$ is the fixed packet transmission duration, and *PIFS* and *SIFS* are the IEEE 802.11 standard Inter Frame Space times [4]. The receiver then sends an ACK packet.

On receiving the ACK, the source and nodes that overhear the ACK, calculate the next scheduled transmission time for the ensuing packet of the current flow and record this in the TRF of their reservation table. The next scheduled receiving window for that neighboring node is calculated as $[(t(n) + t(ct) - PIFS - t(pkt) - SIFS - t(ack)), (t(n) + t(ct))]$, where $t(ack)$ is the transmission duration for an ACK packet. On successful RTS/CTS/DATA/ACK transmission, the real-time flow has a session reserved in the following cycle, and thus can transmit a DATA/ACK during its corresponding session. A similar procedure is followed by other real-time flows to reserve their sessions.

The reservation table is updated in every cycle as illustrated in Figures 2 and 3 below. The reservation table carries information on the start and end time of the transmission, the back-off duration, the source and destination address, the flow ID of the transmission, the end time and flow ID of the previous reservation and a session number that keeps track of the session for a flow (since flows can reserve multiple sessions per cycle) in addition to the receive and transmit reservation fields.
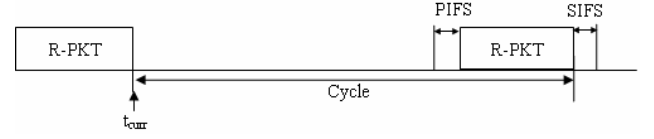
Fig. 2. Renewing reservation window in Transmitter Reservation Table. The start and end times of the next scheduled transmission for the packet are $t_{curr} + cycle - t_{pkt} - PIFS$ and $t_{curr} + cycle + SIFS$, respectively. Here R-PKT represents a real-time flow packet. t_{curr} is the current time, PIFS is point coordination function inter-frame spacing slot, t_{pkt} is the transmission duration of R-PKT, and SIFS is short inter-frame spacing slot.
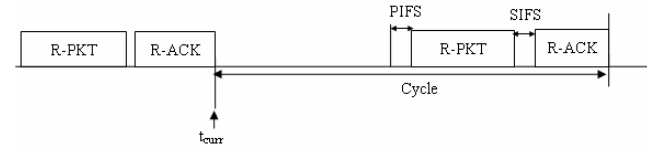
Fig. 3. Renewing reservation window in Receiver Reservation Table. The start and end times of the next scheduled transmission for the packet are $t_{curr} + cycle - t_{pkt} - PIFS - SIFS - t_{ack}$ and $t_{curr} + cycle$, respectively.

2.2 Resource Reservation (RR)

Depending on the packet generation rate, a real-time flow/burst may require more than one session in a cycle to transmit its packets. Hence we have introduced a resource reservation scheme as in [13]. When a node receives a new flow, the minimum and maximum bandwidth requirements (*MinBW* and *MaxBW*, respectively) are assigned based on its traffic generation rate. We define bandwidth of a particular flow in terms of the number of sessions that the flow requires in one cycle. Each node maintains a '*flow_session* table' to keep track of the session's usage and the following flow information: flow_ID, priority, destination address, amount of minimum, maximum and actually allocated bandwidth for the flow, the amount of bandwidth soft reserved and hard reserved for the flow, the contention-window value for the flow, the transmission end time for the last packet sent in this flow, the number of sessions the previous node has for this flow and two other information that are used by the "release bandwidth" feature, namely the number of sessions that are to be temporarily released for new flows and the time after which the hard reserve expires.

The RR scheme is as follows. Let us consider a flow that initiates at source node 'S' and travels through an intermediate node 'I' to a destination node 'D'. On

receiving the real-time flow from the upper layer, node 'S' calculates its available bandwidth (*AvaBW_s)* and updates its flow_session table. It then transmits RTS to node 'I' (all RTS packets will include the *MinBW* and *MaxBW* information of the flow). On receiving RTS, the intermediate node calculates its available bandwidth (*AvaBW_i)* and updates its flow_session table. It then sends CTS to node 'S' which is followed by DATA from node 'S' to node 'I'. Node 'I' transmits an ACK back to node 'S'. Node 'I' then transmits an RTS to node 'D'. On receiving the RTS, node 'D' calculates its available bandwidth (*AvaBW_d)* for this flow and sends a CTS to node 'I'. Node 'I' then transmits DATA to node 'D'. On successful DATA reception, node 'D' transmits an ACK, along with *AvaBW_d*. Upon reception of this ACK, Node 'I' sets its *AvaBW_i* to *min {AvaBW_d, AvaBW_i}*. The new *AvaBW_i* is conveyed to node 'S' along with an ACK packet transmitted by node 'I' in response to the next successful reception of a DATA packet from node 'S'. Node 'S' updates its *AvaBW_s* to *min {AvaBW_s, AvaBW_i}*. Hereafter this common available bandwidth value is referred to as *AvaBW*. This sequence of communications results in the end-to-end reservation of a session from source to destination nodes (called *end_to_end_session*). The new sessions can be initiated by the source only if the number of on going *end_to_end_sessions* is less than *AvaBW*. This enables MMMP to sequentially reserve multiple sessions from source to destination as constrained by *AvaBW*.

