Adaptive Quality-of-Service Provisioning in Wireless and Mobile Networks

by

Chun-Ting Chou

A dissertation submitted in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering-Systems)
in the University of Michigan
2005

Doctoral Committee:

Professor Kang G. Shin, Chair
Professor Demosthenis Teneketzis
Associate Professor Brian Noble
Assistant Professor Achilleas Anastasopoulos
Assistant Professor Mingyan Liu

ABSTRACT

Adaptive Quality-of-Service Provisioning in Wireless and Mobile Networks

by

Chun-Ting Chou

Chair: Kang G. Shin

The problem of adaptive QoS provisioning in wireless and mobile networks is studied in this thesis. A mathematical model is established to analyze the impact of adaptive bandwidth allocation on both system performance and user-perceived QoS. With this model, network service providers can dynamically adjust — based on the network load or available network capacity — user bandwidth with *controllable* degradation on user-perceived QoS. To facilitate adaptive QoS support in time-division multiplexed wireless networks (such as the IEEE 802.11 wireless LANs), a distributed airtime usage control is also developed. By using the proposed airtime control, wireless stations using the contention-based medium access method are shown to be able to provide users the parameterized QoS, which can only be achieved by using the polling-based medium access method in the current IEEE 802.11e standard. The proposed distributed airtime usage control is also shown to be able to provide QoS support in ad hoc IEEE 802.11 wireless LANs.

In order to further improve the user's QoS, the concept of "spectral agility" is introduced to wireless networks (especially, the IEEE 802.11 wireless LANs). An analytical model is established in order to derive the achievable improvement gained by using spectral agility. To fully exploit spectral agility, a comprehensive framework for spectral-agile networks is also developed. This framework and the associated functionalities are integrated with the IEEE 802.11 wireless LAN in the *ns*-2 simulator to demonstrate the effectiveness of the resulting spectral-agile wireless networks. Finally, the mobility support for QoS provisioning in the IEEE 802.11 wireless LAN is investigated, and a unified smooth-and-fast handoff is developed for both intra- and inter-subnet handoffs based on the Inter-Access Point Protocol.

To my dear Mom

ACKNOWLEDGMENTS

Many individuals have contributed to this thesis by giving me their support during my doctoral studies. First of all, I would like to express my deepest gratitude to Professor Kang G. Shin. As my research advisor, he has provided constant encouragement and invaluable suggestions, which help me not only complete this thesis but also prepare for my career in a long time to come. I also would like to thank Professors Demosthenis Teneketzis, Brian Noble, Achilleas Anastasopoulos, and Mingyan Liu for serving on my diseertation committee.

I am also grateful to many members of the Real-Time Computing Laboratory, especially, Daji Qiao, Hani Jamjoom, Mohamad El-Gendy, Jian Wu, Chang-Hao Tsai, KyuHan Kim, Katharine Chang, and Hyoil Kim for their friendship and advice. Thanks also go to Drs. Sai Shankar and Stefan Mangold of Philips Research USA for their valuable suggestions and discussions.

My special thanks go to my family since they have always believed in me and shown me their unconditional love and support. My final acknowledgement goes to my dear girl friend Annie for her understanding, encouragement and companion during my study.

CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

CHAPTER 1

Introduction

Over the last decade, wireless communication has evolved from the synonym of cellular phone service to an integrated audio/video/data service. Such evolution is driven by not only new hardware/software development but also the increasing dependence of human's daily life on wireless communication. For example, people expect to use smart phones for all personal communication needs, to maintain ubiquitous connections to corporate/enterprise networks at work, or to establish a wireless entertainment network at home. To satisfy these diverse demands for wireless communication, the next-generation wireless network has to provide users/applications certain Quality-of-Service (QoS).

The main task of providing QoS guarantees is to ensure that users' requirements are satisfied throughout the entire service period. The most common QoS requirements include the minimum/maximum throughput, delay bound or delay jitter, and packet loss rate. Unlink the best-effort service, service with these QoS requirements calls for integrated support from the content servers, the core network (e.g., the Internet) and the wireless access network, with each relying on different mechanisms for service differentiation, resource reservation or admission control.

Among these, supporting QoS in a wireless network is more difficult than in its wired counterparts. First, the radio is a very limited and precious resource. Although new modulation, coding or medium access schemes allow more efficient utilization of the radio resource, these improvements cannot keep pace with the explosive growth of bandwidth-demanding applications. Second, users of wireless/mobile networks may not keep connected via a fixed attachment point (e.g., an access point) due to user mobility. Therefore, users may experience unpredictable disconnection from the core

network while they are moving, hence resulting in service disruption. Because of these two unique properties, providing *absolute* QoS guarantees in wireless/mobile networks is very difficult, if not impossible.

Adaptive QoS has been considered as the only option to the problem of QoS provisioning in wireless/mobile networks. The key idea of adaptive QoS is to provide users the QoS that is adapted to (1) network conditions such as the network load or available network capacity, and (2) individual users' characteristics such as the physical-layer parameters. Unlike the case of absolute QoS guarantees, adaptive QoS may require certain degradation of users' performance—within a tolerable range—so that the aforementioned adaptivity can be applied to improve both the network and users' performance. In this thesis, we study the problem of adaptive QoS provisioning in wireless/mobile networks and investigate its impact on both system utilization and individual users' QoS.

1.1 Overview

The problem of adaptive QoS provisioning is divided into three parts: (1) adaptive resource allocation, (2) opportunistic resource utilization, and (3) user mobility support. The problems and objectives of each part are outlined as follows.

- *Adaptive Resource Allocation*: The key idea is to adapt the allocation of system resource to network conditions and user characteristics. Thus, the system resource can be utilized more efficiently while individual users can still receive acceptable QoS. We focus on the problems of (1) how much bandwidth to be allocated to users based on network capacity and users' QoS requirements, and (2) how to realize the bandwidth allocation given by the answer of (1) via an efficient medium access control (MAC).
- *Opportunistic Resource Utilization*: Since a wireless network's capability of providing QoS is determined by its transmission capacity, an effective method to enhance the QoS is to increase the network's operating bandwidth. Spectral agility is introduced to wireless networks for this purpose. With spectral agility, a wireless network can locate radio resources — in time, frequency and space

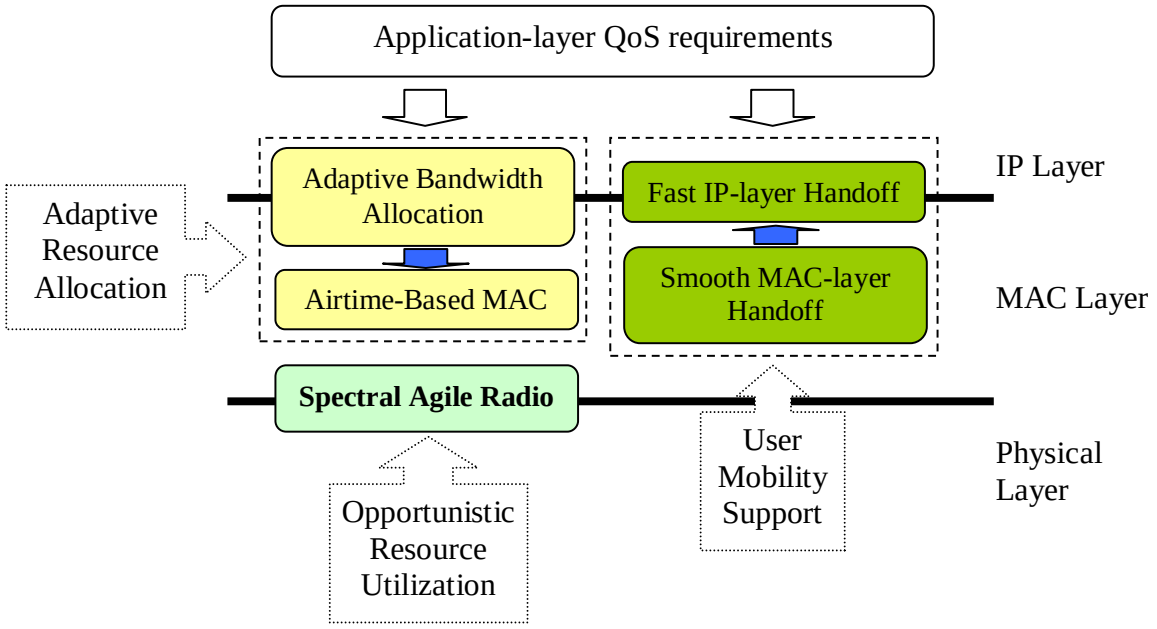


Figure 1.1. The system architecture for adaptive QoS provisioning in wireless networks

domains — and utilize them in an opportunistic way. We analyze the performance of wireless networks with spectral agility, and develop the network architecture and protocols to exploit the potential of spectral agility.

- *User Mobility Support*: Handoffs due to user mobility disconnect the users from their access points. The main goal here is to hide such disconnections, which may cause handoff latency or packet losses, from the users or applications. The concept of cross-layer — between the MAC and IP layers — optimization is applied to improve both IP- and MAC-layer handoffs.

The relation between these three parts in the OSI protocol stack is illustrated in Figure 1.1.

By solving the problems in each part, the adaptive QoS provisioning in wireless networks can be achieved as follows. First, an initial amount of bandwidth is assigned to each user according to his QoS requirement and network capacity. This bandwidth allocation may only be made possible by degrading other users' QoS, especially when the network is heavily-loaded. Once the user is admitted into the network, the assigned bandwidth may be adjusted when the network load changes or when more radio resource becomes available if spectral agility is used. The bandwidth adjust-

ments are made by means of users' medium access control so as to ensure that each user uses the resource properly. Finally, the fast handoff mechanism coordinated by the access points helps minimize users' service disruption due to mobility.

The system architecture to support the aforementioned adaptive QoS is shown in Figure 1.1. It is composed of four building blocks: (1) adaptive bandwidth allocation (2) airtime-based medium access control, (3) spectral-agile radio, and (4) unified smooth-and-fast handoff, with each performing its tasks as follows.

- *Adaptive Bandwidth Allocation* dynamically adjusts the system bandwidth constellation to improve the network utilization, and to provide users QoS with controlled service degradation.
- *Airtime-based Medium Access Control* allocates proportional transmission times to users — in a distributed and autonomous manner — so as to provide users the bandwidth determined by the *Adaptive Bandwidth Allocation*.
- *Spectral-Agile Radio* seeks available spectral resources, provides the *Adaptive Bandwidth Allocation* more radio resources for better user QoS, and coordinates the use of the available spectral resources with other radio devices/systems.
- *Unified Smooth-and-Fast Handoff* provides a unified mechanism for smooth and fast MAC-layer (i.e., intra-subnet) and IP-layer (i.e., inter-subnet) handoffs, based on a smooth MAC-layer handoff mechanism.

1.2 Related Work

Various approaches and algorithms adopting the idea of adaptive bandwidth allocation have been proposed. A graceful degradation mechanism was proposed to increase bandwidth utilization by dynamically adjusting the bandwidth allocation based on user-specified QoS profiles [1]. Sen *et al.* [2] proposed an optimal degradation algorithm to maximize their revenue function. Sherif *et al.* [3] proposed an adaptive resource allocation algorithm to maximize bandwidth utilization and to maintain fairness by means of a generic algorithm. To analyze individual users' QoS, Kwon *et al.* [4] derived a *degradation period ratio* to represent the average time a user stays in the degraded quality-level. To our best knowledge, this is the only analytical model

to investigate the impact of quality degradation on individual users' QoS. Some other algorithms also considered user mobility, by differentiating new and handoff users, when determining users' bandwidth allocation. Lin *et al.* [6] proposed an analytical model for a so-called *Guard Channel* system where a portion of bandwidth is reserved for handoff users while in [10, 11], handoff users have a higher probability to be accepted once the network load exceeds some pre-defined threshold. Other algorithms treat new and handoff users equally but estimate the traffic loads of the adjacent cells [9], or the handoff rates from the adjacent cells [7] to decide the amount of bandwidth to be reserved.

The bandwidth determined above must be allocated to users by allocating a proportional amount of transmission time, either with the help of scheduling algorithms or medium access control (MAC) mechanisms. Different scheduling algorithms, originally designed for wired networks [62, 63, 64], have been adapted to wireless networks. For example, the self-clocked fair queueing (SCFQ) was modified so that it can work in a distributed environment [21, 72]. Some other distributed scheduling algorithms have also been proposed based on the random backoff mechanism of the IEEE 802.11 wireless standard [22, 23]. It has also been shown that individual users/stations can acquire a proportional amount of transmission time by choosing different distributed-coordination-function (DCF) parameters in the IEEE 802.11 wireless standard, such as contention window size or inter-frame space (IFS) [81, 82]. A Markovian model that takes into account both the IFS and contention window size was proposed to determine the corresponding DCF parameters [83]. The problem with this model is the scalability of the resulting 3-dimensional Markovian chain. A lightweight Markovian model based on [28] was also proposed [84], but neither of them considered the reset mechanism of contention window size in the IEEE 802.11 wireless standard. In [53], an opportunistic auto rate (OAR) protocol was proposed to maintain an equal share of transmission time by controlling the "More Fragment" bit in the header of a multi-rate IEEE 802.11 wireless LAN, but allocating a proportional time can also be achieved similarly to [81].

Since transmissions via the wireless medium are more vulnerable than those via

wired media, scheduling algorithms or MAC mechanisms in wireless networks must also take transmission errors into account. WPS [68, 69], CIF-Q [24] and CSDPS [70] addressed the inefficiency/unfairness problems (resulting from transmission errors) by deferring the transmission of error-prone users/flows and compensating them after the transmission condition improves. A long-term fairness server was proposed to reduce the impact of compensation mechanisms on error-free user/flows [25]. Adaptive weights were also used to dynamically adjust the weights of error-prone users/flows to compensate for their throughput losses. The *power factor* [71] and *compensation index* [72] are the main control parameters to adjust the weights for compensation without degrading error-free flows too much.

The advances in software defined radios (SDRs) [87, 88] have stimulated the development of flexible and powerful radio interfaces to support spectral agility, which has recently drawn considerable attention for its potential to improve spectral efficiency. For example, the US Federal Communications Commission (FCC) has issued a Notice of Public Rulemaking and Order regarding *cognitive radio* technologies [79]. The Defense Advanced Research Projects Agency (DARPA) has also started the neXt Generation (XG) Communications Program to develop new technologies which allow multiple users to share the spectrum through adaptive mechanisms [80]. The US Army has also been exploring the so-called "Adaptive Spectrum Exploitation" (ASE) for real-time spectrum management in the battlefield [85, 86]. Although the focuses of these programs are somewhat different, their basic principles are the same: if radio devices can explore the wireless spectrum and locate sparsely-used spectral bands, they can exploit them opportunistically to improve not only the devices' performance but also the overall spectrum utilization.

There is a significant amount of research into supporting user mobility in wireless networks. For example, basic support of IP-layer mobility such as MobileIPv4 and MobileIPv6 has been proposed [96, 97]. A hierarchical foreign agent scheme for micro mobility in MobileIP networks was proposed [40] to confine the binding update within the local domain so that the signaling overhead and binding delay can be reduced. To further reduce the handoff latency in MobileIP networks, different

"fast-handoff" schemes using link-layer indications have also been proposed using the "swiftness" of link-layer handoff processes [98]. For example, some fast-handoff schemes used link-layer indications to initiate a MobileIP binding update (i.e., the handoff process) [49, 39], [99]-[104]. Therefore, instead of relying on the original MobileIP movement detection mechanism, users/stations can initiate the IP-layer handoff procedures much earlier. Some other schemes used link-layer indications to "skip" the IP-layer handoffs. In [106], a bi-directional edge tunnel (BET) is established between the current and new MobileIP mobility agents once the link-layer handoff indicates an upcoming IP-layer handoff. The packets destined for the current mobility agent can then be forwarded to the new agent via this BET such that the wireless station can receive the packets without executing any IP-layer handoff procedure. A handoff-dedicated link-layer bridge was also used to skip the IP-layer handoffs [110]. This bridge only forwards link-layer frames with destination MAC addresses already registered in its filtering database. After a station completes a link-layer handoff, an update frame is sent to this bridge to update the filtering database. The packets being sent to the old mobility agent can then be forwarded to the new agent via the link-layer bridge.

Since a handoff also causes packet losses, many proposals focused on how to eliminate packet losses during a handoff to achieve a smooth handoff. Smooth handoffs can be realized in many ways. For example, multicast was used to support a smooth handoff [37, 38, 30, 50]. The idea is to multicast packets to some/all neighboring mobility agents so that the packets — which may get lost during a handoff — are ready to be sent to users/stations via the new mobility agent, once the handoff is completed. Some smooth handoff schemes adopted a simple packet buffering-and-forwarding technique to achieve a smooth handoff [39, 40, 41]. Some other schemes concealed the packet losses from upper-layer applications, instead of eliminating them, to realize a smooth handoff (from users' perspective). Many proposals adopted this idea to enhance the TCP performance in wirless/mobile networks. Indirect-TCP [29] and Snoop TCP [30] divided a TCP session in two separated ones such that transmission errors over the wireless link can be made invisible to the TCP sender. Delayed

duplicate ACK [31] was proposed to prevent any undue invocation of fast-retransmit, and the "persistent mode" of TCP is also used for the same purpose [36].

1.3 Contributions

The main contributions of this thesis are listed as follows.

- Established a mathematical model to analyze adaptive bandwidth allocation problems, and investigated the tradeoff between system performance and user-perceived QoS.
- Developed a distributed airtime usage control for adaptive QoS support in time-division wireless networks such as IEEE 802.11 wireless LANs. This airtime usage control also has potential for provisioning QoS in ad hoc wireless LANs.
- Analyzed the performance gain of spectral-agile communication, and developed a comprehensive framework to realize spectral-agile communication for better QoS support.
- Developed a unified, smooth-and-fast handoff scheme for both intra- and inter-subnet handoff processes to reduce QoS degradation due to user mobility.

1.4 Thesis Structure

The rest of this thesis is organized as follows. In Chapter 2, the algorithms for adaptive bandwidth allocation are introduced and a Markovian model is provided to analyze user-perceived QoS metrics, including *probability of blocking new users*, *probability of terminating handoff users*, *degradation ratio* and *upgrade/degrade frequency*. With this model, we can evaluate the effects of bandwidth-allocation algorithms on QoS provisioning, and investigate the tradeoffs between system performance and the user-perceived QoS. Chapter 3 discusses a distributed airtime allocation algorithm, which provides users differented accesses to the shared wireless medium, based on the IEEE 802.11e wireless LAN standard. A Markovian model is also established to determine the parameters needed for a precise, quantitative control on users' airtime usage. With the adaptive bandwidth allocation and distributed airtime allocation algorithms, the adaptive QoS provisioning can be realized in a distributed manner and

is discussed in Chapter 4. Chapter 5 presents the network architecture and control protocols for supporting spectral-agile wireless networks. A mathematical model is also established to provide a performance benchmark for spectral-agile communications. In Chapter 6, we generalize the spectral-agile communications in Chapter 5 to further improve the spectrum utilization. In Chapter 7, the smooth and fast handoff mechanism based on the Inter Access Point Protocol (IAPP) (i.e., the IEEE 802.11f standard [112] is proposed and evaluated. Finally, the conclusions and future work are discussed in Chapter 8.

CHAPTER 2

Adaptive Bandwidth Allocation

Dynamic or adaptive bandwidth allocation has been shown to be an effective solution of provisioning QoS in wireless networks. In contrast to the static allocation which gives each user a fixed amount of bandwidth, the adaptive allocation dynamically adjusts user bandwidth based on the underlying network condition. By using adaptive bandwidth allocation, service providers can release some of existing users' bandwidth for new users when the network is heavily-loaded, so that more users can be served with acceptable QoS. On the other hand, if more capacity is added to the network (e.g., via spectral-agile radio), the service providers can distribute the extra capacity to all existing users.

From the service provider's perspective, using adaptive bandwidth allocation can reduce the probability of blocking new users, and achieve a higher resource utilization. However, the service perceived by individual users is not necessarily improved. For example, some users are forced to accept degraded service due to bandwidth reduction made by the adaptive allocation. Although the user receives a service upgrade when the network load is reduced or the network capacity is increased, such a service upgrade may be undesirable if the users end up with switching between degraded/upgarded service very frequently. Take audio streaming as an example. A steady and slightly poor-quality audio connection should be more desirable to the users than an unsteady, higher-quality audio. Therefore, the real challenges of using adaptive allocation for QoS provisioning are to understand its effects on user-perceived QoS, and then to control these effects within the user's acceptable range.

In this chapter, we propose an adaptive bandwidth allocation scheme with integrated admission control for generic wireless networks. An analytical model for

the proposed scheme is developed, and four QoS metrics, namely, blocking probability, forced-termination probability, degradation ratio, and upgrade/degrade frequency are derived mathematically. By using these four metrics, we can quantify the advanatages/disadvantages of using adaptive bandwidth allocation, and investigate the tradeoff between system and user-perceived performances. Based on these findings, we can tailor the proposed adaptive allocation algorithms for different networks and users' QoS profiles.

2.1 System Model and Assumptions

We consider a generic wireless cell (Figure 2.1), in which a mobile node communicates with others via a base station while residing in the cell of that base station. When a mobile node leaves a cell, it could be either successfully handed off, or dropped in case of resource shortage in the new cell. Since dropping hand-off connections is usually less desirable and less tolerable than blocking newly-initiated connections, hand-off connections are given priority over new connections. This is achieved by restricting newly-initiated connections into the system (i.e., only hand-off connections are considered to be admitted into the system), once the total number of connections exceeds a pre-specified threshold N_{thresh}. Obviously, this threshold is a design parameter, and one of our objectives in this chapter is to determine the proper value of the threshold. After admitted into the system, both hand-off and newly-initiated connections are treated equally. We assume that each connection could receive degraded service as long as this degraded service is within the user-specified QoS profile. The service requirement we are concerned here is the bandwidth requirement. We assume that each connection can receive one of the K service levels. The bandwidth requirement of the i-th service level is denoted as W_i (in units of channels),[1] and we assume $W_1 = W_{min} < W_i < W_{max} = W_K$. Therefore, once the total required channels exceed the cell capacity, the system may try to degrade the QoS level of some ongoing connections in order to admit more (both new and hand-off) connections,

[1]A channel can be a specific frequency band in a Frequency-Division-Multiplexing or a time slot in Time-Division-Multiplexing system

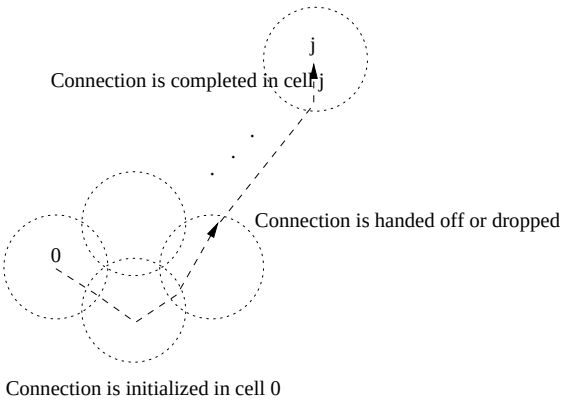


Figure 2.1. A generic wireless network

hence achieving high bandwidth utilization and also reducing the blocking and/or forced-termination probability.

In a system with degradable service, a connection may receive different QoS levels, depending on the system load or capacity during its connection lifetime. Even though a connection receives the maximal QoS level upon its arrival, it may be degraded when the system tries to accept more connections. From the users' perspectives, this may raise two important questions: (1) how long does it stay at each individual QoS level? and (2) how often does the received service switch between these QoS levels ? Even though these two questions are inter-related, the first question does not necessarily imply the second, or vice versa. Therefore, two performance metrics associated with these questions, degradation ratio and upgrade/degrade frequency, are defined as follows.

- *Degradation ratio* (DR): the fraction of time a connection receives degraded service. Since we consider a multi-level QoS system, if a connection receives level-i service for a time period T_i, DR $= \frac{\sum_i \frac{(W_{max}-W_i)}{W_{max}} \cdot T_i}{\sum_i T_i}$.
- *Upgrade/degrade frequency* (UDF): the frequency of switching between QoS levels an admitted connection receives.

In the following, we assume that the arrival process of connection requests is Poisson with the new connection-arrival rate λ_0, and the connection-holding time is exponentially distributed with mean $\frac{1}{\mu_0}$. To evaluate the effects of user mobility on system performance, the connection-sojourn time, which is the time a connection stays in a

cell, is also taken into account and is assumed to be exponentially-distributed with mean $\frac{1}{\eta}$ as in [10, 11, 18] for mathematical tractability. However, we will show via simulation later that the formulas for QoS metrics derived under this model are still valid even when different mobility distributions are used.

Under these assumptions, the hand-off rate can be derived as in [6]:

$$\lambda_h = \frac{\eta(1-p_b)}{\mu_0 + \eta p_f}\lambda_0, \tag{2.1}$$

where p_f is the forced-termination probability of hand-off connections and p_b is the blocking probability of new connections. The channel-occupancy time of an admitted connection in a cell is the minimum of the remaining connection-holding time and the connection-sojourn time. Since we assume that both connection-holding time and connection-sojourn time are exponentially-distributed, the distribution of channel occupancy time is

$$f_{c0} = (\mu_0 + \eta)e^{-(\mu_0+\eta)t}. \tag{2.2}$$

Under this degradation scheme, both connection blocking and forced-termination probabilities are improved. However, some connections may receive severely degraded service. In the following section, we investigate the tradeoff among the QoS metrics, especially between the blocking probability and the other three QoS metrics.

2.2 Analysis

Since there are K different QoS levels, we define the system state, $\bar{\mathbf{n}}$, as

$$\bar{\mathbf{n}} = (n_1, n_2, \ldots, n_K),$$

where n_i is the number of service level-i connections in the system. Such a system can be easily modelled as a Markov chain once the transition probabilities are obtained. In our model, the transition probabilities depend on the admission control (i.e, N_{thresh} value), and the degradation policy. Let W_a be the number of idle channels, and $N_{_T}$ be the total number of existing connections in the system upon the arrival of a connection request. The admission control and bandwidth degradation algorithm is presented in Figure 2.2.

```
01.  if (the connection is a hand-off connection or a new connectionr
       but N_T < N_thresh) {
02.    if (W_a ≥ W_min)
03.      W_allocated = min(W_max, W_a)
04.    elseif (W_a ≤ W_min & (C − N_T * W_min) ≥ W_min)
05.      { W_allocated = 0.
06.        for (i = K, i > 0, i − −)
07.          while (W_allocated < W_min & N_i > 0) {
08.            Randomly degrade one of the n_i connections by
               min(W_min, W_i − W_min) units of channels.
09.            n_i = n_i − 1;
10.            n_j = n_j + 1, where j is such that
               W_j = min(W_min, W_i − W_min)
11.            W_allocated = W_allocated + W_i − W_j; }}
12.    else
13.      Reject the connection request. }
14.  else
15.    Reject the connection request.
```

Figure 2.2. A pseudo-code of the bandwidth degradation algorithm

Allocating only W_{min} units of channels to an incoming connection, when there is a shortage of bandwidth, minimizes the need to degrade the QoS levels of the existing connections, and hence, a smaller DR and UDF can be achieved. On the other hand, fairness is an important issue when considering the service degradation (i.e., bandwidth reallocation) in a multi-service class system. One may expect a tradeoff between the fairness and UDF, because the probability that a connection is degraded increases (consequently, the value of UDF increases) when using a fair degradation algorithm while using an unfair algorithm as shown in lines 06–11 of Figure 2.2 ensures a lower value of UDF. This tradeoff will be investigated more thoroughly later. The corresponding upgrade algorithm is shown in Figure 2.3, when a level-i connection leaves the system such that $W_r = W_i$ units of channels are returned to the system. Here, a fair upgrade algorithm is used to ensure the fairness among the existing connections.

```
01.  n_i = n_i − 1
02.  for (i = 1, i < K, i++);
03.      while (W_r > 0 & N_i > 0) {
04.          Randomly upgrade one of the N_i connections
             by one unit of channel.
05.          n_i = n_i − 1.
06.          n_{i+1} = n_{i+1} + 1.
07.          W_r = W_r − 1.}
```

Figure 2.3. A pseudo-code of the bandwidth upgrade algorithm

2.2.1 Stationary Distribution of the Number of Connections in a Cell

In order to obtain the stationary distribution of the system state upon each arrival of a connection request or departure of an exiting connection, first we need to know the transition probability. Given a state $\bar{\mathbf{n}} = (n_1, n_2, \ldots, n_K)$ and $\sum_i n_i < N_{thresh}$, if a connection request arrives before the departure of any existing connection in the system,

$$P_{\bar{\mathbf{n}},\bar{\mathbf{n}'}} = \frac{\lambda_0 + \lambda_h}{\sum n_i\mu + \lambda_0 + \lambda_h}, \tag{2.3}$$

where $\bar{\mathbf{n}'}$ is decided by lines 06–11 of Figure 2.2. If a level-i connection leaves the system,

$$P_{\bar{\mathbf{n}},\bar{\mathbf{n}'}} = \frac{n_i\mu}{\sum n_i\mu + \lambda_0 + \lambda_h}, \tag{2.4}$$

where $\bar{\mathbf{n}'}$ is decided by the algorithm in Figure 2.3. If $\sum_i n_i \geq N_{thresh}$, the transition probabilities can still be obtained as Eqs. (2.3) and (2.4) by replacing $\lambda_0 + \lambda_h$ with λ_h. The stationary state distribution can be obtained by solving the equation

$$\pi P = \pi. \tag{2.5}$$

Figure 2.4 shows the resulting Markov chain for a special case, where $K = 2$, $W_1 = 1$ and $W_2 = 2$. If new-initiated connections are not differentiated from hand-off connections (i.e., $N_{thresh} = C$), the stationary distribution of the number of connections in a cell can be obtained by Erlang's formula by setting the arrival rate λ_i to $\lambda_0 + \lambda_h$ (the arrival rate of new connection requests plus the arrival rate of hand-off connections) and service rate μ_i to $i \cdot (\mu_0 + \eta)$. If $N_{thresh} < C$, the stationary distribution can still be obtained as a general Erlang's formula with variable arrival rates.

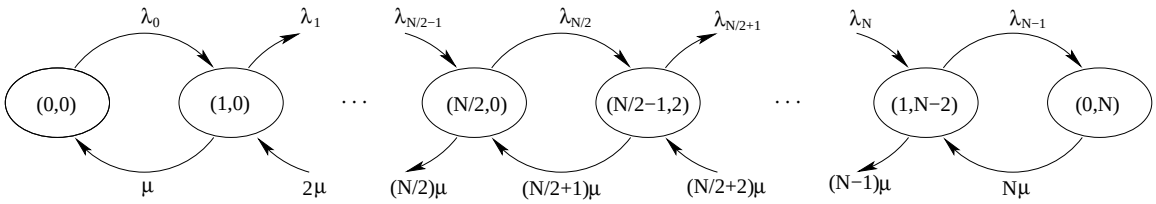


Figure 2.4. State transitions of the number of connections in one cell

The stationary distribution is given as

$$\pi_{n_1,n_2} = \frac{1}{\sum_{i=0}^{N} \frac{\prod_{k=0}^{i-1} \lambda_k}{\mu^i i!}} \times \frac{\prod_{k=0}^{n_1+n_2-1} \lambda_k}{\mu^{n_1+n_2}(n_1+n_2)!}, \tag{2.6}$$

where $\lambda_k = \lambda_0 + \lambda_h$ if $k < N_{thresh}$ and $\lambda_k = \lambda_h$ for $k \geq N_{thresh}$. In either case, the blocking probability p_b is $\sum_{i+j=m'+n'}^{N} \pi_{i,j}$, and the forced-termination probability p_f is $\pi_{0,N}$, which can be obtained from Eq. (2.6).

Thanks to the assumptions of homogeneous cells, Poisson arrival process and exponential channel occupancy time, the statistics for all cells are identical and independent, so the analysis of only one cell is statistically sufficient. Moreover, this stationary distribution is also the probability distribution of the number of connections observed at the arrival time of each connection request.

2.2.2 QoS Metrics

As we mentioned in the previous section, the QoS level received by an admitted connection varies during its lifetime. From the perspective of an admitted connection, given that the system state is $\bar{\mathbf{n}} = (n_1, n_2, \ldots, n_k)$, it may receive one of the K service levels. In order to analytically derive the DR and UDF of an admitted connection, we need to establish a new state, $\bar{\mathbf{c}} = \bar{\mathbf{n}}^{(\mathbf{i})}$, which correctly reflects the evolution of the QoS levels of an admitted connection. The new state $\bar{\mathbf{c}}$ represents that the system is in state $\bar{\mathbf{n}}$, and the admitted connection receives the level-i service (obvious, $n_i > 0$). For example, consider a system with $K = 4$, and $W_i = i$ for $i = 1$ to K. Assume the system capacity, C, is 20 (units of channels) and $N_{thresh} = 15$. If a newly-initiated connection, r_1, arrives when the system is in state $(2, 0, 2, 3)$, $\bar{\mathbf{c}}_{\mathbf{r_1}} = (3, 0, 3, 2)^{(1)}$, simply because one of the level-4 connections is degraded to level-3 and r_1 receives the minimum service (i.e., level-1 service), according to the algorithms introduced

before. If another hand-off connection joins the system some time later, the service state (from connection r_1's point of view) will be $\bar{\mathbf{c}}_{\mathbf{r_1}} = (4, 0, 4, 1)^{(1)}$ since one of the level-4 connections are degraded, but r_1 still receives the level-1 service. If a level-3 connection leaves the system after that, $\bar{\mathbf{c}}_{\mathbf{r_1}} = (1, 3, 3, 1)^{(2)}$, if r_1 is chosen to be upgraded (with probability $\frac{3}{4}$). Therefore, we can model the transition of r_1's QoS levels as an embedded Markov chain Y_{t_n}. In the above example, $Y_{t_0} = (3, 0, 3, 2)^{(1)}$, $Y_{t_1} = (4, 0, 4, 1)^{(1)}$ and $Y_{t_2} = (1, 3, 3, 1)^{(2)}$, where t_i is the occurrence time of the i-th event (either an arrival of a connection request arrival or a departure of an existing connection). If r_1 leaves the system at t_n, then $Y_{t_n} = A$; that is, A is a completion (absorption) state (i.e., $Y_t = A$ for $t > t_n$). For convenience, we just use $\bar{\mathbf{c}}$ as the QoS state of the admitted connection, r_1. The state transition probability $P_{\bar{\mathbf{c_1}},\bar{\mathbf{c_2}}}$ for r_1 can be obtained, based on the algorithms introduced in the previous section, and the detailed derivation will be presented later for the case of $K = 2$.

Degradation ratio

We now derive the DR of an admitted connection, based on the embedded Markov chain described above. First, we need to derive $N_{\bar{c}_j}$, the number of visits to state $\bar{c}_j$ before entering the completion state A, given that the initial state is $\bar{c}_i$:

$$E_{\bar{c}_i}(N_{\bar{c}_j}) = E_{\bar{c}_i}[\sum_{n=0}^{\infty} 1_{\{Y_n=\bar{c}_j\}}] = \sum_{n=0}^{\infty} P_{\bar{c}_i\bar{c}_j}(n), \tag{2.7}$$

where Y_n is the state after the n-th transition and $P_{\bar{c}_i\bar{c}_j}(n)$ is the n-step transition probability from state $\bar{c}_i$ to state $\bar{c}_j$. The $\sum_{n=0}^{\infty} P_{\bar{c}_i\bar{c}_j}(n)$ is also the (i, j)-th element of potential matrix G, which can be obtained by the following equation:

$$G = \sum_{n=0}^{\infty} P^n. \tag{2.8}$$

P is the transition matrix of the embedded Markov chain, and can be written as

$$P = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{T_A} & \mathbf{T_T} \end{bmatrix},$$

where T_T is the restriction of P to the transient set (note that except the absorption state A, all other states are transient). Since we only consider the number of visits

to the transient states before entering the completion state A, the potential matrix can be rewritten as

$$G = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{F} & \mathbf{S} \end{bmatrix},$$

where $S = \sum_{n=0}^{\infty} T_T^n$ and $E_{\bar{c}_i}(N_{\bar{c}_j})$ is just the (i,j)-th element of matrix S. By matrix manipulation, S can be computed by the following equation [19],

$$\mathbf{S} = (\mathbf{I} - \mathbf{T_T})^{-1}. \tag{2.9}$$

Next we define a conditional DR, given the initial state is $\bar{\mathbf{c}}$,

$$\mathrm{DR}_{\bar{c}} = \mu \sum_{k=1}^{K} \frac{W_{max} - W_k}{W_{max}} \sum_{\{\bar{n}:n_k>0\}} \frac{E_{\bar{c}}(\bar{n}^{(k)})}{\lambda + \sum n_j \mu}, \tag{2.10}$$

where $\lambda = \lambda_0 + \lambda_h$ if $\sum n_i < N_{thresh}$; otherwise, $\lambda = \lambda_0$. Finally, DR can be obtained by Eq. (2.10) as

$$\mathrm{DR} = \sum_{\bar{n}} \pi_{\bar{n}} \cdot P(\bar{c}|\bar{n}) \cdot \mathrm{DR}_{\bar{c}},$$

where $\pi_{\bar{n}}$ is the stationary distribution of the system state, and can be obtained by Eq. (2.5). The conditional probability, $P(\bar{c}|\bar{n})$, is decided by the admission control and degradation policy. Taking the previous example, we get $P(\bar{c} = (3,0,3,2)^{(1)}|\bar{n} = (2,0,2,3)) = 1$.

Upgrade/degrade frequency

Let's consider how to derive UDF — the average number of switches per unit time between different service levels. Since there are K service levels, we should group the states with the same service level into a set. Let T_i be such a set $\{\bar{\mathbf{c}} : \bar{\mathbf{n}}^{(i)} \forall \bar{\mathbf{n}} \in \mathcal{N}\ and\ n_i > 0\}$ for $i = 1$ to $i = K$. Consider the sequence of times, $t(0) = 0$, $t(1)$, $\cdots$, where $t(n)$ is the n-th service switching . Let $\tilde{Y_n} = Y_{t(n)}$, then $\{\tilde{Y_n}\}$ is also a discrete Markov chain as shown in Figure 2.5 with the transient matrix $\tilde{\mathbf{P}}$ obtained as follows:

- If $\bar{c}_i \in \mathbf{T_h}$, then $\tilde{p}_{\bar{c}_i\bar{c}_j} = 0$ for $\bar{c}_j \in \mathbf{T_h}$.

- For $\bar{c}_i \in \mathbf{T_h}$ and $\bar{c}_j \in \{A\} \cup_{k=1,k\neq h}^{K} \mathbf{T_k}$, $\tilde{p}_{\bar{c}_i\bar{c}_j}$ is the probability of being absorbed in the states, $\cup_{k=1,k\neq h}^{K}\mathbf{T_k}$, of the Markov chain with transition matrix $\hat{\mathbf{P}}$:

$$\hat{P} = \begin{array}{c} \\ \cup_{k=1,k\neq h}^{K}\mathbf{T_k} \\ T_h \end{array} \begin{array}{c} \begin{array}{cc} \cup_{k=1,k\neq h}^{K}\mathbf{T_k} & T_h \end{array} \\ \begin{pmatrix} \mathbf{1} & \mathbf{0} \\ B_h & Q_h \end{pmatrix} \end{array},$$

  where B_h is the transition matrix of the set T_h to all other states, $P_{T_h,\ \cup_{k=1,k\neq h}^{K}}$, and Q_h is the restriction of P to the set T_h. Then $\tilde{p}_{\bar{c}_i\bar{c}_j} = (S_h B_h)_{\bar{c}_i\bar{c}_j}$.

Having $\tilde{\mathbf{P}}$ this way, the time to absorption into $\{A\}$ is then the number of switches between T_i's. If we rewrite $\tilde{\mathbf{P}}$ as

$$\tilde{P} = \begin{array}{c} \\ A \\ \cup_{k=1}^{K}\mathbf{T_k} \end{array} \begin{array}{c} \begin{array}{cc} A & \cup_{k=1}^{K}\mathbf{T_k} \end{array} \\ \begin{pmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{T_A} & \tilde{\mathbf{Q}} \end{pmatrix} \end{array},$$

then the average number of service-level switches before a connection is completed or handed off, given the initial state $\bar{c}$, is

$$E[N_d]_{\bar{c}} = (1 - \tilde{\mathbf{Q}})^{-1}\mathbf{1} - 1.$$

Finally ,

$$\text{UDF} = \mu \sum_{\bar{n}} \pi_{\bar{n}} \cdot P(\bar{c}|\bar{n}) \cdot E[N_d]_{\bar{c}}.$$

2.2.3 A Special Case: $K = 2$

Let's consider a simple case with $K = 2$, $W_1 = 1$ and $W_2 = 2$ (e.g., a video telephony with low-motion (=20 kbps) and standard quality (=40 kbps)). The resulting embedded Markov chain for the QoS level of an admitted connection is shown in Figure 2.6, and the transition probabilities can be derived as follows. Since there are only two service levels, we will denote the state $\bar{c} = (n1, n2)^{(2)}$ as f_{n1+n2} ('f' as full service), and $\bar{c} = (n1, n2)^{(1)}$ as d_{n1+n2} ('d' as degraded service). Consider an admitted connection, r_1, in any state. Three different events may occur: arrival of a new connection, departure of r_1, or departure of any other existing connections. We need to differentiate several situations in order to calculate the transition probabilities as follows.

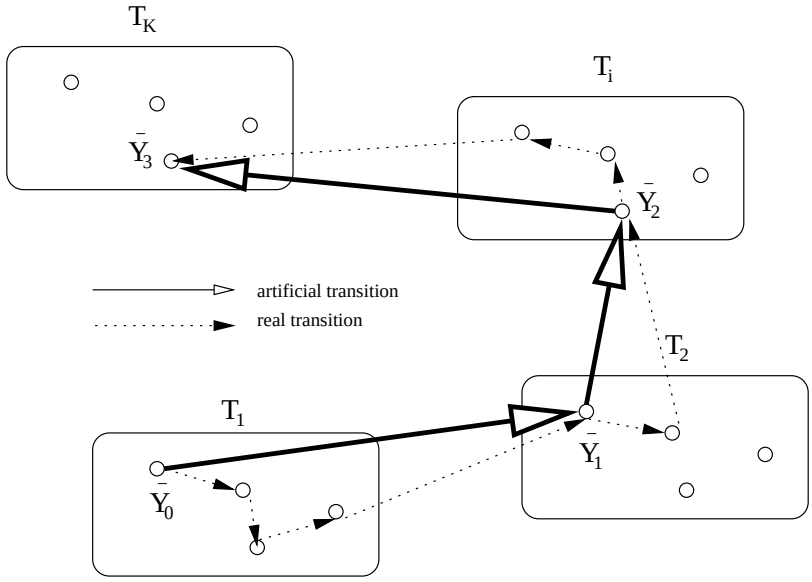


Figure 2.5. Transitions between different QoS levels

- For state f_i, $1 \leq i \leq \frac{N}{2} - 1$, all existing connections receive full service. Three transition probabilities in these states are $P_{f_i,f_{i+1}} = \frac{\lambda_i}{\lambda_i+i\mu}$, $P_{f_i,A} = \frac{\mu}{\lambda_i+i\mu}$ and $P_{f_i,f_{i-1}} = \frac{(i-1)\mu}{\lambda_i+i\mu}$.
- For state f_i, $\frac{N}{2} \leq i \leq N-1$, the arrival of a new connection request may result in two different transitions. One is that connection C is degraded such that the state transits to degraded state $d_{i-\frac{N}{2}+1}$. The other is that C is not degraded so that the state transits to f_{i+1}. The associated transition probabilities are $P_{f_i,d_{i-\frac{N}{2}+1}} = \frac{\lambda_i}{(N-i)(\lambda_i+i\mu)}$ and $P_{f_i,f_{i+1}} = \frac{(N-i-1)\lambda_i}{(N-i)(\lambda_i+i\mu)}$, respectively. The other transition probabilities are $P_{f_i,A} = \frac{\mu}{(\lambda_i+i\mu)}$ and $P_{f_i,f_{i-1}} = \frac{(i-1)\mu}{(\lambda_i+i\mu)}$.
- For state d_i, $1 \leq i \leq N' = \frac{N}{2}$, the departure of any other connections may result in two different transitions. One is that C is upgraded because of the others' departure such that the state transits to $f_{i+N'-1}$. The other is that C continues receiving degraded service and the state transits to d_{i-1}. The associated transition probabilities are $P_{d_i,f_{i+N'-1}} = \frac{N'}{i}\frac{\mu}{\lambda_{i+N'}+(N'+i)\mu}$ and $P_{d_i,d_{i-1}} = (1-\frac{1}{i})(N'+i)\frac{\mu}{\lambda_{i+N'}+(N'+i)\mu}$. The other transition probabilities are $P_{d_i,d_{i+1}} = \frac{\lambda_{i+N'}}{\lambda_{i+N'}+(N'+i)\mu}$ and $P_{d_i,A} = \frac{\mu}{[\lambda_{i+N'}+(N'+i)\mu]}$.
- Note that $\lambda_N = 0$.

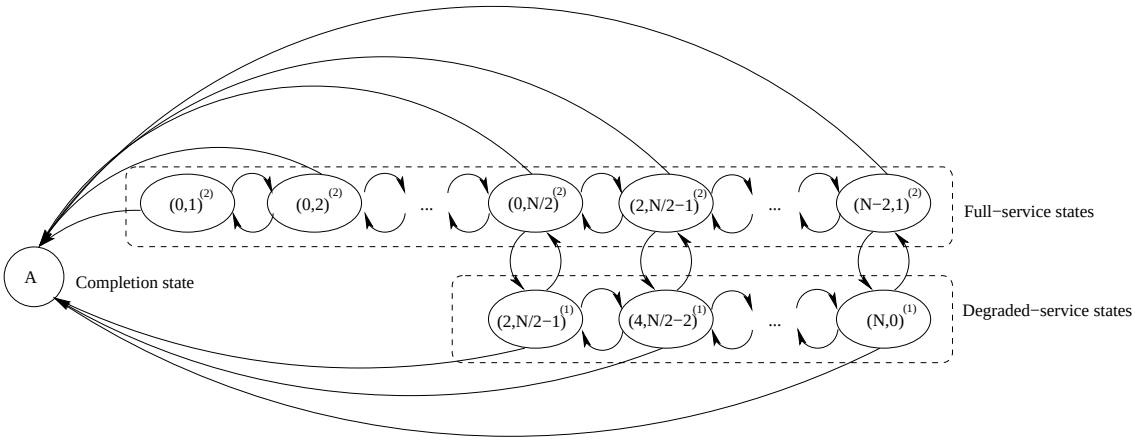


Figure 2.6. State transitions of a connection admitted into any cell

The DR_i can be obtained as Eq. (2.10), but we slightly change it in this special case as

$$\mathrm{DR}_{\bar{c}} = \sum_{d_j \in \{degraded\ class\}} \mu E_{\bar{i}}(N_{d_j}) T_{sojourn,d_j}, \tag{2.11}$$

such that DR will be the fraction of time in degrade service class. The mean sojourn time in state d_j, $T_{sojourn,d_j}$, is $\frac{1}{\lambda_{j+N'}+(j+N')*\mu}$. Then, the degradation ratio can be computed as

$$\mathrm{DR} = \sum_{i=0}^{N'-1} \pi_{0,i} \mathrm{DR}_{f_i} + \sum_{i=N'}^{N-1} \pi_{2i-N,N-i} \mathrm{DR}_{d_i}, \tag{2.12}$$

where π_{n_1,n_2} is given in Eq. (2.6).

Since there are only two kinds of service switching (i.e., service degradation: $f_i \rightarrow d_i$ or service upgrade: $d_i \rightarrow f_i$), we use the first-step analysis for deriving UDF, and the following system of linear equations can be obtained:

$$\begin{aligned} E(D_{f_i}) &= \sum_{j,j\neq i} P_{f_i,f_j} E(D_{f_j}) + \sum_j P_{f_i,d_j} (E(D_{d_j}) + 1) \\ E(D_{d_i}) &= \sum_j P_{d_i,f_j} (E(D_{f_j}) + 1) + \sum_{j,j\neq i} P_{d_i,d_j} E(D_{d_j}) \end{aligned} \tag{2.13}$$

The solution to this system of linear equations can be computed as

$$\mathbf{E}(\mathbf{D}) = (\mathbf{I} - \mathbf{T_T})^{-1} \mathbf{C}, \tag{2.14}$$

where $\mathbf{C}$ is the column vector with the i-th element equal to $P_{f_i,d_{i-N'+1}}$ for $1 \leq i \leq N-1$ or $P_{d_{i-N},f_{i-N'-1}}$ for $N+1 \leq i \leq \frac{3}{2}N$. By using Eq. (2.9), the vector $\mathbf{E}(\mathbf{D})$ can

be rewritten as

$$\mathbf{E}(\mathbf{D}) = \mathbf{SC}. \tag{2.15}$$

UDF can then be obtained as:

$$\text{UDF} = \sum_{i=0}^{N'-1} \mu\pi_{0,i} E(D_{f_{i+1}}) + \sum_{i=N'}^{N-1} \mu\pi_{2i-N,N-i} E(D_{d_{i-N'+1}}). \tag{2.16}$$

Note that the DR and UDF derived so far are the QoS metrics a hand-off connection may experience in each cell. The values of these QoS metrics for a connection in the cell where the connection was initiated, are different, but similar formulas can still be derived by considering the restriction threshold

$$\begin{aligned} \text{DR}_I &= \sum_{i=0}^{min(N_{thresh},N'-1)} \mu\pi_{0,i} T_{d,i+1} \\ &\quad + \sum_{i=min(N_{thresh},N')}^{j-1} \mu\pi_{2i-N,N-i} T_{d,i-N'+1} \end{aligned}$$

$$\begin{aligned} \text{UDF}_I &= \sum_{i=0}^{min(N_{thresh},N'-1)} \mu\pi_{i,0} E(D_{f_{i+1}}) \\ &\quad + \sum_{i=min(N_{thresh},N')}^{j-1} \mu\pi_{N-i,2i-N} E(D_{d_{i-N'+1}}), \end{aligned}$$

where DR_I and UDF_I are the QoS metrics for a connection in the cell where the connection was initiated.

2.3 Numerical Results

We consider a cellular network, in which each cell has 40 units of channels. The arrival process of new connections is assumed to be Poisson, and the connection-holding and connection-sojourn times are exponentially-distributed. The formulas for the resulting hand-off rate and channel-occupancy time can be found in Eqs. (2.1) and (2.2). For illustrative purposes, we fist consider the case with $K = 2$, and assume that each full service requires 2 units of channels and each degraded service requires only 1 unit of channel. The impact of connection-arrival rates, connection-holding time and user mobility on the QoS metrics are discussed. Then, we consider a case

of $K = 3$, which shows how the bandwidth allocation algorithm will affect the QoS metrics.

2.3.1 K=2: Full and Degraded Service

Four QoS metrics — blocking probability of new connections (P_b), forced-termination probability of hand-off connections (P_f), degradation ratio (DR) and upgrade/degrade frequency (UDF) — are evaluated. Since the arrival rate of connection requests, connection-holding time, and mobility ($= \frac{1}{\eta}$) of each connection could significantly affect these metrics, three sets of numerical results are shown for these factors under various settings of the restriction threshold. The restriction threshold ranges from 1 to 40 in each numerical analysis. If the restriction threshold is 1, the traffic restriction is applied at state $(1, 0)$ and higher states as shown in Figure 2.4, and at most one newly-initiated connection could be admitted into the system (e.g., most connections in cells are hand-off connections from the adjacent cells). On the other hand, if the restriction threshold is 40, no channel is reserved for hand-off connections, and there is no distinction between new and hand-off connections. Selection of the restriction threshold under different traffic loads is also discussed at the end of this section.

QoS metrics vs. arrival rate of connection requests

Figure 2.7 plots P_b and P_f under four arrival rates: $\lambda = 20$, 30, 40, 50 connections per unit time. The tradeoff between P_b and P_f is obvious under different restriction thresholds. In the case of light traffic ($\lambda = 20$) with a high restriction threshold, P_b and P_f are negligible. Even in the case of heavy loads ($\lambda = 50$), both P_b and P_f are still only 0.13 and 0.18, respectively (as compared to 0.45 without any degradation and traffic restriction).

Figure 2.8, however, shows that the decrease of P_f and P_b by the degradation scheme results in severe service degradation of individual connections. DR increases with the restriction threshold under different loads and is higher than 0.8 in the case of high loads and high restriction thresholds. UDF increases more quickly than DR as the restriction threshold increases. Even when the system reserves 40% of channels for hand-off connections, UDF is still as high as 5 in the case of moderate traffic load. A

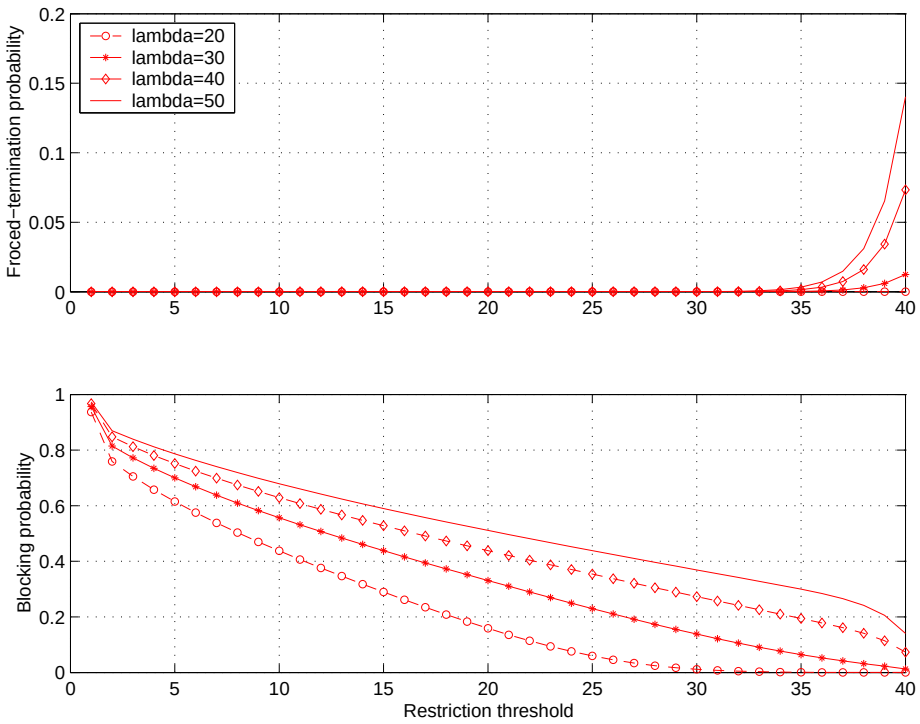


Figure 2.7. P_b and P_f vs. arrival rate of connection requests

drop in UDF can also be observed in case of high loads and high restrictions, because there is a sharp increase of P_f, and consequently the hand-off rate may significantly decrease.

QoS metrics vs. connection-holding time

Figure 2.9 shows P_b and P_f under four different connection-holding times: $\frac{1}{\mu} = 8$, 4, 2, and 1 unit of time. In this case, the arrival rate of connection requests is 20 connections/unit of time. P_b is much more sensitive to connection-holding time than P_f. When the restriction threshold is high (e.g., 35), the blocking probability is still large (e.g., 0.5 in case of $\mu = 0.25$). But we still could simultaneously achieve low probabilities with the help of service degradation, even in the case of a larger connection-holding time.

DR and UDF under the four connection-holding times are plotted in Figure 2.10. In the case of a larger connection-holding time, both QoS metrics show a drop when the threshold is high, because of the sharp increase in the forced-termination probability as shown in Figure 2.9. However, unlike DR, UDF tends to decrease with

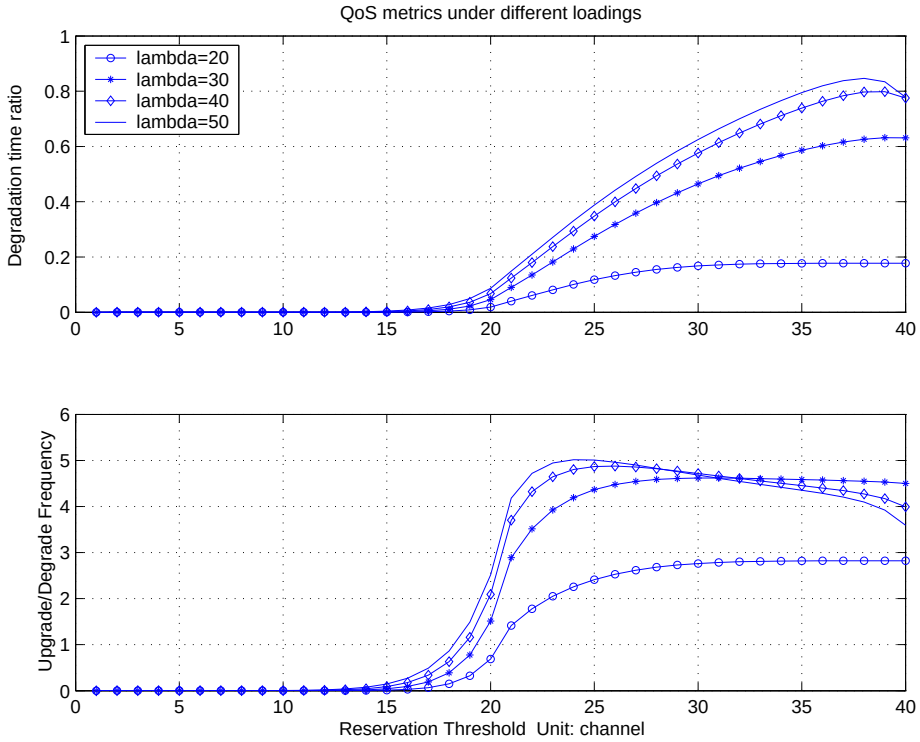


Figure 2.8. DR and UDF vs. arrival rate of connection requests

the increase of connection-holding time. In the case of a higher restriction threshold (e.g., 35), the UDF value when $\mu = \frac{1}{8}$ is half of that when $\mu = \frac{1}{2}$. However, the UDF is not only dependent on μ but also on the threshold as shown in Figure 2.10. When the threshold is high and the connection-holding time is longer, the service switching due to the departures of other connections is lessened and thus, the UDF decreases with the increase of connection-holding time. However, when the threshold is low (more new connections are blocked) and the connection-holding time is shorter, the total traffic load is smaller (note that λ is fixed in this subsection), and thus, most connections would not interfere with one another, which results in a smaller UDF. This explains the crossover of UDF under different μ's when the threshold increases. These different dependencies on connection-holding time also justify the need for considering both metrics.

