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

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.

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)

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
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.

Variable: Octets:

<table>
<thead>
<tr>
<th>Element ID</th>
<th>Length</th>
<th>QoS Info</th>
<th>Reserved</th>
<th>AC_BE Parameters Record</th>
<th>AC_BK Parameters Record</th>
<th>AC_VI Parameters Record</th>
<th>AC_VO Parameters Record</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

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

Octets:

<table>
<thead>
<tr>
<th>Element ID</th>
<th>Length</th>
<th>TS Info</th>
<th>Nominal MSDU Size</th>
<th>Maximum MSDU Size</th>
<th>Minimum Service Interval</th>
<th>Maximum Service Interval</th>
<th>Inactivity Interval</th>
<th>Suspension Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Service Start Time</th>
<th>Minimum Data Rate</th>
<th>Mean Data Rate</th>
<th>Peak Data Rate</th>
<th>Maximum Burst Size</th>
<th>Delay Bound</th>
<th>Minimum PHY Rate</th>
<th>Surplus Bandwidth Allowance</th>
<th>Medium Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

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.

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-

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

Fig. 4. An example of SIP with SDP message bodies: (a) INVITE signal 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 supp-

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 \( \lambda \), 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 \( \lambda \) (and thus independent of which codec is used). Therefore, the system state of a QAP can be denoted as \( s = (s_1, s_2, s_3, \ldots, s_k) \), where \( s_i \) is the number of calls served at the \( i \)th 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.
- \( B_{pi} \): the threshold of occupied bandwidth (below which new calls can be accepted).
- \( B_{free} \): the current free bandwidth of the QAP.
- \( \Delta_{deg} \): the maximum available free bandwidth of the QAP after degrading all existing calls to the lowest QoS level.
- \( P_{def} \): 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,
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 information elements in (a) Probe Request and (b) Probe Response.

![Fig. 5. The proposed message flows of the CAC procedure in IEEE 802.11e networks.](image)

![Order of Information Elements](image)

![Order of Information Elements](image)

### Table 1

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

<table>
<thead>
<tr>
<th>Codec</th>
<th>Data rate (kbps)</th>
<th>Packetization interval (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>G.711</td>
<td>64</td>
<td>113</td>
</tr>
<tr>
<td>G.726</td>
<td>16</td>
<td>84</td>
</tr>
<tr>
<td>G.728</td>
<td>32</td>
<td>94</td>
</tr>
<tr>
<td>G.723.1</td>
<td>5.3</td>
<td>94</td>
</tr>
</tbody>
</table>
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 Median Time (MT) that should be reserved for the call per BI:

\[
MT(\text{codec}, \text{PI}, r) = \left( \text{total\_time\_needed\_per\_BI} \right) = \left( \frac{\text{time\_to\_send\_one\_packet}}{\text{BI/PI}} \right) 
\times \left( \frac{\text{surplus\_bandwidth\_allowance}}{r} \right) 
\times \left( \frac{\text{avg\_contention\_window\_size}}{\text{MT}} \right)
\]

where \( L(c,p) \) is the payload per packet when codec \( c \) and PI \( p \) are used, \( \text{avg\_contention\_window\_size} \) is the average contention window size seen by the QAP, and \( H_{\text{RTP}}, H_{\text{UDP}}, H_{\text{IP}}, \) and \( H_{\text{MAC}} \) are header sizes of RTP, UDP, IP, and MAC packets, respectively. The value of \( \text{avg\_contention\_window\_size} \) 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\( \text{surplus\_bandwidth\_allowance} \) 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 \( MT = \left[ \frac{154}{11} \times (50 + 70 + 192) + (10 + 248) \right] \times (1000/20) \times 1.1 = 37.51 \text{ ms} \). Each QAP will keep its maximum available free bandwidth \( B_{\text{deg}} \), which is equal to \( B_{\text{free}} \) 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 \( B_{\text{deg}} \), 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 \( r \), and we will assume \( p = P_{\text{def}} \). 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 \( B_{\text{deg}} \); otherwise, the call is rejected.
- If this is a new call, there are two cases:
  - If \( MT(c, P_{\text{max}}) < B_{\text{deg}} \) and \( B_{\text{deg}} > (B_{\text{total}} - B_{\text{th}}) \), the call is accepted directly.
  - If \( MT(c, P_{\text{max}}) < B_{\text{deg}} \) but \( B_{\text{deg}} \leq (B_{\text{total}} - B_{\text{th}}) \), 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 \( MT(c, p, r) < B_{\text{free}} \). 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 \( \text{degrade}(c, p, r) \) in Fig. 8. The function will repeatedly select an

