Reservation and Admission Control for QoS support in Wireless Networks

Mustafa Özdemir
mozdemir@ece.neu.edu
RWIN-Lab ECE Dept.
Northeastern University
Boston, MA 02115

A. Bruce McDonald
mcdonald@ece.neu.edu
RWIN-Lab ECE Dept.
Northeastern University
Boston, MA 02115

Ram Ramanathan
ramanath@bbn.com
Internetworking Research Dept.
BBN Technologies
Cambridge, MA 02138

Abstract— **Supporting multimedia streams (e.g. voice/video over IP), in a Wireless LAN requires guaranteed access to channel capacity, which is best provided using capacity reservations. Existing Medium Access Control (MAC) protocols for Wireless LANs either do not provide guaranteed access or do so in a synchronous (e.g. polling, TDMA) context. We present a novel mechanism for reserved access within the asynchronous or contention-based paradigm (for example CSMA/CA, or 802.11 DCF). Our protocol, called Asynchronous Reservation Oriented Multiple Access (AROMA), allows a client to specify a capacity request using a moving window or leaky bucket descriptor. The server performs admission control, reserves the capacity and provides access to the capacity in a simple, efficient manner. The fully distributed version of AROMA provides a scalable control scheme that dynamically adapts backoff parameters for best-effort users based on measured traffic estimates, thus protecting otherwise vulnerable QoS reservations. AROMA is backward compatible with IEEE 802.11 DCF. The performance of AROMA using experimental analysis is analyzed. Experimentally, our comprehensive simulations using an enhancement of OPNET's 802.11 model, shows that AROMA can support more VoIP calls with a better QoS than IEEE 802.11e EDCF.**

I. Introduction

Recent years have seen a dramatic proliferation in the deployment of Wireless LANs (WiFi). In parallel, the use of packetized voice and video (e.g. VoIP) has also increased significantly. Given these trends, the need for real-time multimedia streams over wireless LANs (commonly referred to as VoWLAN) will likely become increasingly pronounced in the near future.

Most Wireless LANs today are based on the IEEE 802.11 b/g/a standard, which, unfortunately, does not provide support for real-time multimedia streams such as packetized voice and video. That is, IEEE 802.11 does not provide differentiation in the quality of service (QoS) perceived by each stream, thereby making it unusable for applications requiring assured QoS. For instance, if user A is videoconferencing over a wireless LAN and user B starts to transfer a large file over the same wireless LAN, user A's session will be disrupted.

We present AROMA - the first medium access control mechanism that provides reserved access in an asynchronous protocol. AROMA enhances CSMA/CA in general and 802.11 DCF in particular to enable capacity request characterization, reservation and admission control. AROMA allows non-QoS traffic access to residual capacity in a best-effort manner. We also describe a specific instantiation of AROMA for IEEE 802.11 based networks, which is backward compatible with IEEE 802.11 DCF. AROMA is vastly simpler than most synchronous-based protocols such as 802.11e HCCS. Finally, although our treatment of AROMA in this paper is restricted to access point based wireless LANs, it is extensible to ad hoc networks because its underlying protocol, CSMA/CA is distributed.

This research was partially supported by NSF Career Award 2004 CISE 0347698 and BBN Technologies.

Approaches to QoS provision can be broadly classified as *prioritized* or *reserved*. In prioritized access, users are classified into multiple priority classes and the network services higher priority traffic before lower priority traffic. In reserved access, users request capacity that is dedicated and guaranteed to them. If the resources are available, while the tradeoffs between reserved and prioritized access have and continue to be hotly debated, a few indisputable points stand out: With prioritized access, a node or session is provided better service than the competition, but is not guaranteed specific capacity. Thus, if other nodes/streams of equal or greater priority join the network, service will degrade. Reserved access not only provides capacity guarantees, but also admission control, and typically also request characterization. Multimedia streams such as VoIP are very sensitive to capacity fluctuations – unlike data flows, it is far better to refuse admission to a voice stream in the first place, rather than admit it and make it susceptible to capacity fluctuations [3].

