Virtual PCF: Improving VoIP over WLAN performance with legacy clients

by

Usman Ismail

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2009

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Voice over IP (VoIP) is one of the fastest growing applications on the Internet. Concurrently, 802.11 Wireless LANs (WLANs) have become ubiquitous in residential, enterprise, campus and public networks. Currently the majority of traffic on WLANs is data traffic but as more people use wireless networks as their primary access medium, a greater portion of traffic will be real-time traffic such as VoIP traffic. Unfortunately 802.11 networks are designed to handle delay-insensitive, bursty traffic and perform poorly for VoIP streams. Experimental and analytical results have shown that a single 802.11b access point operating at the maximum 11 Mbps rate can support only 5 to 10 VoIP connections simultaneously. Intuitively, an 11 Mbps link should support approximately 85 bi-directional 64Kbps (G.711) streams. The reason for this under-utilization lies primarily in the Distributed Coordination Function (DCF) used by 802.11 MAC layer. The problem can be addressed by using the optional Point Coordination Function (PCF). However PCF is not widely implemented in commodity hardware nor likely to be. There is a similar problem with the proposed 802.11e standard for quality of service. To solve these problems we propose Virtual PCF, a legacy-client compatible solution to increase the number of simultaneous VoIP calls. We implement Virtual PCF, a scheme which employs a variety of techniques to improve both uplink and downlink VoIP QoS. This alleviates delays and packet loss due to DCF contention and doubles the number of supported VoIP sessions.

Acknowledgements

I would like to thank Professor Keshav for his guidance and support during the preperation of this thesis and the research presented in it. I would also like to acknowledge the work of Nabeel Ahmed for his tireless efforts in setting up the testbed on which all of my experimentation took place.

Dedication

This is dedicated to my parents for their constant support and encouragement during all my studies.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Voice over IP (VoIP) is one of the fastest growing applications on the Internet [13]. Concurrently, 802.11 Wireless LANs (WLANs) have becoming ubiquitous in residential, enterprise, campus and public networks. According to the WiFi Alliance [6] (a consortium of WiFi chipset manufacturers) 387 million WiFi enabled devices were sold in 2008 alone. Currently the majority of traffic on WLANs is data traffic but major enterprise wireless network solution providers such as Aruba Networks are pushing for all-wireless networks [45]. In addition, a major portion of the sald WiFi devices were small-form factor, hand-held devices such as smart-phones. Such devices have no wired connectivity. These factors are pushing more people to use wireless networks as their primary access medium. As a result of these trends we expect that all forms of traffic will run over WiFi networks. Unfortunately, 802.11 networks are designed to handle delay-insensitive, bursty traffic and perform poorly for VoIP streams.

There are many proposals to improve support for real-time traffic, including VoIP and streaming media traffic, over wireless Local Area Networks (WLANs). However most proposals require significant changes to the client, infrastructure hardware, and firmware. Historical evidence suggests that any proposal that requires changes to clients is unlikely to be widely adopted. This is because clients are generally manufactured by a myriad of vendors each of whom may or may not implement the standard. In addition clients are normally beyond administrative control of the organization deploying the new technology. This makes it difficult to ensure compliance with standards or ensure service to all or even a majority of clients. Furthermore, even infrastructure changes that do not violate the 802.11 standard often require changes to hardware. Much of the prior literature concentrates on VoIP clients on an isolated channel [31, 50]. In real-world deployments VoIP streams must coexist with non-real-time data traffic. We therefore present a legacy-client compatible approach entitled Virtual PCF that makes changes only to the infrastructure software and improves VoIP capacity in isolated channels as well as in the presence of data traffic. We have implemented our approach on a real testbed and show up to a 100% improvement in VoIP call capacity.

1.1 Problem Statement

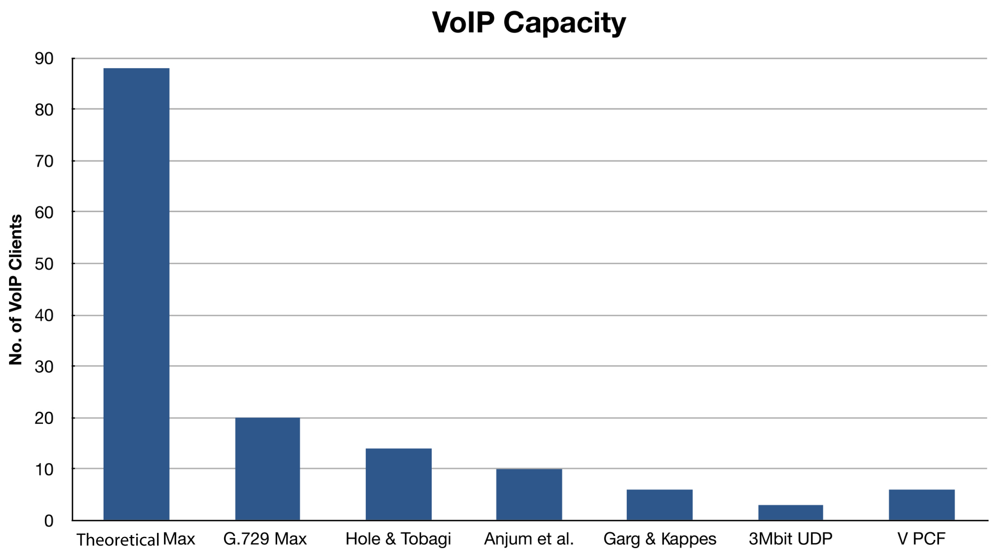


Figure 1.1: VoIP capacity of 802.11b

The worst-case bandwidth requirement of a bi-directional VoIP stream is 128 Kbps [25], i.e., 64Kbps upstream and 64kbps downstream. Modern audio codecs can compress voice streams to as little as 11 Kbps (GSM [17]). Therefore a single 802.11 Basic Service Set (BSS) using a data rate of 11 Mbps should support at least 88 simultaneous VoIP calls (11*1024*1024 / 128 * 1024). However we must realize that each data packet also has associated metadata such as headers of the various layers. VoIP packets generally have little data and the air-time used in transmitting metadata actually exceeds the air-time used to transmit data. Therefore the number of clients actually supported is much smaller (see Figure 1.1). If we use a protocol such as G.729 with a 20 ms packetization interval we can theoretically support up to 20 simultaneous VoIP calls if no air-time is wasted (see Section 4.3 for details). Hole & Tobaagi [22] conduct an analytical study that shows that if we take time wasted in contention into account then only 14 clients can be supported. Anjum et al. [9] conduct an experimental study and find that only about 10 clients can be supported in practice. Garg and Kappes [20] use the G.711a codec with a 10 ms packetization interval for their experiments and show that only about 6 clients can be supported. We ran our own experiments and found that with 3 Mbit CBR data cross traffic running on the same channel that the number of supported clients drops to 3.

The question naturally arises why are so few VoIP streams supported over 802.11 networks. The answer is only partially explained by the air-time used by metadata. A major cause of the problem is the DCF contention protocol; uncoordinated channel access protocols are known to have low air-time utilization as the channel load grows [48]. This is exacerbated by the fact that VoIP clients contend for the channel frequently (in the order of 10s of milliseconds). The problem is even worse for downlink traffic at the Access Point (AP) because for every single VoIP client there is a downlink VoIP stream originating at the AP. This means that an AP serving n clients requires "n" times the channel. However DCF is designed to provide fair channel access and therefore we see downlink VoIP traffic suffer more than uplink.

1.2 Objectives

Our objective in this thesis is to present a comprehensive solution for improving VoIP quality in both the downlink and uplink direction. Our implementation must not only improve VoIP quality in a channel reserved exclusively for VoIP traffic but also when VoIP and data traffic coexist on the same channel. We wish to make our implementation compatible with currently deployed technology. Therefore our approach must achieve these goals with no client-side modifications. Even infrastructure-side changes must be limited to software changes that can be implemented on commodity hardware.

1.3 Contributions

The contributions of work are as follows:

- We present a comprehensive approach to improving VoIP over WLAN (VoWLAN) performance, which improves both uplink and downlink performance. Our approach can increase the number of supported clients over a link by a hundred percent in some scenarios.

- Our approach is able to improve performance in both isolated VoIP deployments as well as deployments where there is simultaneous data traffic.

- Our approach provisions for talk spurts and silence suppression [54].

- Our approach can simultaneously support heterogeneous clients using various protocols.

- Our approach requires no client-side modifications and only software modifications to the infrastructure (Access Points).

- We evaluate our approach on a real testbed of 40 802.11 WiFi nodes deployed throughout two floors of our department building emulating a campus WiFi deployment.

1.4 Thesis Organization

The rest of this thesis is organized as follows; Chapter 2 presents a background of related topics including VoIP and the 802.11 protocol. Chapter 3 discusses the state of the art in VoIP over WLANs (VoWLANs) and discusses the need for our work. We describe our proposal in detail in Chapter 4 and evaluate the work in Chapter 5. Finally we conclude the thesis and present limitations of our work as well as describe possible future work in Chapter 6

Chapter 2

Background

2.1 802.11 WiFi

802.11 (WiFi) [24] networks have become the de-facto standard for short-range wireless networking. In fact due to the low cost and ease of deployment, many small and large scale networks rely entirely on WiFi networks for last-mile connectivity. WiFi solution providers are already predicting completely wireless work places [45]. In such an environment many applications, such as VoIP and audio/video streaming, that traditionally have used only wired networks now have to be supported over 802.11 wireless networks. This section describes the operation of the 802.11 protocol in detail and will provide a basis for the discussion in subsequent sections.

WiFi networks can operate in two modes; ad-hoc mode and infrastructure mode. In ad-hoc mode each client can connect to one or more other clients without central control. This mode of operation is rarely used, most networks are implemented in the infrastructure mode. In this scenario some nodes, Access Points (APs), provide service for all other clients. All clients which are serviced by an AP must first setup a connection with the AP. This process is called *Association*, the AP and all associated clients form a small network known as the Basic Service Set (BSS). A WiFi deployment consisting of multiple BSSs is known as an Extended Service Set (ESS). Normally, neighboring BSSs operate on different frequencies or channels to prevent interference. In order to allow many clients to share the same channel (or medium) the WiFi standard defines two protocols, the distributed DCF protocol and centralized PCF protocol. Although PCF is designed to better support real-time traffic such as VoIP, it is not generally implemented in commodity hardware. Hence any successful approach must operate over DCF. The following sections provide a detailed description of the operation of both DCF and PCF.

2.1.1 DCF Contention

The default medium access protocol used by 802.11 networks is called the Distributed Coordination Function (DCF) and uses CSMA/CA based contention. Any

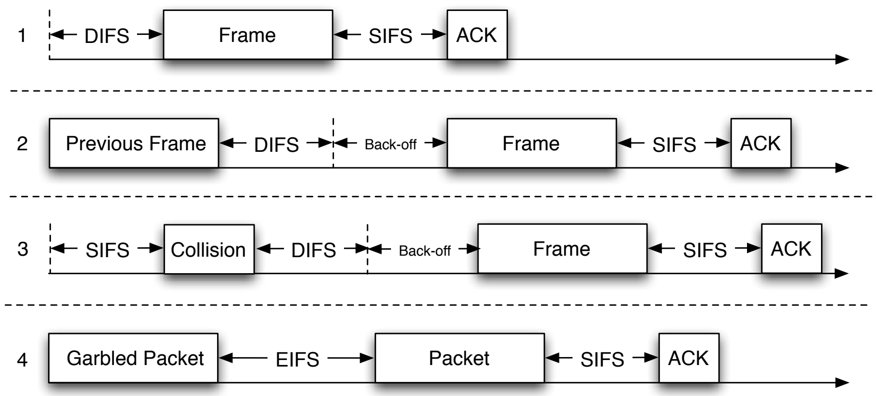


Figure 2.1: DCF Contention Examples

station with data to send first checks to see if the medium is idle the, station before transmitting. If the medium is idle then it waits for a short time known as DCF Inter-frame Space (DIFS) before transmitting a frame (see Figure 2.1, case 1). If the medium is busy or became busy during the inter-frame space then the station initiates the back-off procedure (see Figure 2.1, case 2).

The length of the inter-frame space is determined by the type of frame that the station wants to send. Waiting a short time will increase the probability of gaining access to the channel therefore higher priority frames are allocated shorter inter-frame spaces. There are four possible types of frame spaces, Short Inter-frame Space (SIFS), PCF Inter-frame Space (PIFS), DCF Inter-frame Space (DIFS) and the Extended Inter-frame Space (EIFS) in increasing order. The SIFS is used by stations which receive a frame and want to send an acknowledgement (ACK). An ACK must follow every frame transmission[1] therefore it receives highest priority. For standard data frames a station must wait the DIFS period. If the last frame on the air contained errors then stations must wait the EIFS period before transmitting (see Figure 2.1, case 4).

Once the station has waited the correct inter-frame spacing interval it must choose a number of slots to back-off. The size of a slot is dependent on the physical layer but all stations must use consistent slot sizes to ensure fairness. The number of slots to back-off for is a random number selected between zero and the Contention Window (CW) size. Each possible value has equal probability of being selected. All stations that wish to transmit independently select a back-off value and the station with the smaller value will transmit. If the station is not able to transmit, which happens if two or more stations select the same back-off value, then the CW on all colliding stations is increased and another random back-off value is selected at those nodes (see Figure 2.1, case 3). This process is repeated until any one station is able to transmit a frame. Each collision results in the contention win-

[1]Except broadcast frames

dow of colliding nodes being increased exponentially until it reaches the maximum contention window size (CW_{max}).

802.11 uses two types of carrier sensing *Physical Carrier Sensing* and *Virtual Carrier Sensing*. Physical Carrier sensing is done based on information provided by the underlying Physical Layer. The exact mechanisms are based on the modulation scheme and the medium used for transmission and are transparent to the MAC layer. Virtual Carrier Sense is based on the Network Allocation Vector (NAV) and is described in detail in Section 2.1.2.

2.1.2 Virtual Carrier Sensing

All 802.11 stations have a counter called the Network Allocation Vector (NAV). In addition most frames have a *Duration* field which specifies how long the transmission is expected to last. Stations receiving transmissions read the duration field as the frame is being received and update the NAV value equal to the duration field. Note this procedure is followed even for frames not addressed to the receiving station. The stations also decrement the NAV counter at an identical constant rate. When the NAV value reaches zero the medium is assumed to be idle. The fact that all stations use the same constant rate to decrement the NAV value ensures that they have a consistent view of the network.

