ABSTRACT

MAHBOOB, AHMED M. Access Point-Coordinated Contention Resolution for Channel Access in Wireless LANs. (Under the direction of Assistant Professor Khaled A. Harfoush.)

The most widely deployed wireless networking (WLAN) standard, 802.11, grants access of the wireless channel to contending stations (STAs) through the Carrier Sense Multiple Access/Collision avoidance (CSMA/CA) mechanism. This approach in general is not scalable, is biased against the network's Access Point (AP), and does not satisfy the Quality-of-Service (QoS) requirements of different flow types. These problems persist even in the most recently ratified standard, 802.11e.

In this thesis, we propose a new channel access scheme, ARC (**A**ccess Point-Coordinated Contention **R**esolution for **C**hannel Access) that demotes contention and promotes coordination among wireless STAs. ARC enhances the 802.11 contention-based standards with an efficient messaging protocol between the AP and the STAs, in order to communicate congestion information to the AP and channel assignments to STAs. With all packets in 802.11 Infrastructure mode being either transmitted or received by the AP, the AP is in a unique position to maintain flows' congestion information and to make channel assignment decisions. Simulation results reveal that ARC offers better channel utilization; better support for QoS demanding flows than the 802.11 standard, and resolves the bias against the AP.

Access Point-Coordinated Contention Resolution for Channel Access in Wireless LANs

by

Ahmed M. Mahboob

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Computer Science

Raleigh, North Carolina

2006

APPROVED BY:

Dr. Edward W. Davis Jr.	Dr. David J. Thuente
Dr. Khaled A. Harfoush Chair of Advisory Committee	Dr. Mihail L. Sichitiu

To my parents and my mentor ...

Biography

Ahmed Mishar Mahboob was born in Dhaka, Bangladesh in 1979. During his childhood he had the opportunity to enjoy the comfort and warmth of a joint family comprising his grandparents, uncles and cousins. He passed the Higher Secondary Examination with distinction securing the 10^{th} place out of 200,000 students of his country. Mishar went to the top engineering school to study Computer Science and Engineering which is regarded as the most sought after department for undergraduate study. In his final year he published his thesis in a local conference which motivated him to pursue higher study and research in the field of Wireless Networking. After graduating in April 2004, Mishar started working as a lecturer in a private institution and gathered teaching experience for 8 months. In Spring 2005, Mishar joined NC State. Since then it has been an onward journey to seek state of the art knowledge and quest for excellence in research.

Acknowledgements

I would like to express my heartiest gratitude to the Almighty, without His Blessings I would have not been able to come this far and be surrounded by so many benevolent and erudite minds. Personally, I am grateful to Mr. Manjunath Prabhu with whom I worked closely to come up with an initial foundation of this work. When it was required, I could find the assistance of Dr. Mihail and Mr. Ajit Warrier - a PhD student in our department. The encouragement of my friends deserves special mention. Sumaiya and Vineet; you two were inspirational! I greatly appreciate your words of hope and optimism. I am very thankful to my parents and my sister. My graduation will not only be an achievement of mine but also a milestone in their lives. Finally vote of thanks to my committee members, Dr. Ed Davis and Dr. Thuente. Not to forget my mentor and supervisor Dr. Khaled Harfoush, who lead me all the way this far to earn this laurel from such a coveted institution.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Wireless Local-Area Networks (WLANs) such as 802.11a/b/g standards have quickly become the fastest growing type of consumer networking device. This is in large due to the mobility and convenience that they offer to users. WLANs are not expensive to build and maintain, and provide shared gross data rates from 10 to 50 Mbit/s as opposed to the limited 10-100 kbit/s offered by cellular wide-area networks such as GSM, GPRS, and UMTS. It is not hard to envision that, in the near future, it will be possible to construct large scale wide-area wireless IP networks interconnecting neighboring wireless islands to each other and to the Internet, offering high bandwidth and extended coverage similar to those currently used to offer cellular phone service. Users will not only be able to use this network in the comfort of their homes, in parks, or in coffee shops; but also while riding trains or even while driving their cars. Offering reliable connectivity with reasonable performance for real-time applications such as video streaming and Voice over IP (VoIP) will be the keys to the commercial success of such networks. The hardware aspects of such networks are already becoming a reality with the advent of WiMax technology, which has a maximum range of several miles. There are several existing industry products, such as the CISCO 3200 series routers, which support wireless connectivity to mobile vehicles. While this technology will be able to compete with current broadband services for stationary users, it also opens up new possibilities for mobile users as well, potentially giving rise to large-scale high-speed data networks. Demand for this technology will increase as the number of portable devices available to consumers increases and as the performance of notebooks approaches that of

traditional desktops.

Satisfying different *Quality of Service* (QoS) requirements for different applications in wireless medium is challenging. Some applications like web browsing and file transfer target reliable, high throughput communication; real-time applications like VoIP and video streaming target bounded delay and jitter communication and high throughput is not of high concern; and different distributed gaming applications target different combinations of low latency, low jitter, high reliability, and high throughput communication. In general, MAC layer protocols in wireless environments are designed to satisfy one QoS and not the other; typically reliable, high throughput communication. This results in poor QoS for real-time applications as the MAC layer spends time retransmitting lost packets, while queued packets are missing their deadlines. While throughput, delay and jitter are correlated, they are distinct and are realized through different schedulers. Scheduling for bounded jitter would relate the transmission time of successive packets; scheduling for bounded delay would only be concerned with deadlines of single packets; and scheduling for maximum throughput would *not* be concerned with packet deadlines but rather with the throughput of the flow as a whole. *Scheduling mechanisms capable of dealing with different QoS requirements is a necessity in next generation WLANs.*

Furthermore, the IEEE 802.11 standard mainly grants channel access to wireless stations (STAs) through the Carrier Sense Multiple Access/ Collision avoidance (CSMA/CA) mechanism. CSMA/CD is contention-based and results in high competition over the shared wireless medium when the number of competing flows is large, leading to packet collisions, transmission back-offs, and under-utilized wireless channels. The problem persists even with the introduction of high bandwidth wireless technology, and as we make the case in this thesis, with the introduction of the new 802.11e standard upgrade, designed to support QoS. The 802.11e standard assigns static priority to different flow types to enable real-time to better compete for channel access and/or explicitly poll STAs to allow them transmission windows at pre-defined time intervals. The former mechanism is problematic in highly loaded scenarios and the latter is problematic when STAs have different flow types with different QoS requirements. For example, a polled STA with best-effort traffic will be able to transmit before another STA, which has higher priority traffic.

In this thesis our goal is to develop effective means to accommodate different QoS requirements of applications in WLANs while improving the overall network utilization. Specifically we propose a QoS-scheduler, ARC (**A**ccess Point Coordinate Contention

Resolution for **C**hannel Access), which relies on the following key observations: 1. In the 802.11 standards Infrastructure mode, all packets, which are transmitted by the Access Point (AP) are overheard by all stations (STA) associated with this AP. 2. The AP carries more load than STAs. For each packet transmitted by an STA, the MAC layer of the AP is required to respond with an ACK. Most real-time flows, especially VoIP flows, are bi-directional. Wireless STAs are not expected to work as servers and will mostly download content from the Internet through the AP. The 802.11 standard does not give any preference to the AP over normal STAs when accessing the communication channel.

ARC is implemented at both the STAs and the AP. Any STA computes some measure of its urgency in transmitting packets based on its perceived QoS, like the queuing delay experienced by VoIP packets. Then informs the AP about it through some bits in the MAC header whether in data or ACK packets. We use few additional fields in the Frame Control field of the MAC header for this purpose. The AP gets the urgency feedback from the STAs, and determines which STA should claim the wireless medium sooner. Then, the AP sends a control packet (ACK) to notify all STAs about the lucky STA that will claim the channel for the next transmission period. This arbitration between the STAs by the AP reduces the competition between the STAs especially in times of congestion and avoids the 802.11 bias against the AP. This arbitration also allows for new services such as high priority 911 calls in WLANs, service guarantees for throughput demanding flows, etc. ARC does not assign access to the channel all the time, and allows for periods of CSMA/CD contention when the channel is not congested, which enhances the APs awareness about the state of flows before congestion. Simulation results reveal that ARC offers better channel utilization, better support for QoS demanding flows than the 802.11 standard.

The rest of this thesis is organized as follows. In Chapter 2 we survey related work and compare to ours. In Chapter 3 we provide the design and details of the ARC scheduler. In Section 4 we introduce the simulation results and highlight the design tradeoffs of ARC. We finally conclude in Section 5.

Chapter 2

Related Work

This chapter introduces the fundamental concepts of 802.11 and 802.11e and the limitations in the presence of real time traffic. We also survey research work targeted at dealing with real time traffic.

2.1 IEEE 802.11b

The IEEE 802.11 MAC protocol [2] supports two access methods; named DCF *(Distributed Coordination Function)* and PCF *(Point Coordination Function)*. The DCF access method CSMA/CA *(Carrier Sense Multiple Access with Collision Avoidance)* provides services for asynchronous data transmissions. The PCF has a point coordinator that allows polled STAs to finish its transmission in the assigned slot. Details of these two schemes are provided below.

2.1.1 DCF (Distributed Coordination Function)

In this mode of operation, a station must sense the medium before transmitting its data by PHY and MAC layer virtual sensing. If the medium is found idle for a time interval longer than *DCF interframe spacing* (DIFS) and the backoff counter expires the station starts transmitting. But in case of a busy medium or after the completion of transmission a station will start the random backoff procedure. The station picks up the random time with

the selection of a pseudo random integer uniformly distributed between CW_{min} (initially 7) and CW_{max}. = 255. The backoff time = CW value * slot time, the slot time depends on the physical layer. The backoff timer is decreased only when the medium becomes idle, paused otherwise.

Each time, when the medium becomes idle, the station waits for a DIFS interval and then starts decrementing its backoff counter. Once the backoff counter expires the station transmits. If the station is successful transmitting it will restart backoff after the ack is received. But in case of unsuccessful transmission, the contention window is doubled to reduce the probability of collisions . The contention window is reset to a fixed minimum after each success to improve the channel utilization. The NAV *(Network Allocation Vector)* is used for MAC virtual carrier sensing, by updating the local NAV with the value of other stations' transmission time. By using NAV, a station can know when the current transmission ends and the channel is idle.