AROMA adapts many ideas developed for traffic characterization and congestion control at the network layer [4], including traffic descriptors such as moving window and leaky bucket descriptors, and token bucket management. The identification of CTSs as resources that can be controlled and the realization of the correspondence between CTSs and tokens in this context is a key insight behind the ability to adapt these schemes. Moreover, a best-effort data control scheme is proposed in order to reduce the channel access times at high load for only best-effort stations to further improve scalability. To the authors' knowledge this is the first application of token bucket within a WLAN MAC context [6].

The remainder of the paper is organized as follows: A detailed description of AROMA follows in section II. Performance analysis based on discrete-event simulation is presented for the centralized and scalable distributed scheme in section III. Finally, conclusions are presented in section IV.

0-7803-9305-8/05/$20.00 ©2005 IEEE

II. AROMA Description

AROMA is based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) approach, which is the basis for the IEEE 802.11 wireless LAN standard. Although AROMA applies to most variants of the CSMA/CA approach, it is presented in the context of 802.11 DCF as that is the most prevalent version. First, a brief summary of the 802.11 DCF is given, and then describe the AROMA enhancements is described.

The IEEE 802.11 DCF uses up to four frames for each data packet transfer. A sender first transmits a Request-to-Send (RTS), and the receiver responds with a Clear-to-Send (CTS). Then the sender sends the DATA and finally the receiver completes the transaction with an Acknowledgment (ACK). Both RTS and CTS contain the proposed duration of the data frame. Nodes located in the vicinity of the sender and the receiver that overhear one or both of the RTS/CTS store the duration information in a *network allocation vector (NAV)*, and defer transmission for the proposed duration. The IEEE 802.11 protocol senses the medium and waits for it to be idle before transmitting. Collisions are resolved using a backoff mechanism. A contention window (CW) is maintained by each node and the random backoff interval is chosen from within this window. The CW is increased exponentially when frames are lost (as discerned by the non-receipt of CTS or ACK). The description is necessarily brief, and readers are referred to [2] for further details.

AROMA is a protocol between two logical entities - a client (reservation requester) and a server (reservation granter). In general, it is to any wireless network where a client-server relationship can be established. For example, in a cellular network the hand set contains the client and the base station the server. In an ad hoc network, every node and its immediate downstream node on a flow constitute, respectively, a client-server pair. Intermediate nodes are both a server for the upstream node and client for the downstream node. The AP based client-server interaction is illustrated in Figure 1, and summarized below. Sections following the summary elaborate on each aspect of AROMA.

When a client needs to reserve capacity, it sends a modified RTS referred to as the *Reservation-RTS (R-RTS)*. The purpose of the R-RTS is to let the server know that the ensuing data packet contains information about the reservation request of this client. The data packet following the R-RTS includes the requested capacity. Two well-known traffic descriptors are used for this purpose, which are elaborated on in section II-A. R-RTS is sent to the server. Retransmission rules for R-RTS are identical to the existing rules for regular RTS, as given in IEEE 802.11 standard. Upon receipt of an R-RTS, the server sends a CTS to the client and waits for the data packet, which includes the reservation request information.

Upon receiving the data packet, the server checks to see if the request can be accommodated by executing the *flow reservation* functionality, as described in subsection II-B. This determines, based upon the reservation request and existing reservations, whether this flow can be admitted or not. If flow is admitted, the reservation is successfully made and the server sends an ACK to the client. Both the server and client create