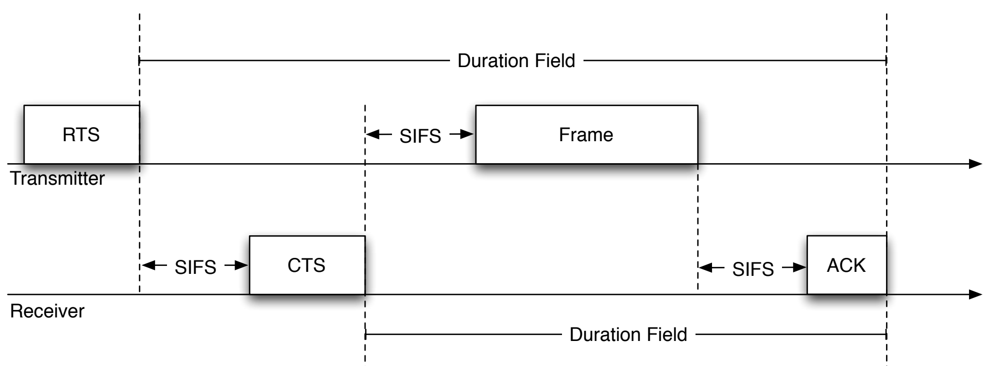


Figure 2.2: RTS/CTS Based Channel Reservation

Most frames use the duration field to reserve the medium for only as long as they actually expect to keep the medium busy. However, some scenarios, e.g., hidden terminal interference, require a station to reserve the medium for a longer period of time. To do this DCF implements the RTS/CTS mechanism. The station wishing to transmit a data frame sends a Request to Send (RTS) frame directed to the intended receiver of the data frame. As shown in figure 2.2, the duration field of the RTS is set such that a Clear to Send (CTS) frame, data frame and ACK can be transmitted all within the reservation interval. The reservation must also accommodate all required inter-frame spaces. All stations, except the receiver, set their NAV counter to the value specified in the RTS duration field. The receiver waits the SIFS interval and sends a CTS frame with the duration set such that it equals the initial duration in the RTS minus the time elapsed since its transmission. All nodes that receive the CTS again update their NAV counters but note that the value of NAV counter and the CTS frames duration field should be identical. The CTS may seem redundant but it is essential for preventing hidden terminal interference (see Appendix A). After receiving the CTS the sender again waits a SIFS interval and then transmits the original data frame. Following the data frame the receiver waits a SIFS interval before sending the ACK. At this point the NAV counter of all stations becomes zero and they wait a DIFS interval before initiating back-off procedure and subsequent transmissions.

2.1.3 Point Coordination Function

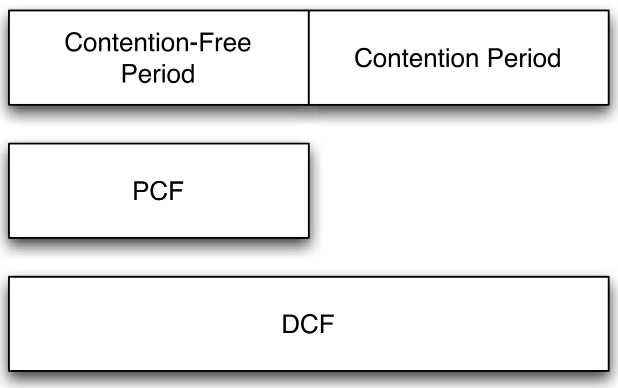


Figure 2.3: Point and Distributed Coordination Functions

The 802.11 specification includes another mechanism for managing medium access, the Point Coordination Function (PCF). In this mode the total air-time is divided into the Contention-Free Period (CFP) and the Contention Period (see Figure 2.3). During the contention period all stations access the medium using DCF contention as described in Section 2.1.1. One node (the Access Point) is selected as the point coordinator and it ensures that medium access is provided to stations without contention during the contention free period. The point coordinator seizes control of the medium and allocates air-time to other stations. PCF is an optional part of the 802.11 standard not widely implemented. However note that PCF is implemented on top of DCF and thus can coexist with non-PCF compliant stations.

The Point Coordinator (PC) signals the beginning of the CFP by transmitting a beacon with a duration field set to the CFPMaxDuration, which is the maximum time that the CFP may last (see Figure 2.4). This causes stations to update their

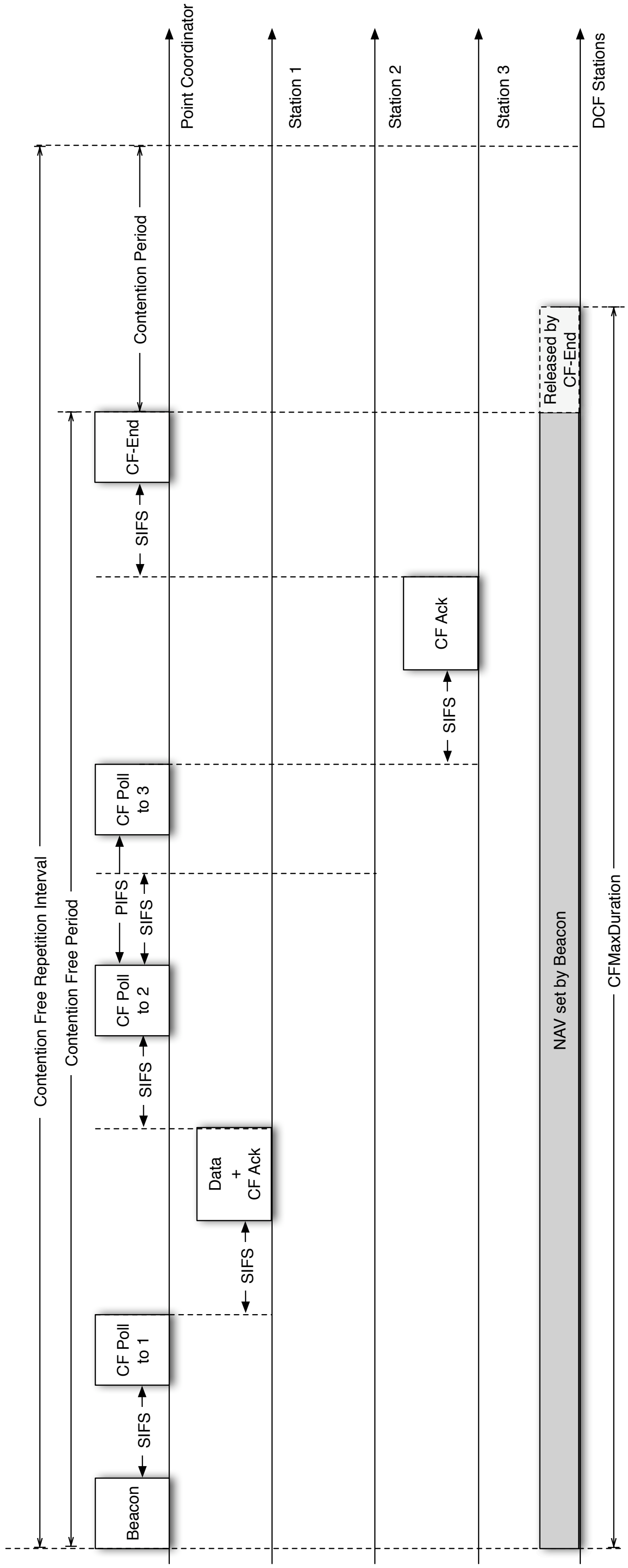


Figure 2.4: PCF Operation

NAV counter and locks out all stations wanting to access the channel via DCF. The PC then sends a Poll Request (CF-Poll) to the first station on its Polling List (Stations can be added to the polling list by setting a flag when they first associate with the AP). Each CF-Poll allows the receiver to send one frame only, and this frame is transmitted after a SIFS interval (For example station 1 in Figure 2.4). By waiting only SIFS instead of DIFS, PCF ensures any DCF station that did not receive the initial beacon is still not able to acquire access to the medium. If the polled station does not respond for a duration equal to the PIFS then the coordinator polls the next station on the polling list (See station 2 in Figure 2.4). As PIFS is shorter then DIFS no DCF station is able get medium access even if the polled station remains silent. After correctly receiving a CF-Poll a station must reply with a CF-ACK, even if it has no data to send for example station 3 in Figure 2.4. Note if the station has data to send it can combine the CF-ACK with the data frame. The access point can also combine a CF-Poll frame with a data frame if it has data to send to the station being polled. In fact many different types of combined packets are used to minimize the wasted air-time (see Appendix B for exhaustive list). When all stations have been polled, the PC sends a CF-End frame with a small duration field so that the larger NAV value set by the beacon is overwritten. The total time taken from the start of one beacon to the start of the next one is called the *Contention Free Repetition Interval*. Every PCF station can expect to be polled at intervals approximately equal to the contention free repetition interval.

To run VoIP clients over PCF we would add the VoIP clients to the polling list and make sure that the Contention Free Repetition Interval is equal to the packetization interval of the audio codec used. Note each VoIP client generates a frame after one packetization interval and we must poll client to make sure the frame is transferred promptly. There are however major problems with using PCF to support VoIP traffic. First, PCF is an optional part of the 802.11 standard and not commonly implemented on commodity systems. Second, if a pre-associated client starts a VoIP session it must re-associate with the access point in order to be added to the polling list. In fact, if the VoIP client uses silence suppression for talk-spurts (see Section 2.2 for details) then each VoIP client will be silent for up to 60 percent of the time. Hence the coordinator either wastes air-time sending CF-Polls to stations which do not require air-time or disassociates the station in order to remove it from the polling list. Finally, a single BSS cannot have clients using different packetization intervals. In such a scenario either some clients will be polled unnecessarily, wasting airtime, or not often enough causing degradation of QoS.

2.1.4 802.11 a/b/g/n

The 802.11 specification contains several different physical layers including 802.11 a, b, g and n. The most heavily deployed is 802.11b, which uses High-Rate Direct-Sequence Spread-Spectrum (HR/DSSS) and supports data rates of 5.5 Mbps and 11

Mbps. A later version, 802.11a uses Orthogonal Frequency Division Multiplexing (OFDM) and supports rates of up to 54 Mbps. In addition it reduces the size of the physical layer (PLCP) header from 196μs to 20μs and also reduces DIFS to 34μs from 50μs. However 802.11a is not interoperable with 802.11b clients and hence an intermediate protocol (802.11g) is specified, which allows for the higher rates and shorter inter-frame spacing of ‘a’ but is still interoperable with ‘b’. Currently another physical layer (802.11n) is being standardized, which allows the use of data rates as high as 300 Mbps and also the use of Spatial Division Multiplexing.

Is there any need for improvement of VoIP over WLANs when we can just use the higher data rates of the high speed physical layers such as ‘a’ and ‘n’ to run many simultaneous clients? There are several problems that remain with high speed physical layers that motivate our work. First, the higher data rates are more susceptible to packet loss making them unsuitable, in some cases, for VoIP traffic. Second, the high speed layers are not supported by all devices, ‘n’ is poorly supported and ‘a/g’ are not supported on many small-form-factor devices. Third, as more bandwidth becomes available more bandwidth intensive applications are also developed. We will always face the problem of data traffic saturating the medium and not allowing VoIP traffic through. Lastly, the more important consideration is the large installed base of 802.11b/g devices that needs to be supported. In addition if there is one node in a BSS that requires the AP to use a slow data rate then that slow rate is used for all nodes. This problem, known as the rate anomaly, occurs because APs do not use per client rate control. Furthermore if a node uses a lower data rate on the uplink then it receives the largest share of the air-time because transmitting a frame at a lower data rate takes longer.

2.2 VoIP Traffic Characteristics

We have mentioned earlier that VoIP traffic is different from data traffic and this makes 802.11 unsuitable to host VoIP clients. In this section we enumerate the characteristics of VoIP traffic and explain why 802.11 is affected by them.

1. *Low bandwidth:* Voice data does not require much bandwidth, in the worst case only 128 Kbps (64 Kbps in each direction). This is why theoretically an 11 Mbps 802.11b connection should support a large number of clients. However, in practice this is not usually realized for a number of reasons. One of the important reasons is that 802.11 (DCF) contention is designed to give all competing stations equal probability of accessing the medium. Unfortunately TCP data traffic always tries to saturate the medium and competes for all available slots. VoIP traffic only competes for a few transmissions slots, and even on these it has an equal probability of gaining access as any of the other stations. This means that VoIP traffic may not even get the meager bandwidth it requires.

2. *Delay Sensitivity:* The most important distinction between VoIP and data traffic is that data traffic is relatively delay insensitive. TCP / UDP traffic can normally tolerate delays of several seconds or more without causing a detrimental effect on the user experience. In contrast VoIP traffic can only tolerate end-to-end delays of 200 to 400 ms. Furthermore VoIP is sensitive to variation in delays (Delay Jitter) between subsequent packets, even variation of the order of a few milliseconds. In order to address this issue, most codecs trade-off additional end-to-end delay for a reduction in jitter using a buffer (Jitter Buffer). This buffer stores incoming packets and outputs them to the decoder at a constant rate. However packets delayed beyond the size of the jitter buffer are considered lost. A larger buffer allows packets to be delayed by a longer period and still be sent to the decoder without jitter but this also causes a larger end-to-end delay. Different codecs use different jitter buffer sizes ranging from 0 ms to 30 ms and larger. In our evaluation we assume the use of ITU G.729 codec [30] which recommends a 5 ms jitter buffer.

   802.11 (DCF) uses CSMA/CA with exponential back-off strategy. This strategy means VoIP clients may back-off and miss their transmission deadline. In addition the back-off time is highly variable based on channel conditions hence consecutive packets are likely to be delayed for different durations causing excessive delay jitter.

3. *Loss Sensitivity* In addition to being sensitive to delay VoIP traffic is also sensitive to frame loss rates. MAC layer retransmissions only partly solve the problem as the delay of retransmissions may be prohibitive. End-to-end retransmissions are not possible as the requisite delay is well beyond acceptable thresholds. Although some packet loss is acceptable, anecdotally around 2% to 5% depending on the encoding scheme used, significant losses can make voice unintelligible. This is one of the reasons VoIP clients should avoid the higher MAC data rates. Furthermore, standard auto-rate selection may not be suitable for VoIP traffic. Auto-rate always tries to maximize data rate used, which can lead to periods of high losses.

4. *Small Packet Size:* Another important characteristic of VoIP traffic is that the data contained in any single packet tends to be small, up to a few tens of bytes. However, each packet has significant metadata including, PLCP, MAC, IP, UDP and RTP layer headers. This leads to a situation where the data accounts for a minute percentage of the total air-time of each packet. For example, a G.729 frame transmitted at 11 Mbps takes 273μs (if we include the ACK frame) Of this time, the eight bytes of voice data take up only six microseconds resulting in a 680% overhead [48]. There are many proposals to address this problem (see Sections 3) however all require modifications to the 802.11 protocol and this makes them unlikely to be deployed in commodity systems.

5. *Periodicity:* The standard sample rate to capture voice data is 8KHz or one sample every 125μs. However, it is impossible to send packets over 802.11

networks at this rate. This is why we aggregate data for several sampling intervals into a single packet. Common encoding schemes typically aggregate data for several 10s of milliseconds (See section 2.3 for details). The time interval for which samples are aggregated is known as the *Packetization Interval*.

6. *Sporadic Nature:* Finally, an important feature of interactive voice communication is that one of the two channels (uplink and downlink) is usually idle and each channel is used in a short burst known as a *Talk Spurt*. This can be explained intuitively because during a conversation one of the two parties is talking and the other is listening. There are however short periods in which both parties are talking and also short periods when both parties are silent. Prior literature [10] suggests that during a typical phone conversation each side is silent for as much as 60% of the duration of the call. Typically a user talks for an average of 1s (a talk spurt) and is then silent for an average of 1.4s. Furthermore the length of the talk-spurts follows an exponential distribution. These characteristics make it difficult to implement scheduling schemes such as PCF as stations' medium access requirement changes dynamically. We look at some prior work that addresses this issue in Section 3.8 and discuss how we address the problem in Section 4.6.

2.3 Audio Codecs

A standard audio stream is created by digitizing an analog audio signal, this is done by sampling the signal at small intervals (Sampling Interval) and capturing information about that sample. The sampling interval, sample size and other details of how the *Encoding* of the analog signal into digital form will take place is specified in a coding standard and implemented in a hardware device called a Coder-Decoder or a *Codec*. An unencoded voice call in a telephone network occupies a 64Kbps stream with a sampling interval of 125μs and 8 bit samples. This codec is defined by the ITU G.711 [25] standard and used in telephone networks worldwide. G.711 is unsuitable for VoIP as it generates many small packets at small intervals. In addition there is no buffer to mask delays in processing and transmissions. More advanced standards use a packetization interval to collect several G.711 samples together before transmitting. In addition they use compression techniques to reduce the number of bits required to store audio samples. However the delay caused by waiting for samples to accumulate and the compression technique used, lowers audio quality. Table 2.1 lists several audio codecs, their bit-rate, sampling interval and the theoretical maximum Mean Opinion Score (MOS)[2]. A score of 5 means perfect fidelity and a score of 1 means random noise. This measurement scheme has been standardized into the e-model [29] so that different codecs can be compared fairly. Appendix C describes in detail how to calculate MOS using the e-model.