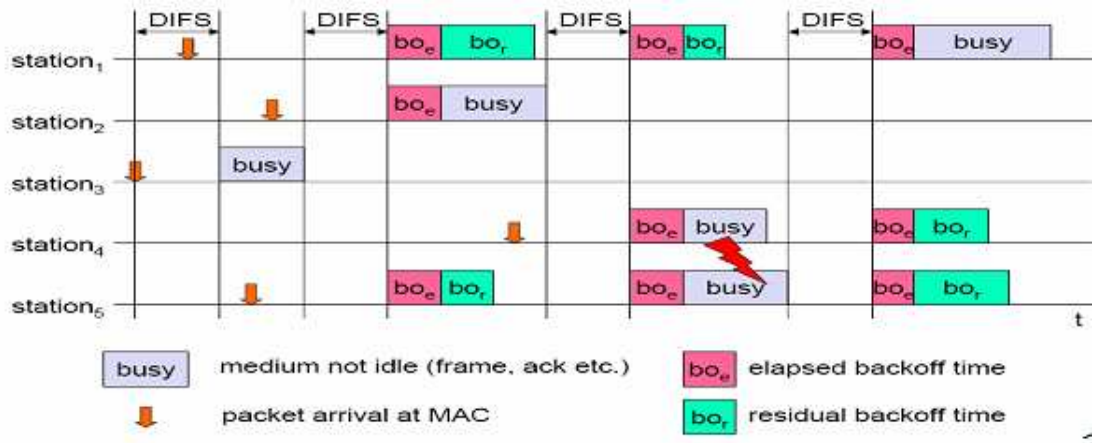

Figure 2.1: Backoff Procedure (Source [16])

A collision occurs if two (or more) stations have detected the medium as idle for DIFS, both are allowed to send and both start their transmissions immediately (as can be seen in Figure 2.1. To resolve repeating collision CW_{min} value has to be set according to equation 2.1 for the first four retransmission and after that it will be set to CW_{max} i.e.

255.

$$CW_{min,new} = 2 * CW_{min,old} + 1 \tag{2.1}$$

2.1.2 PCF Point Coordination Function

The PC (Point Coordinator) maintains a list of registered stations and polls them sequentially. Each station can transmit data in Shortest Interframe Spacing time (SIFS) The PC first senses the channel for a *PCF InterFrame Space* (PIFS) interval and then starts a *CFP* (Contention Free Period) by broadcasting a beacon. PIFS is shorter than DIFS which can help the PC to win the channel contention from the DCF mode. All stations add *CFPMaxDuration* to their NAVs, which prevents them from taking control of the medium during CFP.

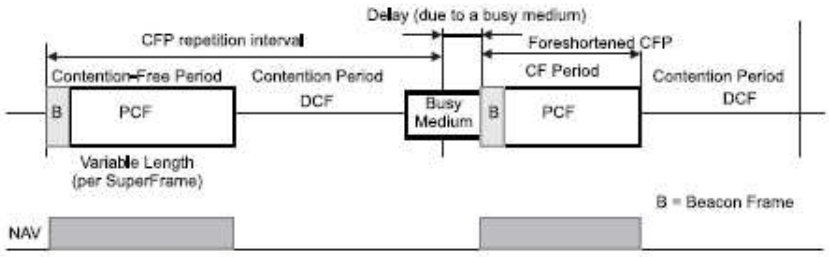

Figure 2.2: PCF/DCF Rotation (Source [2])

The PC can terminate the CFP by transmitting a CF-end packet. All stations receiving the packets reset their NAV. During the CFP if one polled station does not have anything to transmit, the PC will poll another in PIFS period of time. The stations need to register them with the PC through the *association management frame*. Figure 2.2 depicts the PCF/DCF rotation.

2.1.3 Limitations of 802.11b

QoS is the ability of a network element (e.g. an application, a host or a router) to provide some levels of assurance for consistent network performance in some network metric. It can be throughput. Delay, jitter, bit error rate individually or collectively.

With DCF, all the stations in one *Basic Service Set* (BSS) compete for the resources and channel with the same probabilities. There is no service differentiation mechanism to guarantee bandwidth, packet delay and jitter for high-priority stations or multimedia flows. Rather CSMA/CA channel access will support only best effort service quality. AP which holds more data than any other station will suffer if it needs to contend every time.

PCF comes with some guarantee by allocating time to stations that have real-time traffic. But its inefficient and complex central polling scheme deteriorates the performance of high priority traffic when the traffic load increases and registered stations have to wait for their polling slot. Second problem is the incompatible cooperation between the *contention period* (CP) and CFP modes which leads to unpredictable delays. Finally, the transmission time of the polled stations is not known. Also the stations do not provide any specification of their traffic which will not help PC to adopt a polling strategy based on any station's real time need. VoIP and all other traffic are stored in the same queue and are handled thereof in the same manner.

2.2 IEEE 802.11e

IEEE ratified the standard IEEE 802.11e [3] for a quality of service aware MAC protocol in November, 2005. This standard provides multimedia support and QoS features to the existing IEEE 802.11b and 802.11a. Its vision was to provide the home users and broadband service providers to have a quality aware multimedia network that will effectively deploy applications like audio on demand, video on demand and of course voice over IP (VoIP). In this regard, 802.11e allows differentiating the service for different traffic category based on different priority assignment but on top of the same CSMA/CA scheme.

For achieving QoS, 802.11e introduces two access mechanisms: the Enhanced Distributed Channel Access (EDCA) which is activated to delivers traffic based on differentiating *user priorities* (UP) and *Hybrid Coordination Function* (HCF) controlled Channel

Access (HCCA) which works during the controlled channel access. The EDCA manages the medium access in the CP (Contention period) while the HCCA is operable in both CFP and the CP.

The HCF combines functions from the DCF and PCF with some enhanced, QoS-specific mechanisms and frame subtypes to allow a uniform set of frame exchange sequences to be used for QoS data transfers during CP and CFP.. *QoS Station* (QSTA) may obtain transmission opportunities (TXOPs) using one or both of the channel access mechanisms. Each TXOP is defined by a starting time and a maximum length. The TXOP may be obtained by a QSTA winning an instance of EDCA contention (see 2.2.1) during the CP, or by a *non-AP QSTA* receiving a (QoS + CFPoll) frame (see 2.2.2) during the CP or CFP.

2.2.1 EDCA

The EDCA mechanism provides 8 different user priorities (UP). It also defines four access categories (ACs) that provide support for the delivery of traffic with UPs at the QSTAs. The access category (AC) is derived from the UPs as shown in Table 2.1.

Table 2.1: User Priority to Access Category Mapping in 802.11e

Priority	User Priority	Access Category	Designation
Lowest to Highest	1	AC_BK	BACKGROUND
	2	AC_BK	BACKGROUND
	0	AC_BE	BEST EFFORT
	3	AC_BE	BEST EFFORT
	4	AC_VI	VIDEO
	5	AC_VI	VIDEO
	6	AC_VO	VOICE
	7	AC_VO	VOICE

A model of the reference implementation is shown in Figure 2.3 and illustrates a mapping from frame type or User Priority (UP) to Access Category (AC); the four transmit queues and the four independent *enhanced distributed channel access functions* (EDCAFs), one for each queue.

In place of DIFS, 802.11e defines AIFS which is the *arbitrary inter frame spacing* and is at least DIFS long. The relationship of each priority and their inter frame spac-

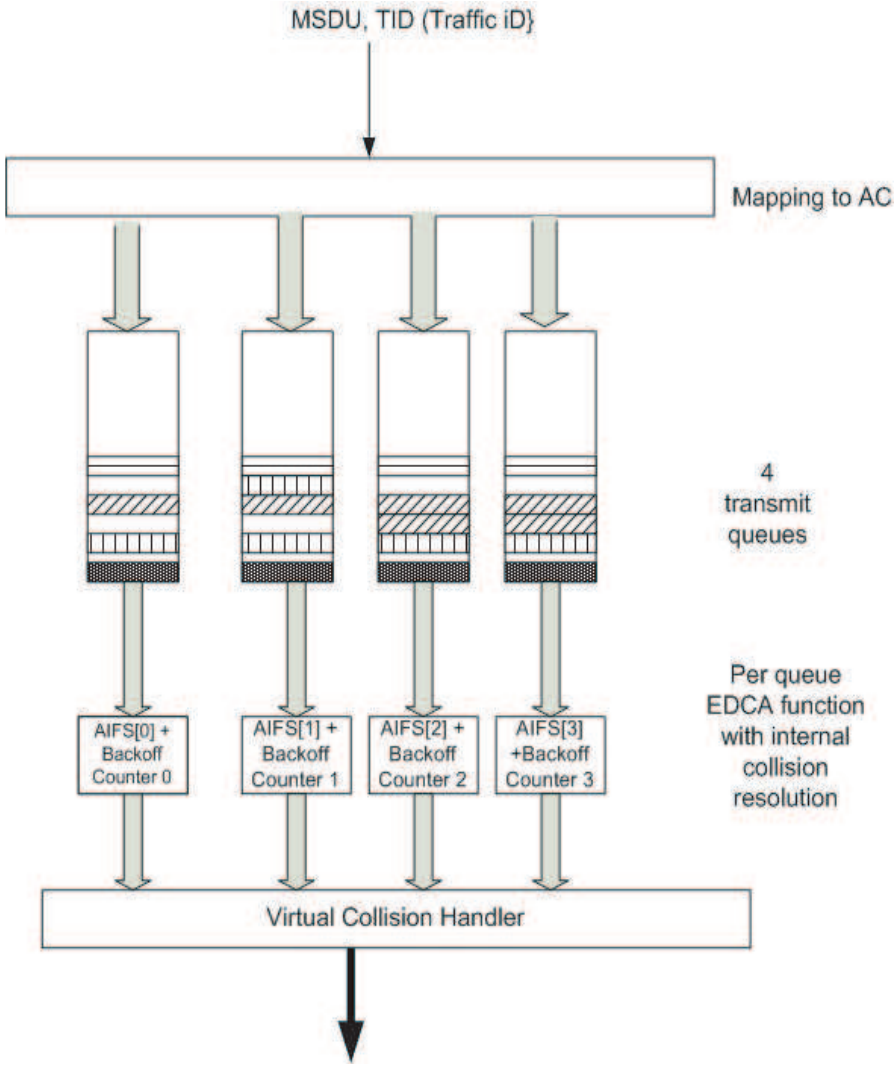

Figure 2.3: Reference Implementation Model

ing compared to other interframe spacing time can be seen in Figure 2.4. An AC uses AIFSD[AC], CW_{min}[AC], and CW_{max}[AC] instead of DIFS, CW_{min}, and CW_{max}, of the DCF. The values of AIFS[AC], CWmin[AC], and CWmax[AC] are announced by the AP via beacon frames. When an 802.11e station seizes the channel, it is entitled to transmit one or more frames for a time interval called *Transmission Opportunity* (TXOP); a TXOP is characterized by a maximum duration, called TXOP Limit. These parameters can be varied by the AP based on the current condition of the network to avoid collisions among traffic flows belonging to the same AC. The TXOP limit duration values are advertised by the *Quality of Service assuring AP* or QAP in the EDCA *Parameter Set Information Element* in Beacons and *Probe Response frames* transmitted by the QAP. Within every 802.11e station, a scheduler solves virtual collisions among the AC queues, always allowing channel access to the flow with higher priority (Figure 2.3). EDCA has been shown to do better than DCF in [9].