The above procedure ensures that the intermediate node does not start a second session (to the destination node) for the same flow, unless there is a second session reserved from source to intermediate node. This avoids control packet overhead and bandwidth wastage that is possible in scenarios where the number of sessions between the intermediate to destination nodes are more than the number of sessions between the source and intermediate nodes.

2.3 Scheduling

Even though MMMP is a reservation based scheme, the first packet in the real-time flow and all datagram packets need to contend for the channel. In order to provide QoS amongst the real-time flows a *priority-based scheduling* is required. We have used the distributed priority scheduling scheme with suitable modifications [11-12]. While details of the scheduling algorithm are beyond the scope of this paper, we briefly present its overview below.

Each node locally gathers and stores scheduling information based on overheard information and incorporates estimate of its traffic's priority into its MAC. In particular, each packet has an associated "priority index" which can be computed with purely local information (e.g.,

user priority and time-to-live). When a node issues a RTS, it piggybacks the priority index of its current packet. Nodes that overhear this RTS will update their database appropriately. If the node is granted CTS, it includes the priority index of its next head-of-line (higher priority) packet in the DATA packet that it transmits. Overhearing nodes again update their database with this information. Each neighboring node can then assess the priority of its own head-of-line packet in relation to the list of other head-of-line packets whose information it would have stored in its database. With this packet-priority information, nodes re-evaluate their back-off values. Thus, the low priority flows defer their transmission for a longer period of time to enable channel reservation for the higher priority flows. This assures service differentiation based on packet priority. This information can be exploited via a minor modification in prioritized backoff schemes in IEEE 802.11, to closely approximate a "global" dynamic priority schedule in a distributed way.

2.4 Smart Drop

A multimedia traffic flow (such as video) is composed of frames. Each frame may be encoded into fixed sized packets. The first packet in a frame (or a group of frames, GOP) may contain frame or GOP header information. Similarly, the first frame in a video GOP is an intra (I) coded frame and certain number (such as 10 frames in MPEG-4) of subsequent predicted (P or B) frames depend on it. Thus, when the header packet or the packets of I-frame are dropped or lost, the remaining predicted frames may be of no use and their transmission would result in wasted bandwidth. To minimize such bandwidth wastage due to partial frame loss, we have introduced a *Smart Drop* feature as discussed below.

The rule is to check and see if the header packet is dropped and if it is, then the whole frame is dropped. Similarly, if the packets of an I-frame in a GOP are already dropped, the subsequent P- and B-frame packets in the GOP are also dropped. We assign a longer TTL (Time-to-live) and more channel access retry limit (say, 5) for header packets compared to non-header packets (having normal retry limit, say 3). If even after the normal retry limit, the I-frame packets cannot be sent, the packets with smaller Time-To-Live (TTL) in the frame are dropped.

2.5 Release Bandwidth

When a network is heavily loaded, a new real time flow cannot acquire bandwidth (BW) regardless of its priority. In order to support a new real time flow even in a congested network, we designed a "Release Bandwidth (RB)" algorithm. In RB, if a new real-time flow is unable to acquire BW, it borrows any excess BW that may be

Table 1. Simulation Parameters (here, CW represents Contention Window)

Parameter	Value	Parameter	Value
Channel Rate (in Mbps)	11	Min CW size for non real time flows	31
Packet Size (in Bytes)	512	Max CW size for non real time flows	1023
Slot Time	20μs	Min CW size for real time flows	10
Cycle Time	32 ms	Max CW size for real time flows	100
Short Spacing (SIFS)	10μs	Time_to_Live (TTL) for data packets	256 ms
Medium Spacing (PIFS)	30μs	TTL for Header packets	266 ms
Long Spacing (DIFS)	50μs	Total Simulation time	40 sec

available amongst existing flows. A node's excess bandwidth is calculated as follows:

For all existing flows whose priority is greater than or equal to the new flow's priority:
Acceptable BW = [(*MinBW* + *MaxBW*) / 2]
Excess BW = ⌊Number of Sessions - Acceptable BW⌋

For all existing flows whose priority is lower than the new flow's priority:
Excess BW = Number of sessions - *MinBW*;

The node then temporarily releases all the excess bandwidth sessions in the next cycle to enable the new flow to contend for the channel. In every transmitting node where session(s) has to be released, the source adds a variable to the data header informing the data's recipient, and all overhearing nodes, about the temporary release of this session in next cycle. All receivers, and the overhearing nodes, then record these sessions. Every receiving node where the session(s) is released adds a new variable to the ACK headers in order to inform the intended receivers of these ACKs and the overhearing nodes about release of this session in the next cycle. After the sessions are released, in a subsequent cycle the source node of the new flow will contend for the channel in released sessions until the RTS/CTS/DATA/ACK transmissions are successful. If the new flow succeeds in reserving a session for its first hop, the intermediate nodes follow the same procedure until the flow has a session reserved from source to destination.