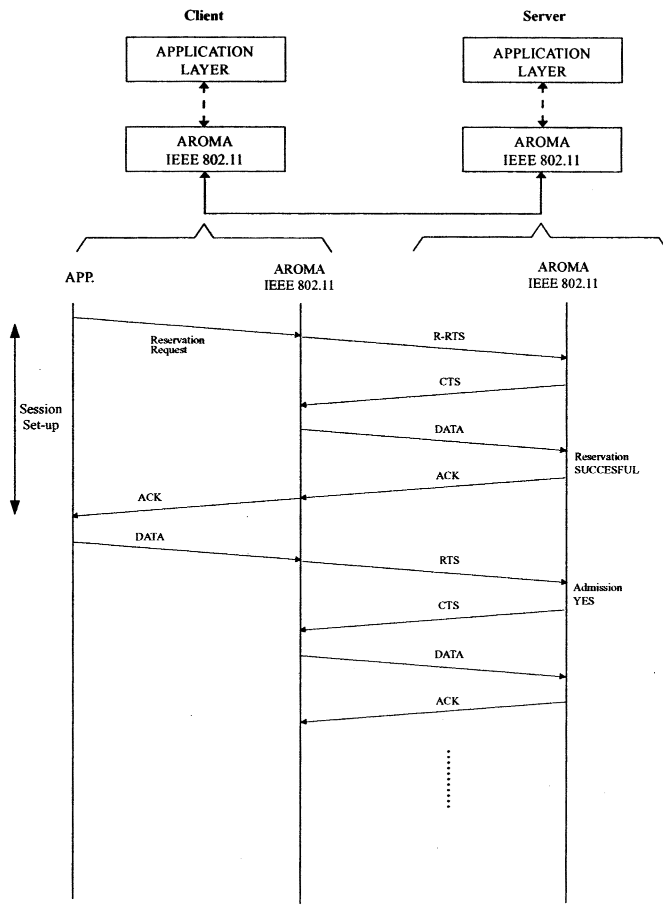

Fig. 1. AROMA Client-Server Interaction

"soft-state" regarding the reservation. If the request cannot be accommodated, the ACK is not sent and the client times out and follows base IEEE 802.11 rules.

Upon receipt of a RTS, the server first looks up the reservation state using the client identifier and decides whether or not to send a CTS by executing the *packet admission* functionality, as described in the subsection II-C. If the RTS is from a client that exceeded its reserved capacity, or if it has no reservation, it is treated as "best-effort", that is, a CTS is returned if there is residual capacity.

All traffic does not require guaranteed capacity. Clients may send packets without having a prior reservation. There may also be clients that are not AROMA compatible which behave as per the 802.11 specification. Packets that do not have a reservation associated with them are treated on a "best effort" basis. That is, CTS is returned for RTS if and only if there is available capacity. Reservations time out after a configured period of time. That is, a reservation is purged at the server if no packet arrives for a predetermined period of time. The client's packets are then treated as "best effort". If the purging of the reservation turned out to be premature, the client must determine to re-establish a reservation.

A. Traffic Descriptors

A traffic descriptor is a set of parameters used to describe the behavior of a source of traffic [4]. AROMA accommodates two kinds of descriptors, both of which are popular in the wired networking literature:

- *Moving Window.* This descriptor has two parameters - Bits (B) and Time (T). The source is allowed to send at most B bits over all windows of length T seconds. The average rate of this source is B/T bits/sec.
- *Leaky Bucket.* Also called the *linear bounded arrival process* (LBAP), this descriptor has two parameters: Rate (R) and Burst (B). The source is allowed to send at most R.t + B bits over any given interval of t seconds. The average rate of this source is R, with a deviation of B bits possible "occasionally".

Reader are referred to [4] for a detailed description of these descriptors . In AROMA, the leaky bucket is used as the primary descriptor and the moving window is mapped to it, thus, both are accommodated. The rate R is further decomposed into token size and token rate. Thus, there are three parameters, Token Size (TS), Token Rate (TR), and Burst (Bu). TS*TR, which is R, corresponds roughly to long-term average rate allocated by server to the client, and Bu is the longest burst a client may send. Setting the token- bucket limit to one token and replenishing the bucket at the average rate makes it a moving-window descriptor.

B. Flow Reservation

Upon initialization, and periodically, the server determines the aggregate *effective* capacity B_{eff} of the medium. This is the capacity seen by the application and represents the total rate of admissible flows. Estimation of B_{eff} can be achieved using theoretical analysis or simulation model of the wireless LAN. A validated theoretical approach is developed but it is not put in this paper because of space limitation. Estimation of the effective channel capacity is an interesting, challenging, problem, but one that is orthogonal to AROMA. Any QoS reservation system would face this problem and AROMA can utilize any mechanism developed for this purpose.

An AP also maintains a reservation list containing information about all active node reservations. Each element in the list is $< client - id, reserved - capacity >$ which identifies a reservation uniquely. Upon node activation or reset, the reservation list is empty. A small amount of minimum capacity is reserved for all best-effort flows, in order to prevent their starvation.

When a reservation request arrives in the DATA frame following the R-RTS, the server extracts the requested capacity rate. If this plus the used capacity rate plus minimum best-effort capacity rate exceeds the total capacity C, the R-RTS is silently discarded. The client times out, notes that it does not have a reservation and may continue to use the channel as a best-effort node, or attempt a reservation at a later time.

