Call Admission Control for Adaptive Bit-rate VoIP over 802.11 WLAN

CUI, Yuanyuan

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Master of Philosophy

in

Information Engineering

The Chinese University of Hong Kong

Aug 2009

摘要

在 IP 網路語音傳輸﹝VoIP 或網絡電話，下稱 VoIP﹞的實際應用中，許多 VoIP 工具， 比如 Skype，Google Talk，MSN， 已經開始使用「自适应變速率」 的語音編碼技術。不同于以往固定速率網路電話﹝Constant bit-rate VoIP，下稱 CVoIP﹞，自适应變速率網路電話﹝Adaptive bit-rate VoIP，下稱 AVoIP﹞可以根據網絡狀況靈活的調整其數據傳送速率。早前研究表明，對于 CVoIP 而言，在保證通話質量的前提下，一個 IEEE 802.11 無線區域網絡﹝下簡稱無線區域網絡﹞可以容納的通話數目是固定的。但是對于 AVoIP，我們發現，由于每個通話的數據傳送速率都是可變的，很難界定一個無線區域網絡究竟可以同時容納多少路通話。隨之而來的一個問題是：如何對接入無線区域網絡的 AVoIP 通話進行呼叫允許控制。這篇論文針對這個問題進行了深入研究。

首先，我們做了大量的實驗來推導 Skype 的自适应變速率通話機制。根據推導結果，我們在 Network Simulator﹝一個網絡模擬仿真軟件﹞中開發了一個 Skype 模擬流量生成器﹝Skype-emulating Traffic Generator，下稱 STG﹞。STG 模擬了 Skype 的流量特征，并且可以像 Skype 一樣針對不同的網絡狀況調整其流量速率。之后，借助 STG，我們對一定數目的 AVoIP 通話共存于一個無線區域網絡的場景進行了仿真。仿真結果表明： 如果單純通過限制通話接入網絡的數目來進行呼叫允許控制，整個系統會陷入一個高度不公平，不穩定的狀態：各路 AVoIP 通話不能公平的分享無線網絡帶寬，并且通話語音質量非常不穩定。爲了解決這兩個問題，我們設計了一個由訪問點﹝Access Point﹞實施的，對無線區域網絡中所有通話進行集中式控制的方案：CFSC﹝Call admission, Fairness,

and Stability Control, 下稱 STG ）。CFSC 融合了呼叫允許控制，公平控制，以及穩定控制三個控制功能。仿真結果表明， CFSC 可以使系統達到一個公平穩定的狀態，同時確保各路 AVoIP 通話的語音質量。

Abstract

Adaptive bit-rate VoIP (AVoIP) has been widely adopted by many VoIP applications, including Skype, Google Talk, and MSN. Unlike Constant bit-rate VoIP (CVoIP) which employs fixed bit-rate speech codecs, AVoIP adopts variable bit-rate speech codecs. For CVoIP, the number of voice sessions that can be supported by an IEEE 802.11 WLAN is fixed. For AVoIP, however, since its bit rate can be adjusted according to the network condition, the number of voice sessions admitted into a WLAN can be flexibly adjusted. An issue is how one would perform call admission control when AVoIP is used.

This thesis approaches the problem in the following way. First, we conduct extensive experiments to deduce the rate adaptation mechanism of Skype, arguably the most popular AVoIP application. We then implement a Skype-emulating Traffic Generator (STG) module in Network Simulator (NS2) that incorporates the mechanism. Simulation results verify that STG can produce Skype-like traffic and react to network congestion in the same way as Skype does. We then demonstrate with STG that traditional call admission control that simply limits the number of admitted calls will cause the system to settle at an operating region where there is a high degree of unfairness and instability among the AVoIP sessions. To solve the problem, we design a comprehensive scheme, which integrates the function of call admission, fairness, and stability control (CFSC), for AVoIP over 802.11 WLAN. CFSC makes use of AVoIP's adaptiveness to packet loss to fairly assign bandwidth to AVoIP sessions. Simulation results show that this scheme can provide fair and consistently good voice quality for all AVoIP sessions.

Acknowledgement

Foremost, I would like to express my deepest gratitude to my supervisor Prof. Liew Soung Chang, who whole-heartedly provided me guidance and invaluable advice throughout the duration of my research. It was under his supervision I chose this topic and completed the thesis. His patience, motivation, enthusiasm, and immense knowledge have stimulated my learning. I could not have imagined having a better advisor and mentor for my master study.

Besides my supervisor, I would like to thank my dear colleagues. During the past two years, they provided me with earnest help and friendship. They also offered me lots of valuable ideas and comments about this thesis, especially MR. Lin Yangyang, Ms. Fu Liqun, Ms. Kai Caihong and Mr. Zhang Shengli.

Last but not least, I am deeply indebted to my parents for their support throughout my life.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Motivations and Contributions

Previous works on VoIP over WLAN have mostly assumed the use of Constant bit-rate VoIP (CVoIP) codecs [1-11]. In practice, however, Adaptive bit-rate VoIP (AVoIP) has taken the place of CVoIP in many VoIP applications, including Skype, Google Talk, and MSN [12-13]. Unlike CVoIP, AVoIP can adjust its bit rate in response to the network bandwidth variations [14-18]. Specifically, when the bandwidth is abundant, AVoIP works at a higher rate to achieve better voice quality; when the bandwidth is scarce, AVoIP lowers its flow rate accordingly. Reducing rate at the source allows more graceful degradation of voice quality compared with letting the network drop its packets arbitrarily under congestion.

The "capacity" of VoIP over WLAN is defined as the maximum number of bidirectional voice sessions that can be supported in a WLAN, subject to a minimum voice quality requirement [5]. For CVoIP, the "capacity" is a fixed number. (e.g. the capacity is 12 when the codec used is GSM 6.10 and the WLAN is 802.11b). Once the capacity is exceeded, the quality of every voice session in the WLAN degrades badly. Therefore, one can simply limit the number of voice sessions admitted into the WLAN to perform call admission control. For AVoIP, however, the "capacity" is not as clear cut. Even when the bandwidth of the WLAN has already been fully utilized, it is still possible to add a new session, since the existing AVoIP sessions could reduce their rates at the sources to make room for the new session. Consequently, how to perform call admission control for AVoIP over WLAN becomes not trivial anymore and needs careful rethink.

This thesis is dedicated to the study of the interaction between the adaptation process at the voice source and the dynamic of the network condition. There are two major contributions:

1. To facilitate our studies, we have built a voice traffic generator that can reproduce AVoIP traffic in NS2. We choose to emulate Skype because it is the most popular VoIP application. Also, the adaptation process of Skype is quite generic and is representative of many other AVoIP applications. Besides our own studies, our traffic generator and the behaviors we unearth about Skype might be useful to the studies by other researchers.
2. We find that traditional call admission control that does not take into account the intricacies of the adaptation process at the source may lead to unfair and unstable performance among the voice sessions. To tackle these problems, we design a scheme that integrates the function of call admission, fairness, and stability control (CFSC). Simulation results show that our CFSC scheme can provide good, fair, and consistent voice quality for every user.

A primary assumption held in this work is that all VoIP sessions in the WLAN are constrained by the same bottleneck - the local WLAN. In practice, different sessions might have different bottlenecks locating at different places. In this thesis, we also explore some methods to perform call admission control without this assumption.

1.2 Related Works

Much work [4-11] has been devoted to the study of VoIP capacity of IEEE 802.11 WLAN. References [6-7] gave an analytical model of studying the capacity of a general wireless network. References [8-9] derived several methods to calculate the capacity for common voice codecs. References [9-11] proposed various schemes to boost the capacity. In this thesis, we continue to use the calculation method in our prior work [9] to obtain the capacity.

As popular but closed-source software, Skype has attracted much reverse-engineering interest. In [19] and [20], Skype's P2P internet protocol was analyzed. In [21] and [22], methods and rules to classify Skype traffic were proposed. Security problems of Skype were also well studied in [23-24]. Our work is most closely related to [25] and [26]. Reference [25] studied the performance of Skype when packet loss occurs due to network congestion. References [26] confirmed that Skype has a rate adaptation mechanism. It also claimed that Skype traffic is not TCP-friendly and not self-friendly. Neither [25] nor [26], however, paid much attention to the characteristics of Skype traffic, and the details of the rate adaptation mechanism are lacking. This thesis, in contrast, attempts to carry out a thorough analysis on the rate adaptation mechanism of Skype, especially under its newly adopted SVOPC codec [27].

Call admission control over WLAN studied in the past can be broadly classified into two types: user-number based [9-11] and residual-bandwidth based [28-30]. References [9-11] used a computed capacity as the criterion to constrain the number of sessions admitted. References [28-30] proposed to admit a new session only if the residual WLAN bandwidth monitored by the Access Point (AP) is sufficient to accommodate it. Residual-bandwidth based approach is not compatible with AVoIP: since existing voice sessions can reduce their flow rates to accommodate the new session, it is possible to add a new session even when the residual bandwidth is zero. The user-number based approach is also not appropriate for AVoIP, because the capacity in terms of an absolute maximum number of admissible calls is not clear cut for AVoIP. In this thesis, we show that unfairness and instability problems can occur under the user-number based approach. We show that our proposed scheme, CFSC, can solve these problems in a comprehensive manner by taking into account the intricacies of the adaptation processes at the sources.

1.3 Organization of the Thesis

The reminder of this thesis is organized as follows. First, we give an overview of the background of VoIP over IEEE 802.11 WLAN in Chapter 2. Chapter 3 then describes the Skype rate adaptation mechanism we uncover from our experiments. It also presents our Skype-emulating Traffic Generator and validates it under different scenarios. Using the generator, we then demonstrate the unfairness and instability problems under the traditional user-number based call admission control in Chapter 4. Chapter 4 also presents our proposed CFSC scheme, which integrates the function of call admission control, fairness control, and stability control. Chapter 5 evaluates CFSC with detailed simulation results. Chapter 6 concludes the thesis.

Chapter 2
Background

In this chapter, we briefly introduce the IEEE 802.11 standards and VoIP techniques. Particularly, we put forth the appealing features of Adaptive bit-rate VoIP. Then we offer an overview of VoIP over WLAN. We also introduce Skype, which is the object of our study on rate adaptation mechanism of AVoIP applications.

2.1 IEEE 802.11

2.1.1 IEEE 802.11 Topologies

The most basic component of an 802.11 WLAN is the station [31]. A station could be any device that contains the functionality of the 802.11 protocol. For example, it could be a laptop PC, a handheld device, or an Access Point.

A set of stations that could communicate with the others compose a Basic Service Set (BSS). The BSS is the basic building block of an 802.11 wireless networks. There are two types of BSS topology supported in IEEE 802.11 standard, the Independent Basic Service Set (IBSS), which is also referred as Ad-hoc network, and the Infrastructure Basic Service Set.

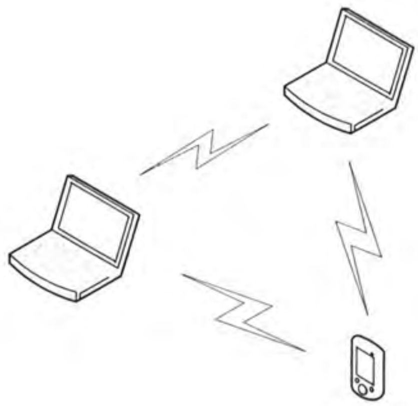


Figure 2.1: An Independent Basic Service Set (Ad hoc network)

Figure 2.1 depicts an independent BSS (IBSS). Stations in an IBSS communicate with each other directly. Since there are no relay functions in an IBSS, all stations need to be within the range of the others. Typically, IBSS is composed of a small number of stations set up for a specific purpose and for a short period of time. A common practice of using IBSS is to create a short-lived network to support a single meeting in a conference room.

Figure 2.2 depicts an Infrastructure BSS. Different from IBSS, an Infrastructure

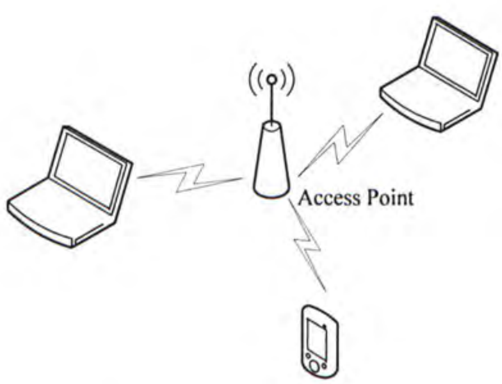


Figure 2.2: An Infrastructure Basic Service Set

BSS has a special station called Access Point (AP). The access point provides a local relay function for the BSS. All communications, including the communications between ordinary stations in the same BSS, must be relayed through the AP. Therefore, the coverage of the Infrastructure BSS is effectively doubled by the local relay function, and is decided by the transmission range of the AP. In an Infrastructure BSS, ordinary stations must actively associate with the AP to obtain network services. Upon receiving an association request, AP may choose to grant or deny access. There is no limit on the number of ordinary stations that an AP may serve in the 802.11 standard.

By chaining BSSs together with a backbone network, an Extended Service Set (ESS) is created and larger network coverage can be achieved. Figure 2.3 depicts an ESS. In an ESS, the APs communicate among themselves to forward traffic from one BSS to another. The AP also performs the wireless-to-wired bridging function. It converts frames on an 802.11 network to another type of frame for delivery to the rest of the world.

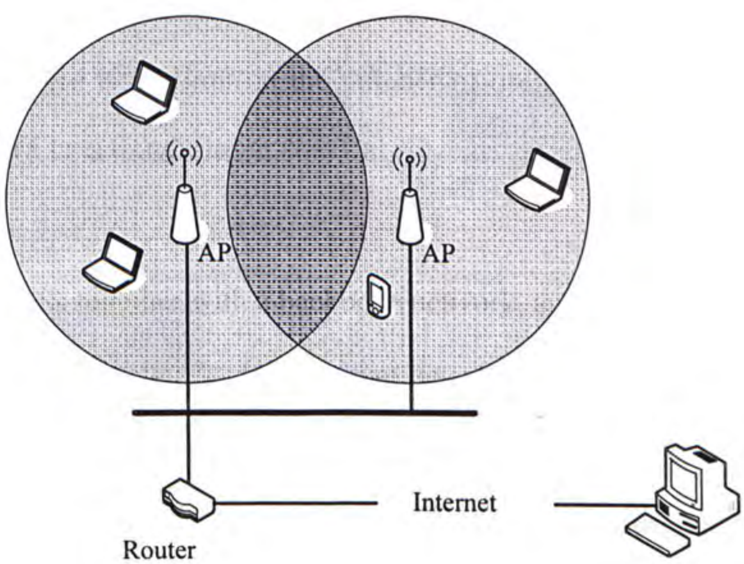


Figure 2.3: An Extended Service Set

2.1.2 IEEE 802.11 MAC

Access to the wireless medium is controlled by coordination functions. The IEEE 802.11 MAC protocol supports two modes of operation, the Distributed Coordination Function (DCF) and the Point Coordination Function (PCF) [32].