3. Simulation Setup

To evaluate the performance of our MMMP scheme, we compare it with IEEE 802.11 DCF [8] and MMACA/PR [9, 10]. We also compare various versions of our basic MMMP scheme with each other. Our basic MMMP scheme is essentially the MMACA/PR scheme, which we refer to as MMMP_B in our simulations. The other versions of our scheme are MMMP_S (i.e. MMMP with priority scheduling but without the smart drop feature),

MMMP_SD (i.e., MMMP with Smart drop feature but without the priority scheduling feature), MMMP_S_SD (i.e., MMMP with Scheduling and with Smart drop features). Note that MMMP_S_SD is our proposed MAC protocol in its entirety. We compare and contrast different versions of our scheme because each version with its associated feature is effective for different traffic types. For example, the smart drop feature might be very useful for real-time multimedia application but not for non real time data traffic. Thus our simulation was designed to study the effectiveness of each of the features individually and when used collaboratively with each other. We used QualNet (version 3.7) as our simulation platform. To isolate the effectiveness of the MAC protocol from associated QoS routing protocol, we consider the case of stationary nodes alone. We simulate an ad hoc network topology (IEEE 802.11 IBSS). We have generated two types of traffic: CBR and VBR. CBR refers to Constant Bit Rate traffic, where the application generates fixed size packets of 512 bytes at regular intervals. VBR refers to Variable Bit Rate traffic, where the application generates packets of fixed size (512 bytes) in bursts at irregular time intervals. We run the simulations for *three network topologies* with a mix of both CBR and VBR real time traffic as discussed below. Each traffic flow is assigned a priority and a particular number of sessions based on its data rate.

- *Light topology* (in Figure 4) has 4 CBR and VBR flows, with up to 2 packets per cycle for each flow and priority from 1 to 5. These flows are independent of each other as each node is transmitting or receiving only one flow.
- *Moderately heavy topology* (in Figure 5) has 5 CBR and VBR flows, with up to 2 packets per cycle for each flow and priority from 1 to 8. Some of these flows are not independent of each other as certain nodes are transmitting more than one flow.
- *Heavy topology* (in Figure 6) has 7 CBR and VBR flows with up to 2 packets per cycle for each flow and priority from 1 to 8. Some of these flows are not independent of each other as certain nodes are transmitting more than one flow.

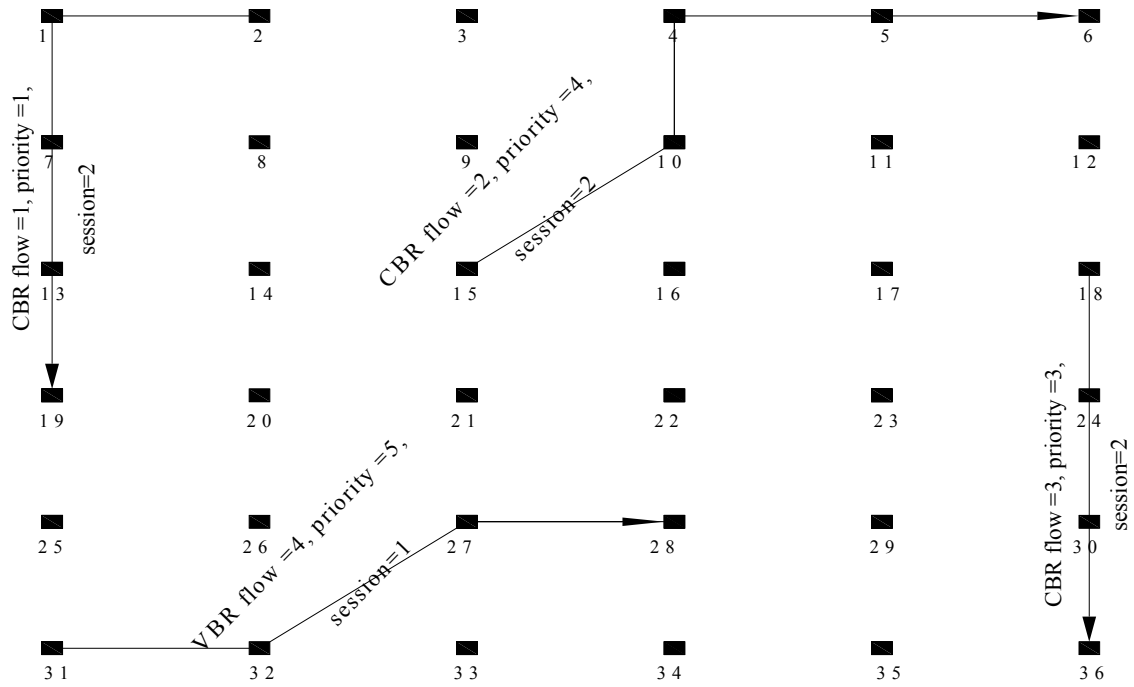A detailed list of parameters used in our simulations is shown in Table 1 above.