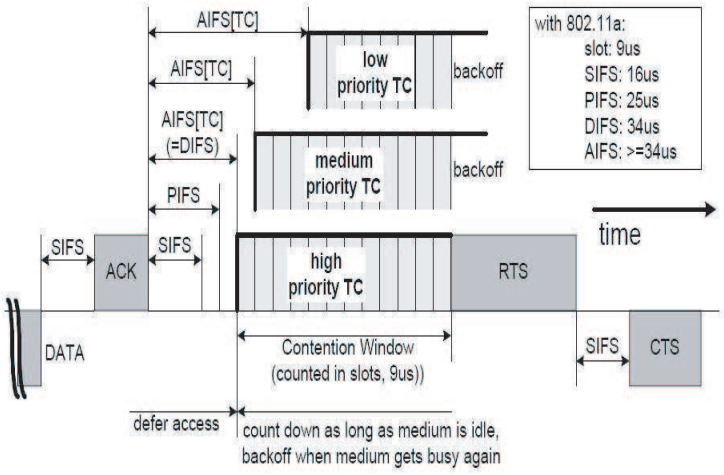

Figure 2.4: IFS relationship in 802.11e (Source [15])

- **Backoff and Retransmit Procedures :**
  Each EDCAF shall maintain a state variable CW[AC], which shall be initialized to the value of the parameter CW_{min}[AC]. If a frame is successfully transmitted by a specific EDCAF, indicated by either the successful reception of a CTS in response to an RTS or on the successful reception of an ACK in response to a unicast *MAC protocol data*

unit(MPDU), CW[AC] shall be reset to CW_{min}[AC]. The backoff counter is selected from [1 + CW[AC]] instead of [0,CW] as in the DCF.

QSTAs shall maintain a short retry counter and a long retry counter for each MSDU (MAC service data unit) or *MAC management protocol data unit* that belongs to a *traffic category* TC and requires acknowledgement. The initial value for the short and long retry counters is zero. QSTAs also maintain short retry counter and long retry counter for each AC and defined as QSRC[AC] and QLRC[AC], each initialized to a value of zero. The counters are incremented after retransmissions.

- **Packet frame Grouping (PFG):** With PFG, a series of small MSDUs are transmitted in a burst at SIFS interval after getting acknowledgement.Therefore the station keeps control over the medium for the whole burst. Sending multiple small packets in a burst avoids contention for each single packet. The PFG is mentioned in 802.11e as the idea of Contention Free Bursting (CFB). In the latest draft it is discussed as the idea of "Obtaining of a Continuation of EDCA TXOP". The time for which a station can consecutively send frames is limited by the Transmission Opportunity Limit (TXOPLimit). This results in a higher efficiency and lower delay as only the first frame needs to contend for the channel, the rest of the frames in the burst are separated by SIFS.

- **Realtime QoS Assurance of EDCA :** EDCA statically allocates priorities to flows irrespective of their current QoS requirements. Hard priorities does not always provide optimal results, especially with the changing requirements of real time traffic. EDCA has not been designed to provide QoS guarantees. It does not guarantee service to any flow once the number of nodes and flows increases. The throughput demanding traffic could also suffer in the presence of VoIP. Unfairness also results between the uplink and downlink flows when the AP is visualized as a node competing for channel access.

2.2.2 HCCA

The HCF controlled channel access mechanism uses a QoS-aware centralized coordinator, called *hybrid coordinator* (HC). The HC is collocated with the QoS access point (QAP) of the QBSS . The HCF has alternating CPs and CFPs forming super frames. Dur-

ing CFP, using CF-Poll frames, the HC can request nodes to inform their QoS requirements. CFP ends by a CF-End frame sent by the HC. During CFP, the TXOP limit is specified by HC in the poll frames. CP is essentially EDCA, although the HC can introduce *Controlled Access Periods* (CAPs), polling hosts to meet their QoS requirements. The CAP ends when the channel is idle for DIFS time when the hosts again switch back to operating in EDCA. During CP, each station gets its TXOP either when the medium is determined to be available under the EDCA rules or when the station receives a poll frame from HC. HCF is more flexible than PCF allowing CAPs to occur anytime during the superframe. The HC may allocate TXOPs to itself to initiate frame transmission after waiting for a time equal to PIFS, which is shorter than DIFS as well as any of AIFS. Thus, HC gets priority over all other stations in medium access.

- **Real time QoS Assurance of HCCA:** There are concerns with HCF's mechanism of getting QoS information updates from STAs. The STAs will get an opportunity to send out a resource request frame during the CFP or when they get an opportunity during the CP. These opportunities are closely based on the requests sent during the CFP. These problems have been highlighted in [4] also. We observe that the problem worsens when coupled with the fact that HC has a greater load of traffic on the downlink, which gives itself higher priority in comparison to the uplink flows.Moreover, HC prioritizes STAs without distinguishing the flows from a STA. As a result, all flows from a STA are taken to be of same priority by the HC. This is not true in reality.Assigning per-station priority can favor the STA with less QoS demanding traffic over the STA with higher QoS demanding flows.

2.2.3 Re-engineering 802.11

EDCA requires strict traffic parameters associated with the different traffic types. In [15] it is shown that with non-overlapping CW, there is an undesirable drop of throughput for the low priority streams. In [13], the authors show that even in the presence of EDCA, the high priority ACs experience deteriorating service quality when the ingress traffic in the BSS approaches saturation. The uplink and downlink unfairness problem has been addressed in [7] with different EDCA parameters at the AP and mobile hosts.

In [4], the authors propose Extended PCF (EPCF) to support VoIP and to provide hard QoS guarantees. They implement a centralized controller at the AP to schedule flows

as per the per packet delay requirements of real time traffic flows. They also evaluate the performance of VoIP traffic in the presence of other traffic types and show that they perform better than HCF in terms of average delay and system throughput. EPCF has not been studied with QoS requirements of downlink traffic, which is important considering that AP is the bottleneck for downlink flows. The polling mechanism is based on uplink traffic requirements i.e. the downlink DATA+Poll packet is sent to a node which has the nearest deadline. The downlink traffic deadlines are ignored. The active information update has been proposed in [17] where the future channel request is piggybacked with the current packet. This update is used by the AP while polling the hosts in the next polling cycle.

We also aim to design a fair scheduling algorithm at the AP. There are several issues involved in providing fair channel access among multiple contending nodes with multiple flows in WLANs. In [5] the authors cite location-dependent and bursty errors, channel contention, and joint scheduling of uplink and downlink flows as some of the issues to be addressed. The previous works mentioned above are not taking into the consideration the imbalance in the traffic size of uplink downlink or if they address the problem its through some random feedback based approach. In these works AP will wait to know about STAs status in a very unsure manner and consequently the STAs will fail to provide QoS to delay sensitive flows. Equally important is to measure the throughput demand of the Best effort traffic. As we have pointed out in the earlier sections, their contention degrades real time STAs. ARC addresses these issues without too much of protocol overhead.

Authors in [6] proposes the Wireless Fair Service (WFS), an adaptive fair queuing algorithm, which provides compensation for lagging flows and graceful degradation of leading flows. Finally it achieves both short term and long term fairness with delay and throughout bounds. Bharghavan et al. illustrates the WFS as an approach to resolve the above mentioned issues in [5]. They also study the performance of Error-Sensitive Vs Delay-Sensitive flows.

A fair dropping algorithm called DS-CFD(Diff-Serv Supported Channel Sensitive Fair Dropping)was proposed in [8]. It implements a drop policy in wireless networks when congestion occurs. The algorithm considers both channel condition and fairness in order to achieve tradeoff between throughput and fair services, and supports different levels of balance to assured services and best effort services. However, they do not study the performance and trade-offs of delay-sensitive traffic in the presence of error-sensitive traffic and vice-versa.

An approach to control and provide fairness to the several flows which pass through the wireless medium is to adaptively adjust the contention parameters (Contention Window and Inter-Frame Space Duration). Kravets et al. proposes a new cross layer framework, named QPART (QoS Protocol for Ad hoc Real time Traffic) in [19], which provides QoS guarantees to realtime multimedia applications for wireless ad hoc networks by adapting the contention window sizes at the MAC. QPART implements priority-based admission control and conflict resolution to ensure that the requirements of admitted real time flows are smaller than the network capacity. QPART is robust to mobility and variances in channel capacity and imposes no control message overhead on the network.

Piggybacking scheduling table during the RTS-DATA-ACK handshakes is one of the ways that neighboring STAs can know from each other about their scheduling tables. But 1. this involves heavy protocol overhead and 2. the carrier sensing range is not taken into the consideration before scheduling. Soft state conflict resolution protocols like INSIGNIA [14] also rely heavily on achievable local service level prediction and message exchange among the intermediate nodes on the route. As the estimation of available bandwidth is done locally these works do not consider how the available bandwidth is reduced, if a node in the carrier sensing range starts transmitting. In case of proportional delay differentiation [18] the normalized waiting time of the packet waiting in the front of MAC queue must be broadcasted to all the nodes. This involves lot of latency which can violate delay deadline of VoIP or make the streaming jittery. Our effort will be to first decide on inter node resolution based on the MAC feedback of congestion from stations. We propose to piggyback the future channel access information in both DATA and CONTROL packets. Also, special management packets are not required to handle the scheduling.

