整合 SIP 與 IEEE 802.11e 支援換手及提供多重品質等級的 VoIP 應用

Integrating SIP and IEEE 802.11e to Support Handoff and Multi-grade QoS for VoIP Applications

研 究 生：李　凌　　　　Student：Ling Lee

指導教授：曾煜棋　　　　Advisor：Yu-Chee Tseng

國 立 交 通 大 學
資 訊 科 學 與 工 程 研 究 所
碩 士 論 文

A Thesis
Submitted to Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer Science

June 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年六月

Integrating SIP and IEEE 802.11e to Support Handoff and Multi-grade QoS for VoIP Applications

Student: Ling Lee
Advisor: Prof. Yu-Chee Tseng

A Thesis
Submitted to Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer Science

July 2006

Hsinchu, Taiwan

整合SIP與IEEE 802.11e支援換手及提供多重品質等級的VoIP應用

學生：李湊　　　　　　　　　　　　　　　　　　　　指導教授：曾煜棋 教授

國立交通大學資訊科學與工程學系（研究所）碩士班

摘　　　要

隨著無線網路技術蓬勃發展，在無線網路環境下支援網路電話的需求也隨之成長，當我們希望在有限頻寬下支援越多通話的同時，如何保障網路電話的服務品質成為一項重要的需求。在本論文中我們探討在IEEE 802.11e無線區域網路下換手及網路電話服務品質的議題。同時在本文中我們假設網路電話可支援多種品質等級，我們展示了如何藉由整合通話初始協定以及IEEE 802.11e下的服務品質機制來達成通話允許控制及換手的目標。我們所提出的方法依據網路的狀況對目前進行的通話做動態資源配置。其中多種品質等級是藉由改變編碼方式跟封裝的時間間隔(PI)來達成。結果顯示我們提出的方法有較好的頻寬使用率，降低對新進通話的阻塞率，同時也能減少換手後的通話被丟棄的機率。除此之外，我們也提出了當使用者離開無線網路基地台時，如何將原本所佔用的資源快速回收利用的方法。最後我們藉由分析及模擬的結果來驗證我們提出方法的成效。

關鍵字：IEEE 802.11e，服務品質，通話允許控制，網際網路電話，換手，無線網路，通話初始協定

Integrating SIP and IEEE 802.11e to Support Handoff and Multi-grade QoS for VoIP Applications

Student: Ling Lee Advisors: Prof. Yu-Chee Tseng

Institute of Computer Science and Engineering
College of Computer Science
National Chiao-Tung University

ABSTRACT

With the increasing popularity of wireless networks and the growing demand for VoIP services, there is a need to guarantee QoS for VoIP calls while supporting as many calls as possible. This paper considers the handoff and QoS issues of VoIP calls under IEEE 802.11e WLANs. Assuming that VoIP calls can be supported by multiple levels of QoS, we show how to conduct call admission control and handoff by integrating SIP and QoS mechanisms of IEEE 802.11e. The proposed scheme is designed to dynamically adjust the resource distribution among existing calls according to the network condition. Multi-level QoS is achieved by adjusting codecs and packetization intervals of calls. The result shows better utilization of bandwidth, decreased blocking rate for new calls, and less dropping rate for handoff calls. In addition, we also show how to achieve early resumption of resources as calls leave a QAP. Both analytical and simulation results are presented to evaluate the performance of the proposed schemes.

Keywords: IEEE 802.11e, quality of service (QoS), call admission control (CAC), voice over IP (VoIP), handoff, wireless network, Session Initiation Protocol (SIP).

Acknowledgments

I want to thank many important people. Because of them, there are possibilities for me to obtain much unexpected happiness in my life. My first note of thanks goes to my advisor, Prof. Yu-Chee Tseng. With the wonderful research conditions he provided and his attentive instructions, I came to discover the pleasure of research. I am also grateful to my senior, Jen-Jee Chen. Without his help and suggestions, I would not be able to have this thesis done. During those days at National Chiao-Tung University, I enjoy the moments studying with all HSCC members. Learning with them benefited me in many ways. Finally, I dedicate this thesis to my family and Wei-Kai for their supports and encouragements.

Ling at CS, NCTU.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As wireless networks become more mature, a lot of research efforts have been dedicated to wireless Voice over IP (*VoIP*) techniques. In particular, VoIP over WLAN is believed to be one of the most important Internet applications to offer a viable alternative to traditional phone services. Before this happens, the critical QoS (*Quality of Service*) issues on WLAN have to be addressed. The *IEEE 802.11 Task Group E* (*802.11e*) [1] has been formed to expand the current 802.11 MAC protocol to support applications with QoS requirements. However, neither call admission control nor resource management algorithm is specified for VoIP applications.

In addition, user mobility is another important issue for wireless VoIP over WLAN services. Due to WLAN's limited coverage, handoff between APs is sometimes inevitable. Failure to reserve sufficient bandwidths for handoff calls may force them being dropped. Dropping ongoing calls has a very negative impact from users' perspective. While resource reservation for handoff calls is necessary, maintaining fairness among handoff calls and existing calls within a WLAN is also important. All these issues are new challenges to WLAN.

In recent years, there has been increasing interest in supporting QoS for multimedia applications. Reference [2] proposes a cross-layer protocol to facilitate VoIP traffics over IEEE 802.11e WLANs. It shows how to increase the number of VoIP sessions that can be supported under a QAP with compromising QoS and how to improve the MAC mechanism of IEEE 802.11e to facilitate the transmission of VoIP traffics. However, it only addresses the call establishment issue; the handoff issue is not discussed.

The number of concurrent VoIP sessions that can be supported in a WLAN is evaluated in [3]. It is reported that besides the bandwidth limitation of the physical layer, the codec, *packetization interval (PI)*, and delay budget may influence the number of VoIP sessions that can be supported. It is further shown that PI has more impact than other factors.

Currently, there is no systematical method for changing PIs and codecs of VoIP calls to improve the overall capacity of a WLAN.

Typical multimedia applications can tolerate some degree of temporary bandwidth fluctuation with little or no perceived degradation in quality by using rate-adaptive codec or hierarchical encoding. Based on this characteristic, several resource management algorithms have been proposed. Reference [4] proposes an adaptive framework for provisioning QoS in multimedia wireless cellular networks by combining call admission control and bandwidth adaption under a *QoS-Adaptive Multimedia Service (QoS-AMS)* framework. The objective is to reduce new call blocking and handoff call dropping probabilities. Reference [5] also proposes an adaptive bandwidth allocation mechanism with degradable QoS. To increase bandwidth utilization, the system can free some bandwidth for new users by lowering the QoS levels of existing users. Nevertheless, these works do not give the details to achieve bandwidth adjustment.

This paper focuses on the QoS and call admission control mechanisms for handoff calls over IEEE 802.11e WLANs. We address this issue especially for VoIP applications. When handoff calls are accepted to a QAP, they may increase the competition among QSTAs and thus reduce the bandwidth shares of existing calls. Since VoIP calls can be supported by different codecs and PIs, we propose mechanisms for IEEE 802.11e QAPs to degrade the QoS levels (reflected by their codecs and PIs) of some existing calls as resources are too stringent. Normally, the choice of PIs is more sensitive than the choice of codecs. So enlarging the PI of a call has more impact on saving bandwidth, but at the cost of increased end-to-end latency for voice packets. (Note that the payloads seen by applications will not be affected by PIs.) The ITU-T recommends of a bound of 150 ms for one-way end-to-end delay [6]. This bound will be followed in our design. Accordingly, we also propose resource degrade and upgrade algorithms as calls enter and leave a QAP, respectively. In particular, the resource upgrade algorithm is to support early resource resumption as some calls terminate or handover out of a QAP.

The rest of this paper is organized as follows. Some preliminaries are given in Chapter 2. The proposed QoS handoff mechanisms are introduced in Chapter 3. Chapter 4 and 5 present our analysis and simulation results, respectively. Finally, conclusions are drawn in Chapter 6.

Chapter 2

Preliminaries

2.1 IEEE 802.11e MAC Protocol

The IEEE 802.11e Working Group is currently defining a supplement to the existing legacy 802.11 MAC sublayer to support QoS. It introduces a new *HCF* (*Hybrid Coordination Function*), which includes two access mechanisms, *EDCA* (*Enhanced Distributed Channel Access*) and *HCCA* (*HCF Controlled Channel Access*), corresponding to the existing *DCF* and *PCF*, respectively, in 802.11. Two new features, *AC* (*Access Category*) and *TXOP* (*Transmission Opportunity*), are introduced in HCF. A TXOP is a bounded time interval during which a QSTA (*Quality of Service Station*) can hold the medium and transmit multiple frames consecutively with SIFS spacing. A station can obtain a TXOP by either contention or scheduled access assigned by polling messages.

EDCA of IEEE 802.11e

To differentiate services, the eight user priorities in 802.1D are mapped to four IEEE 802.11e ACs. Each AC has its own transmit queue with an independent EDCA function to contend the medium. These four ACs are background (AC_BK), best effort (AC_BE), video (AC_VI), and voice (AC_VO), and are prioritized by different AIFS and contention window sizes. If a collision occurs among ACs within a QSTA, the highest priority AC wins the contention and the other AC(s) will backoff. The EDCA_Parameter_Set information elements (Fig. 2.1), which are sent in beacon frames, specify the parameters of ACs.

Admission Control in EDCA

IEEE 802.11e allows a QSTA to request to add a new traffic stream by sending an *ADDTS Request* to its QAP. The information carried in an *ADDTS Request* includes the

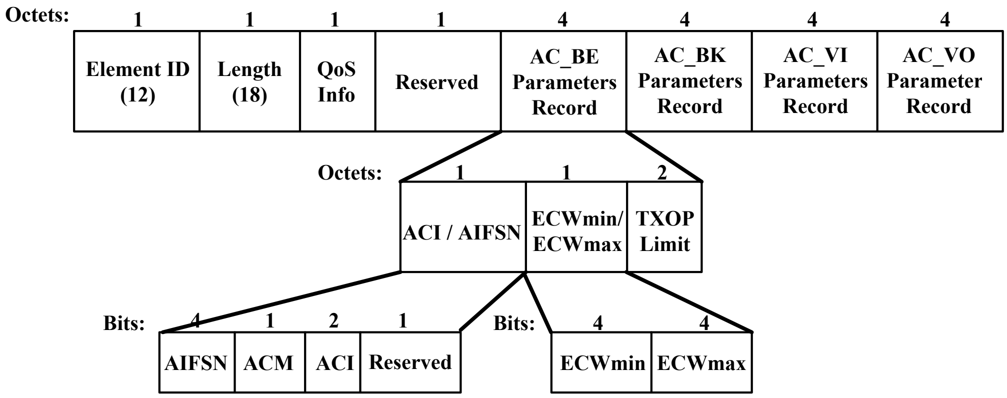


Figure 2.1: Structure of the IEEE 802.11e EDCA_Parameter_Set information element.

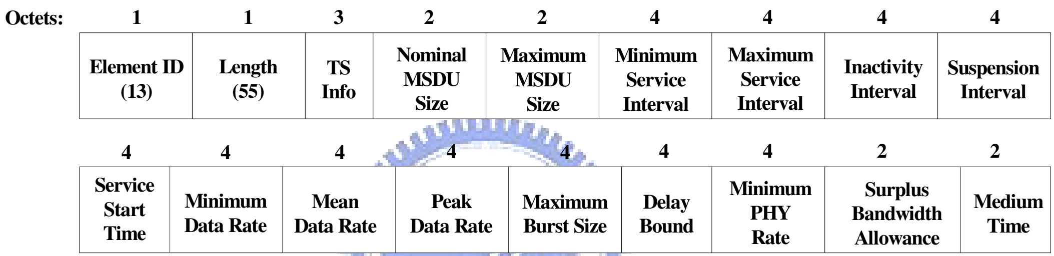


Figure 2.2: Structure of the TSPEC information element.

direction of the stream and a TSPEC (*Traffic Specification*) information element (Fig. 2.2). Admission control is conducted by the QAP by calculating the needed MT (*Medium Time*) as opposed to its remaining MT. Then, an *ADDTS Response* can be replied.