Fig. 4. Light Load Topology depicting flow directions, priority of flow and number of sessions per cycle.
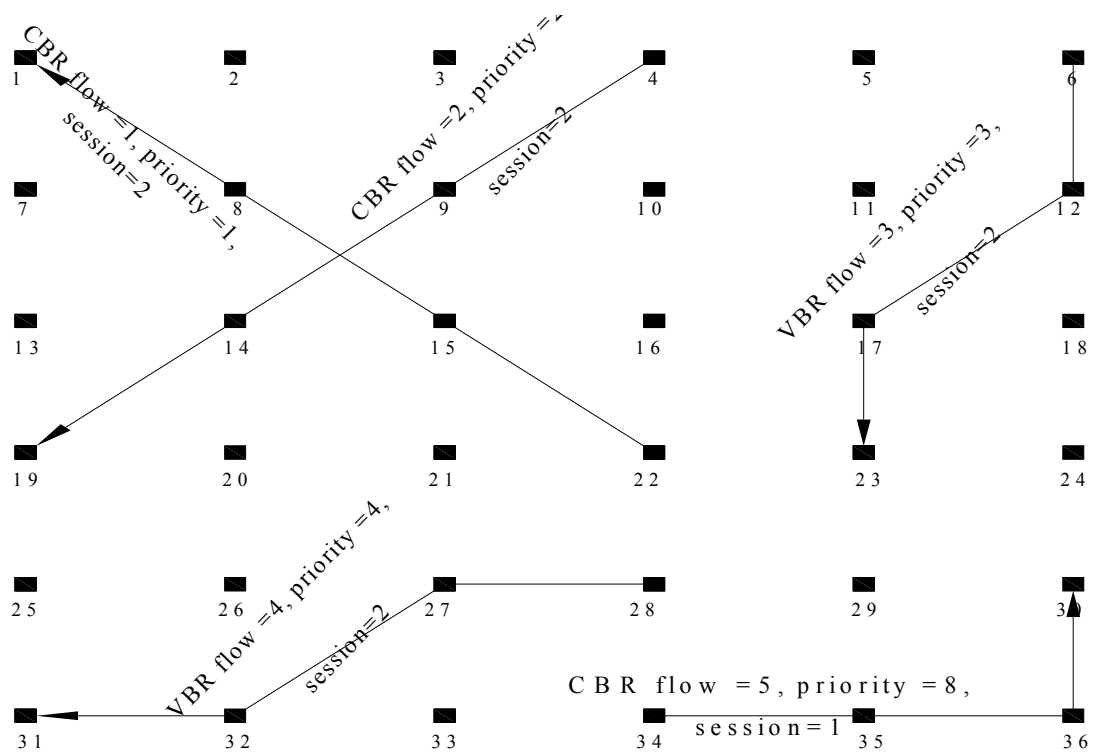
Fig. 5. Moderately Heavy Load Topology depicting flow directions, priority of flows and number of sessions per cycle.
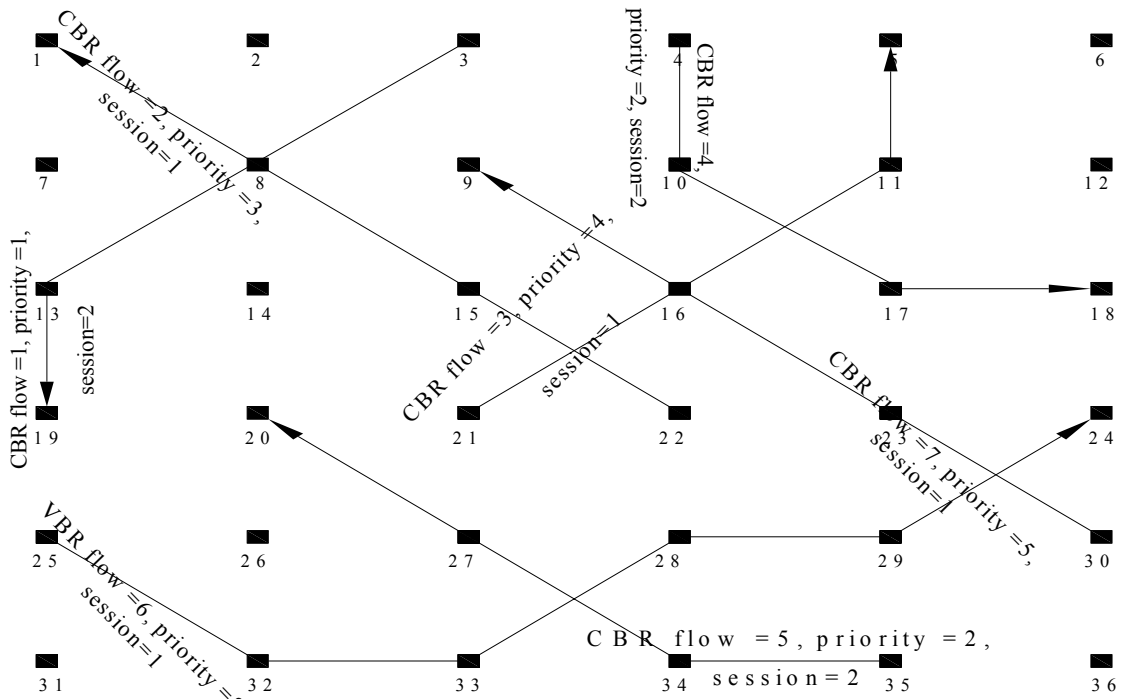
Fig. 6. Heavy Load Topology, depicting flow directions, priority of flows and number of sessions per cycle.

4. Results and Analysis

The five schemes namely MMMP_B, MMMP_S, MMMP_SD, MMMP_S_SD, and IEEE 802_11 are tested for different performance metrics as described below. It is crucial to understand that our MAC layer protocol is geared towards providing QoS guarantees mainly to multimedia traffic. Such traffic is delay sensitive.

4.1 Throughput, Packet and Frame Loss

Overall throughput is calculated as the total number of bytes (including RTS and CTS control packets) transmitted in the network in one second. The overall throughput of each scheme for light, moderately heavy and heavy topologies as obtained from the simulation is represented in Figure 7. We observe that the throughput for IEEE 802.11 compares poorly with the other schemes for all test loads. Since the time-to-live for a real-time flow packet is set to be only 8 cycles in our simulation, IEEE 802.11 cannot transmit the packet from source to destination within that time. Hence a lot of packets are dropped. IEEE 802.11 therefore does not perform well for real time traffic in multi-hop ad hoc networks.