If the request can be accommodated, a new entry for this reservation is added to the reservation list and an ACK is sent to the client. This tells the client that the request is accepted.

C. Packet Admission

When the client needs to transmit a data packet, it first sends an RTS to the server. If this client has a valid reservation, the server associates the corresponding packet with the client. If it has not, the server considers it as best-effort traffic, which does not have a prior reservation. The server decides whether or not to send a CTS in response to the RTS based on the leaky bucket admission controller, which works as follows.

Recall from section II-A that a reservation request has three parameters: *token size (TS)*, *token rate (TR)*, and *burst size (BU)*. Each client i is allocated a token bucket of capacity BU_i that is replenished at a rate TR_i, with tokens of size TS_i. That is, every t seconds, the controller adds $t * TR_i$ tokens to the bucket to produce an effective rate of $R = TR_i * TS_i$ bits/sec. The bucket overflows if the number of tokens exceeds BU_i.

The server honors an RTS if and only if the sum of the token sizes in the bucket adds up to at least the packet's size (indicated in the RTS). The controller rejects a packet if it does not have sufficient tokens for transmission. That is, no CTS is sent in response to an RTS if the admission function fails. The client is forced into an EIFS backoff. If the sum of the token sizes is less than the packet size, then the admission controller treats it as if it is from a best-effort client. Hence, the packet still might be accepted from the best-effort reservation. On a packet admission, the controller removes tokens corresponding to the packet size from the token bucket.

Over the long term, the rate at which packets are accepted from the admission controller is limited by the rate at which tokens are added to the bucket, which in turn is the reserved rate.

D. IEEE 802.11 Compatibility

This section presents the changes in IEEE 802.11 frame format for using AROMA. All changes are backward compatible with IEEE 802.11. That is, upgrading the server (AP) and introducing AROMA-enhanced clients do not affect legacy clients in any way. In particular, all packet format modifications and algorithmic modifications are made in a manner transparent to the legacy clients. Thus, AROMA-enhanced nodes will co-exist with legacy nodes. Further, this allows us to "incrementally" deploy AROMA in an enterprise using WLANs.

AROMA introduces only one new frame, namely the R-RTS. These control fields are intended for DATA frames, and are ignored for RTS, CTS and ACK frames. One of these bits is used, namely the Order bit, to indicate an R-RTS. That is, if the bit is set to 1, the frame is treated as an R-RTS by an AROMA-capable node. A server checks if this field is set to 1 upon reception of RTS. A node which does not have AROMA capability sees the R-RTS as a regular RTS since it does not have to deal with the order bit. Such a node then defers on NAV. This is the only change in the frame format.

III. Simulation Model and Performance Analysis

This section presents results from discrete event simulation modeling that achieves the following goals: (1) a new Best-Effort (BE) data scheme improving AROMA in terms of scalability to the increasing traffic-load conditions; (2) performance analysis of AROMA in the access-point (AP) configuration using one QoS traffic class for VoIP traffic and a "default" class for BE traffic. The performance analysis focuses on delay and packet-loss metrics while varying the ratio of QoS to BE stations given a fixed number of nodes. The experiments include comparison of AROMA to the most significant proposed alternatives.

802.11 Parameters	Values
PYH Layer Specification	DSSS
Channel Transmission Rate	11Mbits/sec
CW_{min}	32
CW_{max}	1024
retry limit	4

TABLE I

IEEE 802.11 SYSTEM PARAMETER VALUES

VoIP Traffic Parameters	Values
Packet Interarrival time	20ms
Voice packet length	160 bytes
RTP layer overhead	12 bytes
UDP layer overhead	8 bytes
IP layer overhead	20 bytes
MAC layer overhead	34 bytes
PHY layer overhead	24 bytes

TABLE II

VOICE TRAFFIC PARAMETERS

A. Simulation Parameters

The discrete-event simulation engine OPNET provided the tools to build an effective model of AROMA for performance analysis. All simulation results reflect statistically significant analysis based on a 95% confidence level and relative precision of 0.05. The important system parameter values are given in table-I. The others can be found in the IEEE 802.11 MAC layer implementation in OPNET. System parameters were chosen to reflect typical installations of IEEE 802.11b;