2.2 SIP and SDP

SIP (*Session Initiation Protocol*) is a protocol for establishing an IP multimedia session. It's an application-layer control protocol to setup, modify, and terminate multimedia sessions. While SIP is not used to transport media traffic, it often chooses RTP (*Real-time Transport Protocol*) as its transportation protocol and uses SDP (*Session Description Protocol*) [7] to specify its session characteristics. Fig. 2.3 shows an example of call setup and tear-down in SIP. When a caller wants to make a VoIP call with a callee, it sends an INVITE including the codecs that the caller supports in a SDP message body. Fig. 2.4(a) shows an example, where G.726 (format 2), and G.723 (format 4) are the offered codecs, with 4400 as its receiving port. If the callee decides to accept the request, it replies a

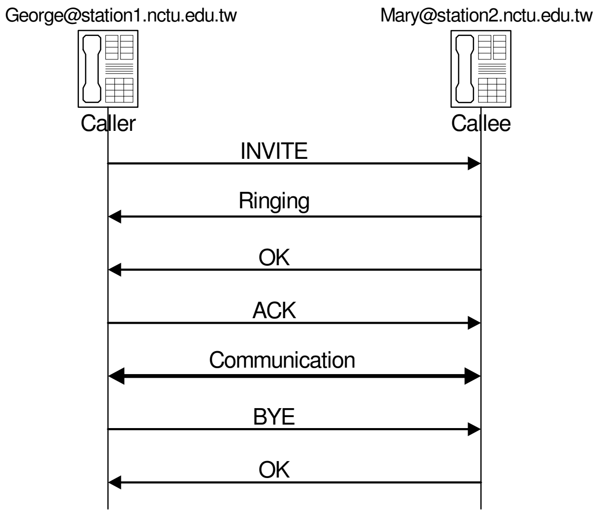


Figure 2.3: An example of SIP call setup and tear-down.

Ringing and an OK signal with the selected codec. In Fig. 2.4(b), the selected codec is G.723, and the receiving port is 888. If the port number is 0, it means a rejection.

```
INVITE sip: Mary@station2.nctu.edu.tw SIP/2.0
From: Caller <sip: George@station1.nctu.edu.tw>; tag=abc123
To: Callee <sip: Mary@station2.nctu.edu.tw >
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Disposition: session

v=0
o= George 123 001 IN IP4 station1.nctu.edu.tw
s=
c=IN IP4 station1.nctu.edu.tw
t=0 0
m=audio 4400 RTP/AVP 2 4
a=rtpmap 2 G726-32/8000
a=rtpmap 4 G723/8000
```

(a)

```
SIP/2.0 200 OK
From: Caller <sip: George@station1.nctu.edu.tw>; tag=abc123
To: Callee <sip: Mary@station2.nctu.edu.tw >
CSeq: 1 INVITE
Content-Type: application/sdp
Content-Disposition: session

v=0
o= callee 456 001 IN IP4 station2.nctu.edu.tw
s=
c=IN IP4 station2.nctu.edu.tw
t=0 0
m=audio 888 RTP/AVP 4
a=rtpmap 4 G723/8000
```

(b)

Figure 2.4: An example of SIP with SDP message bodies: (a) INVITE siganl and (b) OK signal.

Chapter 3

The Proposed QoS Mechanisms

We consider an IEEE 802.11e wireless network operating in the infrastructure mode to support VoIP applications. Although IEEE 802.11e supports QoS, the resource reservation and handoff handling issues for particular applications are left open to designers. In this work, we consider using SIP signaling to support VoIP over IEEE 802.11e networks.

We propose two mechanisms to solve the handover problem. The first one is a Call Admission Control (CAC) algorithm, which is performed whenever a new call or a handoff call arrives at a QAP. The CAC algorithm accepts or rejects an arriving call according to the amount of available resources versus the QoS requirements of the call. The second mechanism is a Resource Adjustment (RA) algorithm, whose purpose is to dynamically change bandwidth allocation among on-going calls in a QAP for better resource utilization and fairness. This is feasible because multimedia services can typically operate under different bandwidths.

In order to dynamically adjust resource allocation, we assume that VoIP calls can be supported by multiple levels of QoS. Each QoS level corresponds to a voice codec and a packetization interval (PI), where PI is the period that voice data is encapsulated into packets for transmission. For most current systems, the default PI is 20 ms. Larger PIs would introduce less header overhead, but may suffer from higher delays and are more sensitive to packet loss. Given a PI and a data generation rate of λ, the amount of data to be transmitted per PI is $(\lambda \times PI + h)$, where h is the header overhead. Therefore, the bandwidth required per time unit is $(\lambda \times PI + h)/PI = \lambda + h/PI$. Clearly, a larger PI will incur less traffic. For example, if we use G.726 at rate 32 kbps, with a header size of 74 bytes and a PI of 10 ms, the required bandwidth is 32 kbps+74 bytes/10 ms=91.2 kbps. However, with a PI of 20 ms, the required bandwidth reduces to 32 kbps+74 bytes/20 ms=61.6 kbps.

Suppose that for each codec there are k QoS levels and each QoS level corresponds to a PI. Note when a call changes from a PI to another PI', the difference of resource usage is $(\lambda - h/PI') - (\lambda - h/PI) = h(1/PI' - 1/PI)$. This value is only dependent of PI and PI', but is independent of λ (and thus independent of which codec is used). Therefore, the system state of a QAP can be denoted as $\overline{s} = (s_1, s_2, s_3, ..., s_k)$, where s_i is the number of calls served at the ith level (these calls may use different codecs). The following terminologies will be used in our CAC and RA algorithms.

- B_{total}: the total bandwidth of a QAP.
- B_{th}: the threshold of occupied bandwidth (below which new calls can be accepted).
- B_{free}: the current free bandwidth of the QAP.
- B_{deg}: the maximum available free bandwidth of the QAP after degrading all existing calls to the lowest QoS level.
- PI_{def}: the default PI for all new calls.
- W_{alloc}: the bandwidth allocated to the request call.

3.1 The Call Admission Control Algorithm

The CAC algorithm is to determine whether a new call or a handoff call can be accepted depending on the available bandwidth in the network. Fig. 3.1 shows the proposed CAC procedure. The caller under QAP1 first establishes a VoIP call with the callee by a SIP INVITE signal containing necessary codec and PI information. QAP1, on receiving this INVITE signal, will estimate its current available resource and possibly filter out some codecs that it cannot support due to bandwidth constraints (refer to box A in Fig. 3.1). If the callee can successfully choose a codec, QAP1 will reserve sufficient resources for the call (refer to boxes C, D and, E in Fig. 3.1). Then, the voice communication can be started. When the caller is moving away from QAP1, due to degraded signal quality, the caller will actively look for the next serving QAP by sending IEEE 802.11 *Probe Requests*. When a QAP receives a *Probe Request*, it will estimate whether it has enough available bandwidth and reply a proper IEEE 802.11 *Probe Response* (refer to box A in Fig. 3.1). The caller will collect all *Probe Responses* and select a new QAP (refer to box B in Fig. 3.1). Suppose that the caller selects QAP3. It will send QAP3 an IEEE 802.11 *Re-Association Request* to trigger QAP3 to execute resource reservation (refer to box C in Fig. 3.1). In the meanwhile, QAP3 and QAP1 will exchange the caller's context by *Inter Access Point Protocol* (*IAPP* [8]). Via IAPP, QAP1 is informed of the departure of the caller and may execute the RA algorithm. Finally, the caller will exchange IEEE 802.11e *ADDTS Request* and *Response* with QAP3 (refer to boxes D and E in Fig. 3.1) to actually reserve the bandwidth. If these steps go through successfully, the caller and the callee can resume their voice communication. In the following, we will explain the detail actions to be taken in boxes A, B, C, D, and E.

A. Resource Estimation at the QAP

When a new call or a handoff call arrives at a QAP, the QAP will evaluate its available resource and the required resource of the call. In the new call event, the INVITE signal from the caller will carry all codec and PI information to the callee. In the handoff call event, we propose that the caller utilizes its *Probe Requests* to convey the codec and PI information with two new IEEE 802.11 *information elements*: codec and PI. Fig. 3.2 shows the proposed orders of the information elements in *Probe Request* and *Probe Response*.

With these information, the QAP can estimate if it has sufficient resource to serve this call. To compute the required resource, we propose that each QAP keeps a *Packet Size Table (PST)* as in Table 3.1. For example, using G.726 with a sampling rate of 32 kbps and PI=20 ms, each packet is of size 154 bytes (80 bytes of voice payload, 40 bytes of IPv4/UDP/RTP/error-checking overhead, and 34 bytes of MAC/error-checking overhead). Therefore, given a call's codec, PI, and current physical rate r, we can compute the required *Medium Time* (MT):

$$\begin{aligned} MT(codec, PI, r) &= (\text{total\_time\_needed\_per\_BI}) \\ &= (\text{time\_to\_send\_one\_packet}) \times (BI/PI) \\ &\quad \times(\text{surplus\_bandwidth\_allowance}) \\ &= \{[(H_{RTP} + H_{UDP} + H_{IP} + H_{MAC}) + L(c)]/r \\ &\quad +(DIFS + averageCW + PHY\_header) + (SIFS + ACK)\} \\ &\quad \times(BI/PI) \times (\text{surplus\_bandwidth\_allowance}), \end{aligned} \tag{3.1}$$

where $L(c)$ is the payload of codec c per packet, $averageCW$ is the average contention window size seen by the QAP, and H_{RTP}, H_{UDP}, H_{IP}, and H_{MAC} are header sizes of RTP, UDP, IP, and MAC packets, respectively. The basic operation of 802.11e is shown in Fig. 3.3. Note that assuming 802.11b, the ACK and the PHY header must be transmitted at 1 Mbps. For an overview, relevant parameters of 802.11e EDCA are listed in Table 3.2. The surplus_bandwidth_allowance is to take the extra medium access overhead (contention,