Figure 8 shows the average throughput per node for different proposed schemes under varying traffic loads. Although the overall throughput is higher for heavy load, the per node throughput is lower due to congestion.

In Figure 9, we see that the number of frames dropped

for the MMMP_SD and MMMP_S_SD schemes are much lower than the other two schemes, which do not have smart drop feature implemented in the protocol. This is because MMMP_SD and MMMP_S_SD schemes prohibit transmission of packets, which will eventually not result in forming a "meaningful" frame. Thus the packets that do get transmitted are all worthwhile packets, which eventually are packaged into meaningful frames at the receiver. On the other hand, schemes without the smart drop feature (MMMP_B and MMMP_S) transmit higher number of packets but not all of them are useful at the receiver in constructing meaningful frames.

4.2 End-to-End Packet Delay

The end-to-end delay is the total time that a packet takes to travel from source to its final destination. The average, minimum and maximum end-to-end delays of different schemes for three topologies are shown in Table 2. The IEEE 802_11 scheme has the lowest end-to-end delay as it does not use reservation and need not wait for the session (Figure 10). As a result, the nodes can transmit packets whenever they get the channel. Even though IEEE 802_11 has low end-to-end delay, it does not result in high performance, due to its low throughput (as shown in Figure 7). The average per-hop delays experienced by the packets for all the schemes at varying traffic loads are shown in Figure 11. The average end-to-end delay increases with traffic load and number of hops that a flow has to travel.

Note here that the packets of 2 of the 4 flows in light topology travel 4 hops whereas packets of all the 5 flows in moderately heavy topology travel only 3 hops. This is why, the delay in Figure 10 is slightly lower for moderately heavy topology as compared to light topology.

The schemes with scheduling feature (MMMP_S and MMMP_S_SD) are seen to have a larger end-to-end and per-hop delay when compared to the other two schemes (MMMP_B and MMMP_SD). This is because of the large delay for the first packets in the low priority flows, which have higher back off.
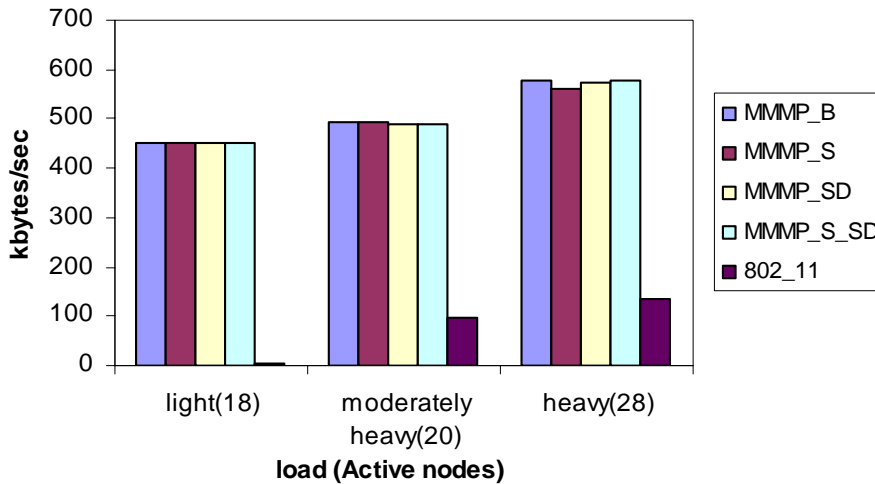
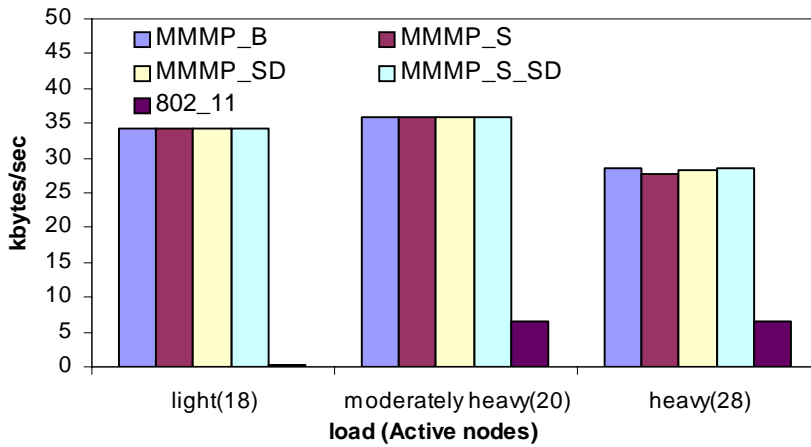**Fig. 7.** Average overall throughput for different schemes under varying traffic loads.