The DCF is basically a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) access mechanism. It provides a standard Ethernet-like contention-based service where multiple independent stations interact without central control. Therefore, DCF can be used in either IBSS networks or in Infrastructure BSS networks. There are two transmission modes in DCF, Basic Mode and RTS/CST Mode. In Basic Mode, a station wishing to transmit senses the medium. If the medium is busy then it defers. If the medium is free for a period of time specified by Distributed Inter Frame Space (DIFS), then the station transmits. In order to avoid collisions among stations, DCF also specifies random backoff, which forces a station to wait for a number of Slots before accessing the medium. This backoff number is randomly chosen between 0 and CW-1. Once the target station receives the packet correctly, it responds with an Acknowledgement after a Short Inter Frame Space (SIFS). Figure 2.4 presents the schematic of the Basic Mode of IEEE 802.11

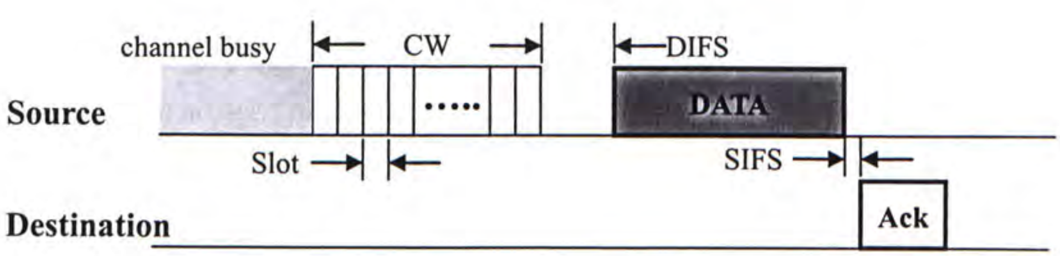


Figure 2.4: Basic Mode of IEEE 802.11 DCF

DCF. Another transmission mode, RTS/CTS Mode, is designed to reduce collisions resulting from hidden nodes. Before transmitting data packet, a station sends an RTS (Ready-to-Send) packet first. The RTS packet has a smaller size than the common data packet. It reserves the radio link for the following data transmission by silencing any stations (except the target station) that hear it. If the target station receives an RTS, it responds a CTS (Clear-to-Send) packet. The CTS packet silences stations in the immediate vicinity. Once the RTS/CTS exchange is complete, the data transmission can proceed. Figure 2.5 presents the schematic of the RTS/CTS Mode. NAV stands for Network Allocation Vector, all stations receiving either the RTS and/or the CTS will not try to access the medium for a certain amount of time indicated by the NAV.

PCF provides contention-free services. It relies on a central node, often an AP, to communicate with other stations and to check whether the airtime is free. Therefore the PCF is restricted to Infrastructure BSS networks. PCF is designed to support applications that require real-time services. It could offer a sort of "packet-switched

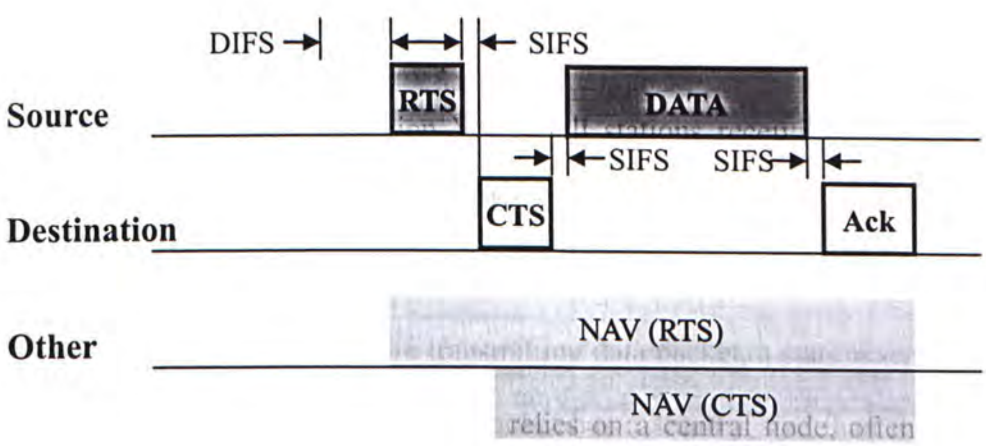


Figure 2.5: RTS/CTS Mode of IEEE 802.11 DCF

connection-oriented" service, thus allow the network to provide delay guarantees necessary for real-time applications like interactive voice [33]. However, in practice, PCF is an option not supported in most commercial products. Most work on VoIP over WLAN assumes the use of DCF, and only some consider PCF.

In this thesis, we assume the use of DCF. Of the two modes of DCF, VoIP over WLAN applications commonly use the Basic Mode (no RTS/CTS handshake). This is because the overhead introduced by RTS/CTS packet is significant when compared with the small size of the voice data packet. Table 2.1 listed the values of the 802.11b [34] DCF parameters that been adopted in both analysis and simulations in this thesis. Since VoIP application is sensitive to delay, the retry limit is 3.

Table 2.1: Parameter Values of 802.11b DCF

DIFS	50 μsec
SIFS	10 μsec
Slot Time	20 μsec
CW_{min}	32
CW_{max}	1023
Retry Limit	3
Date Rate	1,11Mbps
Basic Rate	2Mbps
PHY header *	192 μsec
MAC header	34Bytes
ACK *	248 μsec

* PHY header is transmitted at 1 Mbps, ACK shown above is actually ACK frame + PHY header. The ACK frame is 14 bytes and is transmitted at basic rate, 2 Mbps, regardless of date rate.

2.2 Voice over Internet Protocol (VoIP)

2.2.1 A VoIP system

Figure 2.6 shows a typical VoIP system [35]. At the sender side, audio signal is digitalized and compressed into a low-rate frame stream by speech codecs. Then one or more speech frames might be aggregated into one packet before being sent out. When traveling through the Internet, packets suffer different transmission delay depending on the current network condition and the routing path, and might get lost due to network congestion or transmission errors (often in wireless scenario). At the receiver side, packets are buffered and rescheduled to provide a smooth playout. Any packet arriving after its scheduled playout time is simply discarded and regarded as lost. Consequently, losses as seen by the application may actually consist of two parts, the real losses at the network and the deliberate losses at the receiver. After the playout buffer, the speech frames are finally decoded and the digital signal is transformed into an acoustic signal.

2.2.2 QoS requirements for VoIP

As a real-time application, VoIP has stringent requirements in delay and jitter. According to [36-37], the one way transmission delay for VoIP should be

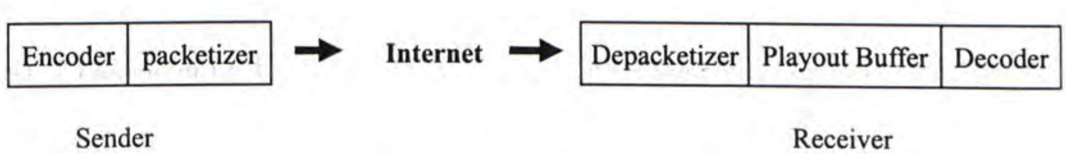


Figure 2.6: VoIP system

preferably less than 150ms, and must be less than 400ms. On the other hand, VoIP is not very sensitive to packet loss. A loss of 3% is acceptable for real-time audio with rate 4kbps to 64kbps. Since delayed packets are not tolerable, retransmission of lost packets is not useful. Therefore, VoIP applications commonly use UDP as transport layer protocol.

In our study, we assume that the tolerable maximum packet loss is 3%, and the end-to-end delay bound is 400ms. An arrived packet with delay that beyond the bound is simply regarded as lost. For a VoIP session with two directional flows, once the overall packet losses of either flow exceed 3%, we say the session fails to meet the QoS requirement. Notice the overall packet losses consist of the packet losses due to network congestion and the packet losses due to unacceptable delay.

2.2.3 VoIP speech codecs

An important component that influences the quality of VoIP is speech codec. There are various standardized speech codecs, such as G.711, G.729, iLBC, iSAC. Different codec has different coding rate (bits/s), frame rate (Hz), algorithmic latency (ms), complexity and speech quality (MOS). Generally, a higher coding rate usually gives a better speech quality. On the other hand, if the packet rate is set low and the packetization time is high, the overall transmission delay increases and conversation call quality is harmed. Each speech codec implements a different trade-off between output speech quality, algorithmic delay, bit rate, computational complexity and robustness to background noise. The optimal codec for VoIP depends on the specific scenario.

VoIP speech codec can be divided into two types, the Constant bit-rate VoIP (CVoIP) speech codec (e.g. G.711, G.729 and iLBC) and the Adaptive bit-rate VoIP (AVoIP) speech codec (e.g. iSAC). In the past, CVoIP codecs were usually employed. Such codecs generate a constant output bit-rate, independently of the network conditions. If the network cannot sustain the traffic, the packets will be delayed or dropped. In the recent years, AVoIP codecs are taking place of CVoIP ones gradually. Based on our observation, many popular VoIP applications have adopted AVoIP codecs, such as GoogleTalk, MSN and Skype. AVoIP codec can adapt its rate to the current condition of the network. If the network is congested, speech is coded at lower bit rates; if the network, instead, is lightly loaded, the speech codec is allowed to operate at higher bit rates. In other words, it could always generate only the traffic that the network is capable of carrying with a given quality at any instant in time.

AVoIP has several appealing features. Firstly, by adapting gracefully to network congestion, AVoIP could provide relatively better speech quality than CVoIP. Secondly, AVoIP does not require a rigid partitioning of the link bandwidth. It can exploit statistical rather than deterministic QoS guarantees. As such, AVoIP uses the bandwidth with higher efficiency. Finally, AVoIP provides operators more flexibility. By trading off perceived QoS and monetary cost, VoIP operators can offer different levels of services. For example, fixed-high-bit-rate calls, fixed-low-bit-rate calls, adaptive-high-bit-rate calls, and adaptive-low-bit-rate calls.

2.3 VoIP over WLAN

Unlike traditional data traffic, which still takes up the majority of bandwidth in IP networks, the sending rate of VoIP session is usually small (less than 130kbps for a

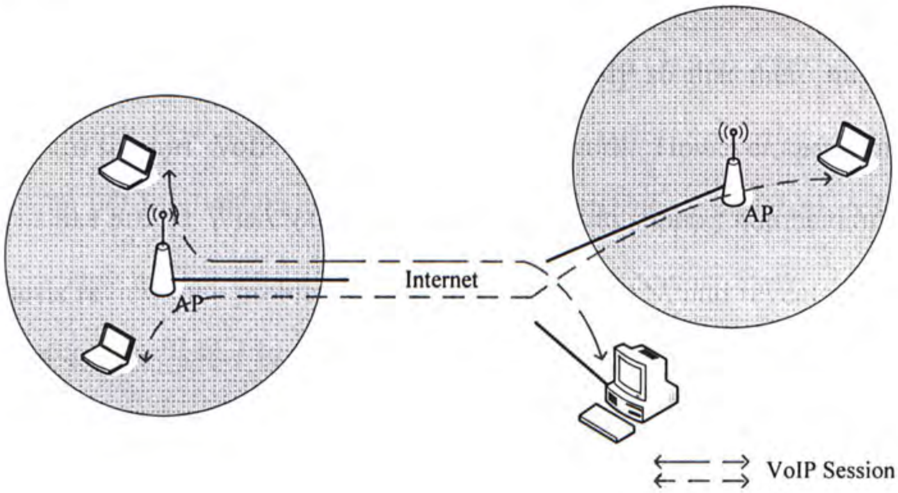


Figure 2.7: System architecture of VoIP over a single-cell WLAN to backbone network

bidirectional VoIP session) and relatively constant over the duration of a call. Available bandwidth is often plenty enough for VoIP so that there might not be a problem to support VoIP traffic in wired network. However, existing wireless networks like 802.11 WLAN might face challenges to support numbers of VoIP users. This thesis reviews the problems for CVoIP over WLAN, and reveals new problems for AVoIP over WLAN.

2.3.1 System Architecture of VoIP over WLAN

The scenario we focus on in this thesis is a "single cell" WLAN, which is an Infrastructure BSS with one AP in the center. Since the coverage of a WLAN is commonly less than 300 meters, we assume users seldom call their neighbors within the same WLAN. In other words, we assume a VoIP session is between a user in the WLAN and a user in the backbone network. All voice packets to or from the WLAN must be relayed by the AP. Figure 2.7 shows the system architecture of VoIP over a single-cell WLAN to the backbone network.

2.3.2 VoIP Capacity over WLAN

The VoIP capacity over WLAN is defined as the maximum number of VoIP sessions a WLAN can support with satisfactory user perceived quality. Using a simple method, the capacity can be easily calculated for common voice codecs [9]. It is well known that the capacity for VoIP over WLAN is severely limited and quite small. It is mainly due to the various inherent header (e.g. IP/UDP/RTP header) and protocol overheads (e.g. backoff countdown time, Physical Header of 802.11 packet, etc.).

Prior works mainly focused on the CVoIP capacity over WLAN and proposed various schemes to boost the capacity. This thesis, however, investigates the AVoIP capacity over WLAN. We focus on AVoIP because of its appealing features and its widely adoption in reality.

A VoIP session has two directional flows, the uplink flow from the wireless station to its AP and the downlink flow from the AP to its wireless station. There exists unbalance problem between the uplink flows and the downlink flows [8]. When the wireless network is saturated, uplink flows always suffer severer packet loss than uplink flows. It is might because that the AP transmits a larger load than the stations and its buffer gets overflow more quickly. In this thesis, to eliminate the influence of unbalance problem and concentrate on issues introduced by AVoIP, we assume that AP has separated buffers for each downlink flows. Another assumption we hold is that all VoIP sessions have the same bottleneck and the bottleneck is the WLAN. In Appendix 2, we also discuss the case without the bottleneck assumption.

2.4 Skype

Skype is beyond doubt the most prominent example of applications providing VoIP calls. Until 2009, the number of Skype registered users over the world has reached 443 million, and the average number of Skype online users everyday hits 17 million [38]. People use Skype mainly for the high quality of voice it offers and especially for its free PC to PC service.

Skype uses a proprietary and closed-source VoIP protocol. The main difference between Skype and other VoIP protocols like SIP, H.323 is that Skype operates on a peer-to-peer model rather than the more usual client-server model. An array of different audio codecs including G.729 and SVOPC are adopted in Skype [39]. SVOPC (Sinusoidal Voice Over Packet Coder) is an adaptive voice codec developed by Skype itself. G.729 is mainly used for Skypeout (PC-to-PSTN) calls while SVOPC is mainly used for PC-to-PC calls.

In this thesis, we limit our study to Skype PC-to-PC calls using SVOPC codec. We show that Skype can adapt to the network condition and we conduct extensive experiments to investigate how Skype manages to do so. The Skype rate adaptation mechanism we conjectured provides important information for us to develop AVoIP traffic generator and to continue our study on AVoIP over WLAN.

Chapter 3
Skype Rate Adaptation Mechanism

In this chapter, we investigate the Skype rate adaptation mechanism with the goal of building an AVoIP traffic generator. To achieve this goal, we have performed extensive experiments in a controlled environment.

3.1 Experimental Setting

Our experimental testbed consists of two PCs connected by a router, as shown in Figure 3.1. Each PC is installed with Skype V.3.8.0.139. Three tools are used in our experiments: i) Wireshark, a protocol analyzer for collecting packet traces; ii) Bandwidth Controller for adjusting link capacity; and iii) Tone Generator

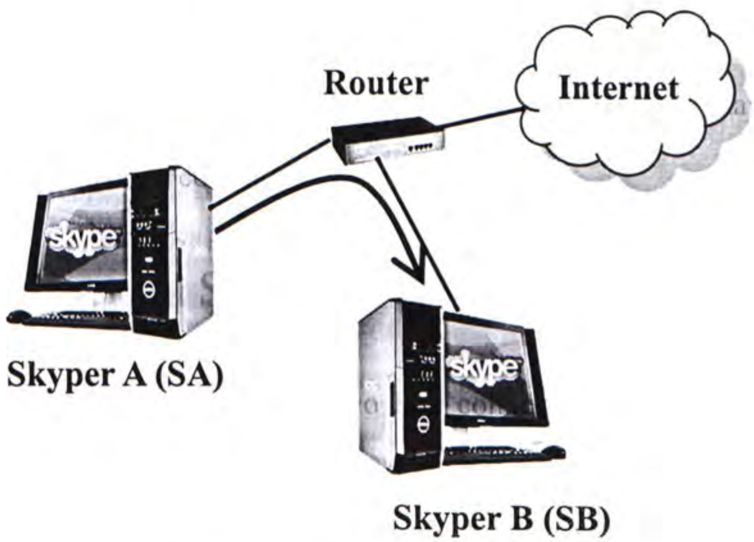


Figure 3.1: Experimental testbed

for generating test tones.