The traffic parameters were selected to model the behavior of the G.711 codec using 20ms packetization intervals for VoIP. Peak allocation was used by the server since the achievable throughput limits can be estimated accurately by the analytical model which is not put here because of space limitation. All traffic sessions assumed a wired server, hence, traffic was not generated between wireless nodes. Based on the G.711 specification the raw packet lengths for voice were fixed at 160 bytes. However, Table II indicates the overhead required by the underlying protocol layers: RTP, UTP, IP and MAC; the aggregate frame lengths came to 258 bytes (significant protocol overheard). Each VoIP session was held for 3 minutes, consisting of unidirectional traffic from the wireless client. The background load was varied by changing the number of best-effort clients. Data traffic was generated by wireless nodes sending fixed 512 byte frames at a mean interarrival rate of 20 ms. The aggregate data load was set at 0.0 and 1.6 Mbps for each fixed number of VoIP nodes. The number of voice stations was varied from 2 to 18.

As the table shows the simulation models the 11 Mbps version of IEEE 802.11. The maximum throughput when using only VoIP traffic can be as low as 960 Kbps using a typical VoIP codec (G711 with 20ms audio data per 200 bytes packet) [1] [5]. Consequently, the maximum number of simultaneous VoIP calls that can be placed in the network cannot exceed 12. Admitting 12 voice stations is feasible if no BE stations are active in the system. The quality of this session would be excessively sensitive to best-effort traffic. Hence, a maximum of 10 voice stations is enforced in the AROMA simulations to ensure minimal acceptable voice performance. Simulation results provide validation for this conclusion.

B. The Scalability Control (SC) Scheme

It is observed in simulations that the performance of voice stations is very sensitive to best-effort traffic. In other words, the system does not scale well with the number of BE stations. When the number of BE stations is reasonably high (hence the load), AROMA is bound to waste a lot of channel capacity and perform poorly. In this section, we propose a measurement-based scalability control mechanism for best-effort data transmissions to overcome the scalability problem. The proposed scheme is fully distributed scalability-control mechanism, since stations dynamically control parameters themselves locally based on traffic load condition. Stations make decision based on the local observed measurements.

Each station measures network traffic condition and estimates appropriate parameters dynamically, i.e., CW_{min} and CW_{max}. The reasons for this scalability control are in the following. First, too many best-effort transmissions degrade the system performance, including the existing QoS streams, since many best-effort transmissions cause a lot of collisions. Hence, the existing QoS streams become vulnerable to best-effort data transmissions. In such a case, a station should observe that the network condition becomes worse by indications such as an increased collision ratio or MAC delay etc. Hence, the station should increase its parameters, i.e., CW_{min} and CW_{max}. On the other hand, if there are very few best-effort data transmissions, the throughput might decrease even though the QoS streams become less vulnerable. In such a case, each station should note that the network condition becomes better by indications such as a decreased collision ratio or MAC delay, etc. Hence, the station should decrease its parameters, i.e., CW_{min} and CW_{max}. The challenging issue is how to adaptively adjust these parameters based on those indications. Second, the reason for handling QoS transmissions need guaranteed service, whereas best-effort transmissions do not.

For best-effort transmissions each station dynamically changes the two parameters (CW_{min} and CW_{max}) with time, based on the measurements of MAC delay which reflects the traffic load condition. We propose one scheme, the Scalability Control (SC) which is based on derivative tendency. In the SC scheme, parameters are changed dynamically according to the same tendency of traffic-load indications. More detail is given in [7].

Here MAC delay is preferred as a traffic load indication since it totally reflects the network condition better and also is easier for stations to find out than collision rates. The average MAC delay for a successfully transmitted packet is defined to be the time interval from the time the packet is at the head of its MAC queue ready to be transmitted, until an acknowledgment for this packet is received. Also, the following moving average integration technique is adopted to take history measurements into consideration. Note that at the first measurement, the history value is equal to the corresponding measurement.

$$D(i) = \alpha \times D(i-1) + (1-\alpha) \times D_{measurement} \quad (1)$$

where α is a smoothing/aging factor and $D(i)$ is the MAC delay for the ith successfully transmitted packet. Here $D(i)$

is the only indication defined and the following difference tendency function for the SC scheme is used

$$
\begin{aligned}
CW_{min}(i+1) &= min\{CW_{min}, \theta \times CW_{max} \times [D(i) - \\
&\quad D(i-1)] + CW_{min}(i), CW_{max}\} \quad (2) \\
CW_{max}(i+1) &= CW_{max} \quad (3)
\end{aligned}
$$

where we have only one parameter θ that needs to be defined. The value θ can be either predefined or dynamically changed based on some other measurements at the run time. The predefined value θ used in the simulations is 0.9. The values of CW_{min} and CW_{max} are given in table-I for QoS stations and 32 and 8192 for BE stations in the simulations.

