Analysis, Enhancement, and Implementation to Support Quality-of-Service in IEEE 802.11 Wireless LAN

Chia-tai Chou

Advisor: Prof. Kwuang-Cheng Chen

June 27, 2002

Abstract

The rapid growth of Internet and mobile computing brings the need of ubiquitous communications. In wireless local area networks, medium access control protocol is the major part that determines the efficiency of sharing the limited bandwidth. Supporting quality of service (QoS) has been an inevitable trend. IEEE 802.11 has held a group to standardize QoS issues in medium access control (MAC) layer. In this thesis, we propose an analytical model to evaluate the theoretical throughput and delay of its enhanced distributed coordination function (EDCF) under integrated traffic sources. The throughput analysis is modeled by a discrete-time Markov chain, and the delay analysis is based on a MAP/G/1 queueing model. The accuracy of the analysis is verified by simulation.

In addition, a group polling algorithm for hybrid coordination function (HCF) is proposed. This proposed enhancement achieves better performance by taking different priorities into consideration in the polling scheme.

Implementation issue also draws our attention. A flexible and cost-effective software implementation model is proposed. We prove that it works well when traffic load is no more than half of the network capacity by simulation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The night opens the flowers in secret
and allows the day to get thanks.
– Tagore

1.1 An overview of IEEE 802.11 Medium Access Control Protocol

The 802.11 MAC protocol [2] introduces two major functions, the point coordination function (PCF) and the distributed coordination function (DCF). PCF provides contention free period (CFP) for frame transfers, while DCF is a contention-based access mechanism. In QoS network configurations, 802.11e MAC enhancements for QoS [1] defines an additional access method called the hybrid coordination function (HCF), which combines DCF and PCF with some QoS-specific functions and frames for QoS frame transfers during both CFP and contention period (CP).

1.1.1 Enhanced Distributed Coordination Function (EDCF)

The fundamental access method of the 802.11 MAC is called distributed coordination function (DCF) known as a carrier sense multiple access with collision avoidance (CSMA/CA). We first briefly review DCF, and then show how differentiated access to the channel is achieved.

A station that needs to initiate transmission shall first sense if the channel is idle for a period equal to distributed inter-frame space (DIFS). If the channel is busy during DIFS, the station shall defer until the end of current transmission and monitors the channel until it is sensed idle for DIFS. At this time, the station selects a random backoff interval and decreases the backoff interval counter only when the channel is idle. The station starts transmitting when the counter reaches zero. This procedure minimizes collisions during contention between multiple stations that have been deferring to the same event.

A basic service set that supports QoS requirements is called a QoS basic service set (QBSS), and the stations that implement QoS functions are called QSTAs. Each QSTA holds separated queues for traffic of different priorities. The HCF contention-based access, called enhanced DCF (EDCF), achieves differentiated access to the wireless medium from the following aspects:

- Sensing the channel idle for distinct idle duration time $AIFS(l)$ for priority l before the backoff procedure starts. Thus idle time is not a constant DIFS as for DCF.

- The contention window (CW) limits for traffic of different priorities are also different. Traffic of higher priority backoff a shorter time before transmission by choosing a smaller $CW_{min}(l)$. However, $CW_{max}(l)$ may be the same for each priority. The backoff interval is computed by one slot time multiplying an integer drawn from a uniform distribution over the interval $[0, CW_l]$, where CW_l is the contention window of priority l and is within the range of $CW_{min}(l)$

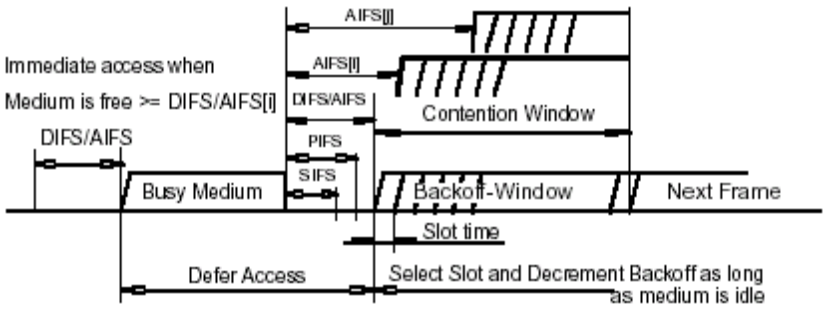


Figure 1.1: Basic access method. (Figure drawn from [1].)

and $CW_{max}(l)$, i.e. $CW_{min}(l) \leq CW_l \leq CW_{max}(l)$.

- Within each QSTA, collisions are resolved such that the higher priority queue gets the opportunity to contend for transmission over the lower one.

The operation of EDCF is illustrated in Figure 1.1.

EDCF operates distributedly. Thus timing synchronization is an important issue. This is achieved by sending a beacon frame at constant repetition intervals.

1.1.2 Point Coordination Function (PCF)

The point coordination function (PCF) provides contention-free frame transfers based on a polling scheme controlled by a point coordinator (PC) operating at the access point (AP) of the service set. The PC gains control of the channel by waiting a shorter time (PIFS) than the stations using the DCF procedure (DIFS). Contention free is achieved by setting network allocation vector (NAV) of each station in the service set. Whenever NAV is set, the station can transmit data only when it is polled. A CFP begins with a beacon frame. After it, the PC shall wait for at least one short inter-frame space (SIFS) period before any transmission. If there is no response to CF-Poll, the AP should poll the next station on the polling list after PCF

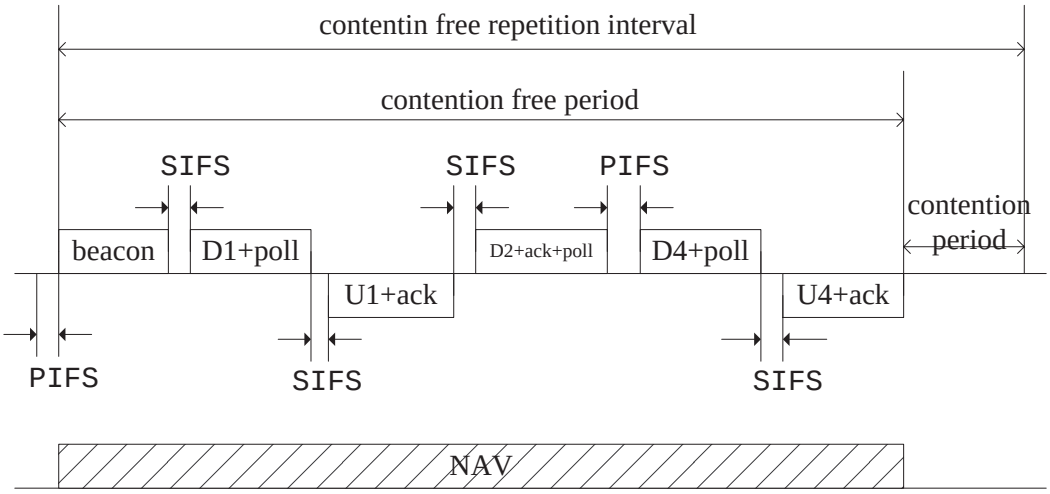


Figure 1.2: PCF frame transfer [2].

interframe space (PIFS) period. A PIFS time is suggested to be one slot time larger than a SIFS time. The operation of PCF is illustrated in Figure 1.2.

1.1.3 Hybrid Coordination Function (HCF)

The hybrid coordination function (HCF) manages bandwidth allocations using a hybrid coordinator (HC) that has higher medium access priority than stations in order to allocate transmission opportunities to stations. Under HCF the basic unit of allocation of the right to transmit onto the wireless medium is the transmission opportunity (TXOP). Each TXOP is defined by a particular starting time, relative to the end of a preceding frame, and a defined maximum length. The TXOP may be obtained by an QSTA receiving a QoS CF-Poll during CP or CFP, or by the QSTA winning an instance of EDCF contention (which will only occur during the CP).

The HCF is a type of point coordinator, but differs from the point coordinator used in PCF in several ways. The most important is that HCF frame exchange sequences may be used among QSTAs during both CFP and CP. Another significant difference is that the TXOP granted by the HC allows multiple frame exchange

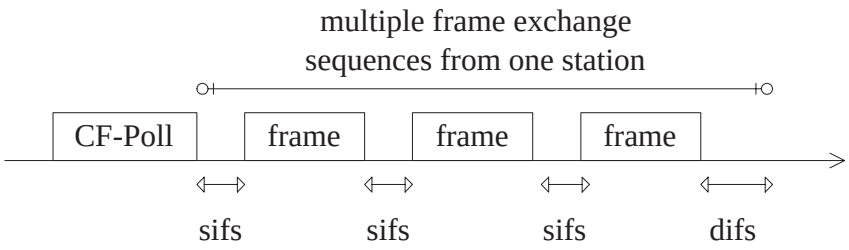


Figure 1.3: Transmission opportunity [1].

sequences within given TXOP (Figure 1.3).

Each frame transmitted within a TXOP is separated by SIFS. The non-QoS frame can be and only be sent as the sole or the final transmission during a transmission opportunity period.

1.2 Related Works

MAC protocols for LANs can be roughly categorized into [3], [4]: random access (e.g. CSMA, CSMA/CD) and demand assignment (e.g. token ring). Most of these protocols are discussed thoroughly in [5] and [6].

While IEEE 802.11 MAC protocol was still a draft, [7] simulates DCF with different MSDU length, bit error rate and offered load. In [8], throughput analysis of 802.11 DCF has been carried out with all exponential backoff windows, but under only data traffic. The authors in [9] suggested an adaptive backoff mechanism tuned on the estimation of the network status. Performance of DCF in presence of hidden terminals is evaluated in [10]. In [11], the delay of voice packets are evaluated in PCF mode. A randomly addressed group polling algorithm for data traffic is proposed in [12]. In [13], a system architecture for link adaptation is introduced.

1.3 Research Objective

The study of wireless local area networks attracts many people in recent years. Wireless communications of only data traffic cannot satisfy our needs. Transferring multimedia sources becomes a basic requirement for mobile communications. The design of wireless networks has to concentrate more on bandwidth consumption issues than wired networks because of limited available bandwidth. Due to the characteristic of shared media in wireless networks, it is necessary to take quality of service (QoS) issues into consideration in the MAC layer. Recently, IEEE 802.11 Group E proposed a draft standard for transport of voice, video, and data traffic over IEEE 802.11 wireless LANs. In order to evaluate the performance of this standard, we propose an analytical model for throughput and delay analysis.

The polling-based HCF lacks efficiency when the addressed recipient does not have frames to send. To resolve this inefficiency, we propose a group polling algorithm which has great improvement over the original HCF by considering different priorities in a group.

There is usually a gap between ideality and practicability. EDCF implemented by placing at least 8 queues in a network interface card costs a lot and is lack of flexibility. We introduce a software implementation model which is more flexible and cost-effective.

1.4 Thesis Outlines

This thesis is organized as follows.

- Chapter 1 first summarizes 802.11 MAC prorocol and its enhancements for QoS. Some related works and our motivations of this study are also presented.
- Chapter 2 provides our analytical model. The throughput analysis is modeled by a discrete-time Markov chain, and the delay analysis is based on a

MAP/G/1 queueing model.

- Chapter 3 verifies the accuracy of our analysis by comparison of the numerical and simulation results. It also evaluates EDCF with exponential backoff contention window size and its performance under burst error channel by simulation.
- Chapter 4 presents our proposed group polling algorithm.
- Chapter 5 describes the proposed software implementation model and compares its performance with the hardware implementation model.
- Chapter 6 concludes our contributions.

Chapter 2

Performance Analysis of EDCF

The brain is wider than the sky.
For put them side by side.
The one the other will contain
with ease and you beside.
– Emily Dickinson

2.1 Model Description

2.1.1 Network Architecture

In this chapter, we will analyze the throughput and delay for enhanced distributed coordination function (EDCF) of IEEE 802.11 standard. In the analysis, only EDCF is considered. That is, the contention free period is assumed zero. There is one QoS access point in a QBSS, and all stations in this service set are QSTAs. Thus in the following, when mentioning a station or an access point, it is a QSTA or a QoS access point.

All stations in the service set are ideally synchronized without sending beacon frames. The backoff procedure is simplified such that the limit of the contention

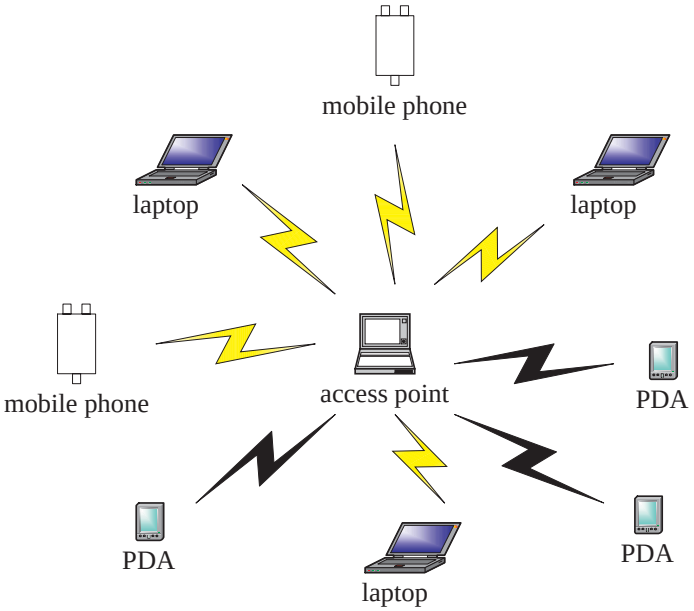


Figure 2.1: Network architecture.

window size is always $CW_{min}(l)$ for frames of priority l. In this chapter, the channel is ideal and error-free such that the only unsuccessful transmission is due to collision. Burst error channel will be discussed in Sec. 3.2. In addition, there are no hidden terminals. Also, no fragmentation is implemented, i.e. all maximum service data unit (MSDU) is equal to maximum protocol data unit (MPDU). Each MPDU contains a MAC header and a payload. Let T_{frame} be the time needed for a MPDU transmission and $T_{payload}$ be the time needed for the transmission of a payload. In the analysis, control frames and acknowledged frames are neglected. Though obviously these frames can be considered with appropriate traffic models together with little changes to the analysis, the contention and transmission duration of them are small enough compared to frames with size T_{frame} and can be neglected. This assumption appears more correct when the size of MPDU is much larger than that of control frames and ACK frames. With the above assumptions, the idle time remains only $AIFS(l)$. They are set by QoS Parameter Set element

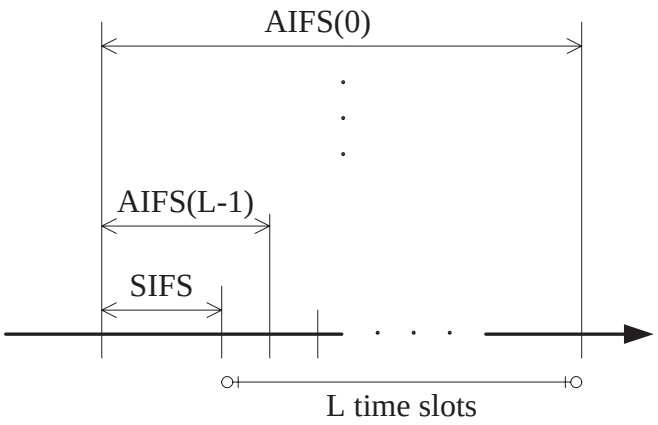


Figure 2.2: Idle time for each priority.

in beacon frames in the draft standard. The $AIFS(l)$ are separated by one slot time (T_{slot}) and $AIFS(L-1) = SIFS + T_{slot}$ as illustrated in Figure 2.2.

Considering a queue in a station, frames start contention in *First-Come-First-Serve* (FCFS) manner. That is, the following frame in the queue does not start to join contention until the frame ahead successfully accesses the channel. Non-preemptive is also assumed. Assume only two states, either idle or busy is known to each station. No collision detection is performed. There are N stations in a service set.

A station generates different prioritized frames with priority 0 to $L-1$, where $L-1$ is the highest priority. Each station holds at least L separated queues for each priority. Each queue has infinite buffer size. As mentioned in Sec. 1.1.1, collisions within a station are resolved such that the higher priority queue gets the transmission opportunity and the lower priority queues act as if there were an external collision. Assume the retry count limit is infinite for each frame, thus no frame is discarded before being successfully transmitted.