Fig. 8. Average per node throughput for different schemes under varying traffic loads.
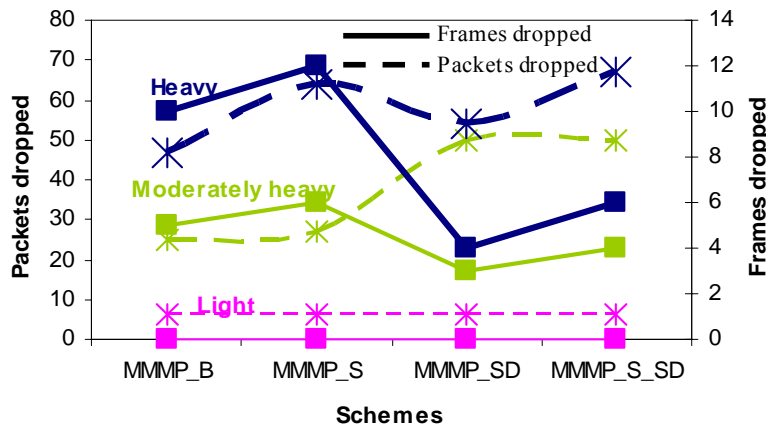
Fig. 9. Packets dropped (dashed lines) and frames dropped (solid lines) under different load conditions for different schemes.

Table 2. Average, minimum, and maximum delays for different schemes and loads (ms)

| MMMP_B	Average	Minimum	Maximum	Std_dev
Light	34.39902	26.47215	40.80645	13.5000
Moderately heavy	30.45752	19.54277	38.01835	10.0000
Heavy	35.4442	7.864876	56.15934	9.0000
MMMP_S				
Light	33.79584	24.7585	40.34241	12.9000
Moderately heavy	32.60819	26.42604	37.01802	11.2000
Heavy	41.84611	11.75028	70.16559	16.3000
MMMP_SD				
Light	34.39902	26.47215	40.80645	13.5000
Moderately heavy	31.22192	19.54277	39.70936	10.4000
Heavy	35.47819	5.780356	53.30997	9.8000
MMMP_S_SD				
Light	33.79584	24.7585	40.34241	12.9000
Moderately heavy	32.66995	26.25981	38.36659	11.2000
Heavy	47.0685	11.68281	80.47328	21.8000
802.11				
Light	4.772958	3.541082	6.717626	
Moderately heavy	4.626736	3.621309	7.460626	
Heavy	5.773868	3.427051	17.00382	

4.3 Control Packet Overhead

Table 3 shows the values of overhead experienced due to control packets as a percentage of the total transmitted data for our proposed schemes under varying traffic loads. As is evident from the table, our schemes have much lower overhead as compared to the IEEE 802.11. This overhead does not increase even when we use more sophisticated scheduling and smart drop features. Similarly, overhead does not vary significantly with traffic load.
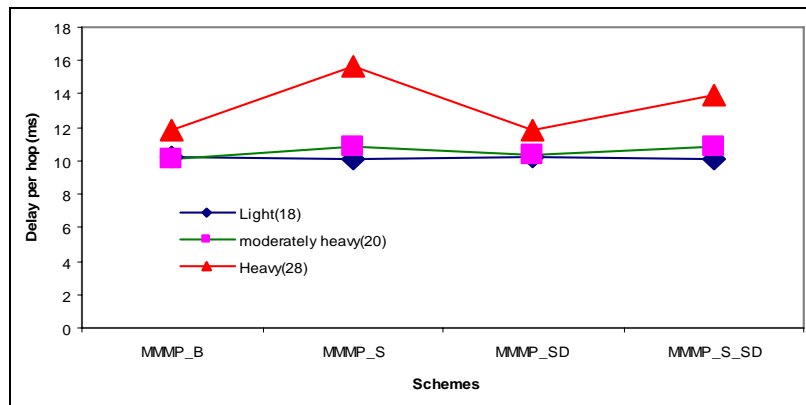
Table 3. Control Packet overhead

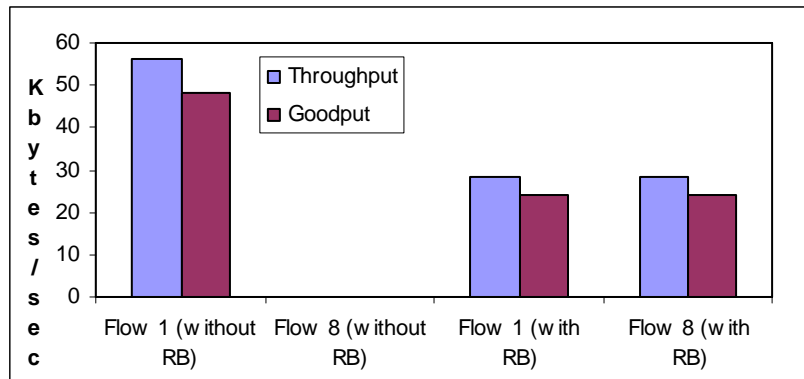|	Light	Moderately Heavy	Heavy
MMMP_B	1.298	1.176	1.480
MMMP_S	1.297	1.177	1.484
MMMP_SD	1.298	1.178	1.481
MMMP_S_SD	1.297	1.181	1.483
802_11	4.26	5.50	5.24

4.4 Release Bandwidth Effect

To simulate the effect of release bandwidth feature
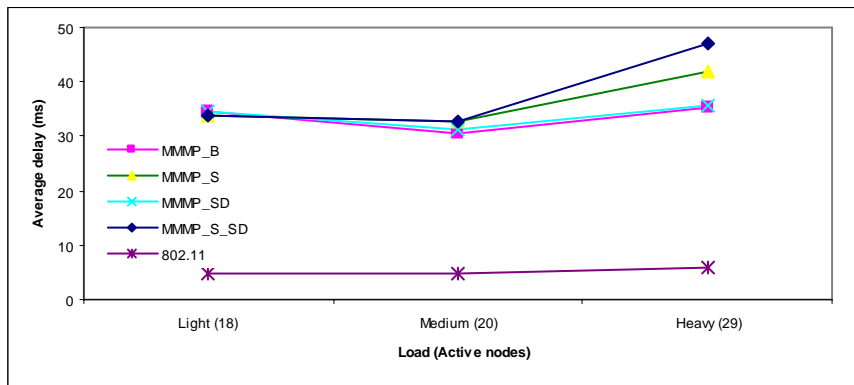
Fig. 10. Average end-to-end delay for various schemes.