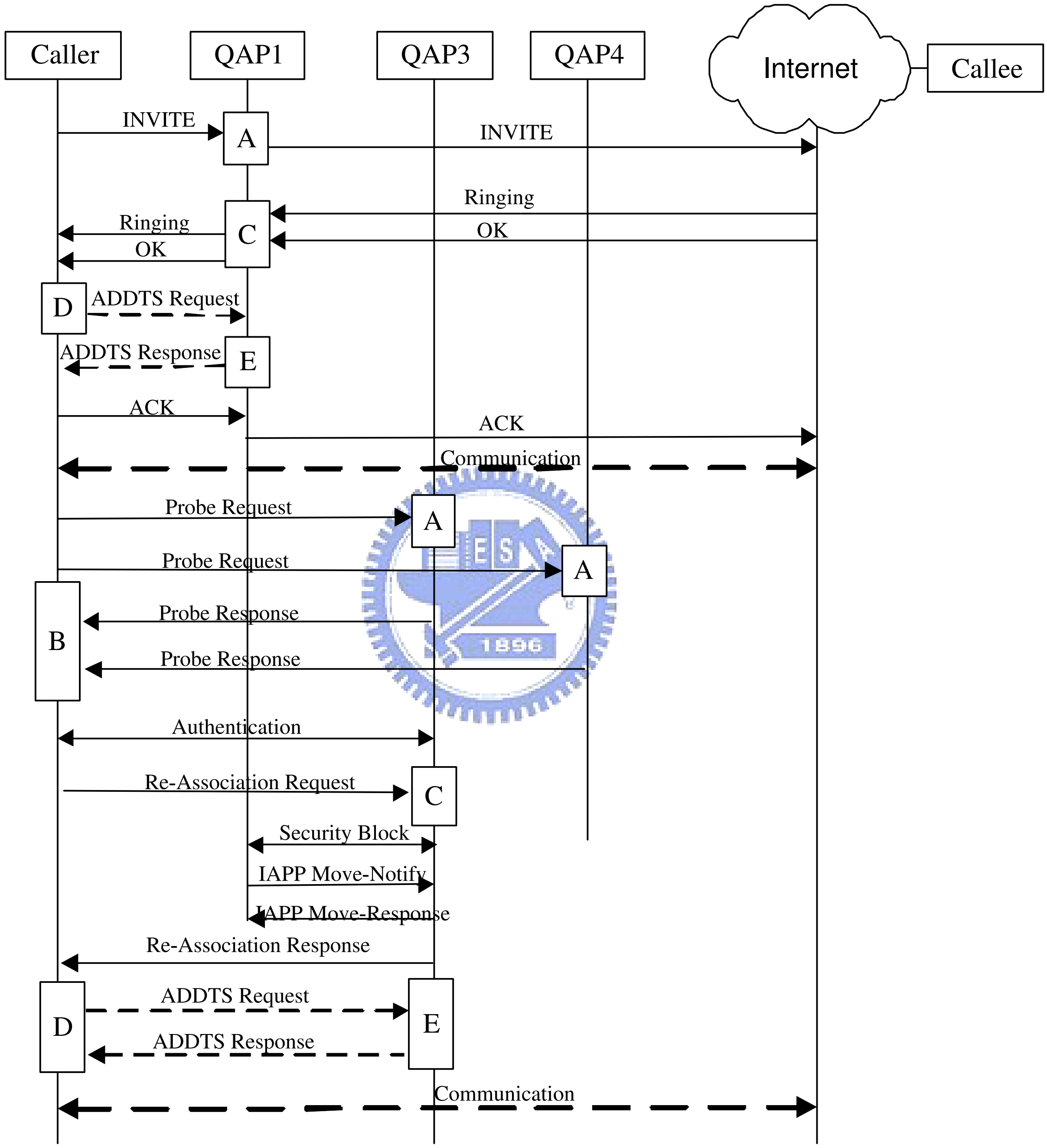


Figure 3.1: The proposed message flows of the CAC procedure in IEEE 802.11e networks.

Order	Information Elements
1	SSID
2	Supported Rates
8	Codec
9	Packetization Interval

(a)

Order	Information Elements
1	Timestamp
2	Beacon Interval
3	Capability Information
...	...
23	QBSS Load
24	EDCA Parameter Set
26	Codec
27	Packetization Interval

(b)

Figure 3.2: The orders of information elements in (a) Probe Request and (b) Probe Response.

Codec	Data Rate (kbps)	Packetization Interval (ms)				
		5	10	20	30	40
G.711	64	113	154	234	314	394
G.726	16	84	94	114	134	154
	32	94	114	154	194	234
G.728	16	84	94	114	134	154
G.723.1	5.3				94	
	6.3				98	

Table 3.1: The Packet Size Table, which contains the packet sizes (in bytes) when different codecs and packetization intervals are used.

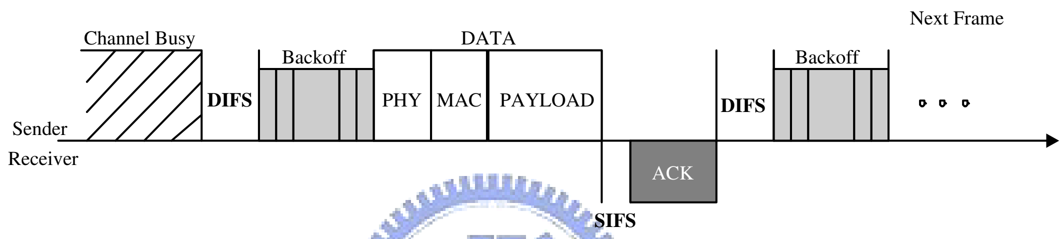


Figure 3.3: Basic operations of 802.11e EDCF.

collision, etc.) into account; in this work, we assume its value to be 1.1. For example, when BI is 1 sec and min_PHY_rate is 11 Mbps, if we use G.726 with rate of 32 kbps and PI of 20 msec, then $MT = [154/11(bytes/Mbps) + (50+70+192) + (10+248)] \times (1000/20) \times 1.1 = 37.51\ ms$.

Each QAP will keep its maximum available free bandwidth B_{deg}, which is equal to B_{free} plus the releasable resource after moving all existing calls to the lowest QoS level. That is, $B_{deg} = B_{free} + \sum_{i=1}^{k-1} h(1/PI_i - 1/PI_k)$. If a codec's required MT is larger than

RTP + UDP + IP header	40 bytes
MAC header for DATA	34 bytes
PHY header	192 μsec
ACK	248 μsec
DIFS	50 μsec
SIFS	10 μsec
Slot Time	20 μsec
CWmin(for voice)	7
CWmax(for voice)	15

Table 3.2: Parameters of IEEE 802.11e EDCA.

the QAP's B_{deg}, the QAP will drop the INVITE or the *Probe Request* silently or reply a SIP response to the caller with a status code of 480, which means "temporarily not available".

B. QAP Selection at the Caller

After scanning all channels, the caller will choose a target QAP based on various criteria, such as signal strength, codec, PI, etc. For example, we may prefer a lighter loaded QAP. Alternatively, we may choose the one with better signal quality. This is outside the scope of this work.

C. Resource Reservation at QAP

First, we will determine the codec c, PI p, and physical rate r to be used by the call. The value of r can be measured from signal quality. In the new call event, the OK signal will contain the value of c, and we will assume $p = PI_{def}$. In the handoff event, the *Re-association Request* will contain the current c and p used by the caller. Then the QAP will decide to accept or reject the call based on the following rules:

- If this is a handoff call, it will be accepted if the requested MT is no more than B_{deg}; otherwise, the call is rejected.
- If this is a new call, there are two cases:
  - If $MT(c, PI_{max}, r) \leq B_{deg}$ and $B_{deg} > (B_{total} - B_{th})$, the call is accepted directly.
  - If $MT(c, PI_{max}, r) \leq B_{deg}$ but $B_{deg} \leq (B_{total} - B_{th})$, the call is accepted with a probability P_r.

Note that the selection of P_r can be based on the *DCRS* (*Dynamic Channel Reservation Scheme*) proposed in [9]. In this work, we will only consider adjusting PI for handoff calls, although adjusting codec is also possible.

If the call cannot be accepted, the QAP will drop the OK silently (for new call) or reply the *Re-Association Response* to the caller with a status code of 37, which means "The request has been declined." (for handoff call). If the call can be accepted, we will check if $MT(c, p, r) \leq B_{free}$. If so, the selected codec and PI will be relayed to the caller via an OK (for new call) or a *Re-association Response* (for handoff call). Otherwise, the current available resource is not able to support the request and we will call function *degrade*(c, p, r) in Fig. 3.4. The function will repeatedly select an existing call to reduce

degrade(c, p, r)

$t\_PI = p$;
while (not all calls are served by PI_{max}) **do**
  let X be the call with the smallest PI in the system;
  in case of tie, the one with the lowest physical rate is selected;
  change X's PI to *next(X.PI)*;
  $B_{free} = B_{free} + MT(X.codec, X.PI, X.rate)$
  $-MT(X.codec, next(X.PI), X.rate)$;
  if ($B_{free} \geq MT(c, t\_PI, r)$) **then**
    return(t_PI);
  else if (there is no call with PI smaller than or equal to t_PI) **then**
    t_PI = *next(t_PI)*;
  end if;
end while;
return(PI_{max});

Figure 3.4: The bandwidth degrade algorithm.

its QoS level. The call with the best QoS level will be degraded first. If there are multiple candidates, the one with the lowest physical rate will be degraded first. Function *next()* will return the next QoS level. This is repeated until sufficient resources are released.

Fig. 3.5 shows an example. Suppose that there are $k = 4$ QoS levels, and the current system state is (3,2,0,2). Also suppose that the required resources for these QoS levels are $(8, 7.5, 7, 6.5)$, $(6, 5.5, 5, 4.5)$, $(4, 3.5, 3, 2.5)$, and $(2, 1.5, 1, 0.5)$, respectively (the four numbers map to four physical rates in an ascending order). The total capacity is equal to 35. So, there is no resource remaining. Suppose that an incoming call requests a QoS level of 2 (physical rate = 1 Mbps). As the resource required is 6, we need to degrade three calls from QoS level 1 to level 2. The next incoming call also requests a QoS level of 2 (physical rate = 1 Mbps). The resource required is 6, too. We need to degrade three of level-2 calls. The calls with lower transmission rates should be degraded first, so we move two calls with 1 Mbps and one with 2 Mbps to level 3. Then the system state will change to $(0, 4, 3, 2)$. The last incoming call requests a QoS level of 2 (physical rate = 5.5 Mbps). According to the algorithm, we move three level-2 calls to level 3. The final system state is $(0, 2, 6, 2)$.

D. ADDTS Request by the Caller

After determining the codec and PI, the caller will send a bidirectional *ADDTS Request* to the QAP by including a TSPEC element to request for resources. We suggest to convey VoIP service requirements by the following fields in TSPEC:

- Minimum_Data_Rate = the acceptable longest packetization interval of the corre-

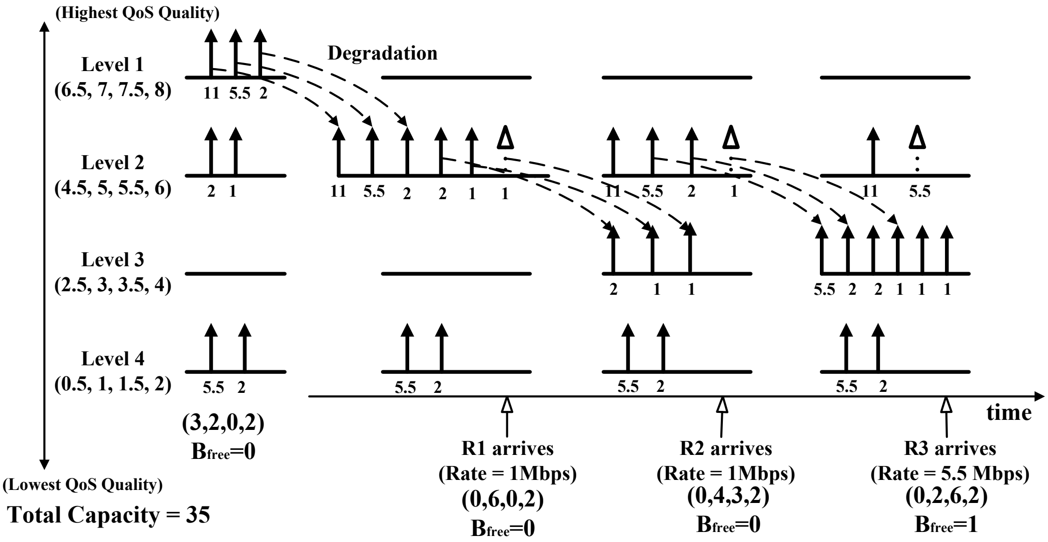


Figure 3.5: An example of bandwidth degrade.

sponding codec.

