Computer Networks 55 (2011) 1719–1734

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Integrating SIP and IEEE 802.11e to support handoff and multi-grade QoS for VoIP-over-WLAN applications

Jen-Jee Chen [a,*], Ling Lee [b], Yu-Chee Tseng [b]

[a] Department of Electrical Engineering, National University of Tainan, Tainan 70005, Taiwan
[b] Department of Computer Science, National Chiao-Tung University, Hsin-Chu 30010, Taiwan

ARTICLE INFO

Article history:
Received 5 July 2009
Received in revised form 2 May 2010
Accepted 6 January 2011
Available online 11 January 2011
Responsible Editor: X.S. Shen

Keywords:
IEEE 802.11e
Quality of Service (QoS)
Call admission control (CAC)
Voice-over-IP (VoIP)
Rate adaptation
Handoff
Wireless network
Session Initiation Protocol (SIP)

ABSTRACT

With the increasing popularity of WLANs and the growing demand for VoIP services, there is a need to guarantee QoS for VoIP calls while support as many calls as possible. Prior efforts focus on call admission control (CAC), but did not address the important handoff and physical rate adaptation issues caused by mobility and wireless channel variation. Assuming that VoIP calls can be supported by multi-grade QoS levels and that handoff and rate adaptation are possible, we show how to conduct CAC and resource management by integrating SIP and the QoS mechanisms of IEEE 802.11e together. When wireless resource is stringent, our resource management can dynamically adjust the resource distribution among existing calls by controlling their supporting codecs and packetization intervals. This not only takes care of calls in bad channel conditions, but also can accept more calls. Thus multi-grade QoS is achieved with 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 for handoff calls. Both analytical and simulation results are presented to evaluate the performance of the proposed schemes.

1. Introduction

A lot of research efforts have been dedicated to Voice-over-IP (*VoIP*) techniques. In particular, VoIP-over-WLAN (wireless local area network) 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 802.11e* [1] has been defined to expand the 802.11 MAC protocol to support applications with QoS requirements. However, neither call admission control (*CAC*) nor resource management algorithm is specified for VoIP applications.

User mobility and rate adaptation are two important factors concerning VoIP-over-WLAN services. With user mobility, handoff between access points (APs) is inevitable. Failure to reserve sufficient bandwidths for handoff calls may force them being dropped. Dropping on-going 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. On the other hand, IEEE 802.11 supports multiple transmission rates to adapt to physical channel conditions. However, to keep the same level of QoS, using a lower transmission rate means that more communication time has to be allocated to that mobile station. It is a challenge to guarantee the QoS of VoIP calls over a multi-rate WLAN concerning user mobility and rate adaptation.

Recently, there have been increasing interests in supporting QoS for running VoIP-over-WLAN. In [2–4], the number of concurrent homogeneous VoIP sessions that can be supported in a WLAN is evaluated. Without considering that calls in a WLAN may use different codecs and packetization intervals (*PIs*) and connections may have

[*] Corresponding author.
E-mail addresses: jjchen@mail.nutn.edu.tw (J.-J. Chen), liling@cs.nctu.edu.tw (L. Lee), yctseng@cs.nctu.edu.tw (Y.-C. Tseng).

1389-1286/$ - see front matter Crown Copyright © 2011 Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2011.01.005

variable rates, these CAC schemes are not generic enough. Ref. [5] proposes to admit streams when the total demand does not exceed the achievable bandwidth in the WLAN, where the achievable bandwidth is evaluated from the payload size per packet and physical rates. References [6–8] propose to replace achievable bandwidth by indexes such as channel utilization and collision rates to determine whether a new call can be admitted or not. In [9], a CAC scheme based on collision probability is proposed, where the collision probability is calculated from the measured channel busy probability. Reference [10] manipulates the CWmin size to accept as many calls as possible and reject calls when there is no applicable CWmin for stations. In [11], it claims that when the collision probability is very small, the channel busy ratio is almost the same as the channel utilization. Moreover, it is shown that the optimal achievable channel utilization is about 0.9 (without RTS/CTS) or 0.95 (with RTS/CTS), which is independent of the number of active nodes and packet sizes. After the optimal point, adding more calls will increase both packet delays and collision rates dramatically and drop network throughput quickly. On the other hand, several works [12–14] have exploited the point coordination function (PCF) to enhance the voice capacity over WLANs. However, PCF is not supported by most available equipments in the market. There are also some model-based admission control schemes [15–17] proposed for CAC. But they all consider a saturated condition, which is not always true since we often experience non-saturated conditions. As we can see, all these works do not consider the issues of handoff and physical rate adaptation in WLANs.

Rate adaptation and call handoff can both vary the required resource in a WLAN. However, existing works as reviewed above all reserve fixed resource for a call based on the initial physical rate when it first joined the WLAN. How to conduct resource management to support VoIP over multi-rate WLANs is a challenge. Typical multimedia applications can tolerate some degree of temporary bandwidth fluctuation with no or little perceived degradation in quality by using a rate-adaptive codec or hierarchical encoding [18–20]. Although this is also applicable to VoIP applications, the range of required resources that can be adapted is somewhat limited. In this work, we propose CAC and resource management mechanisms to support handoff and multi-grade QoS for VoIP over multi-rate WLANs. When handoff calls are accepted to a QAP (*Quality of Service Access Point*) or when physical transmission rates of calls drop, we will see increasing competition among QSTAs (*Quality of Service Stations*) and thus decreasing bandwidth shares of existing calls. Our main idea is to change the codecs and/or PIs of some existing calls to degrade their QoS levels when resources are too stringent. We show that the new call blocking rate and the handoff call dropping rate can both benefit from such adaptations. In particular, we will show that changing the PIs of calls is quite effective to "squeeze" some QSTAs' medium time so as to serve those QSTAs increasing their required medium time due to mobility. Our scheme will adopt the cross-layer architecture in [21] to integrate SIP and IEEE 802.11e together to conduct CAC to facilitate VoIP traffics over IEEE 802.11e multi-rate WLANs with possible handoff among WLAN. Note that SIP is the most promising signaling protocol for VoIP over the Internet. We will also adopt the ITU-T-recommended 150 ms as the bound for one-way end-to-end delay [22]. To summarize, our work contributes in proposing a systematical method for changing PIs and codecs of VoIP calls to improve the overall capacity of a WLAN.

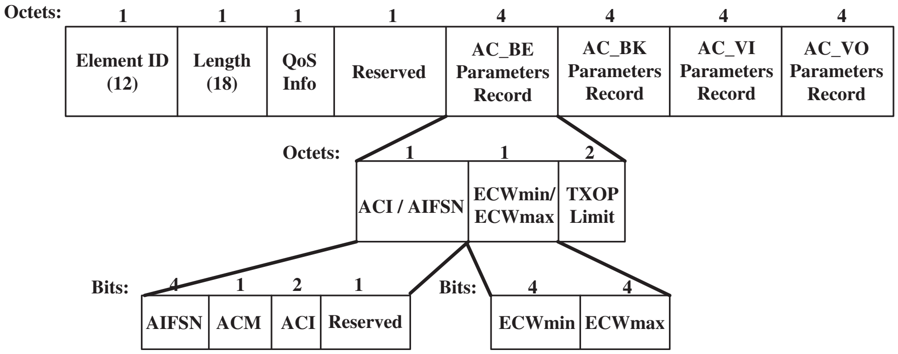


Fig. 1. Structure of the IEEE 802.11e EDCA_Parameter_Set information element.

Octets:	1	1	3	2	2	4	4	4	4
	Element ID (13)	Length (55)	TS Info	Nominal MSDU Size	Maximum MSDU Size	Minimum Service Interval	Maximum Service Interval	Inactivity Interval	Suspension Interval

4	4	4	4	4	4	4	2	2
Service Start Time	Minimum Data Rate	Mean Data Rate	Peak Data Rate	Maximum Burst Size	Delay Bound	Minimum PHY Rate	Surplus Bandwidth Allowance	Medium Time

Fig. 2. Structure of the TSPEC information element.