The general measurement setup is the following: Skyper A (SA) sends audio signal to Skyper B (SB). During the voice communication, we vary the bandwidth between SA and the router using Bandwidth Controller, and collect two kinds of information at both SA and SB: i) packet traces captured by Wireshark and ii) data shown on the Call Technical Information Display (CTID) of Skype.

CTID is a utility of Skype [40], as shown in Figure 3.2. It provides statistics of ongoing Skype calls, such as jitter, round-trip time, and packet loss. In addition, there are other measurements and parameters whose official definitions cannot be found. In our experiments, we look at two such parameters: i) BM (Bps) - we interpret it to be the bandwidth usage target; and ii) a number measured in milliseconds - we interpret it to be the time gap between two successive voice

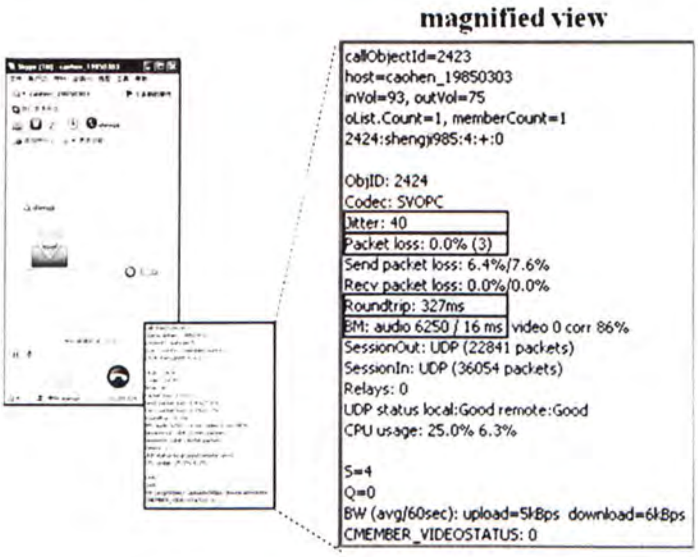


Figure 3.2: Skype Call Technical Information Display

frames generated by SVOPC and name it frame interval Δt. In general, a packet may consist of one or more voice frames.

All our results and analysis in the following are based on the packet trace files collected by ourselves and the information shown on CTID.

3.2 Overview

Figure 3.3 provides the schematic of our "conjectured" Skype rate adaptation mechanism. A node generates a receiver report (RR) to, and receives a receiver report from, the node at the other end of the voice communication. The received RR is used to set a bandwidth usage target (BM), which is fed to a rate controller for the voice codec and packetizer. According to the BM, Skype adapts its flow rate. In general, a Skype receiver continuously monitors the statistics of the received packets and packs the information in the RR sent to its counterpart.

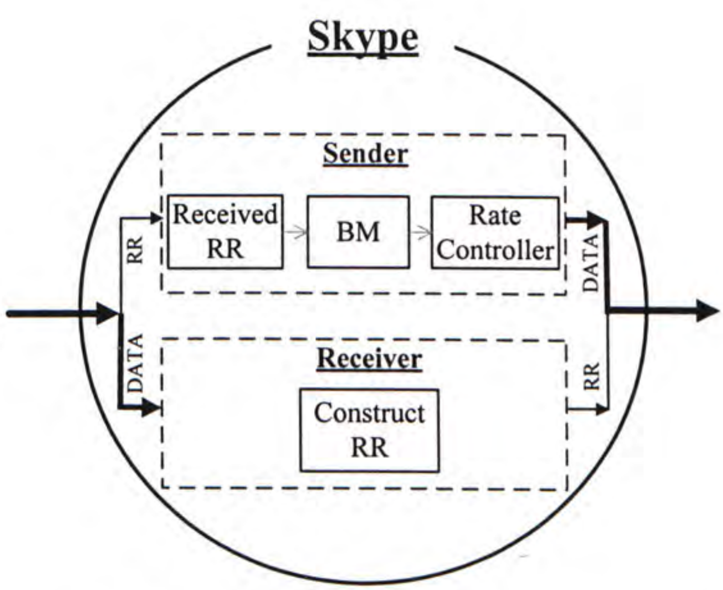


Figure 3.3: Schematic of the Skype rate adaptation mechanism

In the following subsections, we will give full descriptions of the flow rate region, RR, and BM, along with supporting evidence and analysis.

3.3 Flow Rate Region

We have conducted experiments on the testbed shown in Figure 3.1 to investigate the "feasible" flow rates of Skype. In our experiments, we first decrease the link bandwidth between SA and the router, and then increase it back. The variation range is large enough to cover all possible flow rates of Skype. We vary the bandwidth in two manners: 1) step-by-step and 2) sharply, in order to study the dynamics of transience. From the trace files and the information shown on CTID, we summarize the characteristics of the traffic from SA and the traffic from SB in Table 3.1.

As shown in Table 3.1, Skype traffic can be categorized into four levels according to the frame interval Δt. For all levels, the packets of SA have significantly larger size P and smaller packet interval ΔT than those of SB. Since SA has voice input but not SB in our experiments, we believe that Skype can distinguish silence and speech, and treat them differently.

A close look of Table 3.1 reveals two interesting aspects of the rate adaptation.

1. Generally, a higher level, such as L1, has smaller ΔT and larger R. This implies smaller mouth-to-ear delay and larger voice coding rate, thus better voice quality.
2. Although levels can be distinguished without ambiguity according to Δt, the flow rates of adjacent levels may overlap. This suggests that there might be a grey region of flow rates between adjacent levels. We believe this is related to bandwidth probing and we will further discuss this in Section 3.5.

Table 3.1: Characteristics of packets of SA and SB

		SA						SB			
L	Δt (ms)	ΔT (ms)	α	P_{min} (Byte)	P_{max} (Byte)	R_{min} (Bps)	R_{max} (Bps)	ΔT (ms)	α	P (Byte)	R (Bps)
1	**16**	16	1	64	81	4000	5027	48	3	16	333
2	**32**	32	1	95	168	2812	5250	96	3	23	240
3	**48**	48	1	57	215	1187	4480	144	3	30	208
4	**64**	64	1	73	173	1140	2703	128	2	37	289

Our experiments also reveal that transitions can only occur between adjacent levels, even when the bandwidth changes dramatically. The aim might be to avoid over-correction of flow rate to ensure relatively smooth voice quality.

Another trivial finding from our experiments is that a 3-byte packet is generated every 20 seconds from both ends. References [41-42] believed that the 3-byte packet is quality feedback. However, we observed that Skype reacted to network condition changes in no more than 2s. This reaction time is much shorter than 20 seconds. Therefore, feedback is impossible to be these 3-byte packets. In our study we found the real one who plays the role.

3.4 Feedback: Receiver Report (RR)

From the experiments in Section 3.3, we find that Skype piggybacks certain 4-byte information on some packets for the purpose of rate adaption. Figure 3.4 depicts the packet traces of SA and SB. As we can see, packets that are four bytes larger than other packets appear periodically, and SA only adjusts its flow rate (by changing packet size) after receiving such packets from SB. Therefore, we believe the 4-byte extra information plays the feedback role, and we call it Receiver Report (RR).

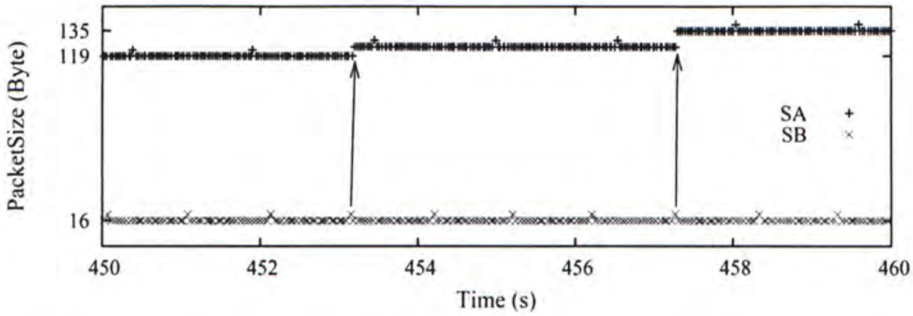
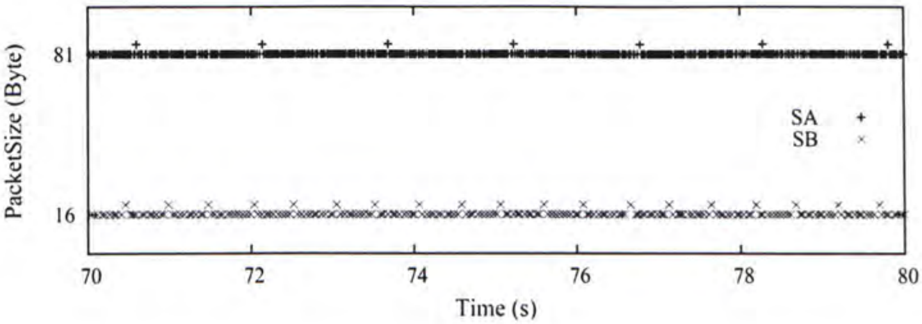
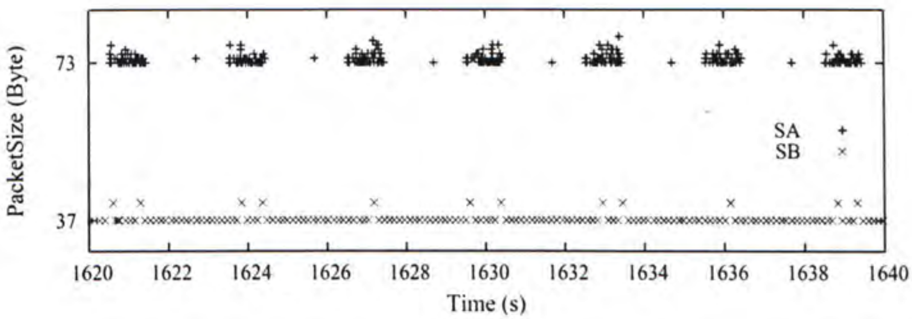


Figure 3.4: Packet traces of SA and SB showing that Receiver Report does exist and can trigger rate adaptation.

We now investigate the generation rule of RR. Figure 3.5 (a) shows the packet traces of SA and SB under a scenario where bandwidth is sufficient to carry L1 Skype traffic. All packets are received successfully. By inspection, packets with RR are generated periodically. We also see that SA sends packets at a rate three times the rate of SB, and SB replies RR at a rate three times the rate of SA. However, the generation of RR is always periodic and the frequency is not always proportional to the received packet rate. Figure 3.5 (b) presents the packet traces of SA and SB under a scenario where bandwidth is not sufficient to even carry L4 Skype traffic. As we can see, RR is not generated periodically, and the time intervals between two successive RRs can be 0.5s and 2.5s. After digging into the trace files of both scenarios, we find the following identical phenomena: i) for a new RR to be generated, there must be at least 30 received packets since the last RR; ii) the interval between two successive RR can only be a multiple of 0.5s. Therefore, we deduce that the rule of RR generation is as follows: every 0.5s, Skype checks whether it has received 30 or more packets. It piggybacks an RR if yes; otherwise it waits another 0.5s.



(a)



(b)

Figure 3.5: Packet traces of SA and SB under different scenarios: (a) bandwidth is sufficient to support L1 traffic; (b) bandwidth is not sufficient even to support L4 Skype traffic.

We believe RR contains two kinds of information: round-trip time (rtt) and a parameter related to packet loss. We deduce this from the fact that Skype can display rtt and packet loss ratio of its counterpart on the CTID. Additionally, rtt and packet loss are commonly used metrics in many network protocols, such as RTCP [43].

We classify RR into three types, good RR, average RR, and bad RR, according to its influence on the receiver. The good RR reports little packet loss. It indicates that the network is in good condition and encourages the receiver to probe for more bandwidth. The bad RR reports large packet losses which exceed the packet loss threshold, $l_{threshold}$. It warns the receiver to reduce the rate. The average RR reports packet losses less than $l_{threshold}$. $l_{threshold}$ will be an important metric in our proposed CFSC scheme in Chapter 6.

3.5 Bandwidth Usage Target (BM)

Upon receiving an RR, Skype precisely controls the BM which in turn shapes the flow rate. The evidence is shown in Figure 3.6. In this experiment, we first decrease the link bandwidth between SA and the router from 8000 Bps to 2000 Bps (refer to Figure 3.1 for experimental testbed), then increase it back to 8000Bps. As we can see, the BM changes before the flow rate changes at time 70s and 175s. Since the BM decides the flow rate, it also decides the level.

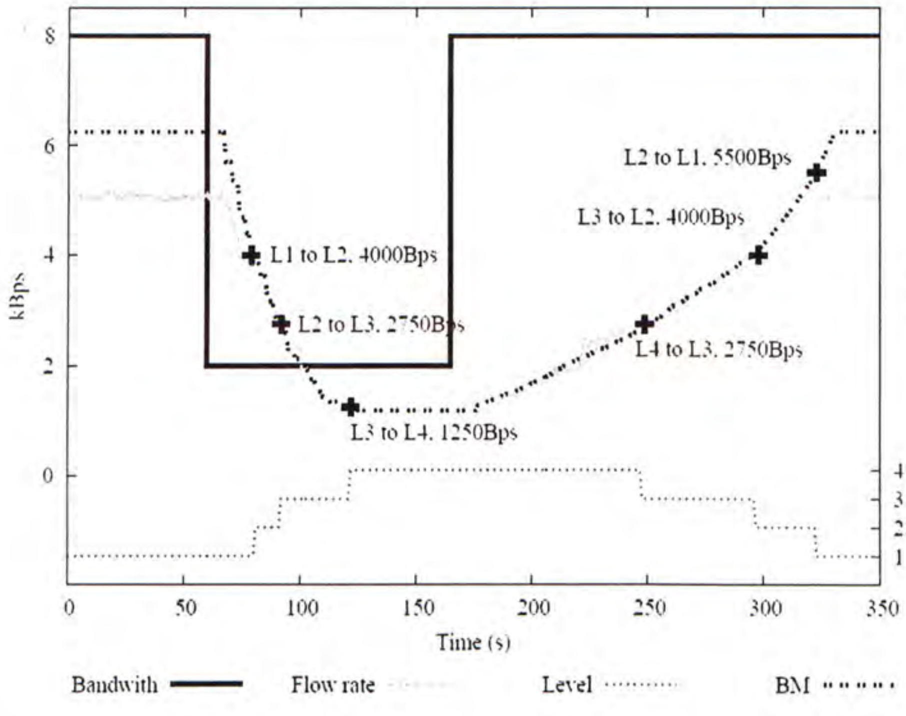


Figure 3.6: BM, flow rate, and level in response to sharp variation of the bandwidth

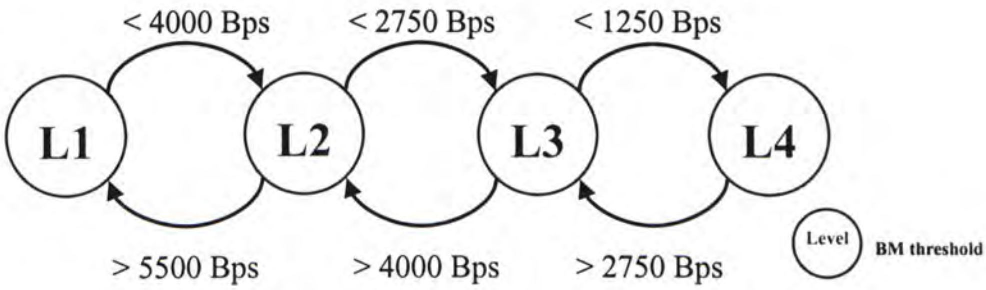


Figure 3.7: Level transition diagram

Embedded in Figure 3.6 is an interesting observation is that Skype sets two BM thresholds for level transition The mechanism is elaborated in Figure 3.7, which we arrive at by observing hundreds of level transitions in our experiments. In general, the level will be maintained to be the same when BM only changes slightly. A level transition will be triggered when the BM changes appreciably. This ensures a more stable voice quality, and that is why the flow rates of adjacent levels overlap with each other.