[2]MOS [28] is a measure of audio quality based in empirical studies of perceived audio quality using test subjects

Coding Scheme	Bit-rate	Sampling Interval	Max MOS
G.711	64Kbps	0.125ms	4.45
G.723.1	6.3Kbps	30ms	3.9
G.723.1	5.3Kbps	30ms	3.62
G.729	8Kbps	10ms	3.9
G.726	16Kbps	10ms	Not Available
G.726	24Kbps	10ms	Not Available
G.726	32Kbps	10ms	4.3
G.726	40Kbps	10ms	4.45

Table 2.1: Various Codecs, associated parameters and maximum MOS

2.4 802.11 VoIP Capacity

Prior to beginning work on increasing VoIP over WLAN capacity we must establish a need for improvement. There is a large body of work studying the performance of WLANs but the work relating to VoIP traffic over WLANs is fairly limited. The work can be divided into two broad categories; Analysis/Model based and Experimental studies. We conduct a survey of literature in both areas and present the findings of various researchers in each category.

2.4.1 Analysis Based Studies

Early analytical work, such as [15], [47] [51], modeling VoIP performance over 802.11 focused on the Point Coordination Function (PCF). Veeraraghavan et al. [47] find that up to 26 simultaneous calls can be made using PCF if a 90 ms packetization interval is used. With a 60 ms interval this drops to 17 calls. Kawata et al. [33] calculate analytically that PCF, with 20 ms packetized G.711 traffic, should support approximately 24 calls but their simulations show that only 18 calls can be supported. With silence suppression the number of supported calls jump to 34 and 30, as determined by calculation and simulation respectively. PCF based schemes give us hope that coordination will help increase the VoIP throughput but few commercial vendors support PCF and hence VoIP solutions based on the protocol are impractical.

Garg et al. [19] present analytical results for the total number of calls supported on several 802.11b rates using the DCF protocol. They calculate the number of supported calls for the G.711, G.729 and G.723 codecs for a range of packetization intervals. For example, at 11 Mbps G.729 using 30 ms packetization will support 21 calls. Using the 54 Mbps rate of 802.11a at the same configuration will allow 95 simultaneous calls. Some of these results are verified by simulation in [22]. There are a number of analytical studies presenting similar results, which we elide in this paper. These results give us an interesting point of reference but for more realistic results we now turn to experimental evaluations of VoIP capacity.

2.4.2 Experiment Based Studies

In a followup to their earlier analytical paper Garg et al. [20] find that only six G.711 VoIP calls can be simultaneously supported by an access point using a packetization interval of 10 ms. This concurs with their earlier analytical prediction. Jeong et al. [31] conduct a similar experiment and find that a maximum of five concurrent clients are supported. Anjum et al. [9] conduct such an experiment using a 20 ms packetization interval and find that ten calls can be sustained whereas Shin et al. [41] use the ORBIT testbed [39] to conduct experiments in which they show that up to fifteen simultaneous G.711 VoIP calls can be sustained at a packetization interval of 20 ms. There is a dearth of similar experimental work relating to VoIP capacity using PCF and therefore we are limited to analytical and simulation based work for PCF.

An interesting dimension to the discussion on VoIP capacity is the effect of co-locating bursty and real-time VoIP traffic. Until now all results have focused on isolated VoIP traffic. Garg et al. [19] find that co-locating the VoIP and data traffic is detrimental to both. Each 128 Kbps G.711 voice stream effectively reduces bandwidth available for bursty traffic (UDP) by 900 Kbps. Anjum et al. [9] experiment with constant bit-rate non-real-time background traffic and see the number of supported VoIP connections falls significantly with even moderate background traffic. Yu et al. [52] show that in the presence of just 5 TCP flows the drop rate can be over ten percent and the delay approaches 100 ms, assuming an AP frame queue size of 50 packets. Larger queue sizes will reduce the drop rate but can make the delay excessive. Kawata et al. [33] simulate mixed traffic over both PCF and DCF modes. Under PCF they assume 30 VoIP clients and up to 3 clients generating ftp traffic. They find that the combined throughput of VoIP and FTP traffic remains below 2 Mb/s on an 11 Mb/s link and the end-to-end delay can be as high as 430 ms. Using DCF about 2.5 Mb/s combined throughput is achieved but the end-to-end delay can be as high as 520 ms.

To summarize using DCF over 802.11b we can run a maximum of 5 to 10 VoIP calls in isolation. Introducing non-real-time traffic on the same channel degrades VoIP QoS significantly such that maintaining even a single VoIP session can be difficult. Using PCF can increase the number of supported clients but as mentioned earlier PCF is not widely implemented. We therefore require a means to increase VoIP capacity for both isolated VoIP traffic and VoIP traffic collocated with data traffic that is compatible with current hardware.

Chapter 3

Related Work

In this section we present the current proposals that seek to improve VoIP over WLAN capacity and QoS. We start with several proposals on improving VoIP capacity over DCF include differentiated contention, priority queueing and ACK aggregation. Although all these approaches improve the state-of-the-art by various degrees they all require client modification. This makes them difficult to deploy in real-world scenarios. We also look at the proposed 802.11e standard, which addresses many of the problems faced by real-time traffic including those faced by VoIP traffic. Furthermore, we look at a proposal called Soft Speak, which presents the most comprehensive solution to the problem, except maybe 802.11e. However even this proposal requires client modification. Finally, we present an approach to support talk-spurts over PCF. We will be using the idea of this approach in our design.

3.1 Differentiated Contention

Distributed Coordination Function (DCF) based contention is designed for bursty delay tolerant traffic. VoIP streams are periodic and delay sensitive and therefore suffer when competing with data traffic for channel access. This is especially relevant for the Access Point (AP). Note, for every VoIP client sending a single stream of packets the AP has to send a corresponding stream of downlink packets. Therefore the AP needs n times the channel access as any client (assuming there are n active clients). However as DCF is designed to allocate channel access fairly the AP may not get desired channel access. Therefore we expect the downlink quality to fall before uplink quality as the number of simultaneous VoIP clients increases. This is confirmed by previous work [48] and our own experimentation (see Section 5.4). To tackle this issue researchers propose differentiated contention for prioritized traffic. Anjum et al. [9] propose a zero-back-off policy for the AP to prioritize its heavier traffic load. This approach is compatible with legacy-clients but can only improve down-link VoIP QoS. We use a similar policy for our approach to improve downlink quality.

3.2 Priority Queueing

In addition to contending with other clients, VoIP packets also have to share air-time with data packets originating from the local node. If a VoIP client (or the Access Point) has a large number of data packets to send at any point in time it will have to use output queues or buffers to store packets waiting to be transmited. Currently most 802.11 nodes use a single FIFO queue to buffer outgoing packets. If VoIP packets are queued behind data packets their transmission will be delayed because data packets tend to be more common and much larger. More importantly, the delay of each packet will become highly variable because of the number of data packets in the queue as well as contention delays of DCF. This has lead several researchers to propose separate queues for real-time traffic [9, 52]. For legacy compatibility reasons we cannot use multiple queues at client nodes. However [9] shows that even priority queuing only at the AP can add call capacity because downlink VoIP streams are more susceptible to queuing effects. We use this approach in our design to increase downlink VoIP capacity.

3.3 ACK Aggregation

Every 802.11 frame requires the receiver to send an acknowledgement (ACK) to confirm correct receipt of the packet. If this ACK is not sent then the packet is marked as lost and retransmitted. Although ACK packets are small, they are sent at the lowest rate and each packet requires the transmission of a PLCP and MAC header. In saturated conditions the air-time used to send the ACKs reduces the number of VoIP calls supported. Therefore [31] propose the use of aggregate ACKs, a single acknowledgement for a train of frames. Their experiments show that this could allow two additional VoIP calls to be added to a channel which was otherwise saturated (using 802.11b at 11 Mbps). However, as with many other proposals, ACK Aggregation requires client modification making it incompatible with the current 802.11 standard.

3.4 Frame Aggregation

As mentioned in Section 1.1, one of the major reasons for the limited VoIP capacity of 802.11 is the high per-packet overhead of protocol headers. In order to reduce the overhead researchers have proposed a technique called Frame Aggregation [31, 50, 53]. Frame aggregation refers to combining the payloads of multiple small VoIP frames in order to avoid the high per-packet overhear of DCF contention. However the added delay of waiting for consecutive packets to be generated will increase end-to-end delay hence reducing voice quality. An interesting variant of this approach is proposed by Wang et al. [50] where the downlink frames of distinct clients are aggregated and then transmitted as single multicast frame. This does not cause extra delay because if there are a number of VoIP clients then there maybe several VoIP packets in the send queue of the Access Point. The drawback of all frame aggregation approaches is that they need client modification and add additional delay. The approach proposed by [50] reduces the additional delay but it can only improve downlink traffic.

3.5 Header Compression

Header compression [31] increases the percentage of data bits in each packet which in turn increases throughput. However note that we are not able to compress PHY layer (PLCP) headers and MAC Layer headers easily as that would require changes to hardware and firmware. There is limited benefit to be derived from header compression as PHY layer headers and MAC layer are a large portion of the metadata associated with each packet.The PHY and MAC layer headers account 32% and 22% respectively (see Section 4.3 for a details). Furthermore header compression requires changes to clients.

3.6 802.11e

The 802.11e [23] standard provides several new tools to wireless network administrators to better provision real-time delay sensitive applications. Traffic is divided into four QoS categories, refereed to as Access Categories (ACs). Each category has its own prioritized queue on every station. Priority queuing is important for avoiding head of line blocking because the probability of a station getting access to the medium is dependent on the AC of the frame(s) at the head of the queue. In addition, each AC has an associated minimum contention window (CW_{min}). A larger minimum increases the probability of a larger back-off period being selected and therefore reduces the probability of gaining access to the channel. For this reason, low-priority ACs have a larger CW_{min} than the higher-priority ACs.

In addition to adding priority queues and tweaking the back-off parameters 802.11e introduces a new MAC layer, the Hybrid Coordination Function (HCF). Much like the traditional 802.11 MAC layer, it has both contention-based and contention-free channel access protocols. Contention-based channel access is provided by the Enhanced Distributed Channel Access (EDCA) protocol. In DCF contention, after a transmission all stations must wait at least a DIFS before attempting to access the channel. With EDCA every station must wait the DIFS interval and then wait an additional inter-frame space known as AIFS. The size of the AIFS is dependent on the AC of the frame being transmitted, high priority ACs use a smaller AIFS increasing their chance of gaining channel access. By ensuring that all frames wait at least the DIFS 802.11e remains compatible with legacy 802.11 stations in the same BSS. However all traffic originating at legacy stations will have highest channel access probability because it is not waiting the additional inter-frame space.

To provide contention-free channel access, 802.11e provides the Hybrid Coordination Function (HCF) Controlled Channel Access (HCCA) protocol. In the Contention-Free-Period (CFP) the Hybrid Controller (HC) allocates a Transmission Opportunity (TXOP) to each station in the polling list. During a TXOP the station can send traffic for a time period equal to *TxOPLimit*. However, the controller does not reserve the channel for the entire time period, instead the station uses a smaller inter-frame space ensuring it always wins channel access. This method is referred to as *Packet Bursting*. 802.11e gives stations the options of ACKing an entire burst of packets as opposed to individual ones. In fact highly delay sensitive applications can specify that ACKs should be omitted altogether. This allows us to reduce the control overhead of frames. However, packet bursting may be of limited use with VoIP streams because packets are generated at fixed intervals and several packets may not be available to transmit during a TXOP. In addition, unlike PCF, HCF allows the Hybrid Controller to poll stations during both the contention-based and the contention-free periods. This allows us to support clients with differing inter-packet intervals.

Experiments by Dangerfield et al. [16] show that even with as many as 18 stations sending saturated data traffic, a voice call can still maintain delays of under 10 ms. Nevertheless, despite its potential there are still some drawbacks of 802.11e which we discuss next.

Differentiated contention schemes such as EDCA are known to cause starvation of low-priority traffic under heavy load scenarios. Furthermore, research has shown that even high priority traffic may receive less aggregate throughput due to increased frequency of collisions caused by smaller contention window sizes [38]. This phenomenon is known as *Performance Inversion*. The Dangerfield study [16] does not see this because they only use one prioritized VoIP client and vary the number of low priority clients.

HCF also has some shortcomings, it lacks the means to dynamically adjust the polling list in response to talk-spurts and pauses. Most conversations are half-duplex and one of the two parties is silent at any given time therefore this is a significant drawback. Finally, both modes of 802.11e suffer from the fact that they are not well-supported by commodity hardware. In fact the HCCA protocol is an optional part of the standard and may not be implemented even when 802.11e-compatible hardware becomes available in the market.

3.7 Packet Scheduling

Another interesting set of proposals seeks to use TDMA-style scheduling for VoIP packets. The Soft Speak project [48] modifies client back-off behavior in order to schedule uplink VoIP traffic. For this purpose, all clients first synchronize their clocks using the AP's periodic beacons. The air-time is then divided into 1 ms slots where each client is assigned a transmission slot by the AP. All clients contend for access in all slots using the default 802.11 behavior. However in the slot assigned to a client it will wait only the SIFS interval plus 40μs (2 802.11 contention window slots). As this is shorter than the interval used by 802.11 by default, all other clients will be locked out. Using this approach, the authors completely eradicate contention for VoIP clients. However this approach is difficult to implement in practice because it requires modification of clients. Instead, we propose a scheduling scheme that allocates transmission slots without requiring client modification.

3.8 Supporting Talk Spurts

As described in Section 2.1.3 the 802.11 standard defines a polling-based channel access mechanism, PCF, for supporting real-time traffic such as VoIP. However, PCF is inefficient in real-world scenarios because it polls stations even if they have no VoIP traffic to send. This is problematic because on average, stations have no VoIP data to send for over 60% of the time. In order to address this issue Kawata et al. [33] propose dynamically updating the polling list. The AP monitors all uplink traffic and detects VoIP packets. Once a VoIP packet is detected the AP adds the station to the polling list and hence the next VoIP packet can be transmitted without contention. When a talk-spurt ends, the station does not utilize its transmission opportunity which implicitly signals the AP. The AP removes the station from the polling list but maintains other metadata associated with the station. When a new talk-spurt starts the station must again contend for the channel using DCF to transmit the first packet. However, as soon as the AP detects this packets it again adds the station to the polling list using the metadata. This approach allows PCF to support talk-spurts and requires no client-side changes. However other issues with VoIP over PCF such as heterogeneous clients and lack of compatible hardware (see Section 2.1.3 for details) still remain with this approach.

Chapter 4

Virtual PCF: System Design and Implementation

In this chapter we propose *Virtual PCF*, a scheme to increase VoIP call capacity on 802.11 links in the presence of data traffic, without requiring client modification. We use a two-pronged approach, using different mechanisms to improve uplink and downlink quality. To improve uplink quality we predict when a VoIP client will attempt to contend for the channel and how long it will require the channel. We then proactively reserve the channel using unsolicited CTS frames as described in Section 4.4. By reserving the channel we ensure all future channel access will be contention free. This reduces quality impairments due to delay variation and packet loss[1], which in turn increases the number of calls that can be placed on a given link while maintaining QoS. As Virtual PCF is implemented over DCF it retains compatibility with commodity clients and non-real-time traffic. Note that non-real-time clients are allowed to contend for air-time as usual except for slots that are explicitly reserved for real-time traffic.