- Mean_Data_Rate = the packetization interval selected by the callee.
- Maximum_Data_Rate = the acceptable shortest packetization interval.
- Medium_Time = the codec selected by the callee.

E. ADDTS Response by the QAP

According to the caller's *ADDTS Request* and the Packet Size Table, QAP can compute the required medium time following Eq.(3.1). Each QAP keeps the following variables:

- $TXOPBudget[AC_i]$ = The remaining bandwidth that can be allocated to $AC_i, i = 0..3$.
- $TxAdDn[AC_i][TSID]$ = The admitted medium time for stream TSID of AC_i in the downlink direction.
- $TxAdUp[AC_i][TSID]$ = The admitted medium time for stream TSID of AC_i in the uplink direction.
- $TxAdDn[AC_i]$ = This value is set to $\sum_{\forall TSID} TxAdDn[AC_i][TSID]$, to record the overall resource allocated to AC_i in the downlink direction.
- $TxUsedDn[AC_i]$ = The summation of used medium time of all downlink streams of AC_i.

Initially, $TXOPBudget[AC_i]$ contains all the bandwidth (in terms of medium time) that is reserved for AC_i. Whenever a new stream is added, the corresponding resource is subtract from $TXOPBudget[AC_i]$, and the resource is assigned to $TxAdDn[AC_i][TSID]$ and/or $TxAdUp[AC_i][TSID]$. Also, each QSTA should keep the following variables:

- $TxAdUp[AC_i][TSID]$ = The admitted medium time for stream TSID of AC_i in the uplink direction in this QSTA per BI.

- $TxAdUp[AC_i]$ = This value is set to $\sum_{\forall TSID} TxAdUp[AC_i][TSID]$, to record the overall resource allocated to ACi in the uplink direction.

- $TxUsedUp[AC_i]$ = The summation of used medium time of all uplink streams of AC_i.

Resource reservation at QAP is done as follows. First, we compute the value of $TXOPBudget[AC_i] - 2 \times MT(c, p, r)$. If the value is non-negative, there is sufficient resource to support this call and we can set variables as follows:

$$TXOPBudget[AC_i] = TXOPBudget[AC_i] - 2 \times MT(c, p, r);$$
$$TxAdDn[AC_i][TSID] = MT(codec, PI, PHY);$$
$$TxAdUp[AC_i][TSID] = MT(codec, PI, PHY);$$
$$TxAdDn[AC_i] = TxAdDn[AC_i] + TxAdDn[AC_i][TSID];$$

Up to this point, the admitted resources have been guaranteed. The QAP will reply an *ADDTS Response* to the caller with the Mean_Data_Rate=PI and Medium_Time=MT(c,p,r) in TSPEC. If there is no sufficient resource, then an *ADDTS Response* is replied with Medium_Time=0.

At the caller's side, if an *ADDTS response* with a positive Medium_Time is received, the QSTA will set its $TxAdUp[AC_i][TSID]$= Medium_Time, retrieves the PI in the Mean_Data_Rate field, and passes it to the upper layer VoIP application program. Otherwise, the call is considered rejected. In both cases, the caller should reply a response signal with the proper status code to the callee.

3.2 The Resource Adjustment Algorithm

Fairness among existing users and handoff users is an important issue. The goal of resource adjustment is to re-allocate bandwidth to calls for fairness. The RA algorithm may be

Resource_Adjustment()

1: On a call X moving to a lower rate r:
   $B_{free} = B_{free} + MT(X.codec, X.PI, X.rate)$;
   if($B_{free} < MT(X.codec, X.PI, r)$)
      degrade(X.codec,X.PI,r);
2: On a call X leaving:
   $B_{free} = B_{free} + MT(X.codec, X.PI, X.rate)$;
   upgrade();
3: On a call X moving to a higher rate r:
   $B_{free} = B_{free} + MT(X.codec, X.PI, X.rate) - MT(X.codec, X.PI, r)$;
   upgrade();

Figure 3.6: The RA algorithm.

triggered by the following two events: departure of calls and transmission rate change of existing calls (refer to Fig. 3.6). On events that a call moves to a lower rate, the function *degrade(c,p,r)* will be called if there is no sufficient resource. On events that a call departs or moves to a higher rate, the value of B_{free} will be updated, and then the function *upgrade()* in Fig. 3.7 will be invoked. This function will repeatedly select an existing call to upgrade its QoS level. The call with the worst QoS level will be upgraded first. If there are multiple candidates, the one with the highest physical rate will be upgraded first. Function *prev()* will return the previous QoS level. This is repeated until B_{free} is not enough to upgrade any existing call.

Fig. 3.8 shows an example. Suppose that there are $k = 4$ QoS levels, and the current system state is $(4, 1, 1, 3)$. The resource requirement is the same as the example in Fig. 3.5. Let the total capacity be 41. Suppose that a level 1 call leaves the network (physical rate = 1 Mbps), releasing a bandwidth of 2. The released bandwidth can upgrade the call at QoS level 4 with rate 11 Mbps to level 3. The system state after upgrade is $(4, 1, 2, 1)$. Next, a level-2 call with rate 2Mbps leaves, releasing a bandwidth of 5.5. This can upgrade the only call at level 4 to level 3, resulting a system state of $(4, 1, 2, 0)$.

upgrade()

while (TRUE) **do**

  let X be the call with the largest PI in the system;
  in case of tie, the one with the highest physical rate is selected;

  if $B_{free} \geq MT(X.codec, prev(X.PI), X.rate) - MT(M.codec, X.PI, X.rate)$ **then**

    change X's PI to *prev(X.PI)*;

    $B_{free} = B_{free} - MT(X.codec, prev(X.PI), X.rate) + MT(M.codec, X.PI, X.rate)$;

  else

    return;

  end if;

end while;

Figure 3.7: The bandwidth upgrade algorithm.

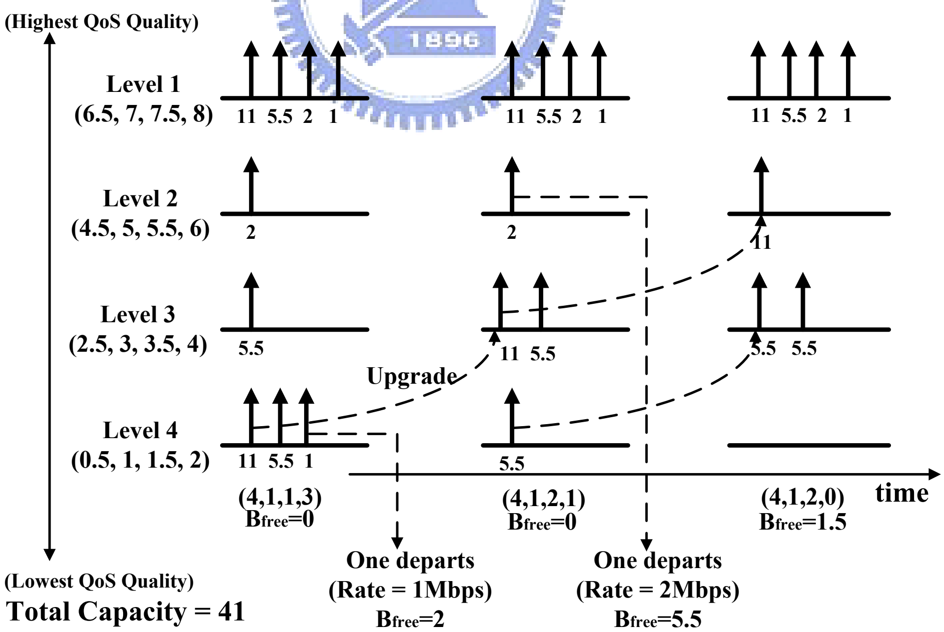


Figure 3.8: An example of bandwidth upgrade.

Chapter 4

Analysis

In this section, we derive an analytical model to evaluate the performance of our QoS mechanisms. Our goal is to analyze the blocking probability of new calls, the dropping probability of handoff calls, and the call dropping probability due to change of transmission rates. Without loss of generality, we assume all calls use the same G.726 codec at the default rate 32 kbps. Thus, during a degrade or upgrade process, calls will only change their PIs, but not codec. Suppose that there are m PIs, $PI_1, PI_2, ..., PI_m$ (in an ascending order), and y transmission rates, $R_1, R_2, ..., R_y$ (in a descending order).

Due to mobility, the rate change of a QSTA is modeled by the state diagram in Figure 4.1. From each state, a QSTA can transit to a higher or a lower rate with a rate ν following a Poisson distribution. In each QAP, new and handoff calls arrive by Poisson distributions with rates $y \cdot \lambda_n$ and $y \cdot \lambda_h$, respectively. These rates are evenly distributed to calls of all physical rates $R_1, R_2, ..., R_y$. Call holding time and cell residence time are exponentially distributed with means of $1/\mu_h$ and $1/\mu_r$, respectively. Thus, the channel occupancy time of a call is exponentially distributed with mean $1/\mu = 1/(\mu_h + \mu_r)$. The required bandwidth of a call with PI_{max} at the transmission rate R_i is denoted by Φ_i.

According to our CAC algorithm, computation of accepting or rejecting a call is all based on the assumption that all calls can be degraded to the lowest QoS level. In other words, a QAP drops or blocks a call when all existing calls use PI_{max} and the sum of their used bandwidth meets some conditions. Therefore, to obtain blocking and dropping probabilities, we can assume that all calls use PI_{max}.

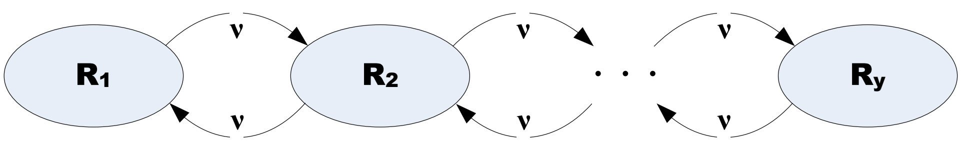


Figure 4.1: The state transition diagram of a QSTA's rate change.

Transmission rate (Mbps)	11	5.5	2	1
Occupied MT (sec)/per session	0.041	0.050	0.083	0.134

Table 4.1: The required MT of a bi-directional voice call under different physical rates under our analytical model.

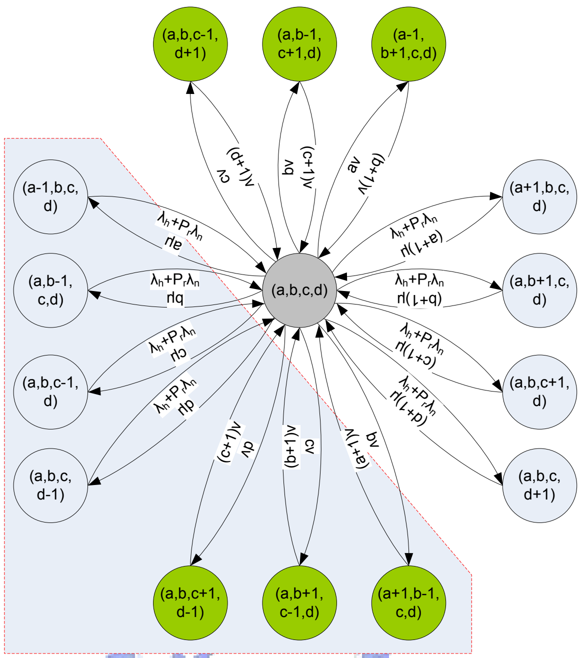


Figure 4.2: Generic state transitions under our analytical model.