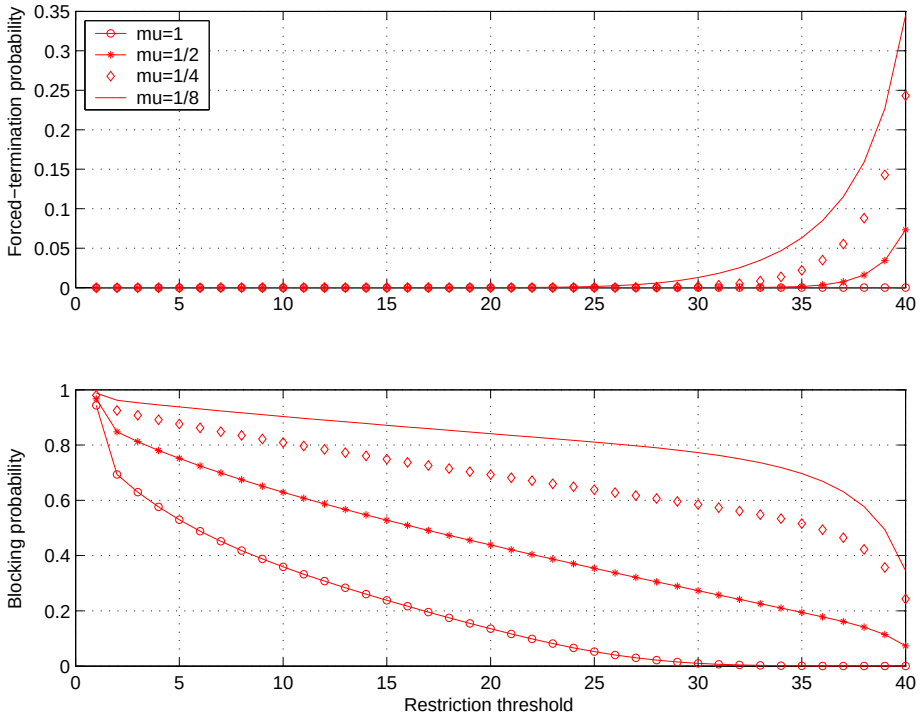


Figure 2.9. P_b and P_f vs. connection-holding time

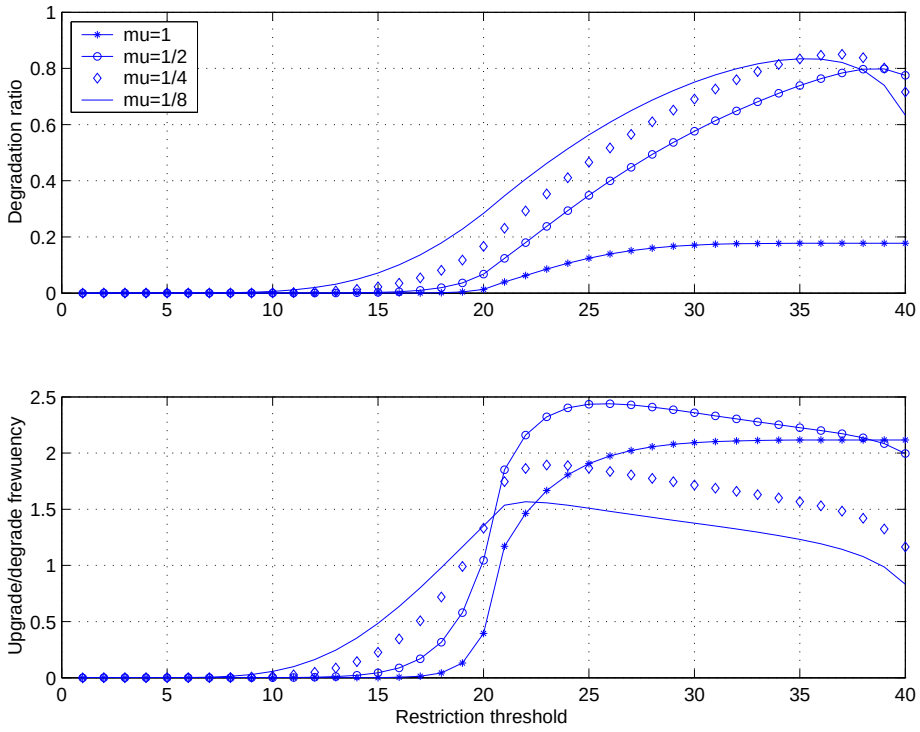


Figure 2.10. DR and UDF vs. connection-holding time

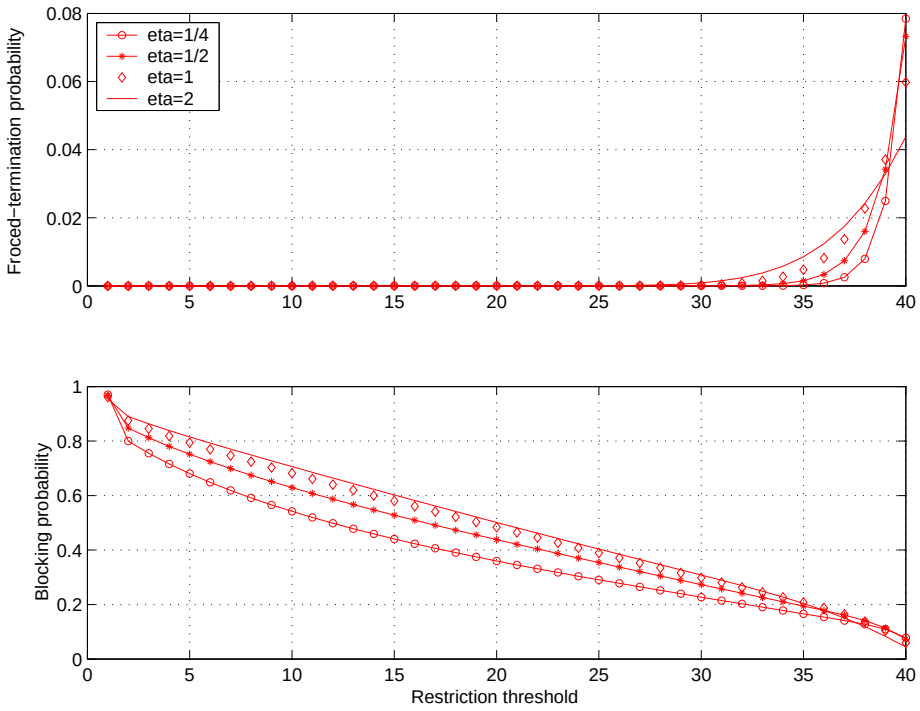


Figure 2.11. Pb and Pf vs. mobility

QoS metrics vs. mobility

Figure 2.11 shows P_b and P_f under four different connection-sojourn times: $\frac{1}{\eta} = 0.5$, 1, 2, and 4 units of time. In all cases, P_b and P_f only slightly increase with mobility. Even in case of higher mobility, both P_b and P_f can be as low as 0.1 or less with the help of a high restriction threshold and service degradation.

DR and UDF are plotted in Figure 2.12, and these two metrics exhibit inverse dependence on mobility. DR remains almost the same under the different cases of mobility. However, UDF can be three times larger in the case of higher mobility than in the case of lower mobility (e.g., UDF$\approx$ 6 when $\eta = 2$, but UDF$\approx$ 2 when $\eta = \frac{1}{4}$, in the case of threshold=27). The reason for this is that high mobility results in frequent switches between different QoS levels, but the amount of time a connection resides in each level is statistically the same. Therefore, we should consider both DR and UDF for QoS provision. In the case of higher mobility, UDF is the dominant factor of QoS for individual connections.

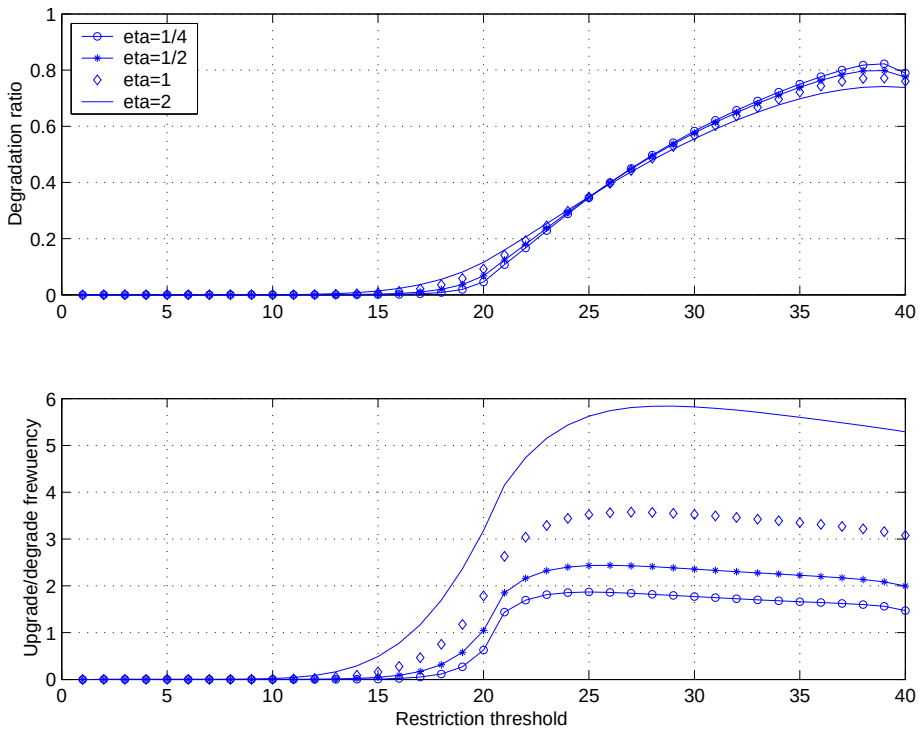


Figure 2.12. DR and UDF vs. mobility

System operation region

There is an obvious tradeoff between the blocking probability of new connections and the other QoS metrics under the proposed degradation and restriction scheme. Therefore, there does not exist an absolutely optimal operation point in terms of all of the four parameters. Since the forced-termination probability rises sharply only when the restriction threshold is close to the system capacity, the possible choice of restriction threshold should be between $\frac{N}{2}$ and N. If we only consider the blocking probability and forced-termination probability, the optimal operation region should be very close to system capacity (e.g., the threshold is 37 or 38 as shown in Figure 2.9). However, DR has a maximal value (≈ 0.8 in Figure 2.10), meaning that connections are severely degraded. If we choose the threshold $\approx$ 25, DR can be significantly improved (from 0.8 to 0.4) with only a slight increase of P_f by 0.12 (P_b is negligible and UDF is almost the same). This means that admitted connections could receive much better service at the expense of blocking only 12% more connections. The same conclusion can be drawn from the results in Figures 2.11 and 2.12. Both DR and UDF

decrease significantly (DR decreases from 0.6 to 0.1 in all cases, and UDF decreases from 6 to 3 in case of high-mobility and from 2 to 0.8 in case of low-mobility) with an increase of P_b less than 0.2 in most cases, if we set the threshold close to one half of the system capacity, instead of setting to the higher values. We show that if only P_b and P_f are considered, even though we can simultaneous achieve low P_b and P_f, each connection endures severely degraded service and frequent switching of service levels. By considering both DR and UDF, each connection can receive much better QoS (much smaller DR and much less service switchings) without sacrificing P_f much.

As the numerical results shown in the previous subsection, the choice of operation point may also vary under different traffic loads and mobility. For example, if customers have longer connection-holding times, the operation point may be chosen to be close to the system capacity. On the other hand, if the mobility of customers is high, the operation point may be chosen to be close to one half of the system capacity such that UDF is acceptable, as suggested in the set of the third numerical results.

2.3.2 K=3: Fairness vs. UDF

As we mentioned in Section III, the upgrade/degrade (i.e. resource reallocation) algorithm may affect not only the DR/UDF but also the fairness among the existing connections. By "fairness" we mean that service provider should allocate the bandwidth to all existing connections in an egalitarian way. Therefore, if a connection is admitted into the system, it should receive a service level as close to that of the existing connections as possible. On the other hand, if service degrade/upgrade of the existing connections is necessary, connections in the highest/lowest service level are *randomly* and *uniformly* chosen to be dergarded/upgraded by a minimum amount (in our case, one unit of channel). Figure 2.13-(a) shows the transitions of system states under this fair reallocation algorithm when $C = 24$ and $K = 3$ with $W_1 = 2, W_2 = 3$ and $W_3 = 4$. For example, when a connection arrives at state $(0, 0, 6)$, in order to allocate as many channels as possible (in this case, 3 units of channels) to the new connection, three level-3 connections are degraded by one unit of channel. The resulting state is $(0, 4, 3)$. Obviously, the fairness is achieved at the expense of more

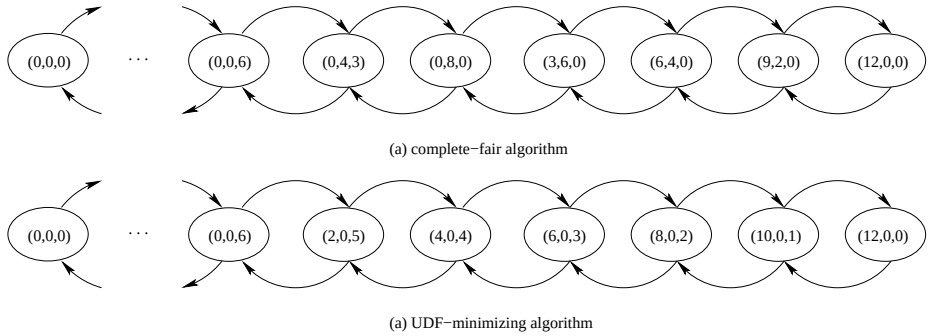


Figure 2.13. State transition diagram

service-level switches of the existing connections. At the other end of the spectrum, we may allocate the minimum number of channels to an incoming connection by degrading as few existing connections as possible. For the departure of a connection, we may reallocate the freed channels with a minimum adjustment of the current channel constellation. The state transitions of this "unfair" algorithm are shown in Figure 2.13-(b). If a connection arrives when the system is in state $(0, 0, 6)$, only 2 channels taken from one existing level-3 connection are reallocated to the new connection, and the resulting state is then $(2, 0, 5)$. Since this unfair (UDF-minimizing) algorithm only requires a minimum adjustment of the current bandwidth allocation, a minimum UDF can be achieved.

Figure 2.15-(a) plots the DR under the completely-fair and UDF-minimizing algorithms. The values of DR under these two algorithms are the same for all the thresholds, because when the system is fully-utilized, the total amount of degradation—if the total number of connections in the system are the same under these two algorithms—is independent of the algorithm used. For example, the total amount of degradation in state $(0, 4, 3)$ of Figure 2.13-(a) is 1*4=(7*4-24)=4 while in state $(2, 0, 5)$ of Figure 2.13-(b), the total amount of degradation is also 2*2=(7*4-24)=4. Therefore, the average degradation of one connection will be the same (i.e., $\frac{4}{7}$) regardless of the algorithm used. However, the impact of the reallocation algorithm on the UDF is significant. As shown in Figure 2.15-(b), the values of UDF under the completely-fair algorithm are almost twice those under the UDF-minimizing algorithm. Even though the UDF can be minimized due to the minimal adjustment

for $(i = K,\ i > 0,\ i - -)$
  while $(W_{allocated} < W_{min}\ \&\ N_i > 0)$ {
    Randomly degrade one of the n_i connections by 1 unit of channel.
    $n_i = n_i - 1$;
    $n_{i-1} = n_{i-1} + 1$.
    $W_{allocated} = W_{allocated} + 1$; }}

(a) fair degradation

for $(i = 1,\ i < K,\ i + +)$;
  while $(W_r > 0\ \&\ N_i > 0)$ {
    Randomly upgrade one of the N_i connections by $\min(W_r, W_{max} - W_i)$ units of channels.
    $n_i = n_i - 1$.
    $n_j = n_j + 1$, where j is such that $W_j = \min(W_r, W_{max} - W_i)$.
    $W_r = \max(0, W_r - W_{max} + W_i.$ }

(b) unfair upgrade

Figure 2.14. Bandwidth reallocation algorithm: Com-2

of resource allocation, it is extremely unfair in the sense that some connections are severely degraded while the others receive full service (e.g., in state (2,0,5), (4,0,4), and etc., in Figure 2.13-(b)). Between these two extremes are the algorithms with the combination of fair/unfair upgrade/degrade algorithms. The "COM-1" is our proposed bandwidth allocation policy which applies the unfair degradation but fair upgrade while "COM-2" enforces the fair degradation and unfair upgrade as shown in Figure 2.14. With the help of this combination, a fairer algorithm with a smaller UDF can be achieved as shown in Figure 2.15-(b). Since the optimal operation region is closer to a half of the system capacity as mentioned in the previous subsection, "COM-1" is preferred as our bandwidth allocation algorithm.

2.4 Simulation

In the above analysis, we assumed that the mobility is exponentially-distributed. In order to verify the applicability of our model to more general cases, we set up simulation as follows. A cellular network of 30 cells is used in our simulation. As

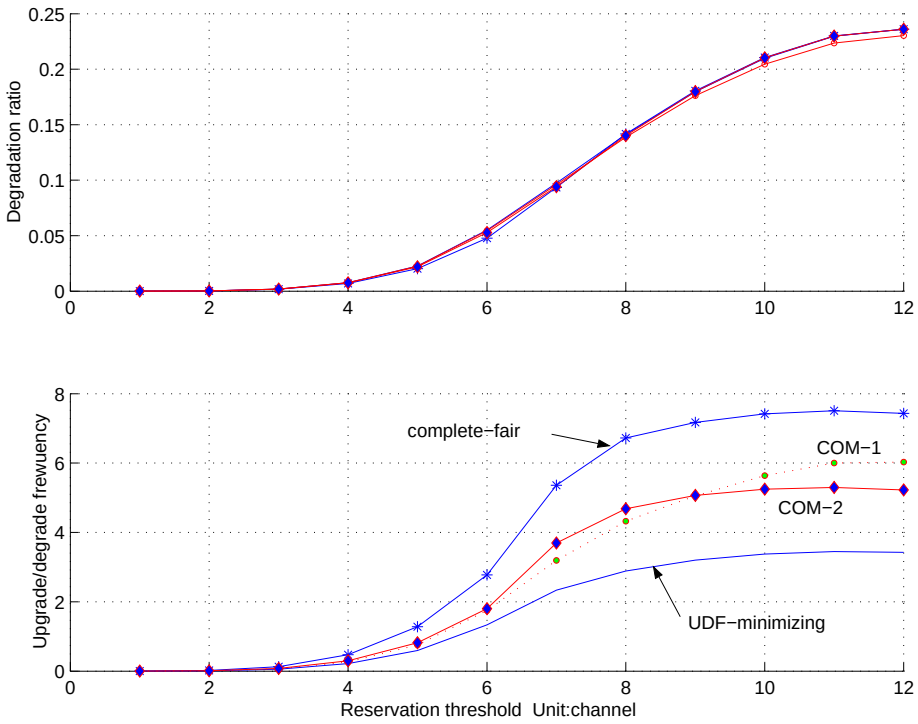


Figure 2.15. Fairness v.s. UDF

shown in Figure 2.16, the statistics of boundary cells (e.g., cells 7, 8, 9, 20) are not taken into account in the comparison with the numerical analysis of the previous section. The arrival process of connection requests is still Poisson, connection-holding time is exponentially-distributed but the assumption of exponentially-distributed connection-sojourn times is relaxed since the stochastic model for mobility may still be arguable. For comparative purposes, we assume that each cell has 40 units of channels. Both heavy-load (40 connections per unit of time) and light-load (20 connections per unit of time) cases are considered. Three distributions of the connection-sojourn time — exponential, uniform, and normal distributions — are considered with mean of 1 unit of time and variance of 1 (except for the case of uniform distribution).

The simulation results are plotted in Figure 2.17. Both DR and UDF are plotted with the numerical results in the previous section (solid lines). In both cases, most of the simulation results are close to the numerical results (the largest error of DR is about 15% when the arrival rate is 40 and the threshold is 25, and the largest error of UDF is 18% when the arrival rate is 40 and the threshold is 20). A reason for this

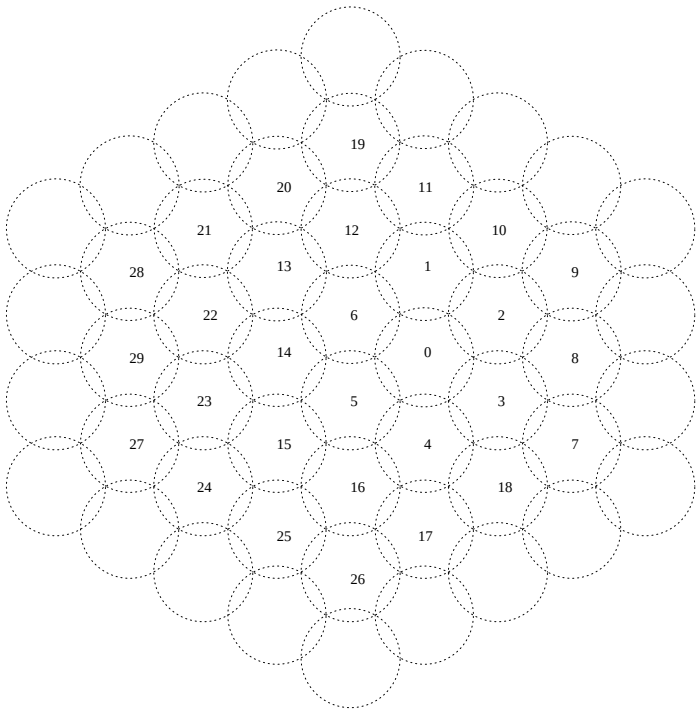


Figure 2.16. The cellular network used in simulation

is that the number of cells is not infinite, and thus, the effect of the boundary cells introduces the error. However, it is surprising to see the phenomenon that, even the distribution of connection-sojourn time is uniformly- or normally- distributed, the results are still consistent with our analytical model. We conjecture that the assumption of independent connection-sojourn times in each cell may possibly contribute to this result. Moreover, the insensitivity of P_b, P_f and DR to different mobility values (as shown in Figures 2.11 and 2.12) could also explain the independence of performance metrics (except UDF) from mobility distributions. This insensitivity to the distribution of mobility implies the applicability of our model to more general cases.

2.5 Conclusion

In this chapter, we derived an analytical model for wireless networks with multilevel adaptive bandwidth allocation and traffic-restriction admission control. Four QoS metrics — blocking probability, forced-termination probability, degradation ratio, and upgrade/degrade frequency — were derived. By using numerical analysis, we

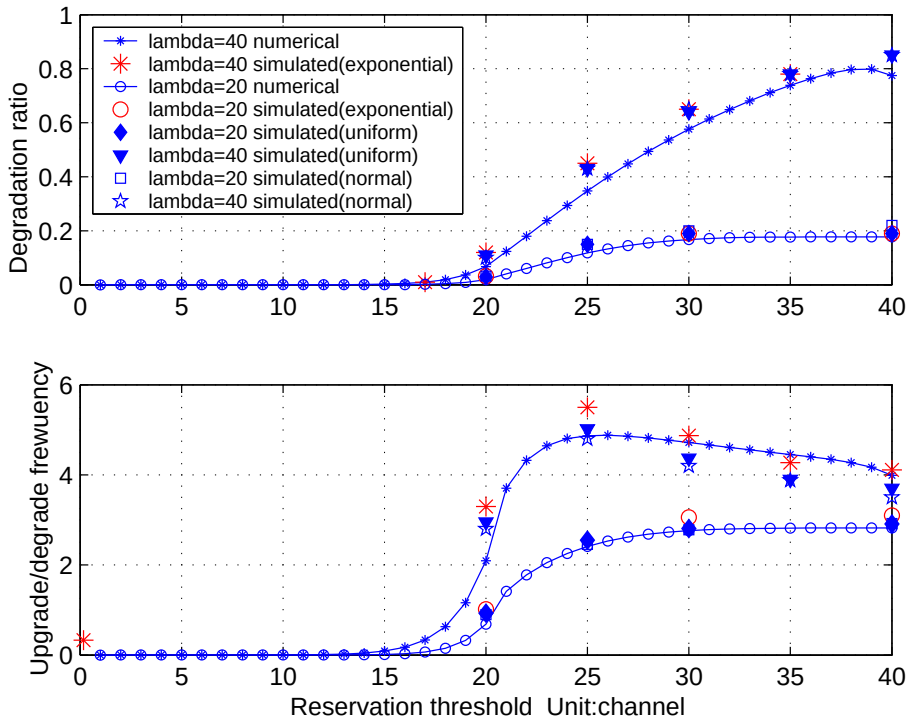


Figure 2.17. DR and UDF under different mobility models

demonstrated the effects of connection arrival rate, connection-holding time, and user mobility on these QoS metrics. A relatively fair admission control and bandwidth allocation algorithm was provided such that lower DR and UDF can be achieved with little increase of the blocking probability of new connections. Our simulation results demonstrated the applicability of our proposed model to the general case with different mobility models. This study provides an analytical framework for predictive or adaptive bandwidth allocation algorithms and helps decide the operating point under different traffic conditions. With this model, more complicated adaptive bandwidth allocation schemes can be analyzed, and their impacts on QoS can also be evaluated.

CHAPTER 3

Distributed Airtime Allocation in IEEE 802.11 Wireless LANs

In a network that supports adaptive QoS, user-/application-perceived bandwidth are subject to vary with the network load and capacity. In general, the user-perceived bandwidth can be changed rapidly through the medium access control (MAC). In a frequency-division-multiple-access (FDMA) network, user bandwidth is changed via redistributing the spectral bands. In a time-division or code-division-multiple-access (TDMA/CDMA) network, user bandwidth is changed via reassigning time slots or spreading sequences (e.g., the multi-code CDMA) to the users. Nevertheless, it is very difficult to change/adjust user bandwidth in a network using a distributed medium access control, such as the IEEE 802.11 wireless LAN. The IEEE 802.11 wireless LAN is a time-division system in the sense that only one user can transmit at any time instant. However, unlike the TDMA system where users transmit within the designated time slot(s) in a round-robin fashion, users in the 802.11 wireless LAN acquire transmission opportunities (i.e., an interval of airtime) using the carrier sense multiple access (CSMA) with collision avoidance (CA) and random backoff. Because of the distributiveness and randomness of the CSMA/CA, it is very difficult to control each user's airtime usage (and thus, the bandwidth), let alone dynamically adjusting user bandwidth for the purpose of adaptive QoS support.

In order to provide adaptive QoS in the IEEE 802.11 wireless LANs, we propose a distributed airtime usage control to facilitate the bandwidth adjustment. The main idea of the proposed control algorithm is to enhance the current CSMA/CA access method so that stations can choose their own CSMA/CA parameters based on the amount of required airtime. With the help of the proposed airtime usage control, the

user bandwidth can be adjusted in a distributed manner.

This chapter is organized as follows. Section 3.1 gives an overview of the IEEE 802.11 medium access control protocol while Section 3.2 discusses the difficulties of controlling users's airtime usage in IEEE 802.11 wireless LANs. Section 3.3 explains the proposed control algorithm and two analytical models are developed to determine the control parameters. Numerical and simulation results are discussed in Section 3.4 and finally, conclusions are drawn in Section 3.5.

3.1 Overview of the IEEE 802.11 Wireless MAC Protocol

The IEEE 802.11 MAC protocol defines two access methods, namely, the distribute coordinate function (DCF) and point coordinate function (PCF). The DCF is known as CSMA/CA and is the fundamental access method on both infrastructure and ad hoc network configurations. The infrastructure network configuration is composed of a station performing the role of access point (AP) and other stations communicating with each other via the AP, while the ad hoc network configuration is composed of stations having direct communication with each other. The PCF is essentially a polling-based access method with the AP performing the role of polling mater to determine which station has the right to transmit. Because of the need of a polling master, the PCF is only usable on infrastructure network configuration and is only an optional access method in the IEEE 802.11 standard. Therefore, we focus our discussion on the mandatory DCF in the rest of this chapter.

3.1.1 CSMA/CA with Random Backoff

In the DCF, a station desiring to initiate the transmission of MAC-layer frames invokes the carrier-sense mechanism to determine whether the medium is busy or idle. If the medium is determined to be idle, the station has to wait for a time duration required by the CSMA/CA algorithm before attempting any transmission. If the medium is determined to be busy, the station defers the transmission until the medium is determined to be idle. After this deferral, the station selects a random backoff interval and decrement the backoff timer while the medium is idle. In case

of a collision or after a successful transmission, the station also waits for a random backoff interval before attempting the next transmission. Once the random backoff timer is decremented to zero, the station can start its transmission.

The random backoff is designed to prevent stations from colliding with each other since stations may all try to use the medium at the end of deferral. The backoff time, BT, is determined by

$$BT = Random([0, CW]) \cdot \mathrm{a}SlotTime,$$

where CW is the station's contention window size and $aSlotTime$ is the duration of a time slot define in the standard. In order to minimize the possibility of collision, each individual station should choose its CW as follows.

1. CW takes an initial value of CW_{min}.
2. CW takes the next value in the series in Eq. (3.1) after an unsuccessful transmission attempt, until CW reaches its maximum value, CW_{max}.
3. Once it reaches CW_{max}, CW will remain there until it is reset.
4. CW will be reset to CW_{min} after (i) a successful transmission of a frame or (ii) the number of retransmission attempts reaches the retry limit. (An IEEE 802.11 station should retransmit any unsuccessful frame up to the number of times specified by the retry limit before discarding that frame).

According to the current IEEE 802.11b standard, the set of CW values should be a sequentially ascending integer power of 2 minus 1, beginning with CW_{min} and continuing up to CW_{max}:

$$\{CW = 2^j - 1 : j = K, K+1, \cdots, K+m\}, \tag{3.1}$$

where m is referred to as the *maximum backoff stage*, which decides the maximum contention window size a station can use, $CW_{min} = 2^K - 1$, and $CW_{max} = 2^{K+m} - 1$.

3.1.2 RTS/CTS/DATA/ACK Frame Exchange

Once acquiring the access to the medium, a station may send a data frame immediately or send a RTS frame first if the size of the data frame exceeds a predefined

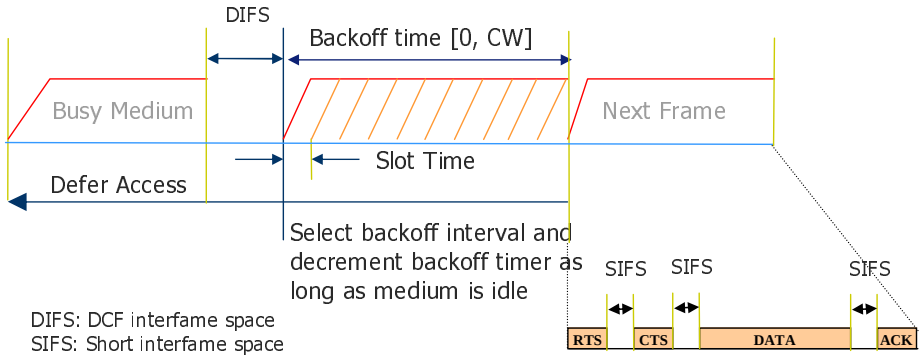


Figure 3.1. The basic DCF in an IEEE 802.11 wireless LAN

threshold. In case that a RTS frame is sent, the station to which the RTS frame is addressed must send a CTS frame to the station from which the received RTS frame is originated. This RTS/CTS frame exchange not only solves the well-known "hidden node" problem but also helps resolve a collision faster. After a successful RTS/CTS frame exchange, the transmission of data frame can proceed. If the transmission succeeds, the station to which the data frame is address sends back an acknowledgement frame (ACK) which concludes the data exchange procedure. The frame exchanges, along with the DCF access method, are illustrate in Figure 3.1.

3.2 Problems for Airtime Usage Control in IEEE 802.11 Wireless LANs

In a time-division system such as the IEEE 802.11 wireless LAN, stations obtain the QoS-required bandwidth by acquiring the corresponding amount of transmission time. Therefore, it is very important for stations to be able to acquire different amounts of transmission time to satisfy different QoS requirements. Unfortunately, the DCF only provides stations an egalitarian access to the wireless medium (and thus an equal share of the total transmission time), primarily due to the distributed CSMA/CA algorithm. As a result, it is impossible to provide QoS in IEEE 802.11 wireless LANs if stations use the basic DCF access method. The new IEEE 802.11e standard addresses this problem by adding an enhanced DCF to provide differential medium access. However, the precise control on each station's usage of transmission

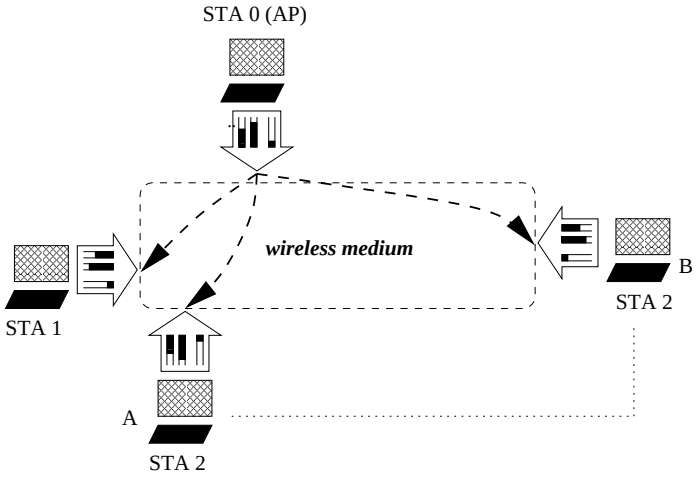


Figure 3.2. An infrastructure IEEE 802.11 wireless LAN

time— which is crucial to QoS provisioning — can still only be achieved by using the polling-based access method.

Another design that complicates the airtime usage control in the IEEE 802.11 wireless LANs is the station's support of multiple transmission rates. For example, an IEEE 802.11b station can transmit at 11, 5.5, 2 and 1 Mbps while an IEEE 802.11a station can transmit at up to 8 different rates. In general, the multi-rate support of the IEEE 802.11 standard is integrated with the link adaptation mechanism. The link adaption is an adaptive rate-control mechanism used by stations to improve transmission efficiency. The idea of link adaption is very simple: a station should use a lower rate for reliable transmission when the channel condition is bad, and use a higher rate to achieve higher transmission efficiency when the channel condition is good. With the link adaption, individual stations in an IEEE 802.11 wireless LAN may use different transmission rates based on the channel conditions. As a result, different stations may occupy the medium for different amounts of time to transmit a data frame, after winning a contention of the medium.

To illustrate how the multi-rate support and link adaption affect the airtime usage control, let us use the IEEE 802.11b wireless LAN as an example. As shown in Figure 3.2, three stations — the AP, STA 1 and STA 2 — consist of an infrastructure IEEE 802.11 wireless, with each being able to transmit at 11, 5.5, 2 or 1 Mbps. We assume that STA 1 and STA 2 communicate with the AP at 11 Mbps before $t = 4$.

As a result, each station use 50% of the total airtime and obtains a bandwidth (i.e., throughput) of 5.5 Mbps.[1] After $t = 4$, STA 2 moves from point A to point B, and adapts its transmission rate to 1 Mbps due to the poor reception. Although STA 1 still transmits as frequently as STA 2 after $t = 4$, STA 1 uses 10 times less airtime than STA 2 does during each possession of the medium. As a result, STA 1 and STA 2 use 9.1% and 90.9% of the total time, respectively, with each receiving a bandwidth equal to 0.909 Mbps. If STA 1 has to provide at least 1 Mbps for certain applications, the bandwidth reduction is unacceptable.

The above example shows that due to the lack of airtime usage control in the IEEE 802.11 wireless LAN, the low-rate station could "overuse" the system airtime easily via the link adaption. As a result, both the low-rate (e.g., STA 2) and high-rate stations (e.g., STA 1) suffer the bandwidth reduction. All though the low-rate station is doomed to loss the bandwidth because of lowering the transmission rate, the high-rate station should not be affected be the low-rate station for the sake of QoS provisioning.

3.3 Distributed Airtime Usage Control

The objective of airtime usage control is to ensure that each station obtains the required amount of airtime throughout the station's service interval. Let $T_i(t_1, t_2)$ be the amount of airtime station i receives in a time interval $(t_1,\ t_2)$, and ϕ_i be the share decided by network conditions and QoS requirements. A perfect airtime usage control should satisfy

$$\frac{T_i(t_1, t_2)}{T_j(t_1, t_2)} \geq \frac{\phi_i}{\phi_j}, \tag{3.2}$$

if station i is continuously backlogged during $(t_1,\ t_2)$. Let $B_i(t_1, t_2)$ be the bandwidth received by station i's within the time interval (t_1, t_2). We have

$$\frac{B_i(t_1, t_2)}{B_j(t_1, t_2)} = \frac{r_i \cdot T_i(t_1, t_2)}{r_j \cdot T_j(t_1, t_2)}, \tag{3.3}$$

where r_i is the physical transmission rate of station i. Eq. (3.3) shows that by controlling station's airtime usage, the bandwidth received by each station can be

[1]For simplicity, we ignore all control overhead and assume that there is no collision.

controlled and adjusted easily. Next, we will show how to achieve Eq. (3.2) in the distributed, multi-rate IEEE 802.11 wireless LANs.

3.3.1 Control Parameters: $AIFS$ vs. CW_{min}

As mentioned in Section 3.1, stations in an IEEE 802.11 wireless LAN contend for the medium using the mandatory DCF access method. Because all stations adopt the same DCF parameters, the stations have the same "chance" to acquire the medium and thus, have an equal share of the total airtime. In order to provide stations differentiated medium access so as to realize airtime usage control, one can either (1) control the mount of airtime each station can use during each possession of the medium, or (2) control the contention process so that the stations access the medium at different "rates". By using method (1), the airtime usage control in Eq. (3.2) can be achieved by

$$\frac{TXOP_i}{TXOP_j} = \frac{\phi_i}{\phi_j}, \tag{3.4}$$

where $TXOP_i$ is the amount of airtime station i can use during each possession of the medium. Although this method provides a simple and effective control on stations' airtime usage, the value of $TXOP_i$ may have some impact on the delay performance. For example, if some stations have large $TXOP_i$ (e.g., large ϕ_i), other stations may have to wait a long period of time before the medium is released (by the stations with large $TXOP_i$) for contention again. In method (2), since each station will use the same $TXOP_i$, the problem of the potential long delay can be minimized. By using method (2), the airtime usage control in Eq. (3.2) can be achieved by

$$\frac{AR_i}{AR_j} = \frac{\phi_i}{\phi_j}, \tag{3.5}$$

where AR_i is station i's medium accessing rate. In order to control the medium accessing rate, stations have to use different DCF parameters, such as inter-frame spacing, minimum or maximum contention window size. In the IEEE 802.11e, a similar mechanism, called Enhanced Distributed Channel Access (EDCA), is proposed to provide stations prioritized medium access. However, our intention here is to control these DCF parameters so as to provide a precise, quantitative airtime usage control.

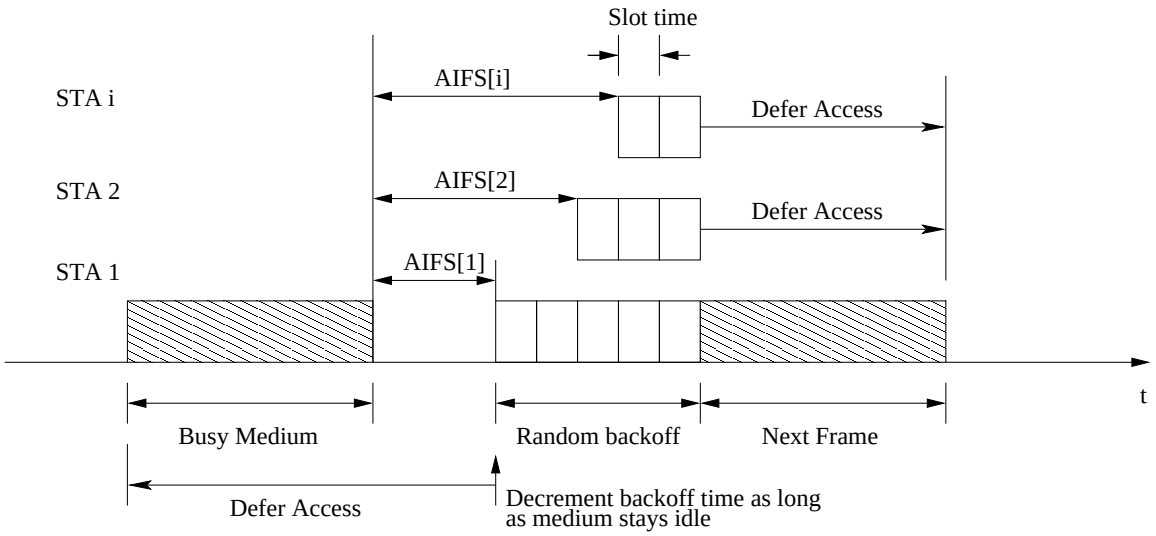


Figure 3.3. Distributed medium access in an IEEE 802.11 wireless LAN

In what follows, we focus our discussion on how to use the method (2) to control stations' airtime usage in a distributed manner.

Figure 3.3 shows the enhanced DCF and the parameters that can be use to control the value of R_i. Compared to the original DCF shown in Figure 3.1, the new DCF allows each station to have its own DCF parameters. By manipulating these parameters, different stations will have different opportunities to acquire the medium, and thus, obtain the required airtime. How to choose these parameters in order to achieve a target airtime usage ratio, however, is never an easy task. Let us consider two stations, STA 1 and STA 2, and assume that they use different AIFS values and contention window sizes. As shown in Figure 3.4, a relation between stations' backoff times can be found if we consider the time interval between two collisions

$$\sum_{i=1}^{n_1} BT_i^{(1)} = \sum_{j=1}^{n_2} BT_j^{(2)} + \sum_{h=1}^{n_1+n_2-1} D_h, \tag{3.6}$$

where $BT_i^{(j)}$ is the i-th backoff time chosen by STA j, n_i represents the total number of times STA i has backed off during this time interval and D_h is referred to as the "decrementing lag" as STA 2 has to wait longer than STA 1 before decrementing its backoff time. In Eq. (3.6), n_i is proportional to a station's airtime usage because whenever its backoff time is decremented to zero, the station is allowed to transmit a frame. The value of random backoff time B_i is determined by the station's contention

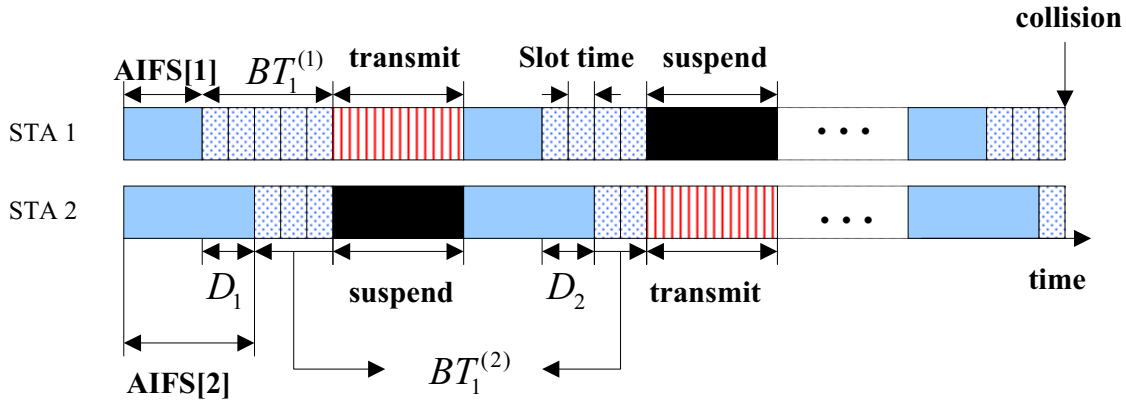


Figure 3.4. Stations' random backoff times between collisions

window parameters, CW_{min}, CW_{max} and *retry limit*, while the decrementing lag is mainly decided by the AIFS value.[2] That is, Eq. (3.6) gives the relation of airtime usage, backoff parameters and AIFS values. Based on this relation, we can choose appropriate parameters in order to control a station's airtime usage.

3.3.2 Controlling AIFS Time

According to Eq. (3.6), we may control stations' airtime usage with the decrementing lag. The only problem is that we do not have direct control over the decrementing lag. Let us assume that STA 1 has a smaller AIFS than STA 2 and AIFS[2]−AIFS[1]=2 time slots. Every time STA 2 starts to decrement its backoff time, STA 1 has already decremented its backoff time by 2 time slots. One may mistakenly think that D is a constant (i.e., D= AIFS[2]-AIFS[1] = 2 in our example), but D is in fact a random variable with possible integer values between 1 and AIFS[2]-AIFS[1]. For example, if STA 1 chooses a backoff time less than 2 (say 1), STA 2 will not even have any chance to start decrementing its backoff time before STA 1 finishes its transmission. In this case, $D = 1$. Therefore, we need a relation between stations' AIFS values and the decrementing lag in order to use AIFS for airtime usage control. We will detail this later in this subsection.

Eq. (3.6) can be extended to the general case of $N > 2$ stations. Here, the

[2]We adopt the notation of AIFS — arbitration interframe space — as in the IEEE 802.11e standard.

stations with the same airtime usage ratio and transmission rate should use the same parameters. Let K_i be the number of stations with ratio ϕ_i. We assume that $\phi_i > \phi_j$ if $i < j$ so that the station with a airtime share ϕ_1 should have the smallest AIFS value (i.e., AIFS[1]), the station with ratio ϕ_2 has the second smallest AIFS value (i.e., AIFS[2]), and so on. In the steady state, Eq. (3.6) can be rewritten as

$$E[n_1]E[BT^{(1)}] = (\sum_{j=1}^{N} K_j E[n_j] - E[N_{col}])E[D^{(k)}] + E[n_k]E[BT^{(k)}], \tag{3.7}$$

for $k = 2$ to N. Here, $E[D^{(k)}]$ is the average "decrementing lag" of stations with ratio ϕ_k, as compared to the stations with the smallest AIFS, and $E[N_{col}]$ is the average number of collisions within the observed interval. In order to emphasize the effects of AIFS values on the stations' airtime usage, we further assume that all stations use the same CW_{min} and CW_{max} values. The effect of these values on stations' airtime usage will be thoroughly investigated in the next subsection. Under this assumption, Eq. (3.7) can be further rewritten as:

$$E[n_1]\frac{CW_{\min}}{2} \approx \sum_{j=1}^{N} K_j E[n_j]E[D^{(k)}] + E[n_k]\frac{CW_{\min}}{2}. \tag{3.8}$$

Here, we simply substitute $\frac{CW_{\min}}{2}$ for $E[BT^{(i)}]$ and assume $\sum_{j=1}^{N} K_j E[n_j] \gg E[N_{col}]$. This is reasonable because the random backoff process is designed to minimize (especially, consecutive) collisions. The probability that a station collides with others more than twice in a row is very small. Later, we will show how to calculate $E[BT]$ when too many collisions occur.

By solving the system of linear equations in Eq. (3.8), the ratio of each individual station's airtime usage ($\propto \frac{E[n_i]}{E[n_1]}$) can be obtained. The desired airtime usage can be achieved by adjusting the AIFS values as follows.

1. Start with an initial set of AIFS values. The initial AIFS values (or more precisely, their differences) are determined by solving Eq. (3.8) with $E[D^{(k)}]$ replaced by $AIFS[k] - AIFS[1]$ and $E[n_k]$ replaced by $\phi_k r_k$.
2. Stations k's number of channel accesses, $E[n_k]$, is computed by solving Eq. (3.8) with the chosen AIFS values. If $\frac{E[n_k]\cdot r_k}{E[n_1]\cdot r_1} \approx \frac{\phi_k}{\phi_1}$, the current set of AIFS values are the parameters we want.

3. If $\frac{E[n_k]\cdot r_k}{E[n_1]\cdot r_1} \neq \frac{\phi_k}{\phi_1}$, a station's AIFS value is incremented if its airtime usage is larger than its assigned share, and decremented otherwise. Repeat Step 2.

As mentioned earlier, the real challenge is how to determine a station's decrementing lag, $E[D^{(k)}]$, based on the given AIFS values. Let us revisit the previous example and assume AIFS[2]−AIFS[1] = d time slots. As shown in Figure 3.5-(a), if STA 1 chooses its first backoff time, $BT_1^{(1)}$, between 1 and $d-1$, the first decrementing lag of STA 2, D_1, will be equal to $BT_1^{(1)}$ because STA 2 is supposed to wait $(d - BT_1^{(1)})$ more time slots before decrementing its backoff. If $BT_1^{(1)} > d$, the computation of decrementing lag is more complicated, but still can be approximated as follows:

- If $BT_1^{(1)} - d < BT_1^{(2)}$, STA 1 will win the current round of contention, and thus, $D_1 = d$.
- Otherwise, STA 2 will win the current round of contention. Since STA 1 still has a nonzero backoff time, it may result in STA 2's second decrementing lag, $D_2 < d$ after STA 2 finishes its current transmission, if $BT_1^{(1)} - d - BT_1^{(2)} < d$ as illustrated in Figure 3.5-(b).

The average decrementing lag for $BT_1 \geq d$ can be calculated as

$$E[D^{(2)}|BT_1 \geq d] = \frac{(CW - (d-1)) \cdot d}{CW} + \frac{\sum_{i=1}^{(d-1)} i}{CW} \tag{3.9}$$

given that the stations choose their own backoff times uniformly from their contention windows. Finally, combining the cases (a) and (b) in Figure 3.5, the average value of $D^{(2)}$ can be calculated as

$$E[D^{(2)}] \approx d - [\frac{d(d-1)}{CW_{\min}} - \frac{d(d-1)^2}{2CW_{\min}}]. \tag{3.10}$$

Here, we calculate the average decrementing lags based on an implicit assumption that if STA 1 "loses" the current round of channel contention (i.e., $BT_1^{(1)} < BT_2^{(2)} + d$), it will "win" the next round (i.e., $BT_1^{(1)} < BT_1^{(2)} + BT_2^{(2)} + 2d$). More precise calculation can be done by considering other possibilities and the difference will be more higher-order terms in Eq. (3.10). Our simulation results in the next section show that a very good estimation of D can be obtained without considering those higher-order terms.

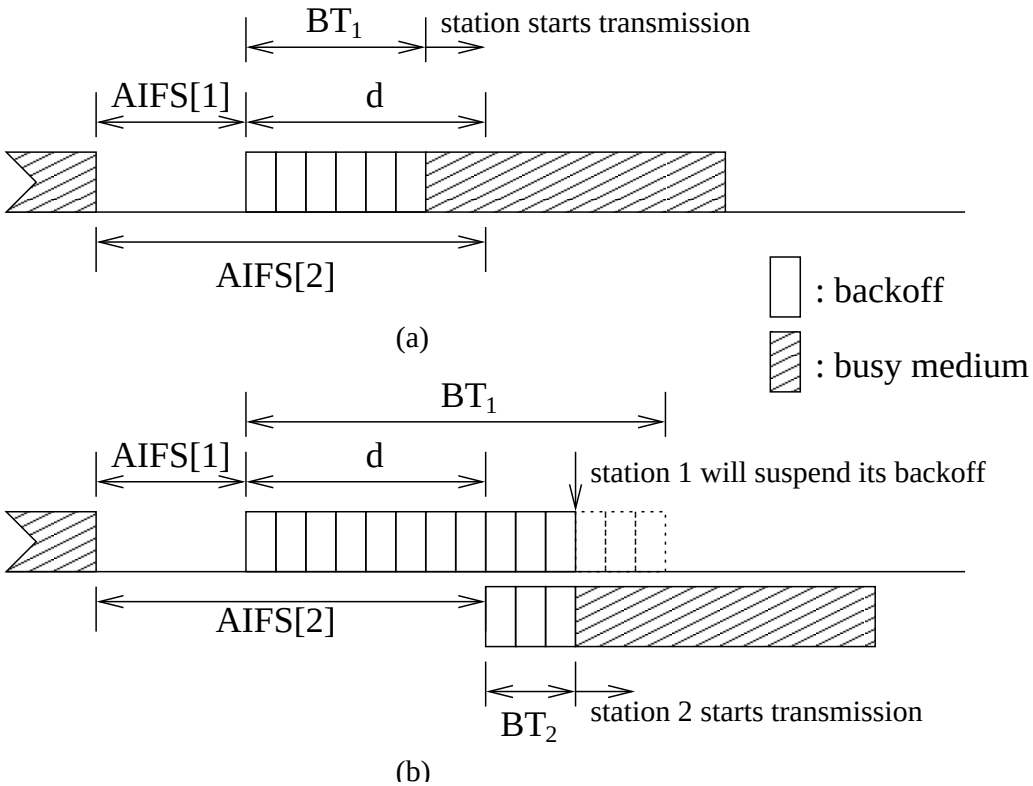


Figure 3.5. Station-2's backoff decrement delay

There are some interesting points to make on the effect of contention window size on the decrementing lag. First, if we choose a very large CW_{min}, STA 2's decrementing lag should be very close to AIFS[2]-AIFS[1]=d since it is very unlikely for station 1 to choose a backoff time less than d. This can be observed in Eq. (3.10). Second, the term inside the square brackets of Eq. (3.10) represents the contribution of STA 1's backoff process to STA 2's decrementing lag. If there are more than one station with smaller AIFS (i.e., AIFS[1]), STA 2's decrementing lag should be smaller because it is more likely for at least one of those stations to choose a random backoff time smaller than d. In fact, STA 2's average decrementing lag can be calculated as above by using the concept of union bound [75]

$$E[D^{(2)}] \approx d - [\frac{d(d-1)}{CW_{min}} - \frac{d(d-1)^2}{2CW_{min}}] * K_1. \tag{3.11}$$

Finally, it should be noted that the number of stations with the same or larger AIFSs will not affect a station's decrementing lag because they are only allowed to decrement their backoff time after this station starts decrementing its backoff time. With this

property and Eq. (3.11), the decrementing lag of the stations with the AIFS values equal to AIFS[k] can be approximated as

$$E[D^{(k)}] = d_1^{(k)} - \sum_{i=1}^{k-1}[\frac{d_i^{(k)}(d_i^{(k)}-1)}{CW_{\min}} - \frac{d_i^{(k)}(d_i^{(k)}-1)^2}{2CW_{\min}}] * K_i, \tag{3.12}$$

where $d_i^{(k)}$=AIFS[k]-AIFS[i] for $i = 1$ to $k-1$. Even though the derivation of Eq. (3.10) and the use of union bound introduce an estimation error to Eq. (3.12), we will show later that it matches the simulation results very well.

3.3.3 Controlling CW_{min} and CW_{max}

In addition to resolving collisions, the random backoff mechanism can also be used to control each station's share of system airtime. In this subsection, we assume all stations use the same AIFS value but use different backoff parameters for differentiated channel accesses. Eq. (3.6) can then be rewritten as

$$\sum_{i}^{n_1} BT_i^{(1)} = \sum_{j}^{n_2} BT_j^{(2)}. \tag{3.13}$$

By taking the expected values of both sides in this equation, we have

$$\frac{E[n_1]}{E[n_2]} \approx \frac{CW_{min}^{(1)}}{CW_{min}^{(2)}}. \tag{3.14}$$

Again, we use $\frac{CW_{min}^{(1)}}{2}$ as the mean value of STA 1's random backoff times $BT_i^{(1)}$'s as we did in the previous subsection. Eq. (3.14) shows that the airtime usage ($\propto n_i$) of a station is approximately inversely proportional to its minimum contention window size. This property provides us an easy way to control each station's share of airtime in a distributed manner. A similar relation can also be found in [23], but the exponential increment of contention window and the reset mechanism of contention window were not considered. In fact, the random backoff process is far more complicated because the contention window size needs to be adjusted, depending on the outcome of each transmission attempt. Even though one can expect that the mean value of a station's random backoff time is close to $\frac{CW_{min}}{2}$ because of the small collision probability, precise control over station's airtime usage cannot be achieved without including

the exponential increment and reset mechanism of stations' contention window sizes, especially when the number of stations in a wireless LAN is large.

In order to accurately analyze each station's share of airtime, we propose an enhanced model based on a previous DCF model [84, 28]. The station's backoff process is observed whenever at least one station's backoff time changes in the wireless LAN, i.e., at the end of an idle slot or at the end of a transmission/collision. Each station's backoff process is represented by a state vector, $(w_i(t), b_i(t))$, at these particular time points. The contention window index of station i, $w_i(t)$, takes the value of j in Eq. (3.1), while $b_i(t)$ is station i's backoff time, in number of slot times. The resulting process, $\{(w_i(t), b_i(t)) : t = t_1, t_2, \cdots\}$ for station i can then be modelled as a 2-dimensional discrete-time Markov chain as suggested in [28]. Our Markovian model differs from their models as follows.

1. When a station has a nonzero backoff time, it will not decrement its backoff time until the medium has been idle for a slot time. If some stations decrement their backoff time to zero at one observation point, the backoff time of the other stations should remain unchanged at the next observation point which is the end of current transmission. Stations finishing their transmission will choose their own new backoff times according to the outcome of their transmission attempts. Therefore, the transition probability from $(w(t_j) = w, b(t_j) = i)$ to $(w(t_{j+1}) = w, b(t_{j+1}) = i - 1)$ for a station with a nonzero backoff time is less than 1. The probability should be computed as

$$P[w(t_{j+1}) = w, b(t_{j+1}) = i - 1 | w(t_j) = w, b(t_j) = i] = P[b_k(t_j) > 0 \text{ for all} k \neq i]. \quad (3.15)$$

   This key property in the random backoff has been overlooked in [28], and the above transition probability was assumed to be 1 there.