To improve downlink VoIP quality we note that downlink VoIP streams suffer because an AP generates much more VoIP traffic then any single VoIP client. However DCF contention is designed to give equal channel service to all nodes. To improve downlink quality we need to prioritize the AP's channel access over other stations. We implement a zero back-off strategy for VoIP packets transmitted by the AP. If the AP has a VoIP packet to send and the medium is not idle it waits for the medium to become idle and then transmits after a DIFS. If any other station also uses a zero back-off (i.e., its randomly chosen back-off value was zero) there will be a collision. The AP and the client both back-off and try to retransmit however the AP again uses a zero length back-off, increasing the probability of gaining channel access significantly. Finally, we also support talk-spurts by dynamically updating the list of *active* VoIP clients in response to pauses and talk-spurts.

[1] Packets that are delayed beyond the play-out buffer interval are considered lost

4.1 Detecting VoIP Packets

The first step in our approach is to detect presence of VoIP frames. We need to detect both uplink and downlink VoIP frames at the access point. In order to identify VoIP frames we inspect all traffic flowing though the access point. Inspection of all packets at line speed may seem impractical, however note that the vast majority of frames can be filtered out after only a few byte comparisons. For example we can check the *Type* and *Sub-Type* fields of the *MAC header*. All VoIP packets will be of the type "Data" and Subtype "Data". Hence we can ignore all other packets. Also note that the MAC header is of fixed size and structure hence the two bits that represent type and four bits that represent subtypes will always be at a fixed location. In fact as the first two bits of the MAC header are constant [2] a single byte comparison is enough to identify possible VoIP frames at the MAC layer. The first byte must be equal to "00100000". We use a similar procedure to filter out packets using the headers of the higher layers.

Once we have determined that the packet is a UDP data packet, we use a protocol specific identifier to determine if a packet is actually a VoIP packet or not. This may involve deeper inspection then before as we may need to look for several characteristics in order to identify VoIP packets. For example [43], [2], [3] describes procedures to detect traffic from a Skype client (one of the popular and encrypted VoIP protocols). Implicit in this procedure is the fact that we also identify which VoIP protocol is being used and therefore the characteristics of the VoIP stream, i.e., the packetization interval, jitter buffer size and whether silence suppression is used. This information is vital for prioritizing uplink traffic (see Section 4.2). We implement our inspection code in the AP 802.11 driver. This has two important implications. First, it requires no modification to the client and no hardware changes to the AP. Second, we are using a cross-layer optimization and hence require access to the entire raw frame at the driver (at the MAC layer) which may not be available on all architectures. However we have access to the raw frame using the SkBuff on all Linux/Unix based operating systems, which we expect the majority of APs to use. In commercial systems such access may be available through the API directly.

4.2 Predicting Channel Access Time

For an AP to reserve the channel for a client it needs to predict all future times when the client will require access to the channel. We predict the next channel access using a scheme proposed Kawata et al. [33]. Clients initially use DCF to contend for channel access. The uplink traffic of all clients is inspected by the AP to identify VoIP packets. From these packets the AP can identify the data size (S) and

[2]The first two bits define the 802.11 protocol version and so far only the protocol number zero has been designated so all 802.11 frames use a protocol version number of zero

periodicity(ρ) of packets that will be sent during the session. Furthermore, using the first packet's arrival time(t_0) and adjusting for transmission and contention delays, the AP can approximate when the client started contending for the channel. This information allows us to estimate t_1, when the client will next require channel access, using Equation 4.2. In the equation t_{trans} is the transmission delay of the packet and t_{back} is the time taken to contend for access to the channel (see Figure 4.1 case 1). We have no information about the actual time taken to acquire the channel but we know it ranges from zero to the contention window size. As a conservative estimate we set t_{back} equal to max contention window size to ensure we do not over-estimate t_1. All subsequent channel access can be predicted using Equation 4.2 (see Figure 4.1 case 2). Note we do not need to account for transmission delay as the generation of frames is independent of actual transmissions.

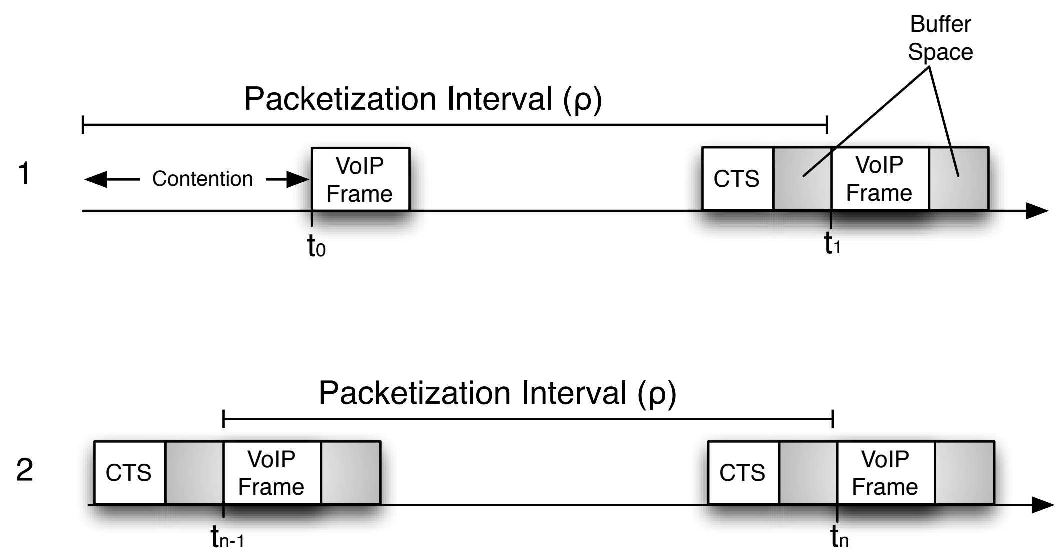


Figure 4.1: Predicting Channel Access of VoIP Frames

$$t_1 = t_0 + \rho - t_{trans} - t_{back} \tag{4.1}$$

$$t_n = t_{n-1} + \rho \tag{4.2}$$

4.3 Calculating Transmission Duration

To reserve air-time for a client we need to know not only when a client will need access to the channel but also the duration of the reservation. To calculate reservation interval we estimate the time required for the client to transmit a VoIP packet.

Figure 4.2: 802.11 Frame Transmission Overheads

Overhead	**a/b/g**	**Bytes**	**Time 802.11(μs)**
Data	all	S	S*8/R
UDP Header	all	8	8*8/R
RTP Header	all	12	12*8/R
IP Header	all	20	20*8/R
MAC Header	all	34	34*8/R
ACK	all	14	14*8/R
PHY Header	a	-	20 * 2
	b	-	196 * 2
	g	-	20 * 2
SIFS	a	-	16
	b	-	10
	g	-	10
DIFS	a	-	34
	b	-	50
	g	-	28
Signal Extension	g	-	6
Total	a	-	$90 + ((88 + S) * 8)/R$
	b	-	$452 + ((88 + S) * 8)/R$
	g	-	$84 + ((88 + S) * 8)/R$

Table 4.1: Packet Transmission Overhead [41, 31]

We already know the voice data size S from packet inspection and the other overheads are fixed (see Figure 4.2 and Table 4.1). For example the time needed to transmit a G.711 frame (S = 8) at data rate (R) = 11-Mbps is approximately 521μs including the ACK and all associated headers (Assuming 802.11b is used). In addition to this we add some buffer time before and after the expected transmission to accommodate delay variation of packet generation. Soft Speak [48] notes that 1 ms reservation leads to best results, a result which we also see in our experiments. Such a large buffer time is needed because of variability in the inter-packet interval and variability in the transmission of the unsolicited CTS. Due to the reservation there will be no contention and therefore no back-off delay. We are currently assuming that a single, fixed data rate is used but in future work we will predict the data rate used by clients to generate the expected reservation duration (see Section 6.2.1).

4.4 Unsolicited CTS Silencing

To reserve a channel for a particular client we use an approach presented in [8]. When the AP wishes to reserve the channel it sends an unsolicited Clear-to-Send (CTS). The NAV value is set to the reservation duration as calculated by the procedure specified in section 4.3. Unlike previous work, we do not use CTS-To-Self instead we direct the CTS to the client for which we wish to reserve the channel. To transmit the CTS packet the AP must acquire the channel and in order to ensure that the AP gains prompt access to the channel we prioritize the AP's access to the channel as described in section 4.7. Having to contend for channel access, albeit with higher priority, may introduce some jitter but the allocation of buffer space around transmissions and the play-out buffer of the coding scheme is sufficient to mask the variation.

4.5 Clock Drift Correction

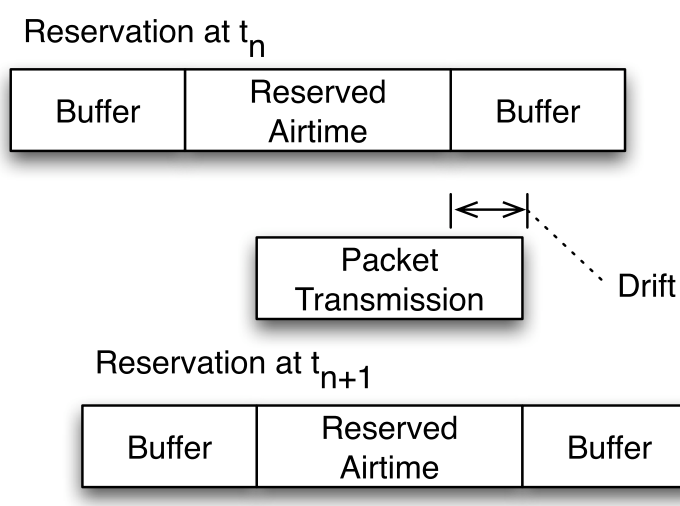


Figure 4.3: Clock Drift Correction

As mentioned earlier, we make a conservative estimate of t_1 (the expected transmission time of the first VoIP packet for which we reserve air-time) by assuming back-off delay for the first packet was equal the maximum contention window size. In worst case the actual back-off duration could be zero and our estimate of t_1 could be off by t_{back}. This causes our reservation to be too early and the VoIP transmission may be delayed. The AP observes the start time of the VoIP packet and if it is delayed then future reservations are moved back. More generally, the AP observes all client transmissions. If a client's transmissions starts moving forward or backwards with respect to expected transmission time then the AP moves the air-time reservation to correct the drift (Figure 4.3).

4.6 Supporting Talk Spurts

The biggest disadvantage of polling schemes such as PCF is that they waste air-time in sending CF-Polls and Null packets when clients have no data to send as verified by Kawata et al.[33]. In normal conversations, only one party is talking at any given time, therefore either the uplink or downlink is idle. In fact each party is silent for 50 to 60 percent of the call duration (See Brady Model [10]). If silence suppression is used, no packets are sent during these idle periods. We therefore use the scheme proposed in[33] to relinquish reservations when clients fail to use several consecutive allocations. The exact number of missed transmissions before reservations are released is dependent on the signal quality of the channel. If the channel has very little noise then missed packets are very likely due to a pause and a reservation can be released after 2 or 3 missed transmissions. If however the channel is lossy then we would wait for 5 to 10 packets before releasing reservations. Note, on average a pause is expected to last a time period equivalent to 50 more more transmissions therefore 5 to 10 unused reservations do not add excessive overhead. When the client resumes sending traffic it must contend for the channel using DCF. Upon seeing this packet we reallocate a reservation as specified in section 4.2. Using this scheme, implemented as a modification to PCF, Kawata et al. [33] were able to increase capacity by 20 percent.

4.7 Prioritizing Downlink Traffic

For each on-going VoIP call there is a stream of packets transmitted by the AP, therefore an AP transmits as many packets as all clients combined. Furthermore the AP has to transmit CTS packets to reserve the channel for clients. DCF however, is designed to be fair to all transmitters including the AP and therefore the AP will receive service equivalent to each of the clients. In order to prioritize access of the AP, we allow the AP to use a zero-length back-off as opposed to a randomized contention window. In addition, if there is an on-going transmission then the AP will utilize the next available time slot by using a smaller IFS then any of the clients. It has been shown in prior work [48] that disabling back-off may cause a drop in QoS as it hinders 802.11's collision avoidance. However, we note the result is due to the fact that multiple clients are contending for the channel without back-off. In our implementation only the AP uses zero back-off. As all other nodes back-off we do not see an increase in collisions.

Chapter 5

Evaluation

To evaluate our design, we use a 40-node testbed deployed throughout our department building. Each node comprises of a VIA EPIA EN12000EG mainboard having a 1.2GHz C7 nanoBGA2 processor with 1GB of DDR RAM. The VIA EPIA EN12000EG mainboard also has Gigabit Ethernet to connect it to the wired infrastructure. Our WLAN testbed is connected to the existing wired backbone of the computer science department. We use a separate VLAN for the testbed which isolates our traffic from the remaining network. To log wireless trace data, we use a 40GB Toshiba IDE hard-drive installed at each node. Every node is equipped with two radios, the primary Intel 2915ABG wireless card (with the ipw2200 driver) is used to perform experiments. A secondary EnGenius EMP-8602 card (based on the Atheros chipset) using the madwifi-ng-r2657 driver is used to observe the medium and capture network traces. Each node uses Debian Linux with the standard 2.6.16 kernel. The nodes have been distributed through the building mirroring the Aruba Networks wireless deployment used by the department see Figure 5.1. For more details of the testbed see [7].

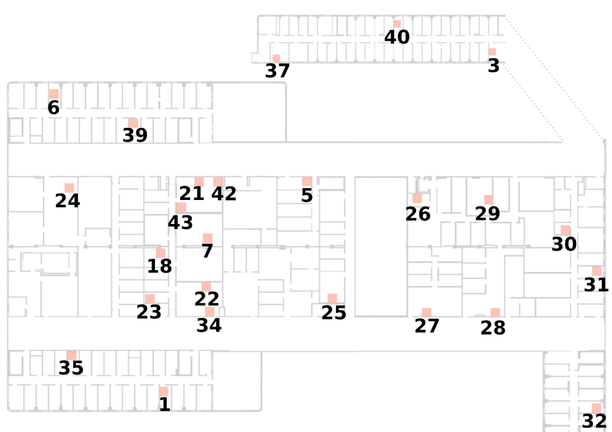
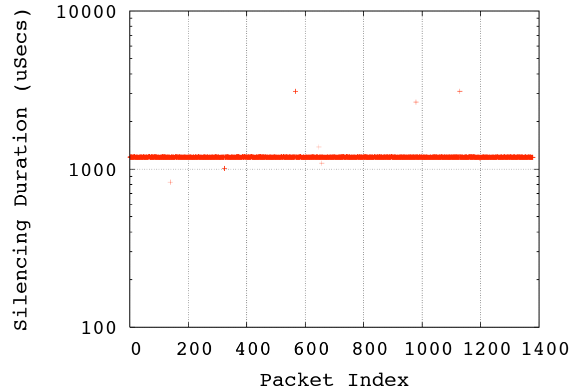
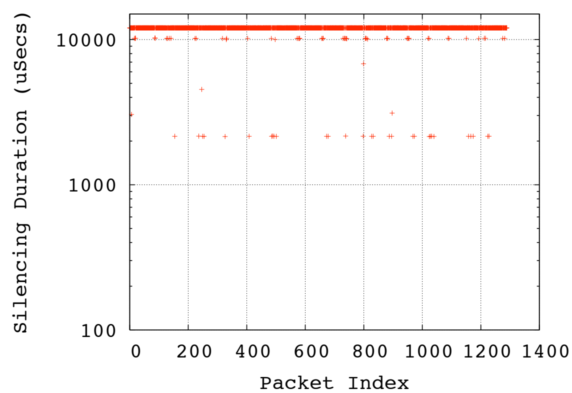


Figure 5.1: Node locations on one floor of our testbed, deployed in a 120m x 65m Office building.