We then investigate how BM changes. After tracking down the times during which BM changes, we find that Skype updates the BM after and only after an RR is received. We also observe that the trend of BM depends on the types of the received RR. Next, let us break down the mechanism.

Decreasing mechanism of BM

After receiving a series of bad RRs, Skype realizes that congestion happens and then reduces the BM. Since RR contains packet loss information, Skype can estimate the minimum available bandwidth, B_{est}, and could have adjusted BM directly to the B_{est}. However, in Figure 3.6, the BM exhibits a smooth decrease rather than a slump in network bandwidth. Therefore, we conjecture that the new value of BM not only depends on the B_{est}, but also depends on the previous values of BM. In other words, the new value of BM is a weighted average of previous values of BM and the B_{est}.

Probing mechanism of BM

After receiving a series of good RRs, Skype will try to probe for more bandwidth by increasing the BM. BM increases in a preset manner. Based on our observation, the value of BM can be divided into four regions, and in each region, the increment of BM grows linearly. The four regions of BM are [1187 Bps, 2000 Bps], [2000 Bps, 3000 Bps], [3000 Bps, 5500 Bps], and [5500 Bps, 6250 Bps]. 1187 Bps and 6250 Bps are the minimum and maximum value of BM observed respectively. For instance, here is a series of values of BM for instance, 1638 Bps, 1679 Bps, 1721 Bps, 1764 Bps, and 1808 Bps. Their corresponding increments are 41 Bps, 42 Bps, 43 Bps and 44 Bps.

All discussions above only focus on the intermediate duration of a Skype session, and there might be something special that Skype set for the initial phase. Therefore, we performed a set of experiments with different initial link bandwidth between SA and the router. Figure 3.8 presents these experimental results. The initial bandwidths in Figure 3.8 (a), (b), (c), and (d) are 8000Bps, 6000Bps, 4000Bps and 2000Bps respectively. Comparing all the subfigures, we have three conclusions about the initial phase of Skype:

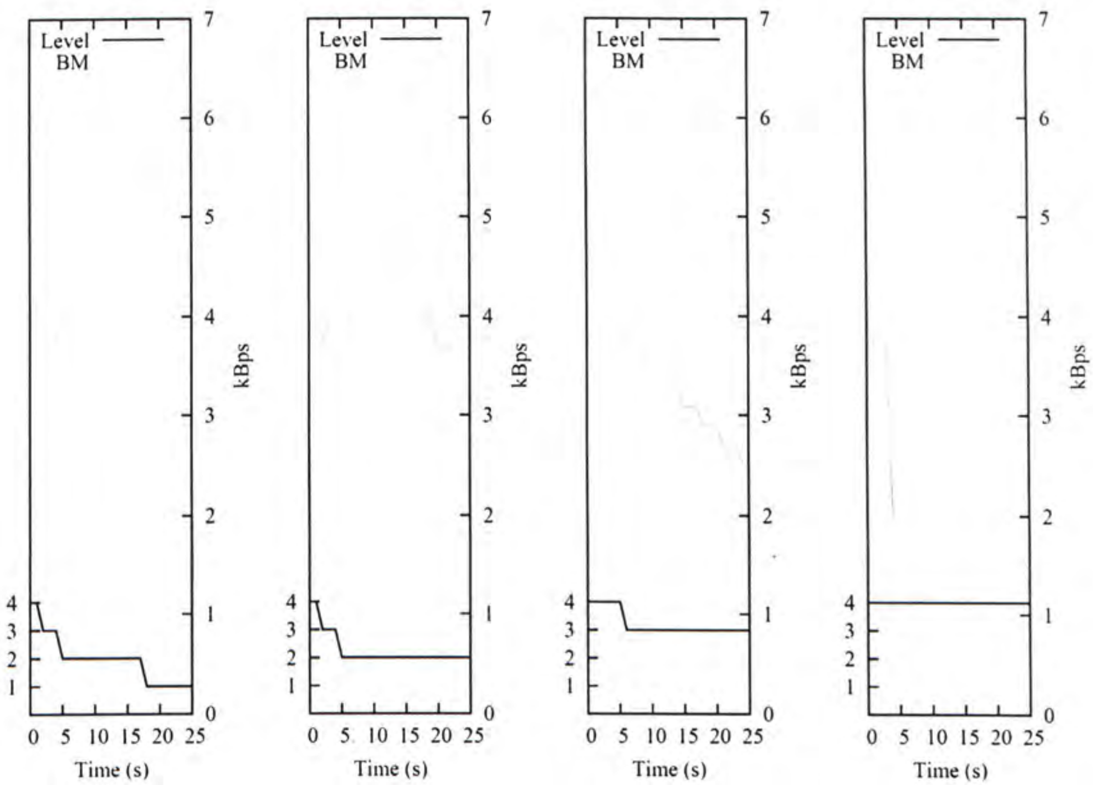


Figure 3.8: The performance of Skype adapting to different initial bandwidths. The initial bandwidths are (a) 8000Bps, (b) 6000Bps, (c) 4000Bps and (d) 2000Bps.

1. The initial BM is 3750 Bps and the initial level is L4.

   It seems that Skype starts quite conservatively at first. An interesting observation is that the initial value of BM (3750Bps) and the level (L4) seems not match each other according to the level transition diagram in Figure 3.7. We think it is a special setting that enables Skype to jump to L3 immediately if the bandwidth is enough to carry L3 traffic during the start phase.

2. Skype can successfully adapt its level to the initial bandwidth quickly. The decreasing and probing of BM still follows the principles discussed above.

3. During the first 20 s after the communication connection is established, the

increase/decrease amplitudes of BM are 3 times larger than those during the subsequent phase.

As shown in Figure 3.8(a), where the initial bandwidth is sufficient to carry L1 traffic, Skype can transit from L3 to L1 within 20 seconds. However, in Figure 3.6, the same transition costs Skype more than 60 seconds (from 250s to 320s).

Based on these conclusions, we divide the duration of a Skype session into two phases: conservative start phase and congestion avoidance phase. Conservative start phase is the first 20 seconds after the connection is set up, while the congestion avoidance phase is the following period.

3.6 Summary of Skype Rate Adaptation Mechanism

The following is a summary of our conjectured Skype rate adaptation operation.

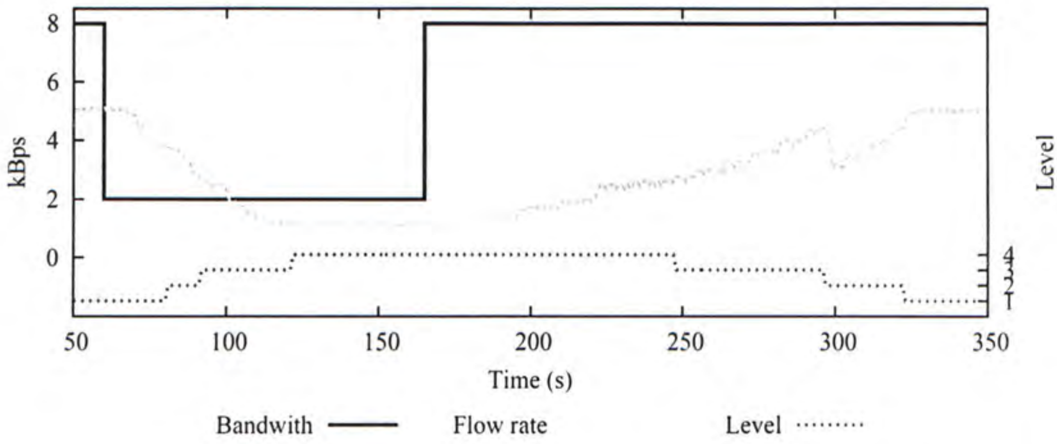
1. Initially set the BM to 3750 Bps and the quality level to L4.
2. Every 0.5 second, check whether 30 or more packets have been received. Piggyback an RR in normal packet if yes; otherwise, wait another 0.5 second.
3. When a series of bad RR are received or a timeout occurs, decrease BM. When a series of good RR are received, increase BM. Otherwise, keep BM unchanged.
4. The increase/decrease amplitudes of BM are 3 times larger during the conservative start phase than those during the congestion avoidance phase.
5. During both the conservative start phase and the congestion avoidance phase, the quality level is adjusted based on BM as depicted in Figure 3.7.

3.7 Skype-emulating Traffic Generator

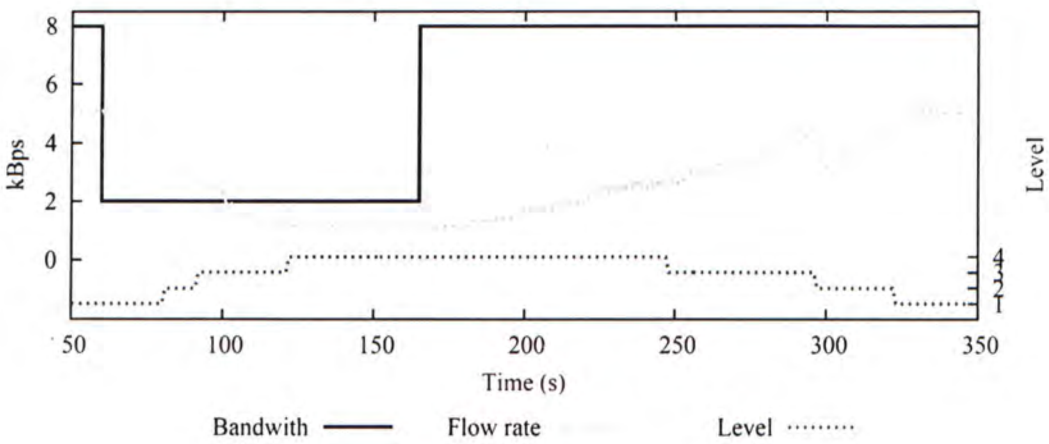
Based on our observations of Skype adaptation process, we have developed a

Skype-emulating Traffic Generator (STG) in Network Simulator (NS2) [44]. Important concepts and mechanisms of Skype rate adaptation, such as flow rate region, RR and BM, are all implemented. The source code of STG is available at CUHK-Wireless Networking Lab [45]. To install and use STG on NS2 platform, just follow the instructions in Appendix 1.

To validate the STG, we compare the traffic it generates with Skype's traffic under the same conditions. In Figure 3.9, the link bandwidth between SA and the router is changed sharply while in Figure 3.10, the bandwidth is changed step-by-step. In each figure, the subfigure (a) presents the flow rates and levels gathered from Skype's experiments; and the subfigure (b) presents the flow rates and levels generated by STG. We can see that STG traffic resembles Skype's traffic in both Figure 3.9 and Figure 3.10.



(a)



(b)

Figure 3.9: Flow rate and level in response to sharp variation in (a) experiment and (b) simulation

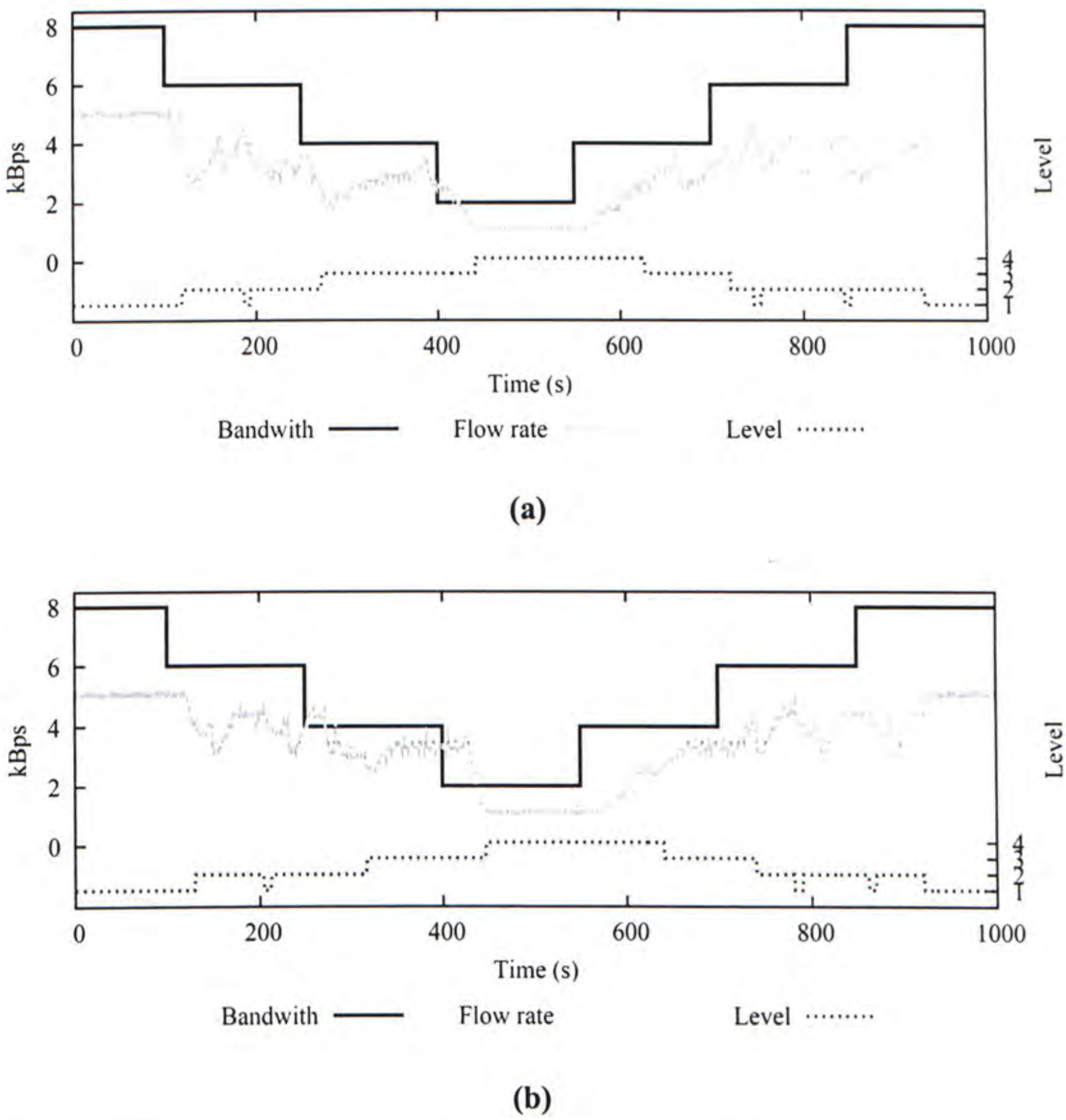


Figure 3.10: Flow rate and level in response to step-by-step variation in (a) experiment and (b) simulation

Chapter 4
Call Admission, Fairness and Stability Control

In this chapter, by using STG in designed simulations, we expose that simply limit the number of admitted calls will cause the unfairness and instability problems among the AVoIP sessions. Some sessions enjoy high voice quality while some others suffer periodic sever packet loss and unacceptable delay. To tackle these problems, we design a control scheme that integrates call admission, fairness, and stability control (CFSC) functions. The three functions work together to ensure that the whole WLAN system settles at an ideal operating point.