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

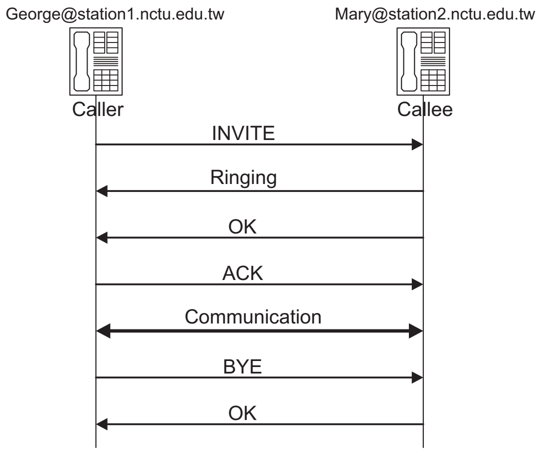


Fig. 3. An example of SIP call setup and tear-down.

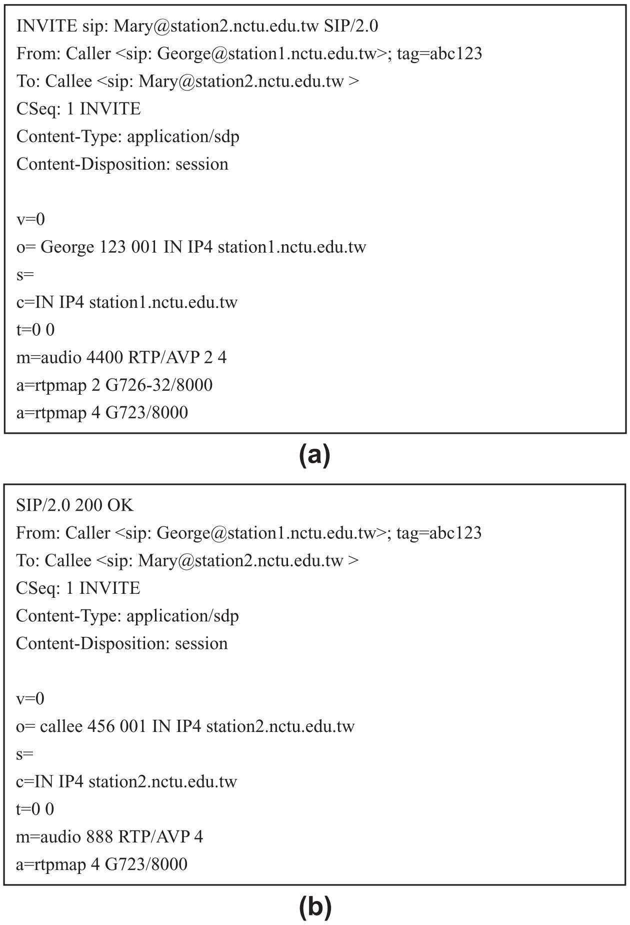
2. Preliminaries

2.1. IEEE 802.11e MAC protocol

The IEEE 802.11e Working Group defines 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 separated by SIFS (short inter-frame spacing). A station can obtain a TXOP by either contention or scheduled access assigned by polling messages.

2.1.1. 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 back-



```
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)

Fig. 4. An example of SIP with SDP message bodies: (a) INVITE siganl and (b) OK signal.

ground (AC_BK), best effort (AC_BE), video (AC_VI), and voice (AC_VO), and are prioritized by different AIFS (*Arbitration Inter-Frame Space*) 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. 1), which are sent in beacon frames, specify the parameters of ACs.

2.1.2. 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 direction of the stream and a TSPEC (*Traffic Specification*) information element (Fig. 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 is 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*) [23] to specify its session characteristics. Fig. 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 an SDP message body. Fig. 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 Ringing and an OK signal with the selected codec. In Fig. 4(b), the selected codec is G.723, and the receiving port is 888. If the port number is 0, it means a rejection.

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 problems caused by handoff and rate adaptation. The first mechanism is a 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 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.

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 $\bar{s} = (s_1, s_2, s_3, \ldots, s_k)$, 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. 5 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 measure its current available resource and the required resource to support this call (refer to box A in Fig. 5). We will use the required *medium time* for the measurement. 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. 5). Then, the voice communication can be started. When the caller moves out of the coverage of QAP1, 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 measure its available bandwidth and reply an IEEE 802.11 *Probe Response* accordingly (refer to box A in Fig. 5). The caller will collect all *Probe Responses* and select a new QAP (refer to box B in Fig. 5). 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. 5). In the meanwhile, QAP3 and QAP1 will exchange the caller's context by *Inter Access Point Protocol* (*IAPP* [24]). Via IAPP,

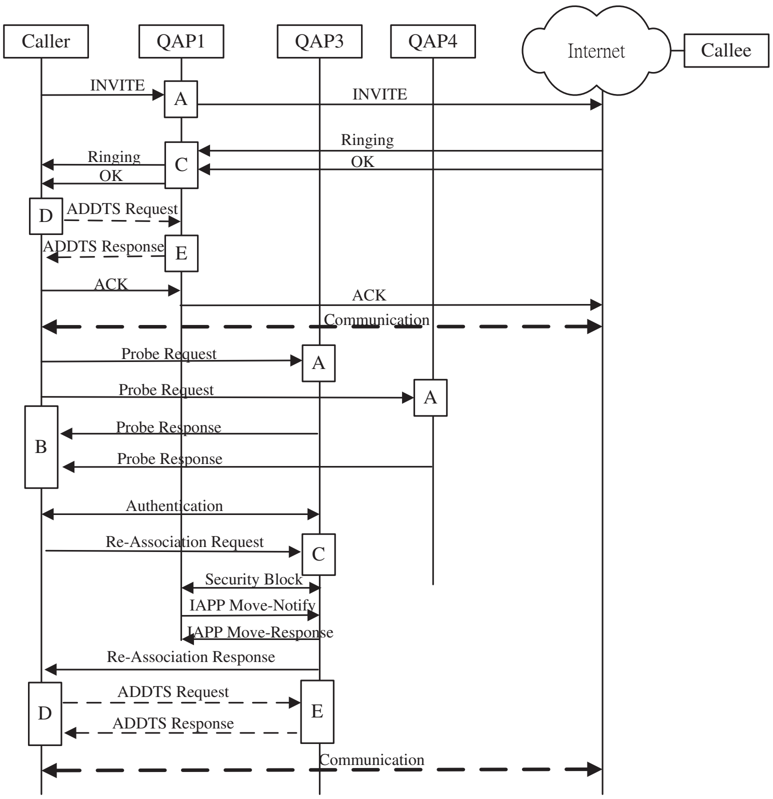
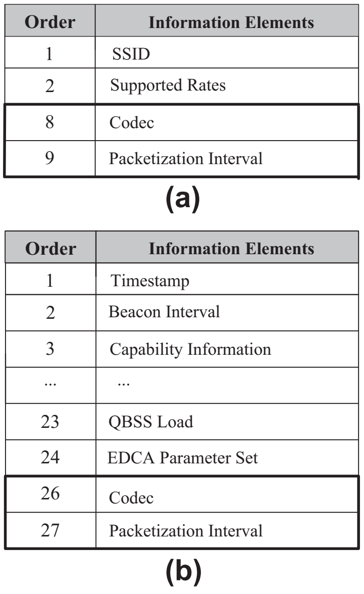


Fig. 5. The proposed message flows of the CAC procedure in IEEE 802.11e networks.

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. 5) 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.

3.1.1. Resource estimation at the QAP (box A)

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. 6 shows the proposed orders of



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)

Fig. 6. The orders of information elements in (a) Probe Request and (b) Probe Response.

Table 1
The Packet Size Table, which contains the packet sizes (in bytes) when different codecs and packetization intervals are used.

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	

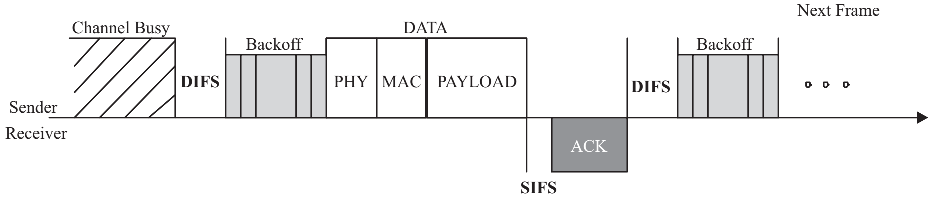


Fig. 7. Basic operations of 802.11e EDCF.

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 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). Using the beacon interval (*BI*) as the time unit, given a call's codec, PI, and current physical rate *r*, we can compute the required *Medium Time* (*MT*) that should be reserved for the call per BI:

$$\begin{aligned} MT(codec, PI, r) &= (\text{total\_time\_needed\_per\_BI}) \\ &= (\text{time\_to\_send\_one\_packet}) \times (BI/PI) \\ &\quad \times (\text{surplus\_bandwidth\_allowance}) \\ &= \{[(H_{RTP} + H_{UDP} + H_{IP} + H_{MAC}) + L(c,p)]/r \\ &\quad + (DIFS + averageCW + PHY\_header) \\ &\quad + (SIFS + ACK)\} \times (BI/PI) \\ &\quad \times (\text{surplus\_bandwidth\_allowance}), \qquad (1) \end{aligned}$$

where $L(c,p)$ is the payload per packet when codec *c* and PI *p* are used, *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 value of *averageCW* may be estimated from recent statistics of the QAP. The basic operation of 802.11e is shown in Fig. 7. Relevant parameters of 802.11e EDCA are listed in Table 2. The surplus_bandwidth_allowance is to take the extra medium access overhead (contention, collision, etc.) into account; in this work, we assume its value to be 1.1. This is consistent with the optimal channel utilization 0.9 found in [11] when RTS/CTS is disabled. For example, when BI is 1 s and min_PHY_rate is 11 Mbps, if we use G.726 with rate of 32 kbps and PI of 20 ms, then $MT = [154/11\,(\text{bytes/Mbps}) + (50 + 70 + 192) + (10 + 248)] \times (1000/20) \times 1.1 = 37.51$ ms.

Table 2
Parameters of IEEE 802.11e EDCA.

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

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. If a codec's required MT is larger than 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".

3.1.2. QAP selection at the caller (box B)

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.

3.1.3. Resource reservation at QAP (box C)

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 $MT(c, PI_{max}, r) \leqslant B_{deg}$ and $B_{deg} > (B_{total} - B_{th})$, the call is accepted directly.
  - If $MT(c, PI_{max}, r) \leqslant B_{deg}$ but $B_{deg} \leqslant (B_{total} - B_{th})$, 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 [25]. In this work, we will only consider adjusting PI for handoff calls, although adjusting codec is also possible. In the above rules, we do not try to degrade a handoff call's PI because a handoff call is more vulnerable to packet loss, so keeping its original PI is essential.

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 $MT(c,p,r) \leqslant B_{free}$. 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. 8. The function will repeatedly select an