For simplicity, we assume that a QAP can support $y = 4$ physical rates, 11, 5.5, 2 and 1 Mbps. For a bi-direction voice stream, assuming $BI = 1sec$, $surplus = 1.1$, and $PI_{max} = 40ms$, the required MT per BI of a call under each rate is listed in Table 4.1.

Our system can be modeled by a $y = 4$ dimensional Markov process. Each system state is written as (n_1, n_2, n_3, n_4), where n_i is the number of calls at rate R_i, $i = 1...4$. For each state (n_1, n_2, n_3, n_4), there 14 possible state transitions, as shown as Figure 4.2, where $n_1 = a$, $n_2 = b$, $n_3 = c$, $n_4 = d$. Horizontal transitions are caused by call arrival or departure events. The arrival rates are all modeled by $\lambda_h + P_r\lambda_n$. The departure rate for rate R_i is $n_i\mu$. For ease of presentation, we let $P_r = 1$ when $n_1\Phi_1 + n_2\Phi_2 + n_3\Phi_3 + n_4\Phi_4 < B_{th}$; otherwise, new calls are accepted with a probability P_r as defined in Sec. 3.1. Vertical transitions are caused by transmission rate change.

A simplified two-dimension Markov process is shown in Figure 4.3 for the case of $y = 2$. The states marked by gray are those with $P_r = 1$, where all new calls can be accepted. Under other states, a new call will be dropped with a fixed probability $(1 - P_r)$.

Based on above state transition diagram, we can derive the steady-state probability

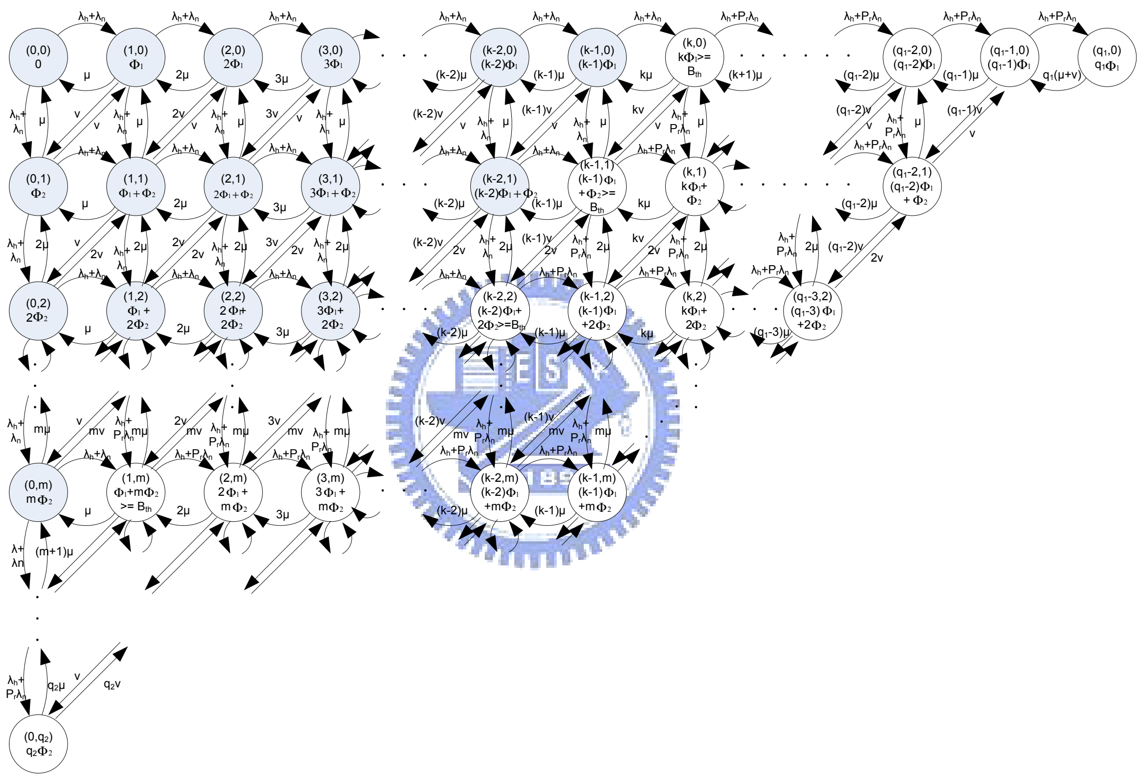


Figure 4.3: A state transition example with $y = 2$ (q_1 and q_2 are the maximum numbers of calls that can be accommodated with PI_{max} at rates R_1 and R_2, respectively).

$P_{n_1,n_2,\ldots,n_y}$ of each state. There are four cases:

Case I: For the state such that $n_1 = n_2 = \ldots = n_y = 0$,

$$y\left(\lambda_h + \lambda_n\right) P_{n_1,n_2,\ldots,n_y} = \mu \sum_{i=1}^{y} P_{n_1,n_2,\ldots,n_i+1,\ldots,n_y}. \tag{4.1}$$

Case II: For states such that $\sum_{i=1}^{y}\left(n_i \Phi_i\right) < B_{th}$,

$$\begin{aligned}
&\left[y\left(\lambda_h + \lambda_n\right) + \left(\sum_{i=1}^{y} n_i\right)\mu + \left(\sum_{i=1}^{y-1} n_i + \sum_{i=2}^{y} n_i\right)\nu\right] P_{n_1,n_2,\ldots,n_y} \\
= &\sum_{i=1}^{y}\left[\left(n_i+1\right)\mu P_{n_1,n_2,\ldots,n_i+1,\ldots,n_y} + \left(\lambda_h + \lambda_n\right) P_{n_1,n_2,\ldots,n_i-1,\ldots,n_y}\right] \\
&+ \sum_{i=2}^{y}\left[\left(n_i+1\right)\nu P_{n_1,n_2,\ldots,n_{i-1}-1,n_i+1,\ldots,n_y}\right] \\
&+ \sum_{i=1}^{y-1}\left[\left(n_i+1\right)\nu P_{n_1,n_2,\ldots,n_i+1,n_{i+1}-1,\ldots,n_y}\right].
\end{aligned} \tag{4.2}$$

Case III: For states such that $\sum_{i=1}^{y}\left(n_i \Phi_i\right) >= B_{th}$ and $\sum_{i=1}^{y}\left(n_i \Phi_i\right) + \Phi_y < B_{total}$,

$$\begin{aligned}
&\left[y\left(\lambda_h + P_r\lambda_n\right) + \left(\sum_{i=1}^{y} n_i\right)\mu + \left(\sum_{i=1}^{y-1} n_i + \sum_{i=2}^{y} n_i\right)\nu\right] P_{n_1,n_2,\ldots,n_y} \\
= &\sum_{i=1}^{y}\left[\left(n_i+1\right)\mu P_{n_1,n_2,\ldots,n_i+1,\ldots,n_y}\right] \\
&+ \sum_{i=1}^{y}\left[\left(\lambda_h + \left(I_i + \bar{I}_i P_r\right)\lambda_n\right) P_{n_1,n_2,\ldots,n_i-1,\ldots,n_y}\right] \\
&+ \sum_{i=2}^{y}\left[\left(n_i+1\right)\nu P_{n_1,n_2,\ldots,n_{i-1}-1,n_i+1,\ldots,n_y}\right] \\
&+ \sum_{i=1}^{y-1}\left[\left(n_i+1\right)\nu P_{n_1,n_2,\ldots,n_i+1,n_{i+1}-1,\ldots,n_y}\right],
\end{aligned} \tag{4.3}$$

where for $z \in \{1, \ldots, y\}$

$$I_z = \begin{cases} 1, & \sum_{i=1}^{y}(n_i\Phi_i) - \Phi_z < B_{th} \\ 0, & \text{otherwise} \end{cases}.$$

Case IV: For states such that $\sum_{i=1}^{y}(n_i\Phi_i) <= B_{total}$ and $\sum_{i=1}^{y}(n_i\Phi_i) - \Phi_y >= B_{th}$,

$$
\begin{aligned}
&\left[\left(\sum_{i=1}^{y} I_i\right)(\lambda_h + P_r\lambda_n) + \left(\sum_{i=1}^{y} n_i\right)\mu + \left(\sum_{i=1}^{y-1} n_i + \sum_{i=2}^{y} n_i\right)\nu\right] P_{n_1,n_2,\ldots,n_y} \\
= \quad & \sum_{i=1}^{y-1}\left[I_i\,(n_i+1)\left(\mu + \bar{I}_{i+1}\nu\right) P_{n_1,n_2,\ldots,n_i+1,\ldots,n_y}\right] \\
&+ I_y\,(n_y+1)\,\mu P_{n_1,n_2,\ldots,n_{y-1},n_y+1} \\
&+ (\lambda_h + P_r\lambda_n)\sum_{i=1}^{y} P_{n_1,n_2,\ldots,n_i-1,\ldots,n_y} \\
&+ \sum_{i=2}^{y}\left[I_{i-1,i}\,(n_i+1)\,\nu P_{n_1,n_2,\ldots,n_{i-1}-1,n_i+1,\ldots,n_y}\right] \\
&+ \sum_{i=1}^{y-1}\left[(n_i+1)\,\nu P_{n1,n_2,\ldots,n_i+1,n_{i+1}-1,\ldots,n_y}\right], \qquad (4.4)
\end{aligned}
$$

where for $z \in \{1,...,y\}$

$$I_z = \begin{cases} 1, & \sum_{i=1}^{y}(n_i\Phi_i) + \Phi_z <= B_{total} \\ 0, & \text{otherwise} \end{cases}$$

and for $(m,n) \in \{(1,2),(2,3),\ldots,(y-1,y)\}$

$$I_{m,n} = \begin{cases} 1, & \sum_{i=1}^{y}(n_i\Phi_i) - \Phi_m + \Phi_n <= B_{total} \\ 0, & \text{otherwise} \end{cases}.$$

Let P_b be the blocking probability of new calls, P_d be the dropping probability of handoff calls, and P_{td} be the call dropping probability due to change of transmission rates. Given any system state $\bar{n} = (n_1, n_2, ..., n_y)$, let the bandwidth requirement $\tau(\bar{n}) = \sum_{i=1}^{y}(n_i\Phi_i)$. We can derive:

$$P_b = \sum_{\tau(\bar{n})\geq B_{th}} P_{n_1,n_2,\ldots,n_y} \qquad (4.5)$$

$$P_d = \frac{1}{y}\sum_{i=1}^{y}\left(\sum_{\tau(\bar{n})+\Phi_i > B_{total}} P_{n_1,n_2,\ldots,n_y}\right) \qquad (4.6)$$

$$P_{td} = \sum_{i=1}^{y-1}\left[\sum_{\tau(\bar{n})-\Phi_i+\Phi_{i+1} > B_{total}}\left(\frac{n_i}{\sum_{i=1}^{y-1} n_i + \sum_{i=2}^{y} n_i} P_{n_1,n_2,\ldots,n_y}\right)\right]. \qquad (4.7)$$

To compute P_b, P_d, and P_{td}, we have to solve the steady-state probabilities $P_{n_1,n_2,\ldots,n_y}$. This can be done by the recursive technique proposed by Herzog *et al.* [10], which states that there exists a subset of the state probabilities, called *boundaries*, such that all other states can be expressed as linear combinations of the boundary states. Therefore, we can determine the boundaries first and then derive the expressions for all remaining state probabilities as functions of the boundary values. This can significantly reduce the complexity of solving of $P_{n_1,n_2,...,n_y}$ as compared to traditional matrix inversion techniques. It has been shown to be suitable to solve a wide class of queuing problems.