4.1 Unfair and Instability problems for AVoIP

4.1.1 Analysis

Unlike CVoIP, AVoIP could flexibly choose its rate in accordance with the network condition. As far as the maximum number of VoIP sessions is concerned, CVoIP has a fixed capacity over WLAN. For example, this maximum number is 12 when the codec used is GSM 6.10 and the WLAN is 802.11b. Admitting more than 12 CVoIP sessions will cause many VoIP packets to be dropped in the WLAN, resulting in unacceptable voice quality. For AVoIP, on the other hand, the capacity is not as clear cut. Even when the bandwidth of the WLAN has already been fully utilized, it

Table 4.1: Capacity under different targeted quality levels in 802.11b WLAN

L	$R_{\min}$ (Bps)	Airtime Usage per second (ms)	Capacity
1	4000	57.00	8
2	2812	29.08	17
3	1187	18.89	25
4	1140	14.35	34

is still possible to add a new session, since the existing AVoIP sessions could reduce their rates at the sources to make room for the new session. Reducing rate at the source allows more graceful degradation of voice quality compared with letting the network arbitrarily drop packets under congestion. The call admission could be more elastic under AVoIP than under CVoIP.

In the case of Skype, there are four levels of quality. By simple calculations [9], we can obtain the capacity (numbers of admissible AVoIP sessions, including two one-way flows) of a WLAN for different targeted quality levels. Table 4.1 presents the figures for 802.11b.

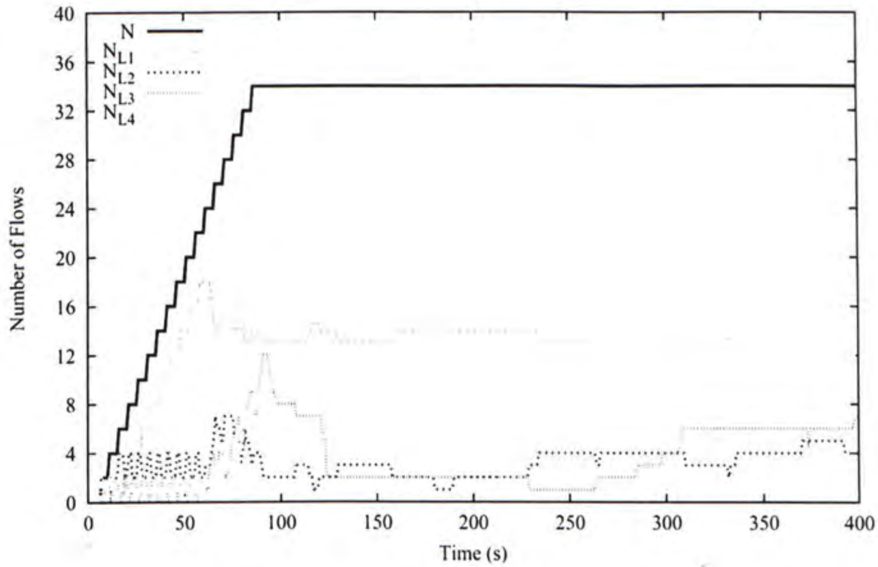
If we wish the users to have a quality of, say, at least L2[1], then we might limit the number of Skype sessions in the WLAN to 17. Note an assumption must be held is that all the admitted sessions share the same bottleneck – the WLAN. We relegate the discussion of the scenario without this assumption to Appendix 2. Under the bottleneck assumption, as long as there are fewer than 17 AVoIP sessions, we would continue to admit new sessions. When the limit is reached, new sessions would be

[1] We provide publicly available recordings for Skype AVoIP at different levels of quality at our website [45]. We consider L2 to be the minimum acceptable voice quality level in our study. The reader can listen to the recordings to make his/her own judgment

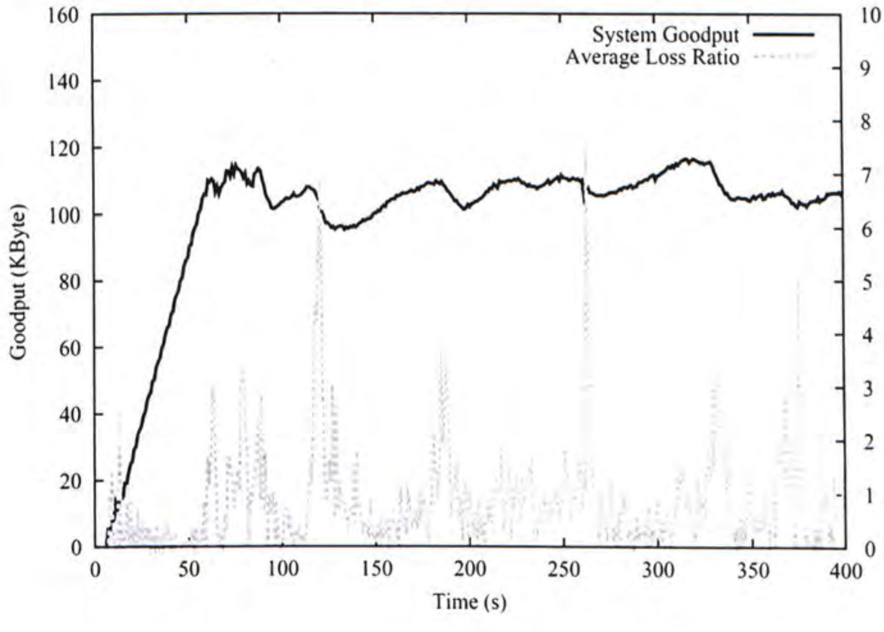
denied. This is similar to the simplistic user-number based cac scheme often used under CVoIP except that we have a higher capacity limit here. When the number of users is 8 or fewer, then all the users will be at L1. When the number of users is between 8 and 17, we expect that there is a mix of users at L1 and L2. It turns out that things may not work as expected because of the intricate dynamics of the rate adaptation process. From our simulations, we discover that there could be users at L3 and L4 even when the number of users is fewer than 17. Meanwhile, there could also be users at L1. In other words, there is a high degree of unfairness among the users if the system is left to evolve according to rate adaptation processes of the respective sources, even though the users use identical rate adaptation mechanism. In the following, we will illustrate this phenomenon with Skype-based simulation results.

4.1.2 Simulation Evaluation

We have performed comprehensive NS2 simulation to evaluate the performance of the simplistic user-number-based cac scheme. We model a "WLAN cell" with a square area of 200m*200m (refer to Table 2.1 for parameter values of 802.11b DCF). An AP is placed at the center of the cell while wireless client stations are placed randomly in the area. All nodes are within the carrier-sensing range of each other and there is no hidden node. Our Skype-emulating Traffic Generator is used to produce the traffic of AVoIP sessions. Altogether, 20 AVoIP sessions request to join the service. Starting from time 5 s, their requests arrive one by one separated by 5-second intervals. The total simulation time is 400 s. With a targeted minimum quality of L2, the capacity limit is set to 17. Thus, the last three requests are denied admission. We have run 20 simulations under the same setting, and all the simulations exhibit similar results.



(a)



(b)

Figure 4.1: User-number-based CAC scheme: (a) Level distribution versus time; (b) System throughput and average loss ratio versus time

Figure 4.1 presents the results taken from one of the simulation runs. Figure 4.1(a) is related to distributions of individual performance. Specifically, we show the numbers of flows at different levels (one bidirectional voice session has two one-way flows).

Figure 4.1(b) is related to the overall system performance. Specifically, we show the overall system goodput and the overall average end-to-end loss ratio. Note the overall end-to-end packet loss consists of packet losses i) at the congested router; ii) at the WLAN due to collisions; and iii) at the application layer due to unacceptable delay. The simulation results expose two problems, as detailed below.

1. Unfairness

   Perhaps the most significant observation is that the qualities enjoyed by users are unevenly distributed. As can be seen from Figure 4.1(a), the qualities of many sessions fall far short of the L2 minimum requirement even though the capacity limit is not exceeded. There are many flows at L1 and L4. Ideally, the flows at L1 should yield some of their bandwidth to those at L3 and L4 to bring about better fairness. However, different sources run their own rate adaptation processes in a distributed manner, and the evolution of the rate adaptation processes at the sources does not bring about this desirable outcome. Another observation is that newly added clients are prone to starvation. As shown in Figure 4.1(a), from 50 s to 100 s, many clients slide into L4. Most of them are new clients who come after time 50 s. We believe this phenomenon can be attributed to the "conservative start" property of Skype: new clients start from L4 and react more aggressively to congestion.

2. Instability

   As a whole, the system suffers from recurrent jumps in packet loss and slumps in goodput, as shown in Figure 4.1(b). These are side effects of the probing mechanism of AVoIP. Periodically, each client will attempt to raise its own rate if there has been no significant packet loss. As each client does that, there eventually will come a point when their aggregate will drive the network to beyond its capacity limit. The overall system instability and oscillatory behavior will in turn cause fluctuations of the voice qualities perceived by individual users. As can be seen in Figure 4.1(b), even after no more new sessions are admitted and the number of sessions is a constant, the distribution of users among the different quality levels fluctuates in a significant manner.

The unfairness and instability problems may not be unique to Skype. In AVoIP, it is natural for a source to probe the network for more bandwidth if it is not enjoying the best possible quality and that it has not experienced significant packet losses or delays. That is how it can make sure that the network bandwidth can be fully utilized. It is also reasonable to begin service conservatively at a lower level before raising it later, but this leads to unfairness for newcomers, as our experiments indicate. Our experimental observations point out the need for a careful reexamination of rate adaptation to take into account the interactions among the processes at the individual sources - without further regulation and coordination among these processes, the system may evolve to an undesirable operating point.

4.2 CFSC scheme

We propose an integrated scheme to solve the unfairness and instability problems. Our scheme, CFSC, consists of three control functions: Call Admission Control (CAC), Fairness Control (FC) and Stability Control (SC). Using CFSC, we could ensure consistent good voice quality to every client.

The AP of the WLAN is responsible for performing the CFSC scheme. It handles the

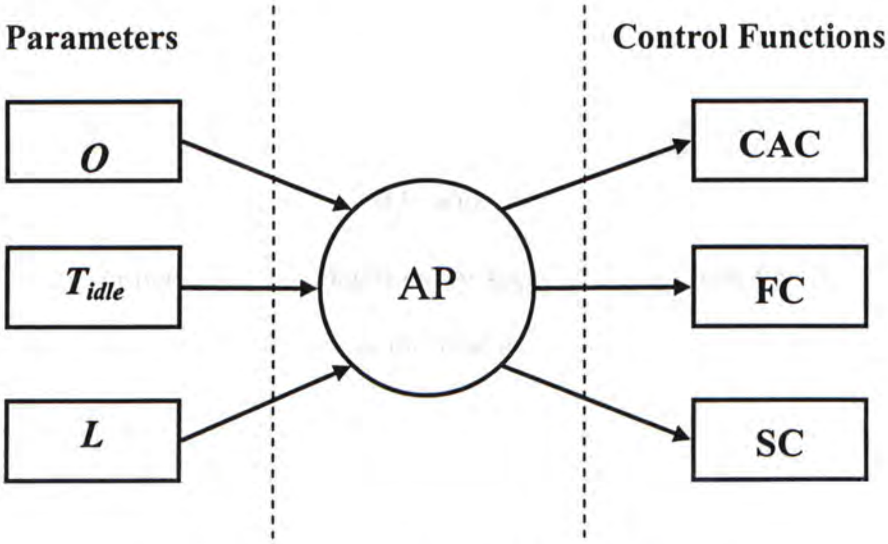


Figure 4.2: Overview of CFSC scheme

access requests from clients, identifies potential problems among clients, and executes the CAC, FC, and SC control functions.

The AP maintains a request queue Q, and two parameters , T_{idle} and L. These parameters are collected and updated every second. They assist the AP to identify and deal with the problems. T_{idle} is the idle time over the period of one second at the MAC layer. L is a vector recording the levels of all flows. Since the network is working in the infrastructure mode, the AP can easily monitor the traffic in both directions and collect these information. T_{idle} can be observed at the MAC layer or estimated at the network layer [29]. In our work, the AP estimates the T_{idle} by calculating the airtime used by all the packets passing it. Although this method induces small error, it needs not cross-layer cooperation between the network layer and the MAC layer. The level of a flow can be determined through the packet interval without accessing the application header.

CAC, FC and SC operate in parallel. Every second, the AP checks Q, T_{idle} and L, and decides which control function should be performed. Figure 4.2 gives an overview of the CFSC scheme. Below we will explain the three control functions separately. We continue to assume L2 is the minimum quality requirement. But note that our scheme works for other requirements and only some control parameters need to be changed accordingly. The overall algorithm for CFSC is contained in the pseudo code of Algorithm 1.

4.2.1 Pre-admission Bandwidth-reallocation Call Admission Control (PBCAC)

We continue to use the capacity limit, 17, as an upbound of the admitted number of sessions in our CAC. When a new client wants to join the service, it first sends a request to the AP. We assume a client will wait for a maximum of 10 seconds for a reply from the AP. Thus, the AP has a budget of 10 seconds to perform the necessary computation and control before a decision has to be given to the client as to whether it is admitted or not.

Upon receiving a new request, the AP immediately admits it if there are less than eight existing clients in the WLAN (the rationale is that, according to Table 4.1, all sessions could be at L1). If there are more than 17 clients, the request will be denied. In other cases, the AP appends the request to the request queue, Q.

Every second the AP checks whether Q is empty. If not, it checks whether T_{idle} can accommodate a new AVoIP session demanding L2 service. If $T_{idel} \geq 2T_{L2}$, where T_{L2} is the minimum airtime requirement for an one-way flow at L2, the AP

Algorithm I CFSC Algorithm

Procedure HANDLE NEW REQUEST (Q)
    if Number of Existing Clients < 8 **then**
        Admit the new request;
    else if Number of Existing Clients > 17 **then**
        Deny the new request;
    else
        Add the new request to Q;
    end if
end Procedure

Loop ;*The AP checks* Q, T_{idle} *and* L
  Procedure CALL ADMISSION CONTROL(Q, T_{idle})
      if Q is not empty **then**
          if $T_{idle} \geq 2T_{L2}$ **then**
              Admit the first request in the Q;
          else
              Blacklist-I 2 L1 flows;
              Blacklist-II the left L1 and all L2 flows;
              Set $l_{d2} < l_{d1} < l_{threshold} < 3\%$;
              Start Deliberate Drop.
          end if
      end if
  end Procedure

  Procedure FAIRNESS CONTROL(L)
      if $(N_{L1} + N_{L2} \neq 0) \cap (N_{L3} + N_{L4} \neq 0)$ **then**
          Blacklist-I $\lceil 0.5N_{L3} + N_{L4} \rceil$ L1 flows;
          Blacklist-II the left L1 and all L2 flows;
          Set $l_{d2} < l_{threshold} < l_{d1} < 3\%$;
          Start Deliberate Drop.
      end if
  end Procedure

  Procedure STABILITY CONTROL(L)
      if $L = L_{ideal}$ **then**
          Blacklist-II all L1 and L2 flows;
          Set $l_{d2} < l_{threshold}$;
          Start Deliberate Drop.
      end if
  end Procedure

end Loop

grants admission to the first request of Q. It then updates T_{idle} accordingly. Otherwise, the AP takes a measure called "deliberate drop", the goal of which is to make the existing sessions at higher level to yield some of their bandwidth to accommodate the new session. The AP rechecks T_{idle} after one second to see if the deliberate drop has succeeded in creating enough idle airtime for the admission of the new client. Deliberate drop is performed until enough idle airtime is created, or the request is withdrawn after waiting 10 seconds.