---

**Table 2**

Parameters of IEEE 802.11e EDCA.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Size (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTP + UDP + IP header</td>
<td>40</td>
</tr>
<tr>
<td>MAC header for DATA</td>
<td>34</td>
</tr>
<tr>
<td>PHY header</td>
<td>192</td>
</tr>
<tr>
<td>ACK</td>
<td>248</td>
</tr>
<tr>
<td>DIFS</td>
<td>50</td>
</tr>
<tr>
<td>SIFS</td>
<td>10</td>
</tr>
<tr>
<td>Slot time</td>
<td>20</td>
</tr>
<tr>
<td>CW/min (for voice)</td>
<td>7</td>
</tr>
<tr>
<td>CW/max (for voice)</td>
<td>15</td>
</tr>
</tbody>
</table>
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 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 a 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
Resource reservation at QAP is done as follows. First, we compute the value of $\text{TXOPBudget}[AC_i] = 2 \times MT(c, p, r)$. If the value is non-negative, there is sufficient resource to support this call and we can set variables as follows:

\[
\begin{align*}
\text{TXOPBudget}[AC_i] &= \text{TXOPBudget}[AC_i] - 2 \times MT(c, p, r); \\
\text{TxAdDn}[AC_i][TSID] &= MT(c, p, r); \\
\text{TxAdUp}[AC_i][TSID] &= MT(c, p, r); \\
\text{TxAdDn}[AC_i] &= \text{TxAdDn}[AC_i] + \text{TxAdDn}[AC_i][TSID].
\end{align*}
\]

Up to this point, the admitted resources have been guaranteed. The QAP will reply an ADDTS Response to the caller with the $\text{Mean}_\text{Data}_\text{Rate} = p$ and $\text{Medium}_\text{Time} = MT(c, p, r)$ in TSPEC. If there is no sufficient resource, then an ADDTS Response is replied with $\text{Medium}_\text{Time} = 0$.

At the caller’s side, if an ADDTS response with a positive $\text{Medium}_\text{Time}$ is received, the QSTA will set its $\text{TxAdUp}[AC_i][TSID] = \text{Medium}_\text{Time}$, retrieves the PI in the $\text{Mean}_\text{Data}_\text{Rate}$ 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 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 $\text{degrade}(c, p, r)$ will be called if there is no sufficient resource. On events that a call departs or moves to a higher rate, the value of $B_{\text{free}}$ will be updated, and then the function $\text{upgrade}()$ 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 $\text{prev}()$ will return the previous QoS level. This is repeated until $B_{\text{free}}$ is not enough to upgrade any existing call.

### Resource Adjustment()

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

Fig. 10. The RA algorithm.
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 $\lambda_1$ and $\lambda_2$, 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

```plaintext
upgrade()
1: while (TRUE) do
2:   let $X$ be the call with the largest PI in the system;
3:   if $B_{free} \geq MT(X, codec, prev(X.PI), X.rate) - MT(M, codec, X.PI, X.rate)$ then
4:     change $X$’s PI to $prev(X.PI)$;
5:     $B_{free} = B_{free} - MT(X, codec, prev(X.PI), X.rate) + MT(M, codec, X.PI, X.rate)$;
6:   else
7:     return;
8:   end if;
9: end while;
```

Fig. 11. The bandwidth upgrade algorithm.

Fig. 12. An example of bandwidth upgrade.
time of a call is exponentially distributed with mean $1/\mu = 1/(\mu_b + \mu_r)$. The required bandwidth of a call with $P_{\text{max}}$ at the transmission rate $R_i$ is denoted by $\Phi_i$.