```
degrade(c, p, r)
1:  $t\_PI = p$ ;
2:  while (not all calls are served by $PI_{max}$) do
3:      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;
4:      change $X$'s PI to next(X.PI);
5:      $B_{free} = B_{free} + MT(X.codec, X.PI, X.rate)$
        $-MT(X.codec, next(X.PI), X.rate)$;
6:      if ($B_{free} \geq MT(c, t\_PI, r)$) then
7:          return($t\_PI$);
8:      else if (there is no call with PI smaller than or equal to $t\_PI$) then
9:          $t\_PI$ = next(t_PI);
10:     end if;
11: end while;
12: return($PI_{max}$);
```

Fig. 8. The bandwidth degrade algorithm.

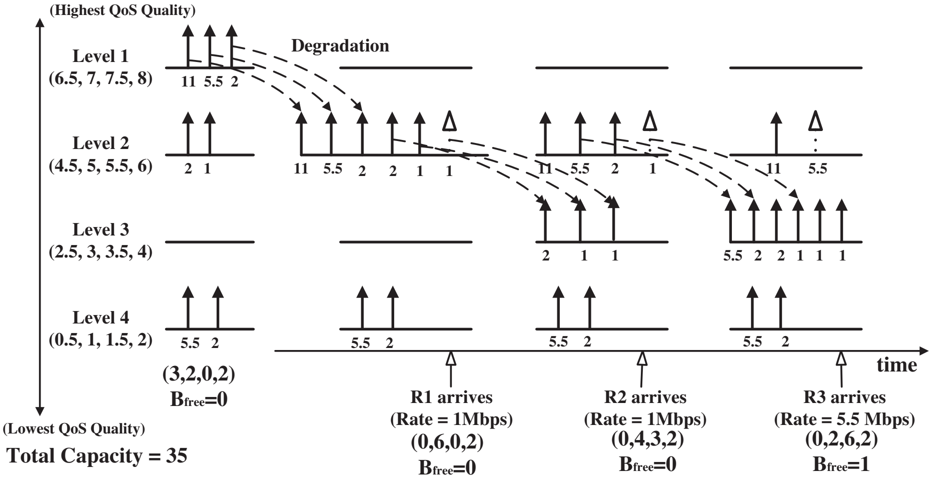


Fig. 9. An example of bandwidth degrade.

existing call to reduce 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. Note that for a new call, we do allow degrading its requested PI, as reflected in line 9 of the algorithm.

Fig. 9 shows an example. Suppose that there are $k = 4$ QoS levels, and there are 3, 2, 0, and 2 calls with QoS levels 1, 2, 3, and 4, respectively, currently in the system, represented by system state of (3,2,0,2). Also suppose that the required resources for these QoS levels are (6.5,7,7.5,8), (4.5,5,5.5,6), (2.5,3,3.5,4), and (0.5,1,1.5,2), respectively (the four numbers map to four physical rates in an descending order). Assuming a total capacity of 35 for the QAP, it has no resource remaining. Now, 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).

3.1.4. ADDTS request by the caller (box D)

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 PI of the corresponding codec.
- Mean_Data_Rate = the PI selected by the callee.
- Maximum_Data_Rate = the acceptable shortest PI.
- Medium_Time = the codec selected by the callee.

3.1.5. ADDTS response by the QAP (box E)

According to the caller's *ADDTS Request* and the Packet Size Table, QAP can compute the required MT following Eq. (1). Each QAP keeps the following variables:

- *TXOPBudget*$[AC_i]$ = the remaining bandwidth that can be allocated to AC_i, $i = 0,\ldots,3$.
- *TxAdDn*$[AC_i][TSID]$ = the admitted MT for stream TSID of AC_i in the downlink direction.
- *TxAdUp*$[AC_i][TSID]$ = the admitted MT for stream TSID of AC_i in the uplink direction.
- *TxAdDn*$[AC_i]$ = this 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 MT of all downlink streams of AC_i.

Initially, *TXOPBudget*$[AC_i]$ contains all the bandwidth (in terms of MT) 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 MT for stream TSID of AC_i in the uplink direction in this QSTA per BI.
- *TxAdUp*$[AC_i]$ = this 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 MT 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:

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

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 = p 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 reallocate bandwidth to calls for fairness. The RA algorithm may be triggered by the following two events: departure of calls and transmission rate change of existing calls (refer to Fig. 10). 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. 11 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.

```
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 $r$:
     $B_{free} = B_{free} + MT(X.codec, X.PI, X.rate) - MT(X.codec, X.PI, r);$
     $upgrade();$
```

Fig. 10. The RA algorithm.