Deliberate Drop

Step 1: If there are 2 or more L1 flows, the AP arbitrarily puts 2 of them on Blacklist-I, and puts the left ones and all L2 flows on Blacklist-II (the rationale is that, according to Table 4.1, the airtime taken by 1 L1 flow can accommodate 2 L2 flows). Otherwise, the AP puts the only L1 flow, and all L2 flows except the ones at the lowest rate of L2 on Blacklist-I, and puts the left L2 flows on Blacklist-II (due to limited space, the pseudo code in Algorithm 1 does not cover this case.). Step 2: In the next second, the AP will deliberately drop some packets of the Blacklist-I flows and the Blacklist-II flows with drop ratios l_{d1} and l_{d2} respectively, where $l_{d2} < l_{threshold} < l_{d1} < 3\%$ (assuming the bearable loss ratio of VoIP traffic is 3%).

Deliberate drop drags down the flow rates of the Blacklist-I flows and prevents the Blacklist-II flows from probing. In other words, deliberate drop forces the L1 flows and L2 flows at higher rate to yield some airtime for the new session, and prevents the L2 flows at lower rate from taking over the idle time created before the new session is added. If there are existing L3 flows also, Fairness Control will blacklist more L1 flows to yield more airtime to the new session and the existing L3 flows. In short, our CAC mechanism forces some L1 and L2 flows to give up some bandwidth so as to make room for the new session. This scheme avoids over-admission and guarantees the new session transits from L4 to higher levels quickly during the conservative start phase.

4.2.2 Fairness Control

The AP detects unfairness by inspecting L. Unfairness is indicated when L1 or L2 flows coexist with L3 or L4 flows. More specifically, unfairness is said to occur when the following event occurs: $(N_{L1} + N_{L2} \neq 0) \cap (N_{L3} + N_{L4} \neq 0)$, where N_{Li} denote the number of flows at level i.

Upon detection of unfairness, the AP performs deliberate drop to force the flows at higher levels to give up some bandwidth to the flows at lower levels. This procedure is similar to the one in CAC except $\lceil 0.5N_{L3} + N_{L4} \rceil$ L1 flows are put into Blacklist-I here (the rationale is that, according to Table 4.1, if 1 L1 flow degrades to L2, it will at most yield about 28 ms airtime per second, which is enough for 2 L3 flows or 1 L4 flow to upgrade to L2).

Fairness Control fairly and efficiently reassigns the bandwidth resource to flows without communication overhead. With Fairness Control, the level distribution of all flows could converge to the ideal operation point, where max-min fairness and maximum system goodput is achieved. We will give detailed description of the ideal operation point in the next part.

4.2.3 Stability Control

Stability Control prevents the system from diverging again after converging to the ideal operating point. The divergence is due to the probing mechanism of AVoIP. We define the ideal operating point, L_{ideal}, as the level distribution when max-min fairness among clients is achieved - an upgrade of any clients must not result in the degradation of some lower-level clients. The ideal operation point can be obtained through calculation [2]. In Table 5.1, the column of "ideal operation" reports the ideal operation points versus different numbers of clients.

The Stability Control function is conducted as follows: the AP checks L every second. When $L = L_{ideal}$, the AP starts Stability Control. It puts all flows, including all the L1 flows and L2 flows, into Blacklist-II. Then it set $l_{d2} < l_{threshold}$ to merely prevent them from probing. Stability Control keeps the system working at the ideal operation point where max-min fairness is achieved.

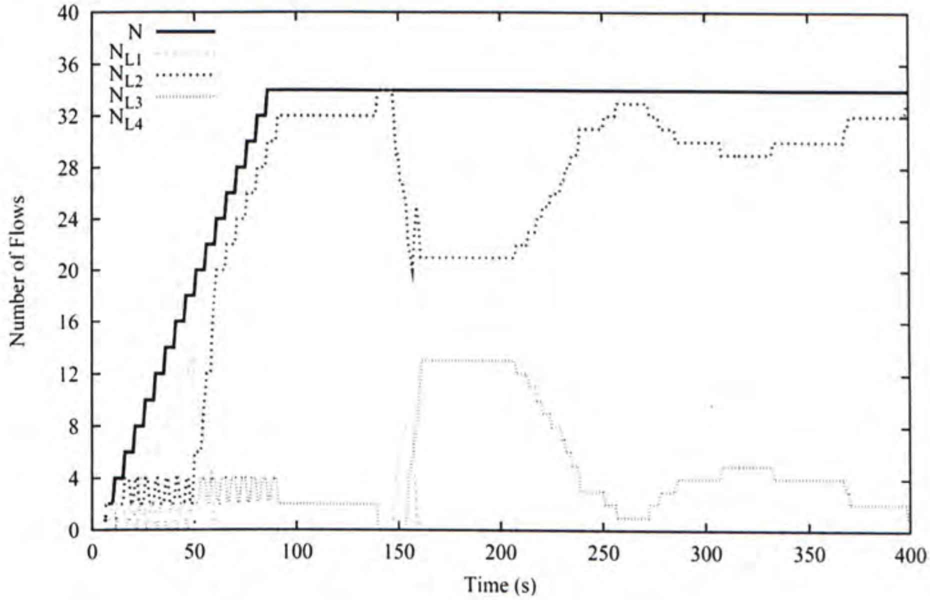
Chapter 5
Performance Evaluation of CFSC

We validate CFSC by simulations in NS2. For comparison, the simulation setting is the same as that in Chapter 4.1.2. An AP is placed at the center of a "WLAN cell" with a square area of 200m*200m. Wireless clients are randomly placed in that area with a uniform distribution. Altogether, 20 clients wish to initiate AVoIP sessions. Starting from time 5s, their requests arrive at the AP one by one every five seconds. In the following, to better identify their roles and effects, we isolate and evaluate Fairness Control, Stability Control and Call Admission Control separately.

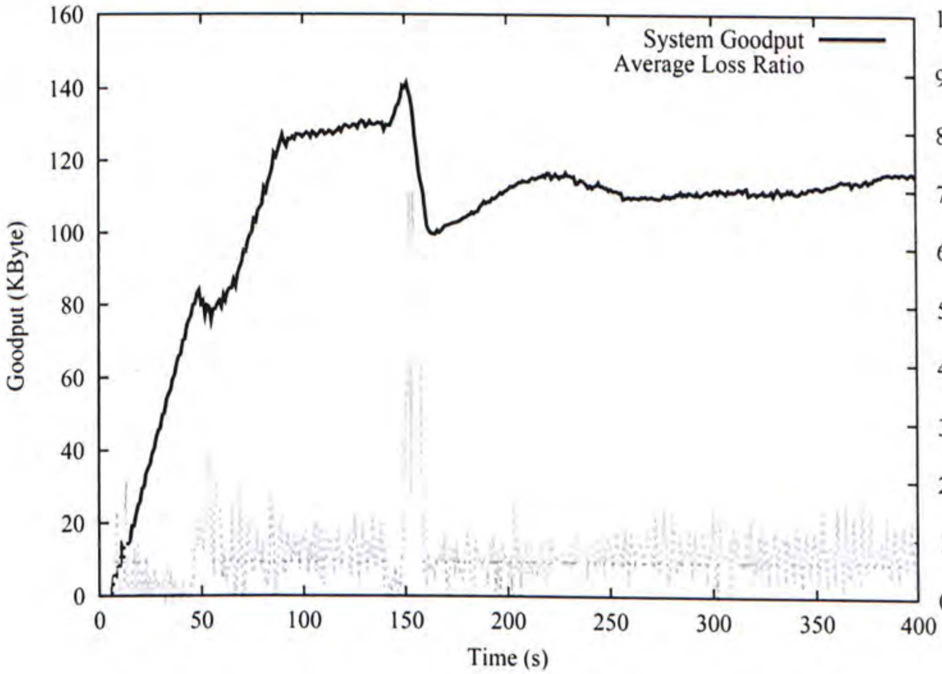
5.1 Evaluation of Fairness Control

In this simulation, the Call Admission and Stability Control functions are disabled, and only the Fairness Control function is activated. For comparison, we adopt the same user-number based CAC that is used in Figure 4.1. Figure 4.1(a) presents the distributions of the performance of individual VoIP sessions; Figure 4.1(b) presents the overall system performance. As we can see, at time 145s, the system converges to the ideal operating point when all clients are at L2. In particular, the unfairness problem in Figure 4.1 has been removed with our fairness control algorithm.

At 150s, however, some clients attempt to go to L1 and succeed in doing so. This causes a surge of packet loss and a slump in system throughput. The system diverges from the ideal operating point. As discussed in Chapter 4.1, this instability problem is due to the probing mechanism of AVoIP. It takes a long time before ideal



(a)



(b)

Figure 5.1: Only Fairness Control activated: (a) Level distribution versus time; (b) System throughput and average loss ratio versus time

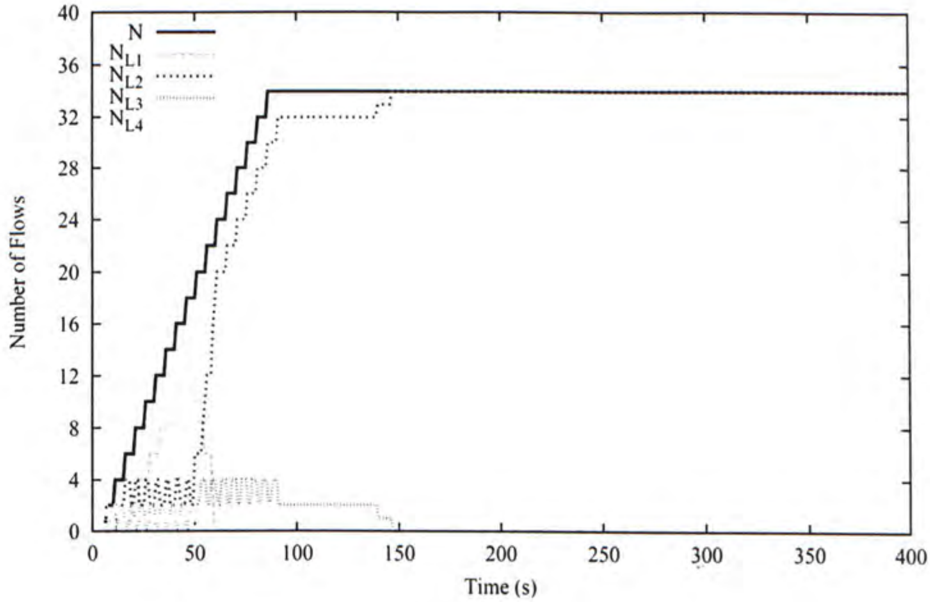
operation is achieved again (at 400s). This result indicates that we need Stability Control to prevent the system from diverging from the ideal operating point.

5.2 Evaluation of Stability Control

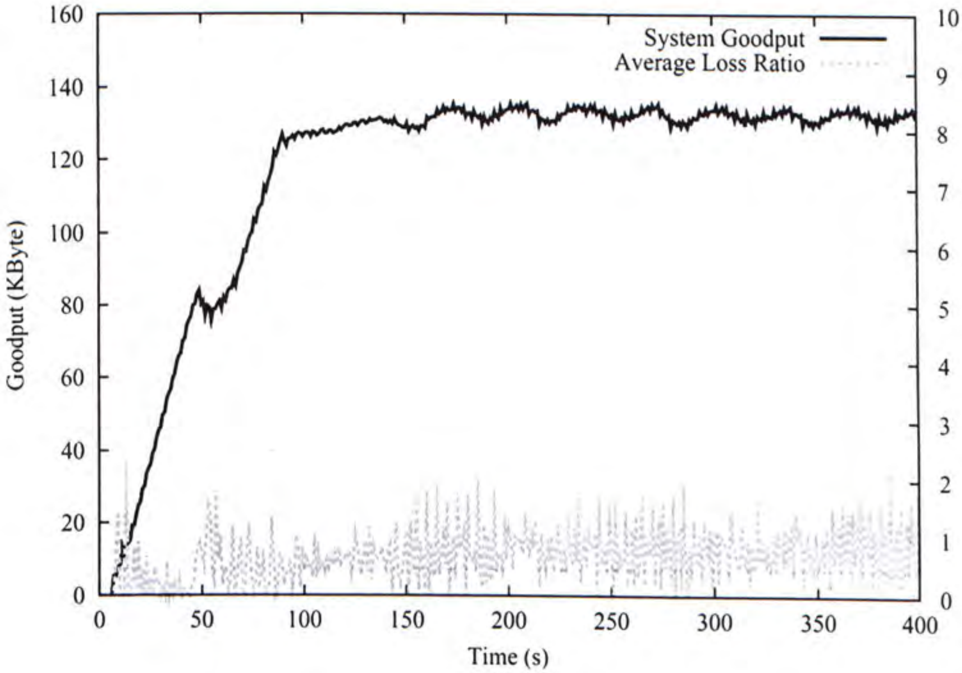
In this simulation, the Fairness and Stability Control functions are activated, together with the user-number based CAC. The distributions of individual performance and the overall system performance are presented in Figure 5.2. As we can see, under the coordination between the Fairness Control and Stability Control, fairness among all clients and consistent good voice quality are guaranteed.

5.3 Evaluation of PBCAC

In this simulation, we replace the user-number based CAC with our own Call Admission Control, and also activate the Fairness Control and Stability Control functions. Simulation results are presented in Figure 5.3. Comparing Figure 5.2 and Figure 5.3, we can see that our CAC outperforms the user-number based CAC in two ways: i) Our CAC achieves better bandwidth usage. Under our CAC, the number of L1 clients decreases more gently when the total number of clients increases in the WLAN between 50s and 100s. We also achieve higher system goodput. ii) Under our CAC, level distribution converges to the ideal operation point more quickly. The reason is that before a new session is admitted, we have reserved enough airtime for it to transit to the higher levels quickly during the conservative start phase. Our improvement is at the expense of longer waiting time of clients. The average waiting time is about three seconds under our CAC while zero under the user-number based CAC.