2. The reset mechanism of the contention window size, after the number of retransmissions reaches the retry limit, is also included. We can therefore study the effects of retry limit on a station's access of the channel. A small retry limit

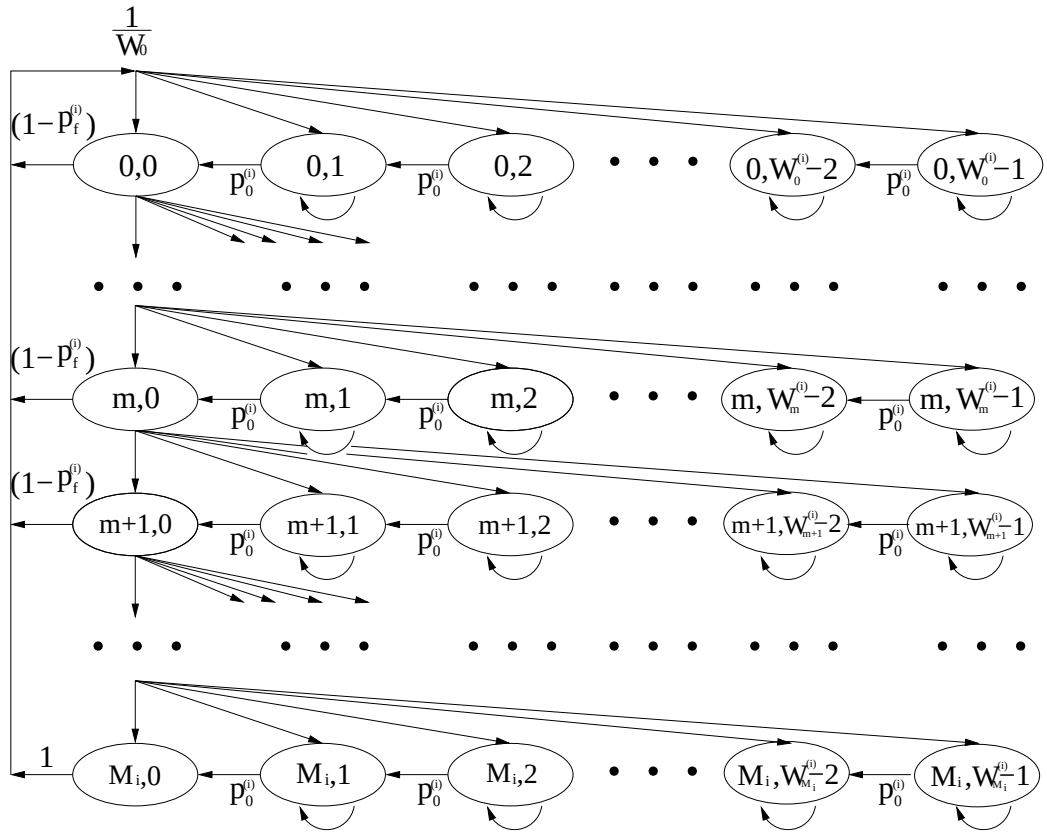


Figure 3.6. Markov model for the enhanced DCF.

can effectively reduce a station's average backoff time because the contention window size gets reset after a few retransmission attempts. A station with a smaller retry limit is likely to have more transmission opportunities. In fact, the retry limit plays a more important role than CW_{max} in a backoff process. If the retry limit is less than the maximum backoff stage, the maximum contention window size a station can use is decided by the retry limit; otherwise, the retry limit determines how many times a station can use CW_{max} before resetting it. This resetting mechanism has not been considered in [84, 28].

3. Each station is allowed to have different values of CW_{min}, CW_{max}, and *retry limit*. In the IEEE 802.11e standard, these values depend on the priority level of a station/application, and our model can handle this general case.
4. The transmission error is included in our model. Moreover, each station may have different transmission-error probabilities (i.e., location-dependent error).

We now consider a tagged station i with the backoff parameters, $W_0 = W_0^{(i)}$,[3] $m = m_i$, *retry limit* $= n_i$, and the probability of transmission errors is $p_e^{(i)}$. If $n_i \geq m_i$, the contention window index may remain unchanged for $(n_i - m_i)$ times after it reaches m_i, because station i will use $CW_{max}^{(i)}$ as the contention window to retransmit a frame up to $(n_i - m_i)$ times. Otherwise, the contention window of station i will never reach $CW_{max}^{(i)}$; instead, the maximum value it can reach is $2^{n_i} \cdot W_0^{(i)} - 1$, and will be reset thereafter regardless whether the transmission succeeds or fails. Figure 3.6 shows the resulting Markov chain with the transition probabilities being computed as follows.

1. After a successful transmission, station i will reset its contention window, and choose a new backoff time:

$$P[0, k|j, 0] = \frac{1 - p_f^{(i)}}{W_0^{(i)}}, \ \forall \ j \neq M_i,$$

for $0 \leq k \leq W_0^{(i)} - 1$. Here, $p_f^{(i)} = 1 - (1 - p_c^{(i)})(1 - p_e^{(i)})$ is the probability of transmission error, $p_c^{(i)}$ is the collision probability of station i, and $M_i = \min(n_i, m_i)$ is the maximum contention window index.

2. After an unsuccessful transmission attempt, station i will use the next contention window in the series of Eq. (3.1), and choose a new backoff time:

$$P[j + 1, k|j, 0] = \frac{p_f^{(i)}}{W_{j+1}^{(i)}}, \ \forall \ j \neq M_i,$$

for $0 \leq k \leq W_{j+1}^{(i)} - 1$.

3. Station i will reset its contention window after the number of retransmissions for a frame reaches n_i, and will randomly choose a new backoff time:

$$P[0, k|M_i, 0] = \frac{1}{W_0^{(i)}}, \ 0 \leq k \leq W_0^{(i)} - 1.$$

4. Station i decrements its backoff time only when all the other stations have nonzero backoff times:

$$P[j, k - 1|j, k] = p_0^{(i)}, \ \forall \ j, \ 1 \leq k \leq W_j^{(i)} - 1,$$

where p_0^i is the probability perceived by station i that the medium is idle.

[3] $CW_{max} = W_0 - 1$

Let $p_{m,n}^{(i)} = \lim_{t\to\infty} P[w_i(t) = m, b_i(t) = n]$ represent the stationary distribution of the Markov chain for station i, where $m = 0, 1, \cdots, M_i$ and $n = 0, 1, \cdots, W_m^{(i)} - 1 = 2^m \cdot W_0^{(i)} - 1$. The following recursive relation holds in the steady state:

$$p_{m-1,0}^{(i)} p_f^{(i)} = p_{m,0}^{(i)},\ 0 < m \le M_i. \tag{3.16}$$

By using Eq. (3.16), we can obtain

$$p_{m,0}^{(i)} = (p_f^{(i)})^m p_{0,0}^{(i)}\ 0 < m \le M_i. \tag{3.17}$$

From the structure of the Markov chain, the following relations can also be found. For $n \in \{1, \cdots, W_0^{(i)} - 1\}$,

$$p_{0,n}^{(i)} = \frac{W_0^{(i)} - n}{W_0^{(i)} \cdot p_0^{(i)}} \left[(1 - p_f^{(i)}) \cdot \sum_{k=0}^{M_i - 1} p_{k,0}^{(i)} + p_{M_i,0}^{(i)} \right], \tag{3.18}$$

while for $0 < m \le M_i$ and $n \in \{1, \cdots, W_m^{(i)} - 1\}$,

$$p_{m,n}^{(i)} = \frac{W_m^{(i)} - n}{W_m^{(i)} \cdot p_0^{(i)}} p_f^{(i)} p_{m-1,0}^{(i)}. \tag{3.19}$$

Substituting Eq. (3.17) into Eqs. (3.18) and (3.19), we can obtain

$$p_{m,n}^{(i)} = \frac{W_m^{(i)} - n}{W_m^{(i)} \cdot p_0^{(i)}} p_{m,0}^{(i)} \forall\ m,\ n \in \{1, 2, \cdots, W_m^{(i)} - 1\}. \tag{3.20}$$

Finally, $p_{0,0}^{(i)}$ can be obtained by using $\sum_m \sum_n p_{m,n}^{(i)} = 1$, Eqs. (3.17) and (3.20):

$$p_{0,0}^{(i)} = \left[(1 - \frac{1}{2p_0^{(i)}}) \frac{1 - (p_f^{(i)})^{n_1}}{1 - p_f^{(i)}} + \frac{W_0^{(i)}}{2p_0^{(i)}} \frac{1 - (2p_f^{(i)})^{n_1}}{1 - 2p_f^{(i)}} \right.$$
$$\left. + (p_f^{(i)})^{m_i+1} (1 + \frac{2^{m_i} W_0^{(i)} - 1}{2p_0^{(i)}}) \frac{1 - (p_f^{(i)})^{n_2}}{1 - p_f^{(i)}} \right]^{-1} \tag{3.21}$$

where $n_1 = \min(n_i, m_i) + 1$ and $n_2 = \max(0, n_i - m_i)$.

One should note that $p_0^{(i)}$ and $p_f^{(i)}$ themselves are functions of $p_{0,0}^{(i)}$. Let $p_t^{(i)} = \sum_{m=0}^{M_i} p_{m,0}^{(i)}$ represent the probability that station i transmits a frame. Then, we have

$$p_0^{(i)} = \prod_{\forall\ k \ne i} (1 - p_t^{(k)}), \tag{3.22}$$

and

$$p_c^{(i)} = 1 - p_0^{(i)}. \tag{3.23}$$

So, a system of nonlinear equations with N parameters has to be solved for a wireless LAN with N stations, if all stations have different backoff parameters.

3.3.4 Optimal Random Backoff Parameters

The optimal random backoff parameters, CW_{min} and CW_{max}, for a given set of stations' airtime usage ratios can be determined by the model established in the previous subsection. In fact, a station's airtime usage ratio is proportional to the product of its probabilities of successfully transmitting a frame and the inverse of the current transmission rates. The probability that station i transmits a frame successfully can be obtained by

$$p_s^{(i)} = p_t^{(i)} \cdot \prod_{\forall\ k-\{i\}} (1 - p_t^{(k)}), \tag{3.24}$$

when $p_t^{(k)}$ is given right before Eq. (3.22). All the probabilities $p_{0,0}^{(i)}$, $p_0^{(i)}$, and $p_f^{(i)}$ in Eq. (3.21) can also be represented by $p_t^{(k)}$, using Eqs. (3.17), (3.22), and (3.23), respectively. Therefore, the optimal parameters for the given airtime usage ratios can be obtained as follows.

1. Start with an initial set of CW_{min} values according to Eq. (3.14). The minimum value of CW_{min} used for the following numerical analysis and simulations is 31.
2. The values of CW_{min} $(= W_0 - 1)$ from Step 1 is substituted in Eq. (3.21), where all probabilities are represented by $p_t^{(k)}$.
3. We compute $p_t^{(k)}$ for $k = 1, \cdots, K$ by solving the system of equations obtained from Step 2. The resultant medium access probability $p_s^{(i)}$ can then be obtained by Eq. (3.24).
4. If $\frac{E[n_k] \cdot r_k}{E[n_1] \cdot r_1} \approx \frac{\phi_k}{\phi_1}$, the current CW_{min}'s are the required values.[4] Terminate the procedure.
5. Else if a station's ratio is larger than the assigned value, increment its CW_{min}. Otherwise, decrement its CW_{min}. Repeat Step 2.

CW_{max}'s are determined according to Eq. (3.1) so that all stations can have similar CW_{max} values. For most of our analyses, we were able to obtain the optimal random backoff parameters in less than 5 iterations.

[4]Within 1% error.

SIFS	DIFS	Slot Time	Preamble length	Packet length	ACK length	MAC heard + CRC
10 usecs	50 usecs	20 usecs	144 usec	1500 bytes	14 bytes	34 bytes

Table 3.1. The parameters for simulation

3.4 Numerical and Simulation Results

We consider an IEEE 802.11 wireless LAN operating in the DCF mode. It is assumed that each station can only transmit/receive frames to/from the AP (i.e., in the infrastructure mode) and may transmit at 11, 5.5, 2, and 1 Mbps. Furthermore, we assume that all frames have the same length and $retry\ limit = 7$. All stations are assumed to be continuously backlogged and each station can only transmit one frame on each transmission opportunity. In order to verify our analytical models, we also implement the DCF mode of an IEEE 802.11 wireless LAN. Only the kernel parts of the DCF mode, namely, the CSMA/CA and exponential random backoff are simulated. We do not include the RTS/CTS and ACK frames, but the associated overheads are considered when calculating the throughput. The simulation is conducted by using an event-driven scheduler written in Matlab code. The parameters used in the simulation are based on the IEEE 802.11b [65] standard and are summarized in Table 3.1.

3.4.1 Control of Stations' Airtime Usage by Using AIFS

Before presenting the results of stations' airtime usage, we first give an example to show the accuracy of Eq. (3.12). Table 3.2 shows the average decrementing lags for the case where there are four types of stations with different airtime usage ratios. Here, we assume that AIFS[i]$-$AIFS[$i-1$]=2 for $i = 2$ to 4 and we change the number of type-i stations, K_i, to investigate their effects on the decrementing lag. Even though deriving Eq. (3.12) needs some approximations, the results show that the estimation error is small. The error may result from the use of $\frac{CW_{min}}{2}$ to approximate the mean value of a station's backoff time and this may not be accurate enough because of the exponential increment and reset mechanism of contention window size in the 802.11

(K_1, K_2, K_3, K_4)	E[$D^{(2)}$]	E[$D^{(3)}$]	E [$D^{(4)}$]
(1, 1, 1, 1)	1.96	3.80	5.42
	1.969	3.817	5.35
(2, 1, 1, 1)	1.93	3.65	5.08
	1.937	3.63	4.91
(3, 1, 1, 1)	1.90	3.50	4.77
	1.906	3.45	4.57
(4, 1, 1, 1)	1.88	3.37	4.51
	1.89	3.31	4.17
(4, 2, 1, 1)	1.88	3.38	4.47
	1.89	3.29	4.04

* The value in the shaded columns are the simulation results.

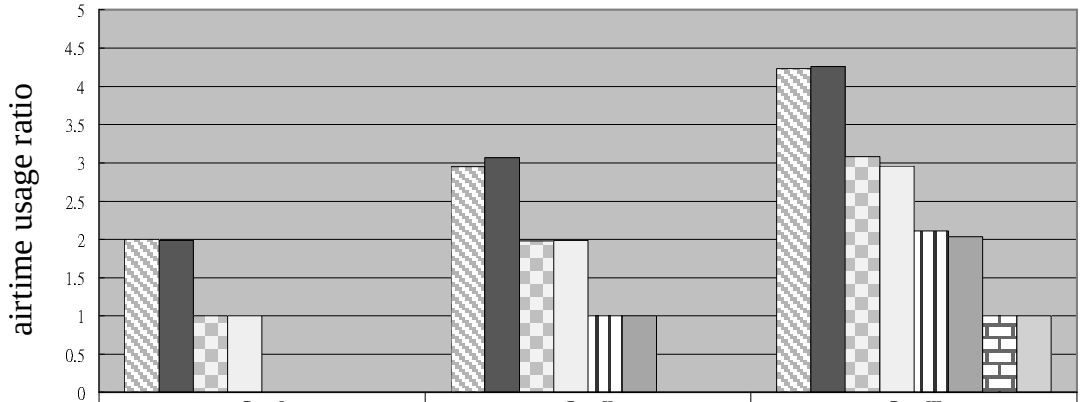
Table 3.2. Decrementing lag: $N = 4$ and $AIFS[i] - AIFS[i-1] = 2$.

standard. This problem can be alleviated by

$$E[BT] \approx (1 - \frac{\sum_i K_i}{CW_{\min}})\frac{CW_{\min}}{2} + \frac{\sum_i K_i}{CW_{\min}} CW_{\min}, \tag{3.25}$$

to include the effects of collisions and the subsequent exponential increase of stations' backoff times. Here, $\frac{\sum_i K_i}{CW_{\min}}$ accounts for the collision probability and $CW_{\min}$ represents the average backoff time a station may choose after the first collision. We do not consider the effect of exponential increase of CW resulting from more than 2 consecutive collisions because they rarely occur. In general, Eq. (3.12) gives a very good estimation of $E[D^{(i)}]$. In this example, the largest estimation error occurs when $K_1 = 4$, $K_2 = 2$, $K_3 = 1$ and $K_4 = 1$ but it is only about 10% .

Next, we show that small differences among stations' AIFS values suffice to provide differentiated airtime usage to different stations. We consider three different cases in which the wireless LAN provides 2 (Case I), 3 (Case II) and 4 (case III) classes, respectively. Stations in each class will be allocated the same amount of airtime. In Case I, we assume $K_1 = K_2 = 3$ and choose $AIFS[2] - AIFS[1] = 4$ according to Section 3.3.2 so that a station in the first class can have twice the airtime of another station in the second class. In Case II, we set AIFS[2]−AIFS[1]=3 and



	Case I	Case II	Case III
class 1 (numerical)	2	2.95	4.23
class 1 (simulated)	1.99	3.07	4.255
class 2 (numerical)	1	1.98	3.08
class 2 (simulated)	1	1.99	2.955
class 3 (numerical)		1	2.11
class 3 (simulated)		1	2.035
class 4 (numerical)			1
class 4 (simulated)			1

Figure 3.7. The stations' airtime usage by controlling AIFS values

AIFS[3]−AIFS[3]=4 so that the ratio of stations' airtime in each class is close to 3:2:1, given that there are 2 stations in each class. Finally, we consider a airtime ratio as 4:3:2:1, given that there are 2 stations in each class in Case III. In this case, AIFS[2]−AIFS[1]=2, AIFS[3]−AIFS[2]=2, and AIFS[4]−AIFS[3]=3. The achievable ratio (by the chosen AIFS value) and the simulation results are plotted in Figure 3.7. The results match well with each other (with the largest error $\approx 6\%$) and show that a small difference among stations' AIFS values suffices to achieve the desired airtime allocation. One of the reasons why we cannot obtain the exact ratio in Case III (4.25 instead of 4.00 in Case III) is that we only use integer multiple AIFS values (to be multiples of time slot). If we are allowed to use any value, the exact ratio can be achieved.

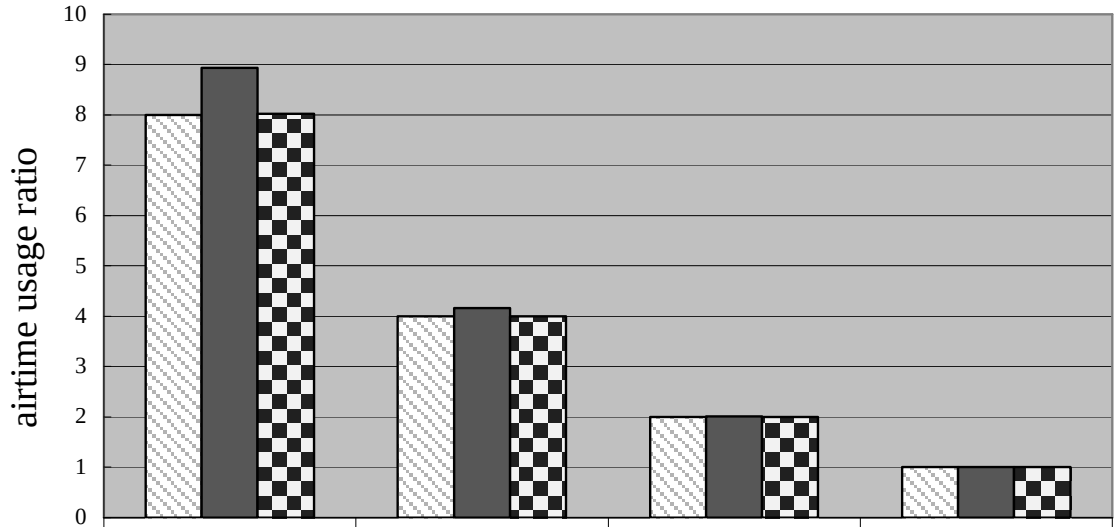
3.4.2 Control of Stations' Airtime Usage by Using CW_{min}

Two sets of analysis are conducted in this subsection — both have 4 different airtime usage ratios assigned to different stations. Consider the first set (Case I and II), in which there are 8 stations with their assigned airtime usage ratios shown in Table 3.3. In Case I, we use Eq. (3.14) so that the value of CW_{min} is inversely proportional to

Station no.		STAs 1-2	STAs 3-4	STAs 5-6	STAs 7-8
Assigned weight		8	4	2	1
I	$W_0(=CW_{\min}+1)$	32	64	128	256
II	$W_0(=CW_{\min}+1)$	35	66	128	254
Station no.		**STAs 1-4**	**STAs 5-8**	**STAs 9-12**	**STAs 13-16**
III	$W_0(=CW_{\min}+1)$	32	64	128	256
IV	$W_0(=CW_{\min}+1)$	64	128	256	512
V	$W_0(=CW_{\min}+1)$	34	64	128	258

Table 3.3. The random backoff parameters for the airtime fairness.

a station's airtime usage ratio. The numerical result plotted in Figure (3.8) shows that this simple control cannot achieve the desired airtime allocation. STA1's or STA2's share of airtime is $\frac{8.94-8}{8} = 12\%$ more than the assigned ratio. Moreover, the largest overuse of airtime (by STA1 or STA2) is almost equal to the airtime received by the station with the smallest airtime usage ratio. Again, the error of Eq. (3.14) results from using $\frac{CW_{min}}{2}$ as the average value of random backoff time. The results can be substantially improved in Case II by using the algorithm in Section 3.3.4. The resultant ratio is almost equal to the assigned value with an error less than 1%. The largest overuse of transmission time by any station is less than 3% of the share of the smallest-ratio station, as compared to 94% in Case I. In Cases III–V, we consider 16 stations. The random backoff parameters used are also shown in Table 3.3. By using the parameters in Case IV, which double the CW_{min} values in Case III, the results can be improved because the number of collisions is reduced by using a larger contention window size. In this case, $\frac{CW_{min}}{2}$ well represents a station's average backoff time. However, there are still some discrepancies between the assigned and the actual ratios. In Case V, we use the parameters obtained from our Markovian model, and it achieves the best result under this scenario as shown in Figure 3.9. The comparison between the numerical and simulation results (using the parameters in Case III and V of Table 3.3) are presented in Table 3.4. The largest error is less than 2%, and it



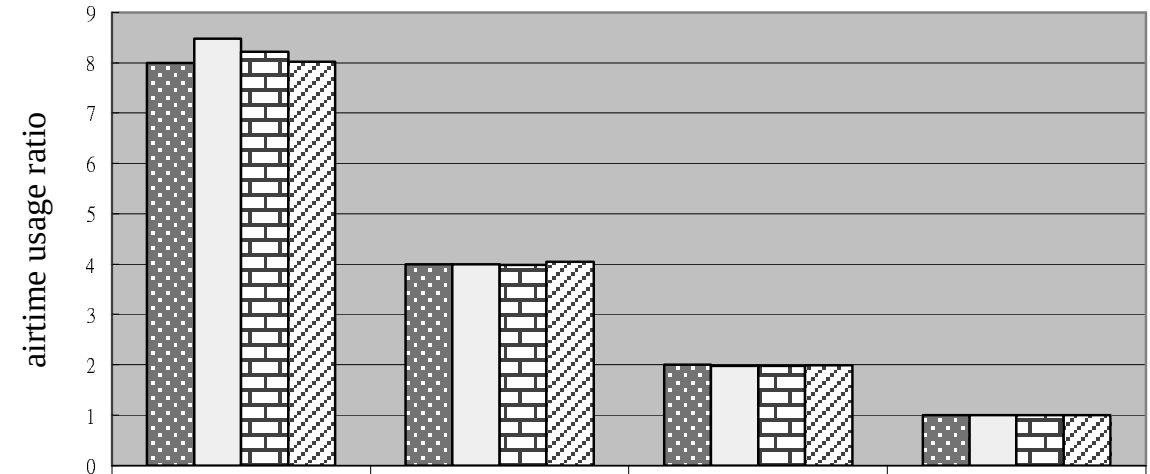
	STA 1-2	STA 3-4	STA 5-6	STA 7-8
assigned	8	4	2	1
Case I	8.9401	4.1556	2.014	1
Case II	8.0256	3.9973	2.0005	1

Figure 3.8. Comparison between basic and optimal controls: 8 stations

shows that the parameters determined by our model can accurately provide stations the share of airtime equal to their assigned ratios.

3.4.3 AIFS vs. CW_{min}

As shown in the previous subsections, controlling AIFS and CW_{min} values can both achieve the desired airtime allocation, and have their own advantages and disadvantages. For the control over AIFS, it only requires small differences among stations' AIFS values. Since stations do not rely on contention window sizes for differentiated airtime usage, they can use the same and smaller CW_{min}. The system airtime wasted due to stations' backoff can then be reduced compared to that of using proportional CW_{min} in Eq. (3.14). This way, it may improve the overall system throughput. in spite of its efficiency, the control over AIFS is sensitive to the number of stations in the wireless LAN. For example, if the number of stations with ratio ϕ_i changes, the required AIFS values of all stations may need to change according to Section 3.3.2 and the simulation results. Controlling CW_{min}, in contrast, is less affected by the changes in the number of stations. If we double the number of stations, using the



	STA 1-4	STA 5-8	STA 9-12	STA 13-16
assigned	8	4	2	1
Case III	8.4828	4.0041	1.9723	1
Casre IV	8.222	3.9826	1.9759	1
Case V	8.0151	4.0469	1.9908	1

Figure 3.9. Comparison between basic and optimal control: 16 stations

Assigned weight		**8**	**4**	**2**	**1**
8 stations	**Numerical results**	8.0256	3.9973	2.0005	1.0000
	Simulation results	8.088	4.004	1.979	1.002
		8.081	3.998	1.996	0.9983
16 stations	**Numerical results**	8.0151	4.0469	1.9908	1.0000
	Simulation results	7.954	3.950	1.998	1.009
		7.894	4.011	1.947	0.9985
		7.942	4.021	1.957	0.9831
		8.002	3.969	1.962	1.0086

Table 3.4. Comparison between analytical and simulation results: 8 and 16 stations

same set of CW_{min} values (e.g., 32, 64, 128, and 256 in Cases I and III of Table 3.3) can still approximate the desired airtime ratio. In fact, if larger values are used (e.g., CW_{min} values in Case IV — 64,128, 256 and 512), the airtime ratio will be closer to the desired values irrespective of whether there are 8 or 16 stations in the system. That is, the larger the CW_{min} values, the more insensitive our control will be to the changes in the number of stations. The disadvantage of using larger CW_{min} values is the waste of more system airtime due to stations' longer backoff times. When the number of stations is small, this may lead to the reduction of overall system throughput.

In general, we may need to change these parameters accordingly when a new station joins/leaves the wireless LAN. In the DCF mode of an infrastructure wireless LAN, the AP should take charge of computing the parameters for these stations and broadcast these parameters to stations via beacons. Therefore, the aforementioned sensitivity of the control over AIFS is not a big problem. This does not contradict our claim of distributed airtime control because transmission of individual frames still relies on stations' CSMA/CA with properly-chosen random backoff parameters. The AP need not schedule the transmission of individual frames for all stations as other centralized scheduling algorithms did. Instead, it only re-adjusts stations' parameters upon arrival or departure of stations. The "scheduling overhead" is far less than that of centralized algorithms. In fact, a new station is supposed to negotiate its share of system resources with the AP when it joins the wireless LAN. This (re)adjustment of parameters can then be included as a part of admission control. If the wireless LAN is in the ad hoc mode, we still can rely on CW_{min} for coarser airtime usage control according to Eq. (3.14). In this case, the insensitivity of controlling CW_{min} to the number of stations removes the need for AP.

3.4.4 Airtime Usage Control in Multi-rate IEEE 802.11 Wireless LANs

In the previous subsections, we assume that all stations use the same transmission rate and show how the stations' airtime usage can be controlled in a distributed manner. To investigate the impact of multi-rate support on stations' airtime usage and

show how the parameters should be adjusted, we consider 8 stations using different transmission rates. We assume that stations 1 and 2 use 11 Mbps, stations 3 to 5 use 5.5 Mbps, and stations 6 to 8 use 2 Mbps. For an illustrative purpose, we assume that they should use an equal amount of airtime. To achieve such airtime allocation, the number of times a station accesses the medium (i.e., n_i in Eq. (3.6)), should be inversely proportional to its transmission rate. For example, station 1 should access the medium twice more than station 3 does because it takes twice the airtime for station 3 to transmit a frame. Based on the ratio of n_i, we choose W_0 to be 35, 66, and 176 for stations 1 and 2, stations 3 to 5 and stations 6 to 8, respectively, according to Section 3.3.4. The values of CW_{max} are $2^5 \cdot W_0 - 1$, $2^4 \cdot W_0 - 1$, and $2^3 \cdot W_0 - 1$ for these three groups of stations. For a comparison purpose, we also let all stations use $W_0 = 32$ and $CW_{max} = 1023$ as in a regular IEEE 802.11 wireless LAN without airtime control. The station's airtime $T_i(0, t)$ is plotted in Figure 3.10 for both cases. Thanks to the optimal CW_{min} values, all stations can have an equal share of airtime regardless of their underlying transmission rates. In contrast, as explained in Section 3.2, stations receive the airtime inversely proportional to their transmission rates if there is no control over their airtime usage. The corresponding throughputs are listed in Table 3.5. If there is no control over stations' airtime usage, all stations will have an equal throughput but the system throughput is reduced. In this simulation, the system throughput with airtime control is 57% higher than that in a regular wireless LAN. Of course, the improvement depends on the stations' transmission rates and the assigned ratios, and may vary case by case. However, our control can yield a higher system throughput since lower-transmission rate stations will not "use up" all network resources.

If stations change their transmission rates, either the AIFS or CW_{min} values have to be changed in order to maintain the negotiated airtime usage. If a station lowers its transmission rate, it should then avoid using too much airtime. That is, it should reduce the frequency of accessing the wireless medium (i.e., smaller n_i in Eq. (3.6)). For example, if STA 1 lowers its transmission rate from 11 Mbps to 5 Mbps, it should access the medium 50% less frequently than before. As in the case of changes in the

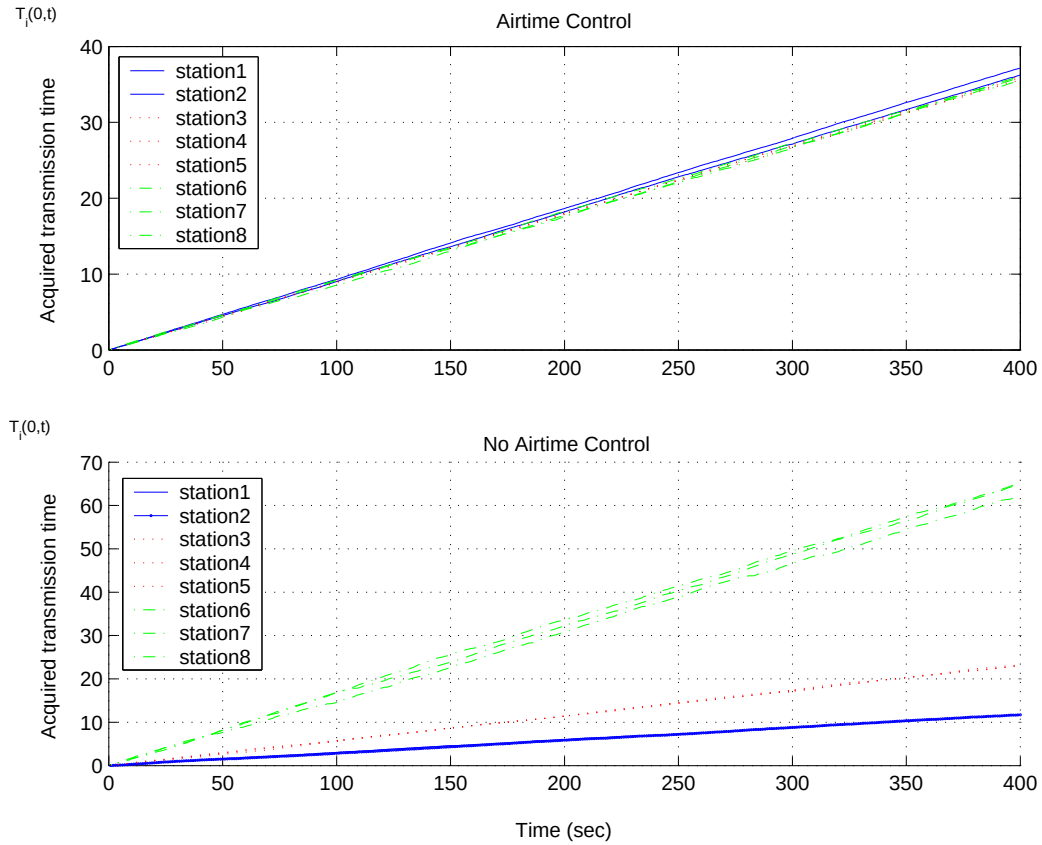


Figure 3.10. Station-received airtime with and without airtime control

	STAs 1-2	STAs 3-5	STAs 6-8	Total
Airtime control	0.988 1.013	0.494 0.490 0.497	0.177 0.179 0.181	4.021
No airtime control	0.322 0.320	0.317 0.321 0.319	0.326 0.310 0.325	2.56

Table 3.5. Throughput (Mbps) performance with and without airtime usage control in multi-rate IEEE 802.11 wireless LAN

number of stations, the AP should re-compute the optimal AIFS or CW_{min} values for such adjustment in the DCF mode of an infrastructure wireless LAN or the station adjusts its CW_{min} according to Eq. (3.14) in an ad hoc wireless LAN.

3.5 Conclusion

In this chapter, we proposed a distributed control on stations' airtime usage in the multi-rate IEEE 802.11 wireless LANs. Two different controls, one using the AIFS parameter and the other using CW_{min} parameter, were developed to achieve the desired airtime allocation in a distributed manner. Both the analysis and simulation results showed that we can finely control the stations' share of airtime by selecting the appropriate control parameters. With this airtime usage control, we can realize the (adaptive) QoS support without using the polling-based medium access method in the IEEE 802.11 wireless LAN.

CHAPTER 4

QoS Support Using the Distributed Medium Access in IEEE 802.11 Wireless LANs

As discussed in Chapter 3, the current IEEE 802.11 wireless LAN cannot provide any QoS support based on the DCF access method. To solve this problem, a new 802.11 standard — the 802.11e standard — has been developed to enable the QoS provisioning in the IEEE 802.11 wireless LANs. The new 802.11e standard uses a new medium access method, called the *Hybrid Coordination Function* (HCF). The word "hybrid" comes from the fact that it combines a contention-based access method, referred to as *Enhanced Distributed Channel Access* (EDCA), and a polling-based access method, referred to as *HCF Controlled Channel Access* (HCCA). The EDCA is essentially the DCF except that the EDCA allows stations to use different CSMA/CA parameters. The HCCA is essentially the PCF with additional signaling mechanisms for QoS negotiation. These two medium access methods are designed to provide two distinct levels of QoS: prioritized and parameterized QoS. The prioritized QoS only requires a station to transmit the data frames based on their assigned priorities. Therefore, the prioritized QoS can be achieved by the contention-based EDCA. However, the parameterized QoS requires a station to transmit the data frames with certain QoS guarantees. Therefore, the parameterized QoS can only be achieved by the polling-based HCCA.

Although the EDCA is designed to provide the prioritized QoS only, it is actually capable of providing the parameterized QoS *if the airtime usage control problem can be solved*. In this chapter, we show that by adding the distributed airtime usage control in Chapter 3 to the current EDCA, we are able to provide the same level of parameterized QoS as the HCCA does. However, unlike the HCCA, this enhanced

EDCA does not require any polling master to schedule the stations' transmission, hence making it very attractive for QoS support in ad hoc IEEE 802.11 wireless LAN.

This chapter is organized as follows. In Section 4.1, we briefly describe the IEEE 802.11e MAC protocol and its new designs for QoS support. Section 4.2 presents the medium time allocation for QoS provisioning in a time-division, multi-rate wireless network. In Section 4.3, we describe how to use the airtime usage control to provide parameterized QoS in the EDCA. Section 4.4 describes the signaling for QoS provisioning in both infrastructure and ad hoc IEEE 802.11 wireless LANs. The performances of the EDCA and the HCCA, in terms of their support for parameterized QoS are discussed in Section 4.5. Conclusions are drawn in Section 4.6.

4.1 Overview of The IEEE 802.11e MAC Protocol

As mentioned in the introduction, the IEEE 802.11e standard defines two medium access methods, namely the EDCA and the HCCA. In general, the EDCA is an enhanced version of DCF and is designed to provide the prioritized QoS. The HCCA is an enhanced version of the PCF and is designed to provide the parameterized QoS. In what follows, we focus our discussion of the IEEE 802.11e standard on these QoS-related enhancements in the EDCA and the HCCA.

4.1.1 Enhanced Distributed Channel Access (EDCA)

The EDCA provides distributed and differentiated access to the wireless medium for 8 user priorities. In order to do so, the EDCA defines access categories (ACs) that provide support for the delivery of traffic with user priorities at wireless stations. Each AC is in fact an enhanced variant of the IEEE 802.11 DCF, which uses CSMA/CA with random backoff to access the wireless medium. The most significant difference between the EDCA and the DCF is that the ACs in the EDCA use different CSMA/CA parameters (i.e., minimum/maximum contention window size, inter-frame space (IFS)) to acquire prioritized access to the wireless medium, while the stations in the DCF use the same CSMA/CA parameter to access wireless medium. In the

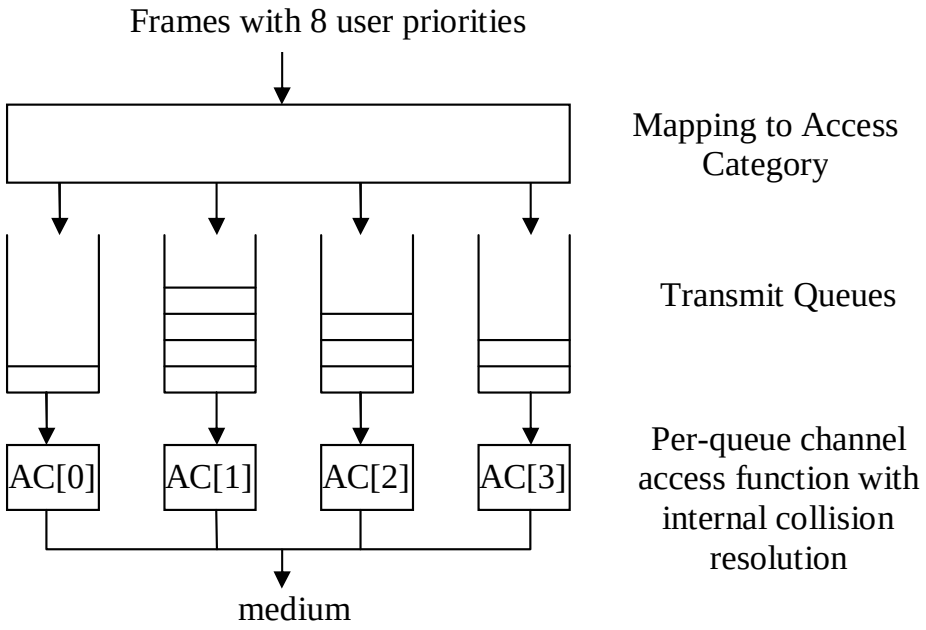


Figure 4.1. Access categories with internal collision resolution in the EDCA

current IEEE 802.11e standard, each station should support four ACs to provide prioritized frame delivery for up to 8 different user priorities as shown in Figure 4.1.

Since each AC is a medium access function as the DCF, it is possible that two ACs in the same station may collide with each other. Such a collision is referred to as an internal collision in the IEEE 802.11 e standard. The internal collision is resolved within the station such that the AC with higher priority receives the access to the medium, and the AC with lower priority behaves as there were an external collision on the wireless medium. The only exception is that the retry count for the frame being transmitted by the lower priority AC is not incremented. Therefore, the data frames will not be discarded due to the internal collisions.

Another difference between the EDCA and the DCF is that during each possession of the wireless medium, the wireless station (i.e., an AC) may initiate multiple frame exchange sequences, separated by a short inter-frame space (SIFS), to transmit data frames within the same AC. However, the total duration of the frame exchange sequences must not exceed a predefined limit called Transmission Opportunity (TXOP) limit. Compared to the DCF in which there is no control on station's usage of the medium time during each possession of the medium, the design of TXOP limitation

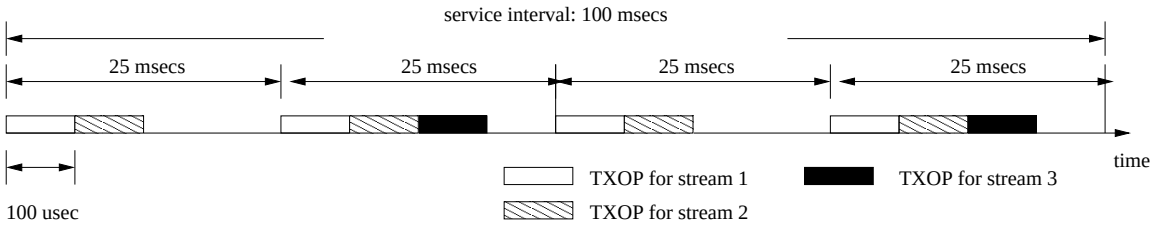


Figure 4.2. Service schedule in the HCCA: the required TXOPs are calculated by the HC and then allocated to streams via polling.

makes it possible to control station's usage of the medium time in the EDCA. We will show later that by controlling the value of TXOP limit, we can also achieve a distributed airtime usage control in the IEEE 802.11 wireless LANs.

4.1.2 HCF-Controlled Channel Access (HCCA)

The HCCA uses a QoS-aware centralized coordinator, called the hybrid coordinator (HC), as a polling master to allocate the medium time (i.e., the TXOP) to itself and other stations. Because of this polling-based mechanism, stations can easily obtain their required medium time as compared to that under the EDCA. What the HC needs to compute are the polling orders and the amount of TXOPs granted to a station for each poll (together called a "service schedule" in the 802.11e standard). Based on the service schedule, the HC polls each station to initiate frame exchange sequences. To give an example of how a service schedule is computed, let us consider 3 multimedia streams that generate packets with size of 600 bytes every 25, 25, and 50 msecs with delay bounds of 100, 100, and 200 msecs, respectively. For the illustrative purpose, we do not consider any polling frames or control overhead, and we assume all streams are transmitted at 48Mbps. To meet the delay bound guarantee, one can choose the polling period (so-called "service interval" in the 802.11e standard) as the minimum of all streams' delay bounds. In this example, we have a service interval of 100=min$(100, 100, 200)$ msecs. Within this interval, the first two streams need $\frac{100}{25} * 600 * 8/48 * 10^6 = 400$ μ secs to transmit four data frames while the last stream only needs 200 μsecs to transmit two frames. One possible implementation of the service schedule in this example is illustrated in Figure 4.2.

Although the HCCA is recommended for parameterized QoS in the IEEE 802.11

wireless LANs primarily because of its efficiency, it is inflexible in the sense that the HC may need to recompute the service schedule every time when a station adds new traffic stream to the wireless LAN, an existing traffic stream leaves the wireless LANs, or a station changes the physical transmission rate. Besides, when two HCCA-coordinated wireless LANs operate on the same medium in the overlapping space, it requires additional coordination between the HCs to avoid any time confliction on their service schedules. More importantly, the HCCA-supported parameterized QoS cannot be realized in the ad hoc IEEE 802.11 wireless LAN.

4.2 Medium Time Allocation For Parameterized QoS

The most important task to achieve the parameterized QoS in the IEEE 802.11 wireless is to ensure that the stations receive their required TXOP. The amount of TXOP needed by a station depends on the QoS requirements of the streams in that station. In the IEEE 802.11e standard, the station specifies these requirements via a so-called traffic specification (TSPEC). The TSPEC element represents a stream's general expectation for the QoS and thus, plays an important role in determining stations' TXOP. In what follows, we give an overview of some important fields in the TSPEC element. Based on the TSPEC, we derive a guaranteed rate that along with the station's physical transmission rate, determines the station's TXOP.

4.2.1 Overview of the TSPEC Element

The TSPEC element contains the set of parameters that characterize the traffic stream that the station wishes to establish. There are 6 important fields in the TSPEC that can be taken into account to determine the required TXOP:

- The *Mean Data Rate (ρ)* field specifies the average data rate of a traffic stream, in bits per second, for transport of MAC service data units (MSDUs) belonging to this stream.
- The *Peak Data Rate (P)* field specifies the maximum allowable data rate in bits per second, for transfer of the MSDUs belonging to a traffic stream.
- The *Maximum Burst Size (σ)* field specifies the maximum data burst in bits that

arrive at the MAC service access point (SAP) at the *peak data rate* for transport of MSDUs belonging to a traffic stream. *This definition is different from the conventional definition for burst size defined in the Resource Reservation Setup Protocol (RSVP) and other protocols where burst may arrive at an infinite rate.*

- The *Minimum Physical (PHY(TX Rate (R)* field specifies the minimum physical transmission rate, in bits per second, required to be operated by the station or the AP in order to guarantee the QoS. As we will show later, this parameter prevents stations from overusing the system medium time via the link adaptation in the multi-rate IEEE 802.11 wireless as we explained in Chapter 3.
- The *Delay Bound (d)* field specifies the maximum amount of time in units of microseconds allowed to transport an MSDU belonging to a traffic stream, measured between the arrival of the MSDU at the local MAC layer and the start of successful transmission or retransmission of the MSDU.
- *MSDU Size (L)* field specifies the size of the frame in a traffic stream. The maximum value of L is fixed in the standard at 2304 bytes.

The Mean Data Rate, Peak Data Rate, and Delay Bound fields in a TSPEC represent the QoS expectations of a stream, and can be used to determine the TXOP in many different ways. For example, the station may request the Peak Data Rate for a stream to provide the best QoS, or just request the Mean Data Rate for the least QoS support. Obviously, these two methods require different amounts of TXOP: the former requires a much larger amount of TXOP and the wireless LAN ends up with admitting fewer streams, while the latter requires a smaller amount of TXOP but barely supports QoS for bursty streams. In order to alleviate the tradeoff between system efficiency and QoS performance, we derive a so-called *guaranteed rate* based on the stream's TSPEC parameters and the dual-token bucket traffic regulation. This guaranteed rate is the minimum data rate at which all frames can be transmitted within the specified delay bound. Obviously, the guaranteed rate is larger than the Mean Data Rate but less the Peak Data Rate.

Figure 4.3 shows the dual-token bucket filter that is associated with each stream and is situated at the entrance of the MAC buffer. In order to ensure that the actual

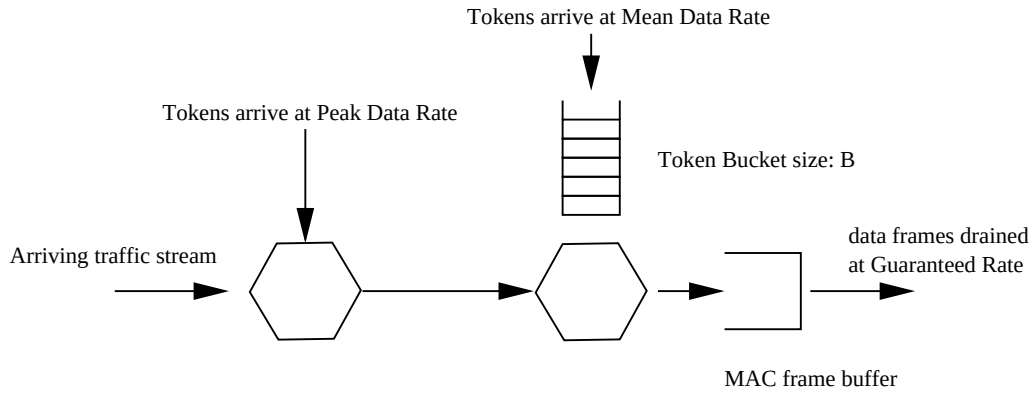


Figure 4.3. The dual-token bucket filter for traffic policing.

arriving frames of the corresponding stream comply with the TSPEC, the bucket size is set as $B = \sigma \cdot (1 - \rho/P)$. One can easily have the arrival process of a stream passing through the dual-token bucket filter constrained by

$$A(t, t+\tau) = Min(P\tau, B + \rho\tau), \tag{4.1}$$

where $A(t, t+\tau)$ is the cumulative number of arrivals during $(t,\ t+\tau)$. From Eq. (4.1) we can construct the arrival rate curve which is drawn in Figure 4.4. Since the guaranteed rate has to be less than the peak rate but large enough to satisfy a stream's delay bound, the relation between the guaranteed rate (g) and the delay bound (d) can be found as illustrated in Figure 4.4. Using the distance formula, one can easily derive the guaranteed rate g_i for stream i

$$g_i = \frac{\sigma_i}{d_i + \frac{\sigma_i}{P_i}}, \tag{4.2}$$

where σ_i, d_i and P_i are the maximum burst size, delay bound and peak data rate of stream i.

Since transmissions on the wireless medium are prone to errors, one may want to provide a larger guaranteed rate to compensate the stream for the failed transmission. By taking into account the error probability of stream i, $P_{e,i}$, we can obtain the new guaranteed rate as

$$g_i = \frac{\sigma_i}{(d_i + \frac{\sigma_i}{P_i})(1 - P_{e,i})}. \tag{4.3}$$

How to estimate $P_{e,i}$ is beyond the scope of this chapter. One simply way is to use the RSSI value from a received data or acknowledgement frame to estimate the error

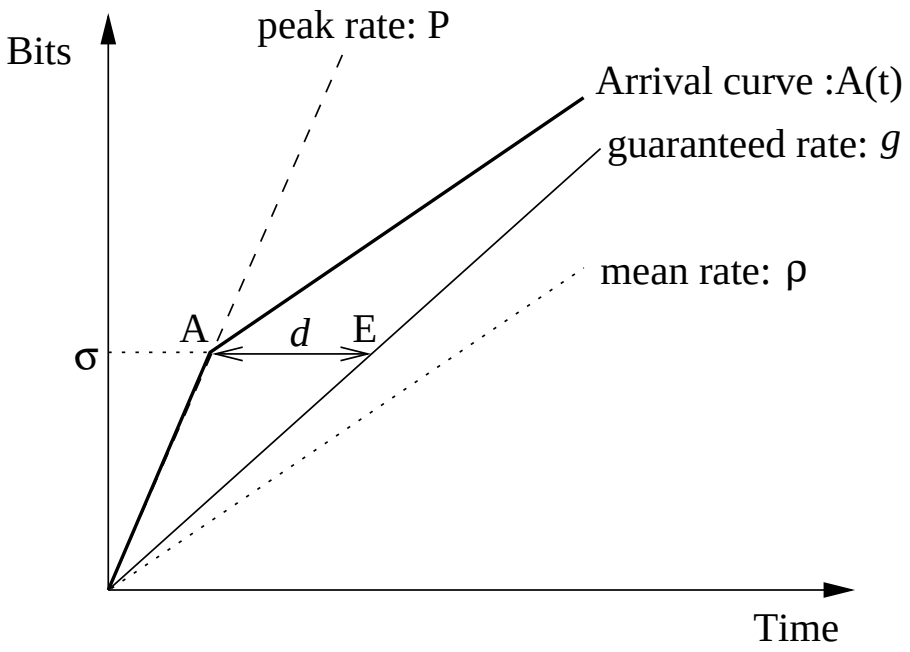


Figure 4.4. Arrival curve at the entrance of MAC buffer and the guaranteed rate for a traffic stream.

probability.

4.2.2 Admission Control Algorithm

With the guarantee rate derived from the TSPEC, the amount of TXOP required by station i for its stream j can be computed by

$$TXOP_{i,j} = \frac{g_{i,j}}{R_i}, \tag{4.4}$$

where $g_{i,j}$ is the guaranteed rate for stream j in station i and R_i is the station i's PHY transmission rate. Here, the $TXOP_{i,j}$ is the amount of medium time station i should obtain for stream j, in an one-second time interval, to guarantee the stream's delay bound. Obviously, $TXOP_{i,j}$ must be less than 1. In other words, the wireless station can only guarantee the stream's delay requirement if and only if it always maintains its PHY transmission rate higher than the guaranteed rate. In fact, the station has to keep its PHY transmission rate higher than a rate determined by the amount of medium time (i.e., the airtime) with which the station is allowed to use for the traffic stream.

Let us consider an HDTV stream in an IEEE 802.11 wireless LAN using 802.11a

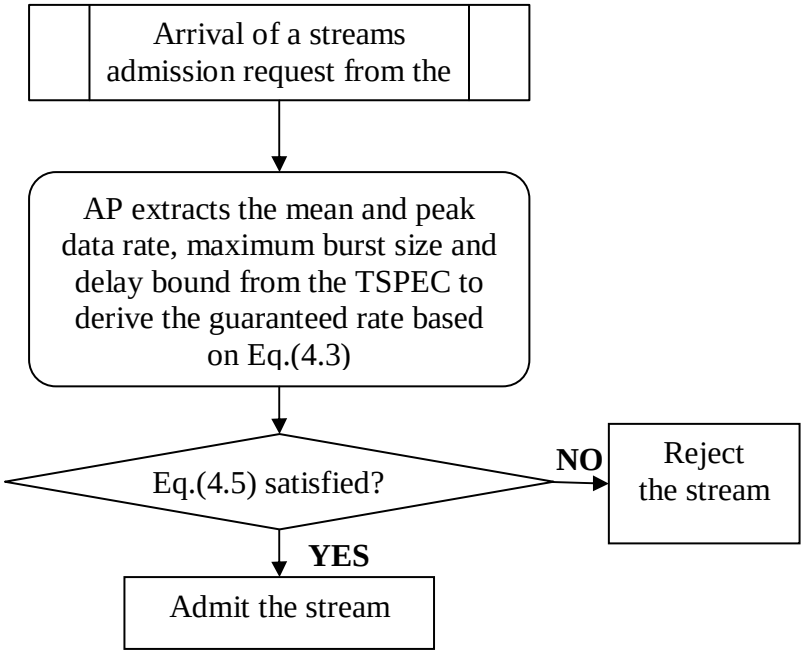


Figure 4.5. Airtime-based admission control algorithm for both the EDCA and HCCA.

PHY layer. If the guaranteed rate for the HDTV stream including the overheads is 30 Mbps, the station may set the minimum PHY rate as 48 Mbps, meaning that the station will occupy 62.5%(= 30/48) of the medium time for this HDTV stream. The station may also set the minimum PHY rate at 36 Mbps, meaning that 83% of the medium is used by that HDTV stream. The more airtime a stream gets, the lower the PHY rate (or a larger range of the PHY rates) a wireless station is allowed to use in order to still satisfy the stream's QoS requirement. However, the wireless LAN may end up with admitting very few traffic streams if the station decides to provide its stream such "wide-range" (in terms of the PHY rates) QoS guarantees. Such a trade-off between QoS guarantees and system utilization, due to the link adaptation, has to be considered when handling the admission control problem in the multi-rate IEEE 802.11 wireless LAN.

Based on Eq. (4.4), we can also obtain the admission control for the parameterized QoS in the IEEE 802.11e wireless LAN as

$$r_i + \sum_{k=1}^{i-1} r_k \leq EA, \tag{4.5}$$

where $r_i = \frac{g_i}{R_i}$ is the fraction of system medium time stream i should obtain and

EA is the fraction of the system medium that can be used for transmitting data frames. Ideally, the value of EA is 1 but the actual value of EA is always less than 1 because of the control overhead incurred by the resource allocation mechanisms. One can expect that using the HCCA can achieve a higher EA than the EDCA because of inevitable collisions due to the contention in the EDCA. The flow chart for QoS negotiation and admission control algorithm is depicted in Figure 4.5.

4.3 Allocation of Airtime in IEEE 802.11e Wireless LANs

The admission control given in Eq. (4.5) requires an effective airtime allocation mechanism to ensure that each station acquires its share of airtime, r_i. Since the HCCA relies on a polling-based mechanism, it can easily allocate the required amount of airtime to wireless stations. As in the example of Section 4.1.2, what the HC needs to do is to calculate the Service Interval (SI) as:

$$SI = \frac{1}{2}\min\{d_1, d_2, \ldots, d_{k+1}\}, \tag{4.6}$$

where d_i is stream i's delay bound. To calculate the required amount of TXOPs for stream i, we need to determine the number of frames that have to be drained from this stream at the guaranteed rate. The number of frames N_i is given by

$$N_i = \left\lceil \frac{SI \times g_i}{L_i} \right\rceil, \tag{4.7}$$

where L_i is stream i's frame size. Then, the TXOP for this stream is obtained as

$$TXOP_i = max\left(\frac{N_i L_i}{R_i}, \frac{M}{R_i}\right) + O, \tag{4.8}$$

where R_i is the negotiated minimum PHY rate for stream i, M is the maximum frame size, and O is the overhead in time units, including the inter-frame spaces, acknowledgement frame and polling overheads. Due to space limitation, details for the overhead calculations are omitted here.

Unlike the polling-based HCCA, the EDCA relies on a distributed, contention-based mechanism. To realize parameterized QoS, we need each wireless station (or its ACs) to use adequate EDCA parameters. In what follows, we focus on how to

determine the EDCA parameters for stations based on the airtime ratio r_i in the admission control. Then, we will compare the HCCA and EDCA from the perspectives of QoS provisioning and system complexity.

4.3.1 Airtime Usage Control in the EDCA

There are two methods to control each station's airtime usage in the EDCA: (1) controlling the TXOP limit of each station and (2) controlling the medium accessing rate of individual stations as described in Chapter 3. By using the first method, all stations choose the same EDCA parameters (as in the DCF) but each station can occupy the wireless medium for a different amount of time during each access. By using the second method, each station occupies the medium for the same amount of time during each access but has a different medium "accessing frequency".

Controlling the TXOP Limit

Let r_i' be the fraction of airtime that station i should obtain and $TXOP_i$ be the value of station i's TXOP limit. Let T_i be the amount of time required to transmit a frame with size of L_i (excluding the frame header) from stream i at the negotiated minimum PHY rate R_i. T_i is obtained by

$$T_i = \frac{L_i}{R_i}. \tag{4.9}$$

Let M be the index of the stream such that $T_M = \max_i T_i$. Then, one an choose $TXOP_i$ as

$$TXOP_i = \frac{r_i T_M}{r_M T_i} \frac{L_i + H}{R_i} + (2 \left\lceil \frac{r_i T_M}{r_M T_i} \right\rceil - 1) SIFS + \left\lceil \frac{r_i T_M}{r_M T_i} \right\rceil T_{ack} \tag{4.10}$$

where H is the MAC frame header size and T_{ack} is the amount of time to transmit an acknowledgement frame. For example, consider four streams with $L_i = 600$, 600, 1200 and 1200 bytes, respectively. We assume these four streams are required to transmit at least at the PHY rates of 48, 48, 48 and 24Mbps, respectively. Based on Eq. (4.9), we have $T_M = 1200 * 8/24 * 10^{-6} = 400$ μsecs. If we assume r_i for each stream to be 0.1, 0.2, 0.2, and 0.1, respectively, we have $N_i = \frac{r_i T_M}{r_M T_i} = 4$, 8, 4, and 1, and N_i is actually the number of data frames that stream i should transmit during each access

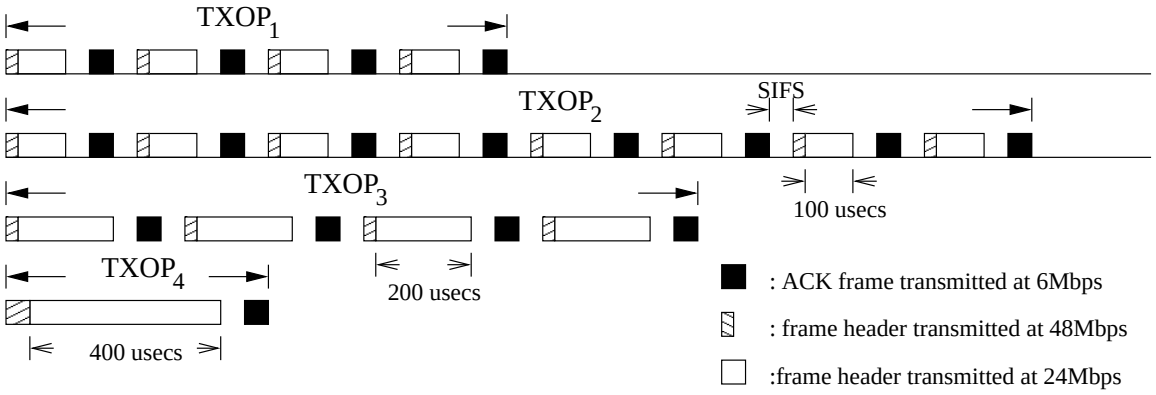


Figure 4.6. Example 1 — Selection of TXOP limits: given that $SIFS$=16 μsecs, frame header size =34 bytes, and ACK frame size = 14 bytes in the IEEE 802.11a standard, we have $TXOP_1$=619.6 μsecs, $TXOP_2$=1255.2 μsecs, $TXOP_3$=1019.6 μsecs, and $TXOP_4$= 512.5 μsecs. *Physical layer overhead is not included in the computation.

to the wireless medium. The values of $TXOP_i$ are illustrated in Figure 4.6. In the case when N_i is not an integer number, frame fragmentation is required for precise airtime control.