```
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;
```

Fig. 11. The bandwidth upgrade algorithm.

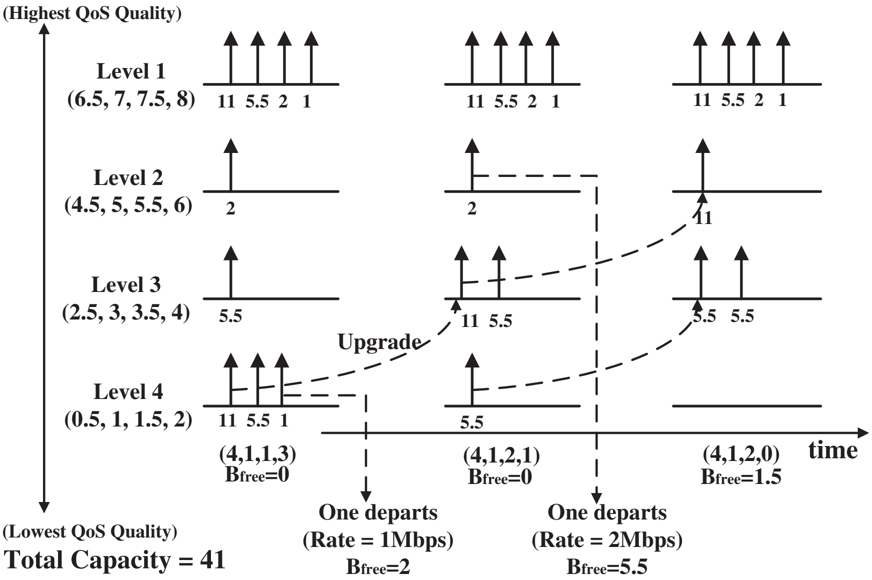


Fig. 12. An example of bandwidth upgrade.

Fig. 12 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. 9. Let the total capacity be 41. Suppose that a level 4 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 2 Mbps leaves, releasing a bandwidth of 5.5. This can upgrade the only call at level 4 to level 3 and the call at QoS level 3 with rate 11 Mbps to level 2, resulting a system state of (4, 1, 2, 0).

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, $PI_1, PI_2, \ldots, PI_m$ (in an ascending order), and y transmission rates, $R_1, R_2, \ldots, R_y$ (in a descending order).

Due to mobility, the rate change of a QSTA is modeled by the state diagram in Fig. 13. From each state, a QSTA can transit to a higher or a lower rate with a rate v 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 $R_1, R_2, \ldots, R_y$. Call holding time (duration of a call) and cell residence time (duration of staying in a QAP) are exponentially distributed with means of $1/\mu_h$ and $1/\mu_r$, respectively. Thus, the channel occupancy

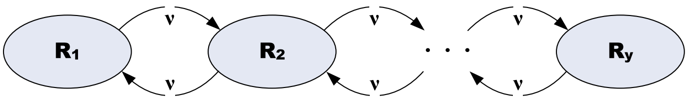


Fig. 13. The state transition diagram of a QSTA's rate change.

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 (refer to the **while** loop in **degrade**), a QAP will reject an incoming call when all existing calls cannot be further degraded (i.e., their PIs have reached PI_{max}). Therefore, to obtain blocking and dropping probabilities, we can assume that all calls use PI_{max}. 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 $BI = 1$ s, $surplus = 1.1$, and $PI_{max} = 40$ ms, the required MT per BI of a call under each rate is listed in Table 3.

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, $i = 1,\ldots,4$. For each state (n_1,n_2,n_3,n_4), there are 14 possible state transitions, as shown in Fig. 14, 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 Section 3.1. Vertical transitions are caused by transmission rate change.

Table 3
The required *MT* of a bi-directional voice call under different physical rates under our analytical model.

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

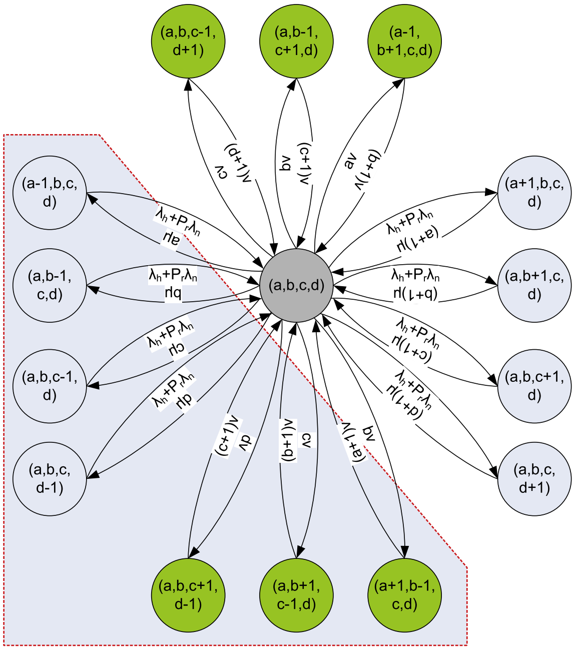


Fig. 14. Generic state transitions under our analytical model.

A simplified two-dimension Markov process is shown in Fig. 15 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 $P_{n_1,n_2,\ldots,n_y}$ of each state. There are four cases:

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

$$y(\lambda_h+\lambda_n)P_{n_1,n_2,\ldots,n_y} = \mu\sum_{i=1}^{y}P_{n_1,n_2,\ldots,n_i+1,\ldots,n_y}. \tag{2}$$

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

$$\begin{aligned}\left[y(\lambda_h+\lambda_n)+\left(\sum_{i=1}^{y}n_i\right)\mu+\left(\sum_{i=1}^{y-1}n_i+\sum_{i=2}^{y}n_i\right)v\right]P_{n_1,n_2,\ldots,n_y}\\ =\sum_{i=1}^{y}\left[(n_i+1)\mu P_{n_1,n_2,\ldots,n_i+1,\ldots,n_y}+(\lambda_h+\lambda_n)P_{n_1,n_2,\ldots,n_i-1,\ldots,n_y}\right]\\ +\sum_{i=2}^{y}\left[(n_i+1)vP_{n_1,n_2,\ldots,n_{i-1}-1,n_i+1,\ldots,n_y}\right]\\ +\sum_{i=1}^{y-1}\left[(n_i+1)vP_{n_1,n_2,\ldots,n_i+1,n_{i+1}-1,\ldots,n_y}\right].\end{aligned} \tag{3}$$

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

$$\begin{aligned}\left[y(\lambda_h+P_r\lambda_n)+\left(\sum_{i=1}^{y}n_i\right)\mu+\left(\sum_{i=1}^{y-1}n_i+\sum_{i=2}^{y}n_i\right)v\right]P_{n_1,n_2,\ldots,n_y}\\ =\sum_{i=1}^{y}\left[(n_i+1)\mu P_{n_1,n_2,\ldots,n_i+1,\ldots,n_y}\right]\\ +\sum_{i=1}^{y}\left[(\lambda_h+(I_i+\bar{I}_iP_r)\lambda_n)P_{n_1,n_2,\ldots,n_i-1,\ldots,n_y}\right]\\ +\sum_{i=2}^{y}\left[(n_i+1)vP_{n_1,n_2,\ldots,n_{i-1}-1,n_i+1,\ldots,n_y}\right]\\ +\sum_{i=1}^{y-1}\left[(n_i+1)vP_{n_1,n_2,\ldots,n_i+1,n_{i+1}-1,\ldots,n_y}\right],\end{aligned} \tag{4}$$

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

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

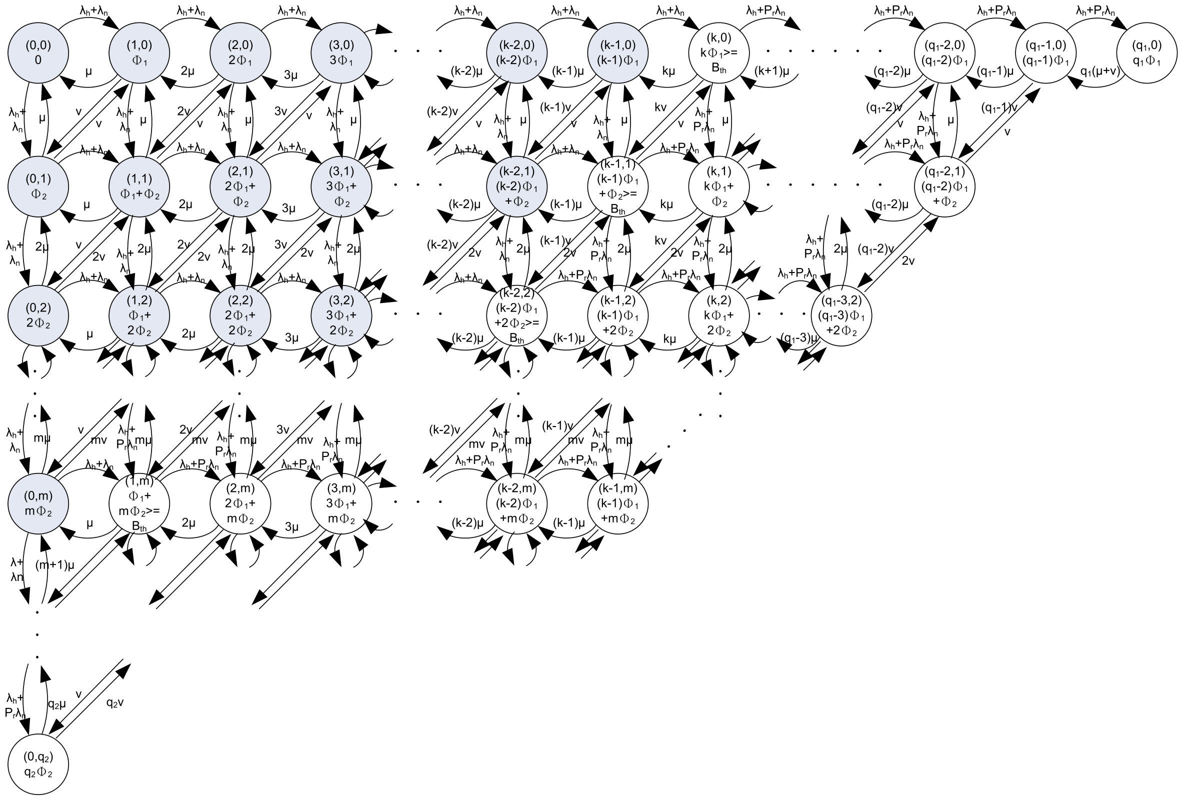


Fig. 15. A state transition example with $y = 2$(q_1 and q_2 are the maximum numbers of calls that can be accommodated with PI_{max} at rates R_1 and R_2, respectively).

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

$$\left[\left(\sum_{i=1}^{y} I_i\right)(\lambda_h + P_r\lambda_n) + \left(\sum_{i=1}^{y} n_i\right)\mu + \left(\sum_{i=1}^{y-1} n_i + \sum_{i=2}^{y} n_i\right)v\right]P_{n_1,n_2,\ldots,n_y}$$
$$= \sum_{i=1}^{y-1}\left[I_i(n_i+1)(\mu + \bar{I}_{i+1}v)P_{n_1,n_2,\ldots,n_i+1,\ldots,n_y}\right]$$
$$+ I_y(n_y+1)\mu P_{n_1,n_2,\ldots,n_{y-1},n_y+1} + (\lambda_h + P_r\lambda_n)$$
$$\times \sum_{i=1}^{y} P_{n_1,n_2,\ldots,n_i-1,\ldots,n_y}$$
$$+ \sum_{i=2}^{y}\left[I_{i-1,i}(n_i+1)vP_{n_1,n_2,\ldots,n_{i-1}-1,n_i+1,\ldots,n_y}\right]$$
$$+ \sum_{i=1}^{y-1}\left[(n_i+1)vP_{n1,n_2,\ldots,n_i+1,n_{i+1}-1,\ldots,n_y}\right], \tag{5}$$