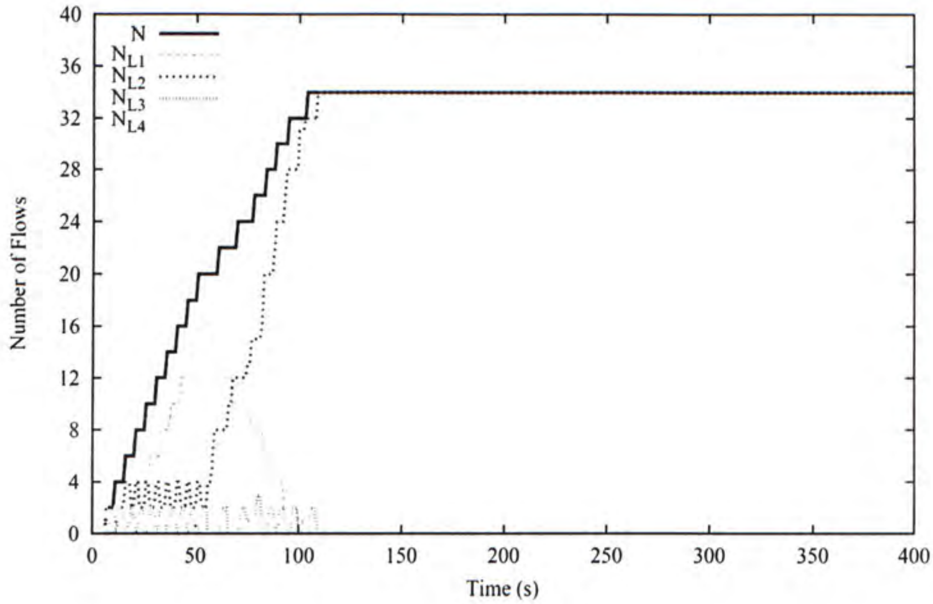


(a)

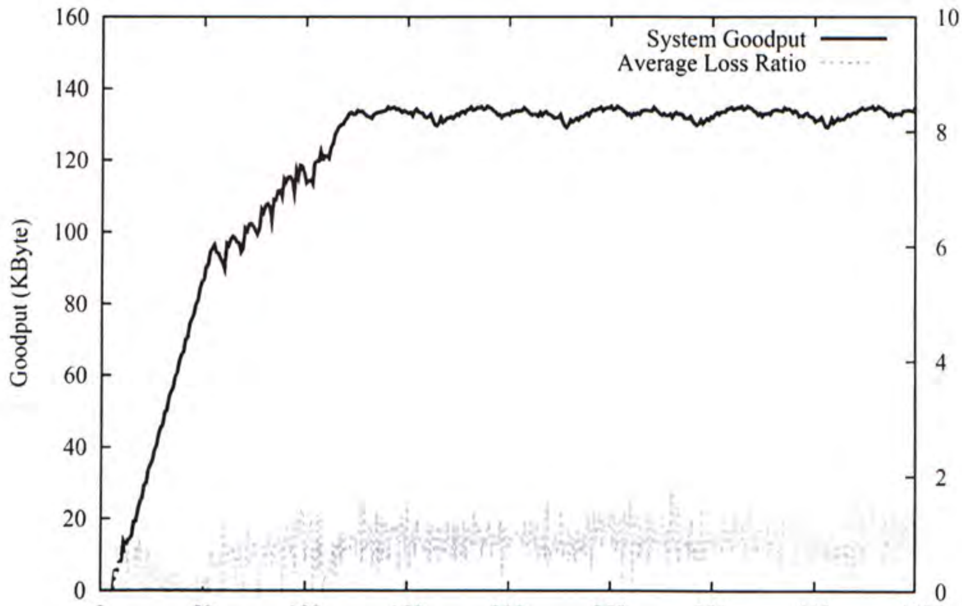


(b)

Figure 5.2: Fairness Control and Stability Control activated: (a) Level distribution versus time; (b) System throughput and average loss ratio versus time



(a)



(b)

Figure 5.3: Complete CFSC scheme: (a) Level distribution versus time; (b) System throughput and average loss ratio versus time

5.4 Evaluation of complete CFSC

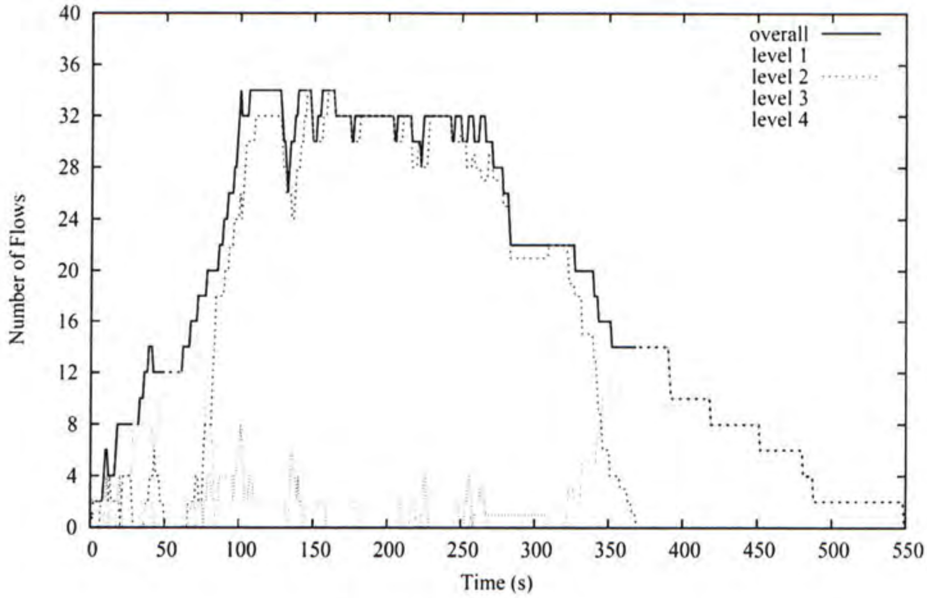
To further evaluate the overall performance of the CFSC scheme, we conduct simulations with different number of clients requesting to initiate AVoIP sessions. For each case, we run 10 simulations. The results of 10 simulations are similar. Table 5.1 presents the results taken from one simulation. It reports the level distribution under the user-number based CAC scheme, and under the CFSC scheme, versus the number of requesting clients. We also list the ideal operation points in the table. In all the 10 simulations, the system can always converge to the ideal operation points regardless of the number of clients under our CFSC scheme. But under the user-number based CAC, the level distribution displays a high degree of unfairness.

Another simulation that demonstrates the performance of our CFSC scheme is presented in Figure 5.4, where 50 clients join and leave the WLAN during a period of 550s. Arrivals of clients are a Poisson process with rate $0.2s^{-1}$, and the duration of a voice session is exponentially distributed with parameter $\frac{1}{150}s^{-1}$. As we can see, even under this dynamic scenario, our CFSC scheme performs well. Voice quality of L2 is guaranteed when there are more than 8 clients in the WLAN from 100s

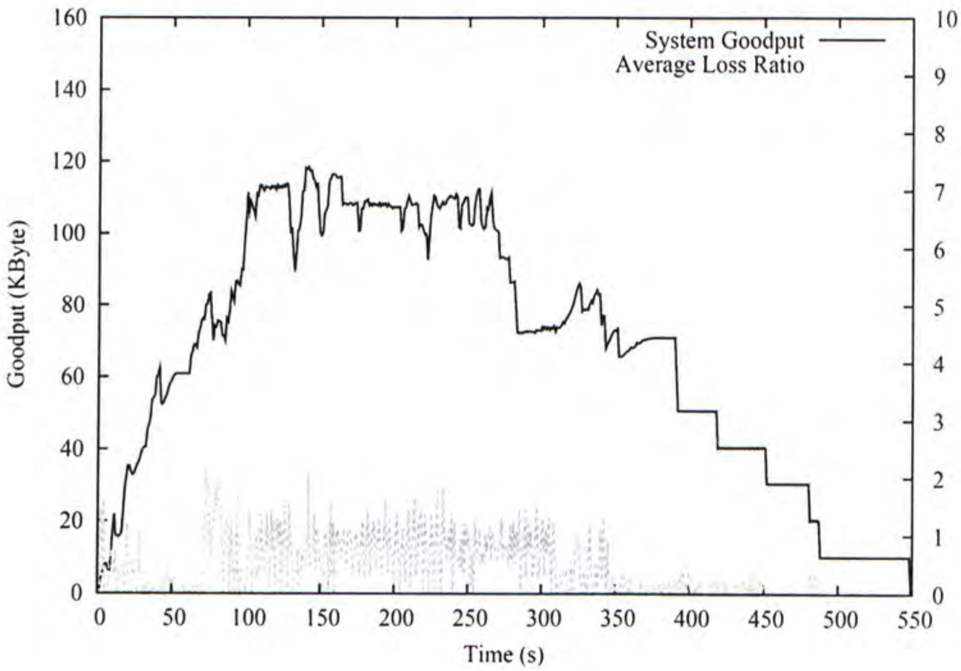
Table 5.1: Level distribution versus number of clients

	User-number-based CAC				CFSC				Ideal Operation			
N	L1	L2	L3	L4	L1	L2	L3	L4	L1	L2	L3	L4
8	16	0	0	0	16	0	0	0	16	0	0	0
10	18	1	1	0	14	6	0	0	14	6	0	0
12	17	1	2	4	10	14	0	0	10	14	0	0
14	16	1	3	8	6	22	0	0	6	22	0	0
16	13	2	7	10	2	30	0	0	2	30	0	0
17	12	2	7	13	0	34	0	0	0	34	0	0

to 300s, and voice quality of L1 is provided when there are less clients after time 300s.



(a)



(b)

Figure 5.4: Complete CFSC: 50 clients join and leave over a period of 50s. (a) Level distribution versus time; (b) System throughput and average loss ratio versus time

Chapter 6
Conclusion

Accelerating adoption of WLAN and increasing maturity of VoIP technology have attracted great interests towards VoIP over WLAN from both industry and academia. Much previous research work has studied Constant bit-rate VoIP over WLAN. This thesis, inspired by the reality that many popular VoIP applications have employed Adaptive bit-rate VoIP codecs, studies AVoIP over WLAN.

We investigate the rate adaptation mechanism of Skype and implement a traffic generator module in NS2 to emulate it. With the help of the traffic generator, we identity two problems associated with the transport of Adaptive bit-rate VoIP (AVoIP) traffic over WLAN. We then devise solutions to solve the problems.

We find that when AVoIP is adopted by the end users, the AVoIP-over-WLAN system could evolve to an operating point where there is a high degree of unfairness among the AVoIP sessions. In particular, some sessions enjoy very good voice quality while other sessions suffer from very poor voice quality. Furthermore, the performance may be unstable in that a session may oscillate between good and poor voice qualities.

To solve these problems, we develop a scheme that integrates the call admission, fairness and stability control functions. Our call admission control makes use of the principle of pre-admission bandwidth reallocation. In essence, if the WLAN is already saturated when a new call arrives, and if some existing sessions are enjoying the highest possible voice quality and that lowering their quality somewhat is still acceptable, then the AP deliberate drops some packets of these existing sessions to cause them to adapt to a lower but acceptable voice quality level. This has the effect of reshuffling the bandwidths of the existing sessions so as to make room for the new call. Doing so allows the new call to adapt to an acceptable quality level very quickly.

Our fairness control makes sure that among the existing calls, their quality levels are comparable and not drastically different. The fairness equilibrium is also achieved by deliberate and selective dropping of packets at the AP. The goal, however, is to prevent the interactions among the users' adaptation mechanisms to cause the system to evolve to an unfair operating point. Our fairness control achieves max-min fairness. Finally, stability control ensures that once fairness is achieved, the system does not diverge from the ideal operating point due to the bandwidth-probing mechanism of AVoIP. Experimental results indicate that our proposed scheme effectively eliminates the unfairness and instability problems we observe for AVoIP over WLAN operation.

Appendices

Appendix 1

Using Skype-emulating Traffic Generator (STG) in NS2

A. Overview

The Skype-emulating Traffic Generator is developed to generate adaptive voice traffic in NS2 simulations for research on AVoIP. The motivation behind is that NS2 lacks support for modeling adaptive voice traffic which is quite popular in practice. As its name indicated, STG can emulate the behavior of Skype adapting to network variations. STG is fully implemented the Skype rate adaptation mechanism described in Chapter 3. Two STGs working as counterparts can produce and feedback receiver reports following the RR generation rule, update BM upon receiving an RR, and adjust flow rate accordingly.

B. Installation

The STG module consists of three files, *skype.h, skype.cc* and *packet.h*. They are downloadable at our lab website http://www.wireless.ie.cuhk.edu.hk.

To install the STG module to your NS2 platform, you should:

Step 1. Download the files *(skype.h, skype.cc)* to a directory under ns-allinone-2.3x/ns-2.3x named "skype" (or any name you wish)

Step 2. Make the following change to the "Makefile":

(i) Add *$skype/skype.o* to the OBJ_CC macro.

(ii) Add *-I./$skype* to the INCLUDES macro.

Step 3. Change some of the files in the ns-2 distribution:

(i) "ns-process.h". Add the following lines before the last ADU in AppDataType enumerator at the begining of the file:

//Skype ADUs

SKYPE_DATA,

SKYPE_DATAwRR,

SKYPE_SYN,

(ii) "packet.h". Download *packet.h* and replace the old one at /ns-allinone-2.33/ns-2.33/common.

Step 4. Recompiling ns with the *configure; make clean; make* command.

After completing those steps, STG is usable on the target NS2 platform. The following section will show how to use STG.

C. Usage

STG is an application layer traffic generator working on User Data Protocol. Its usage is quite like using CBR generator. The only difference is that STG must be used in pair. If the sender side uses STG as generator, the receiver side must also use the STG. It is because that STG needs feedback to perform rate adaptation.

The commands to adopt and use STG are as below:

Step 1. To attach STG to UDP agents:

set skype [new Application/Skype $udp]

Step 2. To start and stop STG traffic:

$skype start

$skype stop

Appendix 2

Performing CAC When Bottleneck is not in the Local WLAN

A. Motivation and Overview

An assumption held in this thesis is that all VoIP flows in the WLAN are constrained by the same bottleneck - the local WLAN. Then if a VoIP flow has poor performance, it must indicate that the WLAN is saturated. As for AVoIP, the poor performance might also be a warning signal of unfairness problem. This assumption is reasonable if we assume that all the counterparts of the VoIP callers in WLAN are in the wired network where bandwidth is usually abundant. However, if one end of a VoIP session is in a wireless network, this assumption does not hold anymore. Figure A.1 gives an example. Apparently the bottleneck of VoIP session aa' is the local WLAN. But for

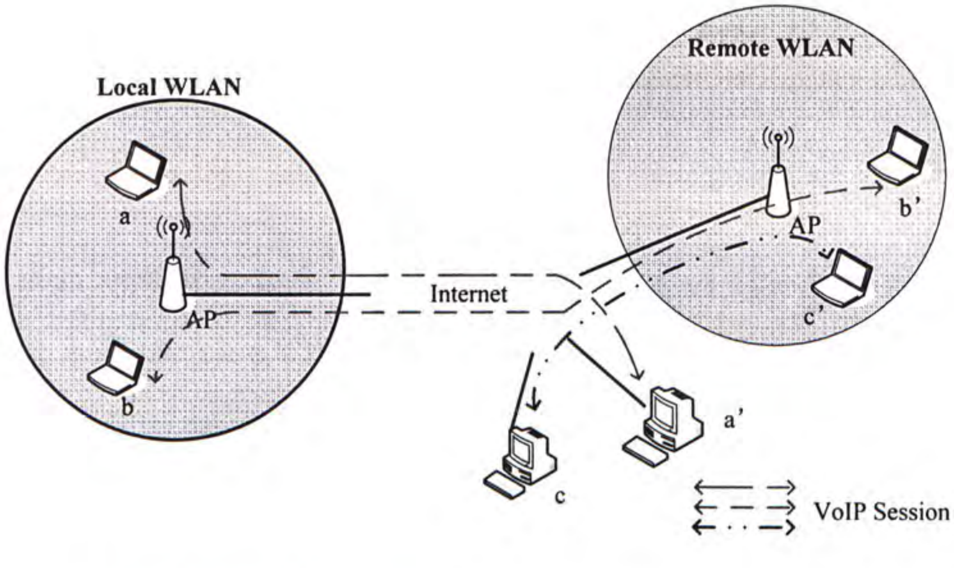


Figure A.1: An example scenario where some counterparts of users in Local WLAN are in the wired network and some ones are in Remote WLAN.

session bb', either the local WLAN or the remote WLAN could be the bottleneck. Besides, the bottleneck of session bb' could even shift back and forth between local and remote WLANs. In this situation, poor performance of a flow might not necessarily indicate that the local WLAN is congested or unfairness problem exists. It is also possible that the poor performance is due to the congestion of the remote WLAN.

If we remove the bottleneck assumption, two issues need to be reconsidered in CFSC scheme:

1. Indication of unfairness that triggers fairness control;

As discussed above, a low quality level does not always indicate unfairness. If we blindly perform fairness control upon every such signal without further identification, the system might enjoy no improvement but suffer higher packet loss induced by deliberate drop (in fairness control).

Define the flows, whose bottlenecks are in the remote networks as "remote flows" and the flows, whose bottlenecks are in the local WLAN as "local flows". To solve this issue, we need to identify the remote flows. A possible method is like this: at the beginning, AP executes fairness control upon poor performance of any flow. After a period T, if the performance of the flow in question remains unimproved, AP assumes that the flow is a remote flow and marks the flow. Then, during the following period T', AP will ignore the poor performance of the marked flow. Note here $T' \gg T$.