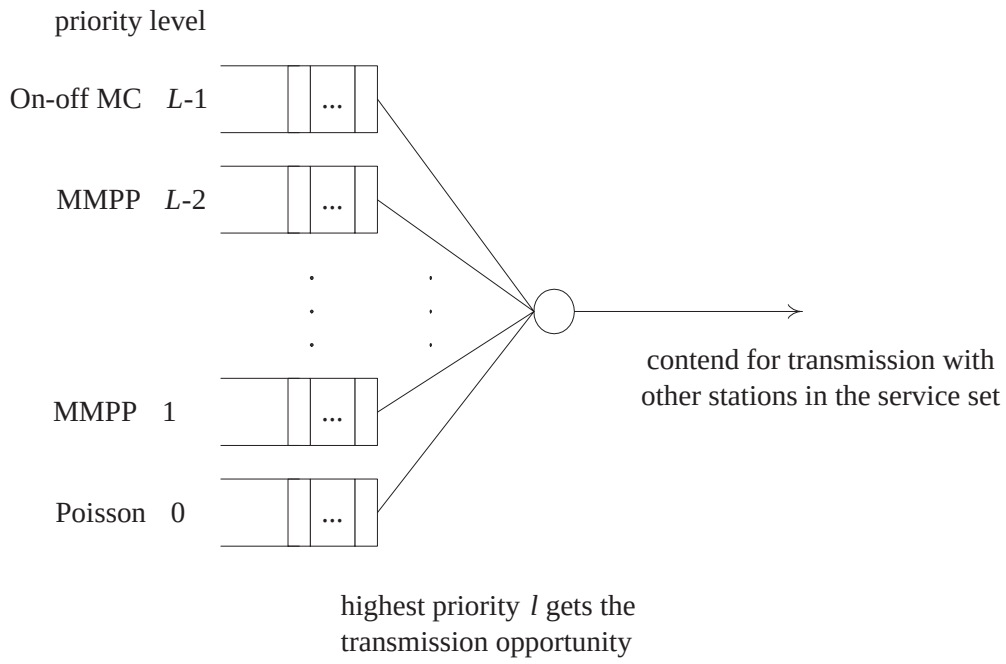


Figure 2.3: The behavior of the L queues in a station.

2.1.2 Traffic Model

The purpose of EDCF is to support LAN applications with QoS requirements, including the transport of voice, audio and video over IEEE 802.11 wireless LANs. Voice traffic is mapped to the highest delivery priority, $L-1$, since humans are more sensitive to hearing than vision. Voice traffic is modeled by an on-off Markov chain. [14], [15]. Data traffic, the best-effort traffic, is mapped to the lowest delivery priority and is modeled by a Poisson process with rate λ_d. Each traffic of delivery priority $L-2$ to 1 is modeled by a two-state Markov modulated Poisson process (MMPP), respectively. These traffic sources include video, audio, etc. Video sources may contain one or more priorities, which is scheduled by the station. The traffic models of each priority and the operation of a station is summarized in Figure 2.3.

For later use, we discuss these traffic models in the following. All traffic models used here, except a single voice traffic model, are special classes of a Markovian

arrival process (MAP).

The Markovian arrival process (MAP)

The following descriptions about the MAP are made based on [16].

Consider an $(m+1)$-state continuous time Markov process, for which the states $\{0,1,\ldots,m-1\}$ are transient and the state $\{m\}$ is absorbing. The arrival process is constructed by: the Markov process evolves until absorption occurs. The epoch of absorption corresponds to an arrival in the arrival process. The mean sojourn time in the transient state $i(0 \le i \le m-1)$ is exponentially distributed with parameter λ_i. When the sojourn time elapsed, the Markov process either enters the absorbing state (which corresponds to an arrival of the MAP) and is instantaneously restarted in the transient state j with probability $p_{ij}, (0 \le j \le m-1)$ or immediately enters the transient state j with probability $q_{ij}, (0 \le j \le m-1, i \ne j)$. Note that

$$\sum_{j=0, j\ne i}^{m-1} q_{ij} + \sum_{j=0}^{m-1} p_{ij} = 1, \quad 1 \le i \le m$$

Define

$$\begin{aligned} D_{ij} &= \lambda_i p_{ij}, \quad 0 \le j \le m-1 \\ C_{ij} &= \lambda_i q_{ij}, \quad 0 \le i, j \le m-1, i \ne j \\ C_{ii} &= -\lambda_i \end{aligned}$$

Then the MAP is parameterized by matrix C and D.

Equivalently, the MAP is a semi-Markov arrival process with transition probability matrix $F(\cdot)$

$$F(x) = \int_0^x e^{Cu} du D \tag{2.1}$$

$$= [1 - e^{Cx}(-C^{-1})]D \tag{2.2}$$

The superposition of n MAPs is still an MAP. The matrices C and D of the

composite MAP are calculated from the individual matrices C_i and D_i as follows.

$$\begin{aligned} C &= C_1 \oplus C_2 \oplus \cdots \oplus C_n, \\ D &= D_1 \oplus D_2 \oplus \cdots \oplus D_n \end{aligned} \tag{2.3}$$

where $\oplus$ is the Kronecker sum [17].

The Markov modulated Poisson process (MMPP)

The MMPP is a doubly stochastic Poisson process [18]. It is constructed by varying the arrival rate of a Poisson process according to an irreducible continuous time Markov chain which is independent of the arrival process. When the Markov chain is in state *i*, arrivals occur according to a Poisson process with rate λ_i. A MMPP is parameterized by matrix Q and Λ, where Q is the infinitesimal generator matrix of the *m*-state modulating Markov chain, and Λ is $diag(\lambda_1, \lambda_2, \ldots \lambda_m)$.

Consider a MMPP. Let π be the steady-state probability vector of the modulating Markov process. Then π is given by solving the equations $\pi Q = 0$ and $\pi e = 1$. The mean arrival rate of the MMPP is Poisson with rate

$$\overline{\lambda} = \pi Q e \tag{2.4}$$

where $e = (1, 1, \ldots, 1)^T$.

The MMPP is a special class of MAP. It is equivalent to MAP with

$$C = Q - \Lambda, \quad D = \Lambda$$

On-off Markov Chain

A single voice source has been widely modeled by an on-off Markov chain [14] as illustrated in Figure 2.4, where r_t^{-1} and r_s^{-1} denotes the mean sojourn time in state talkspurt and silence, respectively. In state talkspurt, voice frames are generated at constant rate, λ_t. In state silence, no frames arrive.

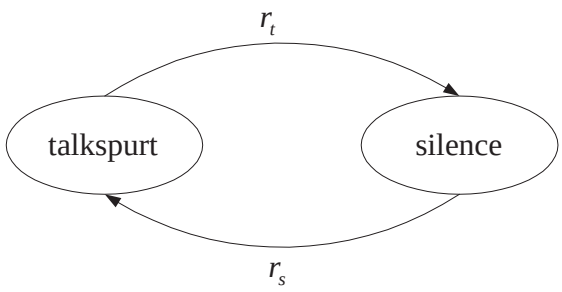


Figure 2.4: Voice source model.

The probability that a voice source is in state talkspurt and has frame arrivals during time T is

$$q_{L-1,on} = \begin{cases} \frac{r_s}{r_t+r_s}\lambda_t T, & \text{if } \lambda_t T < 1 \\ \frac{r_s}{r_t+r_s}, & \text{otherwise} \end{cases} \tag{2.5}$$

The aggregate voice frame arrival rate is a modulated process obtained by modulating the individual voice source packet rate by the number of voice sources in their talkspurt, which is itself a correlated process. It is approximated by an Markov modulated Poisson process (MMPP) [19]. The approximating stream is chosen such that several of its statistical characteristics identically match those of the original superposition.

Interrupted Poisson process (IPP)

An IPP is equivalent to a two-state MMPP with one arrival rate being zero, which means there is no arrival occurs when the modulating Markov chain is in state 2. Thus, $\Lambda = diag(\lambda, 0)$.

Poisson Process

For $C = -\lambda$ and $D = \lambda$, the MAP is a Poisson process.

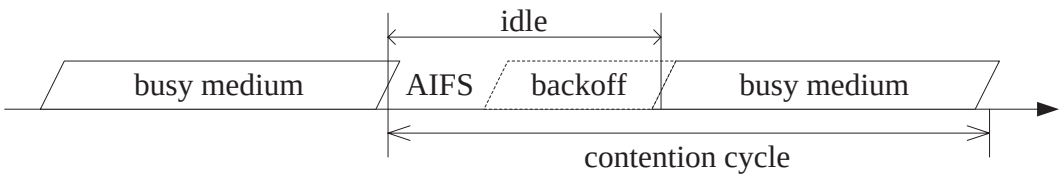


Figure 2.5: Definition of a contention cycle.

2.2 Throughput Analysis

2.2.1 Overview

In this section, we will analysis the throughput of EDCF based on the infrastructure network architecture in Figure 2.1. The definition of a contention cycle is shown in Figure 2.5. A contention cycle includes idle and busy intervals. The idle interval contains the time for sensing channel and the time for backoff. The busy interval may be a successful transmission or a collision.

The throughput analysis focuses on the operation in a service set instead of that in a station. Thus we assume that there are N_l stations having frames of highest priority l at the beginning of each contention cycle, where $N = \sum_{l=0}^{L-1} N_l$. For simplicity, the arrival rate all MMPP is approximated by the mean Poisson arrival rate $\overline{\lambda_l}$ as computed in (2.4). Note that the N_l stations having highest priority l is not specific, but may be "any" N_l stations out of the N stations.

The behavior of the service set can be modeled by a duscrete-time Markov chain with state n denoting the number of active stations. The state is embedded at the beginning of each contention cycle. The word active means that a station has new arrivals or backlogged frames.

When there are v_l active stations having highest priority l (we call it a l^{th} station) at the beginning of current contention cycle, the number of l^{th} active stations after current contention cycle is binomial distributed with parameter $(N_l -$

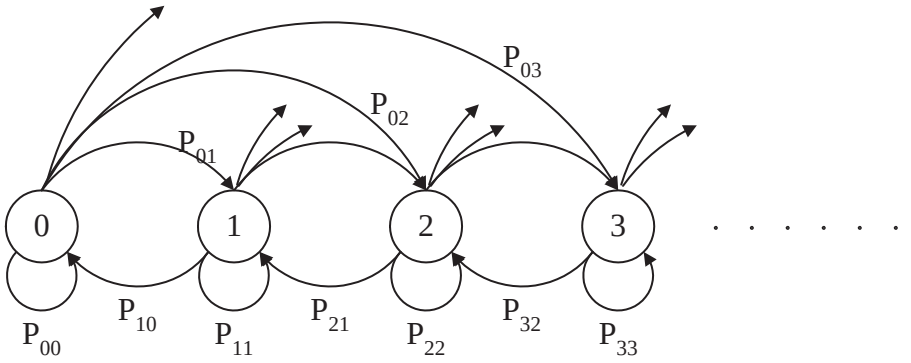


Figure 2.6: Discrete-time Markov chain for throughput analysis.

$v_l + 1, q_{l,on})$ given that current transmission is a successful transfer of priority l, or $(N_l - v_l, q_{l,off})$ given that it is collided or deferred. Note that $q_{l,on}$ is the probability that traffic of priority l arrives during current contention cycle. Let T be the length of current contention cycle. For $0 \leq l \leq L-2$,

$$q_{l,on} = 1 - e^{-\overline{\lambda}_l T}, \qquad q_{l,off} = 1 - q_{l,on}$$

For $l = L-1$, $q_{l,on}$ is shown in (2.5).

Since there is only one access point, the state can decrease at most one per transition, but can increase by an arbitrary amount. This is illustrated in Figure 2.6. In the following, we use the notation $\lfloor x,y \rfloor$ to represent the minimum number of x and y with the constraint that x and y are both nonnegative, that is,

$$\lfloor x,y \rfloor = min(x,y) \geq 0$$

2.2.2 Detailed Derivations

In order to calculate the transition probability, we should condition on which priority gets the transmission opportunity in current contention cycle and current transmission is successful or collided. Then the transition probability from state i to j

is

$$P_{ij}=\begin{cases} p_{on,0}(j) & \\ & i=0, 0\le j\le N \\ \sum_{l=0}^{L-1}\sum_{v_l}\cdots\sum_{v_0} P_i(l;\{v_m\}_{m=0}^{L-1}) \sum_{sp=0}^{l} P_{succ}^{(i,l;\{v_m\}_{m=0}^{L-1})}(sp)\, p_{off}(sp,\{v_m\}_{m=0}^{L-1}) & \\ & 1\le i\le N, j=i-1 \\ \sum_{l=0}^{L-1}\sum_{v_l}\cdots\sum_{v_0} P_i(l;\{v_m\}_{m=0}^{L-1})\cdot & \\ \quad\Big\{\sum_{sp=0}^{l} P_{succ}^{(i,l;\{v_m\}_{m=0}^{L-1})}(sp)\, p_{on,succ}(sp,\{v_m\}_{m=0}^{L-1}, j-i+1) & \\ \qquad +P_{coll}^{(i,l;\{v_m\}_{m=0}^{L-1})} p_{on,coll}(l,\{v_m\}_{m=0}^{L-1}, j-i)\Big\} & \\ & 1\le i\le N, j\ge i \\ \sum_{l=0}^{L-1}\sum_{v_l}\cdots\sum_{v_0} P_i(l;\{v_m\}_{m=0}^{L-1})\cdot & \\ \quad\Big\{\sum_{sp=0}^{l} P_{succ}^{(i,l;\{v_m\}_{m=0}^{L-1})}(sp)\, p_{on,succ}(sp,\{v_m\}_{m=0}^{L-1}, 1)+P_{coll}^{(i,l;\{v_m\}_{m=0}^{L-1})}\Big\} & \\ & i=N, j=i \end{cases} \tag{2.6}$$

The probability $P_i(l;\{v_m\}_{m=0}^{L-1})$ represents when there are i active stations at the beginning of current contention cycle, the highest priority is l, and there are $v_m (0\le m\le L-1)$ stations having frames of priority m. Obviously,

$$v_m=\begin{cases} 0, & \text{if } m>l \\ v_m, & \text{otherwise} \end{cases} \tag{2.7}$$

and

$$\sum_{m=0}^{L-1} v_m = i \tag{2.8}$$

The probability that v_m stations become active in this contention cycle is binomial distributed with parameter $(N_m, q_{m,on})$. The total probability is independent of

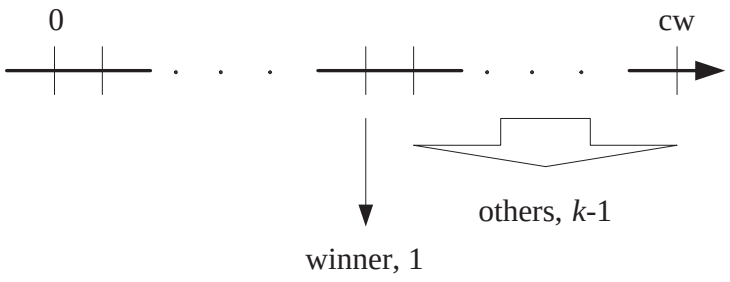


Figure 2.7: Calculation of successful probability with one priority. The station which transmits its frame successfully is called the winner.

each priority. Thus,

$$P_i(l;\{v_m\}_{m=0}^{L-1}) = \prod_{m=0}^{L-1}\binom{N_m}{v_m}(q_{m,on})^{v_m}(1-q_{m,on})^{N_m-v_m}\cdot$$
$$\left\{\sum_{x_{L-1}=0}^{\lfloor i,N_{L-1}\rfloor}\binom{N_{L-1}}{x_{L-1}}(q_{L-1,on})^{x_{L-1}}(1-q_{L-1,on})^{N_{L-1}-x_{L-1}}\cdot\right.$$
$$\sum_{x_{L-2}=0}^{\lfloor i-x_{L-1},N_{L-2}\rfloor}\binom{N_{L-2}}{x_{L-2}}(q_{L-2,on})^{x_{L-2}}(1-q_{L-2,on})^{N_{L-2}-x_{L-2}}\cdots$$
$$\cdots\sum_{x_1=0}^{\lfloor i-\sum_{a=2}^{L-1}x_a,N_1\rfloor}\binom{N_1}{x_1}(q_{1,on})^{x_1}(1-q_{1,on})^{N_1-x_1}\cdot$$
$$\left.\binom{N_0}{x_0}(q_{0,on})^{x_0}(1-q_{0,on})^{N_0-x_0}\right\}^{-1},$$
$$x_0 = i-\sum_{a=1}^{L-1}x_a \le N_0$$

The calculation of successful transmission probability is very complicated. We first show how to derive it when there is only one priority.

A station will transmit its frame successfully if it is the only one with the smallest contention window, which is a uniform random variable drawn from $[0,CW]$. This is illustrated in Figure 2.7.

Thus, when there are k active stations, the successful transmission probability is

$$P_{succ} = \binom{k}{1} p_{cw} \sum_{y=1}^{cw} (1 - y p_{cw})^{k-1} \tag{2.9}$$

and the collision probability is

$$P_{coll} = \sum_{y=1}^{cw} \sum_{a=2}^{k} \binom{k}{a} (p_{cw})^a (1 - y p_{cw})^{k-a}$$

where $p_{cw} = (cw+1)^{-1}$.