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

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

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

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

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 $\bar{n} = (n_1, n_2, \ldots, n_y)$, let the bandwidth requirement $\tau(\bar{n}) = \sum_{i=1}^{y}(n_i\Phi_i)$. We can derive:

$$P_b = \sum_{\tau(\bar{n}) \geqslant B_{th}} P_{n_1,n_2,\ldots,n_y}, \tag{6}$$

$$P_d = \frac{1}{y}\sum_{i=1}^{y}\left(\sum_{\tau(\bar{n})+\Phi_i > B_{total}} P_{n_1,n_2,\ldots,n_y}\right), \tag{7}$$

$$P_{td} = \sum_{i=1}^{y-1}\left[\sum_{\tau(\bar{n})-\Phi_i+\Phi_{i+1} > B_{total}}\left(\frac{n_i}{\sum_{i=1}^{y-1} n_i + \sum_{i=2}^{y} n_i}P_{n_1,n_2,\ldots,n_y}\right)\right]. \tag{8}$$

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. [26], 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 $P_{n_1,n_2,\ldots,n_y}$ 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 $(n_1, n_2, \ldots, n_y)$ such that $n_y = 0$ as boundaries. According to [26], we can rewrite the state probabilities as:

$$P_{n_1,n_2,\ldots,n_y} = \sum_{\alpha_{y-1}=0}^{\left\lfloor \frac{B_{total}}{\Phi_{y-1}} \right\rfloor} \cdots \sum_{\alpha_2=0}^{\left\lfloor \frac{B_{total}}{\Phi_2} \right\rfloor} \sum_{\alpha_1=0}^{\left\lfloor \frac{B_{total}}{\Phi_1} \right\rfloor} C_{n_1,n_2,\ldots,n_y}^{\alpha_1,\alpha_2,\ldots,\alpha_{y-1}} P_{\alpha_1,\alpha_2,\ldots,\alpha_{y-1},\alpha_y=0}, \tag{9}$$

$$C_{n_1,n_2,\ldots,n_{y-1},n_y=0}^{\alpha_1,\alpha_2,\ldots,\alpha_{y-1}} = \begin{cases} 1, & n_1 = \alpha_1, n_2 = \alpha_2, \ldots, \text{ and } n_{y-1} = \alpha_{y-1}, \\ 0, & \text{otherwise.} \end{cases}$$

The coefficients $C_{n_1,n_2,\ldots,n_y}^{\alpha_1,\alpha_2,\ldots,\alpha_{y-1}}$ for $n_y \neq 0$ can be solved recursively. With these state probabilities, we expand (2)–(5) as follows:

$$\begin{aligned} B_{n_1,n_2,\ldots,n_y} P_{n_1,n_2,\ldots,n_y} = & \sum_{i=1}^{y} A_{n_1,n_2,\ldots,n_y}^{r_i} P_{n_1,n_2,\ldots,n_i+1,\ldots,n_y} \\ & + \sum_{i=1}^{y} A_{n_1,n_2,\ldots,n_y}^{l_i} P_{n_1,n_2,\ldots,n_i-1,\ldots,n_y} \\ & + \sum_{i=2}^{y} A_{n_1,n_2,\ldots,n_y}^{u_i} P_{n_1,n_2,\ldots,n_{i-1}-1,n_i+1,\ldots,n_y} \\ & + \sum_{i=1}^{y-1} A_{n_1,n_2,\ldots,n_y}^{d_i} P_{n_1,n_2,\ldots,n_i+1,n_{i+1}-1,\ldots,n_y}, \end{aligned} \tag{10}$$

where the coefficients $A_{n_1,n_2,\ldots,n_y}^{r_i}$, $A_{n_1,n_2,\ldots,n_y}^{l_i}, A_{n_1,n_2,\ldots,n_y}^{u_i}$, $A_{n_1,n_2,\ldots,n_y}^{d_i}$, and $B_{n_1,n_2,\ldots,n_y}$ are abbreviations of those in Eqs. (2)–(5). From Eq. (10), we derive the state probability

$$\begin{aligned} & P_{n_1,n_2,\ldots,n_{y-1},n_y+1} \\ & = \frac{B_{n_1,n_2,\ldots,n_y} P_{n_1,n_2,\ldots,n_y} - \sum_{i=1}^{y-1} A_{n_1,n_2,\ldots,n_y}^{r_i} P_{n_1,n_2,\ldots,n_i+1,\ldots,n_y}}{A_{n_1,n_2,\ldots,n_y}^{r_y}} \\ & - \frac{\sum_{i=1}^{y} A_{n_1,n_2,\ldots,n_y}^{l_i} P_{n_1,n_2,\ldots,n_i-1,\ldots,n_y}}{A_{n_1,n_2,\ldots,n_y}^{r_y}} \\ & - \frac{\sum_{i=2}^{y} A_{n_1,n_2,\ldots,n_y}^{u_i} P_{n_1,n_2,\ldots,n_{i-1}-1,n_i+1,\ldots,n_y}}{A_{n_1,n_2,\ldots,n_y}^{r_y}} \\ & - \frac{\sum_{i=1}^{y-1} A_{n_1,n_2,\ldots,n_y}^{d_i} P_{n_1,n_2,\ldots,n_i+1,n_{i+1}-1,\ldots,n_y}}{A_{n_1,n_2,\ldots,n_y}^{r_y}}. \end{aligned} \tag{11}$$

After some manipulation, Eq. (11) can be converted into the following form:

$$\begin{aligned} & P_{n_1,n_2,\ldots,n_{y-1},n_y} \\ & = \frac{B_{n_1,n_2,\ldots,n_y-1} P_{n_1,n_2,\ldots,n_y-1} - \sum_{i=1}^{y-1} A_{n_1,n_2,\ldots,n_y-1}^{r_i} P_{n_1,n_2,\ldots,n_i+1,\ldots,n_y-1}}{A_{n_1,n_2,\ldots,n_y-1}^{r_y}} \\ & - \frac{\sum_{i=1}^{y} A_{n_1,n_2,\ldots,n_y-1}^{l_i} P_{n_1,n_2,\ldots,n_i-1,\ldots,n_y-1}}{A_{n_1,n_2,\ldots,n_y-1}^{r_y}} \\ & - \frac{\sum_{i=2}^{y} A_{n_1,n_2,\ldots,n_y-1}^{u_i} P_{n_1,n_2,\ldots,n_{i-1}-1,n_i+1,\ldots,n_y-1}}{A_{n_1,n_2,\ldots,n_y-1}^{r_y}} \\ & - \frac{\sum_{i=1}^{y-1} A_{n_1,n_2,\ldots,n_y-1}^{d_i} P_{n_1,n_2,\ldots,n_i+1,n_{i+1}-1,\ldots,n_y-1}}{A_{n_1,n_2,\ldots,n_y-1}^{r_y}}. \end{aligned} \tag{12}$$

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 Eqs. (9) and (12), we can obtain:

$$\begin{aligned} & C_{n_1,n_2,\ldots,n_y}^{\alpha_1,\alpha_2,\ldots,\alpha_{y-1}} \\ & = \frac{B_{n_1,n_2,\ldots,n_y-1} C_{n_1,n_2,\ldots,n_y-1}^{\alpha_1,\alpha_2,\ldots,\alpha_{y-1}} - \sum_{i=1}^{y-1} A_{n_1,n_2,\ldots,n_y-1}^{r_i} C_{n_1,n_2,\ldots,n_i+1,\ldots,n_y-1}^{\alpha_1,\alpha_2,\ldots,\alpha_{y-1}}}{A_{n_1,n_2,\ldots,n_y-1}^{r_y}} \\ & - \frac{\sum_{i=1}^{y} A_{n_1,n_2,\ldots,n_y-1}^{l_i} C_{n_1,n_2,\ldots,n_i-1,\ldots,n_y-1}^{\alpha_1,\alpha_2,\ldots,\alpha_{y-1}}}{A_{n_1,n_2,\ldots,n_y-1}^{r_y}} \\ & - \frac{\sum_{i=2}^{y} A_{n_1,n_2,\ldots,n_y-1}^{u_i} C_{n_1,n_2,\ldots,n_{i-1}-1,n_i+1,\ldots,n_y-1}^{\alpha_1,\alpha_2,\ldots,\alpha_{y-1}}}{A_{n_1,n_2,\ldots,n_y-1}^{r_y}} \\ & - \frac{\sum_{i=1}^{y-1} A_{n_1,n_2,\ldots,n_y-1}^{d_i} C_{n_1,n_2,\ldots,n_i+1,n_{i+1}-1,\ldots,n_y-1}^{\alpha_1,\alpha_2,\ldots,\alpha_{y-1}}}{A_{n_1,n_2,\ldots,n_y-1}^{r_y}} \end{aligned} \tag{13}$$