With the values of $TXOP_i$ chosen by Eq. (4.10) and the fact that each station has a statistically equal probability to access the medium (because of using the same EDCA parameters), each station will obtain the amount of airtime proportional to its r'_i value. The maximum amount of airtime station i can get within an one-second period $r_{max,i}$ is

$$r_{max,i} = \frac{r_i}{\sum_i r_i} EA \geq \frac{r_i}{\sum_i r_i} \sum_i r_i \geq r_i, \tag{4.11}$$

given that Eq. (4.5) is held true. Eq. (4.11) shows that each station can always obtain the required amount of airtime by using this simple control method. *In fact, one of the greatest advantages of using the EDCA is that the amount of airtime a station can get is determined by the ratio of stations' r_i values, not the absolute value of r_i.* For example, assume that station 1 need 0.1 sec out of every one-second period (i.e., $r_1 = 0.1$) for a stream and station 2 need 0.2 sec (i.e., $r_1 = 0.2$) for another stream. Based on Eq. (4.11) and given that $EA = 0.6$, the actual amount of airtime station 1 can obtain is 0.2 sec and that for station 2 is 0.4. When more streams join the wireless LAN, the amount of airtime station 1 can get decreases (automatically adjusted by the EDCA via Eq. 4.11) but it will not get less than 0.1 according to Eq. (4.5).

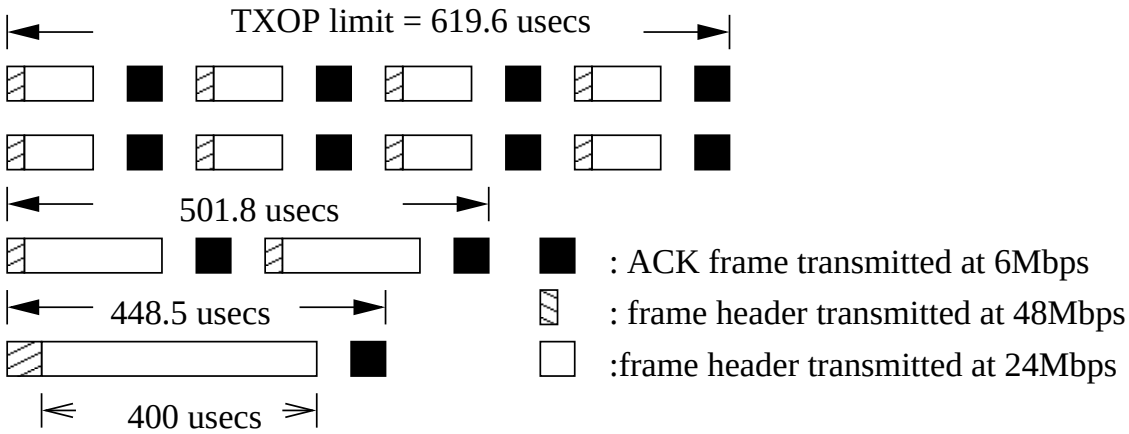


Figure 4.7. Example 2 — Selection of the network-wide unified TXOP limit. In this example, the TXOP limit for all stations is 619.6 μsecs.

Controlling the Medium Accessing Rate

Instead of controlling the duration of a TXOP, we can use a fixed TXOP duration for all stations but control their access rate, AR_i, so that stations can still acquire the desired amount of airtime. This TXOP has to be chosen so that each station uses the same amount of airtime — during each access to the wireless medium — to transmit data frame at the negotiated minimum PHY rate. Therefore, the TXOP limit is chosen as

$$\text{TXOP limit} = \max_i \left\{ \left\lceil \frac{T_M}{T_i} \right\rceil \frac{L_i + H}{R_i} + (2 \left\lceil \frac{T_M}{T_i} \right\rceil - 1) SIFS + \left\lceil \frac{T_M}{T_i} \right\rceil T_{ack} \right\}. \quad (4.12)$$

As shown in Figure 4.7, the TXOP limit of the above example is 619.6 μsecs and all four streams will transmit 400 μsec-worth data frames given this TXOP limit (i.e., streams 1 and 2 send 4 frames, stream 3 sends 2 frames and stream 4 sends one frame).

Several EDCA parameters can be used for controlling AR_i, including the minimum and maximum contention window sizes ($CW_{min,i}/CW_{max,i}$) and arbitration inter-frame space ($AIFS_i$). The relation between these parameters and the accessing rate can be found in Chapter 3 as

$$\sum_{i=1}^{n_1} BT_i^{(1)} = \sum_{j=1}^{n_2} BT_j^{(2)} + \sum_{h=1}^{n_1+n_2-1} D_h, \quad (4.13)$$

where $BT_i^{(j)}$ is the i-th backoff time chosen by STA j and is mainly determined by $CW_{min,j}$ and $CW_{max,j}$, D_h is decrementing lag and is mainly decided by $AIFS_i$ value, and n_i represents the total number of times STA i has backed off during the observing time interval and is proportional to AR_i. Based on Eq. (3.6) and by setting

$$\frac{AR_i}{AR_j} = \frac{r_i}{r_j} = \frac{n_i}{n_j}, \tag{4.14}$$

we can determine the adequate EDCA parameters using the algorithms given in Chapter 3. One approximate but very simple solution is to choose CW_{min} as

$$\frac{CW_{min,i}}{CW_{min,j}} = \frac{r_j}{r_i}, \tag{4.15}$$

which will give a very good control on AR_i. On can easily reach the same conclusion drawn from Eq. (4.11) that stations can always acquire at least the required amount of airtime in a distributed manner.

4.3.2 Comparison of the EDCA and the HCCA

The greatest advantage of using the HCCA for QoS guarantees is higher system efficiency (i.e., a higher EA value), thanks to the HCCA's contention-free nature. Due to this higher efficiency, the HCCA can provide more resource and may admit more traffic streams than the EDCA. Moreover, the HCCA has better control over stations' usage of airtime than the EDCA in which stations have to "cooperate" with each other for airtime usage control. However, there are several potential problems of using the HCCA primarily due to its centralized control over stations' access to the wireless medium.

1. As pointed out in the IEEE 802.11 standard, the operation of the polling-based channel access may require additional coordination to permit efficient operation in cases where multiple polling-based wireless LANs are operating on the same channel in an overlapping physical space. New standard supplements such as the IEEE 802.11k standard are being developed to facilitate the required coordination, but will increase system complexity. On the other hand, the EDCA does not need any coordination between wireless LANs using the same

channel because the EDCA is intrinsically designed to solve the channel sharing problem.

2. The HC in the HCCA needs to recompute the service schedule whenever a new traffic stream joins or an existing stream leaves the wireless LANs. Such re-computation of service schedules may occur very frequently and need coordination as mentioned above when two HCs operate on the same channel in an overlapping physical space. However, the stations in the EDCA assigns the appropriate EDCA parameters set to the new stream and the existing streams may not need to make any adjustment.[1]
3. As mention earlier, the QoS of a traffic stream can only be guaranteed if the wireless station transmits at a (physical) rate higher than the negotiated minimum physical rate. If a station lowers its physical transmission rate (below the negotiated rate), the amount of airtime originally allocated to the stream (by the HC) may not suffice to support the required QoS even though the HC may still have enough unallocated resource to support that stream's QoS at this lower rate. Of course, the HC can temporarily allocate more airtime (by recomputing the service schedule) to support that stream's QoS at this lower rate. However, if more new streams request for QoS later, the HC needs to cut the stream's airtime allocation back to the originally-negotiated amount since the HC needs airtime for new streams. However, using the EDCA will not require the AP to reallocate airtime because wireless stations can automatically obtain the extra amount of airtime according to Eq. (4.11). Consider the previous example again. Stations 1 and 2 can actually halve their PHY rates and still meet the QoS requirements. In other words, *the QoS can be automatically provided by the EDCA, regardless of the rate at which a station is using, as long as the system airtime resource allows. The new streams will not have problems to get the required amount of airtime as the airtime allocation is adjusted automatically according to* Eq. (4.11).

[1]It depends on which airtime control methods of the EDCA is applied

4.4 QoS Signaling for Admission Control and Parameter Negotiation

The IEEE 802.11e standard has specified a set of signaling procedures for adding new QoS streams into an HC-coordinated wireless LAN. We can use these procedures, with little modification, for QoS signaling in the EDCA. In order to better understand how these procedure is implemented in the IEEE 802.11e standard, we briefly introduce the architecture and layer management in the IEEE 802.11e standard.

4.4.1 Architecture and Layer Management of the IEEE 802.11e Standard

Both the MAC sublayer and PHY in the 802.11 standard conceptually include management entities, called MLME (MAC Layer Management Entity) and PLME (Physical Layer Management Entity), respectively. These entities provide the layer management service interfaces through which layer management functions may be invoked. In order to provide correct MAC operation, a station management entity (SME) will be present within each station. The SME is a layer-independent entity that may be viewed as residing in a separate management plane. The SME is responsible for gathering layer-dependent status from the various layer management entities (LMEs), and similarly setting the value of layer-specific parameters. The SME would perform functions on behalf of general system management entities and would implement standard management protocols. Figure 4.8 shows the relationship among management entities. With the overall picture of 802.11e layer management, we can now explain the QoS signaling procedures.

4.4.2 QoS Signaling for Setting up a Stream

Figure 4.10 shows the sequence of messages exchanged during a traffic stream (TS) setup. The SME at the wireless station creates a TS based on the request from the higher layer.[2] The SME also obtains the TSPEC parameters from the higher layer. The SME generates an MLME-ADDTS.request containing the TSPEC. The station's

[2]The decision to create the TS and how to generate the TSPEC parameters are out of scope in the standard.

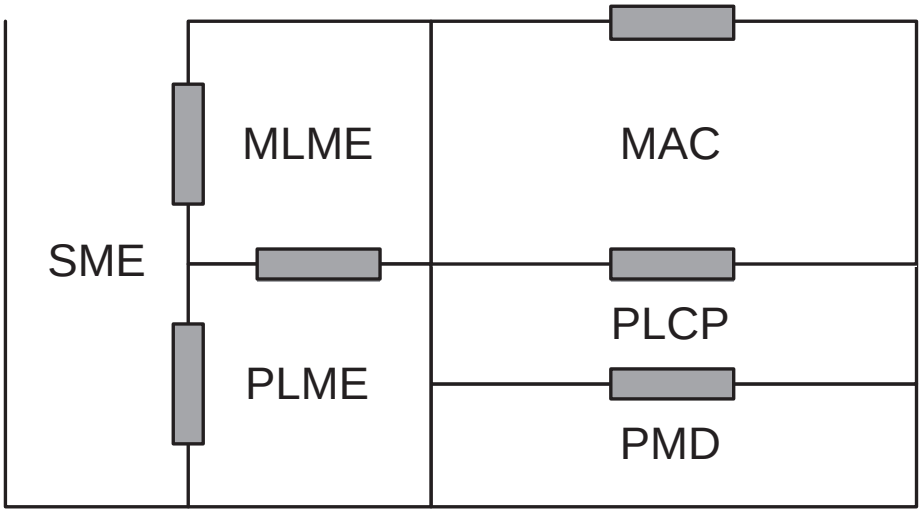


Figure 4.8. Architecture and layer management of IEEE 802.11e standard — SME: Station Management Entity, MLME: MAC Layer Management Entity, PLME: Physical Layer Management Entity, PLCP: Physical Layer Convergence Protocol, PMD: Physical Medium Dependent.

1 octets	1	1	4			1
Element ID	Length	QoS Info	Stream Parameters			Reserved
			AIFS	CWmin	TXOP	

Figure 4.9. The modified EDCA parameter set element for supporting parameterized QoS in the EDCA.

MAC transmits the TSPEC in an ADDTS request in the corresponding QoS Action frame or the (re)association request frame to the HC and starts a response timer called ADDTS timer of duration *dot11ADDTSResponseTimout*. The HC MAC receives this management frame and generates an MLME-ADDTS.indication primitive to its SME containing the TSPEC. The SME in the HC decides whether to admit the TSPEC as specified, or refuse the TSPEC, or not admit but suggest an alternative TSPEC and generates an MLME-ADDTS.response primitive containing the TSPEC and a ResultCode value by employing the admission control algorithm. The HC MAC transmits an ADDTS response in the corresponding QoS Action frame or (re) association response containing this TSPEC and status.

Although the signaling is designed for the HCCA to support parameterized QoS, we can use the same procedures for adding new QoS streams into a wireless LAN using the EDCA. Here, the HC is replaced by the AP since there is no HC in an

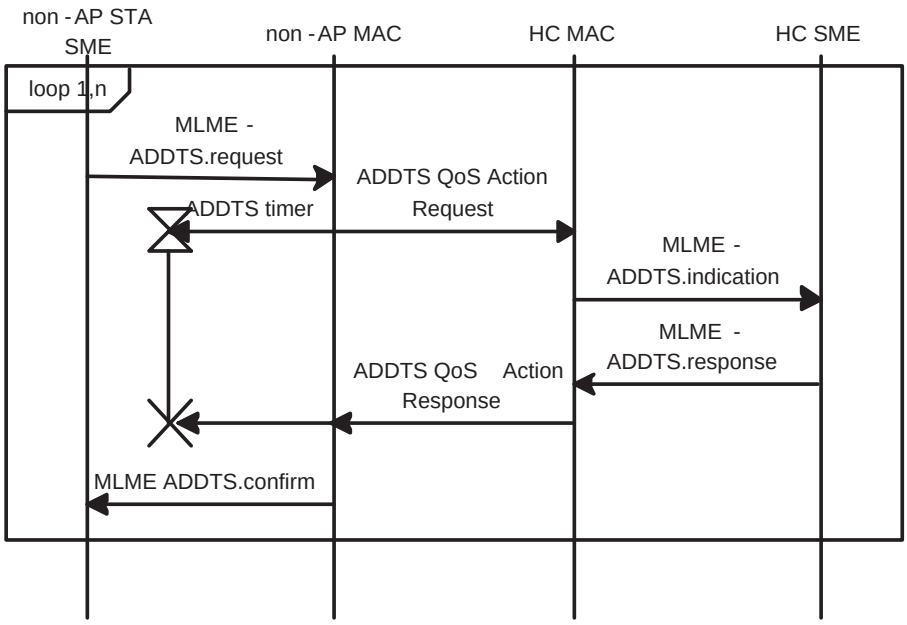


Figure 4.10. Signaling and message exchanges of adding a QoS traffic stream to an HC-coordinated 802.11 wireless LAN.

EDCA-based wireless LAN. The most important task here is to transport the EDCA parameters to the station requesting for parameterized QoS. Fortunately, we can convey these parameters via the *EDCA Parameter Set element* in the frame body of the MAC management frame.[3] We modify the EDCA parameter set element of 802.11e standard as shown in Figure 4.4.2 so that the AP can signal the decision of admission and corresponding EDCA parameters to the station.

If a wireless LAN operates at the ad hoc mode, there will be no AP for admission control and definitely no HC to allocate TXOPs to stations. In this case, stations can only use distributed admission control and the enhanced EDCA for parameterized QoS. Next, we outline how this can possibly be achieved in an ad hoc mode of 802.11e wireless LAN.

4.4.3 Admission Control in the Ad Hoc Mode

For the admission-control purpose, each station has to monitor the channel and determine the current channel utilization. In this chapter we do not consider the hidden

[3]QoS Action frames are a MAC management frame.

terminal effects and assume that all stations hear each other and are not in the power saving mode. Otherwise, the QoS provisioning is almost impossible. Once the channel utilization is determined, each arriving stream's TSPEC element when received at the SME, is passed onto the MAC for determining the guaranteed rate. Note that the signaling is similar to the one discussed earlier with the exception that there is no ADDTS frame that is sent physically on the medium.

Based on the guaranteed rate and the minimum PHY rate, the station can determine the value of r_i. If r_i is found to satisfy Eq. (4.5), the station transmits a RTS frame with the value of r_i to the destination station. Once the destination station responds to the RTS frame with a CTS frame, all stations assume that the new stream's QoS request has been admitted and hence update the system utilization (i.e., $\sum_i r_i$ in Eq. (4.5))for later use. The station requesting admission then contends for the wireless medium with the enhanced EDCA parameters as explained before. In general, this admission control algorithm is similar to that for parameterized QoS in the EDCA, with the exception that the admission control is realized in a distributed manner. Because of this distributed nature and the fact that the minimum PHY transmission rates are determined by individual stations, some stations may over-occupy the wireless medium if they allow the streams to be transmitted at very low PHY transmission rates (and thus, a large r_i). Therefore, it is each individual station's responsibility to use the wireless medium "responsibly".

4.5 Evaluation

In this section, we compare the polling-based HCCA and the contention-based EDCA for their QoS support via simulations. We will focus on the performance of using the enhanced EDCA for QoS support and verify the effectiveness of the integrated airtime-based admission control and enhanced EDCA. The simulations are carried out in OPNET for four scenarios. In scenario 1, we compare the system efficiency, in terms of the number of streams being admitted into a wireless LAN under the EDCA and the HCCA. In scenario 2, we compare the two controlling methods in the enhanced EDCA, namely, controlling TXOP limit and controlling medium accessing

frequency. In scenarios 3 and 4, we compare the performance of the HCCA and the EDCA when some stations vary their physical transmission rates under the heavy- and light-load cases, respectively. We have modified the wireless LAN MAC of OPNET to include the admission control algorithm and the signaling procedures as explained above.

4.5.1 Scenario 1: System Efficiency

We assume that each station carries a single traffic stream which requests a guaranteed rate of 5 Mbps.[4] We also assume that all stations are required to transmit at 54Mbps for QoS guarantees, and do not change their PHY rates. We increase the number of stations, starting from 1, until the wireless LAN cannot accommodate any more stations (or streams). For the EDCA case, we control the TXOP limit for airtime usage control. Since all streams have the same guaranteed rate (g_i =5 Mbps) and minimum PHY rate (R_i =54Mbps), each station uses the same TXOP limit in this scenario. For the HCCA case, we follow the procedures in Section 4.1.

Figure 4.11 plots the total throughput under the HCCA and the EDCA. Since all stations request the same guaranteed rate, one can easily convert the total throughput to the total number of stations (i.e., streams) admitted into the wireless LAN. We increment the number of stations every 5 seconds in order to explicitly show the throughput received by individual streams. Prior to t = 35 second, every admitted stream gets exactly the 5-Mbps guaranteed rate under both the HCCA and the EDCA. It shows that using the enhanced EDCA can achieve the same QoS guarantees as using the polling-based HCCA.

After t = 35 second, the number of stations is increased to 8. The figure shows that using the EDCA cannot guarantee the streams' QoS any more because it needs a total throughput of 40 Mbps to support 8 streams, but the wireless LAN can only provide about 37Mbps. However, under the HCCA, all streams are still provided with the 5-Mbps guaranteed rate. This result is expected because the HCCA uses the polling-based channel access (in contrast to the contention-based EDCA), hence

[4]The average bit rate of a DVD-quality (MPEG-2) video is about 5Mbps.

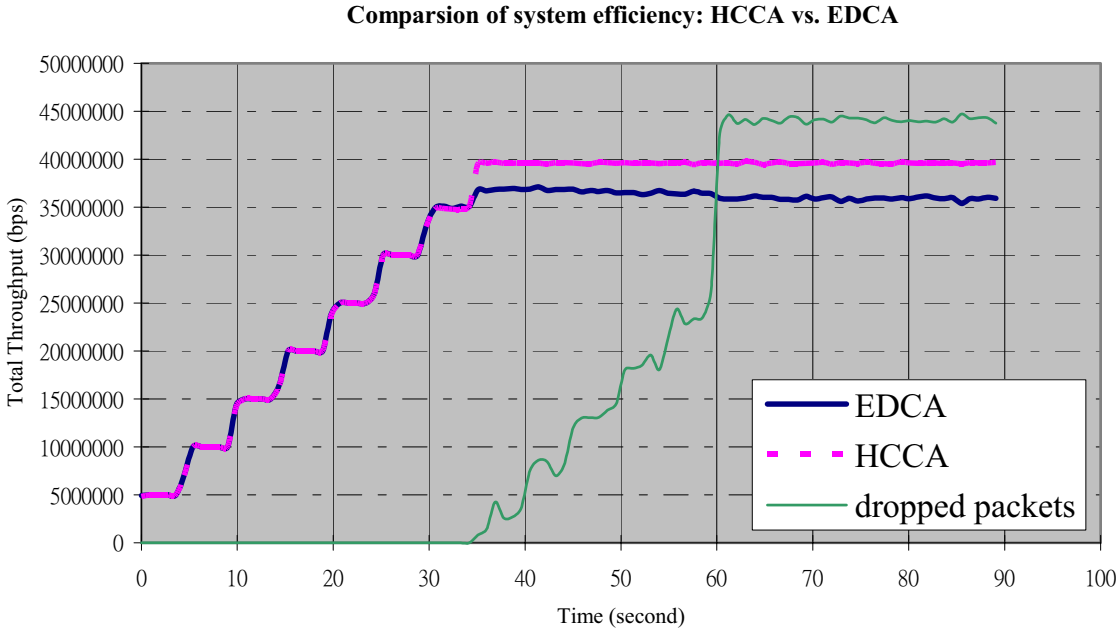


Figure 4.11. Comparison of system efficiency, in terms of the total throughput, between the HCCA and the EDCA. *A new station carrying a single stream is added to the wireless LAN about every 5 seconds and transmits at 54 Mbps. The height of each "stair" in the figure is equal to a stream's guaranteed rate = 5 Mbps.

resulting in a higher efficiency. After $t = 40$, more stations using the EDCA are added to the wireless LAN and the total system throughput starts to drop gradually. At $t = 60$ second where there are 16 stations in the wireless LAN, the system throughput becomes 36 Mbps, compared to the maximum achievable throughput of 37 Mbps. Such decrease in the system throughput results in that more collisions occur when the number of stations increase. The amount of dropped frames under the EDCA is also plotted which shows that frame dropping starts at $t = 35$ second. In contrast, the maximum achievable throughput under the HCCA remain at 40 Mbps based on the parameters we used in our simulation. The efficiency of the HCCA mainly depends on the frame size used by individual stations. If a larger frame size (we use 1500 bytes) is used, the maximum achievable throughput can be increased to 43 Mbps [117].

Based on the simulation results, one can also obtain the values of the effective airtime EA in Eq. (4.5). Because all streams are transmitted at the same PHY rate, the value of EA can be computed by

$$EA = \frac{\text{system total throughput}}{\text{PHY rate}} \tag{4.16}$$

Therefore, we have $EA = 0.67$ under the EDCA and $EA = 0.73$ under the HCCA. Although the value of EA varies under the EDCA (depending on the EDCA parameters used), it is always within the range between 0.65 and 0.68 in our simulation. We use $EA = 0.65$ in Eq. (4.5) for a more conservative admission control under the EDCA.

Although using the HCCA achieves a better efficiency, it only generates $0.06 = 0.73 - 0.67$ second more data-transmission time (within a one-second period) or about 3Mb more data frames when all stations transmit at 54Mbps (the maximal PHY rate in the 802.11a PHY spec.). When stations use smaller PHY rates, the small difference between the EA values of the EDCA and the HCCA results in an even smaller throughput difference. Therefore, one can expect that using the EDCA and the HCCA will generate a similar performance, especially in terms of the total number of admissible streams.

4.5.2 Scenario 2: TXOP Limit vs. Medium Accessing Frequency

In this subsection, we compare the two controlling methods in the EDCA, namely, controlling the stations' TXOP limits and medium accessing frequency. We still assume that each stream requires a 5-Mbps guarantee rate. In order to emphasize the EDCA's quantitative control over stations' diverse airtime usage, we assume that stations 1 and 2 carry a single traffic stream but stations 3 and 4 carry 2 streams. That is, there are six traffic streams in total. We again assume that all stations transmitted at 54Mbps and do not change their PHY rate. Therefore, all streams are able to obtain their guaranteed rate based on the results in Scenario 1. In order to control the stations' medium accessing rate, we choose CW_{min} as the control parameter. Therefore, we choose $CW_{min,1} = CW_{min,2} = 15(2^4 - 1)$ and $CW_{win,3} = CW_{win,4} = 31(2^5 - 1)$ based on Eq. (4.15), and set $CW_{max} = 63(2^6 - 1)$ for all stations. The TXOP limits are chosen according to Eqs. (4.10) and (4.12).

Figure 4.12 plots the total throughput of using the two controlling methods. It shows that both methods generate identical results (in terms of throughput). One can observe that stations 1 and 2 both receive the 5-Mbps guaranteed rate after

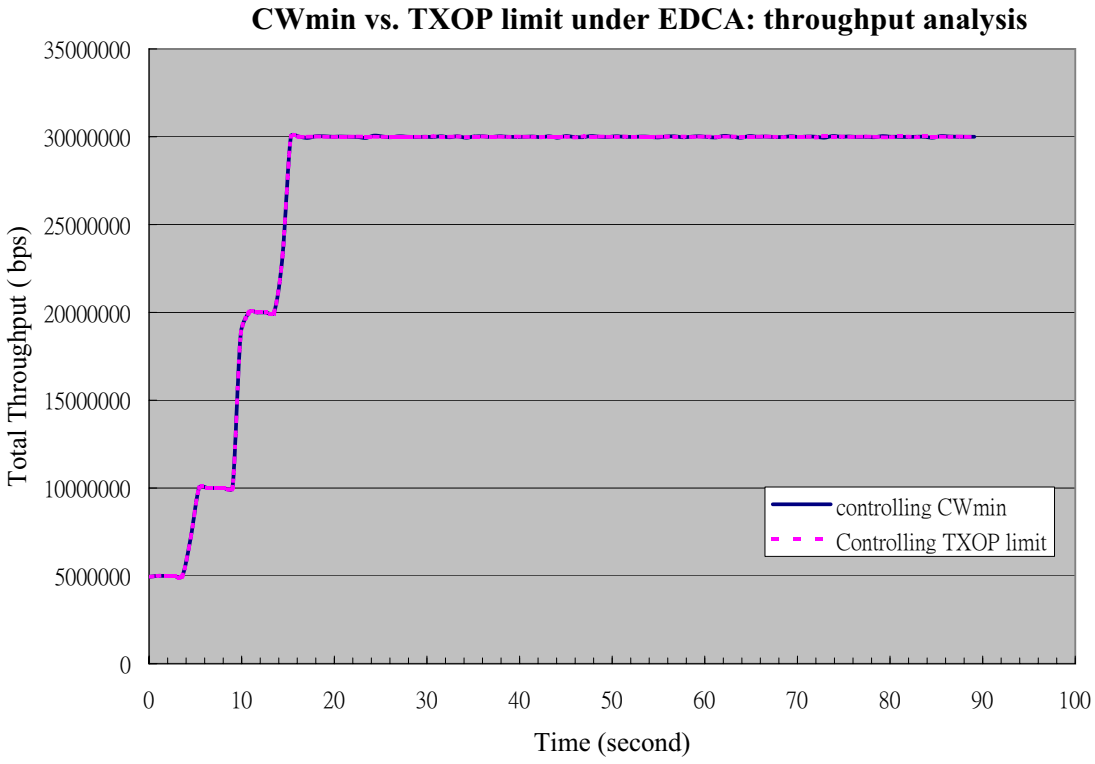


Figure 4.12. Comparison of throughput between controlling stations' TXOP limits and CW_{min} values. *The figures shows that in the EDCA, controlling stations' TXOP limits and CW_{min} values result in the same performance in terms of streams' throughput.

they join the wireless LAN at $t = 0$ and $t = 5$, while stations 3 and 4 both receive 10 Mbps (5 Mbps for each of their own two streams) after they join the wireless LAN at $t = 10$ and $t = 15$. The results show that both controlling methods can realize the distributed and quantitative control over stations' airtime usage. Here, the throughput is proportional to airtime usage since all stations transmit at the same PHY rate.

Figure 4.13 plots the delay under the two controlling methods. Once all 4 stations (all 6 streams) are admitted to the wireless LAN, the delay remains around 0.8 msec if using the TXOP Limit control, or fluctuates around 1.2 msecs if using the CW_{min} control. The reason why the delay fluctuates in the latter is that if stations using larger CW_{min} (i.e., 31) collide with other stations, they use $CW_{max} = 63$ as the contention window size due to the exponential random backoff. Thus, these stations may wait much longer as compared to the case of controlling the TXOP Limit where

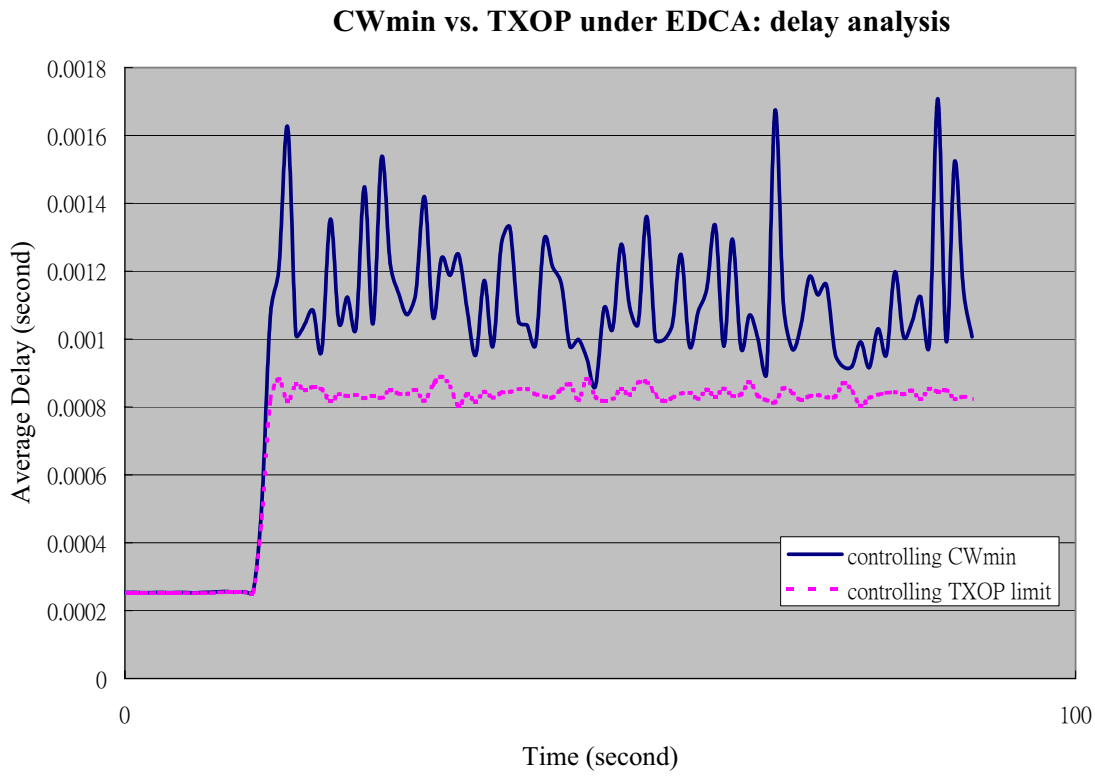


Figure 4.13. Comparison of delay between controlling stations' TXOP limits and CW_{min} values. *The figures shows that in the EDCA, controlling CW_{min} values may result in a large delay variance but still satisfy all stream's delay bound.

stations (rarely) use $CW_{max} = 63$ only when 2 consecutive collisions occur. In any case, the delay under both methods are well below the streams' delay bound, which is 200 msecs in our simulation.

4.5.3 Scenario 3: Time-varying Transmission Rates: a Heavy-load Case

The main advantage of our airtime-based admission control over a rate-based admission control is that when some stations lower their PHY rates, they do not affect other stations' airtime allocation and QoS guarantees. Instead, only the QoS of the stations lowering their PHY rate below the negotiated minimum PHY rates are compromised. To simulate this scenario, we assume that there are 4 stations where station 1 carries a 5-Mbps stream and stations 2-4 each carry 2 5-Mbps streams. All stations are required to transmit at 54Mbps to maintain their QoS. That is, the negotiated minimum PHY rate is 54 Mbps for all stations. Furthermore, we assume that station

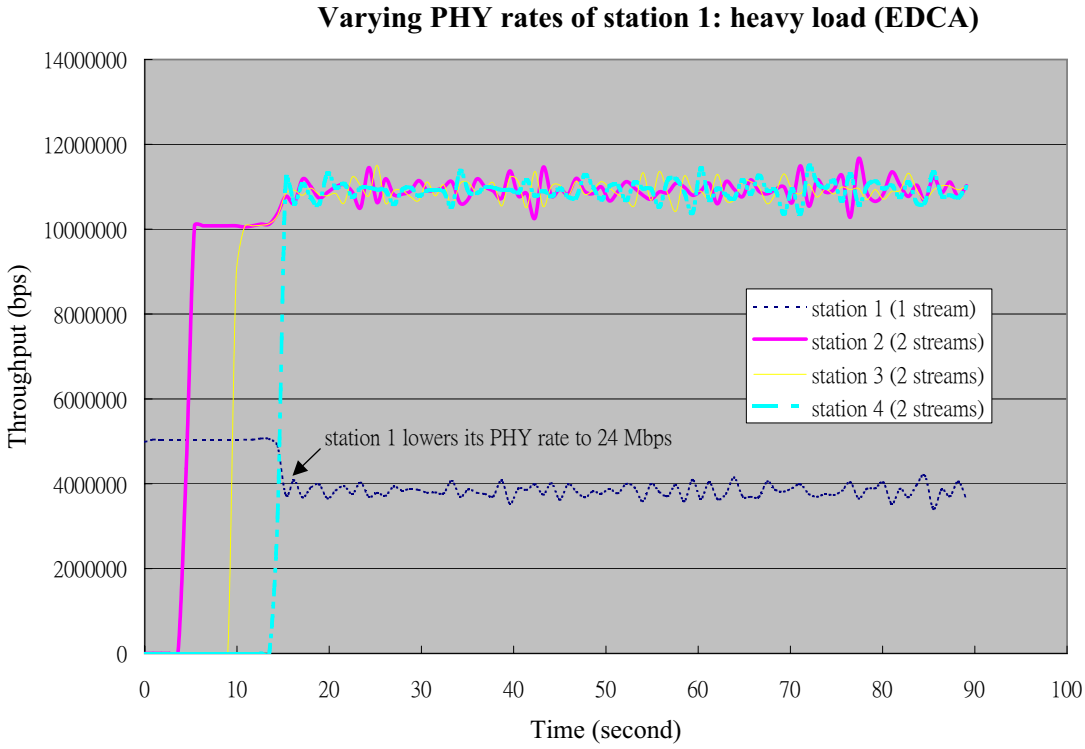


Figure 4.14. Throughput of individual streams in the EDCA: station 1 lowers its PHY rate to 24 Mbps at $t = 15$ second. *The wireless LAN has been heavily loaded before station 1 lowers its PHY rate. Therefore, the wireless LAN cannot provide station 1 the guaranteed rate once station 1 lowers its rate. However, all other stations are not affected as in the HCCA case shown in Figure 4.15.

1 lowers its PHY rate to 24 Mbps due to the link adaptation at $t = 15$ second.

Figures 4.14 and 4.15 plot the throughput of individual stations under the EDCA (controlling the TXOP limits) and HCCA, respectively. These figures show that stations 2-4 that maintain their PHY rate always receive at least 10-Mbps throughput (5 Mbps for each of their own 2 streams) after they join the wireless LAN at t =5, 10, and 15 second, respectively. The only station that receives a throughput less than the guaranteed rate is station 1, which violates the agreement on maintaining the minimum PHY rate at 54 Mbps. The result verifies that our integrated scheme can effectively maintain the QoS for stations complying with the QoS negotiation and "isolates" the stations that violate the QoS negotiation from others in a distributed manner, as compared to the polling-based HCCA.

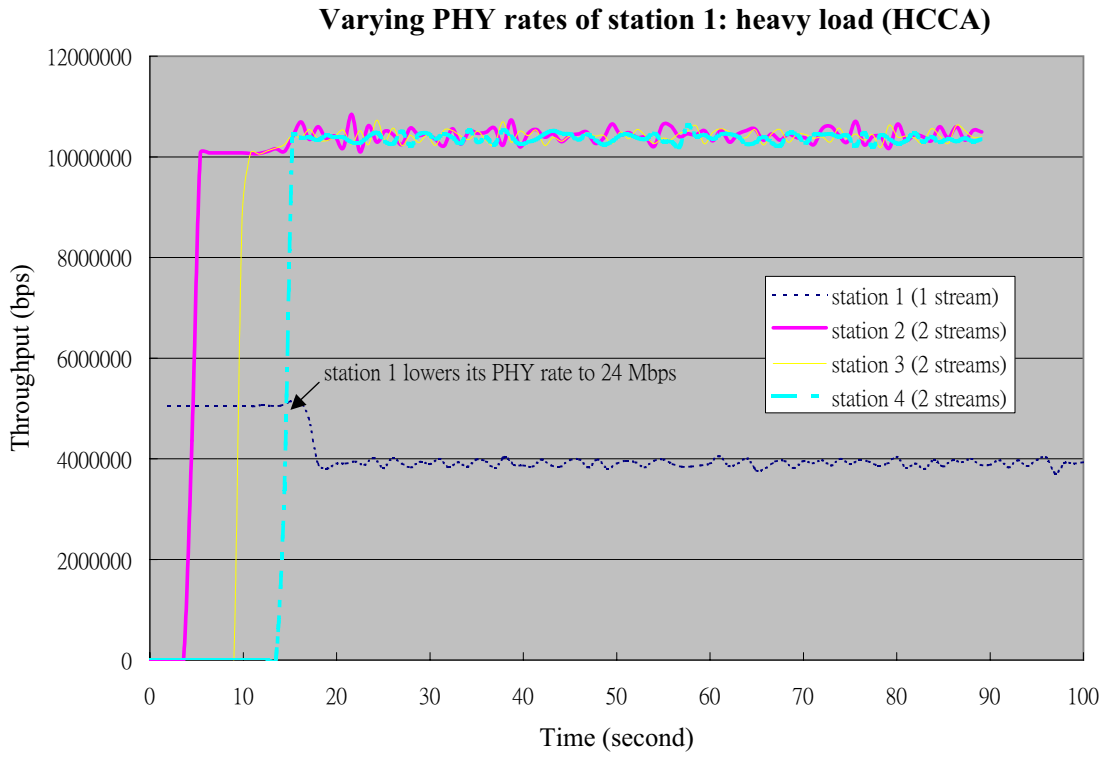


Figure 4.15. Throughput of individual streams in the HCCA: station 1 lowers its PHY rate to 24 Mbps at $t = 15$ second. *The wireless LAN has been heavily loaded before station 1 lowers its PHY rate. Therefore, the HC cannot provide station 1 the guaranteed rate once station 1 lowers its rate.

4.5.4 Scenario 4: Time-varying Transmission Rates: a Light-load Case

In Scenario 3, we conclude that stations lowering their PHY rates below the negotiated minimum PHY rates do not receive the QoS guarantees. However, we also mentioned in Section 4.3 that when a wireless LAN has some unutilized resource (i.e., the airtime), the AP may temporarily allocate more resources to the stations lowering their PHY rates — without violating other stations' QoS — so as to support their QoS at lower PHY rates. This can be done via the HC in the HCCA by computing a new service schedule. In Section 4.3, we claim that these adjustments can be completed without any centralized control if using the enhanced EDCA, thanks to the autonomous distributed airtime control.

To simulate this scenario, we assume that the wireless LAN only admits 4 stations before $t = 15$ second, and stations 1, 2 and 4 carry a single stream and station 3 carries 2 streams. We again assume that each stream requires a 5-Mbps guaranteed rate and

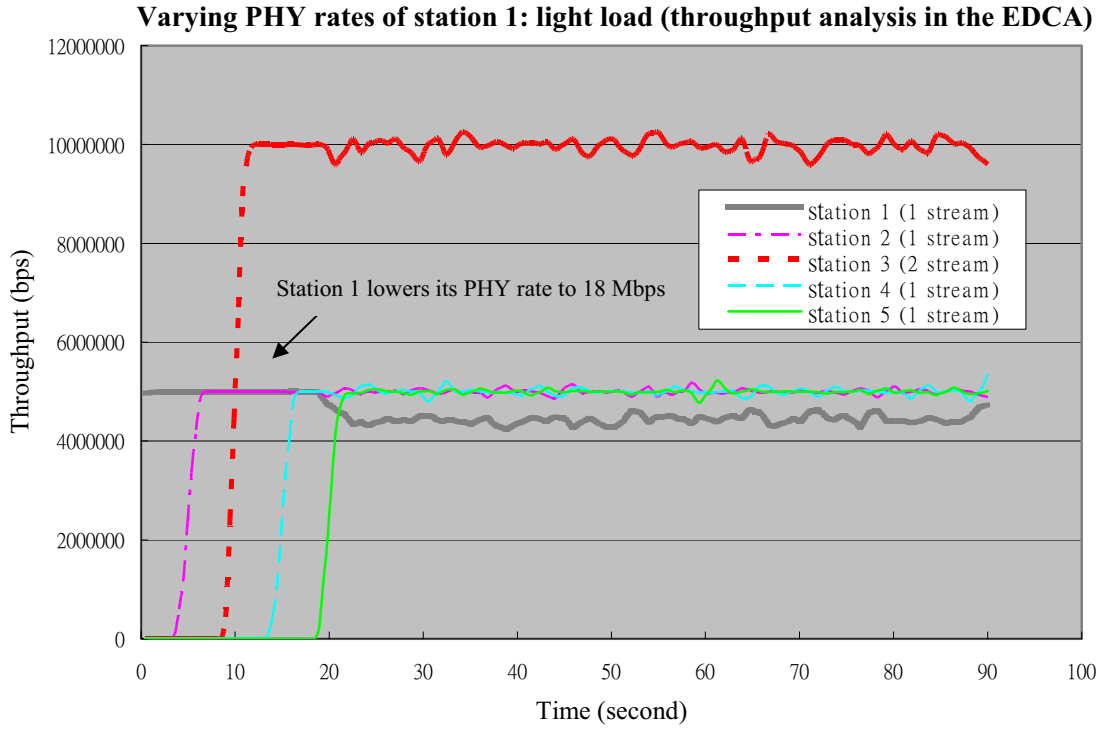


Figure 4.16. Throughput of individual streams in the EDCA: station 1 lowers its PHY rate to 18 Mbps at $t = 15$ second. *The wireless LAN is not heavily loaded when station 1 lowers its PHY rate at $t = 15$ second. Therefore, station 1 can still receive the 5-Mbps guaranteed rate after $t = 15$. However, after $t = 20$ second, station 1 has to "relinquish" the extra airtime it is using so that station 5, which complies the minimum PHY rate of 54 Mbps receives the 5-Mbps guaranteed rate.

that all stations are required to transmit at 54 Mbps to maintain their QoS. We assume that station 1 lowers its PHY rate to 18 Mbps at $t = 15$ second. Unlike Scenario 3, the wireless LAN is still able to (but not necessarily has to) provide the QoS to station 1 without affecting other stations' since there are only 5 streams asking a total amount of airtime (before $t = 20$ second)

$$\frac{4 * 5}{54} + \frac{5}{18} = 0.64 < 0.65 = EA_{edca}. \tag{4.17}$$

We can observe in this figure that station 1 still obtains the required 5-Mbps guaranteed rate even though it violates the agreement upon using a 54-Mbps transmission rate. *Here, we do not need any additional adjustments as required in the HCCA. Instead, station 1 automatically adjusts its airtime usage by contending the wireless medium more frequently via the enhanced EDCA, due to the build-up MAC buffer queue.*

After $t = 20$, we add station 5 which also carries a 5-Mbps stream into the wireless

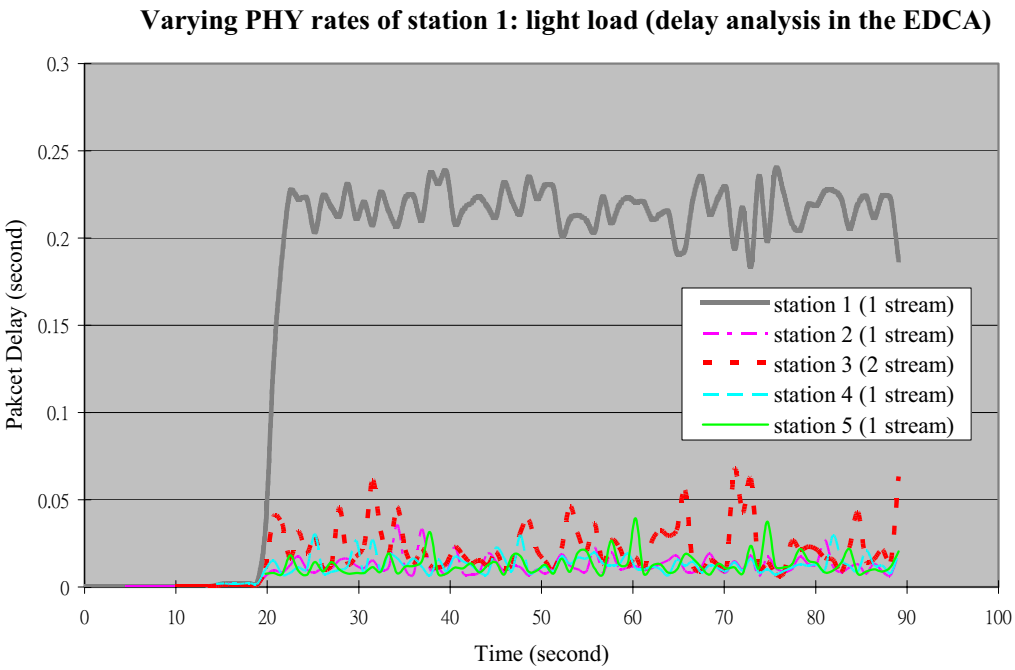


Figure 4.17. Delay of individual streams in the EDCA: station 1 lowers its PHY rate to 24 Mbps at $t = 15$ second. *The wireless LAN is not heavily loaded when station 1 lowers its PHY rate at $t = 15$ second. Therefore, all streams' delay bound are still satisfied after $t = 15$. However, after $t = 20$ second, station 1 has to "relinquish" the extra airtime it is using so that station 5, which complies the minimum PHY rate can receive the QoS. As a result, station 1's stream experiences a delay greater than the required delay bound at $t = 20$ second.

station. When station 5 requests for admission at $t = 20$ second, the AP should admit it based on Eq. (4.5)

$$\frac{6 * 5}{54} = 0.55 < 0.65 = EA_{edca}, \tag{4.18}$$

since all stations are required to transmit at R_i=54 Mbps. However, not all stations actually transmit at 54 Mbps. The total amount of airtime we really need to support QoS for all streams is

$$\frac{5 * 5}{54} + \frac{5}{18} = 0.73 > 0.65 = EA_{edca}, \tag{4.19}$$

where stations 2-5 have 5 streams in total to transmit at 54 Mbps and stations has 1 stream to transmit at 18 Mbps. Obviously, station 1 should not receive the QoS (5-Mbps guaranteed rate). Figure 4.16 again shows this "expected" behavior and *the most important fact is that such adjustment is again achieved automatically (via the EDCA parameters) without any adjustment which is required in the HCCA*. Figure 4.17 shows the delay of data frames from individual stations. Again, before

$t = 20$ second, the delay bound of station 1 is satisfied even though station 1 violates the minimum PHY rate requirement. However, such QoS is not guaranteed any more after $t = 20$ second, because station 5 joins the wireless LAN and complies with the minimum PHY rate requirement.

4.6 Conclusion

In this chapter, we provided a complete set of QoS solutions for the infrastructure-mode 802.11 wireless LAN using both the HCCA and the EDCA, and for the ad hoc-mode 802.11 wireless LAN. In order to provide parameterized QoS guarantees in the EDCA, we exploited the distributed airtime usage control developed in Chapter 3. We also extended the current QoS signaling of the HCCA to do admission control for the parameterized QoS in the EDCA. The simulation results showed that by using the EDCA, we are able to achieve the same level of parameterized QoS support as the HCCA, but results in less complexity than the centralized, polling-based HCCA scheme.

CHAPTER 5

Spectral-Agile Radios

The most important task of a network to support QoS is to provide users their required bandwidth. Therefore, as long as the network has sufficient system bandwidth, providing QoS support is a relatively easy task. Unfortunately, this is not the case in conventional wireless networks where the system bandwidth is a very precious and limited resource. Although such limitation is due to the scarcity of the wireless spectrum, it is the static spectrum allocation policy that prevents wireless networks from utilizing the spectrum more efficiently, and acquiring more usable bandwidth.

Under the current static spectrum allocation policy, wireless devices are only allowed to operate in designated spectral bands. For example, the IEEE 802.11b and 11g wireless stations are only allowed to operate in the unlicensed 2.4 GHz band, and so are the Bluetooth devices and cordless phones. These devices (in the crowded unlicensed bands) are prohibited from using other spectral bands even though those spectral bands may never or rarely be utilized by their designated users. As a result, these wireless devices get stuck in the heavily-used spectral bands, competing with each other for a very limited bandwidth, while many other spectral bands are left unused. One can expect that if the wireless devices (in crowded spectral bands) are allowed to explore and utilize the rarely-used spectral bands opportunistically, not only the performance of individual devices but also the overall spectrum efficiency can be improved.

In this chapter, we propose a new type of wireless communication based on *opportunistic use of the wireless spectrum*. This new type of communication, referred to as the spectral-agile communication, relies on radio devices' capability of seeking and utilizing (in real time) the spectral resources — in time, frequency and space domains.

From the perspective of QoS provisioning, using spectral agility helps a radio device acquire more spectral resources so as to provide users better QoS. Of course, the spectral-agile communication cannot be realized without developing new spectrum access mechanisms. Therefore, we propose a comprehensive framework along with resource monitoring and utilization functionalities to facilitate the adoption of spectral agility. Moreover, we establish a mathematical model to evaluate the potential performance gains of using the spectral agility.

This chapter is organized as follows. Section 5.1 describes the system model and assumptions for our development of spectral-agile communication. In Section 5.2, we present the mathematical model, and discuss and analyze the numerical results. Section 5.3 details the framework for spectral-agile communication, and the associated functionalities. The ns-2 based simulation results are analyzed and discussed in Section 5.4. Finally, conclusions are drawn in Section 5.5.

5.1 System Model

We consider two types of radio devices, namely *primary* and *secondary* devices. A primary radio device has exclusive access to designated spectral bands while a secondary radio device only accesses a spectral band when the corresponding primary device does not use that band. For example, a primary device can be any radio device in licensed bands, and a secondary device can be any an unlicensed-band device such as an IEEE 802.11 wireless station. To realize the secondary device's opportunistic use of primary devices' spectral resources, we assume that a secondary device has spectral agility, which is enabled by the software defined radio (SDR). It is then a secondary device's responsibility to locate available resources, in both spectral and temporal domains, as shown in Figure 5.1.

Even though it is desirable to have the entire spectrum accessible to a spectral-agile device, hardware limitations (such as antenna design) usually determine the accessible range. Therefore, the term "wireless spectrum" in this chapter is referred to as the portion of the wireless spectrum which can be accessed by a spectral-agile. The spectrum is divided into "channels," each of which is the smallest unit of a spec-

tral band. We assume that each secondary device only uses a single channel for basic communication, but should be able to use multiple channels for better performance. For example, a secondary device may adopt a modulation scheme that supports various bit rate or simply adjust the number of subcarriers in the Orthogonal Frequency Division Multiplexing (OFDM) signals, when multiple channels are available.

We assume that the temporal usage of each channel (by the primary devices of that channel) is an independent random process. Since the primary device may not use its designated channel all the time, there exist some "holes" or idle time slots, in that channel which may be exploited by secondary spectral-agile devices. As shown in Figure 5.1, the blank slots represent such holes, each of which is referred to as a *spectral opportunity* in the rest of the chapter. For example, there exists a spectral opportunity in channel 4 after $t = t_1$. Moreover, the entire spectrum is regarded as providing a spectral opportunity during $[t_2, t_3]$. Depending on the primary device's spectrum usage pattern, the duration of a spectral opportunity can be up to several hours or even days (e.g., in spectral bands reserved for emergency), or can be only few milliseconds (e.g., in heavily-used spectral bands). It is relatively easy for a secondary spectral-agile device to use long-lasting opportunities. However, for the short-lasting opportunities, a secondary spectral-agile device may not be able to detect their existence so as to utilize them before they disappear. Therefore, we only focus on the case when spectral opportunities last in the order of seconds or longer.

It should be noted that our problem differs significantly from the problems of using dynamic frequency selection mechanisms in the existing systems, such as Dynamic Channel Selection (DCS) [90] in cellular networks, Dynamic Frequency Selection (DFS) [91] in the IEEE 802.11h standard or Auto Frequency Allocation (AFA) [92] in the HiperLAN. These schemes address the problem of choosing a good channel (either a frequency in the Frequency Division Multiple Access (FDMA) system, or time slots in the Time Division Multiple Access (TDMA) system) so that transmission in that channel may experience less interference or cause less interference to other transmissions in the same channel. In our problem, a spectral-agile device seeks both spectral and temporal opportunities in the wireless spectrum, and utilizes these opportunities

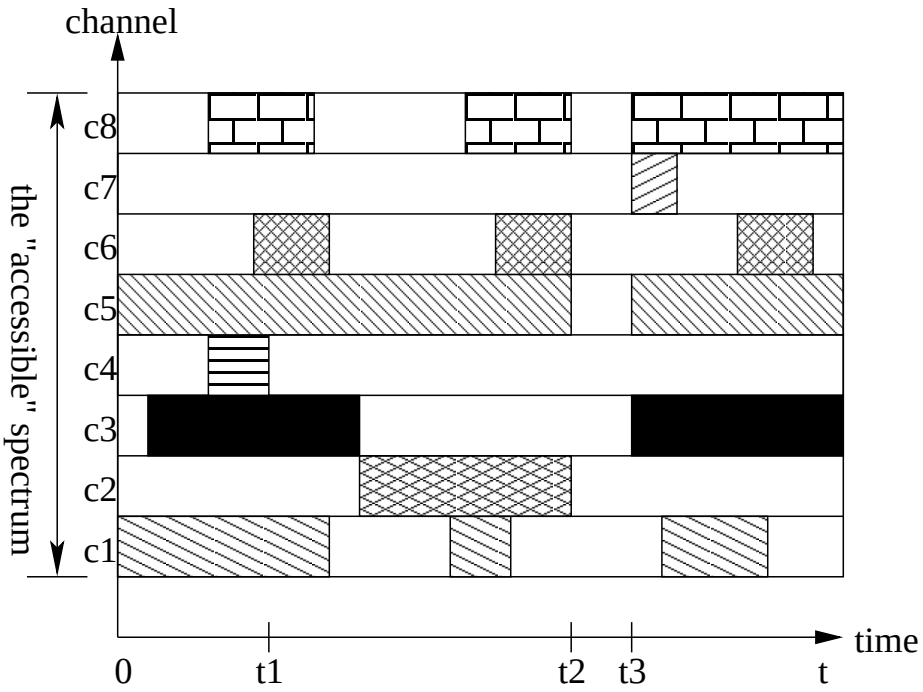


Figure 5.1. Spectrum opportunities for spectral-agile devices

in an opportunistic manner. Among the thus-found opportunities, a spectral-agile device decides which opportunities to use and when to utilize them. If and when activities of a primary device are detected, the secondary spectral-agile devices must vacate the channel in order not to interfere with the primary device. In the case that a set of spectral-agile devices communicate with each other, all these devices must always take the same spectral opportunity to maintain their inter-connectivity. Therefore, the spectral-agile devices belonging to the same communication group may disseminate the information about the found spectral opportunities and how to utilize these spectral opportunities. These procedures are detailed in Section 5.3.

5.2 Analytical Model for Performance Improvements

We establish a mathematical model to analyze the potential performance gains of using spectral agility. In order to measure the performance of spectral-agile devices, we use two performance metrics, namely the spectral utilization and packet blocking time. The spectral utilization is defined as the percentage of time during which a secondary spectral-agile device has the access to some channels for transmission.

One can convert this channel accessing time to the throughput once the underlying medium access control (MAC) and modulation mechanisms are specified. Therefore, we use the channel accessing time so as not to be confined to any specific MAC and modulation schemes. The packet blocking time is defined as the time interval during which a secondary device has no spectral opportunity to utilize (thus, it has to suspend all transmissions).

We assume that there are N channels in total, each with its own designated primary devices, and there are M secondary spectral-agile devices seeking spectral opportunities. The usage pattern of the primary devices in each channel is assumed to be an *i.i.d.* ON/OFF random process with independent ON- and OFF-periods. An ON-period represents that a channel is occupied by its primary devices while an OFF-period is regarded as a potential spectral opportunity for spectral-agile devices. To simplify our analysis, we assume that the distributions of both ON- and OFF-periods in each channel are exponentially-distributed with means equal to T_{on} and T_{off}, respectively. We will explore different distributions using simple simulations at the end of this section.

In order to provide a performance upper-bound, we assume that each spectral-agile device has an infinite amount of traffic to transmit. Moreover, each spectral-agile can scan a channel, vacate a channel (when the channel is reclaimed by primary devices), and switch to a new channel instantly without incurring any control overhead or delay. The control overhead and delays are implementation-dependent, and their impacts on the performance of spectral-agile devices are investigated in Section 5.3. In order to demonstrate the performance gain of using spectral agility, we use performance of non-agile secondary devices as the comparison basis. The no-agile devices listens to a fixed channel, and transmits only when that channel is not used by the primary devices. The spectral utilization of a non-agile secondary device can easily be computed as $\frac{T_{off}}{T_{on}+T_{off}}$, and the average blocking time is T_{on}.