More complex schemes can be designed easily. However, the SC scheme given here is good enough for our scalability control purpose according to simulation results in section III-D.

C. Performance Metrics

The two most important metrics reflecting the delivered QoS for VoIP are packet-loss and end-to-end delay. Packet loss may result from any of the following: (1) buffer overrun, (2) excessive MAC-layer collisions or (3) access denial (rejection) by the AROMA Server. Delay may be incurred in numerous ways; here we consider queuing delay at the source station and MAC delay incurred by the frame in service at the source station. Studies have shown that for acceptable voice quality VoIP can tolerate 200 ms delays with as much as 2% packet loss. The jitter in the delay caused due to steady variance in traffic patterns and transient situations is remedied through using the SC scheme in BE stations. Note that packet loss is represented as a percentage according to the following equation:

$$PL = 100 \cdot \left(1 - \frac{Pkts_{rcvd}}{Pkts_{sent}}\right) \quad (4)$$

where the fraction shows the ratio of received packets to the total number of packets generated by the source.

D. Simulation Results

We conduct extensive simulations to study different parameters such as throughput, delay, delay per voice stream per station, MAC delay per voice stream per station, and effects of traffic load on AROMA with and without the SC scheme, as well as effects of θ on the SC scheme.

Figure-2 shows average delay per voice stream per station for case (a) without AROMA (b) with AROMA and without SC (c) with AROMA and SC (θ = 0.9). Here both QoS and BE traffic load are high. With SC scheme [Figure-2 (c)], delay and jitter in the delay for voice stream has been greatly improved.

Figure-3 shows the effects of θ on voice MAC delay per station for case (a) without the Scalability Control (SC) scheme (b) with SC (θ = 0.2) (c) with SC (θ = 0.5) (d) with SC (θ = 0.9). Here both QoS and BE traffic load are high as well. As the mean and standard deviation value of MAC delay are illustrated in Figure-3, without scalability control, voice performance per station is severely degraded when BE traffic load is high. With the SC scheme, voice performance per station has greatly improved, especially for a large θ value.

Figure-4 depicts the mean packet loss and end-to-end delay for AROMA, AROMA with SC, IEEE 802.11 DCF and IEEE 802.11e EDCF with no best-effort traffic. As expected, packet loss and delay increase with the number of voice stations. There are three curves for AROMA case in the figures. The circled and squared curve represents only admitted voice stations without and with the SC scheme, which is limited to 10 even though active voice station number is increasing. The crossed curve shows the result for all voice stations. The figures clearly demonstrate that AROMA with SC outperforms AROMA, DCF and EDCF by supporting 10 VoIPs all the time.

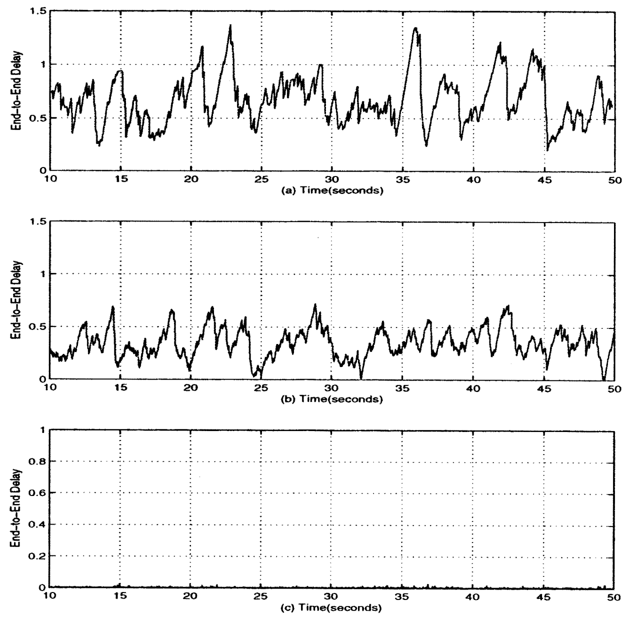

Fig. 2. Average end-to-end delay in seconds per voice station versus simulation time (a) without AROMA (b) with AROMA and without SC (c) with AROMA and SC (θ = 0.9)