Chapter 3

ARC Scheduler

In this chapter we describe the ARC (Access Point Coordinated Contention Resolution for Channel Acess) scheduler and the principles driving its design.

3.1 System Model and Design Overview

Consider a wireless network setup in which all STAs are 802.11e compliant. In 802.11 infrastructure mode (Figure 3.1), AP is acting as a gateway of all the upstream transmissions and transmitting all the downstream packets to mobile STAs. Thus AP is omniscient of all stations' traffic types and of the contention over the shared wireless channel. In order to take advantage of this fact, the ARC scheduler relies on the AP as an arbiter to coordinate channel access between the STAs. Efficient arbitration between the flows requires up-to-date information about the STA flows. This information is communicated to the AP in all data and ACK packets sent from the the STAs to the AP. Arbitration also requires channel assignment decisions, which are embedded in data and ACK packets sent from the AP to the STAs, and overheard by all STAs. Furthermore, in order to avoid long delays for STAs that are not able to get access to the channel and thus cannot report their congestion information to the AP, the AP relies on additional POLL packets, if needed, to contact the deprived STAs.

At the heart of the ARC scheduler is the congestion information maintained at

the AP, in a data structure which we term the *congestion table*, and the logic used to pick the STA that *deserves* to access the channel in the next time slot. This logic is triggered at the AP once it is about to send a data or an ACK packet and a decision is made as to whether to assign the channel to one of the STAs (or to the AP itself), to POLL a deprived STA, or to simply let the STAs resume their EDCA contention.

Overhearing a channel assignment decision, say in an ACK packet transmitted by the AP, the STA which has been assigned the channel acts accordingly and transmits a packet within shortest inter frame spacing (SIFS) time and without contention from other STAs. The 802.11 standard [2] helps to avoid contention with the STA assigned the channel as it requires STAs to wait for *DCF interframe spacing* (DIFS) DIFS time resuming contention. Also, by overhearing the channel through the CSMA/CA meachanism, STAs are not supposed to transmit simultaneously, unless a *hidden* node exists [12]. In this sense, ARC manipulates the EDCS standard and does not change it.

Figure 3.2 displays a timing diagram for the selected ARC node. As shown in the figure, while acknowledging STA C, AP assigns the next time slot to STA D. STA D accesses the channel in the next SIFS time instead of waiting for DIFS time and for its backoff timer to expire. Eventually this leads to better average delay as STAs will aceess channel avoiding the extra latency. This technique also allows the AP to acquire the channel for itself whenever it is needed by accessing the channel in SIFS time after the last successful channel transmission. This guarantees the timely dispatch of downlink packets.

In our model, VoIP is used as the representative real-time application. VoIP is bidirectional and hence we have both upstream (from STA to AP) and downstream (from AP to STA) VoIP flows. Along with VoIP flows, wireless nodes will also receive or send ftp as a best-effort traffic. The later is used to highlight the impact of best effort traffic on realtime flows and to stress test ARC performance.

3.2 Design Considerations

All data and ACK packets from the STAs carry specific fields to provide feedback to the AP about their real-time and best effort flows. The fields included in the MAC/ACK header are:

- Next VoIP packets current queuing delay.

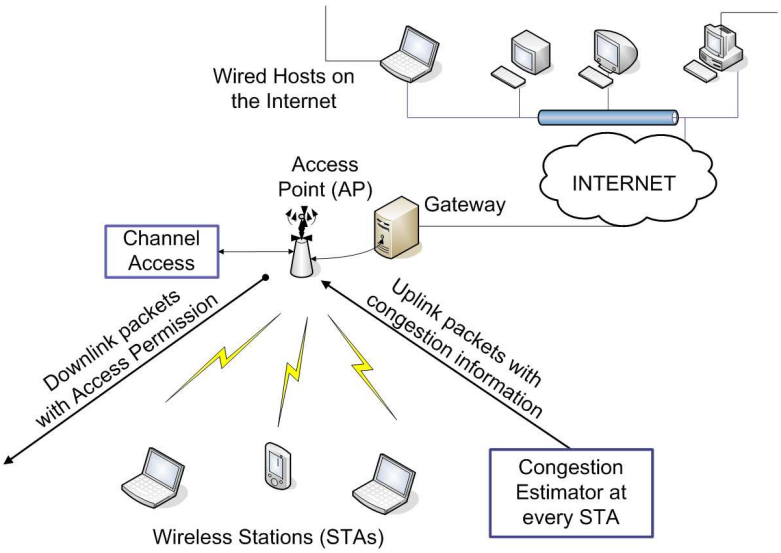

Figure 3.1: 802.11 Infrastructure mode

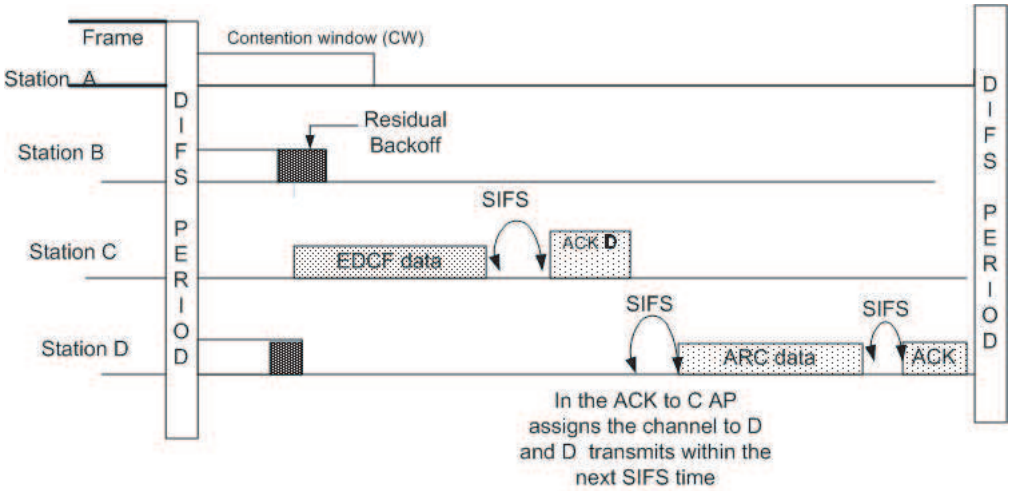

Figure 3.2: Timing Diagram of ARC

Table 3.1: Congestion Table Entry

Node ID	Last reporting time	VoIP Queuing Delay	VoIP queue size	FTP Queue size

- VoIP queue size
- FTP queue size

More details about these fields are included in the Appendix. The AP is able to measure the queuing delay of the next packet to be dispatched from each flow using the queuing delay reported by each STA reported and the time information at the AP. The AP also updates its own status in the table whenever the ARC logic is triggered.

The congestion table at the AP holds this information for each flow type at each STA and for the each flow type at at the AP itself. Furthermore, the congestion table maintains the time it last heard from each STA. This time is used to trigger POLL messages as we elaborate next. Table 3.1 shows the format of one entry of the congestion table and the Appendix provides the code for the definition of the congestion table. Once an entry, corresponding to one STA, has been stored in the table it will be refreshed on each update from the STA.

In addition to the congestion information of each flow type at every STA, the ARC logic is tuned by three three thresholds

- Polling threshold (ms): The polling threshold bounds on the time spent without contacting any STA. Upon violation of this threshold for some STA, the AP explicitly sends a POLL message to this STA, which inserts its congestion information in the next packet it transmits. The AP maintains information about the last-heard-of time from every STA in order to trigger polling, when needed.
- Queuing Threshold (ms): The queuing threshold bounds the queuing delay of VoIP flows, or delay-sensitive flows in general. Upon violation of this threshold for some flow, the AP assigns the channel to the appropriate STA handling this flow.
- FTP Threshold (Number of packet): The FTP threshold bounds the number of queued packets for any FTP flow. Upon violation of this threshold for some flow, the AP

assigns the channel to the appropriate STA handling this flow.

3.3 ARC Scheduler at AP

Figure 3.3 shows the overall view of the ARC logic at AP. Once the ARC algorithm is run at AP, the outcome is a STA ID or -1. In case of -1 the scheduler algorithm did not find a congested node.

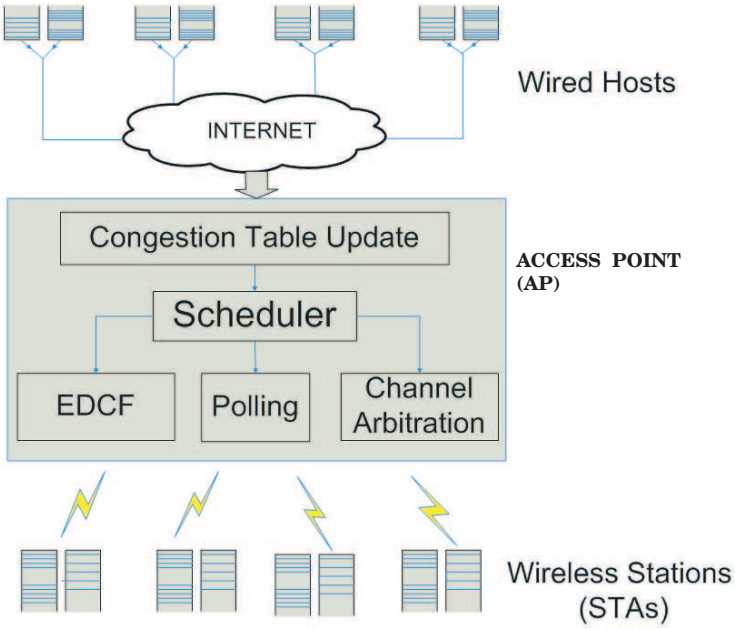

Figure 3.3: ARC Scheduler at AP

Also it can be noted that the scheduler algorithm can return AP as the highest congested and later AP will tell others about winning the channel by putting its ID in the ACK header. The algorithm is illustrated in the following pseudo-code.