5.2.1 A Special Case: $M = 1$

We first consider a special case when there is only one spectral-agile secondary device. As shown in Figure 5.2, the only time interval during which a spectral-agile device has no channel for traffic transmission is when all channels are occupied by the primary devices. Such blocking intervals, denoted as t_{block}, always begin when a channel switches from an OFF-period to an ON-period and ends when one channel switches from an ON-period to an OFF-period. Therefore, t_{block} is computed as

$$t_{block} = \min_{i=1,2,\cdots,N}(T^{(i)}_{remain}), \tag{5.1}$$

where $T^{(i)}_{remain}$ is the remaining ON-period in channel i. Assuming that the ON-periods are independent and exponentially distributed, one can compute the distribution of t_{block} as

$$P(t_{block} = t) = \frac{N \cdot e^{-\frac{T_{on}}{N}t}}{T_{on}}. \tag{5.2}$$

Eq. (5.2) shows that with spectral agility, a secondary device can reduce the average packet blocking time to $\frac{T_{on}}{N}$, as compared to T_{on} in the case of without using agility. The spectral utilization of such a spectral-agile secondary device is obtained by

$$U = 1 - \frac{N(p^{N-1} \cdot \frac{T_{on}}{N})}{T_{on} + T_{off}}, \tag{5.3}$$

where $p = \frac{T_{on}}{T_{on}+T_{off}}$ is the probability that a channel is occupied by the primary devices. Eq. (5.3) is derived based on the fact that a blocking interval starts only if a channel switches from an OFF-period to an ON-period while all other channels have already been in the ON-periods. Eq. (5.3) can be simplified further to

$$U = 1 - (\frac{T_{on}}{T_{on} + T_{off}})^N, \tag{5.4}$$

showing that the spectral utilization of a spectral-agile secondary device is a simple function of the channel load (generated be the primary devices). Finally, the improvement of the spectral utilization achieved by a spectral-agile secondary device is computed as

$$I = \frac{U}{1 - \frac{T_{on}}{T_{on}+T_{off}}} - 1, \tag{5.5}$$

as compared to the no-agile secondary device.

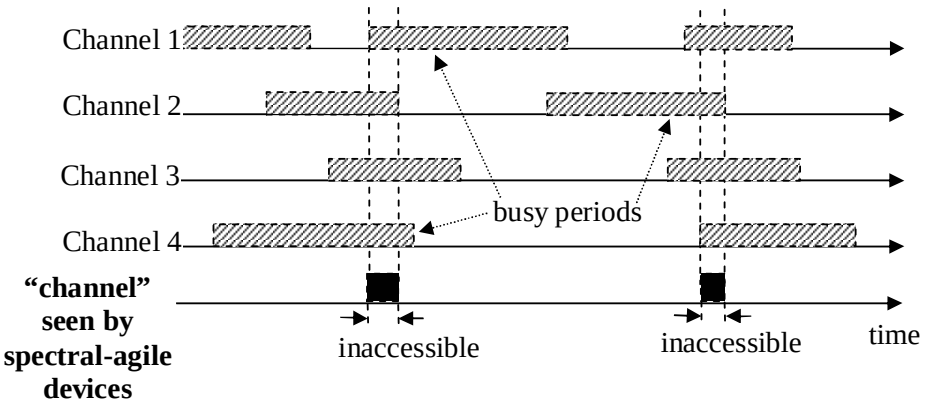


Figure 5.2. A special case: N=4

5.2.2 The General Case: $M > 1$

Eq. (5.4) shows that the spectral utilization of a spectral-agile secondary device is simply a function of the channel load generated by the primary devices, $\tau = \frac{T_{on}}{T_{on}+T_{off}}$. We can generalize this simple equation for the case when different channels have different utilizations, say, channel i with utilization $\tau_i = \frac{T_{on}^{(i)}}{T_{on}^{(i)}+T_{off}^{(i)}}$. Based on Eq. (5.4), the fraction of time during which there are k channels available simultaneously is computed as

$$r_k = \sum_{c=1}^{\frac{N!}{k!(N-k)!}} \left[\prod_{i \in S_c^k} (1-\tau_i) \prod_{j \in \{1,2,\cdots,N\}-S_c^k} \tau_j \right], \tag{5.6}$$

where S_c^k is a set of k channels, chosen from N channels, which are available for spectral-agile secondary devices. For example, we can set $S_1^k = \{1, 2, \cdots, k\}$, $S_2^k = \{2, 3, \cdots, k+1\}$, and so on.

To further generalize our analysis, we assume that there are $M > 1$ spectral-agile secondary devices trying to exploit available spectral opportunities. Obviously, each spectral-agile device obtains exactly one channel if there are no less than M channels available. If less than M channels are available, the spectral-agile devices have no choice but to share whatever available to them. The spectral utilization of each spectral-agile device is then computed by

$$U_{agile} = \sum_{k=0}^{N} \frac{\min(M,k) r_k}{M}. \tag{5.7}$$

As we mentioned in Section 5.1, the SDR enables a radio device to dynamically use a

variety of MAC and modulation schemes, depending on the underlying wireless environment. Therefore, a spectral-agile device can use multiple channels simultaneously, thus acquiring more channel accessing time for better performance. We will discuss how to analyze the performance of using multiple channels in Chapter 6.

As for the non-agile secondary devices, there are two approaches to select channels when $M > 1$: (1) each device randomly selects its own channel independently of others, and (2) all secondary devices cooperate in a way that no more than one secondary device uses the same channel, if possible. The advantage of the first approach is the simplicity while the advantage of the second approach is that each secondary device obtains more channel accessing time.

Random Channel Selection

Given that a non-agile secondary device chooses channel i, the probability that the other k non-agile secondary device also choose the same channel is

$$p_k = \frac{(M-1)!}{k!(M-1-k)!}(\frac{1}{N})^k(\frac{N-1}{N})^{M-1-k}. \tag{5.8}$$

Therefore, the average channel accessing time that a non-agile device can acquire, given that it has chosen channel i, is

$$T_i = \sum_{k=0}^{M-1} p_k \frac{T_{off}^{(i)}}{(k+1)(T_{on}^{(i)} + T_{off}^{(i)})}. \tag{5.9}$$

The spectral utilization of each non-agile device is then computed as

$$U_{random} = \frac{1}{N}\sum_{i=1}^{N} T_i. \tag{5.10}$$

Coordinated Channel Selection

If each non-agile secondary device coordinates its selection of a channel with the others so as to maximize the spectral utilization, the spectral utilization is computed as

$$U_{coordinated} = \frac{\sum_{c=1}^{\frac{N!}{M!(N-M)!}} \frac{1}{M} \sum_{i \in S_c^M} \frac{T_{off}^{(i)}}{T_{on}^{(i)} + T_{off}^{(i)}}}{\frac{N!}{M!(N-M)!}}. \tag{5.11}$$

Here, we simply average all the possibilities of choosing M channels from N channels for non-agile secondary devices. We set $\frac{N!}{M!(N-M)!} = 1$ in case of $M > N$.

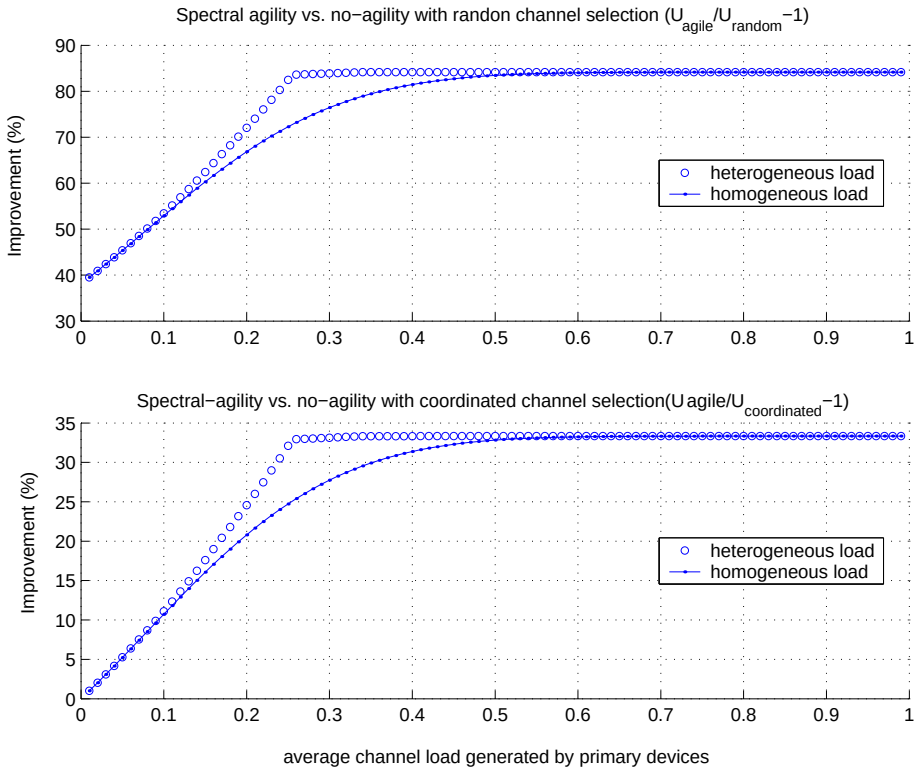


Figure 5.3. Improvement percentage of spectral utilization for spectral-agile devices: $N = 12$ and $M = 9$. *Although the figure shows the maximal improvement percentage (82%) occurs when the channel load approaches 1, it does not suggest that using spectral agility generates the greatest amount of spectral opportunities. Instead, it shows that, for example, with load of 0.99, the average channel accessing time for a spectral-agile device increases from 0.01=1-0.99 sec (i.e., no-agility) to 0.0182 sec out of an one-second period as also shown in Figure 5.4

We can now compare the spectral utilization between secondary devices using (1) spectral agility, (2) no agility with random channel selection (Approach I), and (3) no agility with coordinated channel selection (Approach II) based on Eqs. (5.7), (5.10), and (5.11). We investigate two scenarios with $N = 12$ and $N = 3$. The main reason for choosing these numbers is that there are 12 (non-overlapping) channels in the 5-GHz band for the IEEE 802.11a wireless LAN and 3 (non-overlapping) channels in the 2.4-GHz band for the IEEE 802.11b wireless LAN.[1] Therefore, even though spectral agility cannot be applied immediately to the licensed bands due to the current regulations, the 802.11 wireless LAN may use spectral agility to improve performance in the crowded, unlicensed bands.

Figure 5.3 shows the case of $N = 12$ and $M = 9$ with different average channel loads generated by the primary devices. For each given channel load, we choose the

[1]According to the US regulation, there will be more released channels in the 5-GHz band.

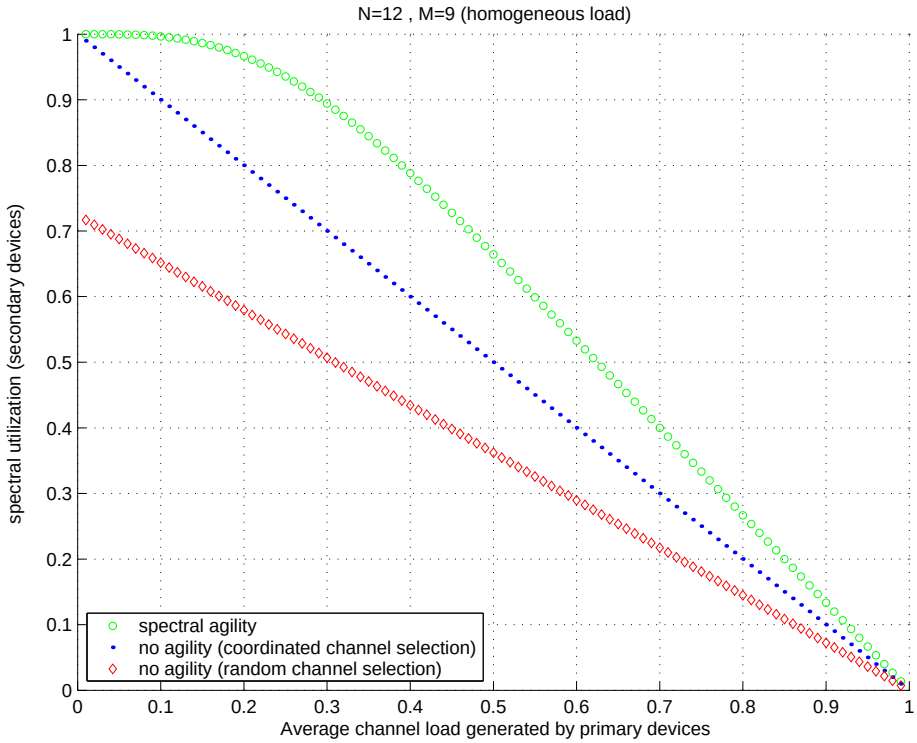


Figure 5.4. Spectral utilization: $N = 12$ and $M = 9$. *This figure, together with Figure 5.3, suggest that a spectral-agile secondary device benefits most from spectral agility when the channel load generated by a primary device is lightly-(0.2) or moderately-loaded ($0.7 \sim 0.8$).

loads of these 12 channels to be homogeneous or heterogeneous. In case of homogeneous loads, each channel is assigned a load equal to the average channel load, while, in case of heterogeneous loads, different channels are assigned different loads with their variance maximized (i.e., the utilization of each channel differs significantly from each other). The improvement shown in Figure 5.3 is defined as

$$\text{improvement } (\%) = (\frac{U_{agile}}{U_{random/coordinated}} - 1) \cdot 100\%, \tag{5.12}$$

where U_{agile}, U_{random}, and $U_{coordinated}$ are given in Eqs. (5.7), (5.10), and (5.11), respectively. The results demonstrate that a spectral-agile secondary device always achieves a higher spectral utilization than the devices without agility, either using random channel selection or coordinated channel selection. Of course, the improvement achieved by a spectral-agile is much smaller (still more than 25% in most cases) when compared to non-agile devices using coordinated channel selection (Figure 5.3-(b)). Note, however, that coordinated channel selection needs off-line channel information. If the channel loads range widely (i.e., heterogeneous loads), it is possible

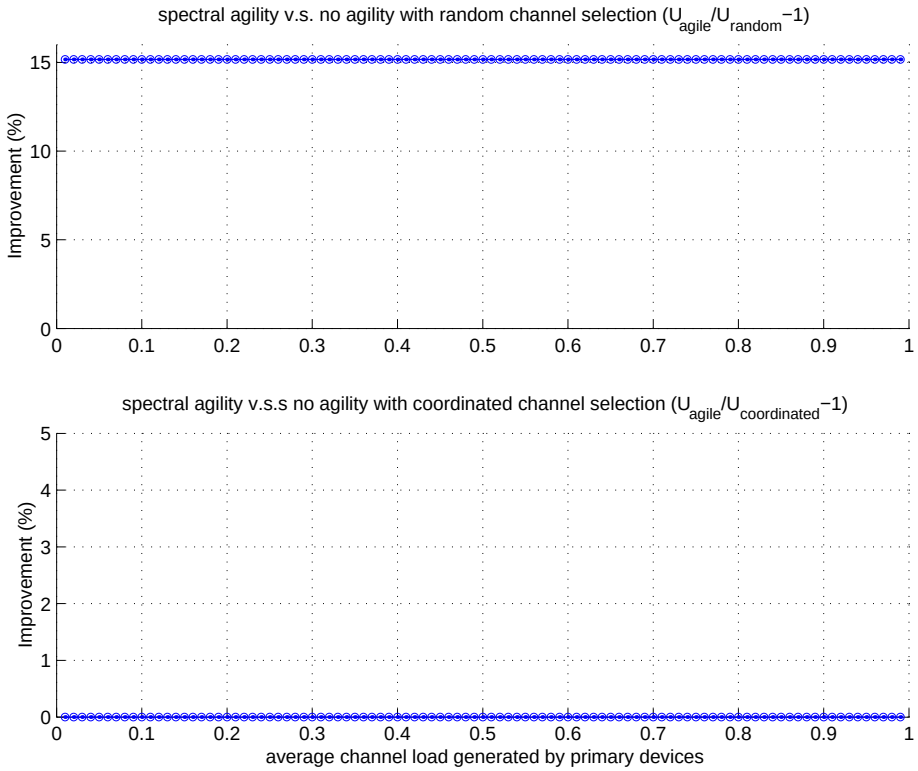


Figure 5.5. Improvement percentage of spectral utilization for spectral-agile devices: $N = 3$ and $M = 5$. *The figures shows that when the number of available channels is less than the number of secondary devices, using spectral agility generates the same performance as that of using static coordinated channel selection. However, spectral agility still outperforms static random channel selection.

that the non-agile secondary device may choose busier channels, regardless of whether or not the coordinated channel selection is used. In contrast, using spectral agility allows a secondary device to dynamically choose the channel with the least activities. Such advantages are also illustrated in Figure 5.3, where we achieve an extra 8-10% improvement under the case of heterogeneous loads when the channel load is around $0.2 \sim 0.3$.

An interesting observation is that the improvement ratio (i.e., Eq. (5.12)) saturates when the average channel load of the primary devices is greater than 0.5. This can be explained by Figure 5.4, in which the spectral utilization of secondary devices linearly decreases with the increase in the average channel load from primary devices beyond 0.3 in all three cases (i.e., with spectral agility, no agility with coordinated channel selection, and no agility with random channel selection). Because of such linearity, the improvement ratio of using spectral agility, as compared to no-agility

cases, remains unchanged when the channel load is greater than 0.3 in Figure 5.3. Figure 5.4 also suggests that when the average channel load of the primary devices is very large, it does not make much sense to use spectral agility as indicated by Figure 5.3 (even though it shows an 80% improvement with the load of 0.9). This is because when the channel is extremely busy, the amount of channel accessing time that each spectral-agile device can obtain is very small (less than 10% of the total time with the channel load of 0.9). Therefore, the control overhead (incurred by using spectral agility) may exhaust most of the channel accessing time a spectral-agile device acquires, hence, easily offsetting the improvement gained with spectral agility.

Next, we consider the case of $M > N$ and choose $N = 3$ and $M = 5$ as an example. Figure 5.5-(b) shows that using spectral agility and using no agility with coordinated channel selection achieve exactly the same performance (i.e., no improvement). The results make sense because when $M > N$, there are simply not enough channels for all secondary devices (so they have to share idle channels with each other). In fact, one can simplify both Eqs. (5.7) and (5.11) as

$$U_{agile} = U_{coordinated} = \frac{1}{M} \sum_{i=1}^{N} \frac{T_{off}^{(i)}}{T_{on}^{(i)} + T_{off}^{(i)}}, \tag{5.13}$$

when $M > N$ and verify the result in Figure 5.5-(b). There are some marginal improvements by using spectral agility as compared to using no agility with random channel selection as shown in Figure 5.5-(a). This is simply because some idle channels may be left unused in the case of random channel selection.

Figures 5.3 and 5.5 show that radio devices can only benefit from spectral agility when there are enough resources for opportunistic uses (i.e., $M < N$). Fortunately, field studies have shown that there are many under-utilized spectral resources in some wireless spectral band [93][94]. Moreover, there are two additional advantages of using spectral agility that we have not yet discussed when $M > N$. First, Eq. (5.2) shows that when the spectral agility is used, the average blocking time is reduced by a factor of N in the special case or reduced from $\frac{\sum T_{on}^{(i)}}{N}$ to $\frac{1}{\sum \frac{1}{T_{on}^{(i)}}}$ in the general case. Thus, even though the spectral utilization is not improved by using spectral

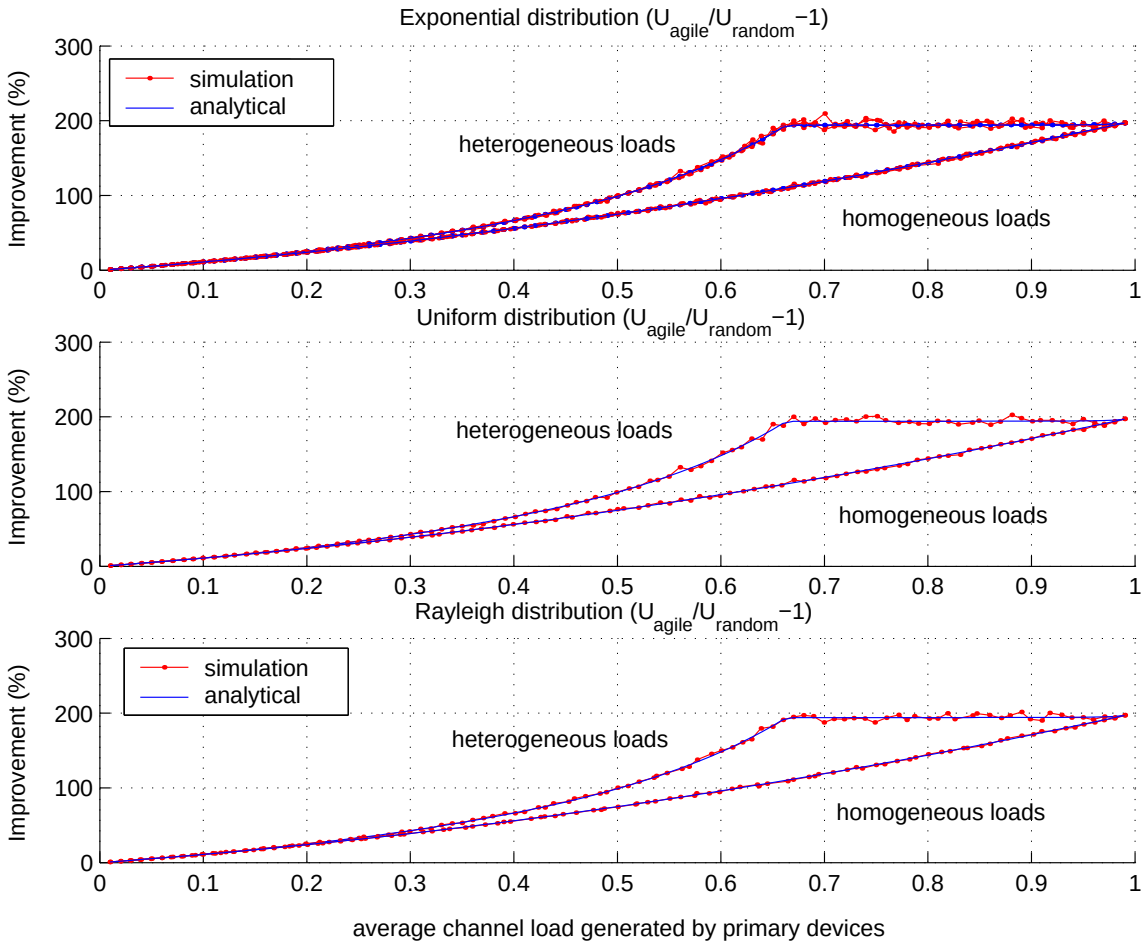


Figure 5.6. Improvement percentage of spectral utilization for spectral-agile devices: different ON/OFF distributions *Although the figure shows the maximal improvement percentage (200%) occurs when the channel load approaches 1, it does not suggest that using spectral agility generates the greatest amount of spectral opportunities. Instead, it shows that, for example, with load of 0.99, the average channel accessing time for a spectral-agile device increases from 0.01=1-0.99 (i.e., no-agility) to 0.03 sec out of an one-second period, similar to what shows in Figure 5.3.

agility when $M > N$, the packet delays are reduced significantly by using spectral agility. Another advantage is the spectral-agile device's capability of using multiple channels. In the above analysis, we assumed that a spectral-agile device always uses a single channel, even when more than one channel are available. We can expect that if a spectral-agile device can use all available channels, the performance must be improved.

Before concluding this section, we investigate the effects of different ON/OFF distributions on the improvement of spectral utilization by using spectral agility. The main purpose of this study is to verify the applicability of our model, which is

established based on the assumption of exponentially-distributed ON-/OFF periods. Here, we use Matlab to simulate the random ON/OFF periods and calculate the total time intervals of overlapping ON-periods (i.e., the blocking intervals for a spectral-agile device) for the case of $N = 3$ and $M = 1$. We use exponential (as in our earlier derivation), uniform, and Rayleigh distributions. Figure 5.6 shows a very good match between our analytical results and the simple simulation results, demonstrating the applicability of our analytical model. The reason why the improvement ratios (again as defined in Eq. (5.12)) are much higher (up to 200%) is that there is only one spectral-agile device seeking spectral opportunities, and thus, it need not share spectral opportunities with other spectral-agile devices. However, as we discussed earlier, such a large improvement ratio, in fact, represents only a very small increase of channel accessing time for a spectral-agile device if the average channel load of the primary devices is extremely high. Therefore, one should not expect improvement in reality, given the control overhead incurred by spectral agility, when the average channel load of the primary devices is very high.

5.3 Implementation of Spectral-agile Communication

In order to achieve the potential performance gains given in Section 5.2, spectral-agile devices must monitor the wireless spectrum, identify the idle channels and utilized the idle channels. In a more general scenario where several spectral-agile devices form a communicating group, these devices have to synchronize their use of spectral opportunities so as to maintain inter-connectivity among them. Moreover, different communicating groups may also need to coordinate with each other in a cooperative and fair manner. A framework to fulfill these tasks is illustrated in Figure 5.7. This framework consists of three parts, namely, spectral-agile devices, intra-group synchronization and inter-group coordination. The spectral-agile device is composed of three major modules: a resource monitor, a resource-use decision maker and a resource coordinator. The resource monitor is responsible for discovering usable spectral resources (referred to as spectral opportunities in this thesis), the resource-use decision maker determines a device's use of spectral resources, and the resource coordinator

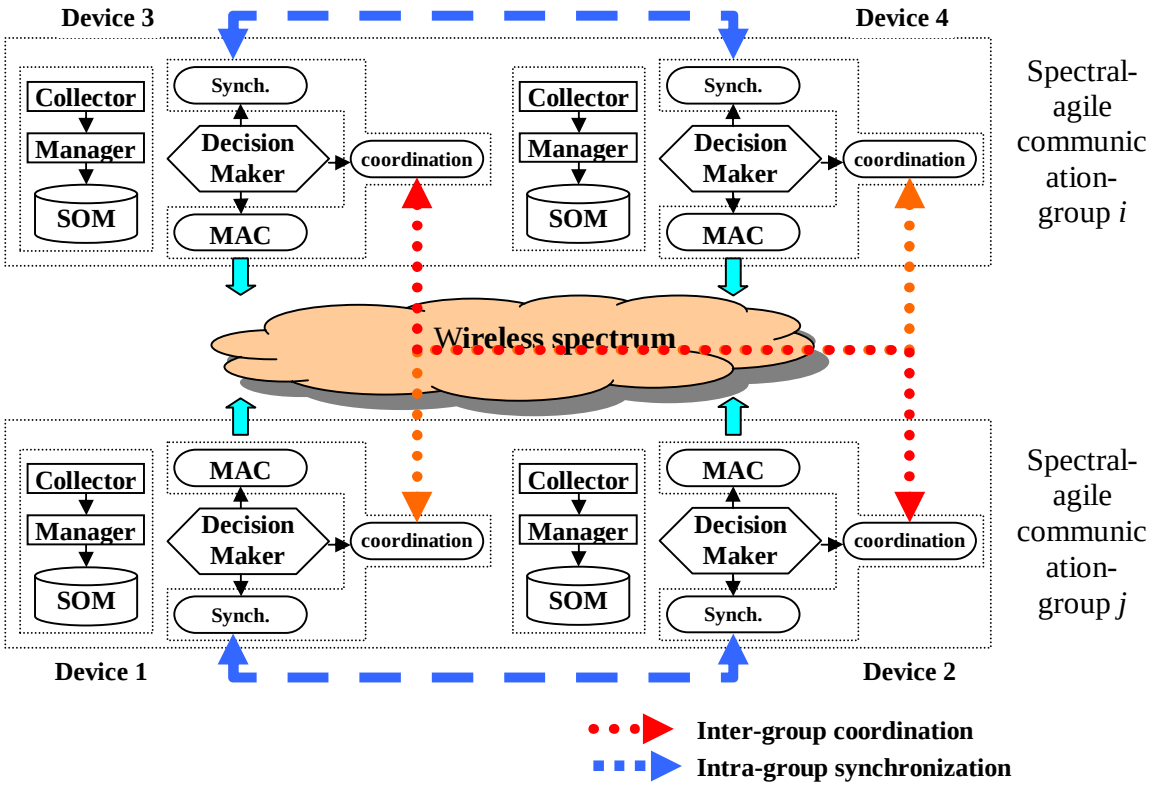


Figure 5.7. System framework for spectral-agile communication

maintains the intra-group synchronization and the inter-group coordination. The basic functions of these three modules are outlined as follows:

- *Resource Monitor* is composed of (i) an information collector that seeks spectral opportunities in time, frequency and space domains, and (ii) a resource manager that updates and characterizes resource information obtained by the information collector. The information collector scans the wireless spectrum regularly, and relies on a dissemination protocol to exchange the information of discovered spectral opportunities among the devices in the same spectral-agile communication group. The information collected by the information collector is used by the resource manager to update a so-called "spectral opportunity map" (SOM). The SOM is basically a database that stores all spectral opportunity information. The resource manager also updates its SOM when receiving the opportunity information update from other devices — via the dissemination protocol — so that information collector in each devices may not need to per-

form the scans very frequently. With the helps of the information collector and the resource manager, each device in the same spectral-agile communication group can keep track of available spectral opportunities and utilize them on a real-time basis.

- *Resource-use decision maker* determines when and how a device should use which channel(s) so as to maximize the utilization of spectral opportunities. The decision maker must make these decisions when (I) informed by the resource monitor that new spectral opportunities are discovered, (II) detecting the presence of primary/licensed devices on the current channel(s), and (III) detecting the presence of other spectral-agile devices. For case (I), the decision maker may decide to use multiple idle channels if the physical layer supports some modulation scheme that can occupy multiple spectral bands. For case (II), the decision maker has no choice but to select an idle channel for the SOM, if possible, and switch to the selected channel. For case (III), the decision maker may also decide to switch to other idle channel(s) so as to maximize the over spectral utilization, or simply decides to stay in the current channel.
- *Resource Coordinator* takes charge of 3 spectrum-access controls to coordinate the use of spectral opportunities among devices and among spectral-agile communication groups. The three controls are (1) intra-group synchronization control, (2) "listen-before-talk" medium access control, and (3) inter-group coordination control. By using (1), the decision makers of individual devices in a spectral-agile communicating group can synchronize with each other to make an unanimous decision on how to use the spectral opportunity, so as to maintain the intra-group connectivity. By using (2) different devices or spectral-agile groups can share the same channel in a distributed manner without interfering with the primary devices. By using (3) different spectral-agile communication groups can utilize the wireless spectrum cooperatively, instead of competing with each other, so as to achieve higher spectral utilization.

In what follows, we describe the detailed operations and algorithms used by these three modules to fulfill the aforementioned tasks. These operations and algorithms

are later implemented in the IEEE 802.11 wireless stations in Section 5.4 for the ns-2 based simulation.

5.3.1 Resource Monitor

The first task to enable the opportunistic use of the wireless spectrum is to discover the potential spectral opportunities. In order to do so, the information collector in each device must scan the wireless spectrum on a regular basis. For each scan, the information collector randomly selects a channel (except the channel the device is occupying and the channel scanned at the last scan) and listens to the selected channel for *SCANNING_INTERVAL* seconds. Since the information collector in each device scans the spectrum independently, it is possible that two devices always scan around the same time (and then scan the same channel occasionally) if each information use the same scanning period. This may cause some problems for disseminating the spectral opportunities as we will explain shortly. Therefore, the information collector should select the scanning period, which is the time interval between two consecutive scans, randomly and uniformly between

$$[0.9 * \text{SCANNING\_PERIOD}, 1.1 * \text{SCANNING\_PERIOD}], \tag{5.14}$$

where *SCANNING_PERIOD* is the average scanning period. By doing so, we can minimize the concurrent scans without using a centralized (scanning) coordination.

There are several special situations that an information collector should cancel a due scan. First, when an device has detected any activity of the primary on the current channel, the information collector in that device should cancel the next scheduled scan. This is because when the primary devices are detected, the decision makers of the devices in a spectral-agile communicating group will invoke the intra-group synchronization control (the details is explained later) to synchronize the vacating (from the current channel). If the information collector performs the channel scanning in the mean time, the device has to leave the current channel and therefore, the synchronization process, which may result in losing connection with other devices permanently. Second, if all devices are synchronized and about to switch to a

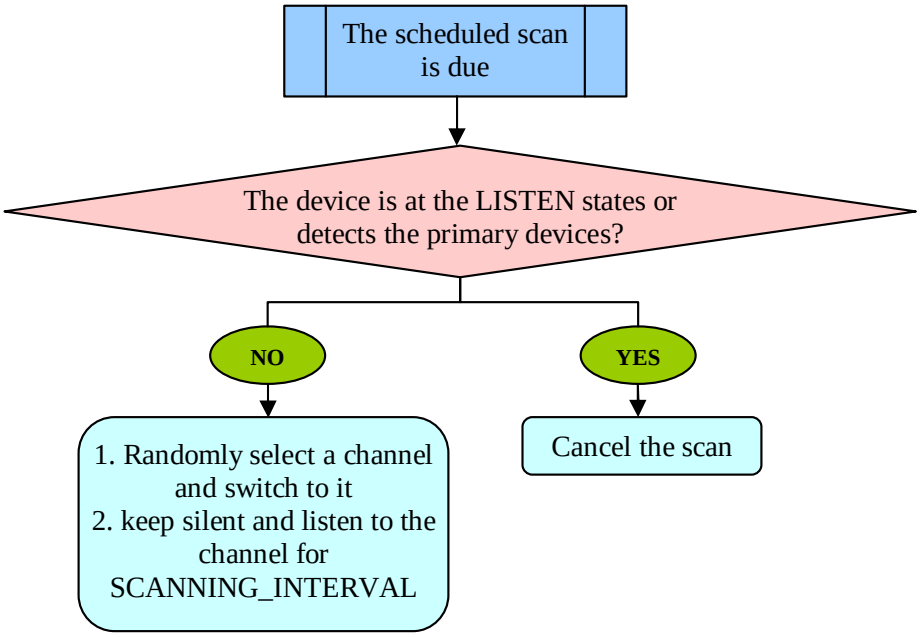


Figure 5.8. Spectral opportunity discovery: before scanning

new channel, any due scan is also cancelled to prevent any disconnection from other devices. Finally, if a device just switches to a new channel and still in the *LISTEN* state,[2], the information collector should also cancel the scan. The scanning procedure described above is illustrated Figure 5.8.

During each channel scan, the information collector records the "activities" detected on the scanned channel. These activities are characterized by several parameters, including the fraction of time that the channel is deemed busy during the scan interval, the average received power and if possible, the activity type (either primary or secondary). These parameters are then used by the resource manager to identify potential spectral opportunities. Upon completion of the scanning, the device switches back to the previous channel and keeps silent for *LISTEN_INTERVAL* second (i.e., the *LISTEN* state) before resuming transmission to make sure the channel is still available. In the meantime, the resource manager updates its SOM— based on the collected parameters and prepares to disseminate the latest opportunity update to the resource mangers of other devices in the same spectral-agile communication

[2]A device must remain in the *LISTEN* state for *LISTEN_INTERVAL* seconds after switching to a new channel to ensure that the new channel is indeed idle and can be used

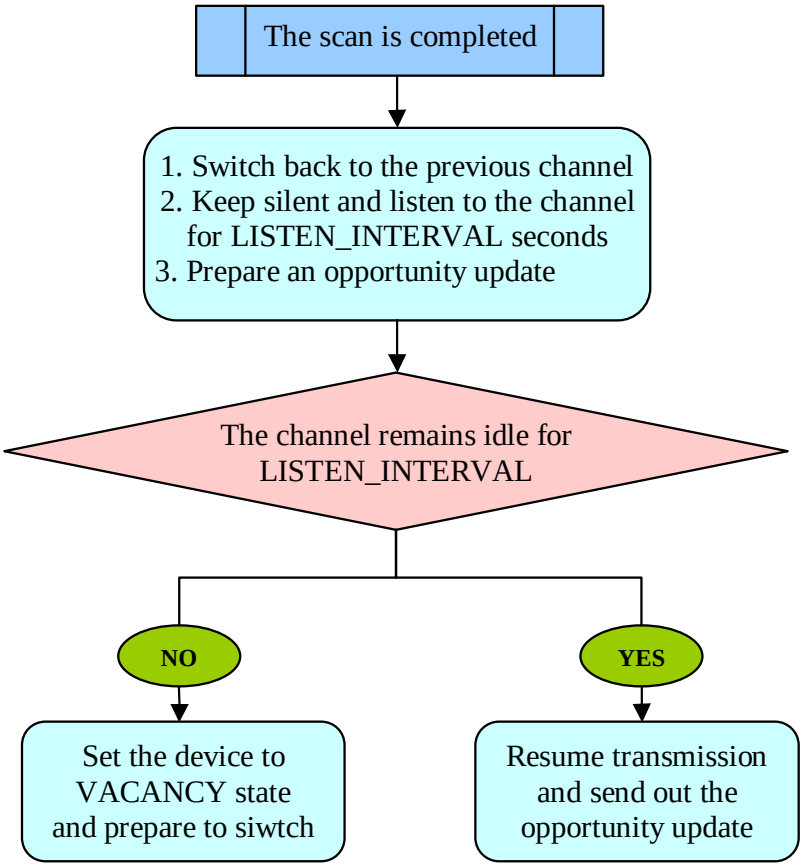


Figure 5.9. Spectral opportunity discovery: after scanning

group. If the current channel remains idle for *LISTEN_INTERVAL* seconds, the resource manager sends out the opportunity update as the normal data frame immediately after the transmission resumes. This post-scanning procedure is illustrated in Figure 5.9.

The resource manager of each device maintains its SOM, which stores the status of all channels in the wireless spectrum. There are two methods to update the SOM: by scanning a channel via the information collector, and by receiving spectral opportunity updates from the other resource managers in the same spectral-agile communication group. As mentioned in the previous subsection, each resource manager disseminates the opportunity update after resuming transmission on the original channel. The information contained in an opportunity update is listed in Figure 5.10-(a), where the "Index" field represents the channel index, the "Duration" field represents the scanning duration, the "P_/S_utilization" field represents the percentage of the

scanning duration when activities from primary/secondary devices are detected, and the last field represents the average detected power of primary devices' transmissions.

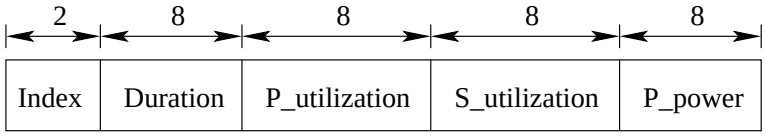
Figure 5.10-(b) shows a possible implementation of the SOM. The "Idle" field indicates if a channel is available or not. For example, a value of 1 means that the channel is idle and considered as a spectral opportunity. This field is set to 0 when the latest spectral opportunity update contains a non-zero P_utilization. The "T_Duration field" represents the accumulative amount of time that has been used for scanning that channel. T_Duration is used to compute the average spectral utilization of primary and secondary devices (i.e., the "avg_P_util" and "avg_S_util" fields in the SOM). The value of avg_P_util is updated by

$$avg\_P\_util = \frac{T\_Duration \cdot avg\_P\_util + Duration \cdot P\_utilization}{T\_Duration + Duration}, \tag{5.15}$$

and so are the values of avg_S_util and avg_P_power. The average specral utilization and average power are useful when multiple idle channels are available, since the statistical information helps a resource-use decision maker choose a "better" idle channels. One should note that the time duration of each potential spectrum opportunity is not included in the SOM simply because it is difficult to predict or estimate such information, given that the primary devices may reclaim the channels at any time. As we will explain in the next subsection, spectral-agile devices uses an idle channel in a reactive way, meaning that spectral-agile devices use a channel until the primary device reclaims that channel. Therefore, the decision maker only needs to know whether or not a channel is available, instead of how long it may last.[3]

It should be noted that different devices in the same spectral-agile communication group may have different SOMs, mainly because a device may miss some opportunity updates sent by the other devices. This could occur if the device switches to another channel for scanning while the other devices are disseminating the opportunity updates. Even though the randomized scanning period helps minimize the loss of opportunity updates, such losses and the resulting "inconsistency" among the SOMs

[3]Of course, any additional information, such as the duration of channel availability, if available, may help a device make a better decision on spectral opportunity use.



(a) Spectral opportunity update

Index	Idle	T_Duration	avg_P_util	avg_S_util	P_power
0	1	10	0.85	0.23	−20 db
1	0	24	0.17	0	−10 db
2	0	N/A	N/A	N/A	N/A
≈	≈	≈	≈	≈	≈
N	1	N/A	N/A	N/A	N/A

(b) Spectral opportunity map

Figure 5.10. Spectral opportunity management (SOM)

cannot be eliminated, Fortunately, our intra-group synchronization control does not require a group-wide, unique SOM to maintain the intra-group connectivity. We will elaborate the intra-group synchronization explained in the next subsections.

5.3.2 Resource-use Decision Maker

The task of the resource-use decision maker is relatively simple thanks to the fact that each spectral-agile device uses at most one channel (at any give time) in a reactive manner. That is, a spectral-agile device vacates only when detecting the presence of primary devices on the current channel(s). Therefore, what a decision maker needs to do is only to choose a "good" idle channel according to the SOM. For simplicity, the decision maker selects an idle channel with the smallest value of "avg_P_util" in our current implementation. If two idle channels have a similar "avg_P_util", the channel with a smaller value of "avg _S_util" is selected. It should be noted that when a spectral-agile device is allowed to use multiple idle channels simultaneously, the decision-making process becomes much complicated since the decision maker has to decide how many and which channels the device (and therefore, the entire spectral-agile communication group) should use so as to optimize not only the individual

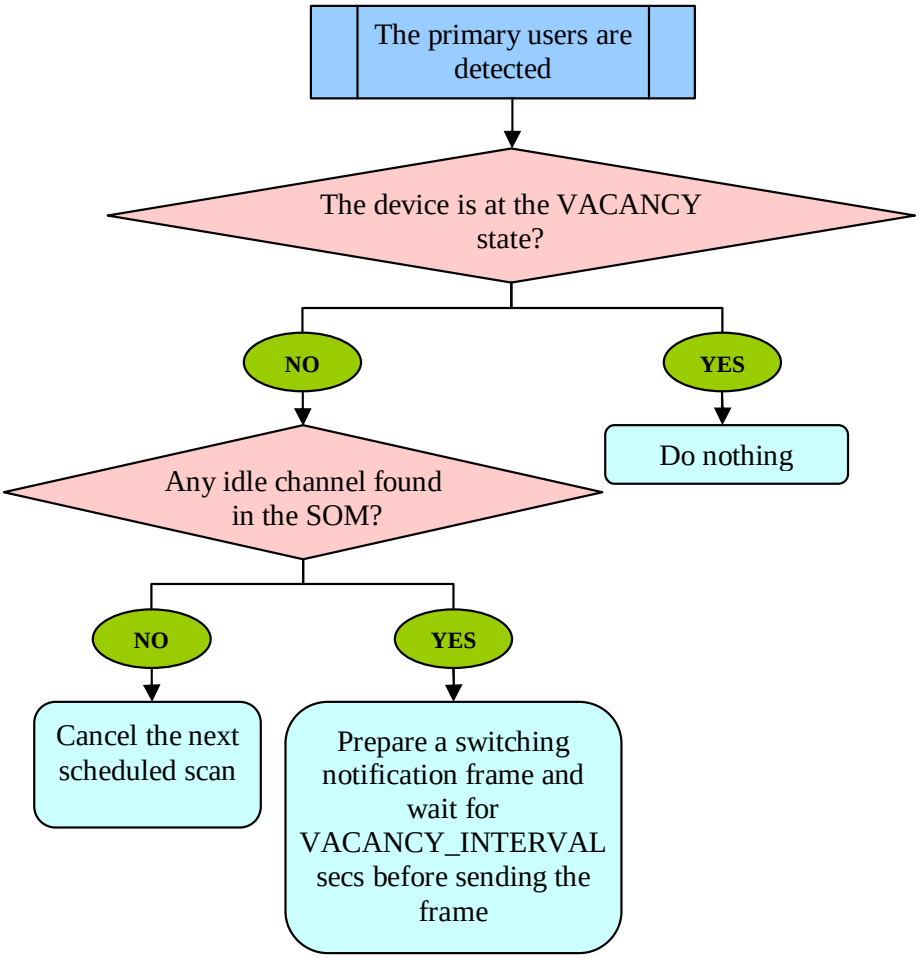


Figure 5.11. Spectral opportunity use: preparation for vacating a channel

device's performance but also the overall spectrum efficiency. We will discuss this issue in the next chapter.

5.3.3 Resource Coordinator

To enable a distributed and cooperative use of the wireless spectrum, the resource coordinators must (1) synchronize the device's use of spectral opportunities so that devices belonging to the same spectral-agile communication group always maintain the inter-connectivity, and (2) coordinate the use of spectral opportunities among different spectral-agile communication group so as to resolve any potential conflict/contention in utilizing these opportunities. The former is referred to as the intra-group synchronization and the latter is referred to as the inter-group coordination.

Intra-group Synchronization

The most challenging task to realize a spectral-agile communication group is to maintain the connectivity among the devices. For example, if some devices decide to switch to channel i while the others decide to switch to channel j, these devices will lose the communication between them. Figure 5.11 depicts the procedure to synchronize the spectral-agile devices's channel switching when their communication group is forced to vacate the current channel. Upon detecting the activities of primary devices, the device enters a so-called *VACANCY* state, and the decision maker searches their own SOM for any available idle channel. If there is no any idle channel found, the device remains in the *VACANCY* state and waits to see if other devices in the same spectral-agile communication group finds some idle channels in their SOM. During the waiting period, the resource monitor cancels the next scheduled scan as explained earlier. If any idle channels is found in the SOM, the decision maker selects an idle channel as the target channel (to switch) and passes the selection to the resource coordinator. The resource coordinator includes the index of the target channel in a so-called *switch notification* frame and waits for *VACANCY_INTERVAL* seconds before sending out this frame. The purpose of the addition "backoff" is to ensure that the other devices which have left the current channel for scanning have enough time to complete the scans, switch back to the current channel, and receive the switching notification frame. This can be achieved by setting the values of *VACANCY_INTERVAL*, *SCANNING_INTERVAL*, and *LISTEN_INTERVAL*

$$VACANCY\_INTERVAL > SCANNING\_INTERVAL + LISTEN\_INTERVAL. \quad (5.16)$$

Once the backoff expires, the resource coordinator sends out the switching notification frame to other devices immediately.

Since the devices may send out their own switching notification frames to each other at the same time, the devices may end up with switching to different channels if the received notification frames indicate different target channels. To avoid this problem, the resource coordinator in each device waits for an additional (and different) transmission offset after the backoff expires. The transmission offsets can be randomly

selected each time or set as a fix value. Once the resource coordinator with a pending transmission of a switch notification frame receives a switch notification frame from the other devices, the resource coordinator must cancel the transmission of its own notification frame. As a result, only one unique switch notification is disseminated and received by all devices in that spectral-agile communication group. The procedure is depicted in Figure 5.12.

Note that it is always possible that some devices may miss a switch notification frame due to transmission errors. Therefore, there is no absolute guarantee for a synchronized channel switching even if other sophisticated retransmission and handshaking mechanisms are applied. A possible solution is to establish a group-wide, unified SOM so that, whenever a spectral-agile communication group needs to vacate a channel, all devices in this group can choose the same channel without requiring the need to notify each other. By doing so, the difficulty shifts from securely disseminating a switch notification packet to securely disseminating *all* spectral opportunity updates. Since sending the opportunity updates to update the SOM is more frequent than sending a switch notification frame, our current implementation should be more reliable. In any case, all devices may either switch back to the previous communicating channel or a pre-defined channel for re-synchronization, when perceiving the existence of a missing device (from the same spectral-agile communication group) after switching to a new channel.

Inter-group Coordination

To make different spectral-agile communication groups utilize the spectral opportunity in a cooperative fashion, we need (i) a multiple access control so that different groups can fairly share the spectrum, and (ii) a "utilization-maximizing" mechanism so that each spectral-agile communication group can utilize a different opportunity, if multiple opportunities exist. The first goal can be easily achieved by using the IEEE 802.11 standard-like carrier-sense-multiple-access/collision avoidance (CSMA/CA) with exponential random backoff. To achieve the second goal, we propose a distributed spectral-sharing etiquette. When a spectral-agile com-

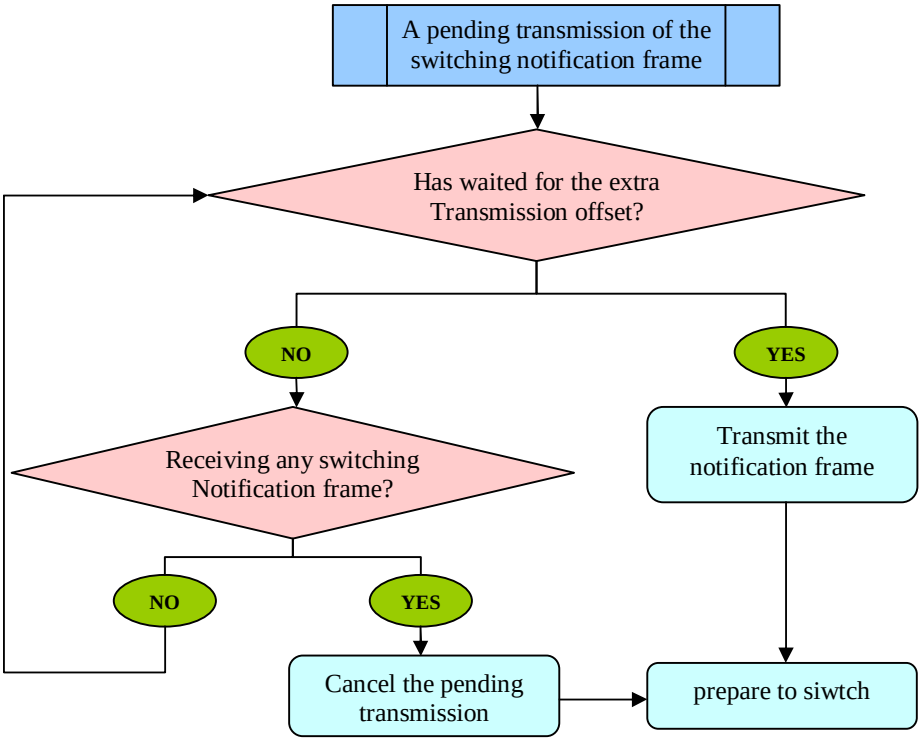


Figure 5.12. Spectral opportunity use: dissemination of a switching notification

munication group detects the presence of any other spectral-agile communication group, the decision makers of all devices in that spectral-agile communication group should immediately check their own SOM for idle channels. If any idle channel other than the currently-occupied channel is found, the resource coordinators follow the intra-group synchronization procedure in the previous subsection. To prevent all involved spectral-agile communication groups from renouncing the currently-occupied channel, the resource coordinators in the same spectral-agile communication groups waits for an additional delay, before actually staring the intra-group synchronization. If a spectral-agile communication group does renounce the current channel, those spectral-agile communication groups that have not vacated yet will cancel their intra-group synchronization procedure, after perceiving the absence of that leaving spectral-agile communication group. These "staying" spectral-agile communication groups may update the channel status in their SOMs and repeat the above procedure, if they are able to locate other idle channels. This way, we can maximize the resource utilization in a distributed manner.

5.4 Evaluation

The three basic components of a spectral-agile device/group in Section 5.3 are implemented in *ns-2* so that we can evaluate the performance (as compared to analytical upper bounds) and the effects of overhead associated with spectral agility. We use the IEEE 802.11 standard as the MAC-layer protocol for spectral-agile secondary devices. The primary devices also use the IEEE 802.11 MAC protocol but they have exclusive access to their designated channels. If an IEEE 802.11 "secondary" device in a spectral-agile communication group detects any activity of an IEEE 802.11 "primary" device, the secondary devices suspend any transmission as explained before.

We assume that there there are two primary devices on each channel, one sender and one receiver. The sender has an *ns-2* ON/OFF traffic generator and transmits packets to the receiver. The average channel load generated by the sender is determined by the mean values of ON- and OFF-periods. We assume that there are 3 spectral-agile devices in a spectral-agile communication group. To fully utilize spectral opportunities, we use the *ns-2* constant-bit-rate (CBR) traffic generator so that devices in the spectral-agile communication group always have packets to transmit as we assumed in Section 5.2. Finally, we assume that the packet size from all traffic generators is 500 bytes and all devices use 1-Mbps for data transmission. Figure 5.13 shows the simulation setup for the case of three channels (i.e., channels 1, 6 and 11 in the IEEE 802.11b standard) with a single spectral-agile communication group (i.e., a spectral-agile wireless LAN).

As explained in Section 5.3, several parameters are needed to control a spectral-agile device/group, namely, the *SCANNING_PERIOD*, *SCANNING_INTERVAL*, *VACANCY_INTERVAL*, and *LISTEN_INTERVAL*. The value of *SCANNING_PERIOD* determines the frequency of seeking a spectral opportunity map (SOM). Obviously, the smaller a device's *SCANNING_PERIOD*, the more accurate the SOM becomes. However, a small value of *SCANNING_PERIOD* incurs more control overhead (e.g., frequent dissemination of opportunity updates to other devices), and interrupts normal transmission more frequently. The value of *SCANNING_INTERVAL* determines

time granularity of the spectral opportunities that a spectral-agile device can detect. If the duration of a spectral opportunity is less than *SCANNING_INTERVAL*, a spectral-agile device cannot detect the existence of such a spectral opportunity because the scanned channel becomes "busy" before the scanning is completed. However, choosing too small a *SCANNING_INTERVAL* value is not a good idea either, simply because not enough "activities" will be collected. The same criteria can be applied to choose the value of *LISTEN_INTERVAL* since choosing too small or too large a value results in either interfering primary devices (because of resuming transmission too fast) or wasting a spectral opportunity (because of waiting too long). Finally, we choose the value of *VACANCY_INTERVAL* according to Eq. (5.16).

Based on the transmission rate and packet size chosen above, we let *SCANNING_INTERVAL* = 20 milliseconds, *LISTEN_INTERVAL* = 10 milliseconds, and *VACANCY_INTERVAL*= 40 milliseconds in all of the simulation runs.[4] However, we change the value of *SCANNING_PERIOD* in order to investigate its impact on both performance improvement and control overhead. In the following simulation, we use $N = 3$ as we want to simulate the case of using spectral agility in the current IEEE 802.11b wireless LAN in the 2.4-GHz band. Of course, these mechanisms can be applied to other types of networks and other spectral bands, once the regulatory restriction is removed.

5.4.1 Throughput Improvement for a Single Spectral-agile Communication Group

We choose *SCANNING_PERIOD* = 0.5 second, $T_{on} = 10 * \textit{channel load}$ seconds, and $T_{off} = 10 * (1 - \textit{channel load})$ seconds in this simulation. Figure 5.14 shows the throughput improvement of the spectral-agile communication-group as compared to the case of no spectral agility. Here, we use throughput as the performance metric since the MAC protocol (i.e., the IEEE 802.11b standard) is specified. We consider both homogeneous and heterogeneous loads, and the simulation results are compared

[4]It should be noted that we only focus on the case when the average duration of a spectral opportunity is in the order of seconds.

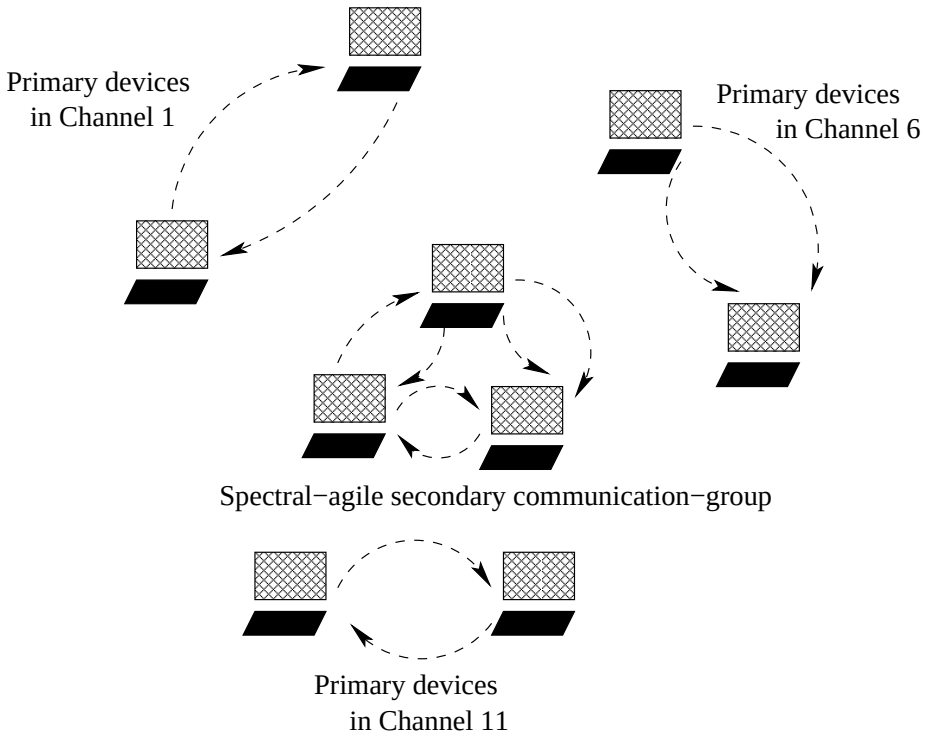


Figure 5.13. Simulation setup for single spectral-agile communication-group: $N = 3$ and $M = 1$

with the analytical results (in solid lines). The improvement obtained from the simulation is shown to be very close to the analytical upper bound in some cases, especially when the average channel load ranges between 0.3 and 0.6. Within this region, the improvement ranges between 40 and 80% for homogeneous loads, and ranges between 50 and 90% for heterogeneous loads. Considering the control overhead incurred by spectral agility, the results verify the effectiveness of our implementation.

One interesting observation is that the improvement is much less than the analytical results as the channel load increases, and using spectral agility is even worse (-22%) than without using spectral agility when the channel is extremely busy. The main reason for this is that when the channel is heavily-loaded, the spectral-agile communication-group has few spectral opportunities. The scanning, listening, and switching simply interrupt the devices' normal transmission without finding many opportunities. Under this circumstance, staying with a fixed channel is better. That is, one should not use spectral agility in extremely busy spectral bands in the first place.