Fig. 11. Average per hop delay for various schemes.

Fig. 12. Throughput and goodput for heavily loaded network using the 'Release Bandwidth' feature

(discussed in Section 2.5), a new flow (Flow #8 between node 3 and node 14 via node 8)) is added to the heavy topology with MinBW =1, MaxBW = 1 and priority =2. This topology is tested on the MMMP_B scheme with the Release Bandwidth (RB) feature added to it. In order to jumpstart the effect of RB feature we increased the network load to its maximum. The nodes 3, 8 and 13 release 1 session (out of 2 sessions) of Flow #1 to accommodate the new flow. Here Flow #2 is not affected as it has higher priority and only 1 reserved session.

As expected, the release bandwidth feature enables the new flow (Flow #8) to share the bandwidth with the existing flow (Flow #1). As seen in Figure 12, Flow #8 had zero throughput without RB feature but gained considerable throughput when the RB feature was enabled. Figure 12 also depicts the "goodput" value for each of these flows. Here, we define goodput as the total amount of raw data (i.e., payload) transmitted in one second in the network. This data does not include headers added at different protocol layers.

5. Conclusions

In this paper we introduced a new MAC protocol called Multi-rate Multihop MAC Protocol (MMMP) that provides QoS for multimedia traffic over an ad hoc network. MMMP is a reservation based asynchronous scheme that provides different QoS support for different multimedia traffics based on their QoS requirements. The Smart Drop feature incorporated in MMMP ensures efficient use of bandwidth while the 'Release Bandwidth' feature targets at maximizing bandwidth utilization.

A range of simulations were executed to test the MMMP performance and the efficacy of its features under varying network load conditions. Our proposed scheme performed much better than IEEE 802.11 in terms of throughput for all traffic load conditions. The simulation results point out

that the performance of four versions of the proposed scheme (with and without scheduling and smart drop) is largely invariant for light and moderately heavy loads. This suggests that the overhead due to the scheduling algorithm can be avoided for such load types. However in heavily loaded networks, for performance measures such as throughput, average delay, number of packets and frames dropped, the overall MMMP scheme with both smart drop and scheduling is seen to be the best when compared with a pre-existing MAC protocol like MACA/PR which is referred to as MMMP_B in our simulation results.

References

[1] S. Kumar, V. S. Raghavan and J. Deng, "Medium Access Control Protocols for Ad-Hoc Wireless Networks: A Survey", *Elsevier Ad-Hoc Networks Journal*, Vol. 4(3), pp. 326-358, May 2006.

[2] R. Jurdak, C. V. Lopes and P. Baldi, "A Survey, classification and comparative analysis of medium access control protocols for ad hoc networks", *IEEE Communications Surveys and Tutorials,* Vol. 6(1), pp. 2-16, 2004.

[3] I. F. Akyildiz, J. McNair, L. C. Martorell, R. Puigjaner and Y. Yesha, "Medium access control protocols for multimedia traffic in wireless networks," *IEEE Network*, Vol. 13, July/August 1999, pp. 39-47.

[4] P. Mahapatra, J. Li and C. Gui, "QoS in Mobile Ad Hoc Networks", *IEEE Wireless Communications*, June 2003.

[5] D. Perkins and H. D. Hughes, "A survey on quality-of-service support for mobile ad hoc networks," *J. Wireless Communication and Mobile Computing*, Vol. 2, 2002, pp. 503-513.

[6] S. Chakrabarti and A Mishra, "QoS Issues in Ad Hoc Wireless Networks", *IEEE Commun. Mag.*, Vol. 39(2), February 2001.

[7] J. L Sobrinho and A. S. Krishnakumar, "Quality of Service in ad hoc carrier sense multiple access networks," *IEEE J. Selected Areas Commun.*, Vol. 17(8), pp. 1353-138, 1999.

[8] "IEEE Standard Information Technology – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (IEEE Std 802.11-1999)", June 2003.

[9] C. R. Lin and M. Gerla, "MACA/PR: An Asynchronous Multimedia Multihop Wireless Network", *Proc. IEEE INFOCOM*, March 1997.

[10] Z. Ling, A. L. Ananda and L. Jacob, "A QoS Enabled MAC Protocol for Multi-hop Ad Hoc Wireless Networks," *Proc. IEEE International Performance Computing and Communication Conf.*, April 2003.

[11] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi and E. Knightly, "Distributed Priority Scheduling and Medium Access in Ad Hoc Networks," *Wireless Networks,* 8, 455-466, 2002.

[12] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi and E. Knightly, "Ordered Packet Scheduling in Wireless Ad Hoc Networks: Mechanisms and Performance Analysis." *Mobihoc'02*, Switzerland.