1: **procedure** CONGESTIONUPDATE($nodeID$, $congestionENTRY$)
2: ▷ insert/update the congestion table with congestion information from STA
3:

```
4:    if (new STA information) then
5:     insert congestion fields from ENTRY at the mapped value for the ID
6:     insert the current simulation time in the last reporting time field of the same entry
7:    else
8:     update the fields for the corresponding ID with the values in the ENTRY
9:    end if
10: end procedure

11: function scheduleFlow(CongestionTable)
12:                 ▷ returns NODE ID if successful or -1 if no node is congested
13:    for (all STA) do
14:          extract POLLED NODE with the maximum value for (current time -last
           reported time)
15:    end for
16:    if (maximum value gt POLLING_THRESHOLD) then
17:     return POLLED NODE
18:    else if (POLLED NODE == -1) then
19:       for (all STA ) do
20:          if (the next VoIP packets reported delay gt 0.0) then
21:              calculate the current queuing delay = Current time - Last reported time
                 + reported queuing delay
22:          end if
23:          Find the Qd_NODE with maximum current queuing delay which has value
                 gt QUEUING_THRESHOLD
24:       end for
25:     return Qd_NODE
26:    else if (Qd_NODE == -1) then
27:          Find a FTP_NODE that has the maximum FTP queue size gt
             FTP_QUEUING_THREHSOLD
28:     return FTP_NODE
29:    else
30:          return -1
31:    end if
```

32: **end function**

The flowchart for the algorithm SCHEDULE FLOW() is shown in Figure 3.4. Once CONGESTION UPDATE() is called, the AP will schedule if it has an ACK or a POLL packet to send.

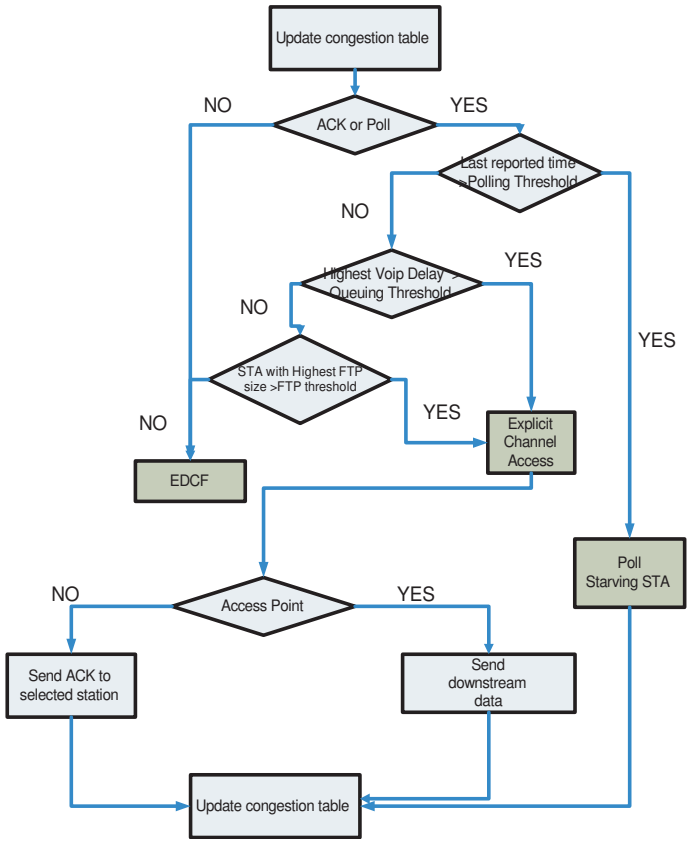

Figure 3.4: Flowchart of ARC Scheduling Algorithm at AP.

As seen from the Figure 3.4, and the pseudo code of SCHEDULE FLOW(), the AP will assign the channel to a STA that violated the polling threshold, to a STA that has a VoIP packet that violated the queuing threshold, to a STA that has an FTP packet which violated the FTP threshold. Otherwise, no STA is explicitly assigned the channel and access to the channel falls back to 802.11 EDCA.

```
1: procedure sendACK(destID, priority)
2: ...
3: ...
4:
5:     if (current sender does not have any VoIP packet that are waiting gte
           QUEUING_THRESHOLD) then
6:      SCHEDULE_FLOW(      )
7:     else
8:      Allow the same station to transmit another VoIP packet
9:     end if
10:    if ( SCHEDULE_FLOW returns AP) then
11:     transmit a packet after the ACK packet in SIFS time
12:    else if SCHEDULE_FLOW returns some STA then
13:     Schedule STA with ACK packet
14:    else
15:     next slot will be according to EDCA
16:    end if
17: ...
18: ...
19:
20: end procedure
21:
```

When AP is sending downlink DATA, it's not possible for it to inform other STA by ACKs. Therefore, it will enact the scheduling decision by broadcasting a small polling packet which the STAs will decipher to determine who will get the channel back from AP. The pseudocode is shown below.

```
1: procedure recvACK(destID, priority)
2: ...
3: ...
4:     SCHEDULE_FLOW(      )
5:    if (returned node  gt -1 and returned node ≠ AP) then
```

```
6:      SendPoll(      )              ▷ (Send poll will contain the node that's selected)
7:    else if AP has a VoIP or FTP packet to send then
8:      AP will schedule its own packet for transmission within next SIFS time
9:    else
10:     next slot will be according to EDCA
11:   end if
12: ...
13: ...
14:
15: end procedure
```

3.4 Scheduler at STAs

The STA will determine whether it has been granted the channel once it overhears/receives the ACK or POLL packet. Once it gets the channel, it will run the following procedure:

```
1: function Algorithm packetSchedule()
2:   if (channel access or polled) then
3:         Send packet(Marked with current congestion information)
4:   end if
5:   if (Recv packet) then
6:         Send ACK(Marked with current congestion information)
7:   end if
8: end function
9:
```

When calculating the congestion, the STA will look into the VoIP queue and measure the current delay of the next VoIP packet after the packet in the head. Once the node transmits data it will provide the AP with the up to date congestion information. Each packet will be time stamped when they enter the MAC queue. This helps us to estimate the queuing delay of each packet. Figure 3.5 shows the diagrammatic view of the algorithm and Figure 3.6 illustrates the congestion estimation technique.

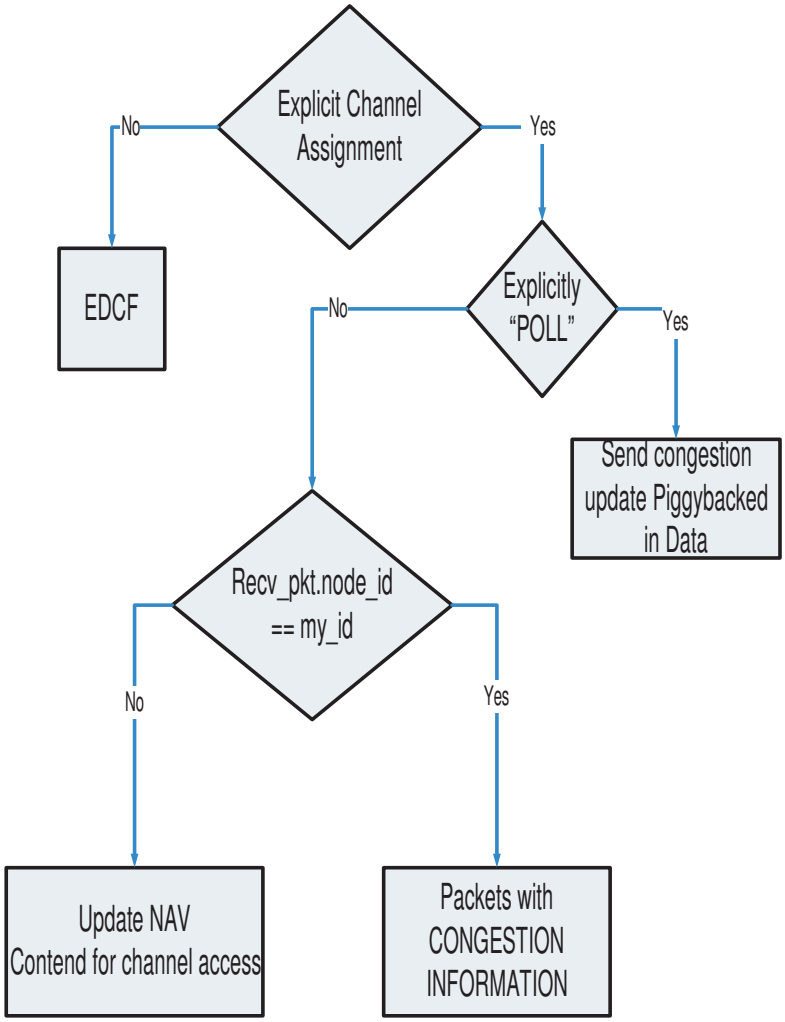

Figure 3.5: Flowchart of Scheduling Algorithm at STA

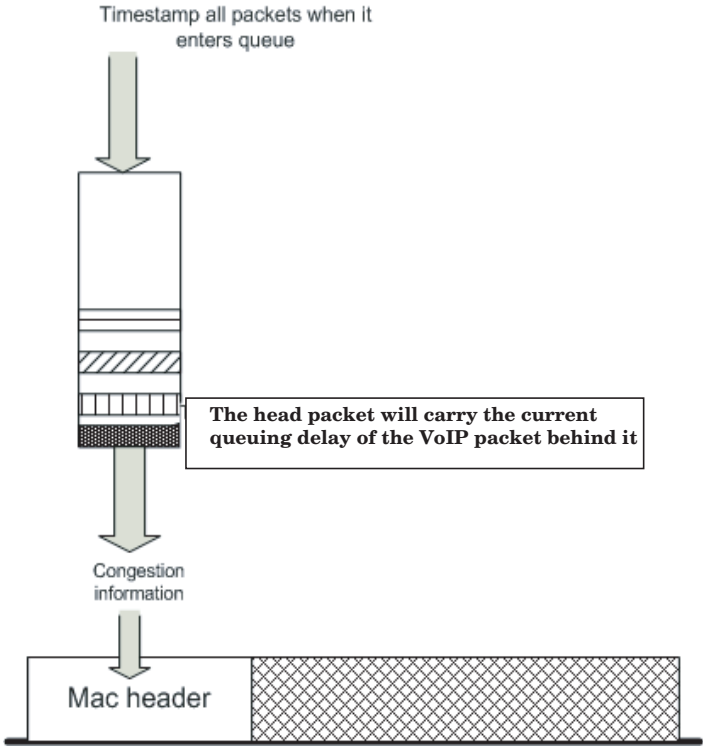

Figure 3.6: Congestion information piggybacking at STA.