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 $P_{\text{max}}$). Therefore, to obtain blocking and dropping probabilities, we can assume that all calls use $P_{\text{max}}$. For simplicity, we assume that a QAP can support $y = 4$ physical rates, 11, 5.5, 2 and 1 Mbps. For a bi-directional voice call under different physical rates, as shown as 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 $\lambda_i$. The departure rate for rate $R_i$ is $n_i \mu$. For ease of presentation, we let $P_r = 1$ when $n_1 \Phi_1 + n_2 \Phi_2 + n_3 \Phi_3 + n_4 \Phi_4 < B_{\text{total}}$; otherwise, new calls are accepted with a probability $P_r$ as defined in Section 3.1. Vertical transitions are caused by transmission rate change.

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 $n_1 = n_2 = \ldots = n_y = 0$,

$$y(\lambda_b + \lambda_n)P(n_1, n_2, \ldots, n_y) = \mu \sum_{i=1}^{y-1} P(n_1, n_2, \ldots, n_i + 1, \ldots, n_y).$$

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

$$y(\lambda_b + \lambda_n) + \left( \sum_{i=1}^{y} n_i \right) \mu + \sum_{i=1}^{y-1} \left( n_i + n_{i+1} \right) \lambda_n P(n_1, n_2, \ldots, n_y)$$

$$= \sum_{i=1}^{y} \left( n_i + 1 \right) \mu P(n_1, n_2, \ldots, n_{i-1}, 1, n_{i+1}, \ldots, n_y) + \sum_{i=2}^{y} \left( n_i + 1 \right) \nu P(n_1, n_2, \ldots, n_{i-2}, 1, n_{i-1}, n_i, \ldots, n_y)$$

$$+ \sum_{i=1}^{y-1} \left( n_i + 1 \right) \nu P(n_1, n_2, \ldots, n_{i+1}, 1, n_i + 1, \ldots, n_y).$$

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

$$y(\lambda_b + P_r \lambda_n) + \left( \sum_{i=1}^{y} n_i \right) \mu + \sum_{i=1}^{y-1} \left( n_i + n_{i+1} \right) \lambda_n P(n_1, n_2, \ldots, n_y)$$

$$= \sum_{i=1}^{y} \left( n_i + 1 \right) \mu P(n_1, n_2, \ldots, n_{i-1}, 1, n_i + 1, \ldots, n_y)$$

$$+ \sum_{i=1}^{y} \left[ \lambda_i + \left( \lambda_b + P_r \lambda_n \right) \lambda_n P(n_1, n_2, \ldots, n_{i-1}, 1, n_{i+1}, \ldots, n_y)$$

$$+ \sum_{i=2}^{y} \left( n_i + 1 \right) \nu P(n_1, n_2, \ldots, n_{i-2}, 1, n_{i-1}, n_i, \ldots, n_y)$$

$$+ \sum_{i=1}^{y-1} \left( n_i + 1 \right) \nu P(n_1, n_2, \ldots, n_{i+1}, 1, n_i + 1, \ldots, n_y).$$

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

$$I_z = \begin{cases} 1, & \sum_{i=1}^{y} (n_i \Phi_i) - \Phi_z < B_{\text{total}}, \\ 0, & \text{otherwise}. \end{cases}$$
Case IV: For states such that $\sum_{i=1}^{y}(n_iF_i) \leq B_{\text{total}}$ and $\sum_{i=1}^{y}(n_iF_i) + \Phi_y \geq B_{\text{total}}$,

$$\left[\sum_{i=1}^{y} \left(\lambda_i + P_{\text{d}} \beta_i\right) + \left(\sum_{i=1}^{y} n_i\right) \mu + \left(\sum_{i=1}^{y} n_i + \frac{1}{2}\right)\right] P_{n_1,n_2,\ldots,n_y}$$

$$= \sum_{i=1}^{y} \left[\left(\sum_{n=1}^{B_{\text{total}}} P_{n_1,n_2,\ldots,n_y}\right) + \lambda_i(n_i + 1)(\mu + \lambda_i + \beta_i) \right]$$