The expected time of a successful transmission is equal to the contention interval plus frame transmission interval. When the winner chooses y as its contention window, the length of the backoff interval is (yT_{slot}). Thus,

$$T_{succ} = \left[\binom{k}{1} p_{cw} \sum_{y=1}^{cw} (y-1)(1 - y p_{cw})^{k-1} \right] T_{slot} + T_{frame} \tag{2.10}$$

$$T_{coll} = \left[\sum_{y=1}^{cw} \sum_{a=2}^{k} \binom{k}{a} (p_{cw})^a (y-1)(1 - y p_{cw})^{k-a} \right] T_{slot} + T_{frame} \tag{2.11}$$

In equation (2.11), with the assumption of no collision detection, the transmission interval of collided frames is also T_{frame}. It is intuitively to see that in systems where collision detection is performed, time wasted on collisions could be reduced.

When there are many priorities in a service set, we need to consider which priorities join contention in this contention cycle. This is explained in Figure 2.8. Note that the contention window size of each priority need to be normalized as in (2.14). This is due to the fact that, when different priorities choose the same contention window size, they are different points at the time axis because of different AIFS time.

In the following , we use the notation $P_{succ}^{(i,l;\{v_m\}_{m=0}^{L-1})}(sp)$ to denote the successful transmission probability when there are i active stations in the service set with v_m stations having priority m's frames. The highest priority available now is l, and the successful transmission belongs to priority sp. Note that the fact (2.7) and (2.8) also

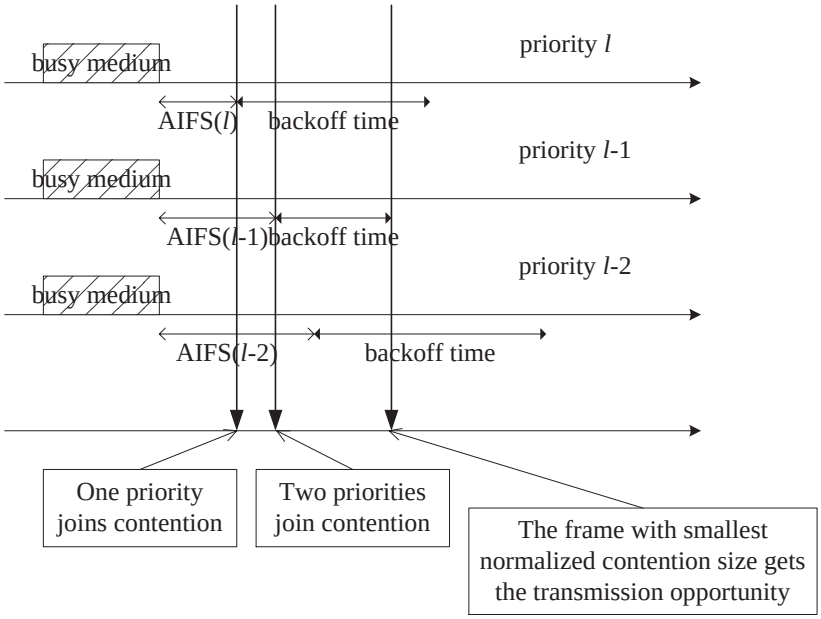


Figure 2.8: Illustration of EDCF when there are many priorities.

holds here. Thus,

$$P_{succ}^{(i,l;\{v_m\}_{m=0}^{L-1})}(sp) = \begin{cases} 0, & \text{if } sp > l \\ \sum_{y_l=1}^{cw_l} p_s(y_l), & \text{if } sp \le l \end{cases} \tag{2.12}$$

When $sp = l$,

$$p_s(y_l) = \begin{cases} \binom{v_l}{1} p_{cw_l}(1-y_l p_{cw_l})^{v_l-1}, \\ \qquad \text{if } AIFS(l) + (y_l-1)T_{slot} < AIFS(l-1) \\ \binom{v_l}{1} p_{cw_l}(1-y_l p_{cw_l})^{v_l-1} \prod_{j=l-k}^{l-1} (1-y_j p_{cw_j})^{v_j}, \\ \qquad \text{if } AIFS(l-k) \le AIFS(l) + (y_l-1)T_{slot} < AIFS(l-k-1) \end{cases} \tag{2.13}$$

where

$$y_j = \frac{AIFS(l) + (y_l-1)T_{slot} - AIFS(j)}{T_{slot}} \tag{2.14}$$

The first equation in (2.13) accounts for that the contention window falls in the interval that only frames of priority l join contention. The second equation accounts

for that the contention window falls in the interval that frames from priority l to priority $l-k$ all join contention.

As Figure 2.8 shows, it is possible that frames of lower priorities successfully access the channel when the highest priority is l.

Thus, when $sp < l$,

$$p_s(y_l) = \begin{cases} 0, \\ \qquad \text{if } AIFS(l) + (y_l - 1)T_{slot} < AIFS(l-1) \\ \binom{v_{sp}}{1} p_{cw_{sp}} (1 - y_{sp} p_{cw_{sp}})^{v_{sp}-1} \prod_{j=l-k, j\neq sp}^{l} (1 - y_j p_{cw_j})^{v_j}, \\ \qquad \text{if } AIFS(l-k) \le AIFS(l) + (y_l - 1)T_{slot} < AIFS(l-k-1) \end{cases} \tag{2.15}$$

At this stage, we could find that during the AIFS time of lower priority, traffic of higher priority acts if only themselves were in the service set. In other words, traffic of higher priority sees a lightly loaded network, which is common in priority networks [20], [21], [22], [23], [24]. However, the problem that traffic of lower priority suffers infinite delay when total load of higher priority is high can be anticipated.

Applying the same procedures, we could derive the collision probability.

$$P_{coll}^{(i,l;\{v_m\}_{m=0}^{L-1})} = \sum_{y_l=1}^{cw_l} p_c(y_l) \tag{2.16}$$

where

$$p_c(y_l) = \begin{cases} \sum_{b=2}^{\lfloor i, v_l \rfloor} \binom{v_l}{a_l} (p_{cw_l})^{a_l} (1 - y_l p_{cw_l})^{v_l - a_l}, a_l = b \\ \qquad \text{if } AIFS(l) + (y_l - 1)T_{slot} < AIFS(l-1) \\ \sum_{b=2}^{i} \sum_{a_l=0}^{\lfloor v_l, b \rfloor} \cdots \sum_{a_{l-k+1}=0}^{\lfloor v_{l-k+1}, k - \sum_{j=l-k+2}^{l} a_j \rfloor} \prod_{j=l-k}^{l} \binom{v_j}{a_j} (p_{cw_j})^{a_j} (1 - y_j p_{cw_j})^{v_j - a_j}, \\ \quad a_{l-k} = b - \sum_{j=l-k+1}^{l} a_j \ge 0 \\ \qquad \text{if } AIFS(l-k) \le AIFS(l) + (y_l - 1)T_{slot} < AIFS(l-k-1) \end{cases} \tag{2.17}$$

The first equation in (2.17) accounts for a collision among priority l's frames when the contention window falls in the interval that only frames of priority l join

contention. The second equation accounts for a collision among priority l to priority $l-k$ when the contention window falls in the interval that frames from priority l to priority $l-k$ all join contention.

Therefore,

$$T_{succ}^{(i,l;\{v_m\}_{m=0}^{L-1})}(sp) = \begin{cases} 0, & \text{if } sp > l \\ AIFS(sp) + \sum_{y_l=1}^{cw_l} T_s(y_l) T_{slot} + T_{frame}, & \text{if } sp \leq l \end{cases}$$

where

$$T_s(y_l) = p_s(y_l)(y_l - 1)$$

and

$$T_{coll}^{(i,l;\{v_m\}_{m=0}^{L-1})} = AIFS(l) + \sum_{y_l=1}^{cw_l} T_c(y_l) T_{slot} + T_{frame}$$

where

$$T_c(y_l) = p_c(y_l)(y_l - 1)$$

The expected length of a contention cycle, given i active stations and the highest priority l, is

$$\begin{aligned} T^{(i,l;\{v_m\}_{m=0}^{L-1})} =& P_{coll}^{(i,l;\{v_m\}_{m=0}^{L-1})} T_{coll}^{(i,l;\{v_m\}_{m=0}^{L-1})} \\ &+ \sum_{sp=0}^{l} P_{succ}^{(i,l;\{v_m\}_{m=0}^{L-1})}(sp) T_{succ}^{(i,l;\{v_m\}_{m=0}^{L-1})}(sp) \end{aligned}$$

The probability $p_{(on,type)}(tp, \{v_m\}_{m=0}^{L-1}; s)$ means that there are s newly active stations in current contention cycle under the condition that current transmission is $type$ (successful or collided) and belongs to priority tp ($t$ransmission $p$riority). Also there are v_m stations having frames of priority m at the beginning of current contention cycle. Note that these newly active stations may have traffic arrival from

voice, video or data. Thus,

$$
\begin{aligned}
&p_{on,type}(tp,\{v_m\}_{m=0}^{L-1};s) = \\
&\quad \sum_{x_{L-1}=0}^{\lfloor s,N_{L-1}-v'_{L-1}\rfloor} \binom{N_{L-1}-v'_{L-1}}{x_{L-1}} (q_{L-1,on})^{x_{L-1}}(1-q_{L-1,on})^{N_{L-1}-v'_{L-1}-x_{L-1}} \\
&\quad \sum_{x_{L-2}=0}^{\lfloor s-x_{L-1},N_{L-2}-v'_{L-2}\rfloor} \binom{N_{L-2}-v'_{L-2}}{x_{L-2}} (q_{L-2,on})^{x_{L-2}}(1-q_{L-2,on})^{N_{L-2}-v'_{L-2}-x_{L-2}} \cdots \\
&\quad \cdots \sum_{x_1=0}^{\lfloor s-\sum_{a=2}^{L-1}x_a,N_1-v'_1\rfloor} \binom{N_1-v'_1}{x_1} (q_{1,on})^{x_1}(1-q_{1,on})^{N_1-v'_1-x_1} \\
&\qquad \binom{N_0-v'_0}{x_0} (q_{0,on})^{z}(1-q_{0,on})^{N_0-v'_0-x_0}, \\
&\qquad\qquad\qquad\qquad\qquad\qquad x_0 = i-\sum_{a=1}^{L-1} x_a \le (N_0-v'_0)
\end{aligned}
$$

where

$$
v'_m = \begin{cases} v_m - 1, & \text{if } m = tp \text{ and } type = succ \\ v_m, & \text{otherwise} \end{cases}
$$

The probability $p_{off}(tp,\{v_m\}_{m=0}^{L-1})$ is that no frame arrives and no new station become backlogged when current transmission belongs to priority tp.

$$
p_{off}(tp,\{v_m\}_{m=0}^{L-1}) = \prod_{m=0}^{L-1} (q_{m,off})^{N_m-v'_m}
$$

With transition probabilities in (2.6) and global balance equations, we could derive the steady state probability $\pi(i)$ at each state. Since the successful transmission probability is conditioned on the number of active stations and the priority, the expected time of a useful transmission (not including header information), $\overline{U}$, in one

contention cycle is

$$\overline{U} = \sum_{i=1}^{N} \pi(i) \sum_{l=0}^{L-1} \sum_{v_l=1}^{i} \cdots \sum_{v_0} P_i(l; \{v_m\}_{m=0}^{L-1}) \sum_{sp=0}^{l} P_{succ}^{(i,l;\{v_m\}_{m=0}^{L-1})}(sp) T_{payload}$$

and the expected time of a contention cycle, $\overline{T}$, is

$$\overline{T} = \pi(i) T_0 + \sum_{i=1}^{N} \pi(i) \sum_{l=0}^{L-1} T(i,l)$$

where

$$T(i,l) = \sum_{v_l=1}^{i} \cdots \sum_{v_0} P_i(l; \{v_m\}_{m=0}^{L-1}) T^{(i,l;,\{v_m\}_{m=0}^{L-1})}$$

Finally,

$$throughput = \frac{\overline{U}}{\overline{T}}$$

2.3 Delay Analysis

2.3.1 Overview

The waiting time of a frame is the duration from its generation to the time it is successfully transmitted. Thus, a frame suffers delay from two aspects, the station and the network. That is, a frame needs to wait for the contention and transmission interval of frames from higher priority queues in its station and it needs to wait for its successful transmission after joining contention in the network. If the frame did not get the transmission opportunity or current transmission were not successful, it suffers additional delay from higher priority queues in its station. In this section, for simplicity, we consider that once a frame is generated, it is the highest priority in its station. Hence we only consider the delay from network. Assume there are L virtual queues in the network. The queued frames in the virtual queue are the head frame

of the highest priority queue in each station. This is illustrated in Figure 2.9. In a service set, though it is possible that a lower priority frame gets the transmission opportunity before higher priority frames, we assume that this case would never happen. In fact, this assumption is what we want to achieve in order to provide QoS guarantees for delay-sensitive traffic and can be achieved by properly choosing the system parameters $AIFS(l)$ and the contention window limits. We analyze the delay under the restriction that the total load of the network is less than one, i.e. $\rho_{tot} < 1$. The service time includes contention and transmission intervals. Thus, the AP is work conserving. The service time distribution of priority l is denoted by $G_l(t)$ with LST $G_l^*(s)$ and finite mean g_l. We could derive g_l when there are i active stations in the service set by the analytical results of Sec. 2.2.

$$g_l = \sum_{v_l=0}^{i} \cdots \sum_{v_0} P_i(l;\{v_m\}_{m=0}^{L-1}) \cdot \left\{ P_{succ}^{(i,l;\{v_m\}_{m=0}^{L-1})}(l) T_{succ}^{(i,l;\{v_m\}_{m=0}^{L-1})}(l) + P_{coll}^{(i,l;\{v_m\}_{m=0}^{L-1})} \left[T_{coll}^{(i,l;\{v_m\}_{m=0}^{L-1})} + g_l \right] \right\} \tag{2.18}$$

by assuming that the lower priority frames will never get the transmission opportunity before the higher priority frames. The first term in (2.18) corresponds to a successful transmission of priority l's traffic and the second term corresponds to a collision. Note that when a collision occurs, the frame needs another g_l to complete its transmission.

With some manipulations, (2.18) becomes

$$g_l = \frac{\sum_{v_l=0}^{i} \cdots \sum_{v_0} P_i(l;\{v_m\}_{m=0}^{L-1}) T^{(i,l;\{v_m\}_{m=0}^{L-1})}}{1 - \sum_{v_l=0}^{i} \cdots \sum_{v_0} P_i(l;\{v_m\}_{m=0}^{L-1}) P_{coll}^{(i,l;\{v_m\}_{m=0}^{L-1})}}$$

Since there are K_l stations having priority l as its highest priority, and a station has exactly one frame joining contention at each time, the virtual queues of the network is finite with maximum size K_l for priority l. Queue size exceeds this value is not possible.

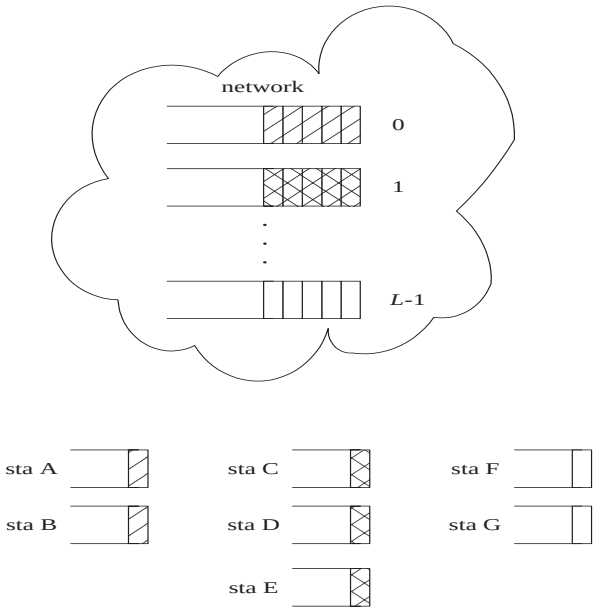


Figure 2.9: Illustration of the virtual queues in the network for delay analysis.

Traffic models are described in Sec. 2.1.2. Since various process could be generalized to MAP, we use MAP as the arrival process. The procedures we apply is a novel but basic approach to analyze a MAP/G/1 queue [16], [25], [26], [27]. In [27], the author analyze an MMPP/G/1 queue with dynamic priority queueing. We apply this method to solve our system, but the priority is static and the behavior of our system is different from [27]. We first find the stationary queue length at departures, then derive the queue length distribution at an arbitrary time. Finally, it comes out the averaged delay. For simplicity, we consider only two priorities now. Consider two MAP with parameters (C_0', D_0') and (C_1', D_1'). Each MAP is calculated from the superposition of N_l identical and independent MAP representing the arrival process at the l^{th} station. That is,

$$C_0' = \underbrace{C^0 \oplus C^0 \oplus \cdots \oplus C^0}_{\text{the number of } C^0 \text{ is } K_0}$$

and

$$D_0' = \underbrace{D^0 \oplus D^0 \oplus \cdots \oplus D^0}_{\text{the number of } D^0 \text{ is } K_0}$$

where (C_0', D_0') parameterize an arrival process of priority 0 and is a MAP. Note that the superposed arrival process is still a MAP. Let M_l be the number of states in the underlying Markov process governing frames of priority l, then $M = M_1 M_2$ is the number of states in the underlying Markov process governing the superposed arrival process. This analysis can be extended to L priorities.