Chapter 4

Experimental Results

We have simulated the ARC scheduler in Network Simulator-2 (NS-2) [1]. The simulator is written in C++ and various well known C++ protocol modules are provided that work in the back-end of TCL front-end interface. To implement ARC, we patched the NS kernel with the TKU Berlin implementation of IEEE 802.11e module [17]. In this chapter we provide the details of the simulated scenarios and comment on the results of each test case.

4.1 Performance Metrics

For real time traffic like VoIP; the scheduler has to guarantee both low drops and low queuing delay (well below the 150 ms deadline [11]). We therefore report on the scheduler performance for VoIP flows in terms of the following performance metrics:

- *Average Queuing Delay for AP*
- *Average Queueing Delay for STA*
- *Transmitted/Successful/Dropped* packets

802.11e fair access mechanism wastes a lot of channel bandwidth because of random access and thereby collisions among the STAs. To highlight this problem, we report on the channel *utilization* and the *useful channel utilization* compared to EDCA. Also, we report on FTP

traffic performance by showing the number of successful packets. This number indicates the total packets that are acknowledged by MAC. As TCP acknowledgement in cumulative, therefore this metric gives us a fair indication of how unbiased the scheduler is for Best Effort traffic.

In order to understand the inner works of our scheduler, we provide statistics explaining ARC scheduling decision by showing a breakdown of the numbers of channel assignments based on each ARC threshold. This allows us to explain the resulting performance and determine how the ARC thresholds should be tuned to improve performance.

4.2 Simulation Details and Test Cases

We use constant bit rate UDP flows with 64kbps data rate to represent the VoIP flows. FTP transmission rate is 1 MBPS (NS default) and the FTP queue is of infinite size whereas the VoIP queue has a limited size. The reason for this is that VoIP is bursty in nature and most of the time the queue remains empty. VoIP has minimum delay requirement and therefore is always guided to the AC queue 0 which is the highest priority queue out of the 4 queues of 802.11e MAC.

Another important test variable is the data rate that the channel will support. Typically the 802.11b offers 11 MBPS of data bandwidth and 2MBPS of basic/ctrl packet bandwidth. But as 802.11e is backward compatible with *802.11a* (which goes all the way up to 54 MBPS) we have also tested on this high bandwidth. In the Appendix we provide the changes in various parameters that will allow us to simulate a 802.11e MAC module compatibly with 802.11a *BSS*.

In the simulations we had the following prerequisites :

- the 802.11e MAC is running on the basic CSMA/CA scheme without the RTS (ready to send)/CTS (Clear to send) extensions.
- STAs are closely situated surrounding the AP, which means there is no hidden station problem. So STAs are static and can overhear any data or ACK packet sent between AP and any other node.
- MAC ACK is optional i.e. it can be switched off if needed to improve performance. But for ARC, MAC ACK is always enabled.

Table 4.2 shows in detail the test setup in terms of the test variables.

Table 4.1: Experimental Setup and Notation

Case#	# of STAs	FTP Direction	Channel Data Rate
Case 1	10	AP to STA	11 Mbps
Case 2	10	STA to AP	11 Mbps
Case 3	20	AP to STA	11Mbps
Case 4	20	STA to AP	11Mbps
Case 5	40	AP to STA	54Mbps
Case 6	40	STA to AP	54 Mbps

4.3 Simulation Results

For brevity, we define the following shortened abbreviations of the ARC thresholds and assignment criteria. These are shown in Table 4.2. For each of the test cases we provide two tables. In the first one will highlight the breakdown of the scheduling decisions and in the later will detail the overall system performance. Each of these test cases has subcases depending on the selected values of the ARC thresholds.

Table 4.2: Scheduler Terminology

Constant Name	Term Used
Polling Threshold	Const1
Queueing delay Threshold	Const2
FTP Queue Size Threshold	Const3
Polling	Cond1
Queuing Threshold Violation	Cond2
FTP Queuing Threshold Violation	Cond3

4.3.1 Case 1

From Table 4.1 this simulation tests our scheme in lightly loaded scenario where there is much less contention and 10 VoIPs can be supported both ways by 11 Mbps. The scheduler configurations are shown in the following table.

Table 4.3: Case 1: Scheduler Configuration and Channel Assignment Results

Test#	Const1 (s)	Const2 (s)	Const3 (packets)	Assigned	unsuccessful	Cond1	Cond2	Cond3	EDCA (runs)
1.1	0.02	0.005	100	195711	11811	43349	60526	91836	31539
1.2	0.02	0.005	200	198701	6211	40506	62716	95476	27911
1.3	0.04	0.005	100	202877	2223	15832	86915	100130	8373

One key point to notice from this table is that if Cond1 is increased because of small Polling Threshold, EDCA runs will increase. Once any station is polled, but it does not have anything to send, then this will result in an unsuccessful assignment. It will prolong time to win the channel in a regular DCF way for others. For this reason, both 1.1, 1.2 has higher unsuccessful assignment than 1.3.

Table 4.4: Case:1 Performance Comparison of ARC

Scheme	Utilization (Channel)	Useful Util. (Channel)	Voip			FTP			Average Voip Delay (AP) (s)	Average Delay VoIP (STA) par (s)
			Transmit	Success	Drop	Transmit	Success	Drop		
1.1	0.835	0.823	108590	97405	11189	106763	98385	8142	0.003	0.0455
1.2	0.855	0.8447	111146	98656	12494	109171	101790	7184	0.003	0.0328
1.3	0.86	0.857	102524	98713	3859	106440	104481	1947	0.0028	0.0611
EDCA	0.8629	0.80957	124699	99234	25465	107636	94243	13334	0.00197	0.00225
CFB	0.869605	0.82323	120338	97913	22428	109084	97850	11212	0.00150	0.0023

The performance from Table 4.4 shows the moderate improvement in the VoIP drops for 1.3. With increasing Const1, polling (Cond1) is reduced. This will eventually allow the stations to be scheduled more on the basis of their reported information. Therefore in 1.3 we have an increase in the number of assignments which ramifies with increase in useful utilization and lesser number of drops. On the other hand, lesser EDCA runs result in lesser collision drops.

Our scheme transmits approximately the same aggregate traffic as of EDCA and CFB but with greater *goodput*.

4.3.2 Case 2

With STAs sending both VoIP and FTP, higher number of polling under Cond1 will prove effective. Hence we have seen a considerable improvement from 2.1 in 2.2. Each STA will have much less FTP individually compared to Case1, where AP downstreams 10 FTP flows. So we have set smaller values for Const3.

Table 4.5: Case 2: Scheduler Configuration and Channel Assignment Results

Test#	Const1 (s)	Const2 (s)	Const3 (packet)	Assigned	unsuccessful	Cond1	Cond2	Cond3	EDCA runs
2.1	0.02	0.005	30	208925	3679	76209	70882	61834	34688
2.2	0.01	0.005	30	219348	1170	112330	97746	9308	9122

The performance in Table 4.6 attributes the improvement in 2.2 to more successful execution of Cond1 . Also it helps to get more VoIP queueing information from the STA and increases the scheduling based on Cond2. As a side effect, increased polling will cause APs traffic to wait longer.

4.3.3 Case 3

On a heavily loaded scenario, scheduling more under Cond2 and 3 will be more accurate. With small Const1 some STAs will be polled excessively without much success. Therefore increasing Const1 will lead to more decision making based on Cond2 and 3. As for Const3, a small value will make the AP schedule more based on FTP traffic.

But it (in 3.5) also means that AP will leave the channel to EDCA. This means the STA will wait less and hence their average delay for VoIP improves over (3.3) as found in Table 4.8. In 3.5, the AP will transmit and drop more resulting in worse useful utilization of the channel. The VoIP transmission will increase as AP will use the opportunity of Cond3-assignment for transmitting its VoIP packet first. But as Cond1/polling executes more, delay for AP starts to increase. Increasing Const1 will prevent hearing back from the STA which increases the STAs delay (3.4).

Table 4.6: Case:2 Performance Comparison of ARC

Scheme	Utilization (Channel)	Useful Util. (Channel)	Voip			FTP			Average Voip Delay (AP) (s)	Average Delay VoIP (STA) (s)
			Transmit	Success	Drop	Transmit	Success	Drop		
2.1	0.91	0.895	113997	97957	16061	126048	112725	13486	0.00635	0.0342
2.2	0.9448	0.94	101090	97990	3100	126357	121860	4497	0.1456	0.0576
EDCA	0.853	0.734	148421	98818	49669	123601	79410	44264	0.6945	0.00411
CFB	0.867175	0.762275	140055	98238	41924	130476	85188	45410	0.0027	0.0053

Table 4.7: Case 3: Scheduler Configuration and Channel Assignment Results

Test#	Const1 (s)	Const2 (s)	Const3 (packet)	Assigned	unsuccessful	Cond1	Cond2	Cond3	EDCA runs
3.1	0.01	0.01	200	191887	72976	136968	5583	49336	144977
3.2	0.04	0.01	200	144761	12587	68003	47478	29280	324605
3.3	0.04	0.005	200	225498	5792	37958	87351	100189	25878
3.4	0.08	0.005	200	235426	11179	27593	113965	93868	34775
3.5	0.04	0.005	100	214715	27267	59639	79298	75778	126509

4.3.4 Case 4

As with any other cases if polling is increased by decreasing Const3 (4.3), the resultant effect will be more EDCA runs (from Table 4.9) and improvement in the average queuing delay of VoIP for the STA. because more VoIP packets are scheduled (in Table 4.10). If we increase Const1, more Cond2 and Cond3 execution will increase accuracy of the scheduler and thereby the channel useful utilization improves. This means STAs are scheduled more by ARC rather than allowing them to contend. It reduces retransmission due to collision drops as observed in 4.2 and 4.3.