$$+ \sum_{i=2}^{y} \left[\left(\sum_{n=1}^{B_{\text{total}}} P_{n_1,n_2,\ldots,n_y}\right) + \lambda_i(n_i + 1)\mu P_{n_1,n_2,\ldots,n_1,n_{i-1},n_{i+1}}\right]$$

$$+ \sum_{i=1}^{y} \left[\left(\sum_{n=1}^{B_{\text{total}}} P_{n_1,n_2,\ldots,n_y}\right) + \lambda_i(n_i + 1)\mu P_{n_1,n_2,\ldots,n_1,n_{i-1},n_{i+1}}\right],$$

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

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

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

$$I_{mn} = \begin{cases} 
1, & \sum_{i=1}^{y}(n_iF_i) - \Phi_m + \Phi_n \leq B_{\text{total}}, \\
0, & \text{otherwise}
\end{cases}$$

Let $P_b$ be the blocking probability of new calls, $P_d$ be the dropping probability of handoff calls, and $P_{\text{ld}}$ 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 $\tau(\bar{n}) = \sum_{i=1}^{y}(n_iF_i)$. We can derive:

$$P_b = \sum_{(\bar{n}) \in B_{\text{total}}} P_{n_1,n_2,\ldots,n_y},$$

$$P_d = \frac{1}{y} \sum_{i=1}^{y} \left(\sum_{(\bar{n}) \in B_{\text{total}}} P_{n_1,n_2,\ldots,n_y}\right),$$

$$P_{\text{ld}} = \sum_{i=1}^{y} \left(\sum_{(\bar{n}) \in B_{\text{total}}} P_{n_1,n_2,\ldots,n_y}\right).$$

To compute $P_b$, $P_d$, and $P_{\text{ld}}$ 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 $P_{n_1,n_2,\ldots,n_y}$ 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:

\[
P_{n_1, n_2, \ldots, n_y} = \sum_{x_y=0}^{n_y} \sum_{x_{y-1}=0}^{n_y} \cdots \sum_{x_0=0}^{n_y} c_{x_0, x_1, \ldots, x_{y-1}} P_{n_1, n_2, \ldots, n_y}
\]

\[
= 0, \quad \sum_{x_0=0}^{n_y} \sum_{x_{y-1}=0}^{n_y} \cdots \sum_{x_1=0}^{n_y} c_{x_0, x_1, \ldots, x_{y-1}} P_{n_1, n_2, \ldots, n_y} = \frac{\sum_{x_y=0}^{n_y} \sum_{x_{y-1}=0}^{n_y} \cdots \sum_{x_1=0}^{n_y} c_{x_0, x_1, \ldots, x_{y-1}} P_{n_1, n_2, \ldots, n_y}}{\sum_{x_y=0}^{n_y} \sum_{x_{y-1}=0}^{n_y} \cdots \sum_{x_1=0}^{n_y} c_{x_0, x_1, \ldots, x_{y-1}}}
\]

\[
= 0, \quad \sum_{x_0=0}^{n_y} \sum_{x_{y-1}=0}^{n_y} \cdots \sum_{x_1=0}^{n_y} c_{x_0, x_1, \ldots, x_{y-1}} P_{n_1, n_2, \ldots, n_y} = \frac{\sum_{x_y=0}^{n_y} \sum_{x_{y-1}=0}^{n_y} \cdots \sum_{x_1=0}^{n_y} c_{x_0, x_1, \ldots, x_{y-1}} P_{n_1, n_2, \ldots, n_y}}{\sum_{x_y=0}^{n_y} \sum_{x_{y-1}=0}^{n_y} \cdots \sum_{x_1=0}^{n_y} c_{x_0, x_1, \ldots, x_{y-1}}}
\]