2.3.2 Preliminaries

As (2.3) shows, the superposition of two MAP is still a MAP. Let

$$\begin{aligned} D_0 &= D_0' \otimes I_1, \qquad & D_1 &= I_0 \otimes D_1' \\ C &= C_0' \oplus C_1', \qquad & D &= D_0 + D_1 \end{aligned} \tag{2.19}$$

where $\otimes$ denotes the Kronecker product [17], and I_l is the identity matrix of the same order as D_l.

Let $\widetilde{M}^l(t)$ be the number of arrivals of priority l during $(0,t]$, and $\widetilde{M}(t) = \widetilde{M}^0(t) + \widetilde{M}^1(t)$. Let $J(t)$ be the state at time t of the underlying Markov process governing the superposed arrival process. The counting function is

$$P_{ij}(n_0, n_1, t) = P\{\widetilde{M}^0(t) = n_0, \widetilde{M}^1(t) = n_1, J(t) = j | \widetilde{M}(0), J(0) = i\}$$

The matrices $P(n,t)$ satisfy the Chapman-Kolmogorov forward equations.

$$P_{ij}'(n_0, n_1, t) = P_{ij}(n_0, n_1, t)C + P_{ij}(n_0 - 1, n_1, t)D_0 + P_{ij}(n_0, n_1 - 1, t)D_1$$

where $P_{ij}(-1, n_1, t)$ and $P_{ij}(n_0, -1, t)$ are zero matrices. The matrix generating

function of $P_{ij}(n_0,n_1,t)$ is

$$\begin{aligned} P^*(z_0,z_1,t) &= \sum_{n_0=0}^{\infty}\sum_{n_1=0}^{\infty} P(n_0,n_1,t) z_0^{n_0} z_1^{n_1} \\ &= e^{(C+z_0D_0+z_1D_1)t} \end{aligned} \tag{2.20}$$

2.3.3 The stationary queue length at departure instants

The embedded Markov renewal process at departure epochs is defined as follows. Let τ_n be the epoch of the n^{th} departure from the queue, with $\tau_0 = 0$. We further define N_n^l to be the queue length of priority l at time τ_n, and J_n to be the state of the underlying Markov process at time τ_n. Then $\{(N_n^0, N_n^1, J_n, \tau_{n+1}-\tau_n) : n \geq 0\}$ forms a semi-Markov chain at departure epochs on the state space $\{0,1,\ldots,K_0\} \times \{0,1,\ldots,K_1\} \times \{1,\ldots,M\}$ with the state transition probability matrix $\widetilde{\mathbf{Q}}(x)$ defined in (2.21). The stationary queue length x is

$$\begin{aligned} x_{k_0,k_1,i} &= \lim_{n\to\infty} P\{N_n^0 = k_0, N_n^1 = k_1, J_n = i\} \\ x_{k_0,k_1} &= (x_{k_0,k_1,1}, x_{k_0,k_1,2}, \ldots, x_{k_0,k_1,m}) \\ x_{k_0} &= (x_{k_0,0}, x_{k_0,1}, \ldots, x_{k_0,K_1}) \\ x &= (x_0, x_1, \ldots, x_{K_1}) \end{aligned}$$

Let (i,j) be the state $(N^0 = i, N^1 = j)$ before a departure occurs and (i',j') be the state $(N^0 = i', N^1 = j')$ just after a departure. Then

$$\widetilde{\mathbf{Q}}(x) = \begin{cases} \widetilde{\mathbf{A}}'_{0,0}(x), & i' = i = 0, j' = j = 0 \\ \widetilde{\mathbf{A}}_{0,0}(x), & i' = i-1, j' = j = 0 \\ 0, & i' = i-1, j' \neq j \\ \widetilde{\mathbf{A}}_{i'-i+1,0}(x), & i' \geq i, j' = j = 0 \\ \widetilde{\mathbf{A}}_{i'-i,0}(x), & i' \geq i, j' = j-1 \\ \widetilde{\mathbf{A}}_{i'-i,j'-j+1}(x), & i' \geq i, j' \geq j \neq 0 \end{cases} \tag{2.21}$$

where $\widetilde{\mathbf{A}}'_{n_0,n_1}(x)$ and $\widetilde{\mathbf{A}}_{n_0,n_1}(x)$ are defined by

$[\widetilde{\mathbf{A}}_{n_0,n_1}(x)]_{rs} = P\{$given a departure at time 0, which left at least one frame in either

priority-0 queue or priority-1 queue or both and the arrival process in state r, the next departure occurs no late than time x with the arrival process in state s, and during that service time there were n_0 arrivals from priority 0 and n_1 arrivals from priority 1$\}$,

$[\widetilde{\mathbf{A}}'_{n_0,n_1}(x)]_{rs} = P\{$given a departure at time 0, which left both priority-0 queue and priority-1 queue empty and the arrival process in state r, the next departure occurs no late than time x with the arrival process in state s, and during that service time there were n_0 arrivals from priority 0 and n_1 arrivals from priority 1$\}$.

Then ([16], [18])

$$\begin{aligned} [\widetilde{\mathbf{A}}_{n_0,n_1}(x)]_{rs} &= \int_0^x P(n_0,n_1,t)dG(t) \\ [\widetilde{\mathbf{A}}'_{n_0,n_1}(x)]_{rs} &= (-C^{-1})D\widetilde{\mathbf{A}}_{n_0,n_1} \end{aligned} \tag{2.22}$$

The steady state vector x of the Markov chain $\{N_n^0, N_n^1, J_n\}$ is given by solving

$$x\widetilde{\mathbf{Q}} = x, \qquad xe = 1$$

2.3.4 The queue length distribution at an arbitrary time

In order to obtain the mean waiting time of each frame, we need to know the stationary distribution of each queue length at an arbitrary time. This is derived by the supplementary variable technique [28], [25], [29]. There are two supplementary variables that can be chosen, one of which is the elapsed service time and the other is the remaining service time. Here we choose the remaining service time as the supplementary variable, that is, from the joint distribution of the queue length and the remaining service time for the frame in service if the server is busy and the remaining vacation time if the server is idle. Then we have much simpler calculations. Let $N^l(t)$ be the queue length of priority l at time t (excluding the frame in service), and $J(t)$ be the state at time t of the underlying Markov chain governing the superposed arrival process. At an arbitrary instant, the station can be characterized

by

$$\varepsilon(t) = \begin{cases} 0, & \text{if the server is busy at time } t \\ 1, & \text{if the server is idle at time } t \end{cases}$$

Define

$$y(0,j) = \lim_{t\to\infty} P(\{N^0(t) = 0, N^1(t) = 0, J(t) = j, \varepsilon(t) = 0\})$$
$$y_0 = (y(0,1), y(0,2), \ldots, y(0,M))$$

be the steady state probability that both queues are empty and the server is idle, and y_n^l be the steady state probability that there are n frames in the queue with priority l and the server is busy,

$$y_n^l(j) = \lim_{t\to\infty} P(\{N^l(t) = n, J(t) = j, \varepsilon(t) = 1\})$$
$$y_n^l = (y_n^l(1), y_n^l(2), \ldots, y_n^l(m)), \quad l = 0,1, 0 \le n \le K_l, 1 \le j \le M$$

Note that it is impossible for the case that there are frames queueing in the buffer and the server is idle because we consider a work conserving system here.

The rest of our work is to find $\mathbf{y}$. We first find the vector y_0 that the server is idle. Relating the stationary queue length at an arbitrary time to the stationary queue length at departures by the *key renewal theorem* [30], we have

$$y_0 = \frac{1}{E} x_{0,0}(-C)^{-1} \tag{2.23}$$

where

$$E = x_{0,0}(-C)^{-1}e + g$$

is the mean interdeparture time of frames. The derivation of these equations are shown in Appendix A.1.

Next we will find the queue length distribution for frames of each priority at an arbitrary time when the server is busy. This is derived by the supplementary method described earlier. Thus

$$y_n^l = P\{\varepsilon(t) = 1\} \int_0^\infty \alpha_l(n,\tau) d\tau \tag{2.24}$$

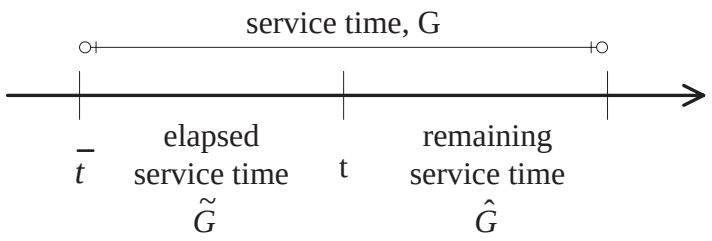


Figure 2.10: Definition of the notations for delay calculation.

where $\alpha_l(n,\tau)d\tau$ is the joint probability distribution of the queue length of priority l and the remaining service time for the frame in service at an arbitrary time τ. The notations needed in the following are illustrated in Figure 2.10.

Let $\widehat{G}$ be the remaining service time for the frame in service. Then

$$\alpha_l(n,\tau) = (\alpha_l(n,1,\tau),\ldots,\alpha_l(n,M,\tau))$$
$$\alpha_l(n,j,\tau)d\tau = P\{N^l(\tau)=n, J_t=j, \tau < \widehat{G} \le \tau + d\tau | \varepsilon(t)=1\}$$

and the LST of $\alpha_l(n,j,\tau)$ is

$$\alpha_l^*(n,j,s) = \int_0^\infty e^{-s\tau}\alpha_l(n,j,\tau)d\tau$$
$$\alpha_l^*(n,s) = (\alpha_l^*(n,1,s),\ldots,\alpha_l^*(n,M,s))$$

By setting $s=0$, (2.24) becomes

$$y_n^l = P\{\varepsilon(t)=1\}\alpha_l^*(n,0) \tag{2.25}$$

where $P\{\varepsilon(t)=1\}$ is the probability that the server is busy. From (2.23),

$$P\{\varepsilon(t)=0\} = y_0 e = \frac{1}{E}x_{0,0}(-C)^{-1}e$$
$$P\{\varepsilon(t)=1\} = 1 - P\{\varepsilon(t)=0\} = \frac{g}{E}$$

We evaluate $\alpha_l^*(n,j,s)$ by conditioning on the number of arrivals from priority l in the elapsed service time $\widetilde{G}$. Then the queue length at an arbitrary time t is the

number of arrivals during the elapsed service time plus the queue length at the last departure epoch before t. Consider priority 1 first. Let

$$\begin{aligned}&\beta_1(n',\tau,j'|j)d\tau\\&\quad= P\{n' \text{ arrivals from priority } l \text{ during } \widetilde{G},\, J(t)=j', \tau < \widehat{G} \le \tau + d\tau | J(\overline{t})=j\}\end{aligned}$$

where $\overline{t}$ is the time at the last departure epoch. Note that $\overline{t} \le t$.

The LST of $\beta_1(n',\tau,j'|j)d\tau$ is defined as

$$\begin{aligned}\beta_1^*(n',s,j'|j) &= \int_0^\infty e^{-s\tau}\beta(n',\tau,j'|j)d\tau\\ &= E[e^{-s\widehat{G}}|N^1(\widetilde{G})=n-n']P[N^1(\widetilde{G})=n-n']\end{aligned} \tag{2.26}$$

and the vector $\beta_1^*(n',s)$ is

$$\beta_1^*(n',s) = (\beta_1^*(n',s,j'|j))_{1\le j,j'\le M}$$

Thus

$$\alpha_1^*(n,s) = x_{0,0}(-C^{-1})D\beta_1^*(n,s) + \sum_{m=1}^{K_0} x_{m,0}\beta_1^*(n,s) + \sum_{m=0}^{K_0}\sum_{k=1}^{n+1} x_{m,k}\beta_1^*(n-k+1,s)$$

$$0 \le n \le K_1 \tag{2.27}$$

where the first term accounts for n frames arriving from priority 1 when both queues are empty, and the second term accounts for n frames arriving from priority 1 when there were m frames of priority 0 and no frame of priority 1 at the last departure instant, hence one frame of priority 0 leaves in current departure epoch. The third term in (2.27) accounts for $n-k+1$ frames arriving from priority 1 when there were m frames of priority 0 and k frames of priority 1 at the last departure instant, hence one frame of priority 1 leaves in current departure epoch.

And $\beta_1^*(r,s)$ is ([26])

$$\beta_1^*(r,s) = \frac{1}{g}\left[\sum_{u=0}^{r} A_{\cdot,u}R_{r-u}(s) - G^*(s)R_r(s)\right] \tag{2.28}$$

where

$$R_r(s) = (sI + C)^{-1}[(-D_1)(sI + C)^{-1}]^r \tag{2.29}$$

and

$$A_{\cdot,u} = \sum_{v=0}^{\infty} \int_0^{\infty} P(v,u,t)dG(t)$$

Substituting (2.27), (2.28), and (2.29) for (2.25) and applying some algebraic manipulations , we have

$$\begin{aligned} y_n^1 = \frac{1}{E}\Bigg[& x_{0,0}(-C)^{-1}D\left\{\sum_{u=0}^{n} A_{\cdot,u}[D_1(-C)^{-1}]^{n-u} - C^{-1}[D_1(-C)^{-1}]^n\right\} \\ & + \sum_{m=1}^{K_0} x_{m,0}\left\{\sum_{u=0}^{n} A_{\cdot,u}[D_1(-C)^{-1}]^{n-u} - C^{-1}[D_1(-C)^{-1}]^n\right\} \\ & + \sum_{m=0}^{K_0}\sum_{k=1}^{n+1} x_{m,k}\left\{\sum_{u=0}^{n-k+1} A_{\cdot,u}[D_1(-C)^{-1}]^{n-k+1-u} - C^{-1}[D_1(-C)^{-1}]^{n-k+1}\right\}\Bigg] \\ & \qquad\qquad 0 \le n \le K_1 \end{aligned} \tag{2.30}$$

By a similar method for the queue with priority 0, we have

$$\begin{aligned} \alpha_0^*(n,s) = x_{0,0}(-C^{-1})D\beta_0^*(n,s) + \sum_{k=1}^{n+1} x_{k,0}\beta_0^*(n-k+1,s) + \sum_{m=1}^{K_1}\sum_{k=0}^{n} x_{k,m}\beta_0^*(n-k,s) \\ 0 \le n \le K_0 \end{aligned} \tag{2.31}$$

where the first term accounts for n frames arriving from priority 0 when both queues are empty, and the second term accounts for $n-k+1$ frames arriving from priority 0 when there were k frames of priority 0 and no frames of priority 1 at the last departure instant, hence one frame of priority 0 leaves in current departure epoch. The third term in (2.31) accounts for $n-k$ frames arriving from priority 0 when there were k frames of priority 0 and m frames of priority 1 at the last departure instant, hence one frame of priority 1 leaves in current departure epoch.

And $\beta_0^*(r,s)$ is

$$\beta_0^*(r,s) = \frac{1}{g}\left[\sum_{v=0}^{r} A_{v,\cdot} R_{r-v}(s) - G^*(s) R_r(s)\right] \tag{2.32}$$

where

$$R_r(s) = (sI + C)^{-1}[(-D_0)(sI + C)^{-1}]^r \tag{2.33}$$

and

$$A_{v,\cdot} = \sum_{u=0}^{\infty} \int_0^{\infty} P(v,u,t)\, dG(t)$$

Thus we obtain

$$\begin{aligned} y_n^0 = \frac{1}{E}\Bigg[& x_{0,0}(-C^{-1}) D \left\{ \sum_{v=0}^{n} A_{v,\cdot} [D_1(-C)^{-1}]^{n-v} - C^{-1}[D_1(-C)^{-1}]^n \right\} \\ & + \sum_{k=1}^{n+1} x_{k,0} \left\{ \sum_{v=0}^{n-k+1} A_{v,\cdot} [D_1(-C)^{-1}]^{n-k+1-v} - C^{-1}[D_1(-C)^{-1}]^{n-k+1} \right\} \\ & + \sum_{m=1}^{K_1} \sum_{k=0}^{n} x_{k,m} \left\{ \sum_{v=0}^{n-k} A_{v,\cdot} [D_1(-C)^{-1}]^{n-k-v} - C^{-1}[D_1(-C)^{-1}]^{n-k} \right\} \Bigg] \\ & \qquad\qquad 0 \le n \le K_0 \end{aligned} \tag{2.34}$$

2.3.5 Mean waiting time

Using the queue length distribution at an arbitrary time, we have the mean queue length L_q^l,

$$L_q^1 = \sum_{i=0}^{K_1} i y_i^1 e, \qquad L_q^0 = \sum_{i=0}^{K_0} i y_i^1 e \tag{2.35}$$

where e is a column vector of 1's.