4.3.5 Case 5

In case of high bandwidth we increase the number of STAs to 40 as there will be enough bandwidth to support the aggregate bandwidth demand. From ARC's point of view, this means AP will always hear from some STA about its VoIP/FTP transmission. For this reason, the congestion table will always have some STAs to be scheduled based on

Table 4.8: Case:3 Performance Comparison of ARC

Scheme	Utilization (Channel)	Useful Util. (Channel)	Voip			FTP			Average Voip Delay (AP) (s)	Average Delay VoIP (STA) (s)
			Transmit	Success	Drop	Transmit	Success	Drop		
3.1	0.732535	0.721455	251079	192934	59482	13341	9008	4330	0.00635	0.03420
3.2	0.732535	0.721455	431968	185691	255627	25864	17052	9090	0.00738	0.0422
3.3	0.894	0.9	211037	193218	16820	46251	42813	3502	0.0040	0.08615
3.4	0.89206	0.8875	212813	193545	19268	46130	42436	3728	0.00432	0.1079
3.5	0.83	0.82	274771	186868	92278	39309	33396	5913	0.00547	0.07056
EDCA	0.906	0.73	340858	174501	172491	37102	24113	13014	5.21519	0.01285

Table 4.9: Case 4: Scheduler Configuration and Channel Assignment Results

Test#	Const1 (s)	Const2 (s)	Const3 (packet)	Assigned	unsuccessful	Cond1	Cond2	Cond3	EDCA runs
4.1	0.04	0.005	30	211554	2258	82952	128147	455	163313
4.2	0.08	0.005	30	217979	1591	63731	154248	0	136998
4.3	0.08	0.005	20	225503	3955	66547	147128	11828	101412

their VoIP's requirement. That's why in both 5.1 and 5.2 we have kept Const1 high enough to reduce unnecessary polling. Also, as there will be multiple candidates at the same time that can be scheduled under Cond2, we observe Cond3 to be executed very less number of times. Table 4.11 gives the breakdown of the ARC's scheduling decision. It's evident that the configuration is allowing more EDCA runs. As a result, we see greater number of collision drops in Table 4.12. Changing the thresholds as in 5.2 does not improve the goodput for VoIP and also increases the average queuing delay of the VoIP packets for the STAs because of the higher unsuccessful assignments resulting in higher EDCA runs. But AP's delay improves compared to 5.1 as Cond3 execution allows better FTP throughput.

4.3.6 Case 6

With FTP being transmitted form STAs; lesser EDCA runs improve the goodput and useful channel utilization, which causes the average delay for the STA become higher. If polling is increased, any new STA will be unable to contend for the channel because Cond1 helps to poll STAs that are already known to be transmitting. Consequently new STAs can not inform the AP of their status. But when more Cond2 is commanding the channel

Table 4.10: Case:4 Performance Comparison of ARC

Scheme	Utilization (Channel)	Useful Util. (Channel)	Voip			FTP			Average Voip Delay (AP) (s)	Average Delay VoIP (STA) (s)
			Transmit	Success	Drop	Transmit	Success	Drop		
4.1	0.868475	0.841445	321238	194240	128283	51699	32547	19666	0.0072	0.051
4.2	0.88	0.86	300426	193374	109166	53177	37039	16645	0.00712	0.0726
4.3	0.894	0.8775	270944	193518	79253	52270	40524	1219	0.0082	0.086
EDCA	0.8812	0.7332	302515	173784	131660	34770	23663	11144	4.8834	0.0069

Table 4.11: Case 5: Scheduler Configuration and Channel Assignment Results

Test#	Const1 (s)	Const2 (s)	Const3 (packet)	Assigned	unsuccessful	Cond1	Cond2	Cond3	EDCA runs
5.1	0.08	0.02	400	352928	5987	165357	187149	422	365978
5.2	0.12	0.015	300	333466	9057	125614	203074	4778	489704

assignment (as found from Table 4.13), there will be lesser possibility for the starving STAs to get a chance to communicate with the AP. Therefore, the polling logic/Cond1 needs to be devised based on previous history of successful assignment for each particular STA. We leave the reengineering of the polling logic to future work.

Table 4.12: Case:5 Performance Comparison of ARC

Scheme	Utilization (Channel)	Useful Util. (Channel)	Voip			FTP			Average Voip Delay (AP) (s)	Average Delay VoIP (STA) (s)
			Transmit	Success	Drop	Transmit	Success	Drop		
5.1	0.855	0.84592	709816	354352	389928	3381	1342	2203	0.0141	0.1021
5.2	0.82	0.80	807707	333181	474526	6961	3067	3894	0.011	0.108
EDCA	0.922	0.4523	1063345	177518	960020	24948	11109	14376	9.2308	0.00712

Table 4.13: Case 6: Scheduler Configuration and Channel Assignment Results

Test#	Const1 (s)	Const2 (s)	Const3 (packet)	Assigned	unsuccessful	Cond1	Cond2	Cond3	EDCA runs
6.1	0.1	0.01	400	374934	2846	106807	268127	0	158357
6.2	0.08	0.02	400	373669	1592	129252	244417	0	138814

Table 4.14: Case:6 Performance Comparison of ARC's

Scheme	Utilization (Channel)	Useful Util. (Channel)	Voip			FTP			Average Voip Delay (AP) (s)	Average Delay VoIP (STA) (s)
			Transmit	Success	Drop	Transmit	Success	Drop		
6.1	0.8964	0.893	527713	375478	152235	3040	453	2587	0.04643	0.258
6.2	0.9045	0.9029	512501	379473	133028	2758	314	2444	0.04732	0.2564
EDCA	0.90	0.468	1017921	195797	891695	2234	881	1410	7.79513	0.00683

Chapter 5

Conclusions and Future Work

In this concluding chapter, we highlight our major findings, pinpoint some of the deficiencies in our scheme and plan how future research should be guided to mitigate the shortcomings.

5.1 Conclusions

Our results show that:

- ARC channel coordination is capable of improving the channel utilization and less packet drops especially as the number of STA increases and contention over shared channel is prevalent, compared to 802.11e EDCA contention mechanism. This feature is crucial as new high bandwidth technologies are introduced such as WiMax. When the number of competing STAs is small, the performance is reasonable and the need for channel arbitration is not urgent.
- Compared to EDCA, ARC avoids the bias against the AP.
- Channel arbitration in ARC can increase the queuing delay at the STAs of which the AP is not aware. EDCA allows STAs to content and on average may get the channel sooner but at the cost of many packet drops and lower channel utilization.

- More polling, lower polling threshold, especially when the number of competing STAs is large is not effective as the many polled STAs may not have packets to send. Balancing the polling rate is a crucial enhancement that needs to be incorporated in ARC.

- ARC falls back to EDCA when maintained congestion information about the STA flows is not sufficient to make an informed decision, which results in packet losses and contention. A more proactive mechanism to report congestion information to the AP is needed. Note that polling STAs in order will not help as it waste channel cycles. This enhancement is more crucial when the number of competing STAs is large.

5.2 Future Work

- Polling logic in ARC is based on one threshold. Polling will be more accurate if any STA is polled based on its profile of traffic load and QoS demand. We can tune the polling threshold with a sophisticated logic given the information the STAs are reporting. But so far we have been assigning a fixed polling threshold.

- The order of the scheduling decisions (Cond1, Cond2 and Cond3) can be adaptively chosen depending on the information AP has stored and the runtime statistics on various metric we have shown in the result.

- Our scheme is tested on a link with no error and no hidden station problems. ARC should be tested in these setups.

- To support compatibility across 802.11x networks, the MAC header can be redesigned to use the reserved subtype fields in 802.11 MAC header.

- Comparison with HCCA will surely pinpoint further areas of improvement. But for the lack of a HCCA NS-2 patch that conforms to the standard and also support the DCF traffic bidirectionally we were unable to simulate it.

- Our polling (i.e.condition 1 assignment) is not discretionary with regards to the QoS requirements. In other words, if a STA is polled and it has no VoIP packet, then that particular STA will transmit FTP or other best effort traffic if possible. This will

hamper real-time flows' performance mostly in the uplink scenarios where the STAs have both VoIP and FTP flows.

- Delayed Packets above an acceptable threshold can be safely dropped. But the first 15 packets in an unvoiced to a voiced transition needs to be preserved for quality of VoIP as shown in [10].

- ARC should be compatible with the 802.11 standard. Testing the performance of ARC in a setup which has non-ARC compatible STAs is also one of our future goals.

Bibliography

[1] Network Simulator NS-2. Network simulator home page available at http://www.isi.edu/nsnam/ns/.

[2] *Wireless Lan Medium Access Control (MAC) And Physical Layer (PHY) Specifications, IEEE Std 802.11*, 1997.

[3] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements, IEEE Std 802.11e*, 2005.

[4] Jamal N. Al-Karaki and J. M. Chang. A Simple Distributed Access Scheme for Supporting QoS in IEEE 802.11 Wireless LANs. In *Proceedings of IEEE Wireless Communications and Networking Conference, (WCNC 2004), Atlanta, Georgia*, March 21-25 2004.

[5] V. Bharghavan, S. Lu, and T. Nandagopal. Fair Queueing in Wireless Networks: Issues and Approaches. *IEEE Personal Communications Magazine*, February 1999.

[6] V. Bharghavan, S. Lu, and N. Thyagarajan. Wireless Fair Service Algorithm: Algorithm and Properties. Technical Report, UIUC. Available at http://timely.crhc.uiuc.edu.

[7] C. Casetti and C.-F. Chiasserini. Improving fairness and throughput for voice traffic in 802.11e EDCA. In *IEEE Personal, Indoor and Mobile Radio Communications (PIMRC 2005),Barcellona, Spain*, September 2004.

[8] Yuan Chen and Lemin Li. A Fair Packet Dropping Algorithm Considering Channel

Condition in Diff-Serv Wireless Networks. In *The Fourth International Conference on Computer and Information Technology (CIT'04)*, September 2004.

[9] S. Choi, J. Prado, S. Shankar, and S. Mangold. IEEE 802.11e contention-based channel access (EDCF) performance evaluation. In *Proceedings of IEEE International Conference on Communications, ICC*, May 2003.

[10] C. Hoene, I. Carreras, and A. Wolisz. Voice over ip: Improving the quality over wireless lan by adopting a booster mechanism - an experimental approach. In P. Mouchtaris, editor, *Proc. of SPIE 2001 - Voice Over IP (VoIP) Technology*, pages 157–168, Denver, Colorado, USA, August 2001.