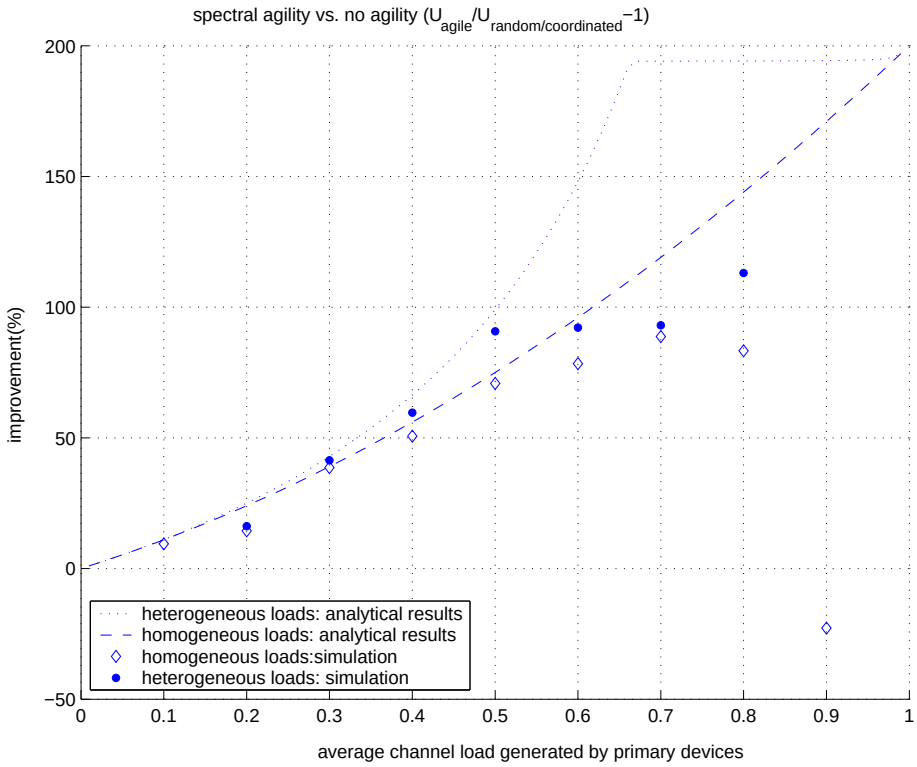


Figure 5.14. A single spectral-agile communication-group: spectral agility vs. no agility with random/coordinated channel selection. *The substantial discrepancy between the analytical and simulation results when the channel load approaches 1 results from that our analytical model does not consider any scanning/control overhead. However, these overheads easily consume the minuscule channel accessing time (as shown in Figure 5.4) gained by spectral agility when the load is close to 1.

Figure 5.14 also confirms that when the loads of the channels are diverse, spectral-agile devices achieves better performance as shown in Section 5.2. One can make an extra 10 to 15% improvement since the spectral-agile devices dynamically search for the least-utilized channels and make use of them more efficiently.

5.4.2 Throughput Improvement of Multiple Spectral-agile Communication Groups

The previous simulation shows that the throughput of a single spectral-agile communication-group increased by up to 90%. We now use $N = 3$ and $M = 2$ to investigate how different spectral-agile groups interact with each other when seeking and utilizing spectral opportunities as shown in Figure 5.15. For an illustrative purpose, we only simulate the case of homogeneous channel loads and set *SCANNING_PERIOD*=0.5 second. In order to make these two spectral-agile communication-groups share the

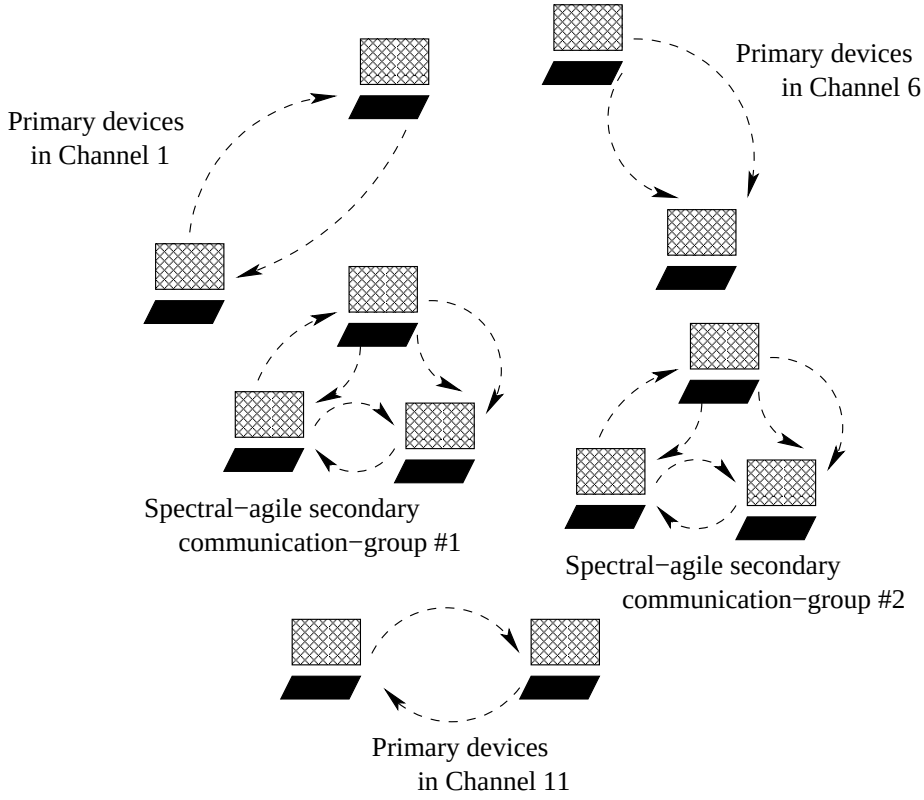


Figure 5.15. Simulation setup for multiple spectral-agile communication-groups: $N = 3$ and $M = 2$

spectral opportunities, instead of letting them compete for these opportunities, we assign different priorities to each spectral-agile communication-group. The priority is used by a spectral-agile communication-group to determine the value of delay in the inter-group coordination algorithm. If a lower-priority spectral-agile group detects the existence of a higher-priority spectral-agile group, the lower-priority group vacates the current channel first ***if and only if*** the SOM indicates that there exist other available spectral opportunities. This way, the lower-priority group is not discriminated in terms of using spectral opportunities. Our simulation results show that these two spectral-agile communication-groups always achieve almost the same throughput.

Figure 5.16 shows the throughput improvement of spectral-agile communication-groups, as compared to the case of using no agility with coordinated channel selection. In general, the improvements are very close to the analytical results (within a 13% margin). One reason why the simulation gives more improvements than the analytical

bound (under moderate channel loads) is that a non-agile secondary communication-group also suspends the transmission for *VACANCY_INTERVAL* seconds before detecting that channel again, if the device/group has detected any activity of the primary device in the assigned channel. For a spectral-agile group, it is less likely to encounter a busy channel because of spectral agility, especially when the channels are moderately-loaded. That is, the overhead of detecting the (channel) idleness in a non-agile secondary device/group is higher than a spectral-agile secondary device/group when the channel is moderately-loaded, and so is the amount of time wasted on waiting. One can also observe that using spectral agility results in poorer performance (-9%) than without using agility, when the channels are heavily-loaded. Again, it does not make any sense to use spectral agility in those heavily-loaded channels as virtually no opportunity exists in those channels. Thus, the overhead easily offsets any improvement made by spectral agility as in the case of a single spectral-agile communication-group.

The simulation results also demonstrate a very important advantage of using spectral agility: *by using spectral agility, we can achieve a higher throughput (more than 30% in many cases, as compared to using no agility with coordinated channel selection, let alone an even higher improvement as compared to using random channel selection) without any off-line planning on spectral resource allocation.* That is, using spectral agility easily achieves the *automated frequency use coordination* and results in a much higher spectral utilization.

5.4.3 Improvements vs. *SCANNING_PERIOD*

We now investigate the effects of *SCANNING_PERIOD* on the throughput improvement of a spectral-agile secondary communication-group. We choose three different loads for the primary devices, 0.2, 0.5 and 0.8, still use $T_{on} = 10 * \textit{channel load}$ seconds and $T_{off} = 10 * (1 - \textit{channel load})$ seconds, and change the value of *SCANNING_PERIOD*. Figure 5.17 shows that for a fixed channel load, the improvement decreases with the increase of *SCANNING_PERIOD*. This is because the less frequently a spectral-agile secondary device scans the spectrum, with a lower probability

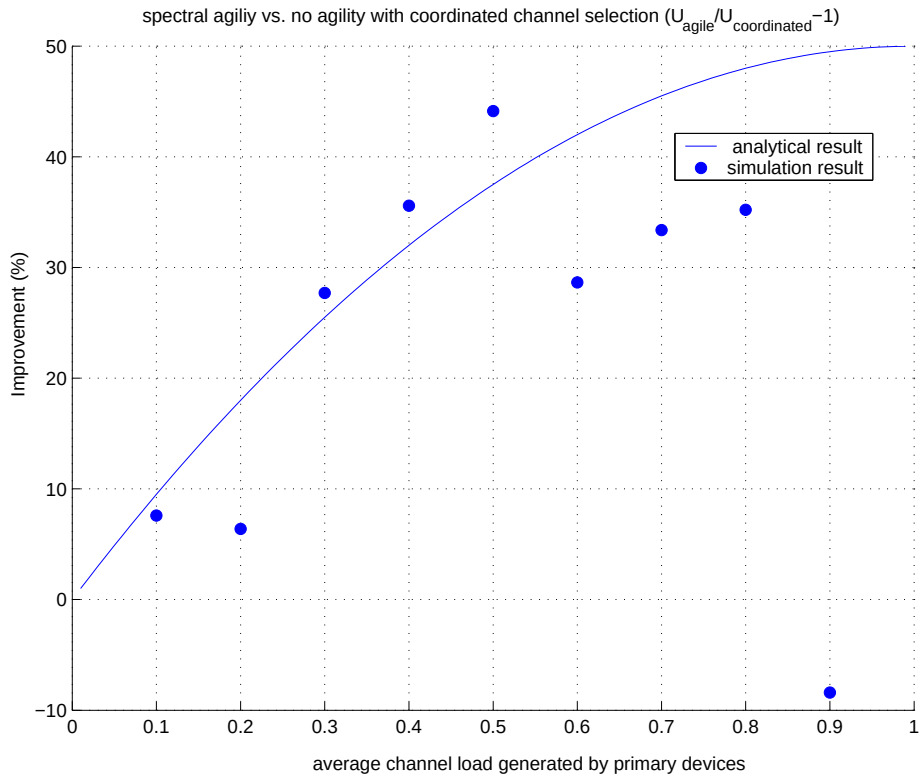


Figure 5.16. Multiple spectral-agile communication-groups: spectral agility vs. no agility with coordinated channel selection. *The substantial discrepancy between the analytical and simulation results when the channel load approaches 1 results from that our analytical model does not consider any scanning/control overhead. However, these overheads easily consume the minuscule channel accessing time (as shown in Figure 5.4) gained by spectral agility when the load is close to 1.

an available channel can be found. Therefore, it is very important for a spectral-agile device/group to choose an appropriate *SCANNING_PERIOD* value since choosing too large a value of *SCANNING_PERIOD* may result in poor performance, especially when the channel is heavily-loaded with the traffic of primary devices. It is when the channel is very busy that a spectral-agile device/group needs spectral opportunities most. Thus, using a large value of *SCANNING_PERIOD* degrades the improvements most when the channel load is high. This explains the decrease of throughput improvement when the load is 0.8.

In fact, one can conclude that the most important control parameter in the spectral-agile device/group is *SCANNING_PERIOD*. A spectral-agile device/group should choose the value of *SCANNING_PERIOD* based on the channel loads, and more importantly, the duration of ON-/OFF-period in each channel. If the channels switch between ON- and OFF-periods very often, a smaller *SCANNING_PERIOD*

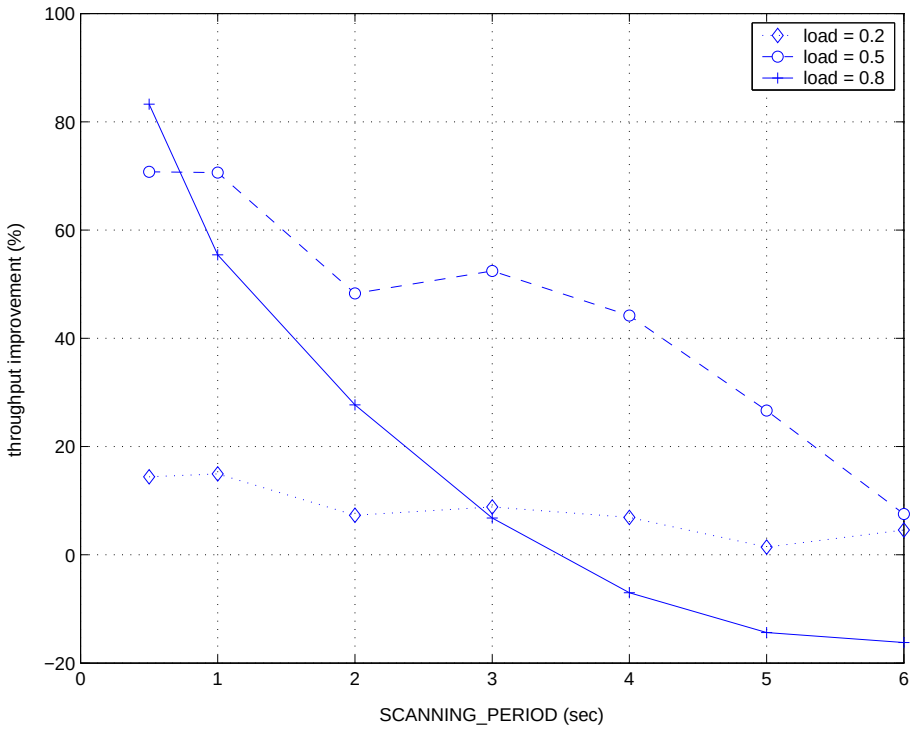


Figure 5.17. Effects of *SCANNING_PERIOD* on the throughput improvement of secondary devices/groups using spectral agility

is required. That is, the degree of agility that a spectral-agile device/group needs, depends on the dynamics of the scanned spectrum. Therefore, using an adaptive *SCANNING_PERIOD* should achieve better performance.

5.4.4 Improvements vs. Duration of a Spectral Opportunity

As discussed above, the throughput improvement of a spectral-agile device/group is determined by *SCANNING_PERIOD* and the average duration of ON-/OFF-periods of primary devices. To be on the safe side, one may choose a very small *SCANNING_PERIOD* in order to exploit the spectral agility. A potential problem with this is that too frequent scanning interrupts too often normal transmission of the spectral-agile devices/groups and also incurs high overhead. We investigate such a trade-off as follows. We choose 3 different values of *SCANNING_PERIOD*. For each *SCANNING_PERIOD* value, we change the T_{on} and T_{off} values but keep the channel load ($= \frac{T_{on}}{T_{on}+T_{off}}$=0.5) unchanged. The total number of packets transmitted (by the spectral-agile devices/group) within a 1000-second interval is plotted in Figure 5.18.

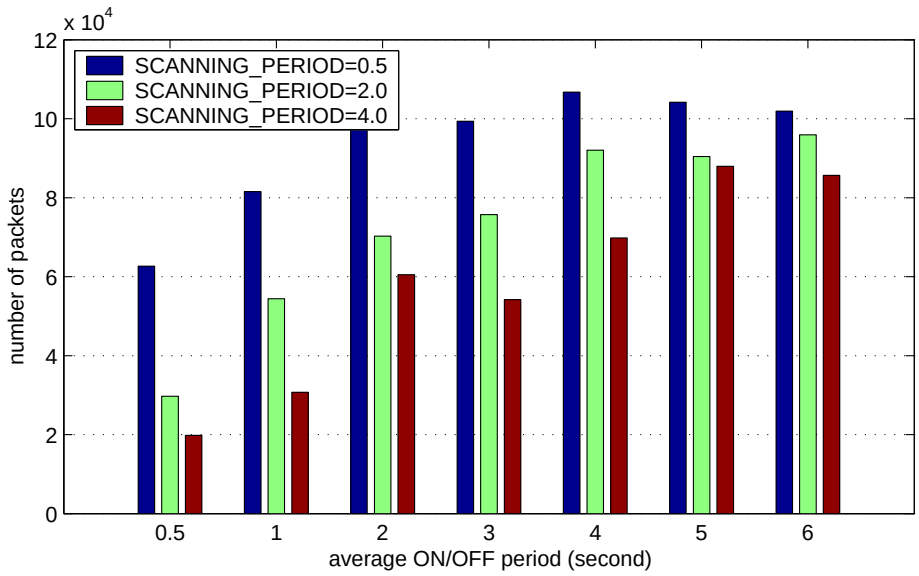


Figure 5.18. Effects of *SCANNING_PERIOD* vs. Effects of average *ON-/OFF-period* on the throughput of secondary devices/groups using spectral agility

For any given value of *SCANNING_PERIOD*, the number of transmitted packets generally increases with the average duration of ON-/OFF-periods (i.e., T_{on} and T_{off}). Of course, a spectral-agile device/group need not scan the channels too frequently when T_{on}/T_{off} is relatively large (compared to *SCANNING_PERIOD*) since the switching also occurs less frequently. This explains the slight decrease for the case of *SCANNING_PERIOD*=0.5 after the average ON-/OFF-periods are larger than 4.0 seconds. However, as compared to using a larger *SCANNING_PERIOD*, using a smaller *SCANNING_PERIOD* always achieves much better performance even though the overhead increases linearly with the scanning frequency. This is because the overhead incurred by scanning is relatively small in our implementation (only *SCANNING_INTERVAL*+*LISTEN_INTERVAL* =0.03 second for every *SCANNING_PERIOD*=0.5 second).

5.5 Conclusion

In this chapter, we investigated the methods of using spectral agility to improve both the efficiency of spectral utilization and the performance of spectral-agile devices. We established a simple mathematical model to analyze the performance gain of using

spectral agility, and provided a performance benchmark by which different implementations of spectral-agile communication can be evaluated. In order to realize the spectral-agile communication, we proposed a comprehensive framework and developed a set of new spectrum access functionalities. These functionalities are added to the IEEE 802.11 wireless LAN in the *ns-2*. The simulation results showed that (1) the throughput of spectral-agile IEEE 802.11 stations can be increased by as high as 90%, (2) such improvement matches the performance benchmark provided by our analytical model, and (3) the improvement is achieved distributively and autonomously with little overhead, and outperforms the improvement of non-agile IEEE 802.11 stations using static coordinated channel selection.

CHAPTER 6

Spectral Agility with Simultaneous Use of Multiple Channels

It has been shown in Chapter 5 that spectral-agile secondary devices/groups can improve their spectral utilization by using spectral agility. Although we assumed that each spectral-agile secondary device can only occupy a single channel at any given time, the improvement is already shown to be very significant. One can expect that if spectral-agile devices are allowed to use all idle channels, the spectral efficiency can be improved further, and so is the secondary device's spectral utilization. However, letting spectral-agile devices/groups use multiple channels can create some new problems. For example, a few aggressive secondary devices/groups may occupy most of the idle channels, hence causing unfair usage of spectral resources. Every secondary device/group may also try to use as many channels as possible, and hence interfere with each other on the shared channels. In order to solve these potential problems, we first investigate the problem of optimal channel allocation and analyze the achievable performance if secondary devices/groups are allowed to use multiple channels. Then, we propose a resource sharing algorithm that not only increases each secondary device/group's resource utilization, but also guarantees fairness among secondary devices/groups. Finally, we provide a framework to integrate the proposed algorithm with the spectral-agile network developed in Chapter 5.

6.1 Optimal Channel Allocation

Assuming that secondary devices are allowed to use multiple channels simultaneously, the first task in developing a resource sharing algorithm is to find the optimal channel

allocation that maximizes the system capacity. Let us assume that N' out of N channels are available, and each channel has a bandwidth of W Hz. If there are M secondary device/groups competing for these channels, the total system capacity C can be obtained by

$$C = \sum_{i=1}^{M} B_i \cdot log_2(1 + \frac{S_i}{N_0 B_i}), \tag{6.1}$$

where $B_i = n_i \cdot W$ is the total bandwidth occupied by secondary device/group i, S_i the transmission power and N_0 the noise power spectral density [75]. Obviously, each feasible allocation should satisfy $\sum_i B_i \leq N'W$. Since the channel capacity function, $B \cdot log_2(1 + \frac{S}{N_0 B})$, is a monotonically increasing function of bandwidth B, one can easily show that the secondary devices/groups should use up all available channels in order to maximize the system capacity. That is, $\sum_i B_i = N'W$.

The problem of finding the optimal channel allocation can then be formulated as

$$\max_{Bi} \; C = \sum_{i=1}^{M} B_i \cdot log_2(1 + \frac{S_i}{N_0 B_i}), \tag{6.2}$$

subject to the constraint

$$\sum_i B_i = N'W. \tag{6.3}$$

By using the Lagrange method, the solution can be obtained by solving the following non-linear system equations

$$\log_2(1 + \frac{S_i}{N_0 B_i}) + \frac{log_2 e}{(1 + \frac{S_i}{N_0 B_i})} \frac{-S_i}{N_0 B_i} + \lambda = 0, \; i = 1, 2, \cdots, M \tag{6.4}$$

where λ is the Lagrange multiplier. The only solution for these non-linear system equations is

$$\frac{S_1}{B_1} = \cdots \frac{S_i}{B_j} = \cdots = \frac{S_M}{B_M}. \tag{6.5}$$

Eq. (6.5) shows that if each secondary device/group obtains an amount of bandwidth proportional to its transmission power, the total system capacity can be maximized. By substituting Eqs. (6.3) and (6.5) into Eq. (6.1), we get the maximum system

capacity as

$$C = \sum_{i=1}^{M} \left(\frac{S_i}{\sum_j S_j} \cdot N'W\right) \cdot log_2\left(1 + \frac{S_i}{N_0\left(\frac{S_i}{\sum_j S_j} \cdot N'W\right)}\right)$$
$$= N'W log_2\left(1 + \frac{\sum_i S_i}{N_0 N'W}\right). \quad (6.6)$$

6.2 The Distributed, Fair Sharing Algorithm

According to Eq. (6.5), the total system capacity is maximized as long as the amount of bandwidth occupied by a secondary device/group is proportional to the transmission power. Therefore, there may exist many possible channel allocations that all maximize the system capacity for given N', M, and $\sum_i S_i$. For example, if $N' = 6$, $M = 3$, and $\sum_i S_i = 0.6$, the allocations $(B_i, S_i) = \{(4, 0.4), (1, 0.1), (1, 0.1)\}$ and $(B_i, S_i) = \{(2, 0.2), (2, 0.2), (2, 0.2)\}$ both maximize the system capacity. However, $(B_i, S_i) = \{(2, 0.2), (2, 0.2), (2, 0.2)\}$ is obviously a better choice because not only the system capacity is maximized but also each secondary device/group obtains an equal share of the idle channels. That is, a good sharing algorithm should be able to (1) ensure that Eq. (6.5) is always satisfied, and (2) guarantee fairness among the secondary devices/groups.

An easiest way to achieve these two objectives is to first distribute the available channels to secondary devices/groups as evenly as possible, and then decide the transmission power according to the resulting bandwidth allocation as well as Eq. (6.5). For example, let us consider the case that 3 spectral-agile secondary communication-groups compete for 5 idle channels as shown in Figure 6.1. Since it is impossible to evenly distribute 5 discrete channels to 3 secondary groups,[1] we can approximate the fair bandwidth sharing by having $(B_1, B_2, B_3) = (2, 1, 2)$ before $t = T_1$, $(B_1, B_2, B_3) = (1, 2, 2)$ before $t = T_2$, and $(B_1, B_2, B_3) = (2, 2, 1)$ after $t = T_3$. By doing so, at least the "long-term" fairness can be maintained.

Unfortunately, there is no central coordinator to allocate idle channels to sec-

[1]Throughout this chapter, we focus on time-division, not code-division, systems. In code-division systems, different groups may occupy the same channels but each is perceived as a noise source to the others.

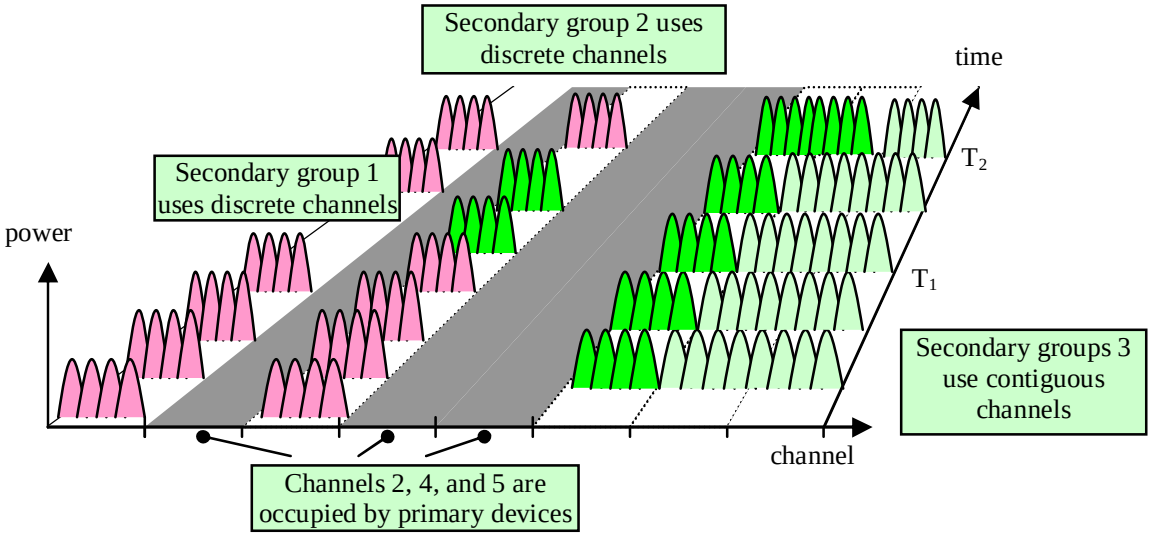


Figure 6.1. Spectral-agile secondary communication-groups use multiple channels: group 1 uses both Channel 1 and Channel , group 2 uses Channel 6, and group 3 uses both Channel 7 and Channel 8.

ondary communication-groups in our distributed, spectral-agile communication. As we discussed in Chapter 5, each secondary device/group scans channels to discover idle channels and utilizes them in a distributed manner. Given that each secondary device/group scans the channel at the same frequency and the channels alternate between ON and OFF states randomly, each secondary device/group should have the same probability to discover a new, idle channel. As long as each secondary device/group occupies idle channels on a "first-discover-first-occupy basis", each device/group should be able to acquire the same share of idle channels in the long run. Based on this observation, we develop our distributed, sharing algorithm as illustrated in Figures 6.2—6.4. Briefly speaking, the left-hand side of Figures 6.2 enforces the first-discover-first-occupy sharing rule, and the right-hand side of Figures 6.2 and 6.3 ensure that a secondary device/group shares an idle channel with others if and only if it is the only idle channel that the secondary device/group can discover. Figure 6.4 ensures that secondary devices/groups vacate the channels once they become busy again.

6.2.1 Theoretical Improvement Ratio

Given that there are N channels in total with an average load $\tau = \frac{T_{off}}{T_{on}+T_{off}}$ on each channel, the total channel time available to all secondary devices/groups is given by

$$N \cdot \frac{T_{off}}{T_{on} + T_{off}}. \tag{6.7}$$

If each secondary device/group fairly shares the total idle channel time given in Eq. (6.7), the average channel occupancy time each secondary device/group can obtain is

$$T_{multiple} = \frac{N}{M} \cdot \frac{T_{off}}{T_{on} + T_{off}}. \tag{6.8}$$

Compared to the case when secondary devices/groups use static channel allocation (i.e., $T_{static} = \frac{T_{off}}{T_{on}+T_{off}}$), the channel occupancy time increases by a factor of $\frac{N}{M}$. Figure 6.5 plots the improvement ratio

$$\frac{T_{multiple}}{T_{static}} \cdot (100\%) \tag{6.9}$$

with different combinations of N and M. As shown in the figure, using spectral agility with simultaneous use of multiple channels always outperforms the case of no agility as long as $N > M$. The improvement ratio can be up to several hundred percents if $M \ll N$.

6.2.2 Improvement Ratio vs. Channel Characteristics

In reality, the channel occupancy time obtained by each secondary device/group is less than that given in Eq. (6.8) because each secondary device/group scans channels at a finite frequency. Therefore, a channel may have become idle for a certain period of time but none of the secondary devices/groups discovers its availability. Obviously, the more frequently a secondary device/group scans the channels, the faster the device/group can discover an idle channel and the less the wasted channel time. Unfortunately, each scan incurs control overhead and interrupts the secondary device/group's normal transmission. If a secondary device/group scans the channels too frequently, the corresponding scanning overhead may offset the improvement.

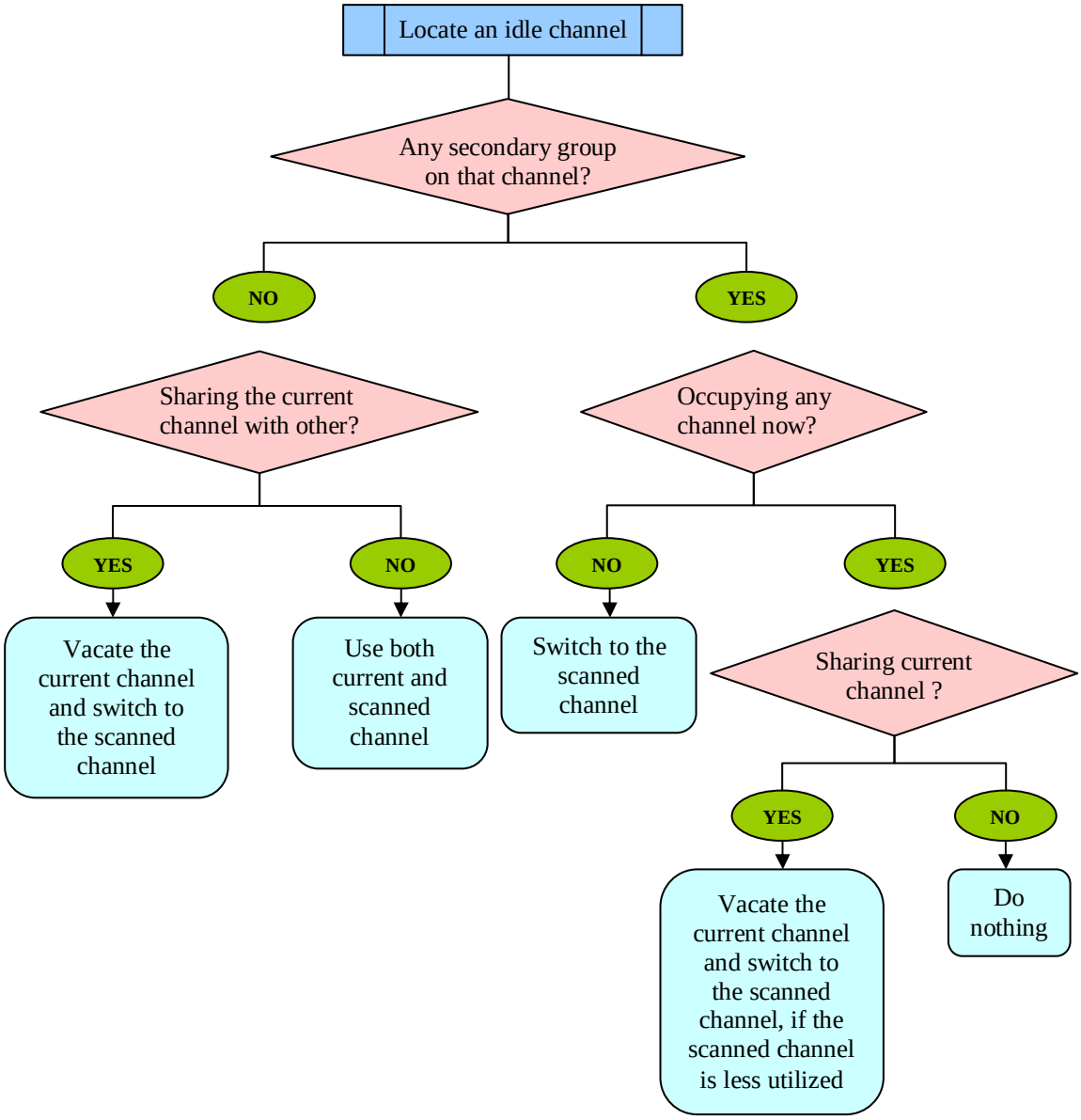


Figure 6.2. The proposed algorithm Part I: Use an idle channel exclusively unless sharing a channel is necessary.

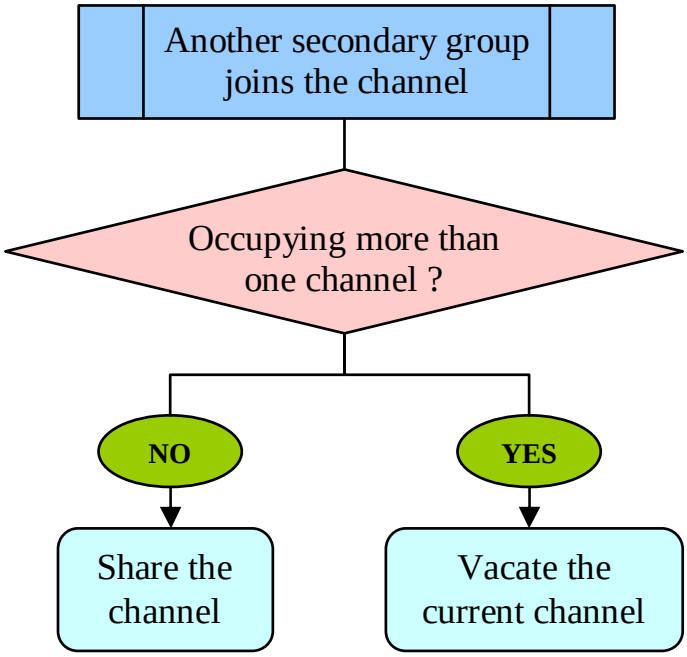


Figure 6.3. The proposed algorithm Part II: Avoid the partial share of currently occupied channels.

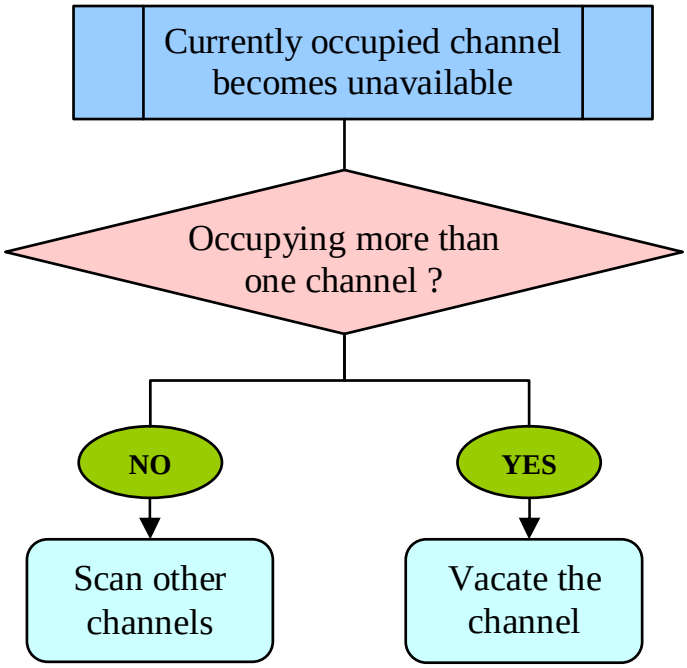


Figure 6.4. The proposed algorithm Part III: Vacate the current channel once the primary devices return to that channel.

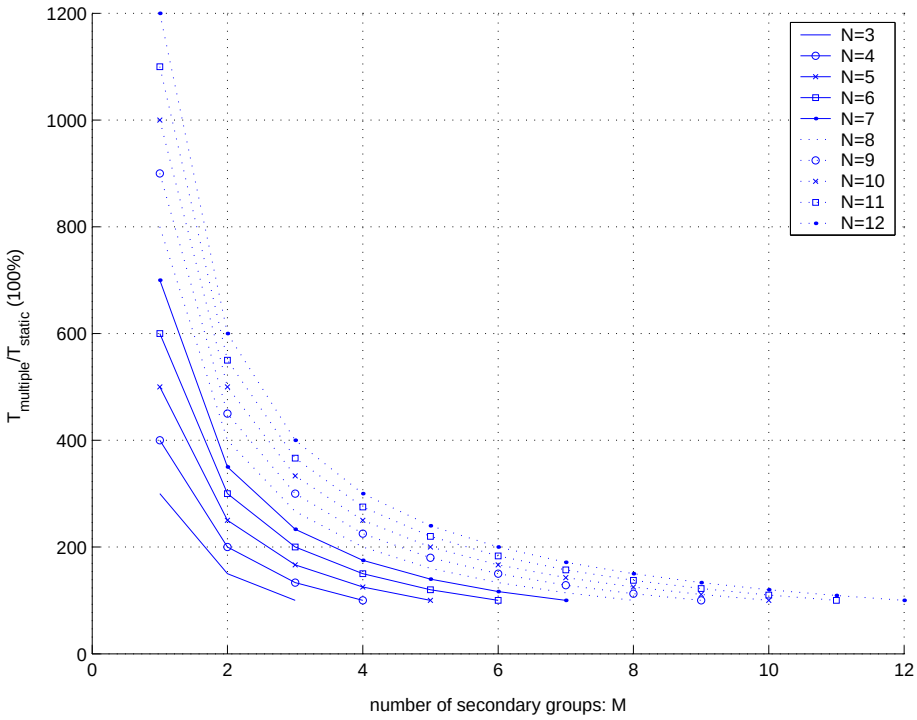


Figure 6.5. The theoretical improvement percentage of the secondary devices/groups' channel accessing time.

One can expect that the optimal scanning frequency depends on the channel characteristics and the scanning overhead. For example, if the channels switch between ON and OFF states frequently, a secondary device/group must scan the channels more aggressively in order to discover the short-lived idle periods before they disappear. Next, we will investigate the effects of the channel characteristics and the scanning frequency on secondary devices/groups' channel utilization.

As illustrated in Figure 6.5, if N is small or $N \approx M$, enabling a secondary device/group to use multiple channels does not make much sense because each device/group can hardly finds an idle channel. In such cases, using spectral agility as in Chapter 5 or even using static channel allocation should suffice. Therefore, we only consider the case when N is larger than M. Figure 6.6 shows the case of $N = 8$ and $M = 3$. In order to investigate the effects of channel characteristics, we vary the channel loads from 0.1 to 0.9, and consider two sets of T_{on} and T_{off} values for each load. We use $T_{on} = 10 * (1 - \tau)$ to represent a fast-varying channel and $T_{on} = 50 * (1 - \tau)$ to represent a slow-varying channel, where τ is the average channel

load. Under these settings, the fast-varying channel alternates its state, on average, 5 times more frequently than the slow-varying channel. One can observe that if $\tau \leq 0.7$, the actual improvement ratio is more than 210% and 230% on fast- and slow-varying channels, respectively, and are quite close to the theoretical improvement of 266% (i.e., the dotted line in the figure). The improvement on a fast-varying channel is less than that on a slow-varying channel mainly because it is more difficult for secondary devices/groups to discover the short-lived idle periods when the channel varies very fast. When the channel load becomes heavier, the secondary devices/groups are more unlikely to discover an idle channel and may switch among different channels frequently. This explains a smaller improvement ratio as compared to the theoretical value for large τ. For example, the improvement ratios are 147% and 188% on fast- and slow-varying channels, respectively, when the average channel load approaches 0.9.

6.2.3 Scanning Frequency vs. Improvement Ratio

As mentioned earlier, one way to increase the channel utilization on fast-varying channels is to reduce the scanning frequency so that secondary device/groups can "capture" short-lived idle periods. We apply this approach on fast-varying channels (i.e., $T_{on} = 10 * \tau$) because of its poorer performance shown in Figure 6.6. The scanning frequency is increased form 0.5 to 10 for the channel loads of 0.9, 0.5 and 0.1. Figure 6.7 shows that by increasing the scanning frequency, we can indeed increase the secondary device/group's channel utilization. For example, when $\tau = 0.9$, the improvement ratio increases from 144% in Figure 6.6 to 182% in Figure 6.7, where the scanning frequency of 4 is used. One can also observe that increasing the scanning frequency is more effective on heavily-loaded channels than on lightly-loaded channels because there exist even less short-lived idle periods on heavily-loaded channels. Therefore, using a higher scanning frequency helps secondary devices/groups greatly to discover the idle periods. For example, the improvement ratio doubles (from 67 % to 144%) if we increase the scanning frequency from 0.5 to 4 for $\tau = 0.9$ but only increases from 201% to 231% for $\tau = 0.9$. However, using too large a scanning fre-

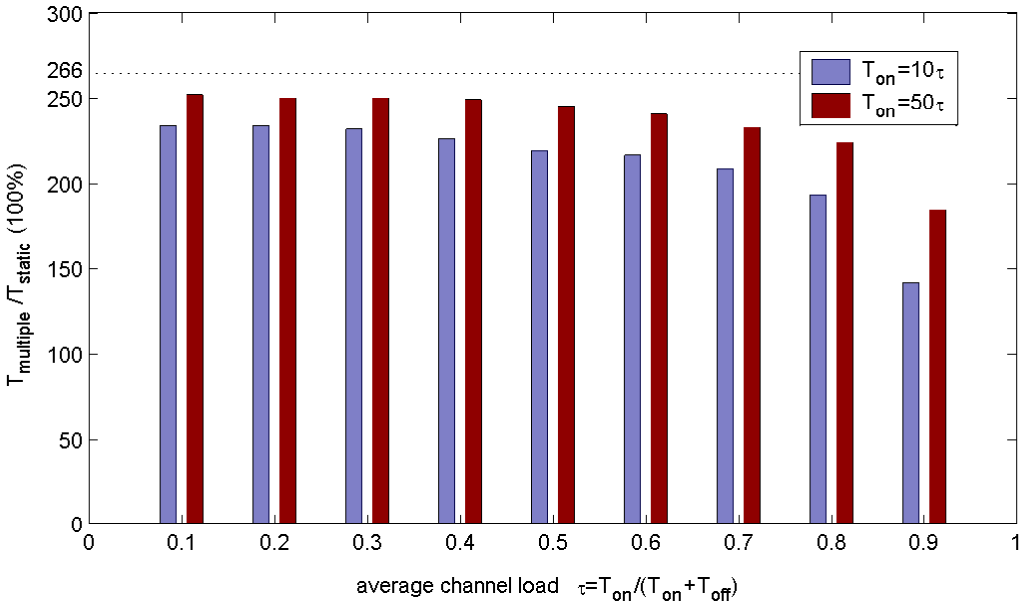


Figure 6.6. The improvement of secondary devices/groups' channel occupancy time achieved by the proposed algorithm under various channel loads and channel dynamics: $N = 8$ and $M = 3$.

quency could also degrade the improvement ratio since the secondary devices/groups spend too much time on scanning, and hence waste channel accessing time. One can see that there exists an optimal scanning frequency that maximizes the secondary device/group's channel utilization. In this particular example, the optimal scanning frequency is 4 for all channel loads.

The relation between the channel utilization and scanning frequency can be analyzed as follows. Assume that each secondary group scans the channels f_{gscan} times every second. Given that there are M secondary groups and N channels, each channel is scanned, on average, by one of M secondary groups $f_{cscan}(= \frac{M \cdot f_{gscan}}{N})$ times per second. Since an idle period cannot be utilized until it is scanned by at least one of the secondary devices/groups, the amount of wasted channel time can be derived as

$$r_{wasted} = \int_0^{T_c} \frac{1}{T_c} \left[\int_0^{T_c - t_1} t_2 f(t_2) dt_2 + \int_{T_c - t_1}^{\infty} (T_c - t_1) f(t_2) dt_2 \right] dt_1, \tag{6.10}$$

where $T_c = \frac{1}{f_{cscan}}$ and $f(t)$ is the probability density function of an idle period. The idea behind this derivation is illustrated in Figure 6.8. The first term in Eq. (6.10) represents the case when an idle period ends before any secondary device/group has a chance to discover it. Therefore, the entire idle period is wasted. The second

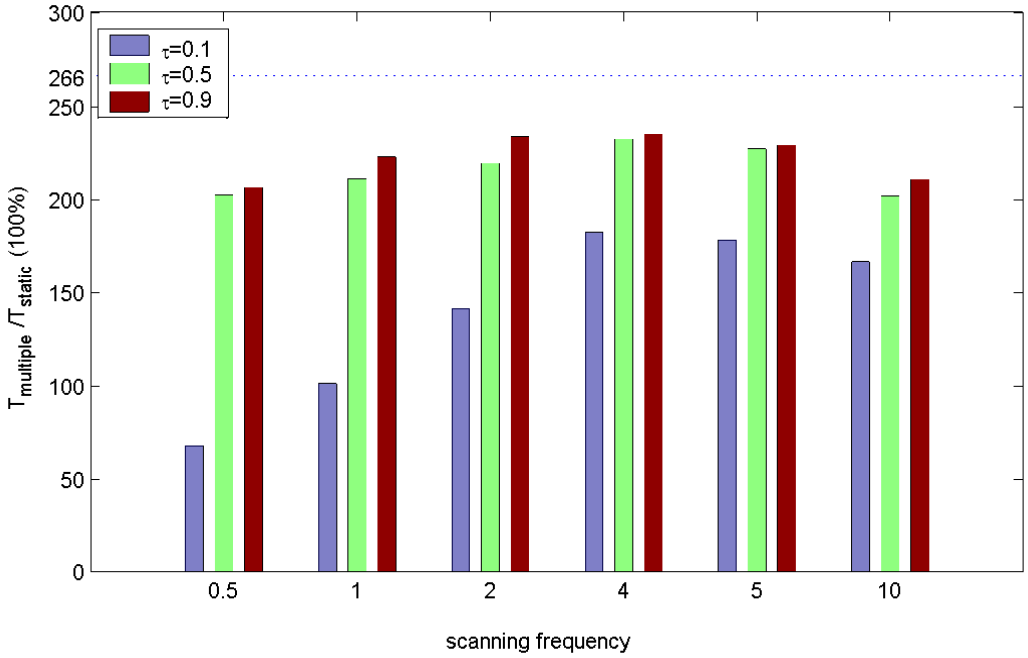


Figure 6.7. The improvement of secondary devices/groups' channel occupancy time achieved by the proposed algorithm for different scanning frequencies on fast-varying channels: $N = 8$, $M = 3$, and $T_{off} = 10 * (1 - \tau)$ for $\tau = 0.1$, 0.5 and 0.9.

term in Eq. (6.10) represents the case when an idle period is discovered by one of the secondary devices/groups so that only a portion of the idle period is wasted. As indicated in Figure 6.8, we assume that the starting time of an idle period is uniformly distributed within two consecutive scans.

The secondary device/group's channel utilization can then be computed as

$$u = 1 - \frac{r_{wasted}}{\int_0^\infty t f(t) dt}. \tag{6.11}$$

If $f(t)$ is an exponential distribution function, we can simplify Eq. (6.11) as

$$u = \frac{1 - e^{-T_{nor}}}{T_{nor}}, \tag{6.12}$$

where $T_{nor} = \frac{T_c}{T_{off}}$ is defined as the normalized scanning period. If $T_{nor} = 0$, the utilization is 1 because there no idle period is wasted if the secondary devices/groups continuously monitor all channels. If $T_{nor} = \infty$, the utilization is 0 because the secondary device/group cannot discover idle channels without scanning. When choosing $f_{gscan} = 4$ (i.e., $T_{nor} = \frac{0.66}{T_{off}}$ given $N = 8$ and $M = 3$), the channel utilizations for the cases of $\tau = 0.1$, 0.5 and 0.9 are 0.73, 0.93 and 0.96, respectively, according to

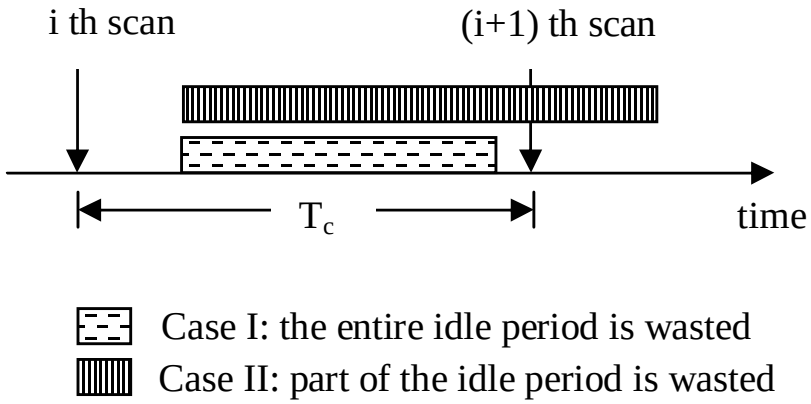


Figure 6.8. The relation between channel utilization and scanning frequency: wasted channel time between two consecutive scans.

Eq. (6.12). Compared to the actual utilizations shown in Figure 6.7 (i.e., 0.744=$\frac{198}{266}$, 0.947=$\frac{252}{266}$ and 0.958=$\frac{255}{266}$ for $\tau = 0.1$, 0.5 and 0.9, respectively), the model provides very accurate estimation.

Using Eq. (6.12), the optimal scanning frequency that maximizes the channel utilization can also be determined. Let the scanning overhead associated with each scan be O_{scan} seconds. The optimal scanning frequency f_{opt} is then the solution that maximizes the utilization function

$$U = \frac{1 - e^{-T_{nor}}}{T_{nor}}(1 - f_{gscan} \cdot O_{scan}), \tag{6.13}$$

where $f_{gscan} \cdot O_{scan}$ is the scanning overhead per unit time, or equivalently, the ratio of time spent on scanning. Since T_{nor} is also a function of the scanning frequency f_{gscan} and can be represented as $T_{nor} = \frac{N}{M \cdot T_{off} \cdot f_{gscan}}$, one can take the derivative of $U(f_{gscan})$ and find the optimal scanning frequency by solving $U'(f_{gscan}) = 0$. Obviously, the optimal scanning frequency is determined by the values of N, M, O_{scan} and T_{off}. The values of M and T_{off} can be estimated by the secondary devices/groups via scanning, and N and O_{scan} are given as operational parameters to the secondary devices/groups.

6.2.4 Fairness vs. Improvement Ratio

Although the proposed algorithm ensures a long-term fair share of idle channels, it is possible that some secondary devices/groups temporarily occupy more channels

than the others, primarily due to the first-discover-first-occupy sharing model. The unfairness may continue until one of the channels changes its state from ON to OFF or vice versa. When the channels have large T_{on} and T_{off}, this may become a serious problem because the channels rarely switch between ON and OFF states. Figure 6.9 shows this potential problem for the case of $N = 8$ and $M = 3$. We assume that T_{on}=15 seconds and T_{off}=45 seconds in each channel, which yields an average channel load of 0.3. As shown in this figure, secondary group no.1 only occupies one channel in $[75, 115]$ while secondary groups 2 and 3 occupy 2, 3 or 4 channels, respectively, during the same time interval. A similar situation occurs again in $[430, 510]$ except that this time the "unfair interval" lasts twice longer and secondary group 2 is "mistreated".

Fairness Index

To quantify the potential unfairness, we define a fairness index F as

$$F = lim_{t\to\infty} \frac{\int_0^t [\max_i n_i(t) - \min_j n_j(t)]dt}{t}, \tag{6.14}$$

where $n_i(t)$ is the number of channels occupied by secondary device/group i at time t and $i, j \in \{1, 2, \cdots, M\}$. The fairness index is the time average of the difference — measured by the number of occupied channels — between the most and the least favored secondary device/groups. Ideally, a fairly-shared system should have $F = 0$ (i.e., $n_i(t) = n_j(t)$). In reality, F is greater than 0 because the channels are not infinitely divisible. For example, if three secondary devices/groups contend for 2 idle channels, the best allocation from the perspective of fairness is to place two of these three devices/groups on one idle channel and the third device/group on the other. That is, $n_1(t) = n_2(t) = 0.5$ and $n_3(t) = 1$. The ideal fair allocation with $n_1(t) = n_2(t) = n_3(t) = \frac{2}{3}$ is actually infeasible. Consider another example where three secondary devices/groups contend for 8 idle channels. The best allocation is that each of the first two secondary devices/groups occupies 3 channels and the third device/group occupies the 2 remaining channels. That is, $n_1(t) = n_2(t) = 3$ and $n_3(t) = 2$, instead of $n_1(t) = n_2(t) = n_3(t) = \frac{8}{3}$. By taking this limitation into

account, the minimum achievable fairness can be computed by

$$F_{min} = \sum_{k=1}^{M} \frac{N! \cdot \tau^{N-k}\tau^{k}}{k!(N-k)!}\left(\frac{1}{floor(\frac{M}{k})} - \frac{1}{ceil(\frac{M}{k})}\right) + \sum_{k=M+1}^{N} \frac{N! \cdot \tau^{N-k}\tau^{k}}{k!(N-k)!} \min(1, mod(k, M)). \quad (6.15)$$

The first term in Eq. (6.15) represents the case that there are not enough channels for secondary devices/groups. In this case, each secondary device/group has to share the channel it occupies with other devices/groups. The second term represents the case that each secondary device/group occupies at least one channel. The difference between the numbers of channels occupied by different secondary devices/groups cannot be more than 1, given that idle channels are **always** allocated to secondary devices/groups fairly. For example, we have $F_{min} = 0.65$, given that $N = 8$, $M = 5$ and $\tau = 0.3$. This implies that the difference in the number of occupied channels cannot be less than 0.65 channel.

Fairness Index Achieved by the Proposed Algorithm

In our proposed algorithm, secondary devices/groups rely on the scanning mechanism to discover idle channels and use them on a "first-discover-first-occupy" basis. Therefore, $n_i(t) - n(j)$ could be much greater than 1 and thus, results in a larger fairness index than that given in Eq. (6.15). In fact, we can estimate the fairness index achieved by our algorithm (i.e., no restriction on a secondary device/group's channel occupancy time) as follows. Assuming that there are K channels available at a certain time instant, the average time interval that these K channels (and only these K channels) remain idle can be computed by

$$T(K) = \frac{1}{\frac{K}{T_{off}} + \frac{N-K}{T_{on}}}, \quad (6.16)$$

given that the ON/OFF period of each channel is independently and exponentially distributed with a mean of T_{on}/T_{off}. In the steady state, the probability that there are K channels available at any time instant can be computed by

$$p(K) = \frac{N!}{N!(N-K)!}\tau^{N-K}(1-\tau)^{K}, \quad (6.17)$$

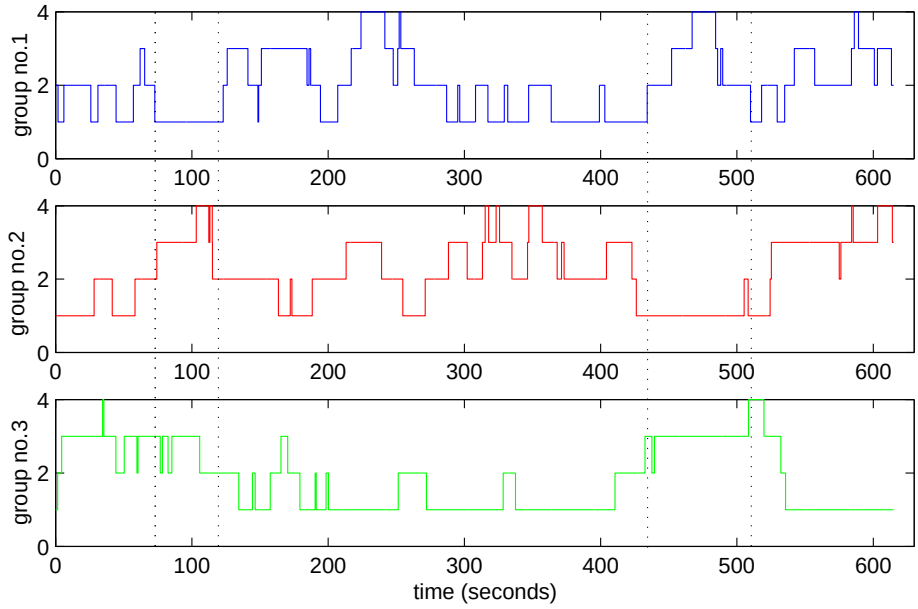


Figure 6.9. The short-term unfairness on slow-varying channels: $N = 8$, $M = 3$, $\tau = 0.3$ and $T_{off} = 50 * (1 - \tau)$.

given that the average load on every channel is τ. The fairness index $F_{proposed}$ can then be obtained by

$$F_{proposed} = \frac{\sum_{K=0}^{N} P(K) \cdot F(M,K) \cdot T(K)}{\sum_{K=0}^{N} p(K) \cdot T(K)}, \tag{6.18}$$

where $F(M,K)$ is the conditional fairness index given that M secondary devices/groups compete for$K$ idle channels. The calculation of $F(M,K)$ involves the operations of permutation/combination and its details are given in the appendix. Take the case of $N = 8$ and $M = 5$ as an example. We have $F(3,0) = 0$, $F(3,1) = 0$, $F(3,2) = 0.5$, $F(3,3) = 0$, $F(3,4) = 1$, $F(3,5) = 1.48$, $F(3,6) = 2.013$, $F(3,7) = 2.271$, and $F(3,8) = 2.567$. Given that $\tau = 0.3$, $T_{on} = 50 * \tau$ and $T_{off} = 50 * (1 - \tau)$, $F_{proposed} = 1.79$. This indicates that although the proposed algorithm exhibits a very good performance in terms of channel utilization, it does not provide fairness since the fairness index is 2.75 times as large as the minimum fairness index $F_{min} = 0.65$

The Enhanced Sharing Algorithm

To improve the fairness of the proposed sharing algorithm, we can either (1) prevent secondary devices/groups from grabbing too many channels in the first place or (2) force secondary device/groups to release the extra channels some time later. Since idle

channels are randomly discovered by secondary devices/groups, the probability that some secondary devices/groups discover much more idle channels than the others is always greater than zero. Moreover, this probability cannot be reduced by increasing the scanning frequency, because the probability to discover an idle channel is equally increased for all secondary devices/groups. This leaves us the only choice — prevent secondary devices/groups from occupying channels for a very long period of time. By doing so, a secondary device/group may still discover and occupy more idle channels than the others, but the secondary device/group has to release those channels after occupying for a predefined amount of time, T_{occupy}. These released channels will then be discovered by other secondary device/groups and be utilized in the same way. The value of T_{occupy} can be derived based on the desired fairness or service requirement but is beyond the scope of this research. We incorporate this restriction mechanism into the previous algorithm, and modify the original operations as follows:

- If a secondary device/group has occupied more than one idle channel, the device/group must enforce the restriction of channel occupancy time on any new idle channel it decides to use according to the original algorithm in Figures 6.2—6.4.
- If a secondary device/group is forced to vacate a channel according to the original algorithm and occupies only one channel thereafter, the device/group must lift the restriction on the remaining channel if restriction has been imposed on that channel earlier.