Applying *Little's Theorem* to (2.35), the mean waiting time is obtained.

$$W_1 = \frac{L_q^1}{\overline{\lambda_1}}, \qquad W_0 = \frac{L_q^0}{\overline{\lambda_0}}$$

where $\overline{\lambda_l}$ is defined in (2.4).

Chapter 3

Numerical and Simulation Results of EDCF

The world is not Conclusion.
A species stands beyond.
– Emily Dickinson

3.1 Performance Evaluation under Error-Free Channel

3.1.1 Numerical Results

In this section, numerical results of Chapter 2 are shown. The system parameters are listed in Table 3.1. These system values are those specified for the orthogonal frequency multiplexing modulation (OFDM) physical layer [31]. OFDM system provides many transmission rates. Here it is chosen as 24Mbps. In this chapter, We use a constant frame size, 2348 bytes, which is equal to the maximum MPDU size. Each AIFS time separates one slot time as the draft standard suggests.

The draft standard provides at most 8 priorities. Here, four priorities are con-

Notation	Description	Value
T_{slot}	a slot time	9μ sec
$SIFS$	a SIFS time	16μsec
$AIFS(3)$	an AIFS time of priority 3	25μ sec
$AIFS(2)$	an AIFS time of priority 2	34μ sec
$AIFS(1)$	an AIFS time of priority 1	43μ sec
$AIFS(0)$	an AIFS time of priority 0	52μ sec
λ_d	data arrival rate	50 (1/sec)
MPDU	MAC protocol data unit	2348 bytes
-	payload size	2312 bytes
-	transmission rate	24Mbps

Table 3.1: System parameters.

Notation	Description	Value (m sec)
λ_t^{-1}	interarrival time in state talkspurt	16
r_t^{-1}	mean sojourn time in state talkspurt	352
r_s^{-1}	mean sojourn time in state silence	650

Table 3.2: Parameters for voice source.

sidered. The lowest priority is data traffic and Poisson with rate λ_d. The highest priority is voice traffic as mentioned in Sec. 2.1.2. The parameters for voice sources are listed in Table 3.2. These parameters are suggested by [14].

Traffic of other priorities are modeled by two-state MMPP [18] as shown in Figure 3.1. The rate of these sources are listed in Table 3.3. Note that the mean arrival rate of priority 2 we use here is the MPEG I type, and the mean arrival rate of priority 1 is the trace video rate.

A two-state MMPP has four parameters to be specified. They are computed based on five values: the mean arrival rate of the overall process m_1, the variance of

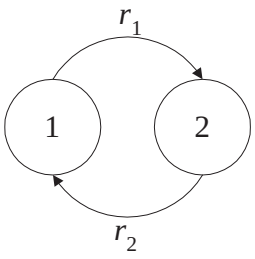


Figure 3.1: Two-state MMPP.

Priority	Mean arrival rate (bps)
2	1.5M
1	350K

Table 3.3: Mean arrival rate for MMPP.

the arrival rate m_2, the third moment of the arrival rate m_3, the time interval $\triangle t$, and the lag-1 autocorrelation coefficient c. Note that $\triangle t$ and c can control the length of the resident time in a state of the Markov chain. For simplicity, we set $m_3 = 0$ and $c = 1/e$. These parameters are related as follows:

$$\tau = -\frac{\triangle t}{\ln c_1} \tag{3.1}$$

$$\delta = \frac{m_3}{\sqrt{m_2^3}} \tag{3.2}$$

$$\eta = 1 + \frac{\delta}{2}[\delta - \sqrt{4+\delta^2}] \tag{3.3}$$

$$\lambda_1 = m_1 - \sqrt{\frac{m_2}{\eta}}, \qquad \lambda_2 = m_1 + \sqrt{\frac{m_2}{\eta}} \tag{3.4}$$

$$r_1 = \frac{\eta}{\tau(1+\eta)}, \qquad r_2 = \frac{1}{\tau(1+\eta)} \tag{3.5}$$

With (3.1) to (3.5) and the mean arrival rate shown in Table 3.3, we could derive the parameters needed to specify an MMPP. The results are listed in Table 3.4 and Table 3.5.

Notation	Description	Value (m sec)
λ_2^{-1}	mean interarrival time in state 2	12.5
λ_1^{-1}	mean interarrival time in state 1	12.538
r_2^{-1}	mean sojourn time in state 2	100
r_1^{-1}	mean sojourn time in state 1	100

Table 3.4: Parameters for MMPP source of priority 2.

Notation	Description	Value (m sec)
λ_2^{-1}	mean interarrival time in state 2	53.4
λ_1^{-1}	mean interarrival time in state 1	54
r_2^{-1}	mean sojourn time in state 2	100
r_1^{-1}	mean sojourn time in state 1	100

Table 3.5: Parameters for MMPP source of priority 1.

The validation of the analytical model is shown in Figure 3.2. It can be seen that the analytical model is extremely accurate. The analytical results (line) practically coincide with the simulation results ($\times$). The simulation results in the plot are obtained with 95% confidence interval (dash). The simulation is an event-driven program, written in C++ programming language, that closely follows all the 802.11e draft protocol details for each independently transmitting stations.

In all figures shown in this chapter, unless otherwise specified, we use 5 stations for each priority.

3.1.2 Simulation Results

We have analyzed the throughput and delay of EDCF with constant backoff window size. In this section, the simulation results of throughput and delay with five exponential backoff window size are shown. The backoff factor is a multiple that a station choose to increase the contention window size when collision occurs. Two

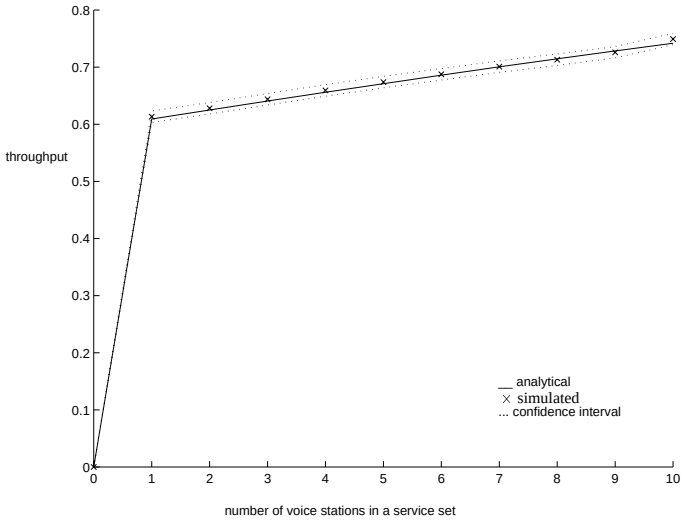


Figure 3.2: Validation of the analytical model.

Priority	CWmin	CW1	CW2	CW3	CW4
3	21	42	84	168	336
2	42	84	168	336	672
1	84	168	336	672	1344
0	168	336	672	1344	2688

Table 3.6: System parameters for simulation. (backoff 2)

groups of backoff window size are listed in Table 3.6 and Table 3.7, where in Table 3.6 backoff factor 2 is chosen and in Table 3.7 backoff factor of 1.5 is chosen. Other assumptions and parameters are the same as those in Sec. 3.1.1.

Each case in our simulation runs 20 times to take the ensemble average, during each time of which it runs for 200 seconds to take the time average.

3.1.3 Conclusions

The EDCF operations, as mentioned in Sec. 2.2, achieves QoS transfers by letting traffic of higher priority see a lightly loaded network. This is accomplished by sens-

Priority	CWmin	CW1	CW2	CW3	CW4
3	21	32	48	72	108
2	32	48	72	108	162
1	48	72	108	162	243
0	72	108	162	243	364

Table 3.7: System parameters for simulation. (backoff 1.5)

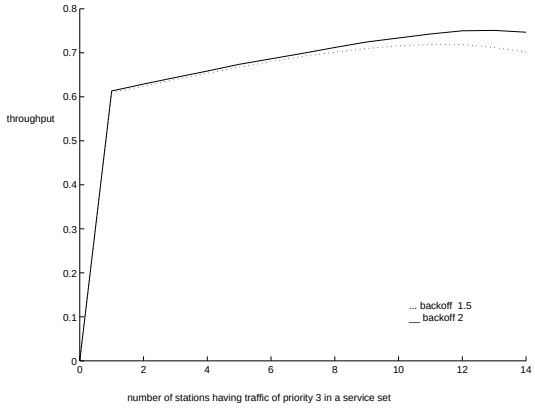
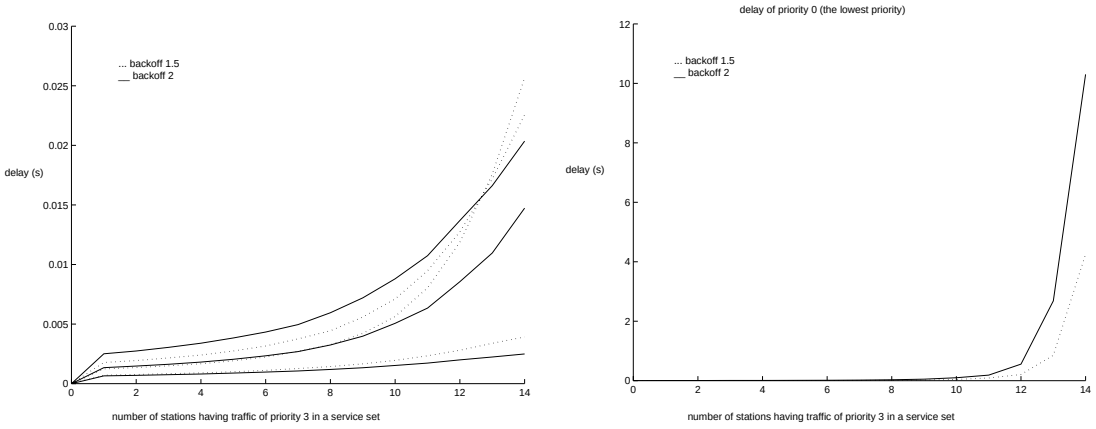


Figure 3.3: Effect of backoff window size on throughput



(a) (b)

Figure 3.4: Effect of backoff window size on delay

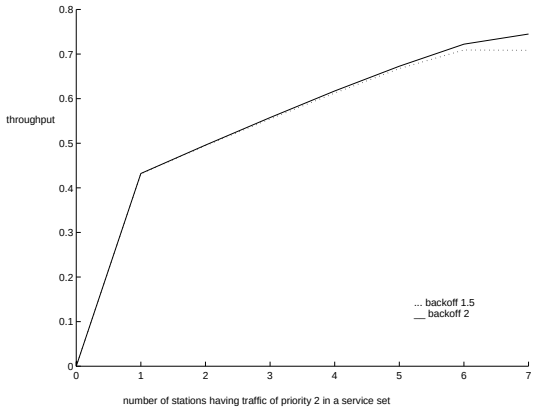
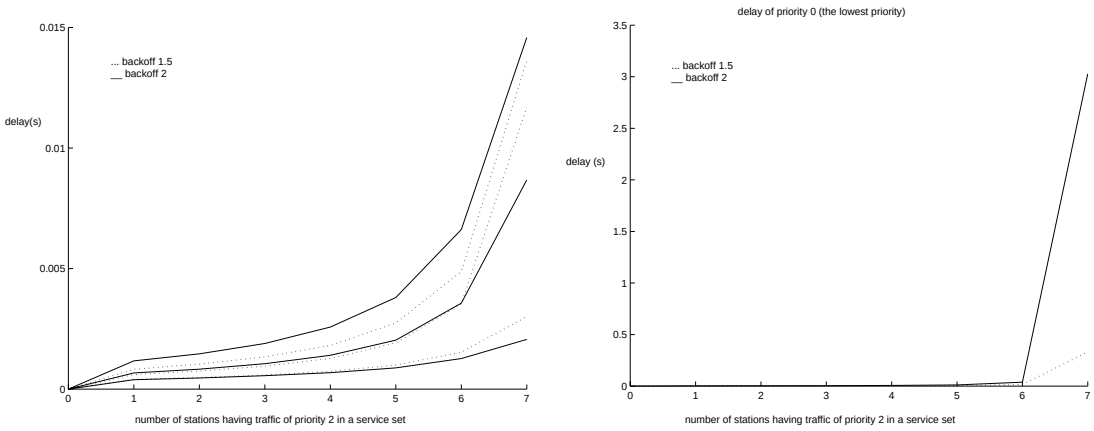


Figure 3.5: Effect of backoff window size on throughput



(a) (b)

Figure 3.6: Effect of backoff window size on delay

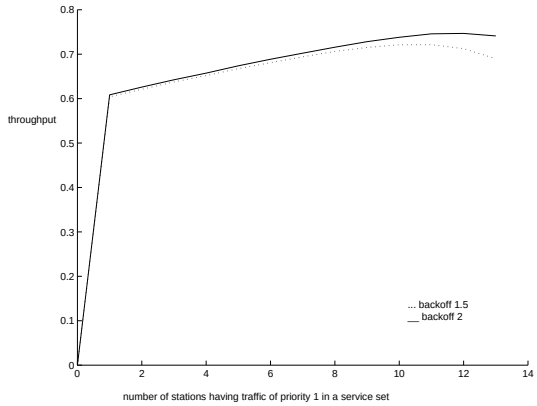
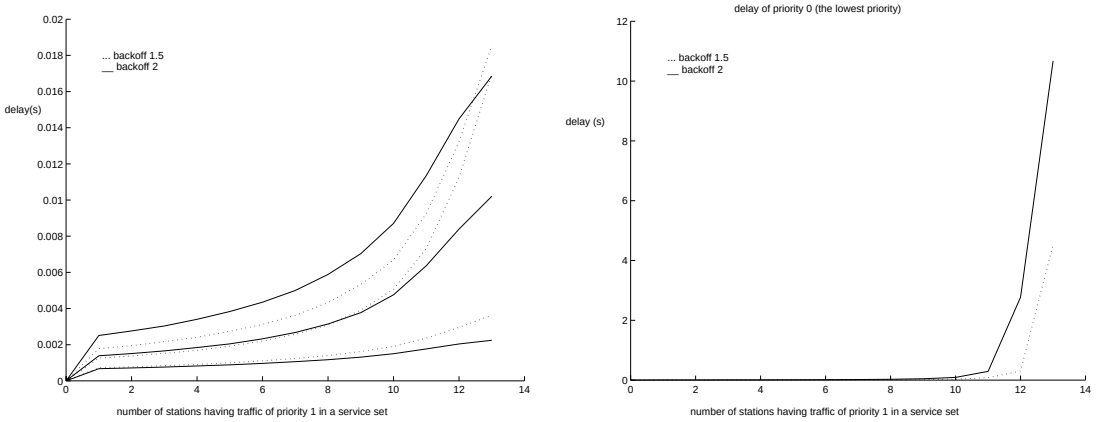


Figure 3.7: Effect of backoff window size on throughput



(a) (b)

Figure 3.8: Effect of backoff window size on delay

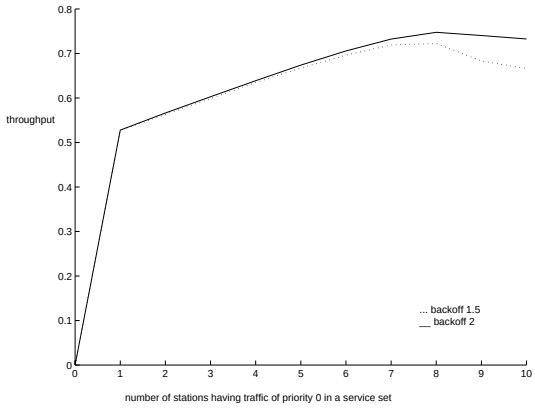
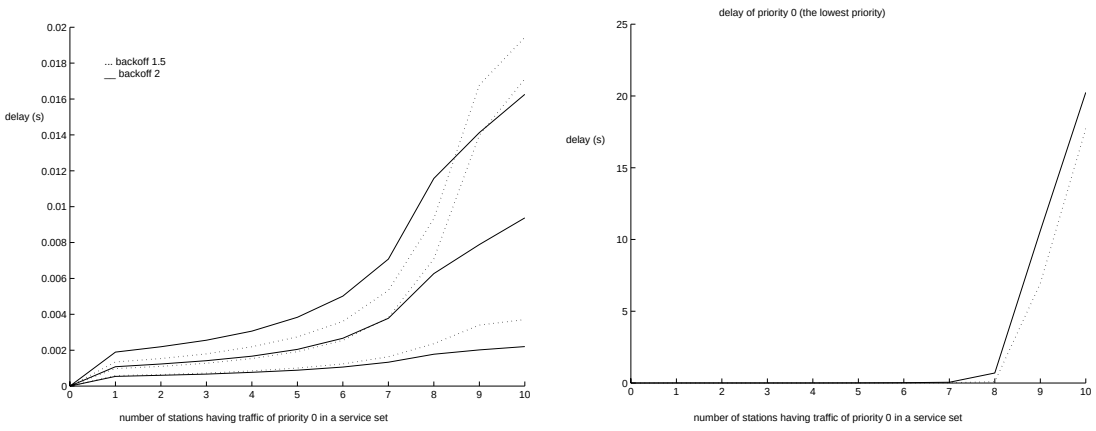


Figure 3.9: Effect of backoff window size on throughput



(a) (b)

Figure 3.10: Effect of backoff window size on delay

ing channel idle for a longer time together with choosing larger contention window limits for traffic of lower priority. Thus when the total effective arrival rate exceeds the channel service rate but the effective arrival rate of priority $L-1$ to l lies in the channel service rate, the delay of priority higher than l can still be bounded to a desirable value. However, the delay of priority lower than l will increase with time.