[13] J. Xue, P. Stuedi and G. Alonso, "ASAP: An Adaptive QoS Protocol for Mobile Ad Hoc Networks", *Proc. 14th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Beijing, China, Sept. 2003.

[14] S. Armenia, L. Galluccio, A. Leonardi and S. Palazzo, "Transmission of VoIP traffic in multihop ad hoc 802.11b networks: experimental results" *Proc. 1st International Conf. Wireless Internet*, 2005.

[15] T. B. Reddy, John P. John and C. SivaRamaMurthy, **"**Providing MAC QoS for multimedia traffic in 802.11e based multi-hop ad hoc wireless networks" *Journal of Computer Networks: The international Journal of Computer and Telecommunications Networking*, Volume 51(1), 2007, pp.153-176.

[16] Y. Yang and R. Kravets, "Distributed QoS Guarantees for Realtime Traffic in Ad Hoc Networks," *Proc. 1st IEEE Int'l Conf. Sensor and Ad Hoc Comm. and Networks (SECON),* June 2004.

[17] A. Rao and I. Stoica, "An Overlay MAC Layer for 802.11 Networks," *Proc. IEEE Conf. Mobile Systems(MobiSys),* 2005.

[18] H. Wu, Y. Liu, Q. Zhang and Z.-L. Zhang, "SoftMAC: Layer 2.5 Collaborative MAC for Multimedia Support in Multihop Wireless Networks," *IEEE Trans. Mobile Computing,* vol. 6(1), pp. 12-25, Jan. 2007.

[19] S.-T. Sheu and T.-F. Sheu, "A bandwidth allocation/sharing/extension protocol for multimedia over IEEE 802.11 ad hoc wireless LANs," *IEEE J. Sel. Areas in Commun.*, Vol. 19(10), Oct. 2001, pp. 2065-80.

[20] N. H. Vaidya, S. Bahl and S. Gupta, "Distributed fair scheduling in a wireless LAN," *6th Annual Intl. Conf. Mobile Computing and Networking*, Boston, August 2000.

[21] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz and L. Stibor, "IEEE 802.11e wireless LAN for quality of service," *Proc. European Wireless*, Florence, Italy, 2002.

[22] D. Thomson, N. Schult and M. Mirhakkak, "Dynamic Quality-of-Service for Mobile Ad Hoc Networks," *MobiHoc 2000, Boston, Massachusetts, USA.*

[23] A. Veres, A. T. Campbell, M. Barry and L. H. Sun, "Supporting Service Differentiation in Wireless Packet Networks Using Distributed Control," *IEEE J. Selected Areas in Communication*, Vol. 19(10), 2001, pp.2081- 93.

[24] Z. N. Kong, D. H. K. Tsang, B. Bensaou and G. Deyun, "Performance Analysis of IEEE 802.11e Contention based Channel Access" *IEEE J. Selected Areas Communication,* Dec. 2004.

[25] M. Benveniste, G. Chesson, M. Hoeben, A. Singla, H. Teunissen and M. Wentink, "EDCF Proposed Draft Text," *IEEE Working Document 802.11-01/12 1r1*, March 2001.

[26] M. Sarkar, S. Gurajala and S. Kumar, "A QoS-Aware Medium Access Control Protocol for Real Time Traffic in Ad Hoc Networks", *18th IEEE Annual International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC'07)*, Athens, Greece, 3-7 Sept. 2007.

Sunil Kumar

He received Ph.D. in Electrical and Electronics Engineering from Birla Institute of Technology and Science, Pilani (India) in 1997. From August 2002 to July 2006, he was an Assistant Professor in the Electrical and Computer Engineering department at Clarkson University, Potsdam, NY. Since August 2006, he has been an Associate Professor and Thomas G. Pine Faculty Fellow in the Electrical and Computer Engineering department at San Diego State University, San Diego, CA. His research interests include (*i*) QoS-aware Cross-layer Protocols for Multimedia Traffic in Wireless Cellular, Ad hoc, Sensor and Cognitive Networks, and (*ii*) Robust Multimedia Compression techniques for wireless. He has published more than 80 research articles in international journals and conferences, including two book/book chapters.

Mahasweta Sarkar

She received Ph.D. in Computer Engineering from University of California at San Diego in 2005. Since August 2006, she has been an Assistant Professor in the Electrical and Computer Engineering department at San Diego State University, San Diego, CA. Her research interests include the scheduling, routing, optimal resource allocation, power management schemes in wireless networks like WLANs, WMANs, Sensor Networks and Ad Hoc Networks.

Supraja Gurajala

She received her M.S. degree in Computer Science in 2005 from Clarkson University, Potsdam, NY, USA. Her research interests are in QoS-aware MAC protocols for Ad Hoc Wireless Networks.

John D. Matyjas

He received Ph.D. in Electrical Engineering from the State University of New York (SUNY) at Buffalo in 2004. Currently, he is a scientist with the Air Force Research Laboratory in Rome, NY, performing research and development in the communications technology branch. His research interests are in the areas of wireless multiple-access communications and networking, statistical signal processing and optimization, and neural networks. Additionally, he serves as an adjunct faculty in the Department of Electrical Engineering at the SUNY Institute of Technology at Utica/Rome. Dr. Matyjas is recipient of the SUNY at Buffalo Presidential Fellowship and the SUNY Excellence in Teaching Award for Graduate Assistants.