(a) Using a 1 ms duration



(b) Using a 10 ms duration

Figure 5.2: Silencing using CTS with 1 ms and 10 ms duration

5.1 Silencing using CTS packets

The first requirement of our design to be evaluated is the effectiveness of CTS based silencing. The RTS/CTS mechanism is a mandatory part of the 802.11 standard and hence essential for stations to implement. However RTS/CTS can be used to launch a denial-of-service attack; a malicious node can send a RTS-to-Self and issue a CTS-to-Self blocking channel access for all other nodes. For this reason some implementations may disable RTS/CTS with large duration values. To evaluate whether we can effectively silence the medium we place two 802.11 nodes in transmission range. One node transmits data continuously, saturating the medium. The other node periodically transmits a CTS packet and then measures how long the medium remains idle. We use a CTS duration of 1 ms and 10 ms. The results are shown in Figures 5.2(a) and 5.2(b) respectively. The Y-axis shows the time between the transmission of the CTS frame and the next packet on the air, the X-axis shows the index of the CTS frame (Note: a logarithmic scale is used for the Y-axis). With a duration field of 1 ms only 1 out of almost 1400 CTS frames fails to silence the medium for the required duration. If a duration of 10 ms is used only a few tens of packets out of 1400 CTS frames fail to silence the medium correctly. We only need the ability to silence the medium for 1 ms so we should not have a problem using CTS based silencing.

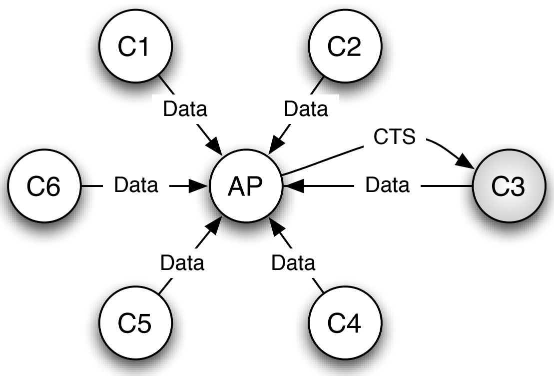


Figure 5.3: Directed CTS Experimental Setup

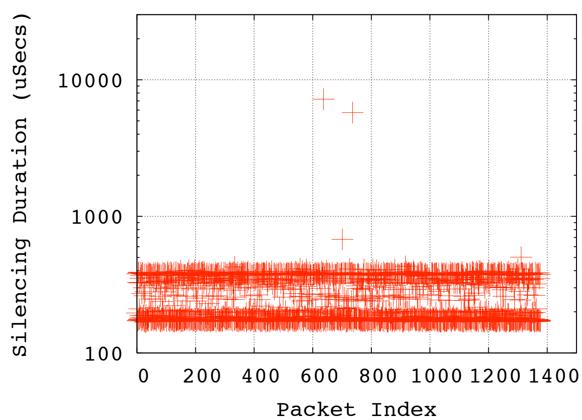


Figure 5.4: Directed Silencing

5.2 Directed Silencing using unsolicited CTS packets

In order to clear the air for a particular client we need to ensure it will avail the transmission opportunity given to it by the unsolicited CTS. As it did not issue a RTS for which it received the CTS we cannot take it for granted that it will be able to transmit. To verify this we collocate several 802.11 nodes and have them all generate saturated traffic. We then have the Access Point (AP) periodically send

CTS packets directed to one of the original nodes. For example in Figure 5.3 we are trying to clear the air for the transmissions from client node *C3*. If our approach is to work then all nodes must continue to obey the CTS, and update their NAV counters, however the node to which the CTS is directed must not be affected. Figure 5.4 shows the interval between the transmission of the CTS frame and the next packet from *C3*. We use a duration field of 1 ms but as shown in the figure almost all the frames were transmitted after 200 to 400 μs confirming the fact that the NAV counter is not updated. Although it is not shown, all other nodes backed off for 1 ms and the delay graphs look identical to figure 5.2(a).

5.3 QoS improvement using silencing

The next question we evaluate is whether clearing the air for a client will significantly improve QoS for VoIP clients. We focus on the delay-jitter and packet loss QoS metrics. End-to-end delay can be ignored because delay on the Internet is an order of magnitude bigger then the additional delay caused by 802.11 contention.

To evaluate the improvement in QoS we set up the experiment described in Section 5.2 with several nodes generating saturated traffic and the AP sending periodic CTS frames for a single client. However, *C3* sends G.729 VoIP frames at 20 ms (20,000 μs) intervals instead of saturated traffic. We measure the Inter-Packet Arrival Time (IPAT), i.e., time between two consecutive packets. Ideally, we would like this value to be close to 20 ms for all packets originating at *C3*. However the interfering nodes will cause delays which manifest themselves as an increase in the IPAT between some packets and a reduction in others. If a packet is delayed it is further apart in time from the previous packet and closer in time to the subsequent packet. A lost packet will show up as a large increase in IPAT, in fact we expect it increase by 20 ms for every consecutive packet loss. Figure 5.5 shows the spread of IPAT without CTS silencing. The X-axis is the offset of a packets IPAT from 20 ms, i.e., if the IPAT is equal to 25 ms then the offset is 5 ms or 5000μs. The Y-axis shows the percentage of packets which were delayed by the selected offset. As the figure shows a large portion of the packets are significantly far from the 20 ms ideal. However when we enable CTS silencing for the client then all the delay variation collapses to few tens of microseconds (see Figure 5.6).

5.4 Prioritizing Downlink Traffic

As mentioned in Section 4.4 CTS silencing will only improve QoS for uplink traffic. For downlink traffic we implement a zero back-off policy for VoIP packets. We now evaluate the effectiveness of downlink traffic prioritization using the zero back-off policy. Although this approach has been proposed and evaluated in prior literature [9] we wish to verify results using our implementation. The experiment is set up as follows, several 802.11 nodes are co-located and associated with a single Access

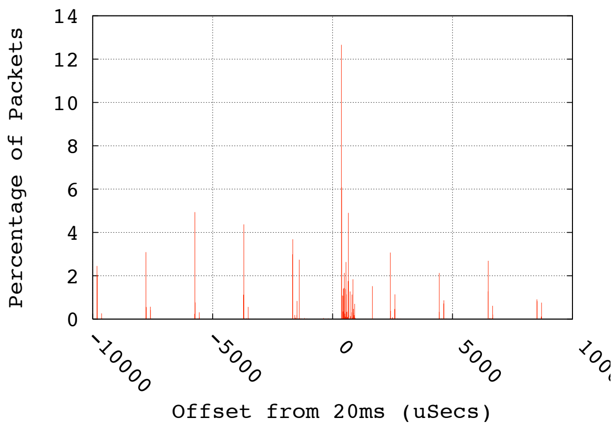


Figure 5.5: Delay Jitter with Saturated Interferers

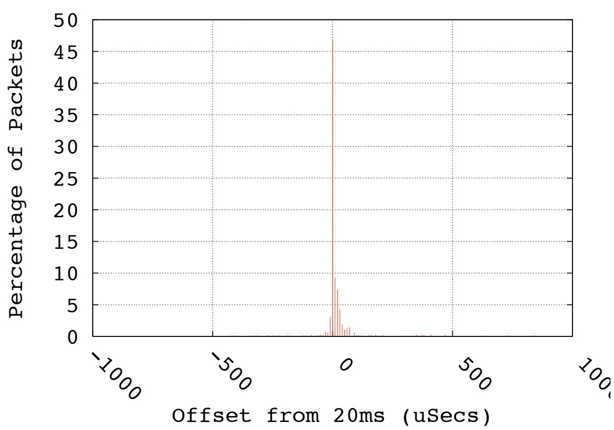


Figure 5.6: Delay Jitter with Saturated Interferers and CTS silencing

Point creating a BSS. Each client node generates a single VoIP uplink stream of G.729 frames at 20 ms intervals. The AP generates a corresponding VoIP downstream for each of the clients. Using the secondary radio on the AP, we measure the Inter-Packet Arrival Time (IPAT) of downlink packets. Any packet delayed by more then 5 ms is considered lost because we are assuming a 5 ms jitter buffer as recommended in the G.729 standard [30]. Over the course of the experiment we increase the number of clients (and downlink streams) and measure the downlink loss rate corresponding to the number of clients. As shown in Figure 5.7, the loss rate increases with the number of clients. We run this experiment once with normal 802.11 DCF and once with zero back-off for downlink VoIP packets[1]. Figure 5.7 shows that after about five clients there is a noticeable difference in packet loss rate when using zero back-off. We see much more dramatic increase in loss rate when we add non-real-time, 3 Mbps Constant Bit Rate (CBR) UDP traffic, as shown

[1]Note the zero back-off results are labeled as Virtual PCF

in Figure 5.8. With background CBR traffic, after 3 clients the packet error rate grows significantly when no prioritization is used. However, with zero back-off the error rate remains at 2%.

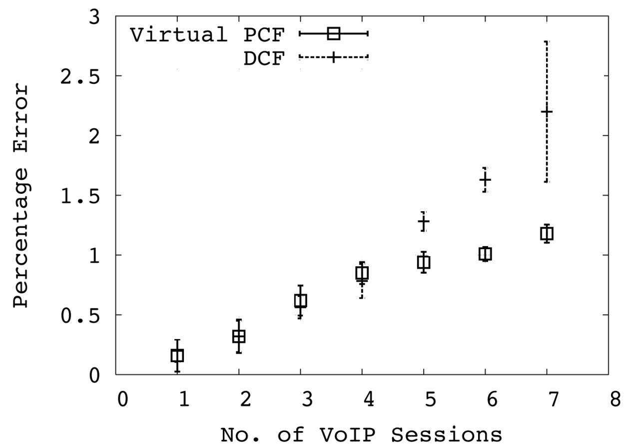


Figure 5.7: Downlink Error Rate With and Without Prioritization

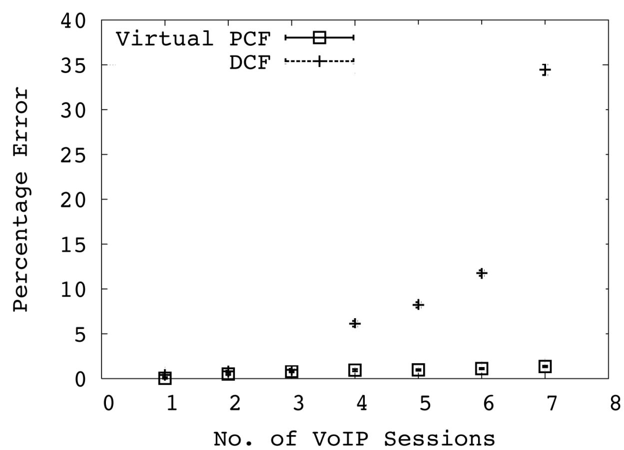


Figure 5.8: Downlink Error Rate With Cross Traffic

Due to the fact that we are prioritizing downlink traffic over uplink traffic, the increase in downlink QoS comes at the expense of uplink QoS. However, note that the increase in loss rate will not be significant because uplink VoIP at any client is an order of magnitude smaller then downlink traffic from the AP. The measurements shown in Figure 5.9 confirm our intuition. Although there is an increase in uplink error with the number of clients, it remains below 1.5%. Prior literature [1] suggests a 2% threshold as acceptable loss rate for VoIP traffic and therefore the drop in QoS is not enough to reduce the number of supported VoIP sessions.

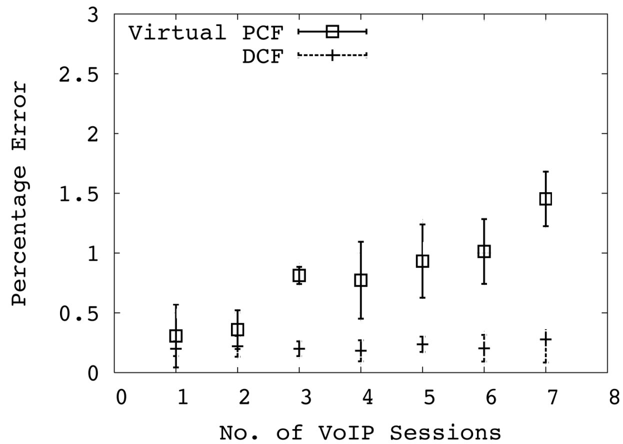


Figure 5.9: Uplink Error Rate With and Without Prioritization

5.5 Increase in VoIP Capacity using Virtual PCF

We have already seen that prioritizing downlink traffic lowers error rates for downlink traffic. We now study the effectiveness of uplink scheduling using unsolicited CTS frames. The experimental setup is exactly as in Section 5.4 with several colocated VoIP clients associated with a single AP and one node generating uplink three Mbit CBR UDP traffic. We again increase the number of VoIP clients, and downlink streams, and measure the corresponding change in loss rate. We collect a network trace for each experiment to identify how many frames are lost or delayed beyond the buffer. We repeat every experiment five times in order to analyze the statistical significance of our findings. The results are shown in Figure 5.10 and it is evident that if four or more clients are active the loss rate is too high to support VoIP. With Virtual PCF the growth in error is much slower even with as many as seven active clients. The five percent error bars (also shown in Figure 5.10) show that the improvement is well beyond the margin of error.

If we combine the uplink and downlink results for Virtual PCF together and compare them with DCF we can see the increased VoIP client capacity. Figure 5.11 shows the uplink and downlink capacity for DCF-based VoIP. Using a 2% error threshold we see that no more than three clients can be supported. However with Virtual PCF (Figure 5.12) 6 clients can be supported on the uplink, a 100% increase over DCF. If seven clients are active then the MOS increases from 1.84 to 4.2 for downlink traffic and from 2.98 to 4.02 in the uplink direction. Note that an MOS value of 4 is normally required for good quality voice communication. (This does not account for the drop in MOS that will be caused by transmission over the wired network and the Internet.)