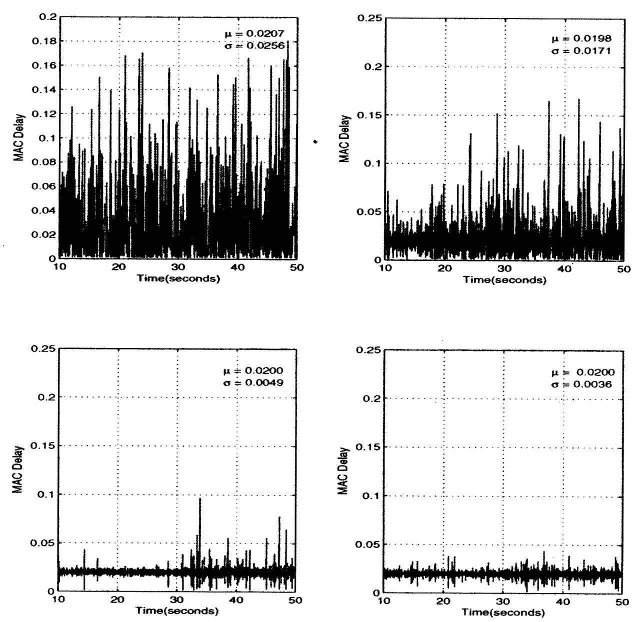

Fig. 3. Effects of θ on MAC delay per voice station (a) without SC (b) with SC (θ = 0.2) (c) with SC (θ = 0.5) (d) with SC (θ = 0.9)

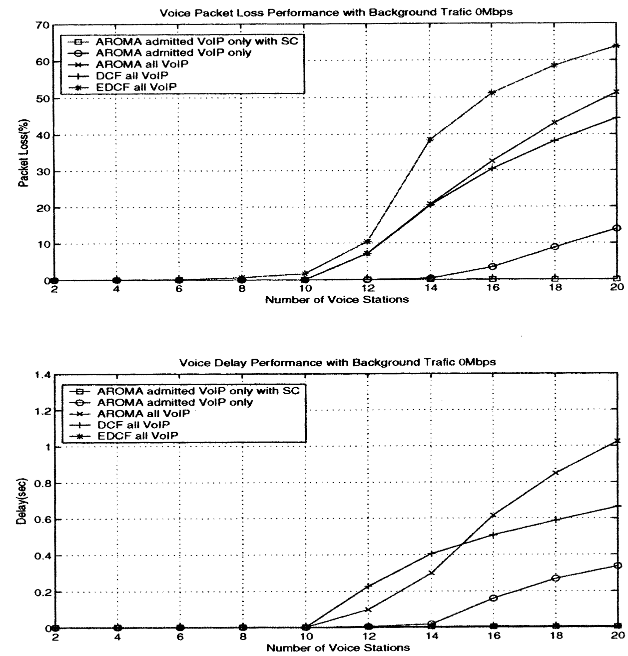

Fig. 4. Voice packet Loss and Delay Performance when the Background Traffic is 0Mbps