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

After obtaining all coefficients $C_{n_1,n_2,\ldots,n_y}^{\alpha_1,\alpha_2,\ldots,\alpha_{y-1}}$, the probabilities of boundaries can be derived by solving the remaining unused $\left\lfloor \frac{B_{total}}{\Phi_1} \right\rfloor \times \left\lfloor \frac{B_{total}}{\Phi_2} \right\rfloor \times \cdots \times \left\lfloor \frac{B_{total}}{\Phi_{y-1}} \right\rfloor$ independent equations in Eq. (10) as well as the normalizing condition:

$$\sum_{n_1} \sum_{n_2} \cdots \sum_{n_y} P_{n_1,n_2,\ldots,n_y} = 1. \tag{14}$$

Having solved the boundaries, all steady-state probabilities $P_{n_1,n_2,\ldots,n_y}$ can be determined from (9). Thus, P_b, P_d, and P_{td} can then be derived.

5. Simulation results

To verify the correctness and applicability of the proposed algorithm, an event-driven C++ simulator is developed. Unless otherwise stated, the following assumptions are made in our simulation. (1) The same call arrival model, call holding time, and call residence time as specified in Section 4 are used in the simulation. (2) Parameters of IEEE 802.11b and 802.11e are used. (3) We set TXOP limit to zero for four ACs, which means that a QSTA can only transmit one packet in each successful contention. (4) The communication channel is assumed to be error-free. (5) No RTS/CTS is used. (6) The BI = 500 ms. (7) For AC_VO traffics, we set CWmin to 7, CWmax to 15, and AIFSN[AC_VO] to 2. For AC_BE traffics, we set CWmin = 31, CWmax = 1023, and AIFSN[AC_BE]=3. (8) Since QAP is more likely to be the performance bottleneck, we give it a higher priority by setting its TXOP[AC_VO] to N packets where N is the expected number of voice QSTAs in the QAP. (9) In QSTAs, each AC has a queue of size 50. (10) A voice packet will be dropped when it is buffered in a queue for more than 100 ms. (11) A packet can be re-transmitted at most 3 times. (12) G.726 with 32 kbps is used as the voice source. The simulated events include call arrival/departure/handoff and rate change of QSTAs. The rate change of QSTAs follows the state transition in Fig. 13. The offered network load is defined as $\rho = (\lambda_n + \lambda_h)/(\mu_r + \mu_h)$. To reach steady states, each simulation case is run with one million arrivals. The performance metrics are new call blocking rate, handoff call dropping rate, channel utilization, and the average voice packet dropping rate, P_d, where the channel utilization is defined as $\frac{1-B_{free}}{B_{total}} \times 100\%$, not the same as the channel utilization in [11]. The quality of a voice call is considered to be acceptable if its $P_d < 2\%$; otherwise, it is considered unacceptable.

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 s. P_r is set to 0.8 when $B_{deg} \leqslant (B_{total} - B_{th})$. Fig. 16 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. So, both our analytical and simulation results are correct and believable.

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 $B_{deg} \leqslant (B_{total} - B_{th})$. 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. 17(a) shows the channel utilization under different offered loads. Clearly, our scheme has very good 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. 17(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 $\rho \geqslant 32$, the medium 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.

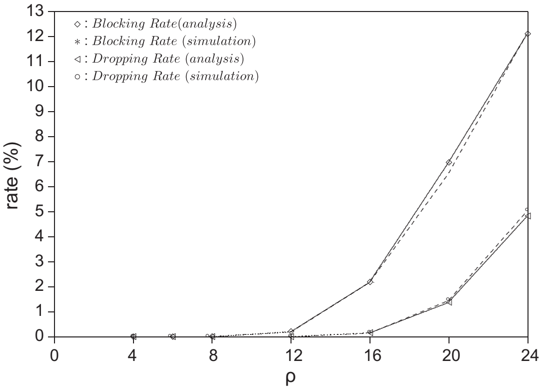


Fig. 16. Comparison of simulation and analytical results on blocking rate and dropping rate ($\lambda_h = 0.8$, $\lambda_n = 1.2$).

Fig. 18 shows the average voice packet dropping rate, P_d, against different ρ for both CAC + RA and no-CAC-no-RA cases. Without CAC, the no-CAC-no-RA curve rapidly exceeds the 2% threshold, which means an unacceptable voice quality. With CAC, our algorithm has a very small P_d because the resource usage is well under control. This result shows that CAC is definitely necessary, or the quality of real-time services will easily become unacceptable.

Fig. 19 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. 19(a), we see that our scheme is only slightly worse than the CAC-only (PI = 40) case after $\rho \geqslant 12$ because of our call acceptance policy. However, the benefit is our lower handoff call dropping rate, as shown in Fig. 19(b).

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. 20 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.

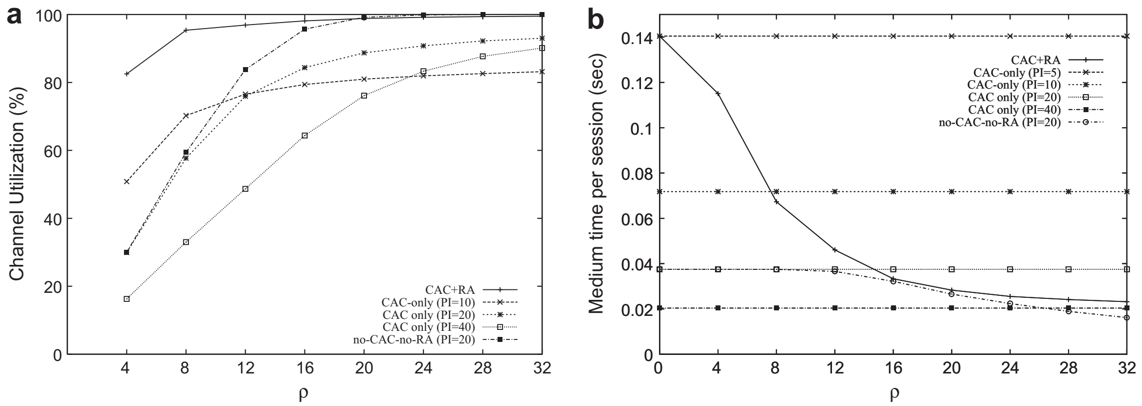


Fig. 17. Comparisons of different schemes on: (a) channel utilization and (b) goodput.

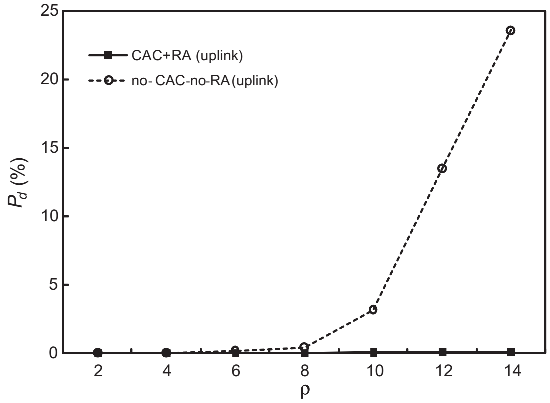