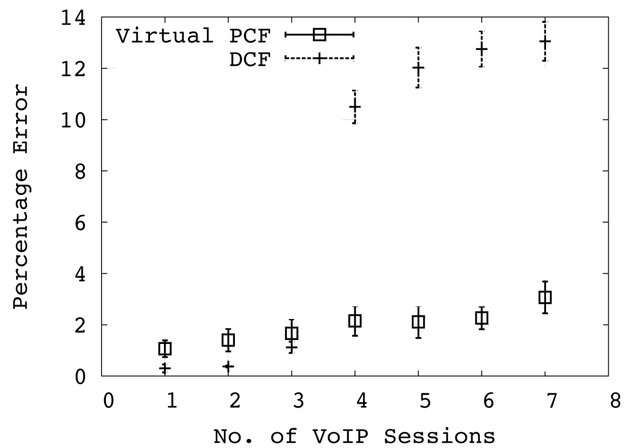


Figure 5.10: Uplink Error Rate With Cross Traffic

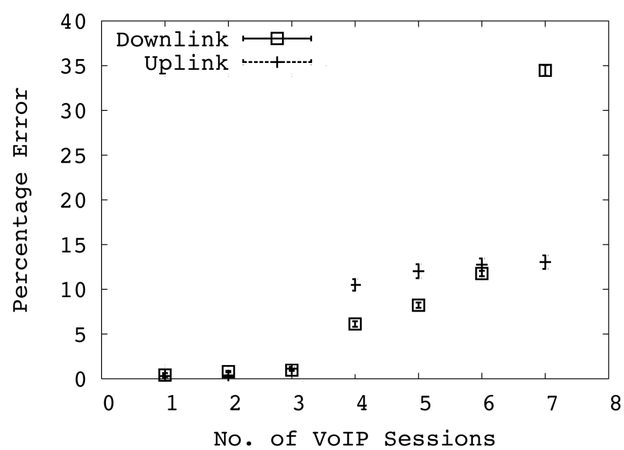


Figure 5.11: VoIP over DCF with Cross Traffic

5.6 iPhone Clients

In order to evaluate whether our implementation is compatible with commodity wifi devices we need to replicate our results using commodity clients. To this end we used six first generation 2g iPhones running firmware version 2.0 as VoIP clients. For our approach to work we must establish that the wireless NIC in the devices obeys CTS frames and utilizes slots allocated via unsolicited CTS frames. Figure 5.13 shows the inter-arrival time of packets which were sent from an iPhone at 3ms intervals. Figure 5.14 shows the inter-arrival time of packets sent at the same rate but with the AP generating a CTS frame with a 10ms NAV value. It is clear that the iPhone wifi chipset is obeying the CTS as there are no packets transmitted with an inter-arrival time less that 10ms. Lastly, we direct the CTS frames to a particular client and observe its back-off behaviour. Figure 5.15 shows that the client ignored the NAV value of directed CTS frames and thus used the transmission

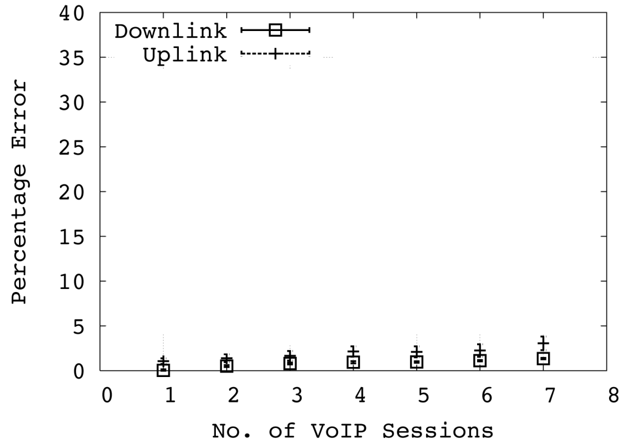


Figure 5.12: VoIP over Virtual PCF with Cross Traffic

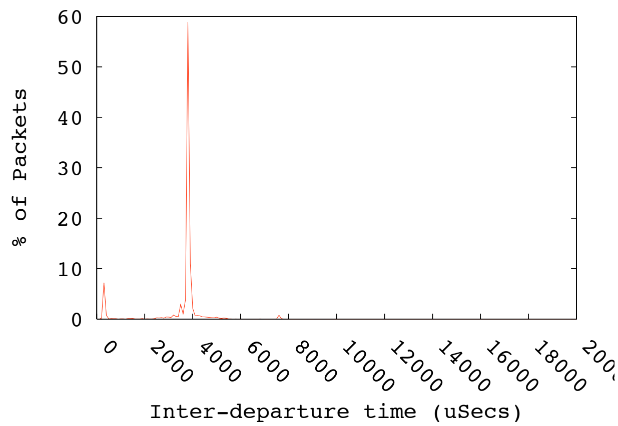


Figure 5.13: iPhone traffic generated at 3ms with no CTS

opportunity allocated to it via the directed CTS. We therefore conclude that our implementation is compatible with iPhones. The remainder of this section evaluates the improvement in the number of VoIP calls supported using our implementation.

To determine the improvement we set up the experiment discussed in section 5.5 with iPhones as clients. The iPhone does not support 802.11a [24]. Using the 802.11b frequency band forced us to conduct our experiments on the same channels used by our campus WLAN deployment. This is in contrast to our earlier experiments which were carried out in an isolated channel. This however provides insight into the operation of our implementation in scenarios with access points beyond administrative control. As shown by Figure 5.16 we are still able to improve VoIP QoS. Virtual PCF increases the number of calls supported (according to the two percent loss-rate rule) from three to five. If we introduce our own 3-Mbit CBR interference (see Figure 5.17) then even a single station is not supported over DCF. However with Virtual PCF we can support as many as four clients. The

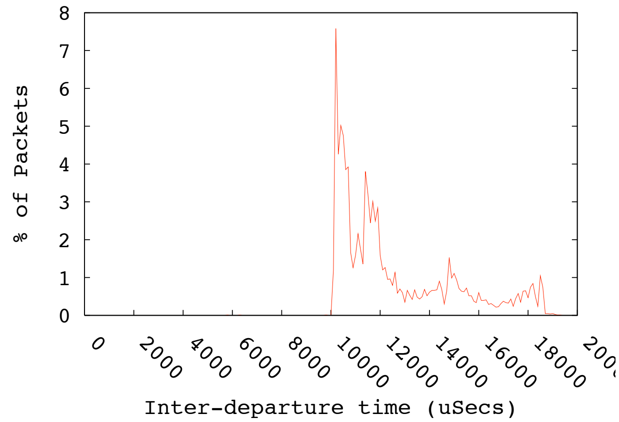


Figure 5.14: iPhone traffic generated at 3ms with 10ms CTS

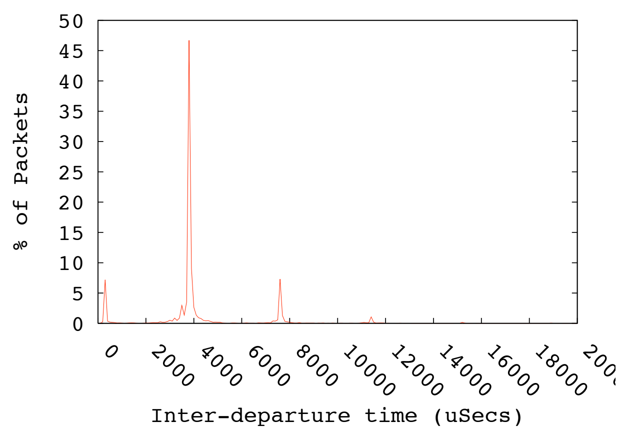


Figure 5.15: iPhone traffic generated at 3ms with directed 10 ms CTS

improvement in number of clients is not as dramatic as we saw when we ran our experiments in an isolated channel. This is due to the fact that some wireless devices (APs and Clients) are not obeying our CTS frames. We confirmed our suspicion as shown in Figure 5.18, where we transmit a CTS frame with a 10ms NAV value and then observe the first packet on the air. Although there is a concentration of packets transmit 11 to 12 ms after the CTS, which we assume is due to CTS, there are a large number of packets which ignored the CTS. These results give us confidence that our implementation is compatible with commodity devices such as the iPhones. Furthermore they show that our approach is effective even in the presence of some nodes that disobey our silencing.

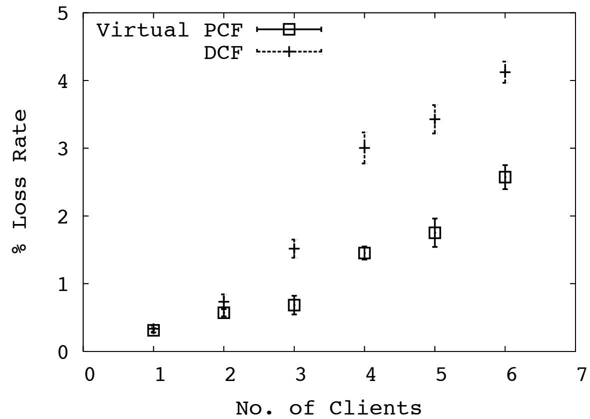


Figure 5.16: iPhone VoIP sessions with background interference

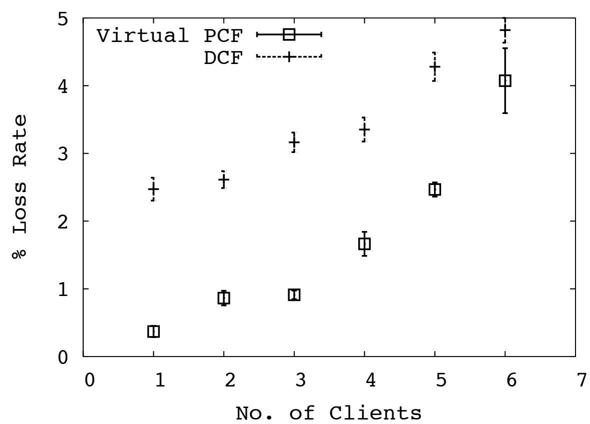


Figure 5.17: iPhone VoIP sessions with background and CBR interference

5.7 Increase in VoIP capacity with talk spurts

Until now all results have used uniform VoIP traffic without any pauses or talk-spurts. However, as mentioned before, VoIP traffic tends to be composed of talk-spurts followed by long pauses. In fact, prior literature [10] suggests that a VoIP stream is silent for the majority of its duration. We use a model similar to the one presented in [10] to study the improvement offered by our approach over DCF on "realistic" traffic. The experimental setup is identical to the one in Section 5.5 except that all VoIP clients now generate traffic in spurts. To generate spurts and pauses each client (and downstream from AP) generates packets at a constant rate for a random time interval with a mean of one second. The client then stops generating packets for a random time period with a mean of 1.5s. This conforms to recommendations of the Brady Model [10]. The results are shown in Figure 5.19 and 5.20, we see that the downlink client capacity increases from six to eleven and

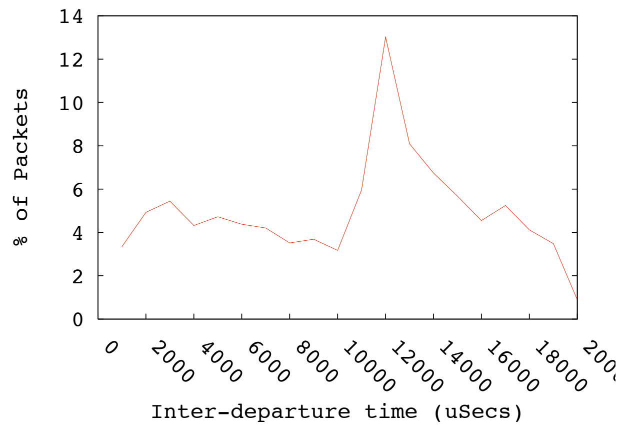


Figure 5.18: Silencing of Campus Network

the uplink capacity increases from seven to ten. Hence the overall client capacity increases from six to ten. The results are less dramatic than the uniform VoIP case. However, we still see a 60% increase in capacity. The reason the change is not as great as with the uniform VoIP case is that at the start of a VoIP talk spurt our approach needs time to synchronize the client and the AP, and during this time higher losses are seen. With uniform traffic this only occurs once per experiment but with talk spurts it happens more frequently. In terms of MOS the uplink MOS with 10 clients increases from 4.04 to 4.13 and downlink MOS increases from 4.09 to 4.18.

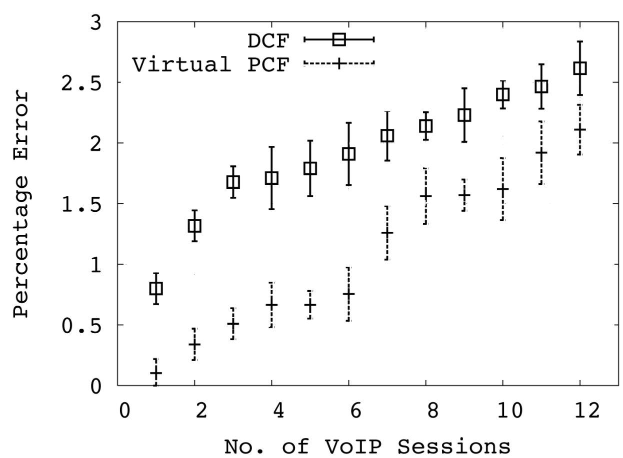


Figure 5.19: Downlink Talk Spurts with Cross Traffic

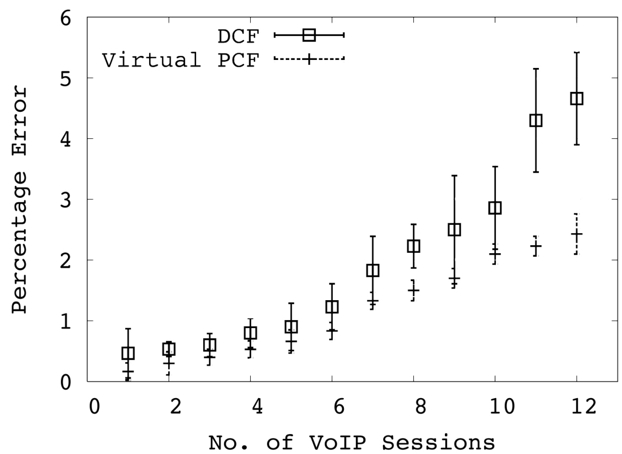


Figure 5.20: Uplink Talk Spurts with Cross Traffic

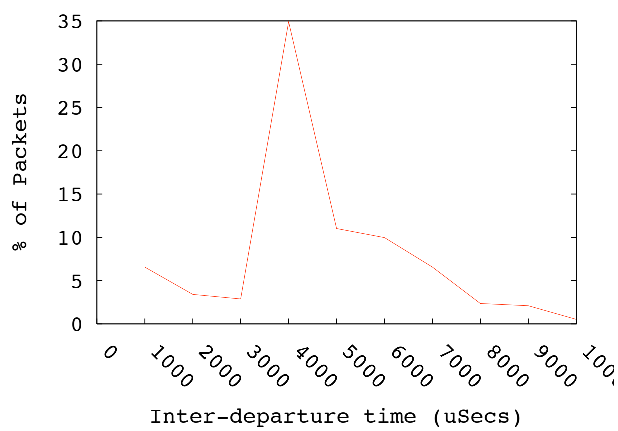


Figure 5.21: Fring Traffic Patterns

5.8 Virtual PCF with real-world clients

In order to use the Directed CTS packets we must be able to accurately predict when a VoIP client will next require channel access. Section 4.2 discusses a formulation to generate such predictions. However, the formulation assumes that VoIP clients generate traffic with predictable inter-packet delay. While this is true for well known protocols, in this section we analyze whether the assumption holds for real-world VoIP clients. For our analysis we use the Fring [4] client with the Skype [5] network. We run the client for 30s observe all packets it generates. Figure 5.21 shows the distribution of the delay between consecutive Fring packets. The horizontal axis shows the inter-packet interval and the vertical axis shows the percentage of packets with the corresponding interval. As the figure shows, the majority of packets are transmitted within 4-ms (4000 μs) of each other but some packets are delayed as much as 8-ms. This variability makes it difficult to clear the air for packets using a

1-ms CTS. A longer reservation is not possible as this would reduce the number of reservations. For example with a 4-ms inter-packet interval and a 1-ms reservation we can allow 2 simultaneous bi-directional VoIP streams. If we use a 2-ms interval then we can only allow one such stream, albeit with a higher QoS.

However, we note that as long as the variation in inter-packet interval does not exceed the jitter buffer our approach is still effective. This is because if a packet misses its reservation it must wait for the next reservation. During this time the next packet will be enqueued behind the first. The reservation is only large enough for a single packet to be transmitted, therefore the second packet must also wait. This effectively adds a consistent delay to all packets. The additional delay absorbs variations in the inter-packet interval. For example, in Figure 5.22 Packet one is generated too late and its reservation is wasted. The packet is transmitted in the subsequent reservation, intended for packet two. Packet two is transmitted in the reservation intended for three and all subsequent packets are delayed by one reservation. As long as the aggregate rate of packet generation is equal to the interval between reservations there will be a consistent rate of transmission of packets. Furthermore, as there is no DCF contention there is less wasted airtime and no unfairness between VoIP streams. We evaluate this by running a Fring client with and without Virtual PCF for 30s each. We then compare the inter-packet interval times in the two cases. Figure 5.23 shows the inter-packet times with Virtual PCF and you can see a marked improvement in comparison to the original inter-packet intervals shown in Figure 5.21. Note, we do not show loss rate results because packets delayed beyond jitter buffer are considered lost and we do not know the jitter buffer size of the fring client.