The effective arrival rate is defined as the average number of bits per second received at the MAC layer. For example, the total effective arrival rate when the number of stations having traffic of priority $(3,2,1,0) = (5,5,5,5)$ is $16m^{-1} \times (352/(352+650))$ [fraction of time in on state] $\times 18784 \times 5 + 1.5M \times 5 + 0.35K \times 5 + 50 \times 18784 \times 5 \approx 16M$. Though the channel capacity is 24Mbps, the service rate does not reach this value. Some bandwidth is wasted on idle time.

Figure 3.5 and Figure 3.6 shows the performance when the number stations having traffic of priority 2 is increasing. Since the mean arrival rate of priority 2 is very high, the system becomes saturation quickly. Note that the delay of the highest priority does not increases so quickly as other priorities.

From all figures shown in this sections, we can see that the performance of EDCF strongly depends on the system parameters, mainly the minimum contention size and the backff factor. We summarize how these parameters would effect the system performance in the following. From Figure 3.3 to Figure 3.10 we can see that

- When the load is light, smaller backoff factor and contention window size is suggested. Intuitively, the channel is usually idle when the load is light, thus a frame is suggested to be sent as soon as possible. Listening to the channel for a long time when the traffic load is light wates more bandwidth.

- When the load is heavy, smaller backoff factor and contention window size would incur higher delay due to higher collision probability.

- The backoff factor also depends on the starting contention window size. For

traffic of the lowest priority, its starting contention window size is large, backoff a longer time would let it almost lose every chance to seize the transmission opportunity, especially when the load is heavy. Thus when the backoff factor is large, the delay of the lowest priority is also very large. However, if traffic of lower priority backoff for a longer time when the load is heavy, traffic of higher priority would have more chance to seize the transmission opportunity, hence their delay would decrease with the compensation of longer delay from higher priority.

Thus we suggest all stations adjust their contention window size and backoff factor to network load. This can be achieved by measuring the number of collisions a frame suffers. When this number increases, the station is suggested to select a larger backoff factor, which provides a lower collision probability.

3.2 Performance Evaluation under Burst Error Channel

In wireless environment, bit error rate is large compared to wired environment. The 802.11 MAC protocol supports retransmissions and an optional forward error correction (FEC) to handle transmission errors. When FEC is not chosen, if the sender does not received the expected ACK frame, it will retransmitted the loss frame by sensing channel idle for SIFS time and applying the backoff procedure. When FEC is chosen, immediate ACK is not required. In this section, we take burst error channel into consideration. We first calculate the successful transmission probability under burst error channel and then show how the error channel would affect the delay if retransmission is applied by simulation. Finally we take FEC into consideration.

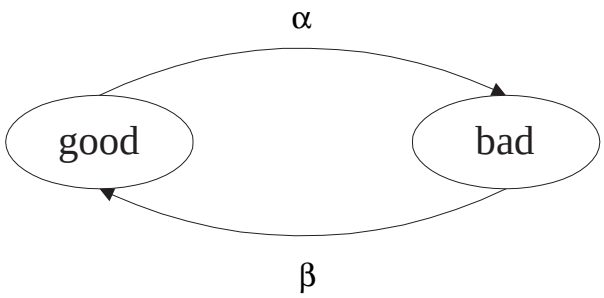


Figure 3.11: Burst error channel model.

3.2.1 Burst Error Channel Model

The burst error channel is modeled by a two-state continuous time Markov chain as shown in Figure 3.11. This model is first introduced by [32]. State good implies lower bit error rate (BER) channel. The mean sojourn time in state good and bad are exponentially distributed with parameter α^{-1} and β^{-1} respectively. The bit error rate in each state is BER_g and BER_b.

During a frame transmission interval, the shift of states can be categorized into three cases [11].

Case 1: When a frame transmission starts, the channel is in the good state, and it remains in the good state until the end of the frame transmission.

Case 2: When a frame transmission starts, the channel is in the bad state, and it remains in the bad state until the end of the frame transmission.

Case 3: When a transmission starts, the channel is in either state, and it undergoes one or more transitions before the frame transmission completes.

3.2.2 Numerical Discussions

Using the numerical results derived in Sec. 2.2, the successful transmission probability under error-free channel is

$$P_{succ} = \sum_{i=1}^{N} \pi(i) \sum_{l=0}^{L-1} \sum_{v_l=1}^{i} \cdots \sum_{v_0} P_i(l; \{v_m\}_{m=0}^{L-1}) \sum_{sp=0}^{l} P_{succ}^{(i,l;\{v_m\}_{m=0}^{L-1})}(sp) \tag{3.6}$$

Considering the three cases introduced in the previous section, the probability of successful transmission under burst error channel could be derived. All assumptions are the same as those depicted in Sec. 2.1. Note that erroneous frames are regarded as collided frames.

Before getting the successful transmission probability, the error frame probability need to be derived first [11]. Let P_i be the probability that case i occurs. Let T_g and T_b be the time that the channel lies in the good and the bad state. It is easy to see that

$$\begin{aligned} P_1 &= \pi_g P(T_g > T_{frame}) = \frac{\beta}{\alpha+\beta} e^{-\alpha T_{frame}} \\ P_2 &= \pi_b P(T_b > T_{frame}) = \frac{\alpha}{\alpha+\beta} e^{-\beta T_{frame}} \\ P_3 &= 1 - P_1 - P_2 \end{aligned} \tag{3.7}$$

where π_g and π_b are the probabilities that a frame starts transmission when the channel is in the good or the bad state. They are equal to the steady state probabilities of the good and the bad state respectively.

The probability of an erroneous frame, ε_i (case i occurs), is approximated by

$$\begin{aligned} \varepsilon_1 &= 1 - (1 - BER_g)^{MPDU} \\ \varepsilon_2 &= 1 - (1 - BER_b)^{MPDU} \\ \varepsilon_1 &\le \varepsilon_3 \le \varepsilon_2 \end{aligned} \tag{3.8}$$

Combining (3.7) and (3.8) yields the error frame probability. Note that a worse-case error probability is assumed here when case 3 occurs, that is, all bits are subject

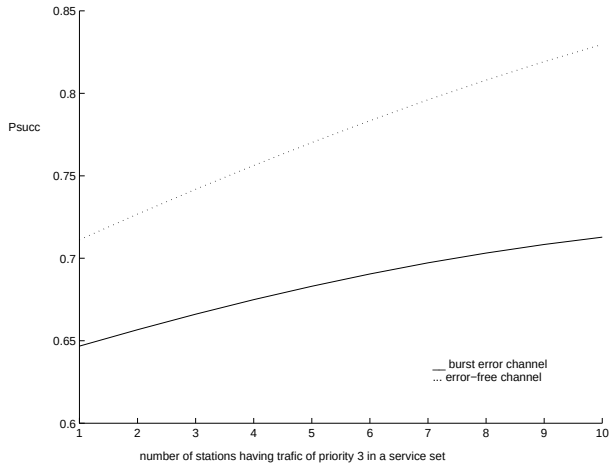


Figure 3.12: Successful transmission probability under burst error channel and error-free channel.

to BER_b

$$P_{err} \leq P_1\varepsilon_1 + P_2\varepsilon_2 + P_3\varepsilon_2 \tag{3.9}$$

Thus, the successful transmission probability in (2.12) now becomes

$$P_{succ}^{'(i,l;\{v_m\}_{m=0}^{L-1})}(sp) = \begin{cases} 0, & \text{if } sp > l \\ \sum_{y_l=1}^{cw_l} p_s(y_l)(1-P_{err}), & \text{if } sp \leq l \end{cases}$$

and the collision probability in (2.17) becomes

$$p_c'(y_l) = p_c(y_l) + p_s(y_l)P_{err} \tag{3.10}$$

where the first term accounts for collisions, and the second term accounts for channel error.

The comparison of successful transmission probability with different number of stations under perfect and burst error channel is shown in Figure 3.12.

The parameters for the burst channel model are listed in Table 3.8. These parameters are suggested by [11]. All traffic sources are the same as those depicted in Sec. 3.1.1.

Notation	Description	Value
BER_g	bit error rate in state good	10^{-4}
BER_b	bit error rate in state bad	10^{-2}
α^{-1}	mean sojourn time in state good	0.1 sec
β^{-1}	mean sojourn time in state bad	0.05 sec

Table 3.8: Parameters for burst error channel model 1.

3.2.3 Simulation Results

We are interested in how burst error channel would affect the performance of EDCF. With the retransmission backoff procedures, some simulation results are shown in this section.

EDCF provides an optional MAC-level forward error correction (FEC) that may be used to reduce both the frequency of retransmissions and the MSDU loss rate for transfers via the wireless medium. The MAC-level FEC uses a (224,208) Reed-Solomon code for decoding the MPDU. FEC coding is performed on successive 208-byte blocks of the MPDU. The FEC coding adds 16 parity bytes per block. The MAC header is encoded using a (48,32) Reed-Solomon code. The code rate is approximately 0.938.

3.2.4 Conclusions

Under burst error channel, network capacity will decrease. Applying (3.7) to (3.9), the effective network capacity could be derived. Considering error model 1 (Table 3.8), we have $P_{err} \approx 0.21$, which yields an effective network capacity approximately equal to 18.96Mbps. The performance degradation under burst error channel is shown in Figure 3.13 and 3.14. Under burst error channel, the delay of lower priority seriously degrades.

Observing (3.10) we know that an unsuccessful transmission results from a col-

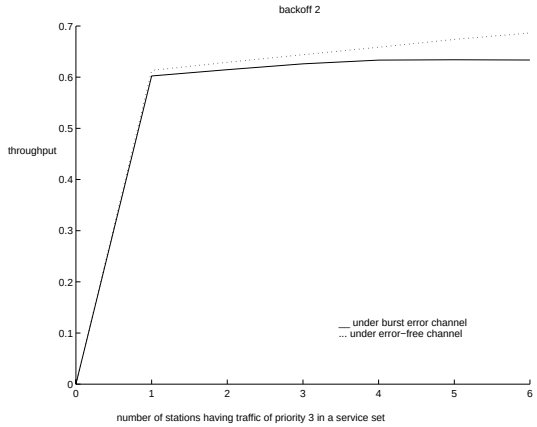
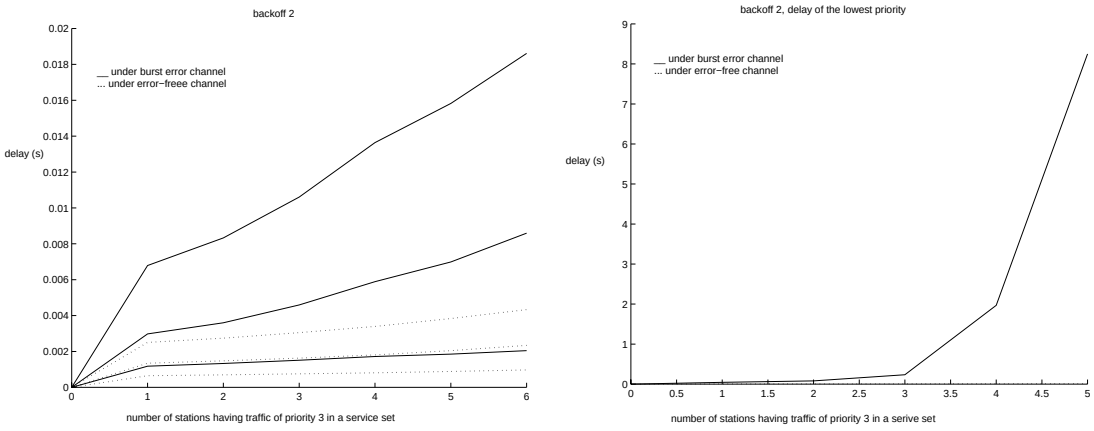
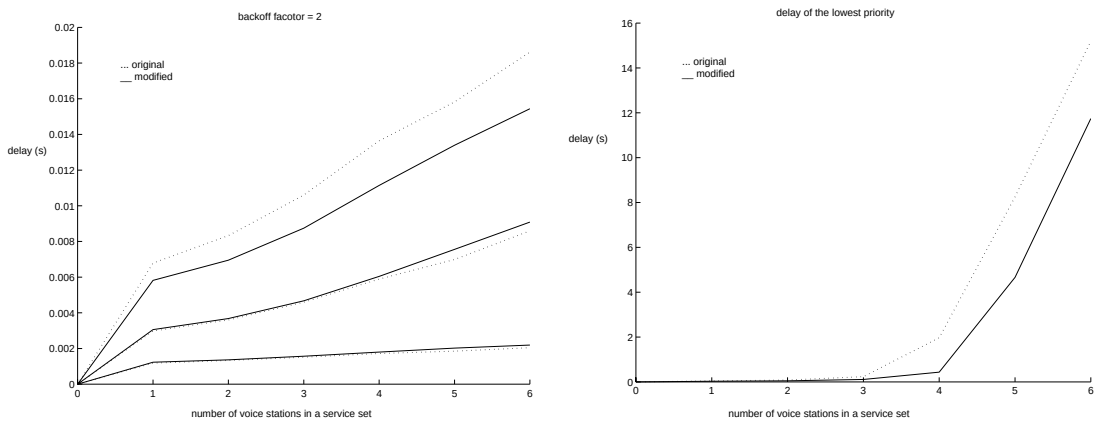


Figure 3.13: Throughput degradation under burst-error channel



(a) (b)

Figure 3.14: Delay degradation under burst error channel. (The dash line in (b) cannot display clearly. Please refer to Figure 3.4 (b).)



(a) (b)

Figure 3.15: Delay comparison of the original EDCF under burst error channel. (Original: Regard an error as a collision. Modified: Distinguish between an error and a collision.)

lided transfer or an error channel. Distinguishing between these two cases first comes into our mind. That is, do not apply backoff procedure when the unsuccessful transmission is due to an error channel, since the backoff procedure mainly operates for avoiding collisions. However, the best policy is not sending any frame when the channel is in bad state because any transmission in this scenario is useless and needs another time to retransmit the erroneous frame. Thus, for frames whose original backoff interval are smaller than the duration of the channel in bad state, regarding an error as a collision and then backoff a longer time exactly fits this need. But for frames whose backoff interval exceeds the duration of the channel in bad state, distinguishing between a collision and an error results in smaller delay (Figure 3.15 (b) and the top line and dash in Figure 3.15 (a)).

A (N,K) Reed-Solomon code is guaranteed to correct up to

$$t = \lfloor 0.5 \times (N-K) \rfloor$$

symbol errors [33]. Thus the R-S code defined in EDCF is capable of correcting

up to 8-byte (64-bit) errors per block (224 bytes). The worst case (case 3 in Sec. 3.2.1) in the burst error channel model shown in Table 3.8 results in 18-bit errors in a block ($224 \times 8 \times BER_b$). When the optional MAC-level FEC is chosen, no erroneous frame need to be retransmitted.