2. The upper bound of the admitted number in call admission control.

Continue to assume that the requirement of voice quality is L2. Recall that in Chapter 5, we limit the upper bound of the admitted number of voice sessions to the WLAN capacity, 17. If we still use the same limitation here, in the worst case where all flows are remote flows, a half of the wireless resource of the local WLAN is wasted (Assume remote flows are all at L4. The rationale is that, according to Table 5.1, the airtime taken by 1 L2 flow equals to the airtime taken by 2 L4 flows.). On the other hand, if we loosen the upper bound and admit more sessions, we may fail to maintain the quality of admitted voice flows. Once some remote flows change to local flows, the system capacity is exceeded and everyone's quality is hurt.

This issue is quite complicated. We will discuss it elaborately in the following sections. In section B, we will give a model to analyze this issue. In section C, D and E, we will present three kinds of CAC scheme for the situation that the bottleneck is not in the local WLAN. The first scheme limits the number of admitted sessions to an estimated upper bound. The second scheme reserves an amount of bandwidth for remote flows. The third scheme guarantees the voice quality of flows with a required probability.

B. Analytical Model

Assume the arrivals of call requests follow a Poisson process with rate λ, and the duration of a call is exponentially distributed with parameter μ. According to its bottleneck location, a directional voice flow may have two states: remote state and local state. Assume the durations that a flow stays in remote state and local state

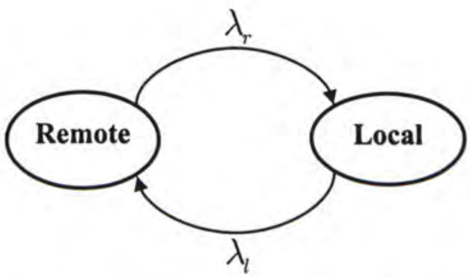


Figure A.2: The State diagram of a flow

follow exponential distribution with parameter λ_r and λ_l respectively. Figure A.2 gives the state diagram for a flow. Further assume the flow transitions are independent.

Suppose at time t_0, a new request arrives and AP has to decide whether to admit it. At this moment AP has the following information: the overall number of flows in the WLAN, N; the number of remote flows, N_r; the number of local flows, N_l; the average airtime taken by a remote flow, B_r (ms); the airtime required by a local flow with accepted quality, B_l (ms) ($B_r < B_l$); and the transition rates between remote state and local state, λ_r and λ_l .

Assume a new admitted flow starts from local state and no voice session leaves during the following period h. Define danger probability as the probability that the WLAN capacity C will be exceeded by adding a new session. Specifically, if at any time during $[t_0, t_0 + h]$, $N_l B_l + N_r B_r > C$ is satisfied, we say the capacity is exceeded. Then the problem can be reformulated as: based on current information, how should AP perform CAC to ensure a low danger probability during $[t_0, t_0 + h]$?

Since $B_r < B_l$, the system can remain stable when N_r becomes larger and larger. However, if N_l increases during $[t_0, t_0 + h]$, the system might enter into danger states where the system capacity is exceeded.

To ensure that the system never enters the danger states, AP must reserve B_l bandwidth for all flows. In the worst case where N_l is zero, a half of the bandwidth is wasted. To ensure that no waste of bandwidth even when N_l is zero, AP must admit a number of sessions which is twice the capacity limit. In worst case, if all N_r remote flows change to local flows, everyone will suffer 50% packet losses. Apparently, there is a trade-off between system bandwidth efficiency and voice quality insurance. A good call admission control scheme should balance well between them.

C. CAC based on Upper bound of Admitted Number

The key assumption in this scheme is that the system is already in a steady state where the numbers of remote flows and local flows are rather constant. Then from Figure A.2, we can derive the steady state probabilities for each state:

$$\pi_r = \frac{\lambda_r}{\lambda_r + \lambda_l}$$
$$\pi_l = \frac{\lambda_l}{\lambda_r + \lambda_l},$$

where π_r and π_l denote the steady state probability for remote state and local state respectively. For a given N, there would be $N_R = N\dfrac{\lambda_r}{\lambda_r + \lambda_l}$ remote flows and $N_L = N\dfrac{\lambda_l}{\lambda_r + \lambda_l}$ local flows in the system. As long as $N_L B_l + N_R B_r \le C$, we say the system can support N voice sessions. Therefore, the upper bound of the admitted number of flows is

$$N = \max\{n \left| \frac{n\lambda_l B_l}{\lambda_r + \lambda_l} + \frac{n\lambda_r B_r}{\lambda_r + \lambda_l} < C \right.\}$$

AP admits a request only if the number of flows in the system is less than the upper bound N.

D. CAC with Bandwidth Reservation

Every time receiving a new request, AP predicts two parameters if it admits the request: the bandwidth B_{RV} that should be reserved for remote flows and the residual bandwidth B_{AV}. $B_{AV} = C - N_r B_r - (N_l + 1)B_l$. If $B_{AV} > B_{RV}$, AP admits the request; otherwise, AP refuses the request.

Now the problem is how to predict B_{RV}. Assume that during $[t_0, t_0 + h]$, each flow makes state transition at most once. Then during $[t_0, t_0 + h]$, the expected number of local-to-remote transitions is $(N_l + 1)\dfrac{\lambda_l}{\lambda_l + \lambda_r}$ while the expected number of reverse transitions is $N_r \dfrac{\lambda_r}{\lambda_l + \lambda_r}$. Overall, there are $\max\left(0, \dfrac{N_r\lambda_r - (N_l + 1)\lambda_l}{\lambda_l + \lambda_r}\right)$ more local flows in the system. Since a local flow takes $B_l - B_r$ more bandwidth than a remote flow, it is necessary to reserve $B_{RV} = \left\lceil \max\left(0, \dfrac{N_r\lambda_r - (N_l + 1)\lambda_l}{\lambda_l + \lambda_r}\right)\right\rceil (B_l - B_r)$ bandwidth for those remote flows that might become local flows.

E. CAC based on Danger Probability

This CAC scheme is based on the predicted danger probability ε and the risk

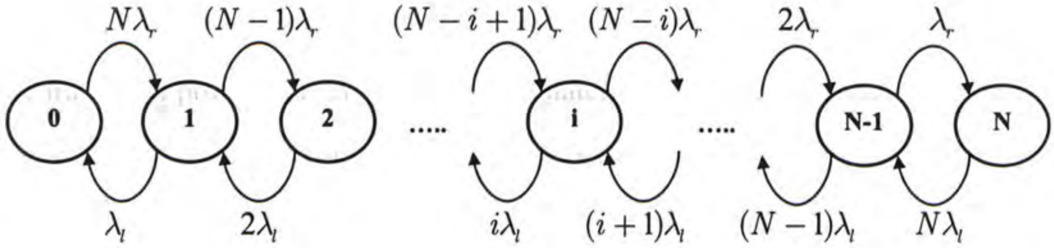


Figure A.3: The state diagram of the number of local flows, N_l

requirement γ. Upon receiving a new request, AP predicts ε for the system with $N+1$ flows. If $\varepsilon \leq \gamma$, AP allows the new session to come in; otherwise, AP turns the request down. The key is how to get ε.

The transition process of N_l can be formulated as a continuous-time Markov chain with the finite state space $\{1, 2, ..., N\}$. Figure A.3 gives the state diagram of N_l.

As i increases, the system might enter the danger state. Define danger state e as the state satisfying $e = \min\left\{i \middle| iB_l + (N-i)B_r > C\right\}$. Then, at time t_0, the transition probability from the current state s to the danger state e in time h is $p_{s,e}(h)$. After knowing all the transition rates, as shown in Figure A.3, $p_{s,e}(h)$ can be calculated using mathematical tools. $p_{s,e}(h)$ reflects the risk of the system being overloaded. Let $\varepsilon = p_{s,e}(h)$. By comparing ε with γ, the admission decision can be made as per the risk requirement.

Larger h implies larger danger probability and stricter requirements. h can be interpreted as the insurance time. Since the average duration of a call is μ, the number of flows in the system is expected to decrease after a period of $\frac{\mu}{N}$. Once the number decreases, the danger probability drops sharply. Therefore, a reasonable value of h is $h = \frac{\mu}{N}$.

F. Implementation Issues

In the proposed CAC schemes, we assume that the transition rates λ_r and λ_l are known. In practice, these parameters can be roughly estimated from past data and statistics. By recording the durations of all flows that stay at the remote state and local state, AP can do such estimation.

References

[1] E. Haghani, S. De, and N. Ansari, *On modeling VoIP traffic in broadband networks*, Proc. IEEE Globecom, Nov. 2007.

[2] S. Garg, M. Kappes, *An experimental study of throughput for UDP and VoIP traffic in IEEE 802.11b networks*, *Proceedings of IEEE WCNC'03*, vol. 3, pp. 1748-1753, Mar. 2003.

[3] F. Anjum et al, *Voice performance in WLAN networks – an experimental study*, Proceedings of IEEE Globecom'03, vol. 6, pp. 3504-3508, Dec. 2003.

[4] S. Garg and M. Kappes, *Can I add a VoIP call?*, IEEE Int. Conf. on Communications 2003.

[5] D. P. Hole, F. A. Tobagi, *Capacity of an IEEE 802.11b wireless LAN supporting VoIP*, Proceeding of the IEEE ICC'04, vol. 1, pp. 196-201, Jun. 2004.

[6] P. Gupta, P. R. Kumar, *The Capacity of Wireless Network*, IEEE Trans. on Information Theory, Vol. 46, No. 2, pp.388-404, Mar. 2000.

[7] Ashish Agarwal and P. R. Kumar, *Capacity bounds for ad-hoc and hybrid wireless networks*. ACM SIGCOMM Computer Communications Review, Special Issue on Science of Networking Design, pp. 71-81, Vol. 34, No. 3, July. 2004.

[8] L. Cai, X. Shen, J. W. Mark, L. Cai, Y. Xian, *Voice capacity analysis of WLAN with imbalanced traffic*, Proceedings of *Qshine'05*, Aug. 2005.

[9] W. Wang, S. C. Liew, Q. X. Pang, and V. O.K. Li, *A multiplex- multicast scheme that improves system capacity of Voice-over-IP on wireless LAN by*

100%, The Ninth IEEE Symposium on Computers and Communications, Jun. 2004.

[10] W. Wang, S. C. Liew and V. O. K. Li, *Solutions to performance problems in VoIP over 802.11 wireless LAN*, IEEE Trans. on Vehicular Technology, Jan. 2005.

[11] A. Chan and S. C. Liew, *VoIP Capacity over Multiple WLANs*, IEEE International Conference on Communications, Jun. 2007.

[12] W. H. Chiang, W. C. Xiao, and C. F. Chou, *A Performance Study of VoIP Applications: MSN vs. Skype*, in Multicomm, 2006.

[13] B. Sat and B. W. Wah, *Analysis and Evaluation of the Skype and Google-Talk VoIP System,* IEEE International Conference on Multimedia, 2006.

[14] A. Barberis, C. Casetti, J.C. De Martin and M. Meo, *A simulation study of adaptive voice communications on IP networks*, Computer Communications, pp 757-767, 2001.

[15] S Shenker, *Fundamental Design Issues for the Future Internet*, IEEE Journal on selected areas in communications, 1995.

[16] C. Mahlo, C. Hoene, A. Rostami, and A. Wolisz, *Adaptive Coding and Packet Rates for TCP-Friendly VoIP Flows*, in Proc. of International Symposium on Telecommunications, Shiraz, Iran, Sep. 2005.

[17] C. Hoene, H. Karl and A. Wolisz, *A Perceptual Quality Model Intended for Adaptive VoIP Applications*, International Journal of Communication Systems, pp.1-20, 2005.

[18] F. Beritelli, S. Casale, and G. Ruggeri, *Performance comparison between VBR speech coders for adaptive VoIP applications*, IEEE Commun. Lett., vol. 5, pp. 423-425, Oct. 2001.

[19] S. A., Baset, H. Schulzrinne, *An Analysis of the Skype Peer-to-Peer Internet*

Telephony Protocol, IEEE INFOCOM'06, Apr. 2006.

[20] S. Guha, N. Daswani and R. Jain, *An Experimental Study of the Skype Peer-to-Peer VoIP System*, 5th International Workshop on Peer-to-Peer Systems, Feb. 2006.

[21] K. Suh, D. R. Figuieredo, J. Kurose, D. Towsley, *Characterizing and detecting relayed traffic: A case study using Skype*, IEEE INFOCOM'06, Apr. 2006.

[22] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi and P. Tofanelli, *Revealing Skype Traffic: when randomness plays with you*, ACM Sigcomm, Aug. 2007.

[23] X. Y. Wang, S. Chen, and S. Jajodia, *Tracking Anonymous Peer-to-Peer VoIP Calls on the Internet*, In Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS), 2005.

[24] T. Berson, *Skype Security Evaluation*, online report, http://www.intelligentzia.ch/inforum /skype_security_evaluation.pdf.

[25] D. Bonfiglio, M. Mellia, M. Meo, N. Ritacca and D. Rossi, *Tracking Down Skype Traffic*, IEEE INFOCOM'06, Apr. 2006.

[26] L. D. Cicco, S. Mascolo and V. Palmisano, *An Experimental Investigating of the Congestion Control Used by Skype VOIP*, WWIC, May. 2007.

[27] J. Lindblom, *A Sinusoidal Voice Over Packet Coder Tailored for the Frame-Erasure Channel*, IEEE trans. Speech Audio Process., vol.13, no.9, pp.1063-6676, 2005.

[28] S. Grag and M. Kappes, *Admission Control for VoIP Traffic in IEEE 802.11 Networks*, IEEE Globecom'03, Dec. 2003.

[29] L. Zhang and S. Zeadally, *HARMONICA: Enhanced QoS Support with Admission Control for IEEE 802.11 Contention-based Access*, Proc. IEEE RTAS, May. 2004.

[30] Hongqiang Zhai, Xiang Chen and Yuguang Fang, *A call admission and rate*

control scheme for multimedia support over IEEE 802.11 wireless LANs, Wireless Networks, July. 2006.

[31] *Introduction to IEEE 802.11*, online tutorial, http://www.intelligraphics.com/introduction -ieee-80211.

[32] M. S. Gast, *802.11 Wireless Networks: Definitive Guide*, O'Reilly, Sebastopol, CA, 2002.

[33] M. Veeraraghavan, N. Cocker and T. Moors, *Support of voice services in IEEE 802.11 wireless LANs*, in Proceedings of INFOCOM'01, 2001.

[34] IEEE Standard for Information Technology - Telecommunications and information exchange between systems - Local and Metropolitan networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher Speed Physical Layer (PHY) Extension in the 2.4 GHz band.

[35] A.P. Markopoulou, F.A. Tobagi and M.J. Karam, *Assessment of VoIP Quality over Internet Backbones*, INFOCOM'02, 2002.

[36] ITU-T G.114, One-way transmission time, 1996.

[37] ITU-T G.1010, End-user multimedia QoS categories, 2001.

[38] Skype, WIKIPEDIA, http://en.wikipedia.org/wiki/Skype.

[39] Skype, http://www.skype.com.

[40] Andrew Sheppard, *Skype Hacks: Tips & Tools for Cheap, Fun, Innovative Phone Service*, O'Reilly, pp 133-138, 2006.

[41] T. Hosfeld, A. Binzenhfer, M. Fiedler, and K. Tutschku, *Measurement and Analysis of Skype VoIP Traffic in 3G UMTS systems*, Research Report No. 377, University of Wrzburg, 2005.

[42] T. Hosfeld, A. Binzenhfer, *Analysis of Skype VoIP traffic in UMTS: End-to-end QoS and QoE measurements*. Computer Networks, 2007.

[43] RFC 3550 - RTP: A Transport Protocol for Real-Time Applications.

[44] The Network Simulator – ns2, http://www.isi.edu/nsnam/ns/.

[45] NS2 source code for Skype-emulating Traffic Generator, and Skype voice traces collected from real network experiments, http://www.wireless.ie.cuhk.edu.hk.