Based on these new operations, a secondary device/group occupies one channel continuously but voluntarily releases other "extra" channels after occupying them for a certain period of time. By doing so, the short-term fairness can be improved since no secondary device/group occupies multiple channels for a long period of time, even when the channel states remain unchanged. The time granularity of the achievable short-term fairness depends on the value of T_{occupy}. The smaller the value of T_{occupy}, the finer the short-term fairness. However, this enhanced algorithm may cause some degradation of channel utilization because secondary devices/groups may vacate a channel that is still usable. As a result, the idle channel is left unused — after be-

ing released by a secondary device/group — until it is discovered again by other secondary devices/groups.

Tradeoff between Fairness and channel Utilization

Figure 6.10 shows the channel occupancy of 3 secondary groups in the case of $N = 8$. We assume that channels are lightly-loaded ($\tau = 0.3$) and switch between ON and OFF states less frequently ($T_{on} = 50 * \tau$) so that the temporary unfairness may become a serious problem. One can observe that by using the enhanced algorithm, each secondary group occupies a "primary channel" continuously and occupies other idle channels by taking turns with other secondary groups. Therefore, each secondary group cannot exclusively occupy multiple channels. However, the channel occupancy of secondary groups becomes more fractured than the channel occupancy shown in Figure 6.11, where secondary groups use idle channels until they are forced to vacate them. The fractured channel occupancy results in degraded channel utilization which is the price to pay for fairness.

Figure 6.12 shows the improvement ratio $\frac{T_{multiple}}{T_{static}}$ and the fairness index under different 10 T_{occupy}'s, for the slow-varying channels with $T_{off} = 50 * (1 - \tau)$. One can easily observe that by enforcing a strict restriction on secondary groups' channel occupancy time (e.g., T_{occupy}=1 second), the fairness index is very close to the minimum value $F_{min} = 0.65$. However, the improvement ratio of secondary groups' channel utilization drops as low as 185%, compared to the theoretical improvement of 266%. On the other hand, each secondary group has a much larger channel utilization by using a larger T_{occupy} but the fairness index also increases. If we use an infinitely large T_{occupy}, namely no restriction on channel occupancy time, we have the improvement ratio very close to the theoretical value (i.e., 266%) but we also have the largest fairness index 1.71 which is also very close to $F_{proposed} = 1.79$). Thus, there is a tradeoff between the fairness and channel utilization, and the choice of T_{occupy} depends on the service or application requirements.

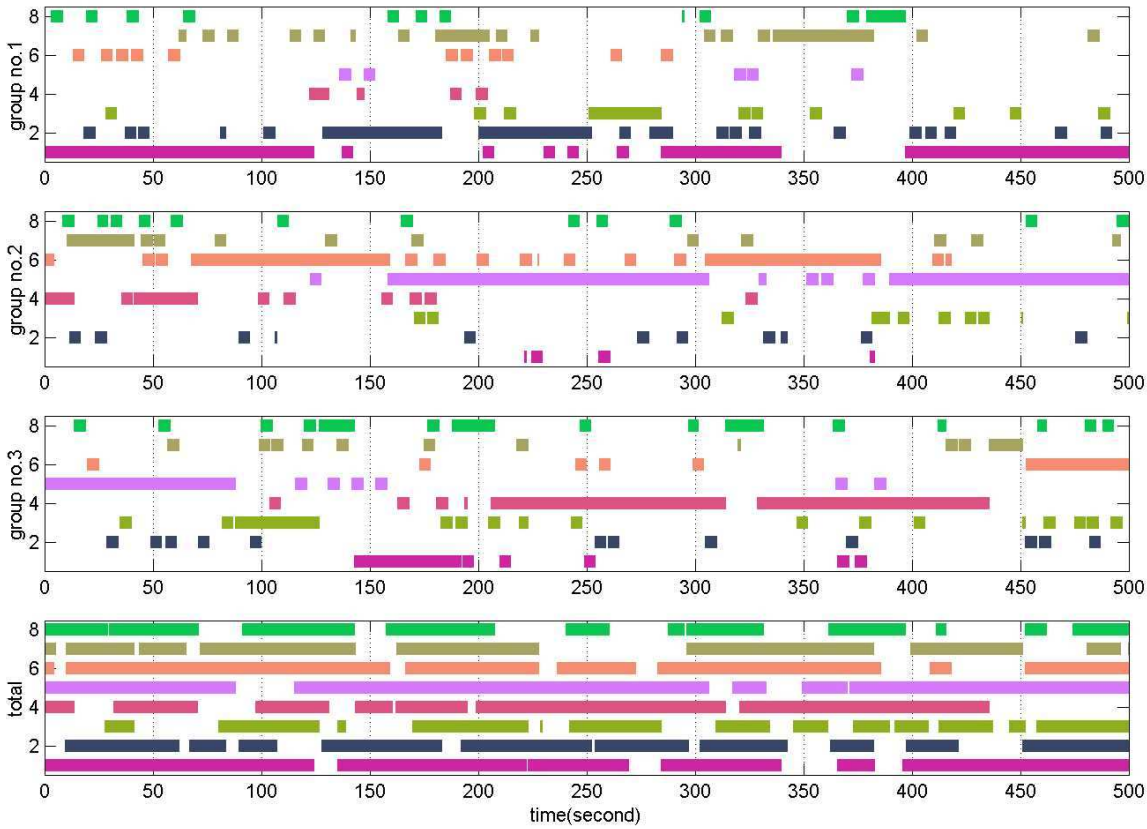


Figure 6.10. Channel occupancy of secondary groups no.1, no.2 and no.3 (from the top) and distribution of available channels (the bottom) — a colored bar represents an idle period: $N = 8$, $M = 3$, $\tau = 0.3$ and $T_{off} = 50 * (1 - \tau)$ with enforcement of restriction on channel occupancy time.

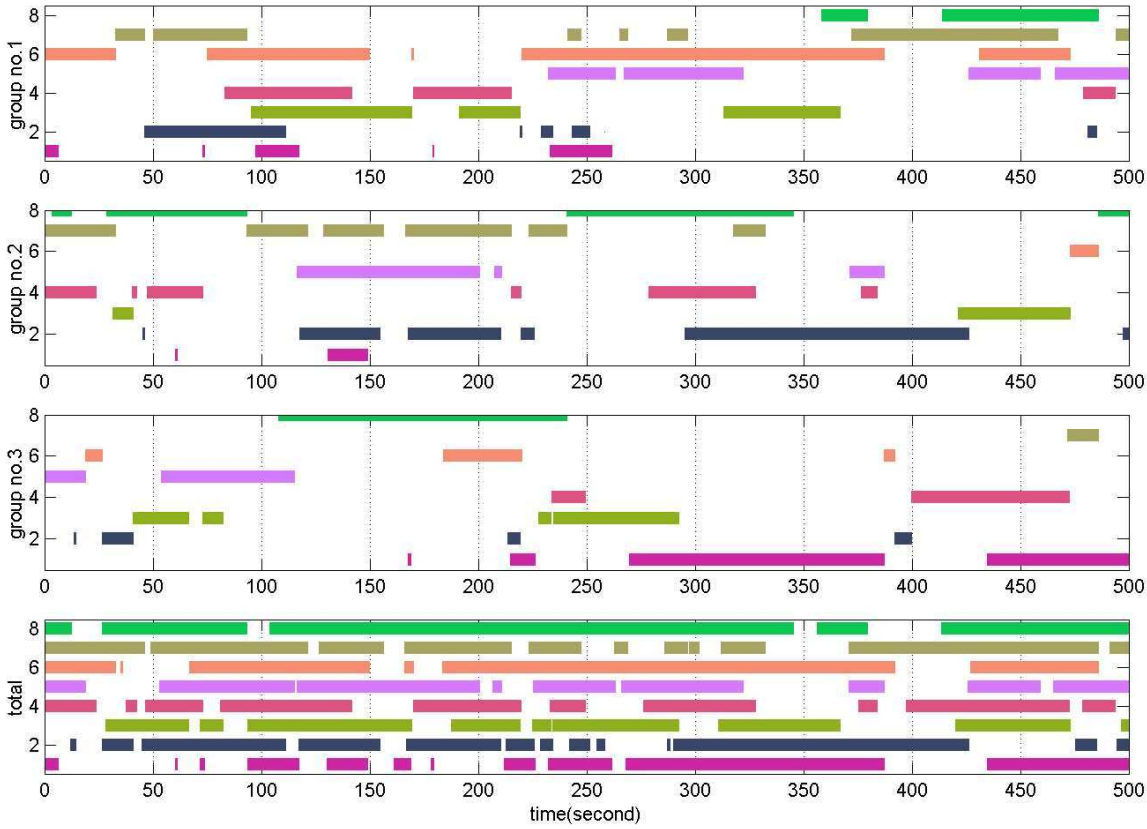


Figure 6.11. Channel occupancy of secondary groups no.1, no.2 and no.3 (from the top) and distribution of available channels (the bottom) — a colored bar represents an idle period: $N = 8$, $M = 3$, $\tau = 0.3$ and $T_{off} = 50 * (1 - \tau)$ without enforcement of restriction on channel occupancy time.

6.3 Cross-band Orthogonal Frequency Division Multiplexing (OFDM)

Since a secondary device/group may simultaneously occupy multiple discrete channels, a modulation scheme that supports effective utilization of multi-channels, such as OFDM, will be needed. OFDM is a modulation technique that uses multiple subcarriers with each being time- and frequency-synchronized so that the subcarriers are orthogonal to each other. By using multiple orthogonal subcarriers, OFDM provides many unique advantages over other modulation techniques. First, the subcarriers can be densely packed without causing inter-carrier interferences, hence making better utilization of spectral resources. Second, the symbol duration in OFDM is larger than that in single-carrier modulation techniques — thanks to the use of multiple subcar-

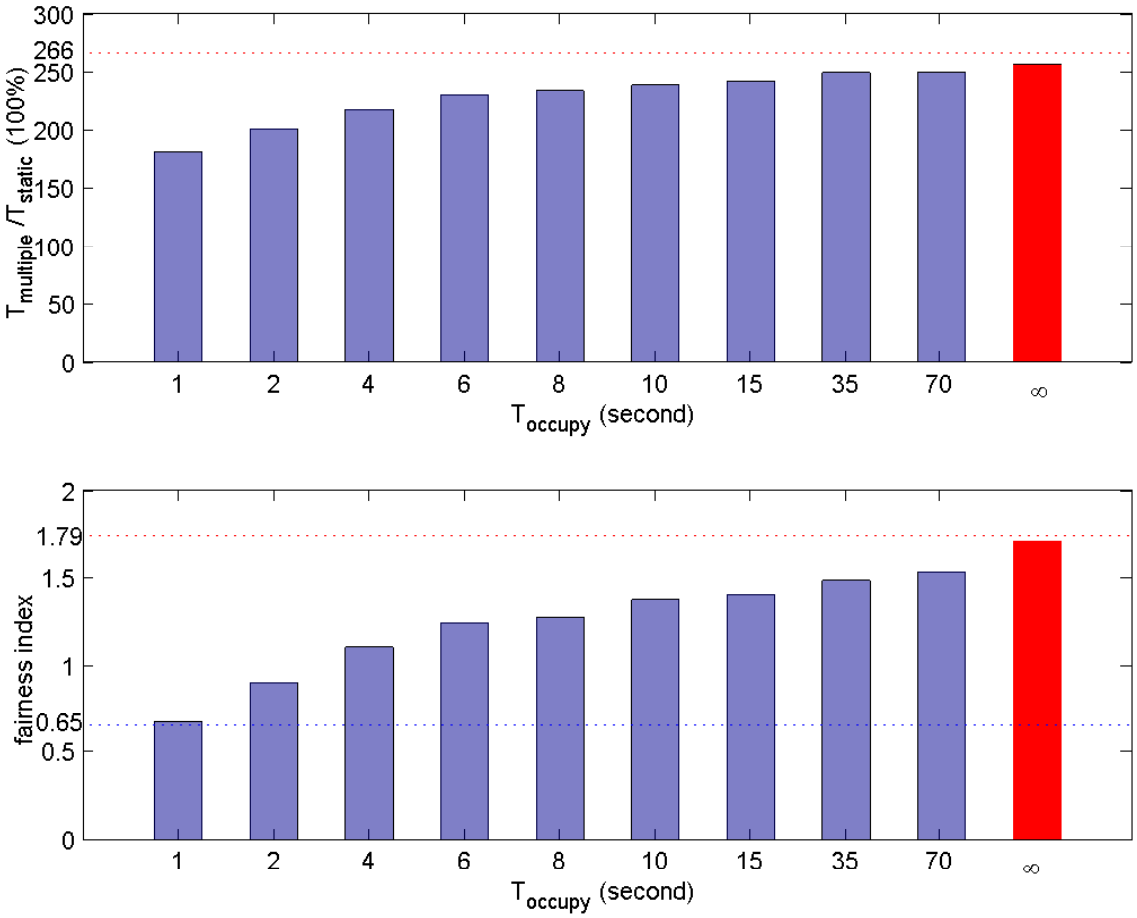


Figure 6.12. Tradeoff between secondary groups' channel occupancy time and the short-term fairness under various values of T_{occupy}: $N = 8$, $M = 3$, $\tau = 0.3$ and $T_{off} = 50(1 - \tau)$.

riers — so that the OFDM symbols are more resistant to inter-symbol interferences. Finally, it is possible to choose desirable subcarriers (from the pool of subcarriers) and modulation schemes on individual subcarriers according to the underlying transmission environment. Such flexibility makes OFDM an attractive option for effective spectral utilization in time-varying wireless networks.

The use of OFDM in our proposed algorithm is also illustrated in Figure 6.1, where we have an 8-channel wireless spectrum with each channel accommodating 4 OFDM subcarriers. As shown in the figure, Channel 2, Channel 4 and Channel 5 are occupied by the primary devices, and thus, are unavailable to the secondary communication-groups. Suppose that based on the proposed algorithm, secondary communication-group 1 will occupy Channel 1 and Channel 3, group 2 will occupy

Channel 6 and group 3 will occupy Channel 7 and 8. Then, secondary group 1 should use OFDM with subcarriers 1$\sim$ 4 and 9$\sim$ 12, secondary group 2 should use OFDM with subcarriers 21$\sim$ 24, and secondary group 3 should use OFDM with subcarriers 25$\sim$ 32. Although secondary groups 1 and 3 both generate an OFDM signal that occupies two channels, the computational overhead for group 1 is larger than group 2, because the modulation/demodulation of an OFDM signal is performed by the Inverse Fast Fourier Transform (IFFT)/Fast Fourier Transform (FFT). For example, secondary group 3 that uses 2 contiguous channels — Channels 7 and 8 — needs only an 8-point IFFT/FFT, but secondary group 1 that uses two discrete channels — Channels 1 and 3— needs 16-point IFFT/FFT. As a result, the latter needs $\frac{16log_216}{8log_28} \approx 2.67$ times more computation time [115]. However, considering the potential increase of spectral utilization, the increased computational complexity should be an acceptable compromise.

A framework to realize the proposed use of multiple channels is illustrated in Figure 6.13. Each radio devices in a secondary communication-group scan the channels as described in Chapter 5. When a radio device detects an idle channel, that device sends a re-synchronization packet to inform the other radio devices of the new OFDM setting (i.e., the new set of OFDM subcarriers). Each device then generates the OFDM signal, via the SDR module, based on the new OFDM setting. In case some of the current occupied channels become unavailable, the radio devices may either cease the use of the corresponding subcarriers or follow the same procedure in Chapter 5 to vacate those channels.

6.4 Conclusion

In this chapter, we derived an optimal allocation of multiple channels for spectral-agile secondary communication-groups and proposed a distributed resource sharing algorithm to approximate the performance of the optimal allocation. We investigated the effects of channel characteristics and scanning frequency on channel utilization, and provided an analytical model to compute the optimal scanning frequency. In order to guarantee a fair use of available resources, we also proposed the use of restrictions on

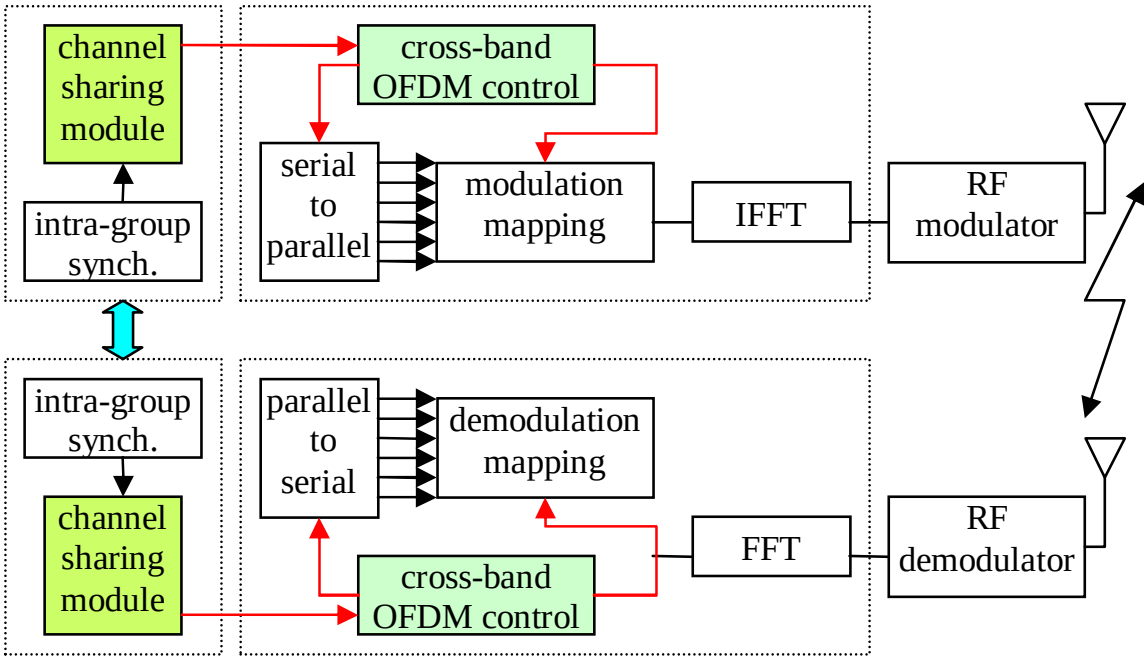


Figure 6.13. Framework of cross-band OFDM

secondary communication-groups' channel occupancy times so as to maintain fairness. A framework to integrate the proposed algorithm with spectral-agile communication — by using the cross-band adaptive OFDM — was also provided.

CHAPTER 7

Unified Smooth-and-Fast Handoff

Wireless networks have two distinct properties — compared to its wired counterpart — that make QoS provisioning very difficult. One is the scarcity of transmission bandwidth and the other is user mobility. As we discussed so far, the QoS problem resulting from bandwidth scarcity can be alleviated by adopting the bandwidth allocation or spectral agility. By using these techniques, users can at least receive QoS support to some extent. However, such QoS support could be compromised by the handoffs resulting from user mobility. If handoffs occur very frequently and incur long delays (i.e, large handoff latency), the resulting QoS may become unacceptable.

A handoff occurs when a mobile station moves from the current radio access cell/network to a new access cell/network. During the handoff, the mobile station cannot send and receive any packet since the current connection (i.e., a link between a mobile station and its previous access point (AP)) has been torn down but the connection with the new AP has not yet been established. This "blackout" interval is referred to as handoff latency, and ranges from hundreds of milliseconds to several seconds depending on the underlying wireless networks. For example, the latency of a handoff between two IEEE 802.11 APs is about 200-400 msecs while that between two MobileIP mobility agents (or access routers) can be up to 3 seconds. Obviously, a handoff latency in the order of second is intolerable from the perspective of QoS provisioning.

In this chapter, we propose a unified smooth and fast handoff scheme to improve both link-layer (e.g., the IEEE 802.11 wireless network) and IP-layer (e.g., the MobileIP network) handoffs. The proposed scheme is based on the IEEE 802.11f standard, namely, Inter-Access Point Protocol (IAPP), and its support for cross-

subnet communication between APs. We enhance the IAPP by adding a cross-subnet frame buffering-and-forwarding mechanism so as to support smooth link-layer handoffs. Based on this smooth link-layer handoff scheme, we show how the IP-layer handoff latency can be reduced and how the IP-layer packet losses can be eliminated — by means of the enhanced IAPP — without modifying the existing MobileIP standard.

This chapter is organized as follows. Section 7.1 discusses the design rationale of the proposed handoff scheme. Section 7.2 elaborates on the problem of frame losses during a link-layer handoff, and discusses the consequence and solutions for this problem. We introduce the current IEEE 802.11 IAPP, and present the enhanced IAPP in Section 7.3. There, we explain how both the link- and IP-layer handoffs benefit from the enhanced IAPP. The detailed implementation of the proposed protocol and the *ns-2* simulation results are presented in Section 7.4. Finally, conclusions are drawn in Section 7.5.

7.1 Handoffs in Wireless and Mobile Networks

There are two types of handoffs in wireless/mobile networks: intra- and inter-subnet handoffs. In an intra-subnet handoff, the APs involved in the handoff reside in the same IP subnet. A wireless station only needs to establish a link-layer connection (with the new AP) without modifying the IP address. Therefore, an intra-subnet handoff is also referred to as a link-layer or layer-2 handoff. A typical example of the intra-subnet handoff occurs when a wireless station moves across between two APs of an IEEE 802.11 wireless LAN. In an inter-subnet handoff, the APs involved in the handoff reside in two different IP subnets. A mobile station not only needs to establish a link-layer connection (with the new AP) as in an intra-subnet handoff, but also needs to obtain a new IP address to maintain IP-layer reachability. Therefore, an inter-subnet handoff is also referred to as an IP-layer or layer-3 handoff. Figure 7.1 depicts these two types of handoffs and the relation between them.

The easiest approach to facilitate the handoff process is to use the beacon-based movement detection mechanisms. For example, in an IEEE 802.11 wireless LAN, the

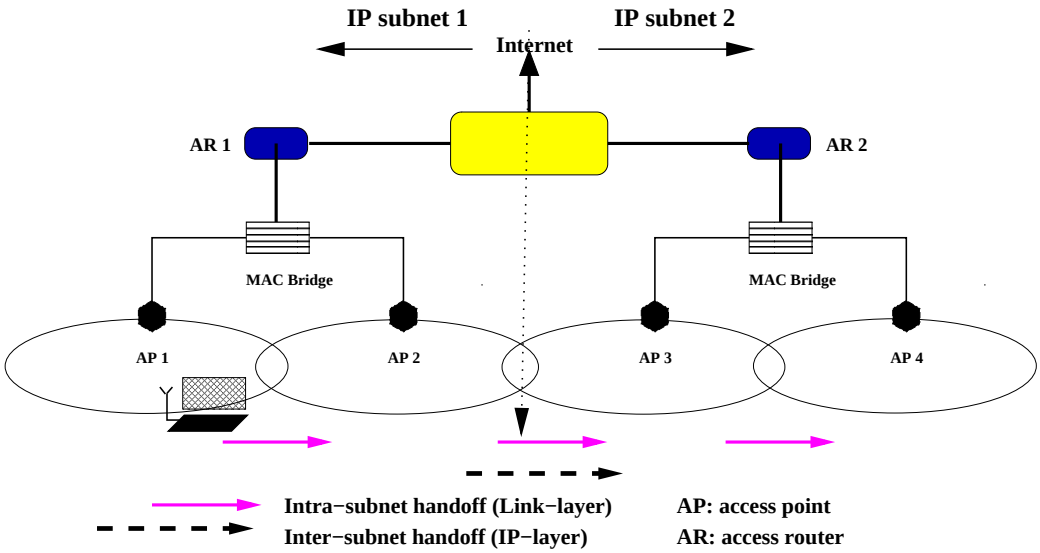


Figure 7.1. Intra-subnet (link-layer) and Inter-subnet (IP-layer) handoffs

APs periodically broadcast the beacon frames to mobile stations. By receiving the beacon frames and comparing the signal strengths, a mobile station can determine whether or not it is about to or has moved out from the current AP and whether or not to initiate a link-layer handoff. In the MobileIP networks, mobility agents or access routers also periodically send out router advertisement containing subnet prefix information. A mobile station can then determine if it has moved to a new IP subnet based on the information provided by the router advertisement and decide whether or not to initiate an IP-layer handoff. By using these beacon-based systems, the handoff latency is primarily determined by the beacon or advertisement interval. In an IEEE 802.11 wireless LAN, the beacon interval is 100 milliseconds, which incurs a link-layer handoff latency of 200∼400 milliseconds [98]. In a MobileIP network, the advertisement interval is 1 second, which may incur an IP-layer handoff latency of up to 3 seconds. In general, a 3-second disconnection from the network is not acceptable for most of the applications.

Since the link-layer handoff is much faster than the IP-layer handoff, one method to expedite the IP-layer handoff is to exploit the link-layer handoff process. For example, a link-layer handoff can be used as a good indication of an upcoming IP-layer handoff given that an inter-subnet handoff involves both link- and IP-layer

handoffs. By using such an indication, a mobile station can initiate the IP-layer handoff right after the link-layer handoff is completed. As a result, one can reduce the inter-subnet handoff latency to the range of the intra-subnet handoff latency. However, there are two problems that still needs to be solved by using such cross-layer schemes. First, both of the intra- and inter-subnet handoffs are not loss-free, primarily due to the non-zero link-layer handoff latency. We will show in the next section that this "believed-to-be-short" link-layer handoff suffices to result in some packet losses which can be very harmful to some applications. Second, not every link-layer handoff indicates the advent of an inter-subnet handoff. Therefore, the IP layer (either in the mobile station or the access router) still needs some extra information to determine whether or not the station already moves out of the current IP subnet. For example, the mobile station may send out a *Router Solicitation* packet, according to the Neighbor Discovery protocol [107], whenever the mobile station receives a link-layer handoff indication. The mobile station can then determine if it has moved to a new IP subnet by examining the solicited *Router Advertisement*. However, the Neighbor Discovery protocol requires a mobile station to delay the initial *Router Solicitation* for a random time (to alleviate congestion when many stations start up on a link at the same time), and also requires an access router to delay the solicited *Router Advertisement* for another random time (so a single advertisement can respond to multiple solicitations). These delays can easily add up to significantly degrade the performance achieved by using link-layer handoff indication.

Based on these observations, we conclude this section by listing some key requirements of a "good" handoff scheme as follows.

- A mobile station should exploit the link-layer handoff indication to reduce the IP-layer handoff latency. However, the mobile station should use such indications in a timely fashion, and require no modification of the existing IP-mobility protocol.
- A mobile station should not experience any packet loss during both intra- and inter-subnet handoffs. Moreover, the packets that cannot reach the mobile station during a handoff should be sent to the mobile station right after the

link-layer handoff is completed.

- A mobile station should not need to differentiate the intra- and inter-subnet handoffs in the sense that the mobile station should follow the unified procedure for both intra- and inter-subnet handoffs.

7.2 Frame Losses in a Link-layer Handoff

Even though the link-layer handoff process is very fast and usually incurs a latency of several hundred milliseconds, A mobile station is still subject to packet loss during an intra-subnet handoff. Such packet losses, as we will show in this section, may degrade the performance of the fast IP-layer handoff schemes using link-layer handoff indications. To show this potential degradation, we establish a test bed and demonstrate how the relatively small link-layer handoff affects the TCP performance. The setup of our test bed is shown in Figure 7.2, where AP1 and AP2 run under the Linux operating system and use D-link IEEE 802.11b wireless LAN cards with Prism2 chipset. The wireless station (STA) also runs Linux but uses a Cisco IEEE 802.11b wireless LAN card. Two FTP servers, one local server (FTP server 1) and one remote server (FTP server 2), are both considered in order to study the impact of round-trip time (RTT) on the TCP performance. FTP server 1 runs Linux with finer timer granularity such that the TCP retransmission timeout (RTO) is about 500 msecs (as shown in Figure 7.3), while the RTO of the FTP sessions with FTP server 2 is about 2 seconds because of the coarse timer granularity and larger minimal RTO value used in Unix machines [116].

7.2.1 Scenario I: Small Round-Trip Time

Figure 7.3 plots the TCP sequence numbers of the STA's FTP session with FTP server 1 throughout a link-layer handoff. The FTP session is interrupted by unplugging the cable between AP1 and the bridge for about 3 seconds (starting at around the 42nd second) before the STA's handoff in order to obtain the RTO value, which is about 500 msecs in this setting. After the handoff takes place at 45.3 sec, all packets destined for the STA get lost. Upon completion of the handoff, one can observe

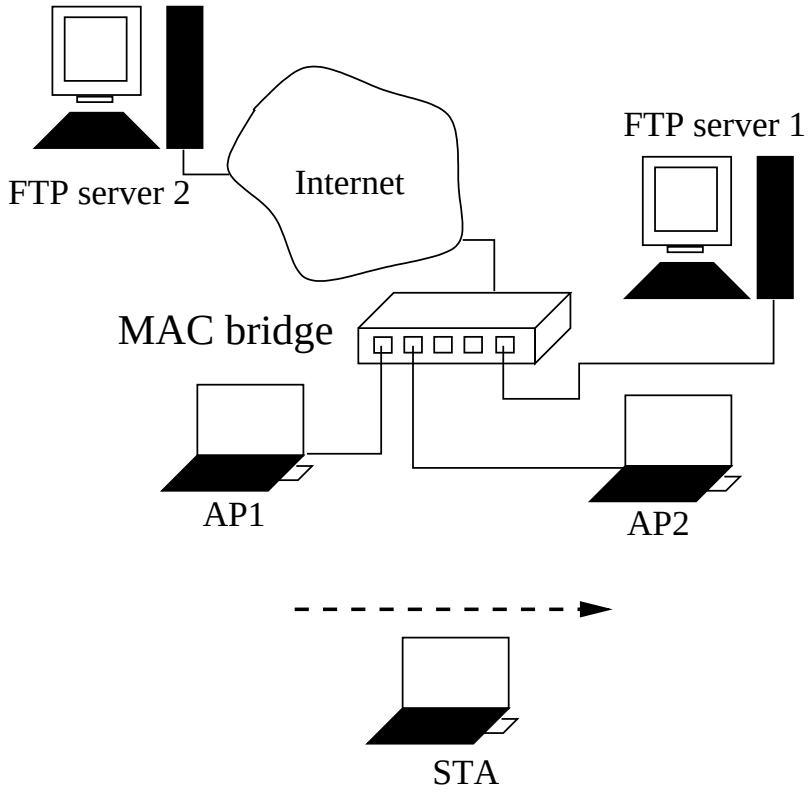


Figure 7.2. A Test bed of TCP performance during a link-layer handoff

that some new packets taking the new route (due to the link-layer update frame) arrive at AP2. These packets are transmitted from the sender's TCP congestion window because the TCP sender receives some acknowledgements right after the handoff. These acknowledgements are those that cannot be sent by the STA before the handoff and are sent via the new AP after the handoff. Due to some packet losses during the handoff, the TCP sender times out eventually and the first lost packet is retransmitted (about 500 msecs after it was transmitted for the first time). This result shows that even though the link-layer handoff latency is small, a TCP retransmission timeout can still be triggered due to packet losses, thus degrading the throughput.

To remedy the problem shown in Figure 7.3, we modify the drivers of the APs' LAN cards in order to support link-layer frame buffering and forwarding for the STA [98, 114]. The TCP sequence numbers under this new setting are shown in Figure 7.4. One can observe that upon completion of the handoff, all packets buffered

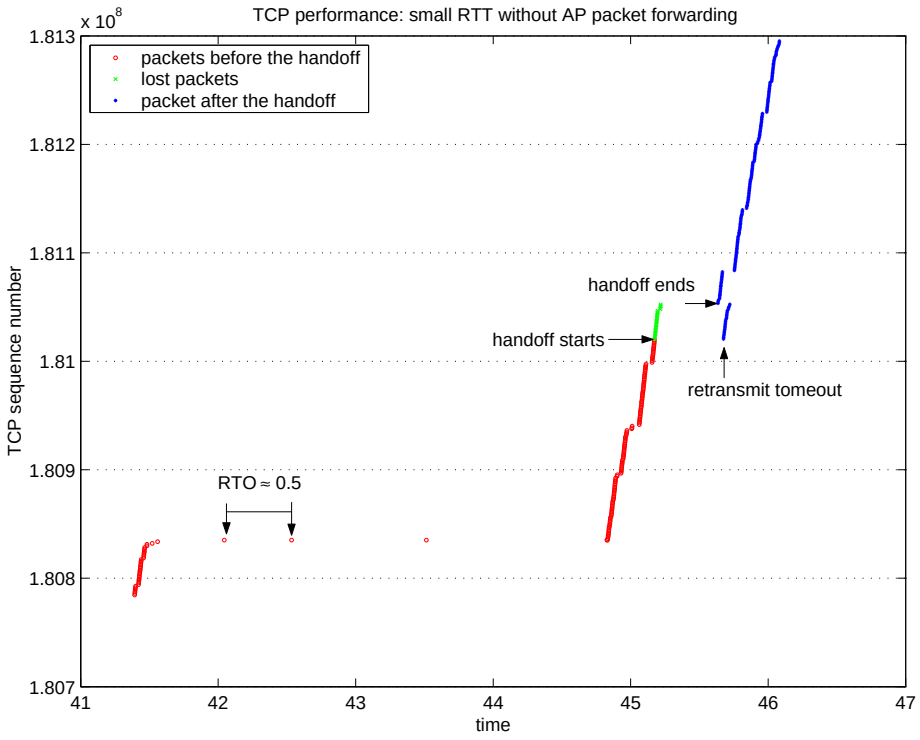


Figure 7.3. TCP performance - scenario I: small RTT without link-layer frame forwarding

at AP1 during the handoff are forwarded to the STA via AP2, and no retransmission timeout occurs. Note that forwarded packets and packets taking the new route (due to the link-layer update frame) arrive at AP2 interleavingly because of the small RTT in this setting. However, TCP can handle this type of out-of-order packet delivery without invoking fast retransmit since the number of out-of-order packets is always less than 3 in our experiment.

7.2.2 Scenario II: Large Round-Trip Time

Figure 7.5 shows the TCP sequence numbers of the STA's FTP session with FTP server 2 during a handoff. All the packets arriving at the AP1 during the handoff simply get lost if there is no link-layer frame buffering and forwarding. Upon completion of the handoff, some new packets arrive at the STA via AP2 as in the previous cases. Unlike the first case in which the RTT is small, no TCP retransmission timeout occurs because of the larger value of RTO and the relatively small link-layer handoff latency. Instead, out-of-order packets (i.e., the new packets via the new route) will

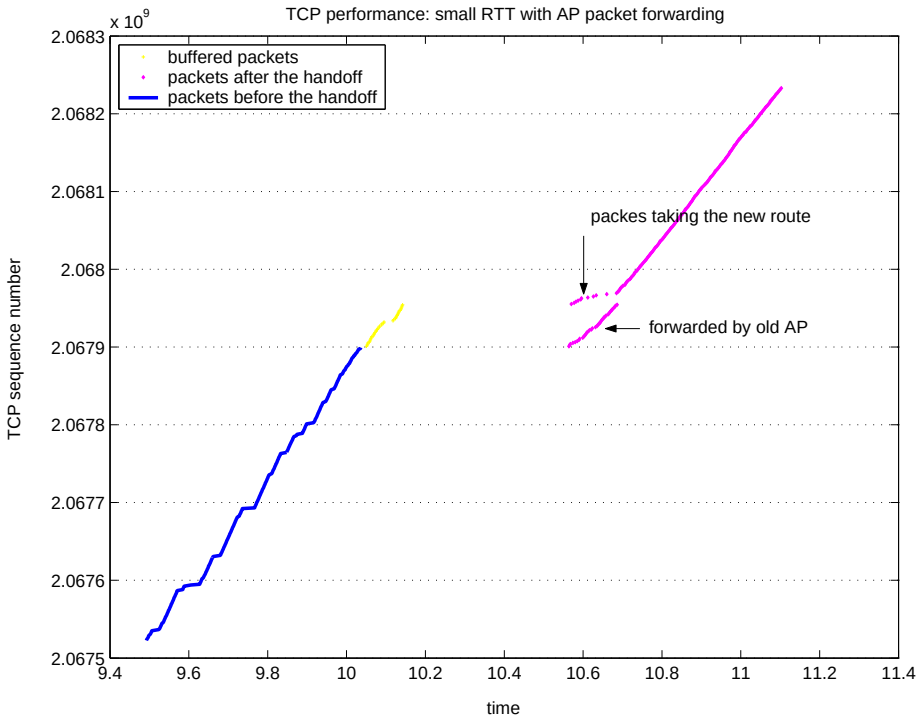


Figure 7.4. TCP performance - scenario I: small RTT with link-layer frame forwarding

invoke TCP fast retransmit such that the lost packets are retransmitted at 27.5 second. This undue invocation of fast retransmit again reduces the TCP throughput. Figure 7.6 shows the TCP sequence numbers in the case where the APs support link-layer frame buffering and forwarding. Upon completion of the handoff, the packets buffered at AP1 are forwarded to AP2. Since the RTT is large in this case, forwarded packets always arrive earlier than the packets taking the new route and therefore, no out-of-order packet delivery occurs. That is, the handoff is completely transparent to the TCP session in this scenario.

The above experiments show that, without link-layer frame buffering and forwarding, either the TCP retransmission timeout or fast retransmit will be invoked during a link-layer handoff. This invocation of TCP congestion control unduely reduces the TCP congestion window and consequently, the throughput. However, if the frame buffering and forwarding is applied, the link-layer handoff becomes transparent to the TCP (and upper-layer applications). That is, this link-layer frame buffering and forwarding helps an already-fast link-layer handoff become an error-free (or smooth)

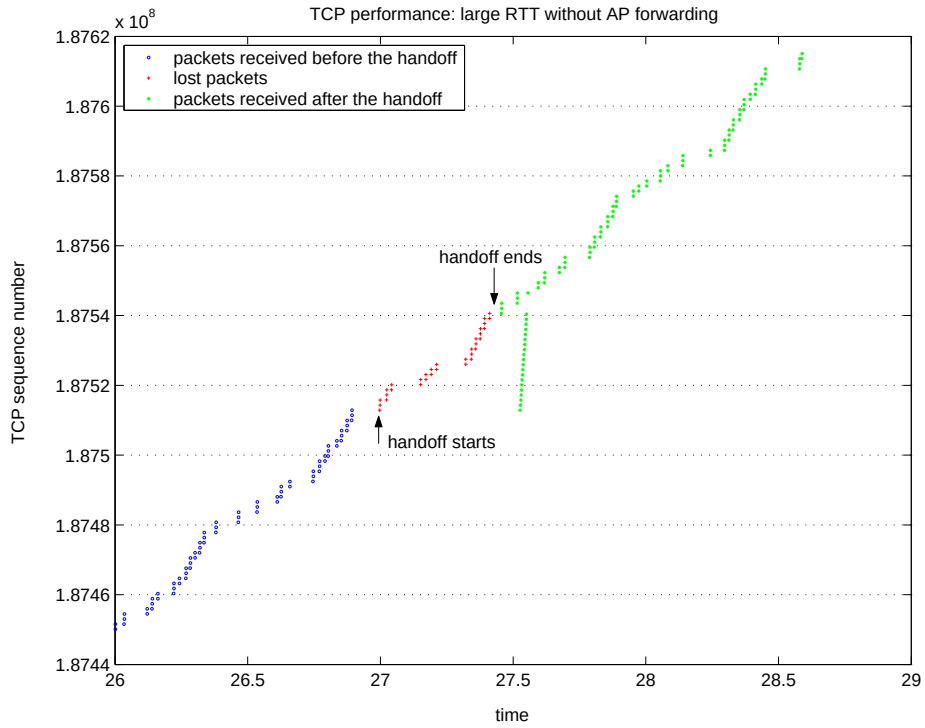


Figure 7.5. TCP performance - scenario II: large RTT without link-layer frame forwarding

handoff. *Unfortunately, the above link-layer frame buffering and forwarding cannot make the fast IP-layer handoff schemes (which use the link-layer handoff indication) error-free because the APs involved in an IP-layer handoff do not reside in the same LAN segment as in our experiment. However, this problem can be solved by using the (enhanced) IAPP as we describe in the next section.*

7.3 Inter-Access Point Protocol (IAPP)

In order to better describe the IAPP, we first introduce some basic concepts of the IEEE 802.11 network architecture. The basic unit in an IEEE 802.11 network is the so-called "basic service set" (BSS), which is also the building block of the well-known Wi-Fi wireless LAN. Within a BSS, wireless stations (STAs) can communicate with each other and access the wired Internet via the STA serving as an AP of the BSS. Instead of being standalone, a BSS may also form a component of an extended form of network that is built with multiple BSSs. This extended form of network is called an "extended service set" (ESS) and the architectural component used to interconnect

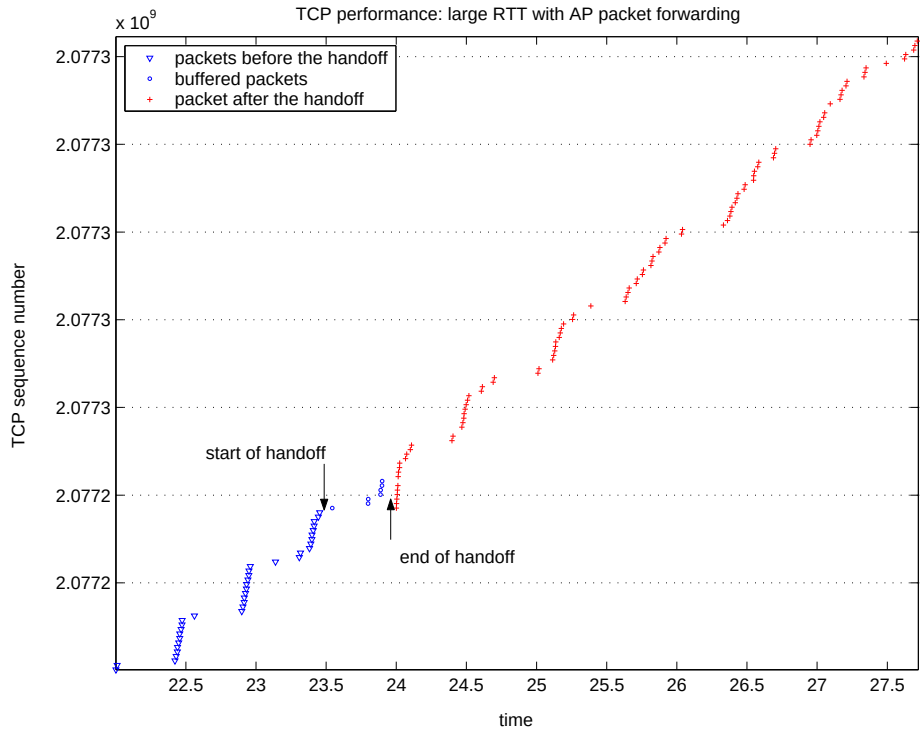


Figure 7.6. TCP performance - scenario II: large RTT with link-layer frame forwarding

BSSs (to form an ESS) is the distribution system (DS). The relations among these components are illustrated in Figure 7.7.

In a common DS, two STAs which cannot communicate directly with each other via wireless medium can still communicate, as long as both STAs belong to the same ESS. That is, an ESS conceptually appears the same to a logical link control layer as a BSS but with a larger "coverage". The IEEE 802.11 standard does not require the DS to be link layer-based or network layer-based as long as the DS can distribute the packet, based on the provided information, to the correct "output" point that corresponds to the desired recipient. The information required by the DS can be obtained from the association-related packets in the IEEE 802.11 standard.

7.3.1 Original IAPP

With the basic concepts introduced above, we can now discuss the IAPP. Briefly, the IAPP is a set of functionalities and a protocol used by an AP to communicate with other APs on a common DS. It is part of a communication system comprising

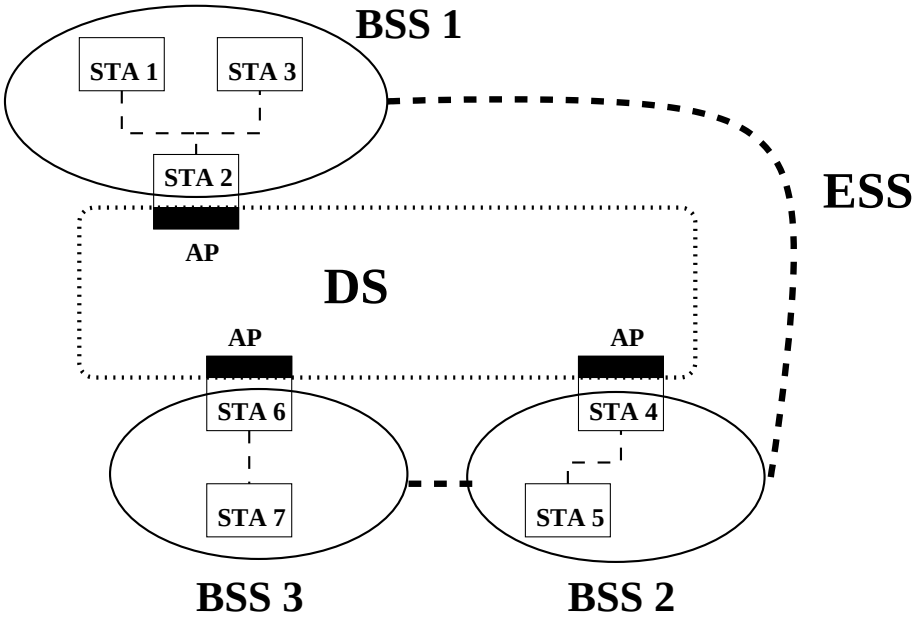


Figure 7.7. The IEEE 802.11 wireless network architecture

APs, STAs, an arbitrarily-connected DS and Remote Authentication Dial In USER Service (RADIUS) servers [113]. The RADIUS servers provide two functions: (i) mapping the BSS Identification (BSSID) of an AP to its IP address on the DS and (ii) distribution of keys to the APs to allow the encryption of the communications between the APs. The functions of the IAPP are to (1) facilitate the creation and maintenance of the ESS, (2) support the mobility of STAs and (3) enable APs to enforce the requirement of a single association for each STA at a given time.

Among the functions provided by the IAPP, we focus on the IAPP's support for STAs' mobility. The events and packet exchanges followed right after a STA moves away from its current AP are illustrated in Figure 7.8. First, the STA starts searching for a new AP by switching to different channels and seeking new beacon frames. If a new AP is located, the STA attempts to reassociate with this AP by sending a reassociation request. This request contains the STA's MAC address and the BSSID of the STA's previous AP. Upon receiving this reassociation request, the new AP

replies to the STA with a reassociation response using the MAC address obtained in the received reassociation request. The new AP also sends an IAPP *MOVE-notify* to the old AP via the DS as required by the IAPP. The old AP then responds to the new AP a *MOVE-response* which carries the context block for the STA's association from the old AP to the new AP.

The IAPP MOVE-notify and MOVE-response are IP packets carried in a TCP session between APs. The IP address of the old AP must be found by mapping the BSSID from the reassociation message to its IP address. This mapping is done using a RADIUS exchange and any standard RADIUS server that support the CALL CHECK service-type should work.[1] Finally, a link-layer update frame is sent by the new AP so that any local layer-2 devices, such as bridges, switches and other APs, can update their forwarding tables with the correct port to reach the new location of the STA.

7.3.2 Enhanced IAPP

Although the current IAPP expedites the link-layer handoff by means of context transfer, there still exists a time period (also shown in Figure 7.8) during which the STA cannot send or receive anything. Therefore, the problems demonstrated in Section 7.2 may still occur. To fix this problem, we include the same technique — the link-layer frame buffering and forwarding — into the current IAPP. *However, unlike the "link-local" frame buffering and forwarding in Section 7.2, the frame buffering and forwarding powered by the enhanced IAPP enables frame forwarding between the APs in the same subnet as well as the APs in different subnets.* The frame forwarding follows right after the old AP sends the MOVE-response back to the new AP and is illustrated in Figure 7.9.

Each link-layer frame forwarded by the old AP is carried in a new IAPP packet called the IAPP *MOVE-forward*, and sent directly to the new AP via TCP/IP. TCP is used, rather than UDP, because of its defined retransmission behavior and the

[1]It can also be done using locally-configured information mapping the BSSID of APs to their IP-address on the DS.

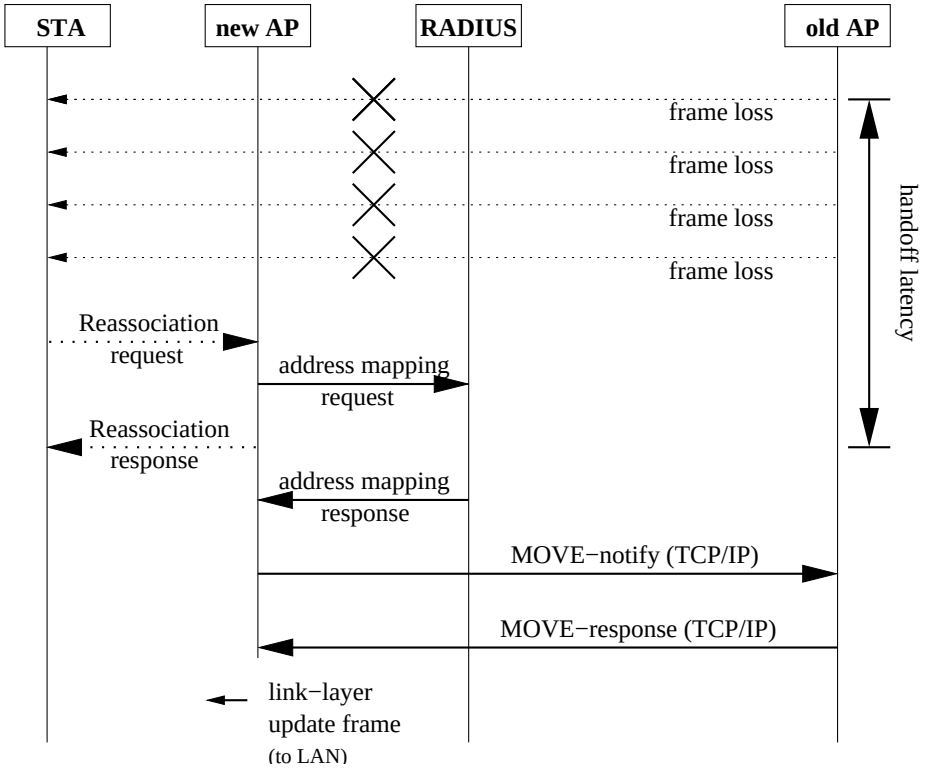


Figure 7.8. The IAPP MOVE-notify and MOVE-response packet exchanges during a link-layer handoff

need for reliable forwarding. The IAPP MOVE-forward packet format is depicted in Figure 7.10. The "Command" field in the IAPP packet header identifies the specific function of the packet. For the IAPP MOVE-forward packet, one can choose any integer value between 7 and 255.[2] The "Data" field contains a subfield "MAC Address" which represents the MAC address of the STA which initiates the reassociation request. This address can be obtained (by the old AP) from the IAPP MOVE-notify packet, and is used by the AP receiving the MOVE-forward packet for transmitting the link-layer frame to its final recipient. The AP retrieves the entire link-layer frame from the "Information" subfield of the "Context Block" in a received MOVE-forward packet, and transmits this link-layer frame to the STA once the authentication or security association between the AP and the STA is completed.

[2] 1-6 are reserved for IAPP MOVE-notify, MOVE-response and etc.

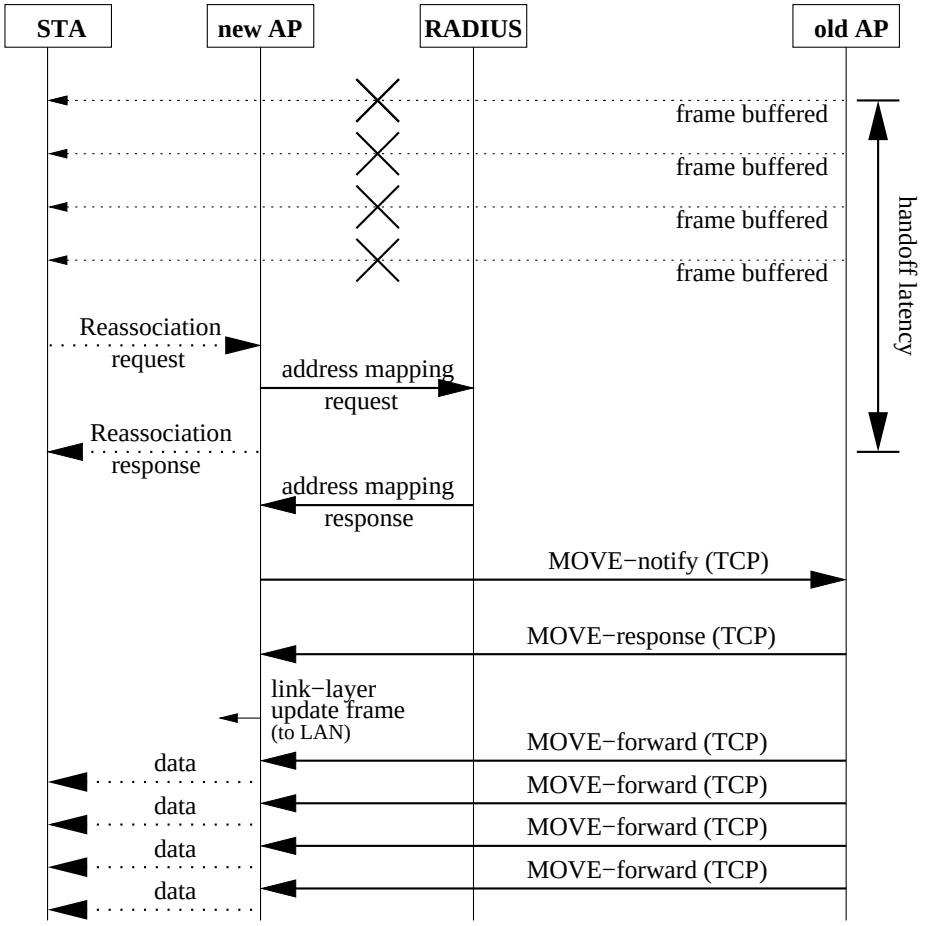


Figure 7.9. The enhanced IAPP packet exchanges during a link-layer handoff: MOVE-notify/MOVE-response packets followed by MOVE-forward packets

7.3.3 Improvements by the Enhanced IAPP

The enhanced IAPP not only improves the link-layer handoff as described in Section 7.2,but also it improves the IP-layer handoff as follows.

1. A mobile station can receive forwarded link-layer frames (from the old AP) via the new AP even when this new AP resides in a different IP subnet, because the IAPP is an IP-based protocol and the forwarded frames are transmitted via TCP/IP.
2. Because of (1), if the mobile station moves to a new IP subnet, it can resume receiving packets (via the IAPP MOVE-forward packets) even before the IP-layer handoff (e.g., the MobileIP procedure) is initiated. From the mobile station's perspective, the IP-layer handoff latency is reduced to the level of the link-

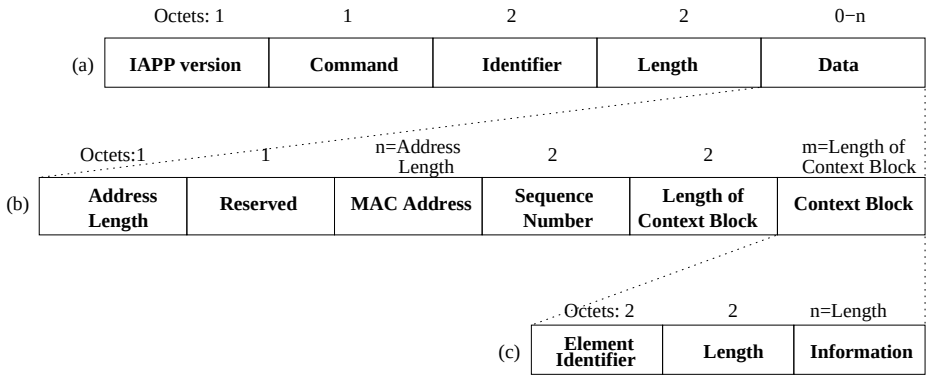


Figure 7.10. IAPP MOVE-forward packet format: (a) General IAPP packet format, (b)MOVE-forward DATA field format, and (c) Information element format

layer handoff latency as in those fast handoff schemes using link-layer handoff indications.

3. The APs function uniformly regardless of the type of handoffs they are involved with, because the enhanced IAPP need not differentiate between a link-layer and an IP-layer handoff for the purpose of packet forwarding. More importantly, access routers are not involved in packet buffering and forwarding. As a result, the intelligence of determining the handoff type in order to initiate a fast handoff is not required any longer.
4. Because of (1)-(3), a fast and smooth IP-layer handoff is achieved "implicitly" (by the enhanced IAPP) without modifying the MobileIP. That is, a fast IP-layer handoff is achieved without coupling link-layer operations with MobileIP operations. Such independence makes the enhanced IAPP applicable to other protocols supporting IP mobility which may emerge in the near future.
5. The mobile station requires neither multiple radio interfaces nor *a priori* knowledge of the new AP it may head for, thanks to the "post-handoff" nature in the enhanced IAPP.
6. No additional *over-the-air* signaling is required as other schemes, except the original reassociation frame in the IEEE 802.11 standard. Of course, the frame buffering and forwarding requires resources at both end APs, and consumes network bandwidth along the path between them. However, the wired network is not the resource bottleneck and such resource requirement should be acceptable

in order to achieve smooth handoffs.

7.3.4 Unified Link- and IP-layer Handoffs

Next, we show via an example how the enhance IAPP can actually achieve all of the above salient features. Let us consider the scenario shown in Figure 7.1, and consider the case when a mobile station moves from AP1 to AP2, and eventually to AP3. As the mobile station is handed off to AP2, it sends a reassociation request to AP2 as required by the IEEE 802.11 standard. Once it receives the reassociation request from the mobile station, AP2 follows the enhanced IAPP shown in Figure 7.9: it sends a reassociation response to the mobile station and an IAPP MOVE-notify to AP1. In the meantime, AP1 buffers all link-layer frames destined for the mobile station (signaled by the frame retry count as we will detail later). Upon receiving the IAPP MOVE-notify from AP2, AP1 replies with an IAPP MOVE-response and forwards all buffered frames to AP2. Then, AP2 sends a link-layer update frame to the local subnet and transmits the link-layer frames received from AP1 to the mobile station via the wireless link. Since the link-layer update frame "refreshes" the local MAC bridge's forwarding table, the new link-layer frames (from the mobile node's corresponding node) will take the direct route to AP2. Under this scenario, the mobile station will soon receive the router advertisement from AR1 and realize that no IP-layer handoff is necessary.