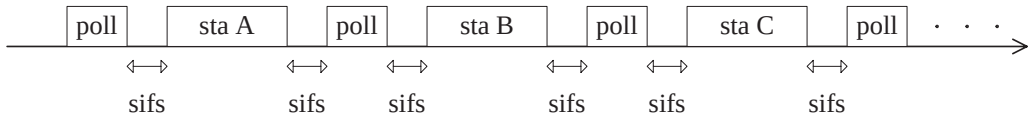
Chapter 4

Proposed Enhancement for HCF

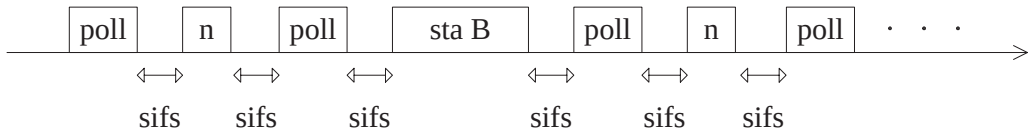
The best does not come alone.
It comes with the company of the all.
– Tagore

4.1 Basic Problem

The PCF mode offers a packet-switched connection-oriented service, which is inherently suitable to provide quality-of-service for delay-guaranteed traffic. Packet-switched services take the advantage of silences in a given voice call by multiplexing voice frames with other frames. Thus is more-bandwidth-efficient than circuit-switched services. In wireless networks, where bandwidth is more constrained, the use of packet-switched techniques for carrying voice traffic is an inevitable trend. The connection-oriented aspect of PCF mode would provide delay guarantees necessary for voice and video traffic. However, as other polling-based operations, the system would suffer inefficiency when the addressed recipient does not have frames to send. When this occurs, the bandwidth is wasted on both polling and the reaction of the polled stations, which in turn makes the delay longer. This problem is illustrated in Figure 4.1. When a polled station does not have frames to send, it informs



(a)



n: null (no frames to be sent by sta A)

(b)

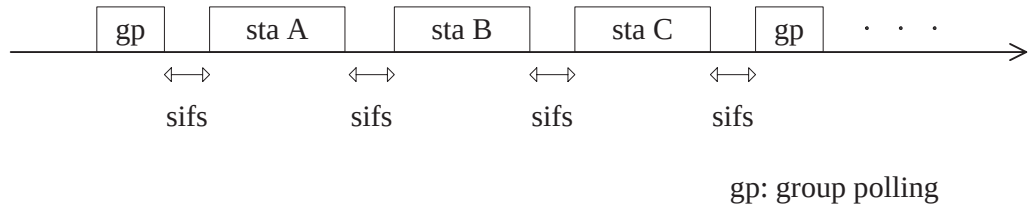
Figure 4.1: (a) Example of HCF. (b) Inefficiency introduced by HCF when a station does not have frames to be sent.

the access point (AP) by sending back a short frame called null.

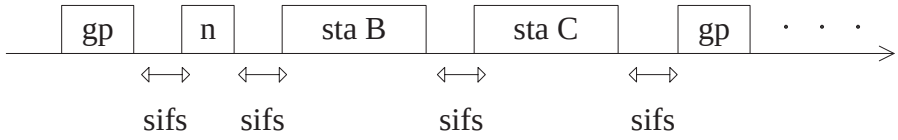
4.2 Proposed Grouping Polling for HCF

In order to resolve the inefficiency introduced in the previous section, we suggest the AP applies group polling. Incorporating with the concept of EDCF, a ”group” consists of stations with traffic of each priority. That is, a group consists of L stations when there are L priorities. The AP holds L ”lists” for each priority. In a group, a station will appear at most once. The conception of group and list is illustrated in Figure 4.3.

We summarize the proposed group polling for HCF operations to two phases, the polling phase and the transmission phase. In the polling phase, the AP sends the group polling frame to a group of stations on the polling list orthogonally in the frequency band. In the transmission phase, the addressed recipients send replied message sequentially.



(a)



(b)

Figure 4.2: (a) Example of the proposed grouping for HCF. (b) When a station does not have frames to be sent.

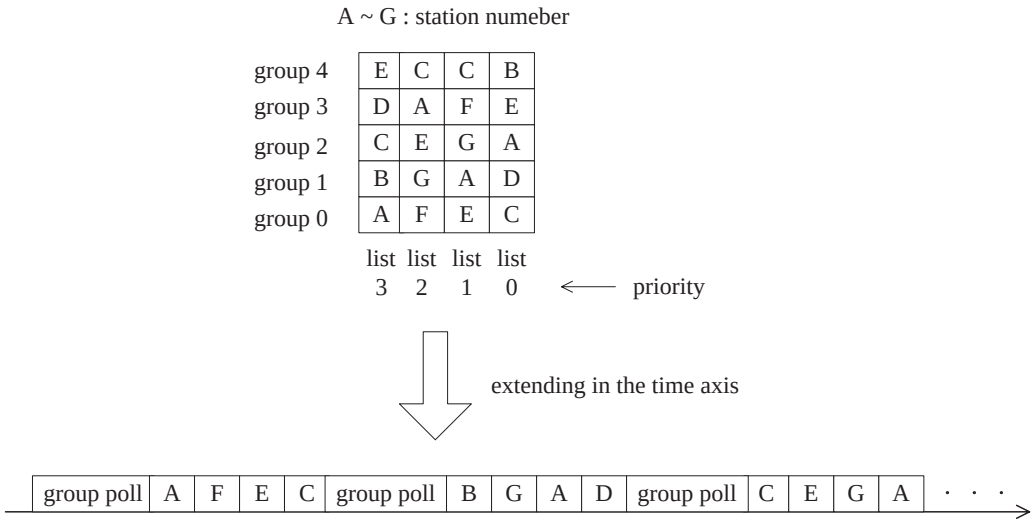


Figure 4.3: Concept of group and list.

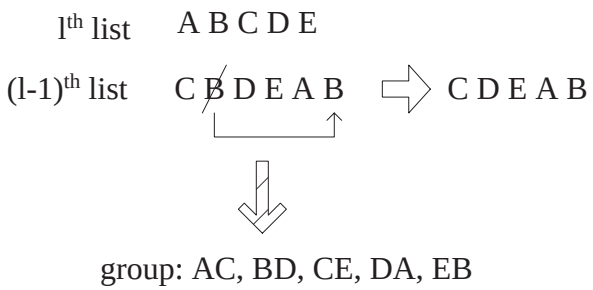


Figure 4.4: Basic operation in the polling phase.

Polling Phase

We first describe the operations in the polling phase. When a station initiates a voice application or other applications generating frames to only one priority, it first signals the AP using EDCF and is added to the polling list. The AP separates all voice stations to different groups (because voice frames belong to only one priority) in the ascending order of their request time.

When a station request for video transfers or other applications generating frames to at least two priorities, the operation is a little different from that of voice application. The video station would signal the AP with its starting priority. Since a video station is able to send frames of different priorities during its life, the AP first adds it to all lists which belong to video traffic.

The constraint that a station appears at most once in a group is reasonable and intuitive because of fairness and efficiency. To meet this constraint, before initiating a group polling, the AP needs to check if any list violates this law. If yes, the station that appears at lists of lower priority is put to the last position of the list. This is illustrated in Figure 4.4.

The HCF operation allows multiple frame exchanges from one addressed recipi-

ent during its transmission opportunity period. The non-QoS frame can be and only be sent as the sole or the final transmission during a transmission opportunity period. This also holds in our proposed scheme. On the other hand, the QoS frame not belonging to the polled priority should be the sole or the final transmission during a transmission opportunity period.

Transmission Phase

All stations in current group would receive the group polling frame consisting of the information that which priority it is addressed now (i.e. at which position on the polling list). In the transmission phase, the station of the highest priority would send its frame after SIFS time. Other stations would listen to the channel. After this frame transfer and another SIFS time, the station of the second highest priority would send its frame. This is illustrated in Figure 4.5.

Note that in this group polling scheme, collision is possible when the addressed recipients in a group cannot hear each other (hidden terminal problem [34]).

4.3 Simulation Results

Based on the parameters listed in Sec. 3.1.1 and Sec. 3.1.2, we compare the original hybrid coordination function with our proposed enhancement scheme by measuring the delay of the frames transmitted using HCF polled channel access. Note that the voice frame size is set to 120 bytes, and the video frame size is set to 180 bytes.

From Figure 4.5, we can see that the delay can be reduced a lot by our proposed grouping polling scheme. This is due to the fact that when group polling is applied, time wasted on polling inactive stations can be greatly reduced.

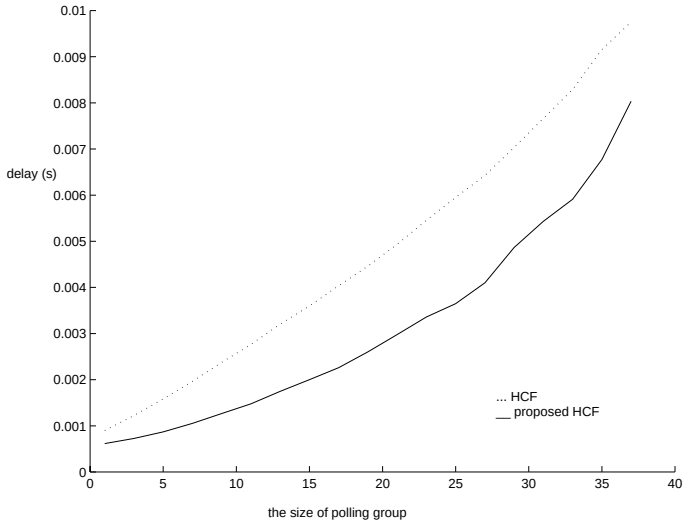


Figure 4.5: Delay comparison of HCF and the proposed group polling for HCF

Chapter 5

Implementation of EDCF

Thoughts pass in my mind
like flocks of ducks in the sky.
I hear the voice of their wings.
– Tagore

In this chapter, we would like to consider the implementation issues of the enhanced distributed coordination function (EDCF). There are two methods for implementing EDCF, one is hardware implementation, and the other is software implementation. We would evaluate their performance by simulation.

Before describing the implementation models, please note that there are two parts in a station, the host and the network interface card (NIC). The interface between them is PCI interface.

5.1 Hardware Implementation Model

The hardware implementation of EDCF exactly follows the standard. The NIC holds L separated queues for each priority. The host generates different prioritized frames and then put them directly into the NIC according to their priority. The latency of PCI access from host to NIC is about several microseconds and is ignored.

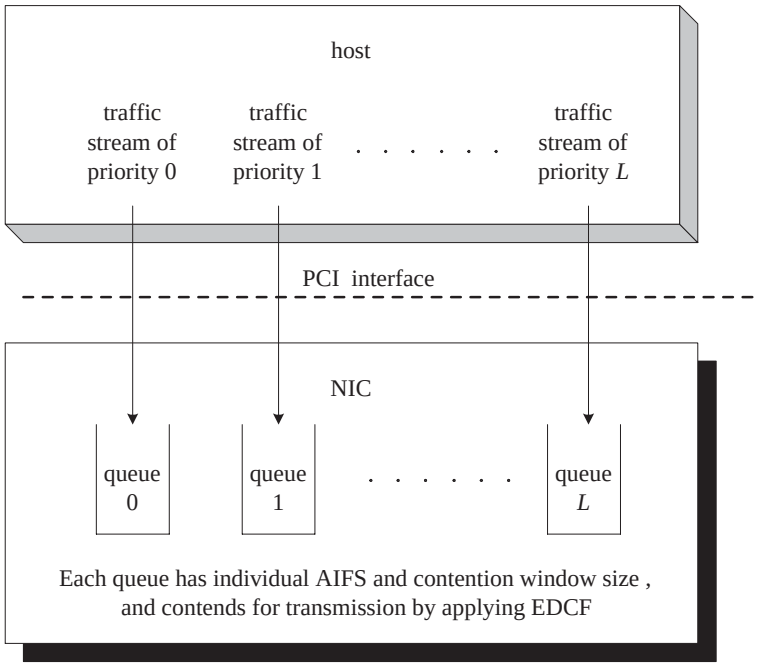


Figure 5.1: Hardware implementation model.

Each queue in the NIC has its parameters ($AIFS$, CW_{min}, CW_{max}, etc.) regarding its priority. It applies EDCF with these parameters to gain the transmission opportunity. When collisions occur (either inside the NIC or in the channel), the backoff procedure is applied if the retry count is not reached or the frame is dropped if the retry count is reached. The queue size is assumed unlimited. This model is illustrated in Figure 5.1.

5.2 Software Implementation Model

The block diagram of the software implementation is shown in Figure 5.2. In the host, the software virtually maintains L queues for each priority. In the NIC, there is only one queue. The EDCF is implemented in the software. That is, the host generates different prioritized frames and then put them in the virtual queue according

to their priority. The NIC operates only EDCF with parameterized AIFS and the contention window size directly calculated from the software. In other words, the NIC simply reads the interval to sense channel idle and the backoff time which were marked on the frame by the software, and then applies DCF procedures with these parameters.

Every frame when generated is marked a transmitted time by the software. This time interval consists of AIFS time and backoff time. The software determines which frame gets the transmission opportunity by applying the virtual EDCF in every queue. Then it puts these frames into the NIC queue by the ascending order of the transmitted time. The NIC does not have the ability to order these frames. Thus every time a frame is generated, the software will reorder the frames in the NIC queue according to the transmitted time marked on each frame. When collision occurs, the collided frame is sent back to the software to reschedule a backoff time using EDCF if the retry count is not reached or to drop the frame if the retry count is reached. The latency for frames returned to the host through PCI interface is not small thus cannot be ignored.

The backoff time of a frame generated by the software should be remembered till the frame is sent. At this time, the transmitted time of all frames in the software queues should be reduced by this interval. This is due to the fact that, in the original hardware implementation, the backoff timer of each queue in the NIC should be decreased when the channel is idle. There is some difference between these two models. The backoff timer of the software implementation model is decreased only when the first frame in the NIC is sent.

There is one criterion in the software implementation model. The new generated frame can be at most the second frame in the NIC queue no matter how small the transmitted time it has. That is, the frame contending for transmission with other stations will keep on contending till it is sent (a successful transmission or a collision). This is because the head frame in the NIC queue has been forwarded to

Description	Value
MPDU	1000 bytes
PCI access from NIC to host	10m sec
PCI access from host to NIC	0
Retry Count	3

Table 5.1: System parameters for implementation models.

the physical layer. The cost of reorder this frame with other frames is high.

5.3 Simulation Results

In this section, we would like to prove that our proposed software implementation model still has the ability to differentiate each priority when traffic load is not heavy. And we would like to see its performance by measuring the delay and dropping probability of each priority.

The system parameters are listed in Table 3.1 with the difference that the frame size (MPDU) is set to 1000 bytes in this section. The retry-count is set to 3, and the PCI access delay from NIC to the host is set to 10m sec. These parameters are listed in Table 5.1. The contention window sizes apply here are shown in Table 3.6. Note that in this section, a station is able to generate frames of each priority and the traffic models are depicted in Table 5.2 and Table 5.3. Only error-free channel is considered here.

As well known in the literature, voice traffic can be appropriated modeled by an on-off Markov chain as shown in Figure 2.4. Here we consider a more complete model shown in Figure 5.3 taking a talkspurt, mini-silence, and the duration of a voice connection into consideration. The parameters needed to specify this process are listed in Table 5.2. The arrival rate in state mini-silence and silence are both zero.

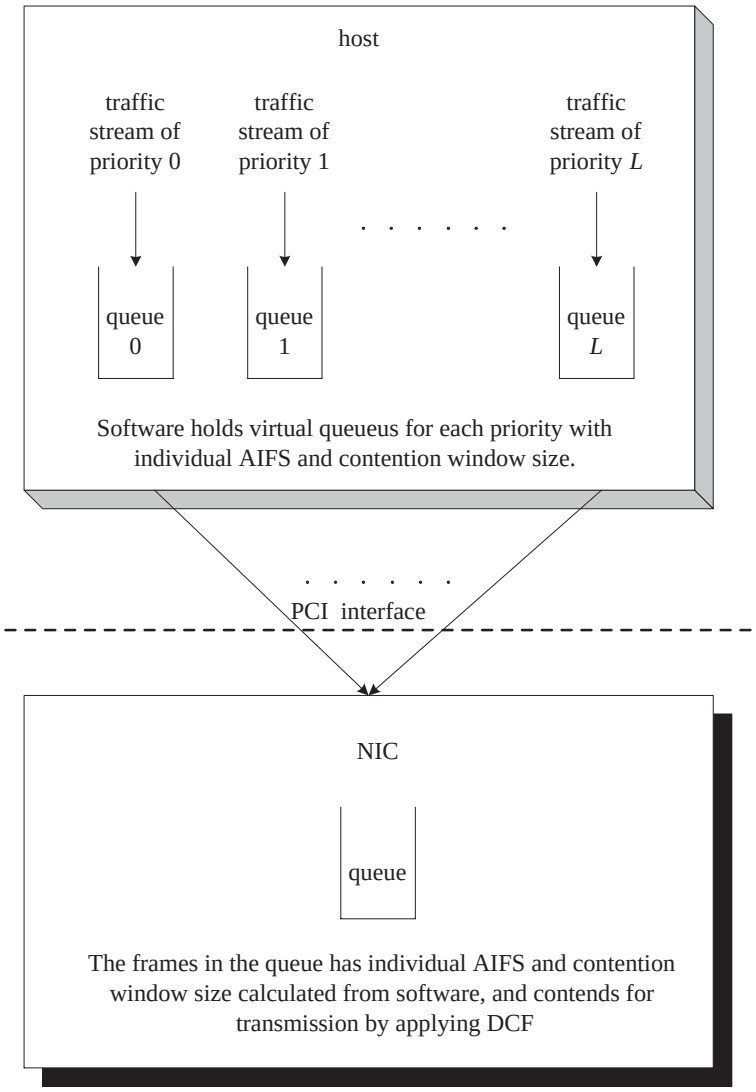


Figure 5.2: Software implementation model.

Notation	Description	Value (sec)
λ_t^{-1}	interarrival time in state talkspurt	0.016
r_t^{-1}	mean sojourn time in state talkspurt	0.352
r_s^{-1}	mean sojourn time in state mini-silence	0.65
r_{off}^{-1}	mean sojurn time in state silence	120
r_{on}^{-1}	average duration of a voice connection	120

Table 5.2: Parameters for voice source with mini-silence.