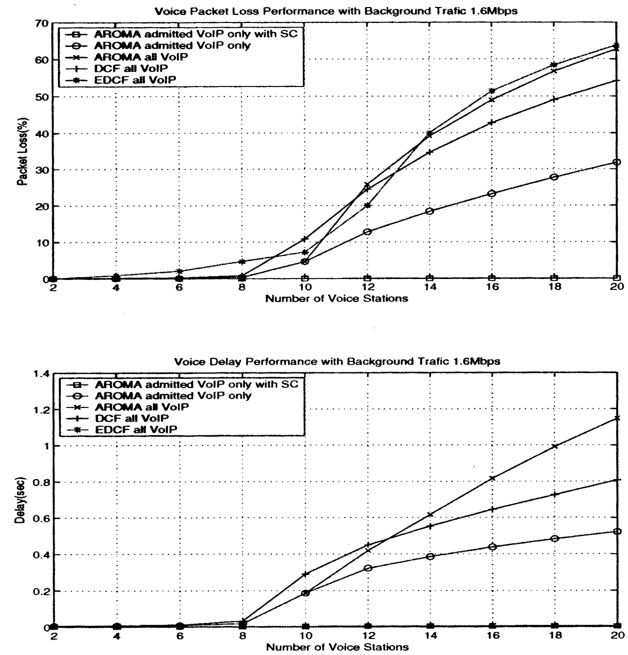

Fig. 5. Voice packet Loss and Delay Performance when the Background Traffic is 1.6Mbps

Results for delay using EDCF show that EDCF significantly outperform DCF and AROMA but not AROMA with SC. This result is due largely to effects of system parameter settings required in EDCF: the minimum contention window size (CWmin) is set to $\frac{W}{4}$ and the maximum contention window size (CWmax) to $\frac{W}{2}$, whereas CWmin and CWmax are set to W and $2^m W$, respectively in the AROMA and DCF simulations. Moreover, EDCF performs so poorly with respect to packet-loss that its improved delay performance has no application level relevance. When the wireless network is used only for VoIP traffic, after maximum effective 10 VoIPs, additional VoIP-streams lead the network to congestion collapse. Thus, all voice traffic cannot be sent through the network under DCF and EDCF. However, AROMA still provides 10 acceptable quality VoIPs given as many as 15 active voice stations and AROMA with SC provides 10 VoIPs in any case.

Results given heavy background traffic of 1.6Mbps are shown in Figure-5. Heavy traffic does not have any affect on the performance of AROMA with SC. AROMA supports 9 VoIPs when compared with 8 and 7 for DCF and EDCF, respectively. When the network is shared between voice and data, additional voice clients increase network access contention causing AROMA, DCF and EDCF performance to become much more sensitive to the additional VoIP-streams than AROMA with SC.

IV. Conclusions

We have presented the first protocol – AROMA – that provides reservations within an asynchronous context. In particular, AROMA enhances CSMA/CA in general and IEEE 802.11 Distributed Coordination Function (DCF) in particular to enable traffic description, reservation requests, and admission control. AROMA is backward-compatible with IEEE 802.11 DCF and therefore can co-exist with non-AROMA-enhanced clients. This allows for incremental deployment in an existing Wireless LAN. AROMA uses well-known traffic descriptors from network layer QoS concepts, such as leaky-bucket and moving window descriptors which facilitates cross-layer integration for end-to-end QoS solutions.

As seen from figures, AROMA scales better with increasing numbers of BE users compared to IEEE 802.11 DCF and IEEE 802.11e EDCF. However, since all the nodes use the same CW and back-off appropriately in AROMA, it does not scale well enough when the number of BE nodes is very high. Implementing adaptive CW and back-off algorithms, the SC scheme, over AROMA provided far better scalability.

References

[1] G. Bianchi, *Performance analysis of the ieee 802.11 distributed coordination function*, IEEE Journal on Selected Areas in Communications **vol. 18** (2000), no. 3, pp. 535–547.

[2] IEEE Standards Board, *Ieee standard 802.11 - wireless lan medium access control (mac) and physical layer (phy) specifications*, The Institute of Electrical and Electronics Engineers, Inc, 345 East 47th Street, New York, NY 10017-2394, USA, June 1997.

[3] S. Garg and M. Kappes, *Admission control for voip traffic in ieee 802.11 networks*, IEEE GLOBECOM, 2001.

[4] S. Keshav, *An engineering approach to computer networking*, 7th ed., Addison Wesley, 2000.

[5] M. Ozdemir and A. B. Mcdonald, *An m/mmgi/1/k queuing model for ieee 802.11 ad hoc networks*, IEEE/ACM PE-WASUN'04 (Venice, Italy), October 2004.

[6] S. Ramanathan, *Method and apparatus for asynchronous reservation oriented multiple access*, U.S. Patent number 6,577,613, June 2003.

[7] H.Li Y.Xiao, *Local data control and admission control for qos support in wireless ad hoc networks*, IEEE Transactions on Vehicular Technology **53** (2004), no. 5, 1558–1572.