Fig. 18. Comparison of different schemes on their average voice packet dropping rates (P_d).

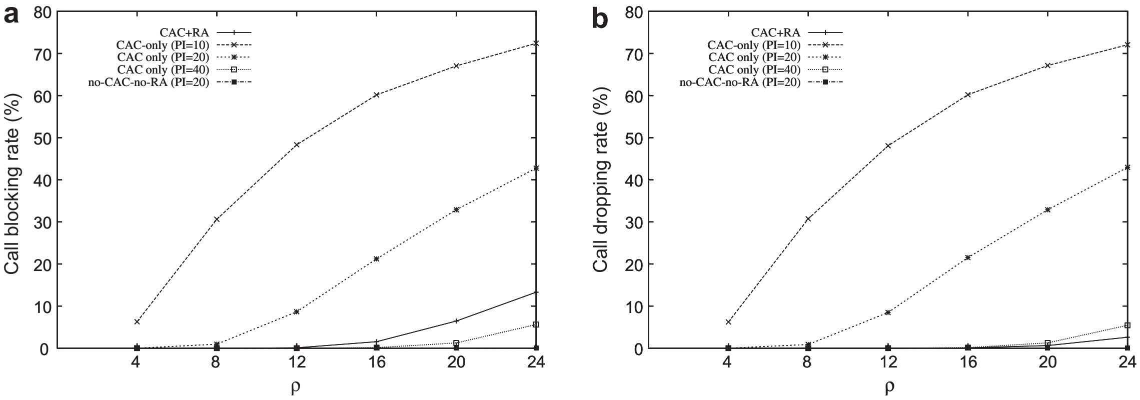


Fig. 19. Comparisons of: (a) call blocking rate and (b) call dropping rate.

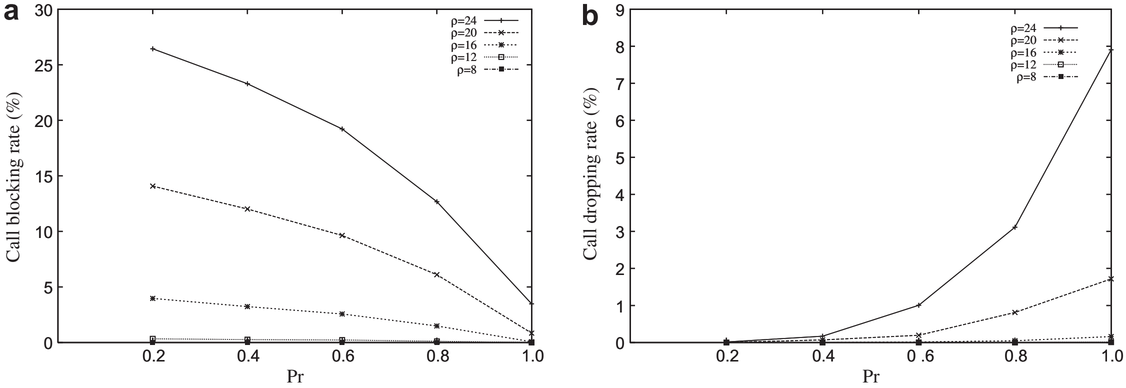


Fig. 20. The impact of P_r on: (a) call blocking rate and (b) call dropping rate.

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. 21 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.

5.5. Influence of additional BE traffics

The above experiments all assume that VoIP is the only traffic in the network. In this experiment, we add some additional static QSTAs, each generating best-effort data

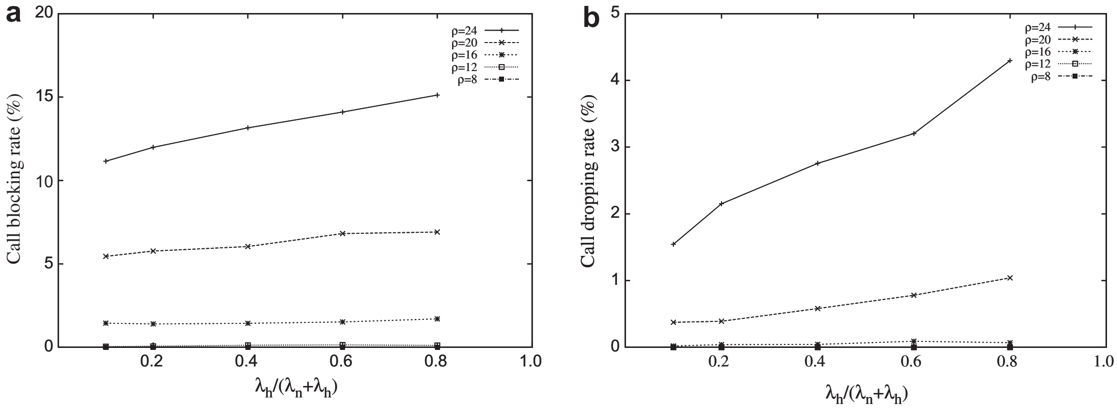


Fig. 21. The impact of the percentage of handoff calls on: (a) call blocking rate and (b) call dropping rate.

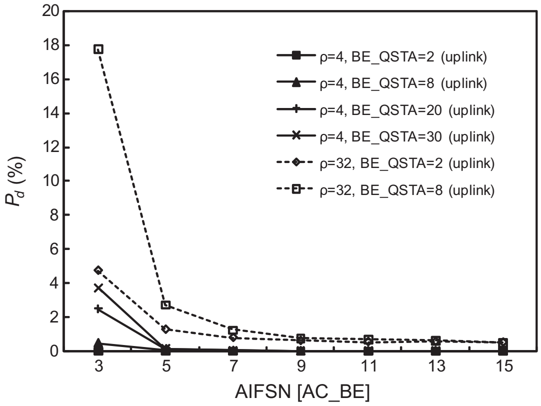


Fig. 22. Impact of interference from AC_BE traffics with various AIFSNs for AC_BE traffics.

(AC_BE) with a rate of 480 kbps to compete with VoIP traffics. As we can see in Fig. 22, if AIFSN[AC_BE] is too small (such as 3), the AC_BE traffics will significantly affect the performance of VoIP traffics. When the voice load is high (such as $\rho = 32$), the value of P_d is much too high. Even in the light load case ($\rho = 4$), we see $P_d > 2\%$ when there are more than 20 BE streams. The effect can be reduced by enlarging the value of AIFSN[AC_BE]. As shown in Fig. 22, with a slightly larger AIFSN for AC_BE, the dropping rates for voice packets are significantly reduced. When $\rho = 32$ and AIFSN[AC_BE] = 15, $P_d = 0.50\%$ and 0.53% for BE_QSTA = 2 and 8, respectively, which are improved by 89.46% and 97%, respectively, as opposed to AIFSN[AC_BE] = 3. This also implies that AIFSN can effectively help differentiate the priorities of voice and best-effort packets.

6. Conclusions

In this paper, we have proposed a CAC and an RA mechanisms to solve the handoff and physical rate adaptation issues over IEEE 802.11e multi-rate wireless networks. Our main approach is to dynamically upgrade/degrade the PIs of calls. Enlarging PIs helps reduce traffics of a QAP, but does not reduce the actual payload that a station can share. We have also derived an analytical model to evaluate our system. 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. Our results should be quite useful for QAPs which are designed to support WiFi phones. One possible future direction is to extend our current rate transition model, which is memoryless, to a more realistic one.

Acknowledgments

Y.-C. Tseng's research is co-sponsored by MoE ATU Plan, by NSC grants 97-3114-E-009-001, 97-2221-E-009-142-MY3, 98-2219-E-009-019, and 98-2219-E-009-005, by ITRI, Taiwan, by III, Taiwan, by D-Link, and by Intel.

References

[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, November 2005.

[2] S. Garg, M. Kappes, Can I add a VoIP Call?, in: Proceedings of the IEEE International Conference on Communications (ICC 2003), vol. 2, 2003, pp. 779–783.

[3] D.P. Hole, F.A. Tobagi, Capacity of an IEEE 802.11b wireless LAN supporting VoIP, in: Proceedings of the IEEE International Conference on Communications (ICC 2004), vol. 1, 2004, pp. 196–201.

[4] W. Wang, S.C. Liew, V.O.K. Li, Solutions to performance problems in VoIP over 802.11 wireless LAN, IEEE Transactions on Vehicular Technology (2005).

[5] M. Li, B. Prabhakaran, S. Sathyamurthy, On flow reservation and admission control for distributed scheduling strategies in IEEE802.11 wireless LAN, in: Proceedings of the 6th ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems, 2003, pp. 108–115.

[6] S. Garg, M. Kappes, Admission control for VoIP traffic in IEEE 802.11 networks?, in: Proceedings of the IEEE GLOBECOM), vol. 6, 2003, pp. 3514–3518.

[7] Y. Xiao, H. Li, S. Choi, Protection and guarantee for voice and video traffic in IEEE 802.11e wireless LANs, in: Proceedings of the IEEE INFOCOM'04, vol. 3, 2004, pp. 2152–2162.

[8] D. Gu, J. Zhang, A new measurement-based admission control method for IEEE802.11 wireless local area networks, in: Proceedings of the IEEE Personal, Indoor and Mobile Radio Communications (PIMRC 2003), vol. 3, 2003, pp. 2009–2013.

[9] Z. Chen, L. Wang, F. Zhang, X. Wang, W. Chen, VoIP over WLANs by adapting transmitting interval and call admission control, in: Proceedings of the IEEE International Conference on Communications (ICC 2008), 2008, pp. 3242–3246.

[10] S. Oh, J. Shin, D. Kwak, C. Kim, A novel call admission control scheme for the IEEE 802.11e EDCA, in: Proceedings of the 10th International Conference on Advanced Communication Technology (ICACT 2008), vol. 3, 2008, pp. 1832–1835.

[11] H. Zhai, X. Chen, Y. Fang, A call admission and rate control scheme for multimedia support over IEEE 802.11 wireless LANs, Wireless Networks 12 (4) (2006) 451–463.