The coefficients \(c_{x_0, x_1, \ldots, x_{y-1}}\) for \(n_y \neq 0\) can be solved recursively. With these state probabilities, we expand (2)–(5) as follows:

\[
B_{n_1, n_2, \ldots, n_y} P_{n_1, n_2, \ldots, n_y} = \sum_{i=1}^{y} A_{i1} P_{n_1, n_2, \ldots, n_y} P_{n_1, n_2, \ldots, n_y+1, \ldots, n_y} + \sum_{i=1}^{y} A_{i2} P_{n_1, n_2, \ldots, n_y} P_{n_1, n_2, \ldots, n_y, 1, \ldots, n_y} + \sum_{i=1}^{y} A_{i3} P_{n_1, n_2, \ldots, n_y} P_{n_1, n_2, \ldots, n_y, \ldots, n_y}
\]

where the coefficients \(A_{i1}^{x_0, x_1, \ldots, x_{y-1}}, A_{i2}^{x_0, x_1, \ldots, x_{y-1}}, A_{i3}^{x_0, x_1, \ldots, x_{y-1}},\) 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

\[
P_{n_1, n_2, \ldots, n_y, n_y+1} = \frac{B_{n_1, n_2, \ldots, n_y} P_{n_1, n_2, \ldots, n_y} - \sum_{i=1}^{y-1} A_{i1}^{x_0, x_1, \ldots, x_{y-1}} P_{n_1, n_2, \ldots, n_y+1, \ldots, n_y}}{A_{i1}^{x_0, x_1, \ldots, x_{y-1}}}
\]

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

\[
P_{n_1, n_2, \ldots, n_y, n_y+1} = \frac{B_{n_1, n_2, \ldots, n_y} P_{n_1, n_2, \ldots, n_y} - \sum_{i=1}^{y-1} A_{i1}^{x_0, x_1, \ldots, x_{y-1}} P_{n_1, n_2, \ldots, n_y+1, \ldots, n_y}}{A_{i1}^{x_0, x_1, \ldots, x_{y-1}}}
\]

For each fixed state \((x_1, x_2, \ldots, x_y, 0)\), if we let \(P_a = 1\) and \(P_d = 0\) for all \(i \neq n\), from Eqs. (9) and (12), we can obtain:

\[
C_{n_1, n_2, \ldots, n_y} = \frac{B_{n_1, n_2, \ldots, n_y} - \sum_{i=1}^{y-1} A_{i1}^{x_0, x_1, \ldots, x_{y-1}} P_{n_1, n_2, \ldots, n_y+1, \ldots, n_y}}{A_{i1}^{x_0, x_1, \ldots, x_{y-1}}}
\]

where \(P_a, P_d,\) and \(P_{id}\) 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 Bit = 500 ms. (7) For AC_VO traffic, we set CWmin to 7, CWmax to 15, and AIFS[N][AC_VO] to 2. For AC_BE traffic, we set CWmin = 31, CWmax = 1023, and AIFS[N][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 \(\rho = (\lambda + \lambda_h)/\mu_v + \mu_h\). 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, \(P_{id}\), where the channel utilization is defined as \(\frac{\text{Traffic}}{\text{Bandwidth}} \times 100\%\), 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 $B_{deg} \leq (B_{total} - B_{th})$. 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 $B_{deg} \leq (B_{total} - B_{th})$. 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. 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 $\rho \geq 32$, 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. 18 shows the average voice packet dropping rate, $P_d$, against different $\rho$ 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 $\rho \geq 12$ 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. 16.** Comparison of simulation and analytical results on blocking rate and dropping rate ($\lambda_0 = 0.8$, $\lambda_H = 1.2$).

**Fig. 17.** Comparisons of different schemes on: (a) channel utilization and (b) goodput.
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
the light load case (such as performance of VoIP traffics. When the voice load is high (such as 3), the AC_BE traffics will significantly affect the traffics. As we can see in Fig. 22, if AIFSN[AC_BE] is too small (AC_BE) with a rate of 480 kbps to compete with VoIP traffics.

![Fig. 22. Impact of interference from AC_BE traffics with various AIFSNs for AC_BE traffics.](image)

(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 ρ = 32), the value of $P_d$ is much too high. Even in the light load case (ρ = 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 ρ = 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 AIFS-N[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**