First, we choose all states $(n_1, n_2, ..., n_y)$ such that $n_y = 0$ as boundaries. According to [10], we can rewrite the state probabilities as:

$$\begin{aligned} P_{n_1,n_2,...,n_y} &= \sum_{\alpha_{y-1}=0}^{\left\lfloor \frac{B_{total}}{\Phi_{y-1}} \right\rfloor} \cdots \sum_{\alpha_2=0}^{\left\lfloor \frac{B_{total}}{\Phi_2} \right\rfloor} \sum_{\alpha_1=0}^{\left\lfloor \frac{B_{total}}{\Phi_1} \right\rfloor} C^{\alpha_1,\alpha_2,...,\alpha_{y-1}}_{n_1,n_2,...,n_y} P_{\alpha_1,\alpha_2,...,\alpha_{y-1},\alpha_y=0} \qquad (4.8) \\ C^{\alpha_1,\alpha_2,...,\alpha_{y-1}}_{n_1,n_2,...,n_{y-1},n_y=0} &= \begin{cases} 1, & \text{if } n_1 = \alpha_1, n_2 = \alpha_2, ..., \text{ and } n_{y-1} = \alpha_{y-1} \\ 0, & \text{otherwise} \end{cases} . \end{aligned}$$

The coefficients $C^{\alpha_1,\alpha_2,...,\alpha_{y-1}}_{n_1,n_2,...,n_y}$ for $n_y \neq 0$ can be solved recursively. Next, we rewrite Eq.s (4.1)–(4.4) by:

$$\begin{aligned} B_{n_1,n_2,...,n_y} P_{n_1,n_2,...,n_y} &= \sum_{i=1}^{y} A^{r_i}_{n_1,n_2,...,n_y} P_{n_1,n_2,...,n_i+1,...,n_y} \\ &+ \sum_{i=1}^{y} A^{l_i}_{n_1,n_2,...,n_y} P_{n_1,n_2,...,n_i-1,...,n_y} \\ &+ \sum_{i=2}^{y} A^{u_i}_{n_1,n_2,...,n_y} P_{n_1,n_2,...,n_{i-1}-1,n_i+1,...,n_y} \\ &+ \sum_{i=1}^{y-1} A^{d_i}_{n_1,n_2,...,n_y} P_{n_1,n_2,...,n_i+1,n_{i+1}-1,...,n_y}, \qquad (4.9) \end{aligned}$$

where the coefficients $A^{r_i}_{n_1,n_2,...,n_y}$, $A^{l_i}_{n_1,n_2,...,n_y}$, $A^{u_i}_{n_1,n_2,...,n_y}$, $A^{d_i}_{n_1,n_2,...,n_y}$, and $B_{n_1,n_2,...,n_y}$ are abbreviations of those in Eq.s (4.1)–(4.4). Then we rewrite Eq. (4.9) as:

$$
\begin{aligned}
P_{n_1,n_2,\ldots,n_{y-1},n_y+1} \quad = \quad & \frac{B_{n_1,n_2,\ldots,n_y} P_{n_1,n_2,\ldots,n_y} - \sum_{i=1}^{y-1} A^{r_i}_{n_1,n_2,\ldots,n_y} P_{n_1,n_2,\ldots,n_i+1,\ldots,n_y}}{A^{r_y}_{n_1,n_2,\ldots,n_y}} \\
& - \frac{\sum_{i=1}^{y} A^{l_i}_{n_1,n_2,\ldots,n_y} P_{n_1,n_2,\ldots,n_i-1,\ldots,n_y}}{A^{r_y}_{n_1,n_2,\ldots,n_y}} \\
& - \frac{\sum_{i=2}^{y} A^{u_i}_{n_1,n_2,\ldots,n_y} P_{n_1,n_2,\ldots,n_{i-1}-1,n_i+1,\ldots,n_y}}{A^{r_y}_{n_1,n_2,\ldots,n_y}} \\
& - \frac{\sum_{i=1}^{y-1} A^{d_i}_{n_1,n_2,\ldots,n_y} P_{n_1,n_2,\ldots,n_i+1,n_{i+1}-1,\ldots,n_y}}{A^{r_y}_{n_1,n_2,\ldots,n_y}}.
\end{aligned} \tag{4.10}
$$

After some manipulation, Eq. (4.10) can be further rewritten as:

$$
\begin{aligned}
P_{n_1,n_2,\ldots,n_{y-1},n_y} \quad = \quad & \frac{B_{n_1,n_2,\ldots,n_y-1} P_{n_1,n_2,\ldots,n_y-1} - \sum_{i=1}^{y-1} A^{r_i}_{n_1,n_2,\ldots,n_y-1} P_{n_1,n_2,\ldots,n_i+1,\ldots,n_y-1}}{A^{r_y}_{n_1,n_2,\ldots,n_y-1}} \\
& - \frac{\sum_{i=1}^{y} A^{l_i}_{n_1,n_2,\ldots,n_y-1} P_{n_1,n_2,\ldots,n_i-1,\ldots,n_y-1}}{A^{r_y}_{n_1,n_2,\ldots,n_y-1}} \\
& - \frac{\sum_{i=2}^{y} A^{u_i}_{n_1,n_2,\ldots,n_y-1} P_{n_1,n_2,\ldots,n_{i-1}-1,n_i+1,\ldots,n_y-1}}{A^{r_y}_{n_1,n_2,\ldots,n_y-1}} \\
& - \frac{\sum_{i=1}^{y-1} A^{d_i}_{n_1,n_2,\ldots,n_y-1} P_{n_1,n_2,\ldots,n_i+1,n_{i+1}-1,\ldots,n_y-1}}{A^{r_y}_{n_1,n_2,\ldots,n_y-1}}.
\end{aligned} \tag{4.11}
$$

For each fixed state $(\alpha_1, \alpha_2, \ldots, \alpha_{y-1}, 0)$, if we let $P_{\bar{n}} = 1$ and $P_{\bar{n}'} = 0$ for all $\bar{n}' \neq \bar{n}$, from Eq.s (4.8) and (4.11), we can obtain:

$$
\begin{aligned}
C^{\alpha_1,\alpha_2,\ldots,\alpha_{y-1}}_{n_1,n_2,\ldots,n_y} \quad = \quad & \frac{B_{n_1,n_2,\ldots,n_y-1} C^{\alpha_1,\alpha_2,\ldots,\alpha_{y-1}}_{n_1,n_2,\ldots,n_y-1} - \sum_{i=1}^{y-1} A^{r_i}_{n_1,n_2,\ldots,n_y-1} C^{\alpha_1,\alpha_2,\ldots,\alpha_{y-1}}_{n_1,n_2,\ldots,n_i+1,\ldots,n_y-1}}{A^{r_y}_{n_1,n_2,\ldots,n_y-1}} \\
& - \frac{\sum_{i=1}^{y} A^{l_i}_{n_1,n_2,\ldots,n_y-1} C^{\alpha_1,\alpha_2,\ldots,\alpha_{y-1}}_{n_1,n_2,\ldots,n_i-1,\ldots,n_y-1}}{A^{r_y}_{n_1,n_2,\ldots,n_y-1}} \\
& - \frac{\sum_{i=2}^{y} A^{u_i}_{n_1,n_2,\ldots,n_y-1} C^{\alpha_1,\alpha_2,\ldots,\alpha_{y-1}}_{n_1,n_2,\ldots,n_{i-1}-1,n_i+1,\ldots,n_y-1}}{A^{r_y}_{n_1,n_2,\ldots,n_y-1}} \\
& - \frac{\sum_{i=1}^{y-1} A^{d_i}_{n_1,n_2,\ldots,n_y-1} C^{\alpha_1,\alpha_2,\ldots,\alpha_{y-1}}_{n_1,n_2,\ldots,n_i+1,n_{i+1}-1,\ldots,n_y-1}}{A^{r_y}_{n_1,n_2,\ldots,n_y-1}}
\end{aligned} \tag{4.12}
$$

for all combinations of $n_1, n_2, \ldots, n_y$.

After obtaining all coefficients $C_{n_1,n_2,...,n_y}^{\alpha_1,\alpha_2,...,\alpha_{y-1}}$, the probabilities of boundaries can be derived by solving the remaining unused $\lfloor \frac{B_{total}}{\Phi_1} \rfloor \times \lfloor \frac{B_{total}}{\Phi_2} \rfloor \times ... \times \lfloor \frac{B_{total}}{\Phi_{y-1}} \rfloor$ independent equations in Eq. (4.9) as well as the normalizing condition:

$$\sum_{n_1} \sum_{n_2} ... \sum_{n_y} P_{n_1,n_2,...,n_y} = 1 \tag{4.13}$$

.

Having solved the boundaries, all steady-state probabilities $P_{n_1,n_2,...,n_y}$ can be determined from (4.8). Thus, P_b, P_d, and P_{td} can then be derived.

Chapter 5

Simulation Results

To verify the correctness and applicability of our proposed algorithm, an event-driven simulator written is developed in C++. The following assumptions are made in our simulation. (1) The same call arrival model, call holding time, and call residence time as specified in Chapter 4 are used in the simulation. (2) The communication channel is assumed to be error-free. (3) No RTS/CTS is used. (4) The BI=500 ms. (5) There is only AC_VO traffic in the network, and its CW_{min}=7, CW_{max}=15, and IFSN=2. (6) G.726 with 32 kbps is used as the voice source. The offered network load is defined as $\rho = (\lambda_n + \lambda_h)/(\mu_r + \mu_h)$. To obtain steady states, each simulation case is run with one million arrivals. The performance metrics are new call blocking rate, handoff call dropping rate, and channel utilization.

5.1 Validation of Analytical Results

In this experiment, we assume that 40% of arrival calls are handoff calls. The channel occupancy time is 2 second. P_r is set to 0.8 when $B_{deg} \leq (B_{total} - B_{th})$. Fig. 5.1 shows the blocking rate and dropping rate of both analytical and simulation results. The maximal difference of blocking rate between simulation and analytical results is about 0.44%, which appears when $\rho = 20$. It can be seen that analytical results match well with simulation results.