[12] P. Wang, H. Jiang, W. Zhuang, IEEE 802.11e enhancement for voice service, IEEE Wireless Communications 13 (1) (2006) 30–35.

[13] C. Li, J. Li, X. Cai, A novel self-adaptive transmission scheme over an IEEE 802.11 WLAN for supporting multi-service, Wireless Communications and Mobile Computing 6 (4) (2006) 467–474.

[14] P. Wang, H. Jiang, W. Zhuang, Capacity improvement and analysis for voice/data traffic over WLANs, IEEE Transactions on Wireless Communications 6 (4) (2007) 1530–1541.

[15] D. Pong, T. Moors, Call admission control for IEEE 802.11 contention access mechanism, in: Proceedings of the IEEE GLOMECOM, vol. 1, 2003, pp. 174–178.

[16] A. Banchs, X. Perez-Costa, D. Qiao, Providing throughput guarantees in IEEE 802.11e wireless LANs, in: Proceedings of the 18th International Teletraffic Conference, 2003.

[17] N. Hegde, A. Proutiere, J. Roberts, Evaluating the voice capacity of 802.11 WLAN under distributed control, in: Proceedings of the 14th IEEE Workshop on Local and Metropolitan Area Networks, 2005.

[18] N. Nasser, Adaptability enhanced framework for provisioning connection-level QoS in multimedia wireless networks, in: IEEE and IFIP International Conference on Wireless and Optical Communications Networks (WOCN), March 2005, pp. 275–279.

[19] C.-T. Chou, K.-G. Shin, Analysis of adaptive bandwidth allocation in wireless networks with multilevel degradable quality of service, IEEE Transactions on Mobile Computing 3 (1) (2004) 5–17.

[20] D. Chen, A.K. Elhakeem, X. Wang, A novel call admission control in multi-service wireless LANs, in: Proceedings of the 3rd International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WIOPT 2005), 2005, pp. 119–128.

[21] P.-Y. Wu, J.-J. Chen, Y.-C. Tseng, H. Lee, Design of QoS and admission control for VoIP services over IEEE 802.11e WLANs, Journal of Information Science and Engineering 24 (4) (2008) 1003–1022.

[22] G. Karlsson, Asynchronous transfer of video, IEEE Communication Magazine 11 (Aug) (1996) 118–126.

[23] M. Handley, V. Jacobson, SDP: session description protocol, RFC2327, in: IETF, 1998.

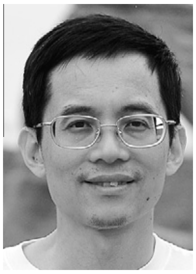
[24] Recommended Practice for Multi-Vendor Acess Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation, IEEE Draft 802.1f/Final Version, January 2003.

[25] Y.-C. Kim, D.-E. Lee, B.-J. Lee, Y.-S. Kim, B. Mukherjee, Dynamic channel reservation based on mobility in wireless ATM, in: Proceedings of the IEEE wmATM'99, 1999, pp. 100–106.

[26] U. Herzog, L. Woo, K. Chandy, Solution of queueing problems by a recursive technique, IBM Journal of Research and Development 19 (May) (1975) 295–300.

Jen-Jee Chen received his B.S. and M.S. degrees in Computer Science and Information Engineering from the National Chiao Tung University, Taiwan, in 2001 and 2003, respectively. He was a Visiting Scholar at the University of Illinois at Urbana-Champaign during the 2007–2008 academic year. Then, he obtained his Ph.D. in Computer Science from the National Chiao Tung University, Taiwan, in October of 2009. He was a postdoctoral research fellow (2010–2011) at the Department of Electrical Engineering, National Chiao Tung University, Taiwan. He is Assistant Professor (2011-present) at the Department of Electrical Engineering, National University of Tainan, Taiwan. His research interests include wireless communications and networks, personal communications, mobile computing, cross-layer design, and Internet communication service. Dr. Chen is a member of the IEEE and the Phi Tau Phi Society.

Ling Lee received her B.S. and M.S. degrees in Computer Science and Information Engineering from the National Chiao Tung University in 2004 and 2006, respectively. Her research interests include wireless VoIP, QoS, and WiMAX. She is currently working at Yahoo!, Taiwan as a software engineer.



Yu-Chee Tseng got his Ph.D. in Computer and Information Science from the Ohio State University in January of 1994. He is/was Professor (2000-present), Chairman (2005–2009), and Associate Dean (2007-present), Department of Computer Science, National Chiao-Tung University, Taiwan, and Chair Professor, Chung Yuan Christian University (2006–2010).

Dr. Tseng received Outstanding Research Award (National Science Council, 2001, 2003, and 2009), Best Paper Award (Int'l Conf. on Parallel Processing, 2003), Elite I. T. Award (2004), and Distinguished Alumnus Award (Ohio State University, 2005), and Y. Z. Hsu Scientific Paper Award (2009). His research interests include mobile computing, wireless communication, and parallel and distributed computing.

Dr. Tseng serves/served on the editorial boards for *Telecommunication Systems* (2005-present), *IEEE Trans. on Vehicular Technology* (2005–2009), *IEEE Trans. on Mobile Computing* (2006-present), and *IEEE Trans. on Parallel and Distributed Systems* (2008-present).