Next, suppose that the mobile station moves from AP2 to AP3. The mobile station and AP3 follow exactly the same procedures as above (since it is just a link-layer handoff so far). AP2 also reacts exactly the same as AP1 during the first handoff. The only difference is that now the forwarded link-layer frames take a longer, cross-subnet path. However, this is perfectly fine since the APs communicate with each other via the DS, which is an IP-based distribution system required by the IAPP. Then, AP3 sends a link-layer update frame to its local subnet and transmits the forwarded link-layer frame to the mobile station via the wireless link. Until this time instant, the mobile station (more precisely, the MobileIP entity) has not been informed of an upcoming IP-layer handoff by the link layer (and, in fact, the

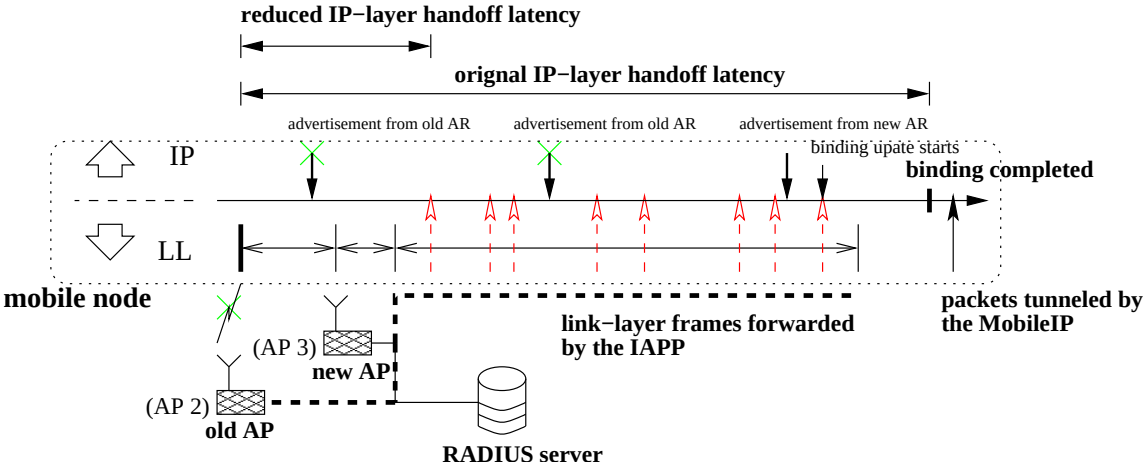


Figure 7.11. Smooth and fast IP-layer handoffs by using the enhanced IAPP: (i) IP-layer handoff latency is reduced to the level of link-layer handoff latency and (ii) packet losses are eliminated by link-layer frame buffering and forwarding

MobileIP entity will never be informed by the link layer in our scheme). It is until the mobile station receives a new router advertisement from AR2 that the MobileIP entity starts the normal MobileIP binding update. In the mean time, the packets still reach the mobile station via the IAPP MOVE-forward packet, along the route from AP2, via the MAC bridges and the routers, to AP3. This handoff process is illustrated in Figure 7.11. As shown in the figure, the IP-layer handoff latency is reduced significantly and is equal to that in the post-registration fast handoff schemes. More importantly, all APs react uniformly to both handoffs and the MobileIP is left intact.

7.4 Simulation and Evaluation

The proposed enhanced IAPP is implemented in the *Network Simulator* (ns-2) since at present there is no off-shelf wireless LAN card supporting the IAPP. Without giving too much of implementation details, we list the essential operations in the AP and the mobile station for supporting the enhanced IAPP. Especially, we describe how the AP gets signaling of packet buffering based on the existing IEEE 802.11 standard.

7.4.1 Operations of APs

Since an AP works differently depending on whether it is acting as an old AP or a new AP for the mobile station, we separate discussions of the AP's operations accordingly.

Old AP

The most important tasks of an old AP are to (i) buffer the packets destined for the mobile station once it lost the connection with the mobile station, and (ii) forward the packets after it is informed by another AP about the mobile's handoff. For packet buffering, an old AP needs some signaling mechanism to initiate the buffering process. Although the IEEE 802.11 standard defines the disassociation procedure between an AP and a mobile station, using disassociation packets as the signaling is not reliable because the disassociation packet may never reach the old AP before the mobile station loses the link-layer connection.[3] In our implementation, we use the packet retry count as the signaling for packet buffering.

In the IEEE 802.11 wireless LAN, a frame can be retransmitted up to *retry count limit* (=7) times before it is discarded. If the old AP has retransmitted a packet 7 times, it is a strong indication that the mobile station may have moved out of the old AP's coverage area. Of course, the frame may happen to collide with others, but the probability that a packet collides with others for 7 consecutive times is extremely small due primarily to the exponential random backoff in the IEEE 802.11 standard. Another possibility of consecutive packet retransmissions is that the mobile station suffers a bad reception due to multi-path fading. We handle this situation as follows.

1. An AP buffers any frame which is supposed to be discarded based on the IEEE 802.11 standard (that is, any frame with the retry count exceeding *retry count limit*). The AP also starts a timer which expires 500 msecs after the first frame is buffered.
2. Whenever a frame from the mobile station is received, the AP discards all buffered packets[4] and stops the timer.

[3]Most existing IEEE 802.11 wireless LANs do not support disassociation between APs and mobile stations via the wireless link.

[4]For better performance, the AP can send the buffered packets to the mobile station but this is

3. If the timer expires but the AP does not receive an IAPP *Move-notify* from other APs, the AP discards all buffered frames and stops the timer.
4. If the AP receives an IAPP *Move-notify* regarding a mobile station whose MAC address matches the destination MAC address of a buffered frame, the matched frame is forwarded and the timer is stopped. Moreover, the AP sets a *forwarding flag* associated with the mobile station to TRUE so that in-flight frames destined for the mobile station will also be forwarded once they arrive at the old AP.

By following the above procedure, the old AP can accurately buffer the frames for the mobile station during a link-layer handoff. One should note that all of these operations (in the old AP) are at the MAC layer as required by the IEEE 802.11 standard, except the operations involved with other APs (including MOVE-notify, MOVE-response and MOVE-forward), which are regulated by the IAPP.

New AP

The new AP follows the procedure as we explained in the previous section. In addition, the new AP will

- set the *forwarding flag* associated with the mobile station to FALSE once the AP completes the reassociation process of the mobile station. This way, the new AP can stop any frame forwarding that may have been activated for the mobile station when last time the mobile station is handed off from this AP.
- check the list of associated mobile stations for every received MOVE-forward packet. If the MAC address contained in the IAPP header of the MOVE-forward packet matches any one of the mobile stations in the list, the new AP retrieves the link-layer frame from the received MOVE-forward packet, and transmits it to that MAC address via the wireless link immediately. Otherwise, the new AP discards the received MOVE-forward packet.

7.4.2 Operation of a Mobile Station

The mobile station follows the normal reassociation procedures defined in the IEEE 802.11 standard during a link-layer handoff. In addition, the mobile station also

out of the scope of a handoff.

follows the procedure below.

1. The mobile station buffers any frame which is supposed to be discarded based on the IEEE 802.11 standard (that is, the frame with the retry count exceeding *retry count limit*). The mobile station also starts a timer which expires 500 msecs after the first frame is buffered.
2. Whenever a frame from the current AP is received, the mobile station discards all buffered frames[5] and stops the timer.
3. If the timer expires but the mobile station does not receive any beacon frame from other APs, the mobile station discards all buffered frames and stops the timer.
4. If a new beacon frame is received before the timer expires, the mobile station stops the timer and forwards the buffered frame to the new AP once the reassociation with the new AP is completed.

By following this procedure, the mobile station can prevent any uplink (from the mobile station to the AP) packet loss during a handoff. As a result, both uplink and downlink transmissions are error-free during both intra- and inter-subnet handoffs.

7.4.3 Simulation and Evaluation

The network topology used throughout the simulation is shown in Figure 7.12. All APs in the figure are the IEEE 802.11 wireless APs. AP1 and AP2 reside in an IP subnet and are connected by a MAC bridge, while AP3 and AP4 reside in another IP subnet and are also connected by a MAC bridge. The purpose of using the MAC bridges is to separate the APs in the same IP subnet so that they are in two different "segments". This way, we can capture the effects of link-layer update frame (in the IAPP protocol) on a intra-subnet handoff process. In order to better monitor the mobile station's handoffs, we choose transmission power and receiving power threshold in a way that the mobile station loses its connection to both APs when it is in the middle of the two APs, which are separated by 40m.

[5]For better performance, the mobile station can send the buffered packets to the current AP but this is out of the scope of a handoff.

The mobile station in the figure follows a very simple movement pattern. The mobile station starts at AP1 and heads toward AP2 at a fixed speed S. Once reaching AP2, the mobile station turns right and heads toward AP3 with same speed. The mobile station repeats the same rules after it arrives AP3, then AP4 and eventually AP1. After that, the mobile station starts all over again. This way, the mobile station will experience 2 intra-subnet handoffs (between AP1 and AP2, and between AP3 and AP4) and 2 inter-subnet handoffs (between AP2 and AP3, and between AP4 and AP1). For each inter-subnet handoff, the mobile station has to perform a link-layer handoff (between the APs) and also a IP-layer handoff (between the ARs).

In order to initiate a handoff, a mobile station needs to seek a new beacon frame (for a link-layer handoff) or a router advertisement (for an IP-layer handoff) after waiting for some time and still receiving no beacon or advertisement from the current AP or AR. This waiting time is usually chosen to be multiple beacon frame intervals (for a link-layer handoff) or multiple router advertisement intervals (for an IP-layer handoff). Of course, one can choose a waiting time equal to a beacon/advertisement interval to expedite a handoff. However, the mobile station may miss a beacon/advertisement simply because of a transmission error or a packet collision. Therefore, choosing too small a waiting time may force a mobile station to switch to other radio channels for seeking new beacons/advertisements which may be unnecessary in the first place. That is, the beacon/advertisement waiting time creates a trade-off between the handoff latency and accuracy of initiating a handoff process. Since the link-layer handoff latency is relatively small (usually hundreds of milliseconds), we choose the beacon waiting time to be twice of the beacon interval (=100 msecs) to prevent any "premature" channel switching. For the router advertisement waiting time, we consider the value of a single router advertisement interval (=1 second) and twice of the interval (=2 seconds).

Finally, we use the TCP-based application as the traffic source in our simulation. The mobile station and its correspondent node establish a FTP session with an approximated end-to-end throughput of 2.4 Mbps, based on the chosen packet size (=1500 byte), average round-trip time ($\approx$ 100 msecs) and the maximal TCP

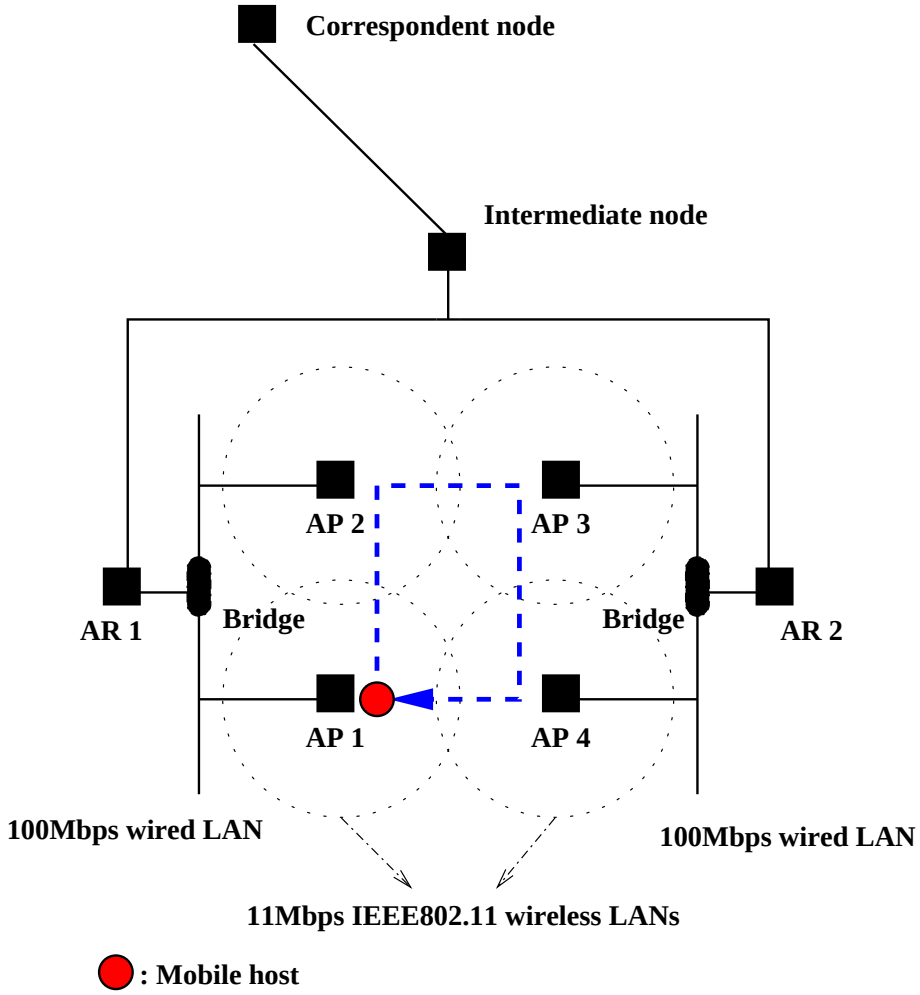


Figure 7.12. Network topology in the *ns-2* simulation

congestion window size (20). In what follows, we show how the enhanced IAPP improves handoff process in terms of handoff latency and overall throughput, and investigate the impacts of user mobility and router-advertisement waiting time on these improvements.

Reduced IP-layer Handoff Latency

Since we have already shown the effects of link-layer packet buffering and forwarding on intra-subnet handoffs in Section 7.3, we now focus on the inter-subnet handoff in this subsection. The trace of TCP sequence numbers (in the mobile station side) under the enhanced IAPP is plotted in Figure 7.13-(a). Here we only show an inter-subnet handoff between AP2 and AP3 around $t = 12$ second. At $t = 12.48$ second, the

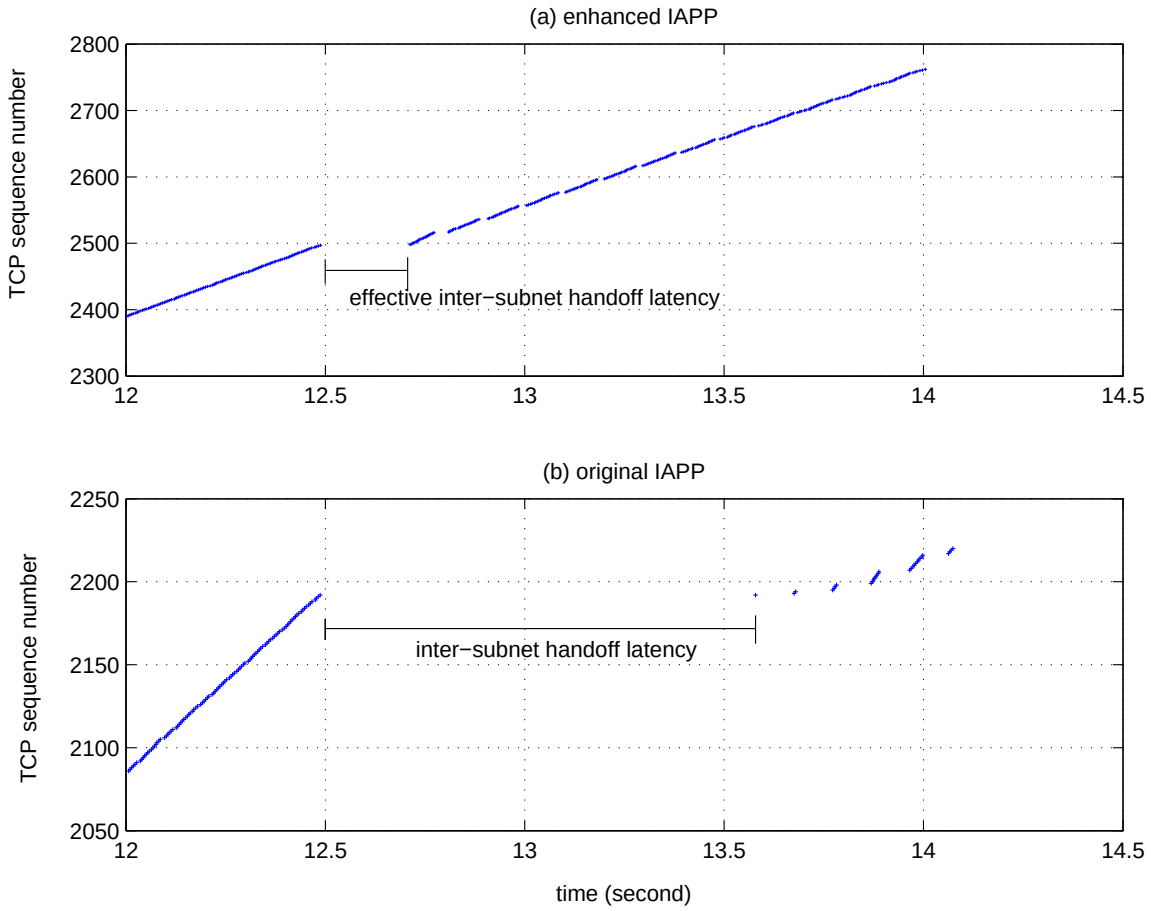


Figure 7.13. Reduced IP-layer handoff latency as compared to the original MobileIP-only scheme

mobile station loses its connection with AP2 when it is heading for AP3. However, the mobile station has not detected the situation since it just received a beacon frame from AP2 at $t = 12.4$ second and believes it is still connected. It is until $t = 12.62$ second that the mobile station starts seeking new beacon frames because the beacon-frame waiting time has expired (200 milliseconds in our simulation). At $t = 12.7$ second, the mobile station receives a new beacon frame rom AP3 and attempts to re-associate with AP3. After the reassociation is completed, the mobile station starts to receive forwarded TCP packets from AP2 via AP3 (note that it is a batch of 20 packets). It should be noted that at this time point, the mobile station has not discovered yet that it has moved to a new IP subnet. It is until $t = 13.4$ second that the mobile station receives a router advertisement from AR2 (via AP3), and then starts the binding update. Once the binding update is completed, the TCP packets

will take the new route instead of being forwarded by AP2. Under this scenario, the "effective" intra-subnet handoff latency is equal to the link-layer handoff latency, which is around 210 milliseconds in our simulation.

Figure 7.13-(b) shows the same scenario as above except that we use the original IAPP. As in the previous case, the link-layer handoff process is completed around $t = 12.7$ second. However, without packet buffering and forwarding, the mobile station receives nothing from the correspondent node until the TCP packet #2192 times out at $t = 13.52$ second (note the exponential increase of TCP congestion window size thereafter). Unfortunately, the TCP retransmission timeout reduces the correspondent node's TCP congestion window size, hence reducing the throughput. We will investigate this issue in the next subsection. In regard to the handoff latency, the resulting inter-subnet handoff latency is around 1 second, which is 790 milliseconds more than that of using the enhanced IAPP. Of course, the inter-subnet handoff latency also depends on the router-advertisement waiting time. So far, we use the minimal waiting time (equal to the router advertisement interval). One can expect an even longer inter-subnet handoff latency (without the enhanced IAPP) if we allow the use of a longer router-advertisement waiting time. We will also discuss this issues in the following simulations.

User Mobility

Based on the mobility pattern described in the beginning of this section, we choose 3 different speeds for the mobile station, namely $S = 2m/s$, $S = 5m/s$ and $S = 10m/s$. These three different speeds represent *low-mobility*, *medium-mobility*, and *high mobility*, respectively. We set the router-advertisement waiting time as a router-advertisement interval, which is the minimal value one can choose. This way, the mobile station is more "agile" in seeking new router advertisements and initiating a handoff process.

Figure 7.14 shows the number of TCP packets received by the mobile station in an 85-second time interval (so that a mobile station can visit all APs at a speed of 2 m/s) at different speeds. For each speed, we use the original IAPP and the enhanced

IAPP for comparative purposes. As shown in the figure, the mobile station receives more packets at all three speeds if the enhanced IAPP is applied. These improvements originate from the fact that neither the TCP fast retransmit nor retransmission timeout is invoked, thanks to the loss-free, much faster handoff process enabled by the enhanced IAPP. In contrast, the TCP fast retransmit may occur during an intra-subnet handoff and the TCP may time out during an inter-subnet handoff, if the original IAPP is used.

The percentage improvements (compared to the original IAPP) are also shown in the figure indicating that the higher the user mobility, the larger the percentage improvement. This is because when the mobile station moves fast, it experiences more handoffs and thus, the effects of the enhanced IAPP can kick in. The improvement can be as up to 50% for the high-mobility case. Of course, the improvement also depends on the router-advertisement waiting time used by a mobile station. In the simulation, we use the smallest value (=1 second) given that the router-advertisement interval is 1 second as suggested in the MobileIP standard. One can expect that if a larger waiting time is used, the transmission of a mobile station will stall longer, under the original IAPP, due to the longer inter-handoff latency. In contrast, the transmission of a mobile station is not affected by the value of router-advertisement waiting time under the enhanced IAPP as we will show next.

Router-Advertisement Waiting Time

As mentioned earlier, there exists a trade-off between the handoff latency and accuracy of initiating a handoff process. Although choosing a small router-advertisement waiting time can reduce an intra-subnet handoff latency, doing so may sometimes invoke movement-detection operations which should not take place at all, hence incurring control overhead. For example, a mobile station may simply miss a router advertisement due to transmission errors. To investigate the impact of this waiting time on the handoff performance, we consider both 1-second and 2-second waiting times. A 2-second waiting time allows a mobile station to miss one router advertisement without trying to initiate an inter-subnet handoff. In the original MobileIP

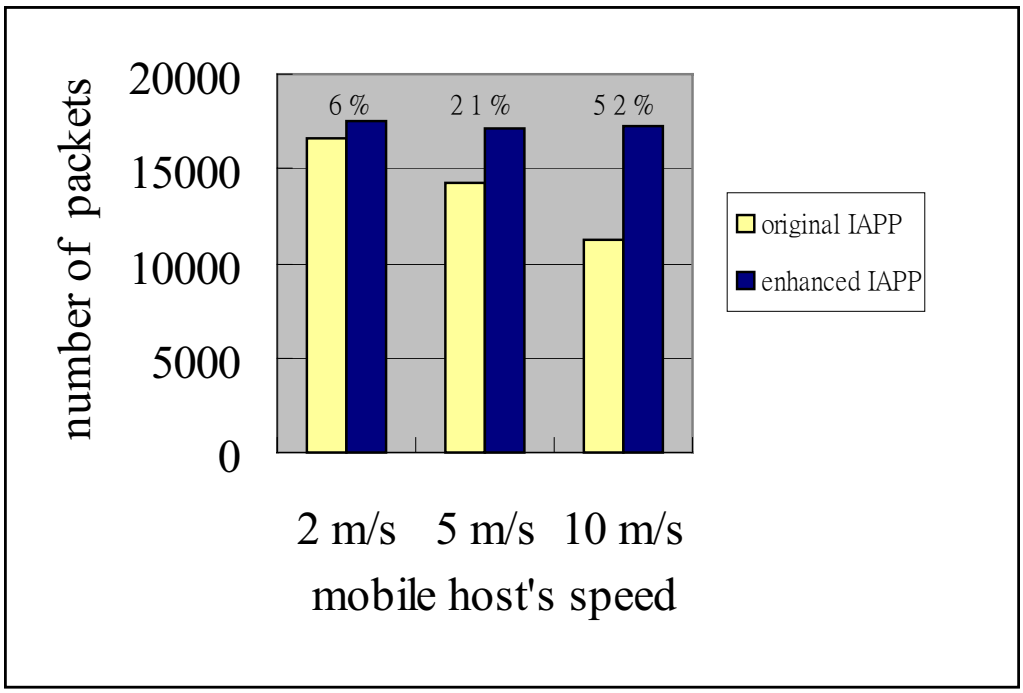


Figure 7.14. Throughput improvement made by the enhanced IAPP under different user mobility

standard, the waiting time should not exceed 3 seconds (that is, allowing a mobile station to miss two consecutive router advertisements).

The number of TCP packets received by the mobile station are shown in Figure 7.15 for both waiting times under the original IAPP and the enhanced IAPP. One can observe that the mobile station receives 42% less packets if a larger waiting time under the original IAPP is used. This is because the larger waiting time suffices to cause 2 consecutive TCP retransmission timeouts during an inter-handoff latency. Note that an unacknowledged TCP packet will time out within around 1 second under our simulation setting. Therefore, if a packet gets lost when an inter-subnet handoff starts (under the original IAPP), the packet is retransmitted again after 1 second, and will get lost again since the handoff is not completed (may take up to 2 seconds to re-configure the IP-layer reachability in the case of a larger waiting time). The exponential increase of the second retransmission timeout further degrades the TCP performance. However, a TCP retransmission timeout does not occur under the enhanced IAPP, thanks to the small "effective inter-subnet handoff" as we explained in the first subsection. Since this effective inter-subnet handoff is

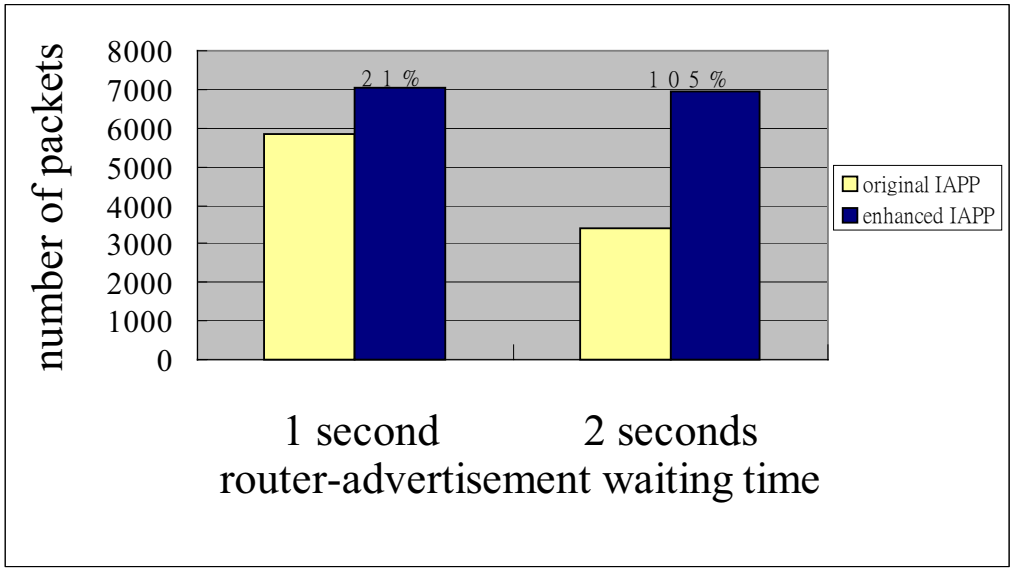


Figure 7.15. Throughput improvement made by the enhanced IAPP for different MobileIP router-advertisement waiting times

solely decided by the link-layer handoff latency, the TCP performance is not affected by the router-advertisement waiting time as also shown in Figure 7.15.

Based on the simulation results, we can conclude that the enhanced IAPP allows the use of a larger router-advertisement waiting without sacrificing the TCP performance or increasing inter-subnet handoff latency. In other words, the enhanced IAPP optimizes the aforementioned trade-off between the handoff latency and accuracy of initiating a handoff process caused by the router-advertisement waiting time.

7.5 Conclusion

In this chapter, we proposed a simple but effective enhancement for the IEEE 802.11 IAPP to improve both intra- and inter-subnet handoff processes. We showed that the enhanced IAPP can reduce the inter-subnet handoff latency significantly without modifying the MobileIP standard. Unlike other existing schemes which require the MobileIP entity to process link-layer handoff indications, our enhanced IAPP decouples the MobileIP operations from the underlying link-layer handoff process. Such decoupling makes the enhanced IAPP applicable to other IP-mobility solutions. The simulation results showed that the enhanced IAPP supports high user mobil-

ity, and requires no user intervention in the sense that the fast IP-layer handoff is automatically achieved by means of the IAPP-enabled, cross-subnet frame buffering-and-forwarding. The enhanced IAPP was also shown to allow the MobileIP to use a less aggressive movement detection, thus reducing the handoff overhead.

CHAPTER 8

Conclusion and Future Work

This thesis explored the problems of adaptive QoS provisioning in wireless and mobile networks. First, we developed a mathematical model to analyze the effects of adaptive bandwidth allocation on both system performance and user-perceived QoS. With this model, a wireless network can dynamically adjust the user's bandwidth — based on the network load or network capacity — with *controllable* degradation of user-perceived QoS. We then developed a distributed airtime usage control to facilitate adaptive QoS support in time-division wireless networks such as the IEEE 802.11 wireless LANs. By using the proposed airtime control, stations using the contention-based medium access method are shown to be able to provide users the parameterized QoS, which can only be achieved by using the polling-based medium access method in the current IEEE 802.11e standard. Moreover, the distributed airtime usage control has potential for providing QoS support in ad hoc IEEE 802.11 wireless LANs.

In order to further improve the user's QoS, the concept of "spectral agility" is introduced to the wireless networks (especially, the IEEE 802.11 wireless LANs). We established an analytical model to study the achievable improvement gained by using spectral agility, and developed a comprehensive framework to fully exploit spectral agility. This framework and the associated functionalities are integrated with the IEEE 802.11 wireless LAN in the *ns*-2 simulator to demonstrate the effectiveness of the resulting spectral-agile wireless networks. Finally, we studied the mobility support for QoS provisioning in the IEEE 802.11 wireless LAN, and developed a unified smooth-and-fast handoff for both intra- and inter-subnet handoffs based on the Inter-Access Point Protocol.

8.1 Contributions

The main contributions of this thesis are summarized as follows.

- Developed a mathematical model to analyze adaptive bandwidth allocation problems, and investigate the tradeoff between system performance and user-perceived QoS. This model provides an analytical framework for developing predictive or adaptive bandwidth allocation algorithms in wireless and mobile networks.
- Developed a distributed airtime usage control that can be used to adjust user bandwidth for adaptive QoS support in time-division wireless networks. This airtime usage control can also be used to support QoS without using centralized resource allocation, which makes the proposed airtime control an attractive solution for QoS provisioning in ad hoc IEEE 802.11 wireless LANs.
- Analyzed the performance gain of using spectral agility, and developed a comprehensive framework to realize spectral-agile communication. The spectral-agile communication not only improves the overall spectral efficiency but also provides a better QoS support for individual users.
- Developed a smooth-and-fast handoff scheme that uses a unified procedure for both intra- and inter-subnet handoff processes. The inter-subnet handoff latency can be reduced to the range of intra-subnet handoff latency without modifying the IP-mobility protocols.

8.2 Future work

As future work, we would like to first study the problem of using the proposed airtime usage control for QoS provisioning in ad hoc IEEE 802.11 wireless networks. As outlined in Chapter 5, such QoS support requires a distributed admission control that can only be achieved by each wireless station via monitoring the network load. We would like to study the performance of using integrated distributed admission control and airtime usage control in ad hoc IEEE 802.11 wireless LANs. We would also like to improve the performance of the proposed spectral-agile communication.

First, we would like to investigate a more effective scanning mechanism which combines the current proactive scanning (on a regular basis) and reactive scanning (on an on-demand basis), to reduce the scanning overhead while still providing accurate information about spectrum availability. Second, we would like to consider the proactive channel switching, in addition to the current reactive switching mechanism, so as to eliminate any potential interference with the primary users. Finally, we would like to study the effects of spectral-agile radios on user QoS provisioning and develop adaptive QoS support based on the spectral-agile radios. In summary, we would like to:

- study QoS support in ad hoc IEEE 802.11 networks using the proposed distributed airtime usage control algorithm;
- enhance the spectral-agile communication by using the reactive spectrum scanning and proactive channel switching mechanism, and analyze its performance; and
- study the interaction between the adaptive bandwidth allocation and the opportunistic use of spectral resource, and integrate these two mechanisms for better adaptive QoS support.

APPENDICES

APPENDIX A

Computation of Conditional Fairness Index

Let $\bar{n} = (n_1, n_2, \cdots, n_M)$ be the vector that represents the numbers of idle channels occupied by the M secondary groups, the conditional fairness index $F(M, K)$ is defined as

$$F(M, K) = E\left[\max(\bar{n}) - \min(\bar{n}) | \sum n_i = K\right], \tag{A.1}$$

where $E[X|A]$ is the expected value of random variable X given that event A occurs, and $\max(\bar{n})/\min(\bar{n})$ is the maximum/minimum element of vector $\bar{n}$.

The channel occupancy vector, $\bar{n}$, is jointly decided by the secondary group's scanning mechanism and the proposed algorithm in Figures 6.2-6.4. In order to simplify our analysis, we divide the decision process for $\bar{n}$ into two independent stages: (I) the idle channels are discovered by all secondary groups based on the scanning mechanism and (II) the channel occupancy decided in (I) is adjusted according to the proposed algorithm. If the secondary group's scanning period is much less than the channels' mean ON/OFF period, this is a good approximation because the channels switch rarely and every idle channel can be discovered by the secondary groups.

Let $\bar{n}'$ be the vector that represents the numbers of idle channels occupied by the secondary groups in stage I. Given that there are K idle channels and each secondary group has an equal probability to discover an idle channel, there exist M^K different instances of channel occupancy, each with a probability of $\frac{1}{M^K}$. Since all idle channels will be discovered given $T_{on}/T_{off} \gg f_{gscan}$, the constraint

$$n'_1 + n'_2 + \cdots + n'_M = K, \tag{A.2}$$

must be satisfied. Therefore, the probability of $\bar{n}' = (n'_1, n'_2, \cdots, n'_M)$ can be com-

puted by

$$p(\bar{n}') = \frac{K!}{n_1! \cdot n_2! \cdots n_M!} \cdot \frac{1}{M^K}. \tag{A.3}$$

It should be noted that if $(n'_1, n'_2, \cdots, n'_M)$ is a solution of Eq. (A.2), any permutation of $\{n'_1, n_2 1, \cdots, n'_M\}$ is also a solution for Eq. (A.2) and has the same probability as given in Eq. (A.3). These permutations all represent the same "channel allocation" form the the perspective of fairness provisioning as implied by Eq. (A.1).

Having $\bar{n}'$ in stage (1), we can determine $\bar{n}$ according to the proposed sharing algorithm in Figures 6.2-6.4. For example, if $\bar{n}' = (4, 1, 0)$ in the case of $M = 3$ and $K = 5$, the third secondary group will eventually acquire one channel from the first secondary group according to Figures 6.2 and the first secondary will vacate that channel according to Figures 6.3. That is, $\bar{n} = (3, 1, 1)$. If $\bar{n}' = (3, 2, 0)$, we have $\bar{n} = (2, 2, 1)$ with a probability of 0.6 and $\bar{n} = (3, 1, 1)$ with a probability of 0.4 because the third secondary group will randomly discover a channel from the five idle channels. As a result, it is either that the first secondary group vacates one of its three channels for the third secondary group, or the second secondary group vacates one of its two channels for the third secondary group. Since it is difficult to explicitly express $\bar{n}$ as a function of $\bar{n}'$, the relation is denoted as $\bar{n} = f(\bar{n}')$.

Finally, the conditional fairness index can be obtained by

$$F(M, K) = \sum p(\bar{n}')[\max(f(\bar{n}')) - \min(f(\bar{n}'))], \tag{A.4}$$

where there are $\frac{(K+M-1)!}{K!(M-1)!}$ different $\bar{n}'$'s that satisfy the constraint $n'_1 + n'_2 + \cdots + n'_M = K$. As we mentioned earlier, any permutation of the elements in $\bar{n}'$ is also a solution

$\bar{n}'$	$p(\bar{n}') = \frac{5!}{n_1'! n_2'! n_3'!} \cdot \frac{1}{3^5}$	$\bar{n} = f(\bar{n}')$	$\max(\bar{n}) - \min(\bar{n})$
(5,0,0)	1/243	(3,1,1)	2
(4,1,0)	5/243	(3,1,1)	2
(3,2,0)	10/243	(3,1,1) or (2,2,1)	1.4
(3,1,1)	20/243	(3,1,1)	2
(2,2,1)	30/243	(2,2,1)	1

Table A.1. Computation of $F(3, 5)$.

for the constraint. Take the case of $M = 3$ and $K = 5$ as an example. There are $\frac{(5+3-1)!}{5!(3-1)!} = 21$ different $\bar{n}'$'s that satisfy $n'_1 + n'_2 + \cdots + n'_M = 5$. However, there are only 5 different types of "channel allocation", namely $\{5, 0, 0\}$, $\{4, 1, 0\}$, $\{3, 2, 0\}$, $4\{3, 1, 1\}$, and $\{2, 2, 1\}$ from the perspective of computing $F(M, K)$. For example, $\bar{n}' = (5, 0, 0)$, (0,5,0) and (0,0,5) all have the same probability and result in the same $\max(\bar{n})$-$\min(\bar{n})$. Therefore, the computation can be further simplified. Table A shows these five different channel allocations and the corresponding elements needed in Eq. (A.4). Based on Table A.1, we can compute $F(3, 5)$ as

$$F(3,5) = \frac{3*1*2}{243} + \frac{6*5*2}{243} + \frac{6*10*1.4}{243} + \frac{3*20*2}{243} + \frac{3*30*1}{243} = 1.48, \quad \text{(A.5)}$$

where the first term in the numerator of each fraction is the number of permutations for a given $\bar{n}'$.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] S. Singh, "Quality of Service guarantees in Mobile Computing", *Computer Communications*, no.19, 1996, pp. 359-371.

[2] S. Sen, J. Jawanda, K. Basu, and S. Das,"Quality-of Service Degradation Strategies in Multimedia Wireless Network", *IEEE Vehicular Techonology Conference*, vol.3, May 1998, pp. 1884-1888.

[3] M. R. Sherif, I. W. Habib, M. N. Nagshineh, and P. K. Kermani,"Adaptive Allocation of Resources and Call Admission Control for Wireless ATM Using Generic Algorithm", *IEEE Journal of Selected Areas in Communications*, vol.18, no.2, Feb. 2000, pp. 268-282.

[4] T. Kwon, Y. Choi, C. Bisdikian, and M. Naghshineh, "Call Admission Control for Adaptive Multimedia in Wireless/Mobile Network", *Proceedings of first ACM international workshop on Wireless mobile multimedia*, Oct. 1998, pp. 111-116.

[5] S. Choi and K. G. Shin, "Location/Mobility-Dependent Bandwidth Adaptation in QoS-Sensitive Cellular Networks", *IEEE Vehicular Technology Conference*, vol.3, 2001, pp. 1593 -1597.

[6] Y.B Lin, S. Mohan, and A. Noerpel, "Queueing Priority Channel Assignment Strategy for PCS Handoff and Initial Access", *IEEE Transaction on Vehicular Technology*, vol.43, no.3, Aug. 1994, pp. 704-712.

[7] M. Naghshineh, and M. Schwartz, "Distributed Call Admission Control in Mobile/Wireless Networks", *IEEE Journal of Selected Areas in Communications*, vol.14, no.3, May 1994, pp. 289-293.

[8] W. Lee, and B. Sabata, "Admission Control and QoS Negotiations for Soft-Real Time Applications", *IEEE International Conference on Multimedia Computing*

and Systems, vol.1, 1999, pp. 147-152.

[9] A. Sutoving, and J.M. Peha, "Novel Heuristic for Call Admission Control in Cellular Systems", *IEEE International Conference on Universal Personal Communications*, vol.1, 1997, pp. 129-133.

[10] R. Ramjee, R. Nagarajan, and D. Towsley, "On Optimal Call Admission Control in Cellular Networks", *IEEE INFOCOM '96*, vol.1, pp. 43-50.

[11] K. Mitchell, and K. Sohraby, "An Analysis of the Effects of Mobility on Bandwidth Allocation Strategies in Multi-Class Cellular Wireless Networks", *IEEE INFOCOM '01*, vol.2, pp. 1075-1084.

[12] S. Choi, and K. G. Shin, "Predictive and Adaptive Reservation for Handoffs in QoS-Sensitive Cellular Networks", *Proceedings of ACM SIGCOMM '98*, pp. 155-166.

[13] A. Aljadhai, and T. Znati, "A Framework for Call Admission Control and QoS Support in Wireless Environments", *IEEE INFOCOM '99*, vol.3, pp. 1019-1026.

[14] Z. Liu, M.J. Karol, M.E. Zarki, and K.Y. Eng,"Channel Access and Interference Issues in Multi-code DS-CDMA Wireless Packet (ATM) Networks", *Wireless Networks*, vol.2, no.3, 1996, pp. 173-193.

[15] J.C. Haartsen,"The Bluetooth Radio System", *IEEE Personal Communications*, vol.7, no.1, Feb. 2000, pp. 28-36.

[16] C. Fragouli, V. Sivaraman, and M.B. Srivastava,"Controlled multimedia wireless link sharing via enhanced class-based queuing with channel-state-dependent packet scheduling", *IEEE INFOCOM '98*, vol.2, pp. 572-580.

[17] D.A. Eckhardt, and P. Steenkiste,"Effort-limited Fair (ELF) Scheduling for Wireless Networks", *IEEE INFOCOM '00*, vol.3, pp. 1097-1106.

[18] C. Chao, and W. Chen, "Connection Admission Control for Mobile Multiple-Class Personal Communications Networks", *IEEE Journal on Selected Areas in Communications*, vol. 15, no.8, 1997, pp. 1618-1626.

[19] P. Bremaud, "Markov chains : Gibbs fields, Monte Carlo simulation and queues", Springer, New York, 1999.

[20] S. J. Golestani, "A Self-Clocked Fair Queueing Scheme for Broadband Appli-

cation", *IEEE INFOCOM'94*, 1994, pp. 636-646.

[21] R. Kautz and A. L. Garcia, "Distributed self-clocked fair queueing architecture for wireless ATM networks", *IEEE PIMRC'97*, 1997, pp. 189-193.

[22] N. H. Vaidya, P. Bahl, and S. Gupta, "Distributed Fair Scheduling in a Wireless LAN", *ACM MobiCom'00*, 2000, pp. 167-178.

[23] D. Qiao and K. G. Shin, "Achieving Efficient Channel Utilization and Weighter Fairness for Data Communications in IEEE 802.11 WLAN under the DCF", *International Workshop on Quality of Service (IWQoS'2002)*, May 2002.

[24] T.S.E. Ng, I. Stoica and H. Zhang, "Packet fair queueing algorithms for wireless networks with location-dependent errors", *IEEE INFOCOM'98*, pp. 1103-1111.

[25] P. Ramanathan, and P. Agrawal, "Adapting Packet Fair Queueing algorithm to Wireless Networks", *ACM MobiCom'98*, pp. 1-9.

[26] H. Huang, D. Tsang, R. Sigle and P. Kuhn, "Hierarchial Scheduling with Adaptive Weights for ATM", *IEICE Transations on communication*, Vol. E83-B, no. 2, Feb. 2000, pp. 313-320.

[27] G. Bianchi, L. Fratta, and M. Oliveri, "Perofrmance Evaulation and Enhancement of the CSMA/CA MAC Protocol for 802.11 Wireless LANs", *IEEE PIMRC'97*. Oct. 1996, pp. 407-411.

[28] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed coordination Function", *IEEE Journal on Selected Areas in Communications*, vol. 3, no. 3, Mar. 2000, pp. 535-547.

[29] A. Bakre and B. R. Badrinath, "I-TCP: Indirect TCP for mobile hosts," *Proc. International Conference on Distributed Computing Systems*, 1995, pp. 136–146.

[30] H. Balakrishnan, S. Seshan and R. H. Katz, "Improving reliable transport and handoff performance in cellular wireless networks," *Wireless Networks*, vol. 1, no. 6, 1995, pp. 469–481.

[31] N.H. Vaidya, M.N. Mehta, C.E. Perkins, and G. Montenegro, "Delayed duplicated acknowledgements: a TCP-unaware approach to improve performance Of TCP over wireless," *Wireless Communications and Mobile Computing*, Vol. 2,

no. 1, 2002, pp. 59–70.

[32] S. Floyd, "TCP and Explicit Congestion Notification," *ACM Computer Communications Reviews*, vol. 24, no. 5, 1994, pp. 10–23.

[33] S. Y. Wang and H. T. Kung, "Use of TCP decoupling in improving TCP performance over wireless networks," *Wireless Networks*, vol. 7, no. 3, 2001, pp. 221–236.

[34] M. Mathis, *et al.*, "TCP selective acknowledge options," em RFC 2018, 1996.

[35] J. Kempf, *et al.*, "Bidirectional Edge Tunnel Handoff for IPv6," *IETF Internet Draft*, 2001.

[36] T. Goff, J. Moronski, D. S. Phatak and V. Gupta, "Freeze-TCP: A true end-to-end TCP enhancement mechanism for mobile environments," *IEEE Infocom'00*, pp. 1537–1545.

[37] H. Balakrishnan, S. Seshan and R. H. Katz, "Improving Reliable Transport and Hnadoff Performance in Cellular Wireless Networks", *Wireless Networks*, vol. 1, no. 6, 1995, pp. 469–481.

[38] S. Ohzahata, S. Kimura and Y. Ebihara, "A Proposal of Seamless Handoff Method for Celluar Internet Environments", *IEICE Transactions on communications*, Vo1. E84-B, no. 4, Apr. 2001, pp. 752-759.

[39] J. C. Wu, C. W. Cheng, N. F. Hunag, and G. K. Ma, "Intelligent handoff for mobile wireless Internet," *Mobile Networks and Applications*, vol. 6, no. 1, 2001, pp. 67–79.

[40] C. E. Perkins and K. Y. Wang, "Optimized smooth handoffs in mobile IP," *IEEE ISCC'99* , pp. 340–346.

[41] R. Caceres and V. N. Padmanabhan, "Fast and scalable wireless handoffs in support of mobile Internet audio," *Mobile Networks and Applications*, vol. 3, no. 4, 1998, pp. 351–363.

[42] C. E. Perkins, *et al.*, "IP mobility support for IPv4," *IETF Ingternet Draft*, 2002.

[43] G. Krishnamurthi, *et al.*, "Buffer management for smooth handoffs in IPv6," *IETF Internet Draft*, 2001.

[44] R. Caceres and L. Iftode, "Improving the performance of reliable transport protocols in mobile computing environments," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 5, 1995, pp. 850–857.

[45] R. Ludwig and R. Katz, "The Eifel algorithm: Making TCP robust against spurious retransmissions," *ACM Computer communications Reviews*, vol. 30, no. 1, Jan. 2000, pp. 30–36.

[46] W. R. Stevens, "TCP/IP Illustrated. Volume 2", Addison Wesley, MA, 1994.

[47] A. K. Parekh and R. G. Gallager, "A Generalized Process Sharing Approach to Flow Control in Integrated Services Networks — The Single Node Case", *IEEE INFOCOM'92*, 1992, pp. 915-924.

[48] "Low latency handoff in mobile IPv4," *IETF Internet Draft*, 2001.

[49] "Link layer assisted mobile IP fast handoff method over wireless LAN networks," *Proc. ACM Mobicom 2002*.

[50] S. Ohzahata, S. Kimura and Y. Ebihara, "A proposal of seamless handoff method for celluar Internet environments," *IEICE Transactions on communications*, vo1. E84-B, no. 4, Apr. 2001, pp. 752–759.

[51] A. Elwalid and D. Mitra, "Traffic Shaping at a Network Node: Theory, Optimum Design, Admission Control", *IEEE INFOCOM*, pp. 4b.4.1-4b.4.11, 1997.

[52] X. Liu, K. P. Chong andd N. B. Shroff, "Transmission Scheduling for Efficient Wireless Utilization", *IEEE INFOCOM'01*, pp. 776–785.

[53] B. Sadeghi, V.Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic Media Access for Multirate Ad Hoc Networks", *ACM Mobicom'02*, pp. 24–35.

[54] A. Elwalid, D. Mitra and R. H. Wentworth, "A New approach for allocating buffers and bandwidth to heterogeneous regulated traffic in an ATM node", *IEEE JSAC*, 13(6): 1115-1127, August 1995.

[55] L. Georgiadis, R. Guerin, V. Peris and K. N. Sivarajan, "Efficient network QoS provisioning based on per node traffic shaping", *IEEE/ACM Trans. Net.*, 4(4): 482-501, August 1996.

[56] E. W. Knightly and H. Zhang, "D-BIND: an accurate traffic model for providing QoS guarantees to VBR traffic", *IEEE/ACM Trans. Net.*, 5(2): 219-231, April

1991.

[57] F. LoPresti, Z. Zhang, D. Towsley and J. Kurose, "Source time scale and optimal buffer/bandwidth trade-off for regulated traffic in an ATM node", *IEEE INFOCOM 97*.

[58] S. Rajagopal, M. Reisslein and K. Ross, "Packet Multiplexors with adversarial regulated traffic" *IEEE INFOCOM 98*, pp. 347-355.

[59] D. Werge, E. Knightly, H. Zhang, and J. Liebeherr, "Deterministic delay bounds for VBR video in packet switching networks: Fundamental limits and tradeoffs", *IEEE/ACM Trans. Net.*, 4(3): 352-362, June 1996.

[60] X. Liu, E. K. P. Chong and N. Shroff, "Opportunistic transmission scheduling with resource-sharing constraints in wireless networks", *IEEE JSAC*, 19(10):2053-2064, 2001.

[61] B. Sadeghi, V. Kanodia, A. Sabharwal and E. W. Knightly, "Opportunistic Media Access for Multirate Ad Hoc Networks", *IEEE Mobicom 2002*.

[62] A. K. Parekh and R. G. Gallager, "A Generalized Process Sharing Approach to Flow Control in Integrated Services Networks — The Single Node Case", *IEEE INFOCOM'92*, pp. 915–924.

[63] S. J. Golestani, "A Self-Clocked Fair Queueing Scheme for Broadband Application", *IEEE INFOCOM'94*, pp. 636–646.

[64] J.C.R. Bennett and H. Cheng, "WF2Q: Worst-case Fair Weighted Fair Queueing", *IEEE INFOCOM'96*, pp. 120–128.

[65] IEEE Std 802.11-1999, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*, Reference number ISO/IEC 8802-11:1999(E), IEEE Std 802.11, 1999 edition, 1999.

[66] Daji Qiao and Sunghyun Choi, "Goodput Enhancement of IEEE 802.11a Wireless LAN via Link Adaptation," in *Proc. IEEE ICC'01*, Helsenki, Finland, Jun. 2001.

[67] F. M. Guillemin, N. Likhanov, R. R. Mazumdar and C. Rosenberg, "Extremal traffic and bounds for mean delay of multiplexed regulated traffic streams", *IEEE INFOCOM 2002*.

[68] S. Lu, V. Bharghavan, and R. Srikant, "Fair Scheduling in Wireless Packet Networks", *ACM SIGCOMM'97*, pp. 63–74.

[69] S. Lu, T. Nandagopal, and V. Bharghavan, "A Wireless Fair Service Algorithm for Packet Cellualr Networks", *ACM MOBICOM'98*, pp. 10–20.

[70] C. Fragouli, V. Sivaraman, and M.B. Srivastava, "Controlled Multimedia Wireless Link Sharing via Enhanced Class-Based Queuing with Channel-State-Dependent Packet Scheduling", *IEEE INFOCOM '98*, pp. 572–580.

[71] D. Eckhardt and P. Steenkiste,'Effort-limited Fair(ELF) Scheduling for Wireless Networks", *IEEE INFOCOM'00*, pp. 1097–1106.

[72] M. Jeong, H. Morikawa and T. Aoyama, "Fair Scheduling Algorithm for Wireless Packet Networks", *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E84-A, no. 7, Jul. 2001, pp. 1624–1635.

[73] C.T. Chou and K.G. Shin, "Analysis of Combined Adaptive Bandwidth Allocation and Admission Control in Wireless Networks", *IEEE INFOCOM'02*, pp. 676–684.

[74] IEEE 802.11e/D4.3, *Draft Supplement to Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)*, May 2003.

[75] J. G. Proakis, "Digital Cimmunications", *Mcgraw-Hill*, 1995.

[76] Sai Shankar N and Sunghyun Choi, "QoS Signaling for IEEE 802.11e", *Lect. Notes in Computer Science, Springer Verlag*, August 2002.

[77] C. T. Chou, K. G. Shin and Sai Shankar N, "Inter Frame Space (IFS) Based Service Differentiation for IEEE 802.11 Wireless LANs", *IEEE VTC Fall 2003*.

[78] Chun-Ting Chou, Kang G. Shin and Sai Shankar, "Distributed Control of Airtime Usage in Multi-rate Wireless LANs," under review of the IEEE/ACM Transactions on Networking.

[79] "Facilitating Opportunities for Flexible, Efficient, and Reliable Spectrum Use Employing Cognitive Radio Technologies," *the FCC Notice of Proposed Rulemaking and Order — ET Docket No. 03-108*.

[80] http://www.darpa.mil/ato/programs/XG/rfc_vision.pdf — *Vision RFC.*

[81] I. Aad and C. Castelluccia, "Differentiation mechanisms for IEEE 802.11", in *Proc. IEEE INFOCOM'01*, pp. 209–218.

[82] L. Romdhani, Qiang Ni and T. Turletti, "Adaptive edcf: enhanced service differentiation for IEEE 802.11 wireless ad-hoc networks", in *IEEE Proc. IEEE WCNN'03.*

[83] Jun Zhao and etc., "Performance study of MAC for service differentiation in IEEE 802.11", in *Proc. IEEE GLOBECOM'02*, pp. 778–782.

[84] Jianhua He and etc., "Performance analysis and service differentiation in IEEE 802.11 WLAN", in *Proc. IEEE ICC'03*, pp. 691–697.

[85] C. Bergstrom, S. Chuprun and D. Torrieri, "Adaptive spectrum exploitation using emerging software defined radios", *IEEE Radio and Wireless Conference 1999*, pp.113-116.

[86] P. K. Lee, "Joint frequency hopping and adaptive spectrum exploitation", *IEEE MILCOM 2001*, Vol.1, pp.566-570.

[87] J. Mitola, "The software radio architecture", *IEEE Communications*, Vol.33, No.5, 1995, pp.26-38.

[88] "White paper on regulatory aspects of software defined radio", *SDR forum document number SDRF-00-R-0050-v0.0.*

[89] http://www.darpa.mil/ato/programs/XG/rfc_af.pdf — *Architecture RFC.*

[90] J.B. Punt, D. Sparreboom, F. Brouwer and R. Prasad, "Mathematical analysis of dynamic channel selection in indoor mobile wireless communication systemse", *IEEE Transactions onVehicular Technology*, Vol.47, No.4, Nov. 1998, pp.1302-1313.

[91] "IEEE 802.11h standard (Amendment to IEEE 802.11 Standard, 1999 Edition)", 2003.

[92] J. Khun-Jush, G. Malmgren, P. Schramm and J. Torsner, "Overview and performance of HIPERLAN type 2-a standard for broadband wireless communications", *IEEE VTC 2000-Spring, Tokyo*, Vol.1, pp.112 - 117.

[93] "Additional spectrum for unlicensed devices below 900 MHz and in the 3 GHz

band", *FCC ET Docket No. 02-380.*

[94] "Future trends in defence antenna technology", http://www.bcba15324.pwp.blueyonder.co.uk/consulting/Bibliography/Paper-16.pdf.

[95] D. Bertsekas and R. Gallager, "Data Networks", *Prentice-Hall*, New York, 1992.

[96] C. Perkins and *et al*, "IP Mobility Support", *RFC 2002*, Oct. 1996.

[97] D. Johnson and C. Perkins, "Mobility Support in IPv6," *draft-ietf-mobileip-ipv6-17.txt*," a work in progress.

[98] M. Portoles, "IEEE 802.11 Link-Layer Forwarding for Smooth Handoff,", *IEEE PIMRC'03*, pp.1420-1424.

[99] Jon. C. Wu, C. Cheng and N. Hunag, "Intelligent handoff for Mobile Wireless Interne," *Mobile Networks and Applications*, no.6, 2001, pp. 67-79.

[100] K.E. Malki and *et al*, "Low Latency Handoff in Mobile IPv4," *draft-ietf-mobileip-lowlatency-handoffs-v4-02.txt*, a work in progress.

[101] G. Dommetry and *et al*, "Fast Hnadovers for Mobile IPv6," *draft-ietf-mobile-fast-mipv6-04.txt*, a work in progress.

[102] A. E. Yegin and *at al*, " Supporting Optimized Handover for IP Mobility - Requirements for Underlying System," *draft-manyfolks-12-mobilereq-02.txt*, a work in progress.

[103] IEEE 802.21: A Generalized Model for Link Layer Triggers (An abstract). http://www.ieee802.org/handoff/march04_meeting_docs/Generalized_triggers-02.pdf

[104] J. Choi, "Detecting Network Attachment in IPv6 Problem Statement," *draft-jinchoi-dna-dnav6-prob-00.txt*, a work in progress.

[105] M. Ohta, "Smooth Handover over IEEE 802.11 Wireless LAN,", *draft-ohta-smooth-handover-wireless LAN-00.txt*, a work in progress.

[106] J. Kempf and *et al*, "Bidirectional Edge Tunnel Handover for IPv6," *IETF draft: draft-kempf-beth-ipv6-02.txt*, Mar. 2002.

[107] T. Narten, "Neighber Discovery for IPv6," *RFC 2461*, Dec. 1998.

[108] P. Tan, "Recommendations for Achieving Seamless IPv6 Handover in IEEE

802.11 Networks," *draft-paultan-seamless-ipv6-handoff-802-00.txt*, a work in progress.

[109] Y. Gwon and *at al*, "Fast handoffs in wireless LAN networks using mobile initiated tunneling handoff protocol for IPv4 (MITHv4)," *IEEE WCNC'03*, pp.1248-1253.

[110] T. Yokota and em at al, "Link Layer Assisted Mobile IP Fast Handoff Method over Wireless LAn networks," *ACM Mobicom'02*.

[111] C.E. Perkins and K. Wang, "Optimized Smooth Handoffs in MobileIP," *Proceedings of IEEE International Symposium on Computers and Communications*, Jul. 1999, pp.340-346.

[112] "IEEE 802.11f: Recommended Practice for Multi-Vender Access Point Interoperability via an Inter-Access Point Protocol Access Dostribution Systems Supporting IEEE 802.11 Operation," *IEEE standard 802.11f/D1*, Jan. 2002, Draft.

[113] C. Rigney and *et al*, "Remote Authentication Dial In User Service (RADIUS)," *RFC 2058*, Jan. 1997.

[114] J. Malinen, "Host AP Driver for Intersit Prism2/2.5/3, http://hostap.epitest.fi/

[115] J. G. Proakis and D. G. Manolakis, "Digital Signal Processing", *Prentice-Hall, New Jersey*, 1996.

[116] P. Sarolahti and A. Kuznetsov, "Congestion Control in Linux TCP ", *2002 USENIX Annual Technical conference*, Jun. 2002.

[117] J. P. Pavon and S. Shankar, "Impact of Frame Size, Number of Stations and Mobility on the Throughput Performance of IEEE 802.11e", Mar. 2004, IEEE WCNC'04.