5.2 Influence of CAC and RA

In this experiment, we want to evaluate the impact of CAC and RA. P_r is set to 0.8 when $B_{deg} \leq (B_{total} - B_{th})$. We compare our scheme against the CAC-only and "no-CAC-no-RA" cases. For the CAC-only case, the PIs of calls are fixed. Fig. 5.2(a) shows the channel utilization under different offered loads. Clearly, our scheme has very good

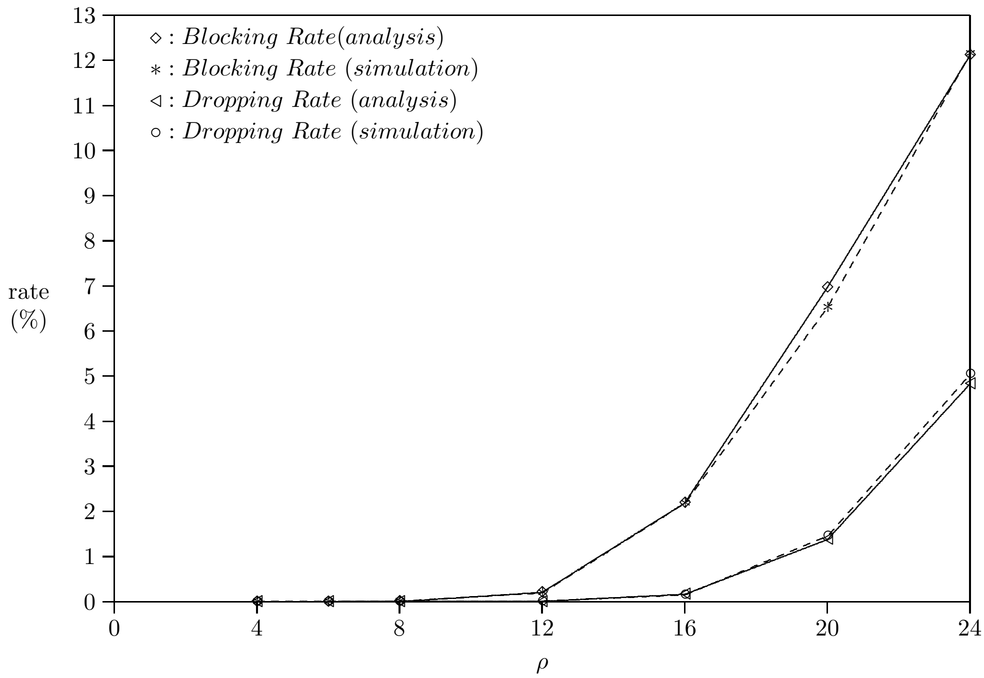


Figure 5.1: Comparison of simulation and analytical results on blocking rate and dropping rate ($\lambda_h = 0.8,\ \lambda_n = 1.2$).

utilization because calls can always be upgraded when there are extra resources. The no-CAC-no-RA (PI=20) case outperforms the CAC-only (PI=20) case because it accepts every incoming request in all network situations. Fig. 5.2(b) shows the medium time per session receives (which is approximated by the total used medium time divided by the total number of ongoing calls). With call admission control, the medium time of our scheme is better than that of the no-CAC-no-RA case. Even when the work load is high, our scheme can still guarantee the minimum bandwidth requirement of all calls. As $\rho \geq 32$, the minimum time of the no-CAC-no-RA case will drop to an unacceptable level. This shows that our scheme can well utilize network resources while guarantee the quality of calls.

Fig. 5.3 shows the new call blocking rate and handoff call dropping rate versus different offered loads. The rates of the no-CAC-no-RA case are all zero because every incoming request is accepted. From Fig. 5.3(a), we see that our scheme is only slightly worse than the CAC-only (PI=40) case after $\rho \geq 12$ because of our call acceptance policy. However, the benefit is our lower handoff call dropping rate, as shown in Fig. 5.3(b).

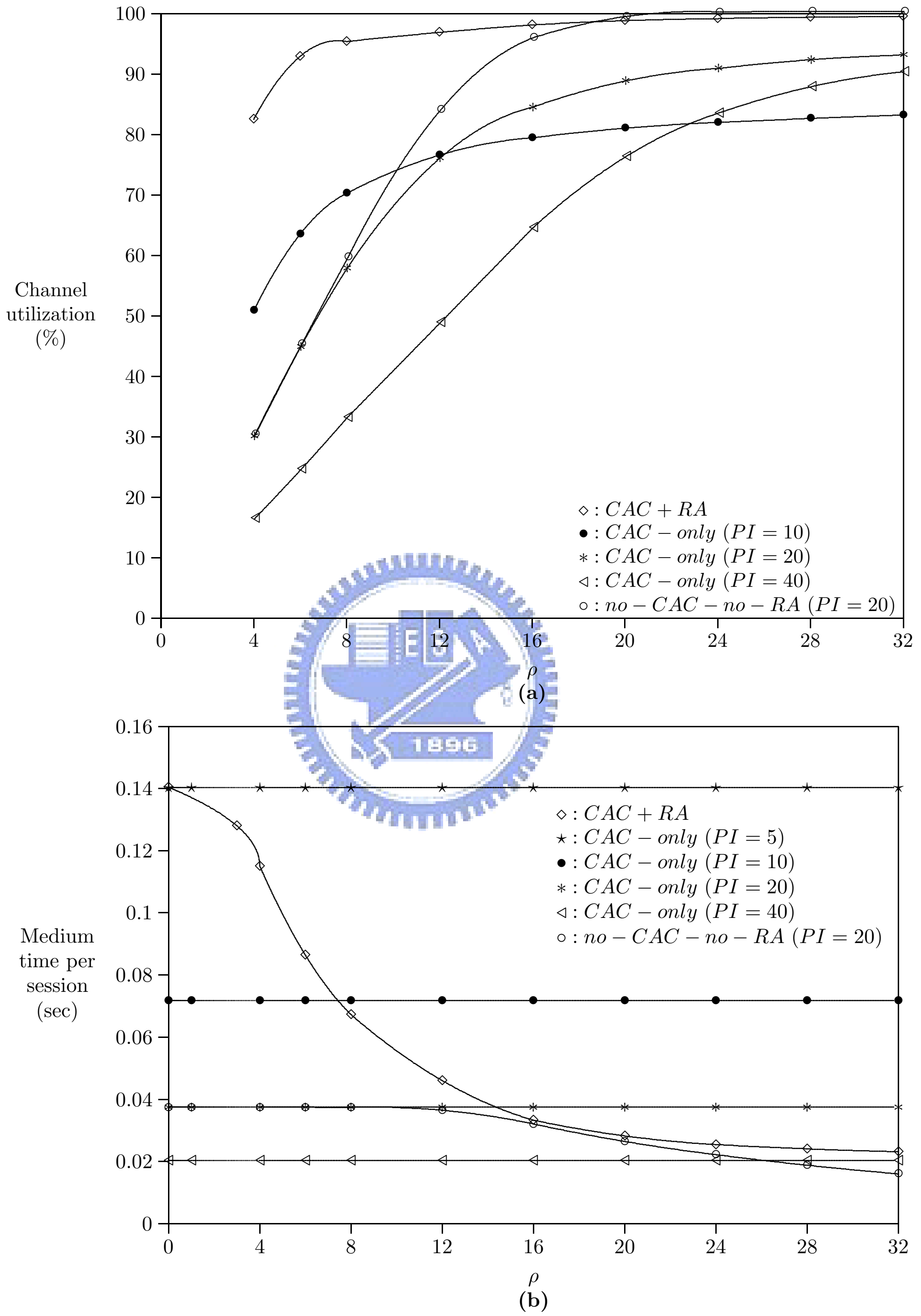
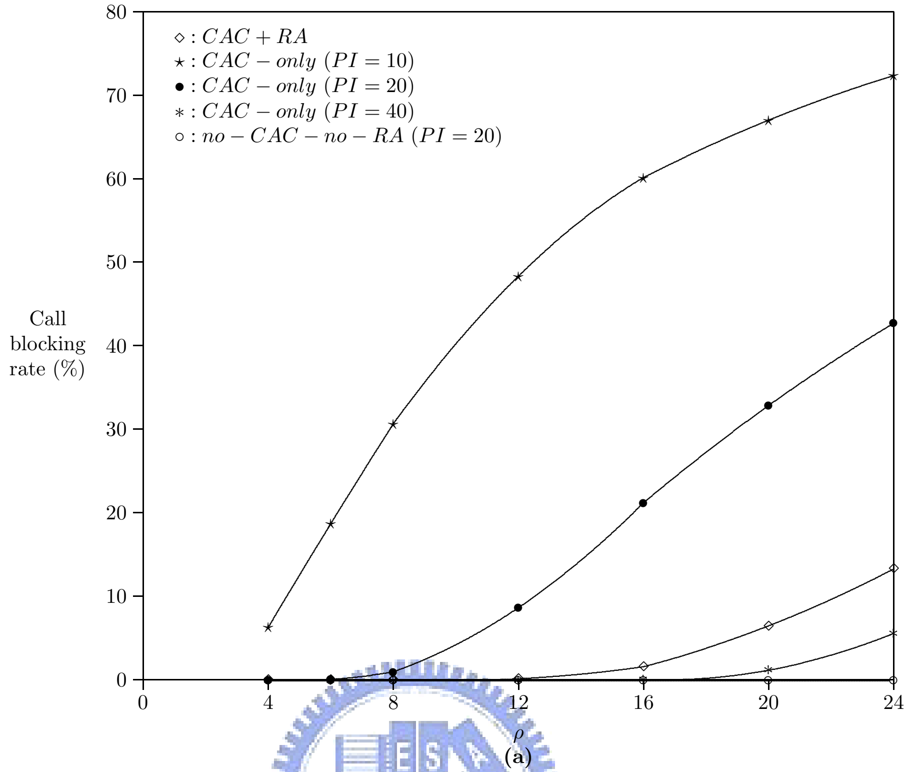


Figure 5.2: Comparisons of different schemes on: (a) channel utilization and (b) goodput.



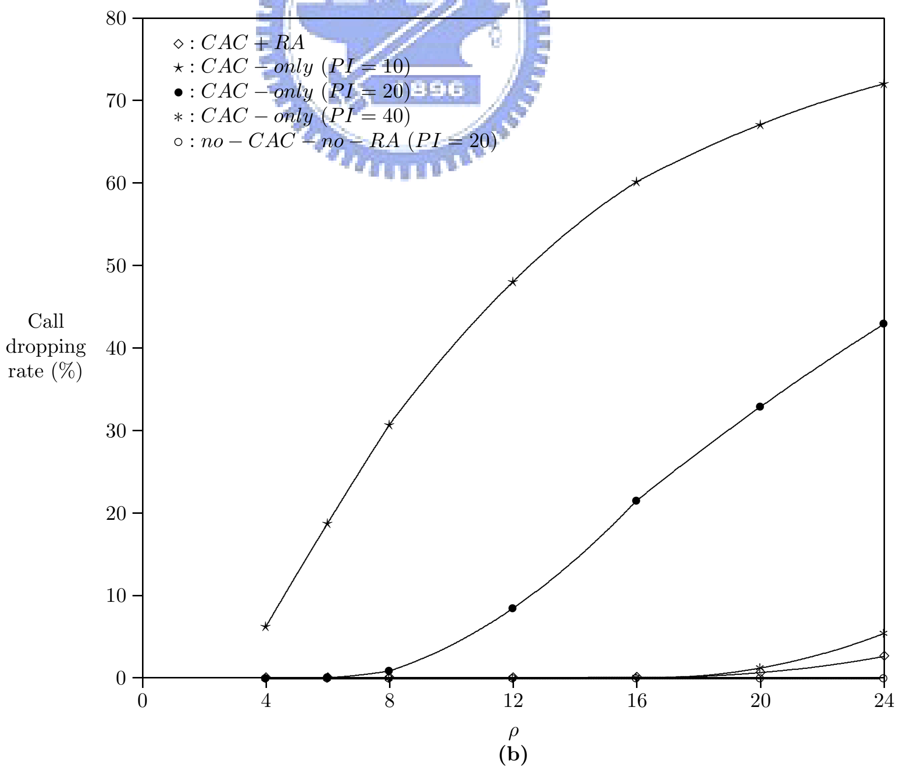


Figure 5.3: Comparisons of: (a) call blocking rate and (b) call dropping rate.

5.3 Influence of P_r

The value of P_r reflects the possibility that a QAP permits new calls to start. Clearly, a larger P_r will benefit new calls but hurt handoff calls. Fig. 5.4 shows the impact of P_r on call blocking and dropping probabilities. From these curves, a suggested value of P_r could range from 0.2 to 0.6.