[11] D.P. Hole and F. A. Tobagi. Capacity of an IEEE 802.11b wireless LAN supporting VoIP. In *IEEE International Conference on Communications*, June 2004.

[12] L. Kleinrock and F. Tobagi. Packet switching in radio channels part ii - the hidden node problem in carrier sense multiple access nodes and the busy tone solution. *IEEE Transactions on Communications*, COM-23(12):1417–1433, 1975.

[13] Cheng Kuan and Ng ZhengYong. Providing QoS in Congested IEEE 802.11 Hot Spots. In *First International Conference on Distributed Frameworks for Multimedia Applications (DFMA'05)*, February 2005.

[14] Seoung-Bum Lee, Gahng-Seop Ahn, Xiaowei Zhang, and Andrew T. Campbell. INSIGNIA: An IP-based quality of service framework for mobile ad hoc networks. *Journal of Parallel and Distributed Computing*, 60(4):374–406, 2000.

[15] Stefan Mangold, Sunghyun Choi, Peter May, Ole Klein, Guido Hiertz, and Lothar Stibor. IEEE 802.11e Wireless LAN for Quality of Service. In *in Proc. European Wireless (EW2002), vol. 1, pp. 32-39, Florence, Italy*, February 2002.

[16] Jochen Schiller. Chapter 7: Wireless LANs. *Mobile Communications*, 2003.

[17] A. Velayutham and J. M. Chang. An Enhanced Alternative to the IEEE 802.11e MAC Scheme.

[18] Yuan Xue, Kai Chen, and Klara Nahrstedt. Achieving proportional delay differentiation in wireless lan via cross-layer scheduling. *Journal of Wireless Communications and*

Mobile Computing, Special Issue on Emerging WLAN Technologies and Applications, 4(8):849–866, 2004.

[19] Yaling Yang and Robin Kravets. Distributed QoS Guarantees for Realtime Traffic in Ad Hoc Networks. In *IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, October 2004.

Appendix

Chapter 6

Appendix

6.1 Key Terms

Access Category A label for the common set of enhanced distributed channel access (EDCA) parameters that are used by a quality of service (QoS) station (QSTA) to contend for the channel in order to transmit medium access control (MAC) service data units (MSDUs) with certain priorities.

AIFSD[AC] An AC in 802.11e uses AIFSD[AC] instead of DIFS to determine the time for which it will defer. It is determined by $AIFSD[AC] = SIFS + AIFS[AC].SlotTime$, where AIFS[AC] is an integer greater than zero.

BSS Coverage of one Access Point is called a BSS. An access point acts as a master to control the stations within that BSS. Each BSS is identified by an SSID. The most basic BSS is two STAs in IBSS mode. In infrastructure mode, a basic BSS consists of at least one STA and one Access Point (AP).

CAP Controlled Access Phase is during when the HC (Hybrid Coordiantor in the HCCA mode allocates TXOP to itself and other QSTAs in order to provide this limited duration for contention-free transfer of QoS data.

CFP contention-free period (CFP): The time period during operation of a point coordina-

tion function (PCF) when the right to transmit is assigned to stations (STAs) solely by a point coordinator (PC), allowing frame exchanges to occur between members of the basic service set (BSS) without contention for the wireless medium.

Channel Utilization in percentage is the achieved throughput related to the physical data rate in bit/s of a digital communication channel. For example, if the throughput is 70 Mbit/s in a 100 Mbit/s Ethernet connection, the channel utilization is 70%.

CP The time period outside of the contention-free period (CFP) in a point coordinated basic service set (PCF BSS). In a BSS where there is no point coordinator (PC), this corresponds to the entire time of operation of the BSS

CSMACA Carrier Sense Multiple Access/Collision Avoidance, a network contention protocol that listens to a network in order to avoid collisions, unlike CSMA/CD that deals with network transmissions once collisions have been detected. CSMA/CA contributes to network traffic because, before any real data is transmitted, it has to broadcast a signal (RTS- CTS) onto the network in order to listen for collision scenarios and to tell other devices not to broadcast.

Downlink Opposite of uplink. Its directed from the base station to the STAs.

EDCAF A logical function in a quality of service (QoS) station (QSTA) that determines, using enhanced distributed channel access (EDCA), when a frame in the transmit queue with the associated access category (AC) is permitted to be transmitted via the wireless medium (WM). There is one EDCAF per AC.

IEEE 802.11a 802.11a supports bandwidth up to 54 Mbps and signals in a regulated frequency spectrum around 5 GHz. This higher frequency compared to 802.11b limits the range of 802.11a networks to 30 meters. For higher frequency 802.11a has lesser penetration ability and also for this reason it's non compatible with 802.11b

Infrastructure Wireless LAN Infrastructure mode wireless networking bridges (joins) a wireless network to a wired Ethernet network. Infrastructure mode wireless also supports central connection points for WLAN clients.A wireless access point (AP) is required for infrastructure mode wireless networking. To join the WLAN, the AP and all wireless clients must be configured to use the same SSID (*service set identifier; a 32*

character unique identifier attached to the header of a packet that acts as a password when a mobile device wants to connect to a WLAN). The AP is then cabled to the wired network to allow wireless clients access to, for example, Internet connections or printers.

Goodput is the amount of useful information that is delivered per second to the application layer protocol. Dropped packets, packet retransmissions and protocol overhead are not counted. Because of that, the goodput is lower than the throughput

Piggyback The overloading of a data frame with an acknowledgment of a previously received medium access control (MAC) protocol data unit (MPDU) and/or a poll to the station (STA) or any specific protocol information to which the frame is directed.

QAP An access point (AP) that supports the QoS facility specified in this amendment. The functions of a QAP are a superset of the functions of a non-QAP (nQAP), and thus a QAP is able to function as an nQAP to non-QoS stations (nQSTAs).

QSTA A station (STA) that implements the QoS facility. A QSTA acts as an non-QSTA (nQSTA) when associated in a non-QoS basic service set (nQBSS).

TXOP An interval of time when a particular quality of service (QoS) station (QSTA) has the right to initiate frame exchange sequences onto the wireless medium (WM). A TXOP is defined by a starting time and a maximum duration. The TXOP is either obtained by the QSTA by successfully contending for the channel or assigned by the hybrid coordinator (HC).

TC A label for medium access control (MAC) service data units (MSDUs) that have a distinct user priority (UP), as viewed by higher layer entities, relative to other MSDUs provided for delivery over the same link. A 802.11e QoS enabled MAC determine the UP for MSDUs belonging to a particular traffic category using the priority value provided with those MSDUs at the MAC service access point.

TID or Traffic Identifier. Any of the identifiers usable by higher layer entities to distinguish medium access control (MAC) service data units (MSDUs) to MAC entities that support quality of service (QoS) within the MAC data service. There are 16 possible TID values; 8 identify TCs, and the other 8 identify parameterized TSs. The TID is assigned to an MSDU in the layers above the MAC.

Throughput is the amount of digital data per time unit that is delivered to a certain terminal in a network, from a network node, or from one node to another, for example via a communication link. The throughput is usually measured in bit per second (bit/s or bps).

User Priority A value associated with an medium access control (MAC) service data unit (MSDU) that indicates how the MSDU is to be handled. The UP is assigned to an MSDU in the layers above

Useful channel utilization is the percentage of througput that is used to deliver per second useful information. Dropped packets, retransmissions and protocol overheads are not counted while calculating useful channel utilization.

Uplink In Wireless terminology uplink refers to any data transmission happening from the STAs to the base station or AP.

6.2 NS

6.2.1 802.11a changes

Table 6.1: 802.11a PHY configuration

SlotTime	90μs
CCATime	3μs
RxTxTurnaroundTime	2μs
SIFSTime	16 μs
PreambleLength	96bits = 16μs
PLCPHeaderLength	40bits
PLCPDataRate	6 * 10^6Mbps

To simulate 802.11e mac module on the IEEE 802.11a PHY we have modified the mac-802.11e header file of NS 802.11e module. The configurations for the various parameters are shown in the table 6.1.

6.2.2 ARC related changes in NS

To implement ARC we incorporated some changes in the definition of NS2.28's wirless module. The headers were appended with new fields so that it's possible to provide the feedback and the ordering to and from AP respectively.Following is a description of these modificiation.

Data Structures

- MAC/ACK Header The MAC header of the 802.11 e data and acknowledgement will contain the piggybacked congestion table information which includes the FTP and VoIP queue length and also the next VoIP packets queuing delay. The changes have been marked by the comments in both the MAC data and acknowledgement header" //ARC CHANGES//"

```
In 802.11e module mac-802_11e.h file
struct hdr_mac802_11e {
struct frame_control dh_fc;
u_int16_t dh_duration;
u_char dh_da[ETHER_ADDR_LEN];
u_char dh_sa[ETHER_ADDR_LEN];
u_char dh_bssid[ETHER_ADDR_LEN];
u_int16_t dh_scontrol;
u_char dh_adddr4[ETHER_ADDR_LEN];
struct QoS_control dh_qos;
u_char dh_body[0]; // XXX Non-ANSI
//ARC CHANGES//
u_int32_t node_id; // node id of the STA
double cong; // value of the next VoiP's delay
int q_len; //  queue length of the VoiP queue
int fq_len; // FTP queue length
//***************//
}
```

  The ACK packet contains both the selected node, when AP is sending MAC acks

or it will contain the congestion information when the STAs are sending ACKs to AP.

```
In mac-802_11.h file
struct ack_frame {
struct frame_control af_fc;
u_int16_t af_duration;
u_char af_ra[ETHER_ADDR_LEN];
u_char af_fcs[ETHER_FCS_LEN];
//ARC CHANGES//
u_int32_t af_node_id;
double cong;
int vq_len;
int fq_len;
/***********************/
};
```

- Congestion Table The congestion table is a hash table or map type of data structure. Its class definition is shown below.

```
class H {
public:
H();
m cong_table;  m is a STL map class with the congestion information
as value and STA id as key
double highest_congestion;
int max_node;
void push_hash(u_int32_t node, cong_entry cong); //push the congestion
information of a node in the map
int pop_hash();//Determine which STA to assign the channel next
};
```