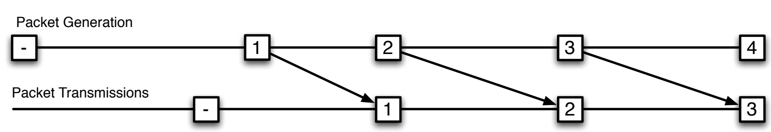


Figure 5.22: Delayed Fring Traffic

5.9 Conclusion

In this section we have shown that we are able to silence the medium for a particular client using unsolicited CTS frames and that using this approach we can prioritize its traffic. We have also shown that using downlink prioritization we can improve downlink VoIP over WLAN capacity without causing significant loss in uplink capacity. By combining these approaches into a single implementation entitled Virtual PCF we have found a 100% increase in VoIP capacity using uniform traffic patterns. Although realistic traffic patterns are circumspect, using one common model we have shown a 60% improvement in capacity. Furthermore, we

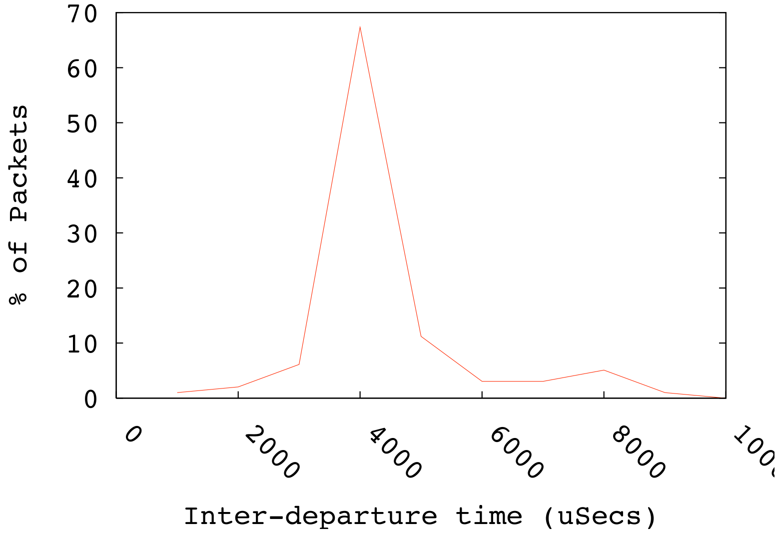


Figure 5.23: Fring traffic with Virtual PCF

have tested one real-world VoIP client (fring) and are able to improve the delay variation of its packets. This improvement comes as a result of changes made only to the AP firmware and driver, and does not require modification of clients or AP hardware.

Chapter 6

Conclusions and Future Work

6.1 Contributions

The contribution of this work is to provide a practical approach (Virtual PCF) to improve VoIP over WLAN capacity without requiring any change to existing hardware. We provide a comprehensive solution to improving QoS of both uplink and downlink VoIP streams without violating the 802.11 protocol. Our approach can be used with commodity clients (laptops and smartphones) already owned by users. We only require changes to software on access points. Virtual PCF improves QoS both when VoIP traffic is on an isolated channel and when it is sharing the channel with data traffic. Our approach significantly improves QoS in scenarios where VoIP traffic and data traffic is co-located, by 60% to 100%. As most modern clients are running many applications in parallel running VoIP traffic on an isolated channel is not practical. Therefore improvements in this scenario are particularly valuable.

Virtual PCF is compatible with real-world practices and technologies. We support talk spurts and silence suppression because most modern VoIP codecs use such techniques. Although not as dramatic as with uniform traffic, we still see an improvement of 60% when using Virtual PCF with talk spurts. As part of our effort to ensure real-world compatibility, we evaluate our approach on a real testbed instead of a simulated environment. In addition to using physical hardware, commodity drivers and firmware for our access points and clients we also located the nodes randomly throughout our department building (The Davis Center). This building, at any given time, is home to several hundred undergraduate students, graduate students, faculty and administrative staff. This creates a dynamic radio propagation environment and gives us confidence our results can be replicated in any scenario.

Despite our efforts there are still some challenges that need to be addressed before Virtual PCF can be deployed in commodity systems. We hope to address these challenges in future work and are presented in Section 6.2.

6.2 Limitations and Future Work

6.2.1 802.11 Multi-rate traffic

We currently assume that a single fixed data rate is used for the duration of a VoIP session. This is because the data rate affects the expected transmission duration of a packet and therefore the size of the air-time reservation. If auto-rate selection is used the data rate will change dynamically and we will have to update the reservation size. Furthermore, the data rate used by a client for each packet can change and we have no information regarding this change at the access point which makes the reservations. In order to address this issue we can make the assumption that between two consecutive VoIP packets the data rate can change by only one, i.e., a packet can only use a rate one higher or one lower then the previous one. We can then update the reservation based on the data rate used by the last packet from a client. We also provision enough buffer space in the reservation so that if a client uses the lower rate it still has enough time to transmit its data. We hope to implement and evaluate this approach in future work.

6.2.2 Local Cross-Traffic

If a VoIP client (or the access point) is sending a lot of non-real-time traffic it may cause local queuing delays. These delays are problematic because we reserve the channel for a client for a short time. If the client uses its reserved slot for data packets then it will not see any benefit from Virtual PCF. We can rectify this problem at the AP by using a separate queue but this not possible on legacy-clients. Another solution is to use a larger silencing duration to increase the probability that the VoIP packet will reach the head of the queue while silencing is in effect. However, increasing the silencing duration for each client will reduce the total number of clients that can be supported. There is a trade-off between the reduction in the number of supported clients (due to higher loss because of missed silencers) and the reduction in number of supported clients because air-time is saturated by long silencing slots.

In order to determine the impact of local cross traffic on the VoIP stream we set up an experiment with two nodes, one acting as the Access Point (sending CTS silencers) and one generating a single VoIP stream. We then vary the duration of silencing of the CTS packets and also the amount of data traffic. We measure the loss rate in the VoIP stream and find the amount of local-cross traffic required for the error to exceed 2% [1]. Our results (See figure 6.1) show that at a rate of 6-Mbps and a 3 ms silencing interval we can have over 2.5-Mbps of data traffic. If a data rate of 12-Mbps is used then a 2 ms silencing interval shows similar results. The results illustrate that a reasonably large amount of local cross traffic can be

[1] 2% loss rate is defined as an acceptable loss rate in prior literature [1]

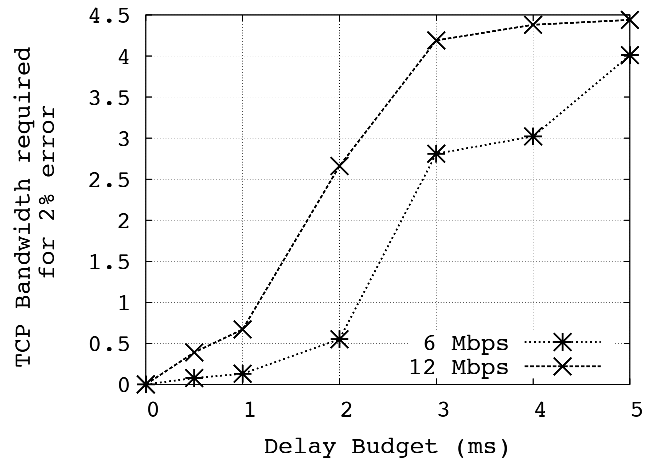


Figure 6.1: Local data bandwidth required to cause 2% error

tolerated. As long as users do not generate too much non real-time traffic, their VoIP sessions should be unaffected by local cross traffic.

We are exploring ways to actively deal with local cross traffic without requiring client modification. One avenue we are plan to study in future work is to actively monitor the uplink traffic and detect if a client utilized its reservation for a VoIP packet. If the reservation is used for data traffic we temporarily allocate another slot to the client at the earliest available time. This procedure may have to be repeated several times if there are multiple packets in front of the VoIP packet. Note that this will not disturb reservations of other clients because we only issue extra transmission slots if the air-time is not reserved for another client.

Furthermore, we note that the procedure described above has two shortcomings, First, we can only issue extra transmission slots if the medium is relatively idle. Second, we may only issue extra transmission slots if the delay in the packet is smaller then the jitter buffer size. For example, if the jitter buffer is 5 ms then we can only issue transmission slots up to 5 ms after the expected packet transmission time. In practice, at lower data rates, this means that only one or two data packets need to be in front of a VoIP packet in order for the delay to exceed jitter buffer size. In order to address this issue, if an AP observes that a client is consistently utilizing reserved slots for data traffic then it can randomly drop some TCP data packets. This will cause a TCP back-off, if the packets are UDP then we have no way of forcing the client to back-off.

6.2.3 Air-Time Fragmentation

Virtual PCF assigns transmission slots at arbitrary times depending on the expectation of when a client will try to access the channel. Unlike other reservation protocols such as Reservation Aloha there is no requirement that the boundaries of the reserved slots coincide. In other words, there can be an arbitrarily long time interval between two successive reservations. This may lead to a situation where the unreserved time interval between two consecutive reserved slots is not large enough to transmit another packet. If this happens often there may be a situation where the total free airtime is sufficient to transmit data but there is no single contiguous transmission opportunity. We term this phenomenon *Airtime Fragmentation*. To prevent this scenario, we can move the preceding allocation back to absorb the airtime. This may cause additional delay but the added delay should be well within the jitter buffer interval of the VoIP session. Note that the time taken to transmit a data packet is much larger then the time taken to transmit VoIP packets. Therefore there may be many gaps that are too large to absorb but still not large enough to allow transmission of data traffic. Therefore we may cause excessive delays to non-real-time traffic in certain scenarios. If this is the case we will need to use an admission control scheme such as [18] to limit the number of simultaneous VoIP calls.

6.2.4 Lack of Mobility

A large number of WiFi enabled devices sold today are small-form-factor hand held-devices. Users of such devices have a natural tendency to be mobile while using them. This is problematic as our approach does not support cross-BSS mobility. This is mainly because seamless hand-off between access points is a difficult problem. Moving from one access point to another requires re-association which is a process that can take over a second [35]. During this interval the users call will be disrupted. There are several proposals on achieving faster re-associations or seamless hand-off; [37], [40], and if one is widely implemented we can extend our approach to interoperate with that scheme.

6.2.5 Encrypted / Obfuscated Traffic

Our approach relies on examining characteristics of VoIP traffic in order to identify the encoding scheme used and also the relevant configuration parameters such as packetization interval and jitter buffer size. However, some protocols, especially proprietary protocols, may try to obfuscate this information and avoid detection. In addition, if end-to-end encryption is used we may not be able to discern the configurations used by the clients during the VoIP session. For this reason, in a real-world deployment of our approach, we constrain users to use a VoIP client using a known open encoding scheme and to not use encryption. Users who use other clients will not benefit from our approach unless they use a protocol which is known to the system *a priori*. We do not feel this is a strong requirement because few protocols are in common use and new protocols can be added at any time if means are discovered to identify their packets.

6.2.6 Malicious Clients

In our current discussion we have assumed all clients act in good faith. However, in the real world this may not be the case. Some clients may use this protocol to prioritize their non-real-time traffic or to gain unfair channel access. In addition, a malicious client may use the airtime reservation system to craft a denial-of-service (DOS) attack against the network. We have not developed a complete threat model and hence cannot give a guarantee of security. However, we note that a DOS attack is possible using unsolicited CTS frames regardless of whether our approach is used. In addition, VoIP traffic is known to use little bandwidth in comparison to data traffic. Hence any attempt to prioritize data traffic should be fairly evident.

6.2.7 Accurate sub-millisecond event scheduling

For our evaluation we have used a data rate of 6-Mbps at which the transmission of a frame takes to the order of several hundred microseconds. We are using a reservation size of 1 ms for each VoIP frame. Our current implementation makes it difficult to achieve tighter synchronization of such ephemeral events. However if our approach is to be used at higher rates then we would need smaller reservations and more accurate synchronization of events. Most commercial access points use a real-time operating system that would allow such tight synchronization.

6.2.8 Cross-Layer Optimizations Required

In order to detect VoIP traffic we intercept IP, UDP and application layer headers contained in 802.11 frames from within the 802.11 NIC driver. This is possible because all Linux/Unix based systems use a structure called a Socket Buffer (SkBuff) to store packets. In order to make the processing of packets efficient, Linux/Unix uses a zero copy approach. The SkBuff is held in memory shared between all layers of the network stack. Each layer adds or removes headers by moving pointers but does not allocate or release memory. This makes it possible for us to process headers from all layers at all other layers, which is essential for packet inspection in the driver. Our approach is not implementable if we are not able to access the headers from other layers at the MAC layer.

6.2.9 CTS Silencing Support

For our approach to work we require clients to back-off in response to CTS frames, we have verified this to be the case for the Intel 2915ABG wireless card, with the ipw2200 driver (see Section 5.2) and also on the EnGenius EMP-8602 card (based on the Atheros chipset) using the madwifi-ng-r2657 driver. However due to the possibility of a denial-of-service attack using such frames, some platforms may ignore CTS frames. In such a scenario our approach will not be practicable. We however note that RTS/CTS is a mandatory part of the 802.11 specification and therefore we expect most clients to support this functionality.

6.3 Concluding Note

We have presented in this thesis a legacy compatible approach (Virtual PCF) to providing scheduled uplink traffic similar to PCF and Soft Speak [48]. Virtual PCF runs over DCF and requires no modification to clients and only software modifications on the Access Points. Although there are still several issues that need to be addressed to the best of our knowledge we present the only legacy-client compatible approach to improving both uplink and downlink traffic. We envisage our approach being used in campus and enterprise WiFi deployments. In such a scenario, this will be a much more cost-effective solution than providing specialized VoIP AP and clients which need to be isolated from the normal data network.

APPENDICES

Appendix A

Hidden Terminal Interference

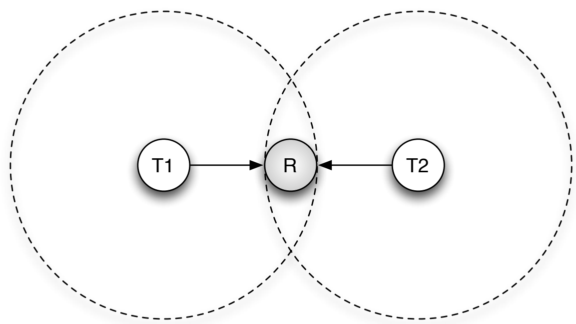


Figure A.1: Hidden Terminal Interference

Hidden terminal interference occurs when a receiver is able to decode the signal from two transmitters but the two are not able to carrier sense each other. For example in Figure A.1 receiver *R1* is able to receive transmission from both transmitters, *T1* and *T2*. However *T1* and *T2* are not within each other's carrier sense range. In such a scenario both transmitters may transmit simultaneously and will not detect each others transmission and hence will not back-off. If one of the two transmissions is destined to *R1* it will not be decoded as the two transmission collide causing the signals to degenerate into noise. Hidden terminal interference is addressed in the 802.11 using the RTS/CTS mechanism. If RTS/CTS is enabled then transmitter *T1* will first send a small RTS (Request-to-Send) frame to receiver *R1*. This frame contains a duration field which specifies how long *T1* needs to transmit its data frame. All stations which receive the RTS will back-off and allow *T1* access to the medium. However as we already know *T2* is not able to decode transmissions from *T1* so in response to an RTS *R1* generates a CTS (Clear-To-Send) frame. All stations which receive the CTS frame also back-off and allow *T1* to access the medium without interference.