5.4 Influence of Traffic Characteristic

Next, we evaluate the influence of traffic characteristic. We change the percentage of handoff calls while keep the offered load unchanged. Fig. 5.5 shows this impact on call blocking and call dropping rates. We can see that our scheme is quite insensitive to this change, unless the offered load is very high. This concludes that our scheme can provide good QoS to handoff calls.

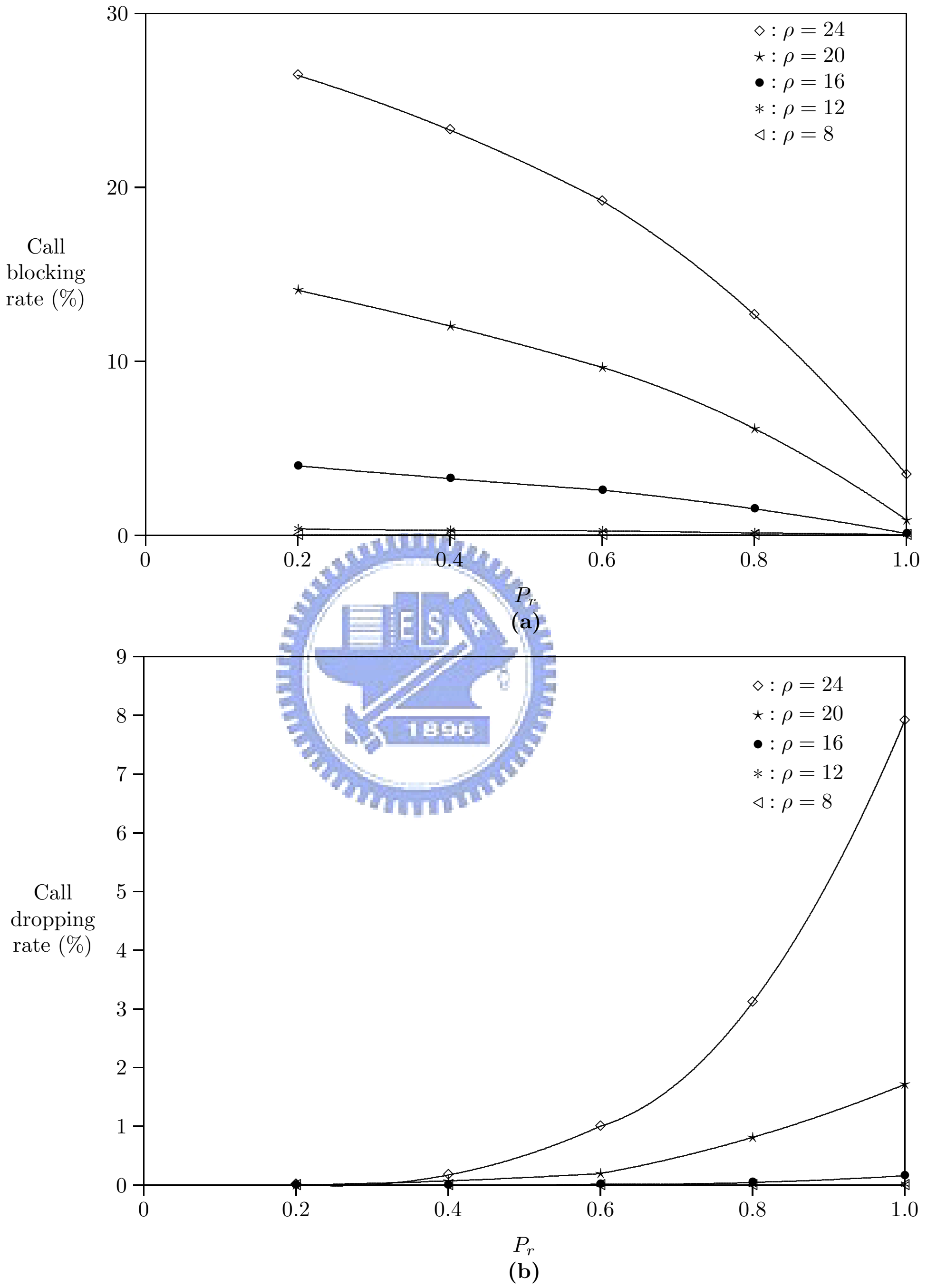


Figure 5.4: The impact of P_r on: (a) call blocking rate and (b) call dropping rate .

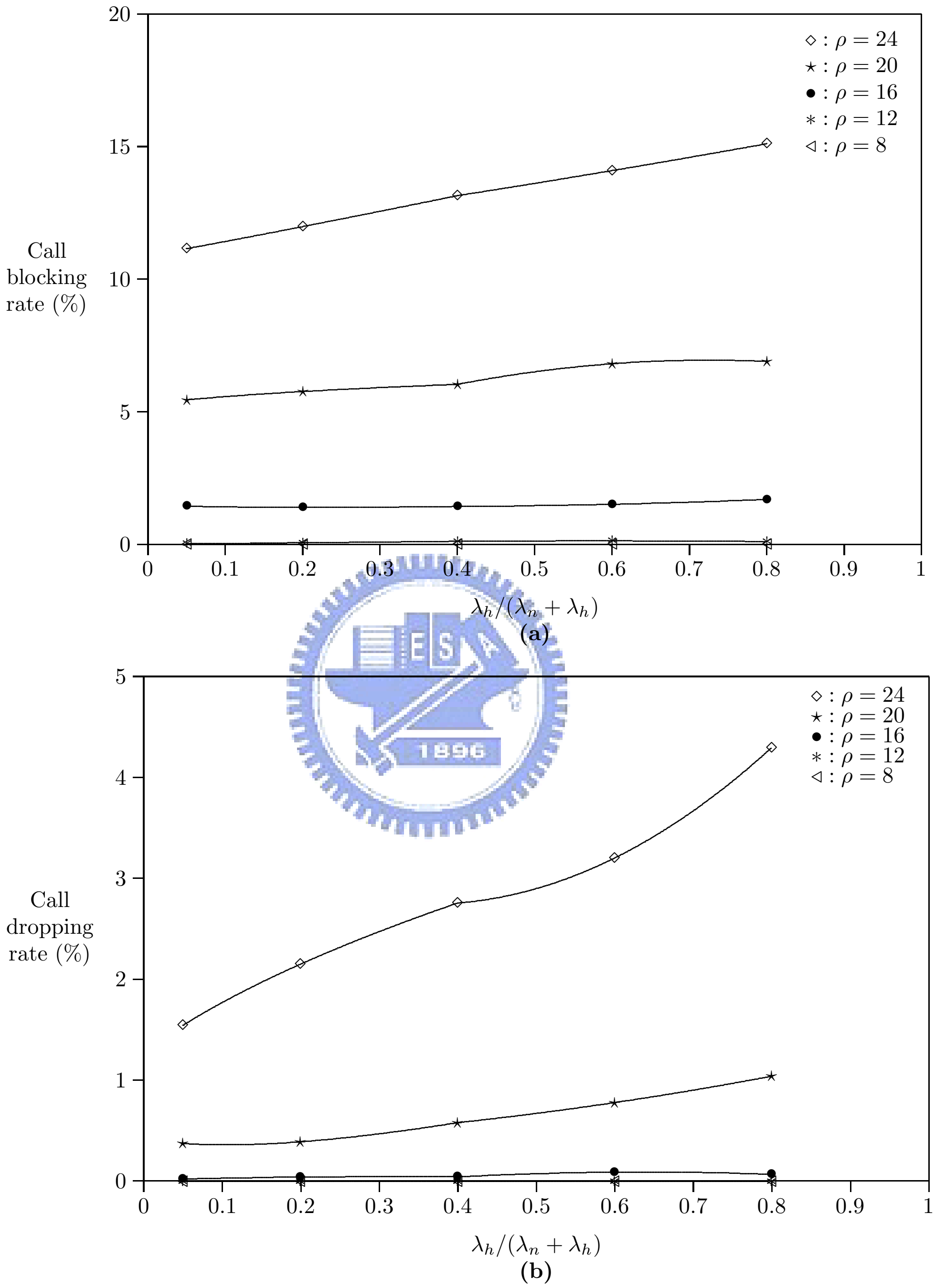


Figure 5.5: The impact of the percentage of handoff calls on: (a) call blocking rate and (b) call dropping rate.

Chapter 6

Conclusions

In this paper, we have proposed a CAC and a RA mechanisms to solve the handoff and resource redistribution issues over IEEE 802.11e wireless networks. By upgrading/degrading resources allocated to calls, we can make better use of the network resource. We have also derived an analytical model to evaluate our system with multi-level QoS support. Three performance metrics, blocking probability, dropping probability, channel utilization, have been derived. Our numerical analysis shows the importance of CAC and RA mechanisms, especially when user mobility is high and fairness is important. Our simulation results also support our conclusions.

Bibliography

[1] IEEE Std 802.11e-2005, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements. Nov 2005.

[2] P.-Y. Wu, Y.-C. Tseng, and H. Lee. Design of QoS and Admission Control for VoIP Services over IEEE 802.11e WLANs. *National Computer Symposium*, 2005. Taiwan.

[3] S. Garg and M. Kappes. Can I add a VoIP Call? *In Proc. of International Conference on Communications (ICC 2003)*, 26(1), May 2003.

[4] N. Nasser. Adaptability Enhanced Framework for Provisioning Connection-level QoS in Multimedia Wireless Networks. *IEEE and IFIP International Conference on Wireless and Optical Communications Networks (WOCN)*, pages 275–279, Mar 2005.

[5] C.-T. Chou and K.-G. Shin. Analysis of Adaptive Bandwidth Allocation in Wireless Networks with Multilevel Degradable Quality of Service. *IEEE Trans. on Mobile Computing*, 3(1):5–17, Jan./Feb. 2004.

[6] G. Karlsson. Asynchronous transfer of video. *IEEE Communication Magazine*, 11:118–126, Aug 1996.

[7] M. Handley and V. Jacobson. *SDP: Session Description Protocol, RFC2327*. in IETF, 1998.

[8] Recommended Practice for Multi-Vendor Acess Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation. *IEEE Draft 802.1f/Final Version*, Jan 2003.

[9] Y.-C. Kim, D.-E. Lee, B.-J. Lee, Y.-S. Kim, and B. Mukherjee. Dynamic Channel Reservation based on Mobility in Wireless ATM. *Proc. of IEEE wmATM'99*, pages 100–106, 1999.

[10] U. Herzog, L. Woo, and K. Chandy. Solution of queueing problems by a recursive technique. *IBM J. Res. Dev.*, 19:295–300, May 1975.

[11] J. Rosenberg, H. Schulzrinne, E. Schooler, M. Handley, G. Camarillo, A. Johnston, J. Peterson, and R. Sparks. SIP: Session Initiation Protocol, RFC3261 in IETF. June 2002.

[12] D. Conllins. *Carrier Grade Voice over IP.* McGraw-Hill Companies, Inc, second edition, 2003.

[13] 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.

[14] A. Malla, M. El-Kadi, S. Olario, and P. Todorova. A Fair Resource Allocation Protocol for Multimedia Wireless Networks. *IEEE Transactions On Parallel And Distributed Systems*, 14(1), Jan 2003.

Curriculum Vita

Ling Lee (liling@csie.nctu.edu.tw) received her B.S. degree in Computer Science from the National Chiao-Tung University, Taiwan, in 2004. Her research interests include wireless networks, VoIP, and QoS issues.