Appendix B

Data Frame Sub-Types

Frame Type	**DCF**	**PCF**
Data	$\surd$	
Data + CF-ACK		$\surd$
Data + CF-POLL		$\surd$
Data + CF-POLL + CF-POLL		$\surd$
NULL	$\surd$	
CF-ACK		$\surd$
CF-POLL		$\surd$
CF-ACK + CF-POLL		$\surd$

Table B.1: Possible frame Sub-types of Data frames

Appendix C

MOS Calculations

The following section describes computation of the Extended E Model as presented in the ETSI standard [46].

$$MOS = \begin{cases} 1 & \text{if } R \leq 0, \\ \text{1+0.035R+R(R-60)*(100-R)*(7*10}^{-6}\text{)} & \text{if } 0 > R < 100, \\ 4.5 & \text{if } R \geq 100 \end{cases} \tag{C.1}$$

To calculate the MOS we first calculate the transmission quality factor 'R', which can then be converted to MOS using Equation C.1, first presented in [14]. R can be calculated using Equation C.2 where R_0 is the signal-to-noise ratio and I$_s$ is a combination of all impairments which occur simultaneously with the voice signal. Both variables are independent of the underlying network,. We therefore use default values of 94.77 and 1.41 for R_0 and I$_s$ respectively, as prescribed by the standard [46]. I$_d$ represents the impairments caused by delay, however we note that the delay on the wireless link is small portion of the end-to-end delay and therefore has little impact on MOS. We therefore omit the I$_d$ delay impairment factor from our calculations. 'A' the Advantage factor is used to account for users' expectation of voice quality based on the type of network that is being used. A conventional "wire-bound" network has an advantage factor of zero however networks providing mobility within a geographical area are expected to offer compromised service and hence receive a 10 point advantage score. We use an advantage score of 5 as per recommendation of [29] because WLANs provide mobility within a building. Finally, I$_e$ represents the equipment impairment factor and encompasses: codec affects, packet loss and delay jitter.

$$R = R_0 - I_s - I_d - I_e + A \tag{C.2}$$

C.1 Equipment Impairment Factor

$$I_e = I_e(LOSS) + I_e(PDV) + I_e(CODEC) \tag{C.3}$$

The equipment impairment factor (Equation C.3) is a function of impairment due to the codec used, packet loss and Packet Delay Variation (PDV). The impairment due to various codecs is standardized in ITU Recommendation G.113 [27], several common codecs and their impairment values are shown in Table C.1.

Codec	Bit-rate	I_e
G.726, G.727	40 Kbps	2
G.721(1988), G.726, G.727	32 Kbps	7
G.726, G.727	24 Kbps	25
G.726, G.727	16 Kbps	50
G.728	16 Kbps	7
G.728	12.8 Kbps	20
G.729	8 Kbps	10
G.729-A + VAD	8 Kbps	11
G.723.1	5.3Kbps	19
G.723.1	6.3Kbps	15

Table C.1: Selected Codecs and their corresponding I_e values

C.2 Impairment due to Packet Delay Variation

The impairment due to delay variation is computed using the average adjusted inter-arrival time of packets. If the packet inter-arrival time 't' is smaller than the codec jitter buffer delay 't_{jb}' then the voice quality is not effected as the buffer will correct the jitter. If the inter-arrival time is larger then the discard time $t_{discard}$ then the packet is dropped and the drop in quality is reflected by impairment due to loss. If the inter-arrival time is in between these two limits then impairment due to jitter is given by the adjusted average inter-arrival time as shown in Equation C.4 and C.5.

$$t' = \begin{cases} 0 & \text{if } t < t_{jb}, \\ t - t_{jb} & \text{if } t_{jb} \leq \text{t} \leq t_{discard} \\ \text{Discard Measurement} & \text{if } t > t_{discard} \end{cases} \tag{C.4}$$

$$I_e(PDV) = 0.1 * Avg(t') \tag{C.5}$$

C.3 Impairment due to Packet Loss

To compute the impairment due packet loss, we note that it is important consider the density of losses. For example, losing one packet every second and losing 10 packets in a second every ten seconds will lead to different perceptions about call quality. For this reason, impairment due to packet losses is modeled in reference to *Burst* or *Gap* of errors (Note that it is unrelated to talk bursts). If the number of packets between two successive lost packets is greater then g_{min}, then the lost packets and all packets between the two are considered part of a burst of errors. If a sequence of g_{min} or more packets is received correctly then that sequence is said to be part of a gap in errors. In addition, it has been shown that changes in actual quality of do not lead to sudden changes in perceived quality. Prior work [44] shows that it takes about four to five seconds for perceived quality to fall in response to a sudden fall in actual quality and 15 to 30 seconds for perceived quality to return to actual quality in response to a sudden improvement. The change in perceived quality follows an exponential curve. These factors are formalized into Equation C.6, where b and g are the length in seconds of a burst and gap respectively. I_{eb} and I_{eg} are these burst and gap equipment impairment factors respectively. t_1 and t_2 are estimates of the time taken for perceived quality to match actual quality in case of deterioration and improvement respectively, note t_1 is normally set to 5 and t_2 to 15. I_1 and I_2 are the time averages of the perceived quality as it changes from gap to burst value and burst to gap value respectively. The detailed derivation of this equation can be found in [46].

$$I_e(LOSS) = (bI_{eb} + gI_{eg} - t_1(I_{eb} - I_2)(1 - e^{-b/t_1}) + t_2(I_1 - I_{eg})(1 - e^{-g/t_2}))/(b + g)) \tag{C.6}$$

C.4 Recency Effects on Impairment

The Extended E model also proposes an impairment factor relating to the location of the errors within the call itself. This incorporates the findings that losses early in the call and near the end of a call cause a greater loss in received quality then errors in the middle of the call. However these findings are not of relevance to VoIP over WLANs and hence, for the sake of simplicity, we omit this factor from our calculations.

References

[1] Telecommunications and internet protocol harmonization over networks (tiphon) release 3; end-to-end quality of service in tiphon systems; part 1: General aspects of quality of service (qos). Technical Report TR 101 329-6 V3.1.3, ETSI TIPHON, 2002.

[2] Digging deeper into deep packet inspection (dpi). Technical report, Allot Communications, Woburn, MA, 2007.

[3] Blocking skype through deep packet inspection. In *HICSS '09: Proceedings of the 42nd Hawaii International Conference on System Sciences*, pages 1–5, Washington, DC, USA, 2009. IEEE Computer Society.

[4] Fring voip client `http://www.fring.com/about/about_us/`, 2009.

[5] Skype `http://www.skype.com/`, 2009.

[6] Wifi alliance press release: `http://www.wi-fi.org/pressroom_overview.php?newsid=770`, 2009.

[7] Nabeel Ahmed and Usman Ismail. Designing a high performance wireless testbed for centralized control. In *IEEE TridentCom 2009*, 2009.

[8] Nabeel Ahmed, Usman Ismail, Srinivasan Keshav, and Dina Papagiannaki. Online estimation of rf interference. In *ACM CoNext*, 2008.

[9] F. Anjum, M. Elaoud, D. Famolari, A. Ghosh, R. Vaidyanathan, A. Dutta, P. Agrawal, T. Kodama, and Y. Katsube. Voice performance in wlan networks - an experimental study. *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, 6:3504–3508 vol.6, Dec. 2003.

[10] P. T. Brady. A technique for investigating on-off patterns in speech. In *BSTJ*, volume 44, pages 1–2, 1963.

[11] Lin Cai, Yang Xiao, Xuemin (Sherman) Shen, and Jon W. Mark. Voip over wlan: voice capacity, admission control, qos, and mac: Research articles. *Int. J. Commun. Syst.*, 19(4):491–508, 2006.

[12] A Chan and S.C. Liew. Voip capacity over multiple ieee 802.11 wlans. In *IEEE ICC*, 2007.

[13] Imrich Chlamtac and Yi bing Lin. *Wireless And Mobile Network Architectures.* Wiley, Hoboken, New Jersey, 2001.

[14] R. G. Cole and J. H. Rosenbluth. Voice over ip performance monitoring. *SIGCOMM Comput. Commun. Rev.*, 31(2):9–24, April 2001.

[15] Brian Crow, Indra Widjaja, Jeong Geun Kim, and Prescott Sakai. Investigation of the ieee 802.11 medium access control (mac) sublayer functions. In *INFOCOM '97: Proceedings of the INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution*, page 126, Washington, DC, USA, 1997. IEEE Computer Society.

[16] I. Dangerfield, D. Malone, and D.J. Leith. Experimental evaluation of 802.11e edca for enhanced voice over wlan performance. *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium on*, pages 1–7, April 2006.

[17] ETSI. *Digital cellular telecommunications system (Phase 2+); AT command set for GSM Mobile Equipment (ME)*, 1998.

[18] S. Garg and M. Kappes. Admission control for voip traffic in ieee 802.11 networks. In *IEEE Global Telecommunications Conference.*, volume 6, pages 3514– 3518, Baskin Ridge, NJ, USA, 2003. Avaya Labs.

[19] S. Garg and M. Kappes. Can i add a voip call? In *ICC '03. IEEE International Conference on Communications, 2003*, volume 2, pages 779–783 vol.2, 2003.

[20] S. Garg and M. Kappes. An experimental study of throughput for udp and voip traffic in ieee 802.11b networks. *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, 3:1748–1753 vol.3, March 2003.

[21] Matthew S. Gast. *802.11 Wireless Networks: The definitive guide.* O'Reilly, Sebastopol Callifornia, 2002.

[22] David P. Hole and Fouad A. Tobagi. Capacity of an ieee 802.11b wireless lan supporting voip. In *in Proc. IEEE International Conference on Communications (ICC 2004*, pages 196–201, 2004.

[23] IEEE. *P802.11e/D13.0 (C/LM) Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment: Medium Access Method (MAC) Quality of Service Enhancements*, 2005.

[24] IEEE. *Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific*

requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2007.

[25] ITU-T. *G.711: Pulse Code Modulation (PCM) of Voice Frequencies*, 1989.

[26] ITU-T. G.726: 40, 32, 24, 16 kbit/s adaptive differential pulse code modulation (adpcm). Technical report, ITU-T, 1990.

[27] ITU-T. *G.113: General Characteristics of International Telephone Connections and International Telephone Circuits*, 1993.

[28] ITU-T. *P.800: Methods for Subjective determination of Transmission Quality*, 1996.

[29] ITU-T. *G.107: The E-model, a computational model for use in transmission planning*, 2005.

[30] ITU-T. *G.729: Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP)*, 2007.

[31] Yeonsik Jeong, Eep Kakumanu, Cheng lin Tsao, and Raghupathy Sivakumar. Improving voip call capacity over ieee 802.11 networks. In *Broadnets*, 2007.

[32] Wenyu Jiang, K. Koguchi, and H. Schulzrinne. Qos evaluation of voip end-points. *Communications, 2003. ICC '03. IEEE International Conference on*, 3:1917–1921 vol.3, May 2003.

[33] T. Kawata, S. Shin, A.G. Forte, and H. Schulzrinne. Using dynamic pcf to improve the capacity for voip traffic in ieee 802.11 networks. *Wireless Communications and Networking Conference, 2005 IEEE*, 3:1589–1595 Vol. 3, March 2005.

[34] Athina P. Markopoulou, Fouad A. Tobagi, and Mansour J. Karam. Assessment of voip quality over internet backbones. In *IEEE Infocom*, pages 150–159, 2002.

[35] Rohan Murty, Jitendra Padhye, Ranveer Chandra, Alec Wolman, and Brian Zill. Designing high performance enterprise wi-fi networks. In *NSDI'08: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, pages 73–88, Berkeley, CA, USA, 2008. USENIX Association.

[36] M. Narbutt and M. Davis. Gauging voip call quality from 802.11 wlan resource usage. *World of Wireless, Mobile and Multimedia Networks, 2006. WoWMoM 2006. International Symposium on a*, pages 8 pp.–, June 2006.

[37] Eun Kyoung Paik and Yanghee Choi. Prediction-based fast handoff for mobile wlans. In *ICT 2003. 10th International Conference on Telecommunications, 2003.*, volume 1, pages 748– 753, 2003.

[38] N. Ramos, D. Panigrahi, and S. Dey. Quality of service provisioning in 802.11e networks: challenges, approaches, and future directions. *Network, IEEE*, 19(4):14–20, July-Aug. 2005.

[39] D. Raychaudhuri, M. Ott, and I. Secker. Orbit radio grid tested for evaluation of next-generation wireless network protocols. In *TRIDENTCOM '05: Proceedings of the First International Conference on Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities*, pages 308–309, Washington, DC, USA, 2005. IEEE Computer Society.

[40] Minho Shin, Arunesh Mishra, and William A. Arbaugh. Improving the latency of 802.11 hand-offs using neighbor graphs. In *MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 70–83, New York, NY, USA, 2004. ACM.

[41] Sangho Shin, Andrea G. Forte, and Henning Schulzrinne. Experimental measurement of the capacity for voip traffic in ieee 802.11 wireless networks. *INFOCOMM*, 2007.

[42] N. Smavatkul, Ye Chen, and S. Emeott. Voice capacity evaluation of ieee 802.11a with automatic rate selection. *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, 1:518–522 Vol.1, Dec. 2003.

[43] Kyoungwon Suh, Daniel R. Figueiredo, Jim Kurose, and Don Towsley. Characterizing and detecting skype-relayed traffic. *IEEE Infocom (Infocom 2006)*, 2006.

[44] France Telecom. *ITU-T SG12 D.139: Study of the relationship between instantaneous and overall subjective speech quality for time-varying quality speech sequences: influence of a recency effect*, 2000.

[45] Peter Thornycroft. The all-wireless workplace is now open for business: Using 802.11n as your primary network lan deployments. Technical report, Aruba Networks, 2007.

[46] ETSI TIPHON. *TS 101 329-5 QoS Measurement Methodologies*, 2001.

[47] M. Veeraraghavan, N. Cocker, and T. Moors. Support of voice services in ieee 802.11 wireless lans. *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 1:488–497 vol.1, 2001.

[48] Patrick Verkaik, Yuvraj Agarwal, Rajesh Gupta, and Alex C. Snoeren. Softspeak: Making voip play fair in existing 802.11 deployments. In *6th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2009.

[49] S. Wang, A. Sekey, and A. Gersho. An objective measure for predicting subjective quality of speech coders. *Selected Areas in Communications, IEEE Journal on*, 10(5):819–829, Jun 1992.

[50] Wei Wang, Soung Chang Liew, and V.O.K. Li. Solutions to performance problems in voip over a 802.11 wireless lan. *Vehicular Technology, IEEE Transactions on*, 54(1):366–384, Jan. 2005.

[51] J.Y. Yeh and C. Chen. Support of multimedia services with the ieee 802-11 mac protocol, in: Ieee icc (2002). In *ICC*, 2002.

[52] Jeonggyun Yu, Sunghyun Choi, and Jaehwan Lee. Enhancement of volp over ieee 802.11 wlan via dual queue strategy. *Communications, 2004 IEEE International Conference on*, 6:3706–3711 Vol.6, June 2004.

[53] Yuan Yuan, Daqing Gu, W. Arbaugh, and Jinyun Zhang. High-performance mac for high-capacity wireless lans. *Proceedings. 13th International Conference on Computer Communications and Networks (ICCCN)*, pages 167–172, Oct. 2004.

[54] Maria C. Yuang, Po L. Tien, and W. J. Hung. Intelligent voice smoother for silence-suppressed voice over internet: Theory and practice. *IEEE Journal on Selected Area in Communications*, 17:29–41, 1997.