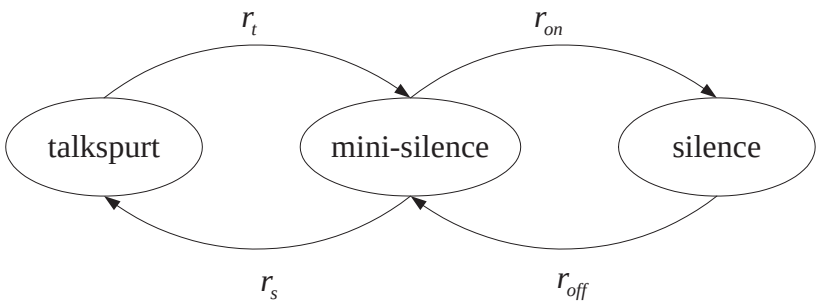


Figure 5.3: Voice model with mini-silence.

Priority	Description	Value
2	mean arrival rate	1.5Mbps
	average duration of a video connection	300 sec
	mean sojourn time in state off of a video connection	180 sec
1	mean arrival rate	350Kbps
	average duration of a video connection	300 sec
	mean sojourn time in state off of a video connection	120 sec

Table 5.3: Parameters for video sources.

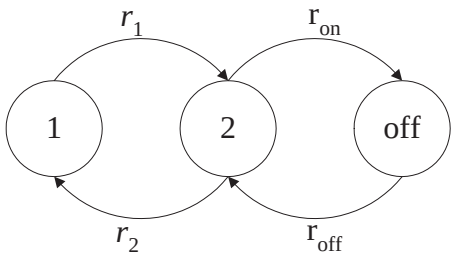


Figure 5.4: Three-state MMPP.

For video traffic, we also take the duration of a video call into consideration. Thus a video source is modeled by a 3-state MMPP here as illustrated in Figure 5.4. The arrival rate in state off is zero. Other parameters are listed in Table 5.3 and Table 3.3.

Each simulation in this section runs for 2000 sec to take the time average.

The simulation results are listed from Table 5.4 to Table 5.13. Note that the value shown on top of each table is the normalized arrival rate. The unit of delay is millisecond if no further specified, and P_{drop}, the dropping probability, is calculated in percentage.

From Table 5.4 to Table 5.9, the delay bound is set to 10m sec. Thus in the software implementation model, all frames collided in the channel will be dropped

8 stations, 0.5584				
	Hardware		Software	
priority	delay	P_{drop}	delay	P_{drop}
3	1.476	0	1.153	0.44
2	1.591	0	1.223	0.63
1	2.236	0	2.331	0.59
0	9.712	0	3.662	0.04

Table 5.4: data rate = 50 pkt/sec

	4 stations, 0.3459				5 stations, 0.4324			
pri-	Hardware		Software		Hardware		Software	
ority	delay	P_{drop}	delay	P_{drop}	delay	P_{drop}	delay	P_{drop}
3	0.243	0	0.427	0.17	0.288	0	0.656	0.06
2	0.464	0	0.581	0.04	0.577	0	0.694	0.07
1	0.804	0	0.966	0.26	1.005	0	1.104	0.09
0	1.621	0	6.822	1.40	2.220	0	1.907	0.0012

Table 5.5: data rate = 100 pkt/sec

due to the PCI access delay from NIC to the host (10m sec). From Table 5.10 to Table 5.13, the delay bound is set to 20m sec.

5.4 Conclusions

In this section, we summarize the simulation results of the two implementation models described in Sec. 5.1 and Sec. 5.2 by comparing their performance with respect to the normalized arrival rate from all stations in a service set. The normalized arrival rate is defined as the average number of bits per second received at the MAC sublayer from the upper layer divided by the channel capacity (transmis-

	6 stations, 0.5188				7 stations, 0.6053			
pri-	Hardware		Software		Hardware		Software	
ority	delay	P_{drop}	delay	P_{drop}	delay	P_{drop}	delay	P_{drop}
3	0.342	0	1.240	1.02	0.445	0	2.482	3.77
2	0.732	0	1.020	0.59	1.070	0	2.243	3.46
1	1.289	0	2.174	0.82	1.874	0	3.021	3.30
0	2.939	0	24.31	1.54	5.928	0	5.044	0

Table 5.6: data rate = 100 pkt/sec

	3 stations, 0.3594				4 stations, 0.4792			
pri-	Hardware		Software		Hardware		Software	
ority	delay	P_{drop}	delay	P_{drop}	delay	P_{drop}	delay	P_{drop}
3	0.259	0	0.610	0.02	0.304	0	1.489	0.07
2	0.493	0	0.716	0.04	0.611	0	1.632	0.08
1	0.870	0	0.989	0.04	1.057	0	1.895	0.06
0	2.055	0	1.746	0.01	2.895	0	3.152	0

Table 5.7: data rate = 200 pkt/sec

	5 stations, 0.5590				6 stations, 0.7188			
pri-	Hardware		Software		Hardware		Software	
ority	delay	P_{drop}	delay	P_{drop}	delay	P_{drop}	delay	P_{drop}
3	0.485	0	3.566	3.54	0.595	0	11.87	16.66
2	1.144	0	2.491	4.06	1.553	0	4.582	19.76
1	2.103	0	2.847	3.87	2.961	0	6.400	17.61
0	323.5	0	4.710	0.16	73580	0	8.924	0.13

Table 5.8: data rate = 200 pkt/sec

3 stations, 1.1594				
	Hardware		Software	
priority	delay (sec)	P_{drop}	delay (sec)	P_{drop}
3	0.360×10^{-3}	0	nan	100
2	0.745×10^{-3}	0	0.0030	99.99
1	1.348×10^{-3}	0	0.0047	99.99
0	580.4	0	127.34	54.66

Table 5.9: data rate = 1000 pkt/sec

8 stations, 0.5584		
priority	delay	P_{drop}
3	1.476	0.22
2	1.590	0.39
1	2.236	0.25
0	9.712	0.86

Table 5.10: Software, delay bound = 20 m sec, data rate = 50 pkt/sec

	4 stations		5 stations		6 stations		7 stations	
	0.3459		0.4324		0.5188		0.6053	
pri	delay	P_{drop}	delay	P_{drop}	delay	P_{drop}	delay	P_{drop}
3	0.530	0	0.635	0.00101	1.072	0.22	2.221	0.94
2	0.626	0.00204	0.789	0.00657	1.157	0.45	2.139	0.78
1	1.145	0.00115	1.147	0.00320	1.710	0.41	3.221	0.63
0	1.753	0	2.034	0.00220	9.498	0.98	11.11	0.82

Table 5.11: Software, delay bound = 20m sec, data rate = 100 pkt/sec

	3 stations		4 stations		5 stations		6 stations	
	0.3594		0.4792		0.5990		0.7188	
pri	delay	P_{drop}	delay	P_{drop}	delay	P_{drop}	delay	P_{drop}
3	0.710	0	1.553	0.056	3.198	2.05	7.891	9.38
2	0.840	0.00058	1.598	0.062	2.804	2.38	6.133	10.63
1	1.168	0.0024	1.957	0.051	4.009	1.79	6.480	10.01
0	2.032	0.035	3.244	0.00038	51.51	1.78	30.27	1.02

Table 5.12: Software, delay bound = 20m sec, data rate = 200 pkt/sec

3 stations, 1.1594		
priority	delay (sec)	P_{drop}
3	nan	100
2	9.262×10^{-3}	99.99
1	15.59×10^{-3}	99.99
0	475.768	0

Table 5.13: Software, delay bound = 20 m sec, data rate = 50 pkt/sec

sion rate). The mean arrival rate (from all priorities) of a station when data rate is d bps is $16m^{-1} \times 8000 \times (352/(352+650)) \times 0.5 + 1.5M \times (300/480) + 0.35K \times (300/420) + d = (1.275 + d)$ bps.

When traffic load is light, the performance of the software implementation model is good. It has the ability to differentiate each priority even when there are many stations in a service set (but the total traffic should be light). Our proposed software implementation model has good performance between stations in the network. This is because virtual EDCF is applied in the software, and hardware reads the AIFS time and contention window size marked on frames by the software for backoff. Thus the system parameters are exactly set according to the original EDCF.

The major drawback of our proposed software implementation model lies in the operation between all queues in a station. Thus when traffic load is heavy because of high arrival rate from lower priority queues or because of too many stations in a service set, performance of higher priority frames in the software implementation model seriously degrades. This is due to the constraint that any new generated frame can be at most the second frame in the NIC queue. Thus when the frame of lower priority gets the transmission opportunity in a station (this frame is sent to the NIC queue), it contends for the wireless medium with other stations which may have frames of higher priority. The lower priority frame cannot seize the channel till a long time. Thus frames of higher priority in this station will be blocked in the NIC. On the other hand, the delay of each priority cannot be differentiated since many high-priority frames are blocked in a station. Similar phenomenon happens when there are many stations in a service set. Note that in the hardware implementation model, this bad scenario does not happen since every queue in a station contends for transmission at one time. The lower-priority frame will not block higher-priority frames. In the hardware implementation model, when the load is heavy because of high arrival rate from traffic of lower priority, only the delay of lower priority frames will increase with time. Please refer to Sec. 3.1.3 for more detailed explanations.

The PCI access delay from NIC to the host is another unavoidable problem in the software implementation model. This explains why the delay and the dropping probability of software implementation model are larger than the hardware implementation model.

Let us give some brief summaries from our simulated data.

- When total effective traffic load is less than approximately half of the network capacity, the software implementation model works well. The dropping probability is less than 1%, and the delay of each priority is small.

- When total effective traffic load exceeds approximately 50% of the network capacity, the software implementation model cannot work. But higher priority frames in the hardware implementation model still can work as long as their effective arrival rate is less than the service rate.

The hardware implementation model, though has the better performance, is not cost-effective especially when the number of priorities (queues) is increasing. The software implementation model is more flexible and cost-effective. Its performance is not as good as the hardware implementation model but is still in a tolerable range.

Chapter 6

Conclusions

The service of the fruit is precious,
the service of the flower is sweet,
but let my service be the service of the leaves
in its shade of humble devotion.
– Tagore

In Chapter 2, we have presented an analytical model to evaluate the throughput and delay performance of IEEE 802.11 enhanced DCF under integrated traffic sources with the assumption of constant backoff window size and independent traffic generation. Our model also assumes fixed number of stations having traffic of each priority in a service set and the channel is error-free.

In Chapter 3, comparison with the simulation results shows that our proposed analytical model is extremely accurate in predicting the system throughput. We also give some numerical discussions on the probability of successful transmission under burst error channel. Then we evaluate the system performance under error-free and burst error channel with different contention window size and backoff factor. These two parameters strongly effect the system performance. This is summarized in Sec. 3.1.3 and 3.2.4.

In Chapter 5, we have introduced a model to implement EDCF by software instead of hardware. Software implementation usually has the benefit of lower cost and higher flexibility. However, two constraints on this model make the performance of software implementation worse than the hardware implementation. One is that the first frame in the network interface card queue cannot be reordered for transmission, and the other is the unnegligible PCI access delay from NIC to host. The performance degradation is still in a tolerable range when traffic load is less than half of network capacity. Detailed explanations of this phenomenon are presented in Sec. 5.4.

In Chapter 4, we have proposed a group polling algorithm which resolves the inefficiency when the addressed recipient does not have frames to send. A group consists of stations from each priority. Simulation results show its better performance over the original HCF.

Appendix A

Derivations of Equations

A.1 (2.23)

From the *key renewal theorem,* we have

$$y(0,j) = \sum_{h=1}^{N} \frac{1}{s(0,0,h)} \int_0^\infty P(0,0,t)_{hj} dt$$

where $s(0,0,h)$ denotes the mean recurrence time of the state $(0,0,h)$ in the Markov chain $\{N_n^0, N_n^1, J_n\}$ defined in Sec. 2.3 and $\int_0^\infty P(0,0,t)dt$ is the mean idle period starting from state j,

$$\int_0^\infty P(0,0,t)dt = \int_0^\infty P^*(0,0,t)dt = \int_0^\infty e^{Ct} dt = -C^{-1}$$

by setting $z = 0$ in (2.20). And

$$m(0,0,k) = E x_{0,0,k}^{-1}$$

where E is the interdeparture time of frames [26],

$$\begin{aligned} E &= x_{0,0}[(-C)^{-1} + g]e + (1 - x_{0,0}e)g \\ &= x_{0,0}(-C)^{-1}e + g \end{aligned} \tag{A.1}$$

Note that the first term accounts for the interdeparture time when both queues are empty and the second term accounts for the interdeparture time when there are frames queued in the buffer.

Then it follows (2.23).

Bibliography

[1] *Draft Supplement to IEEE Standard, Part 11: Wireless Medium Access Control and Physical Layer Specifications: Medium Access Control Enhancements for Quality of Service*, P802.11/D2.0a. Nov. 2001.

[2] *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, P802.11. Nov. 1997.

[3] J.L. Hammond and P.J.P. O'Reilly, *Performance Analysis of Local Computer Networks*, Addison-Wesley, 1988.

[4] S. Tasaka, *Performance Analysis of Multiple Access Protocols*, MIT Press, 1986.

[5] R. Rom and M. Sidi, *Multiple Access Protocols – Performance and Analysis*, Springer-Verlag, 1990.

[6] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, 2nd edition, 1992.

[7] J.G. Kim B.P. Crow, I. Widjaja and P.T. Sakai, "IEEE 802.11 wireless local area networks," *IEEE Commun. Mag.*, pp. 116–126, Sep. 1997.

[8] Giuseppe Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Select. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[9] M. Conti F. Cali and E. Gregori, "IEEE 802.11 protocol: design and performance evaluation of an adaptive backoff mechanism," *IEEE J. Select. Areas Commun.*, vol. 18, no. 9, pp. 1774–1786, Sep. 2000.

[10] A. Kahol S.K.S. Gupta S. Khurana and P.K. Srimani, "Performance evaluation of distributed coordination function for IEEE 802.11 wireless LAN protocol in presence of mobile and hidden terminals," *MASCOT*, pp. 40–47, Oct. 1999.

[11] N. Cocker M. Veeraraghavan and T. Moors, "Support of voice services in IEEE 802.11 wireless LANs," *INFOCOM*, pp. 488–497, 2001.

[12] K.C. Chen and C.H. Lee, "RAP- a novel medium access control protocol for wireless data networks," *IEEE GLOBECOM*, pp. 1713–1717, 1993.

[13] D. Qiao and S. Choi, "Goodput enhancement of IEEE 802.11a wireless LAN via link adaptation," *Proc. IEEE ICC*, pp. 1995–2000, Jun. 2001.

[14] K. Sriram and W. Whitt, "Characterizing superposition arrival processes in packet multiplexers for voice and data," *IEEE J. Select. Areas Commun.*, vol. 4, no. 6, pp. 833–846, Sep. 1986.

[15] J.N. Daigle and J.D. Langford, "Models for analysis of packet voice communication systems," *IEEE J. Select. Areas Commun.*, vol. 4, no. 6, pp. 847–855, Sep. 1986.

[16] K. Meier-Hellstern D.M. Lucantoni and M.F. Neuts, "A single server queue with server vacations and a class of non-renewal arrival process," *Adv. Appl. Prob.*, vol. 22, pp. 676–705, 1990.

[17] Graham, *Kronecker Products and Matrix Calculus with Applications*, Ellis Horwood, Chichester, 1981.

[18] W. Fischer and K. Meier-Hellstern, "The Markov-modulated poisson process (MMPP) cookbook," *Perform. Eval.*, vol. 18, pp. 149–171, 1992.

[19] H. Heffes and D.M. Lucantoni, "A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance," *IEEE J. Select. Areas Commun.*, vol. 4, no. 6, pp. 856–868, Sep. 1986.

[20] H. Takagi, *Queueing Analysis, Volume 1: Vacation and Priority Systems, Part 1*, North-Holland, 1991.

[21] Jr. R.G. Miller, "Priority queues," *SIAM Ann. Math. Stat.*, vol. 31, no. 1, pp. 86–103, Mar. 1960.

[22] J.M. Harrison, "Dynamic scheduling of a two-class queue: small interest rates," *SIAM J. Appl. Math.*, vol. 31, no. 1, pp. 51–61, Jul. 1976.

[23] D.P. Heyman, "A priority queueing system with server interference," *SIAM J. Appl. Math.*, vol. 17, no. 1, pp. 74–82, Jan. 1969.

[24] L. Takacs, "Priority queues," *Oper. Res.*, vol. 12, pp. 63–74, 1964.

[25] T.T. Lee, "M/G/1/N queue with vacation time and exhaustive service discipline," *Oper. Res.*, vol. 32, no. 4, pp. 774–784, Jul.-Aug. 1984.

[26] D.I. Choi B.D. Choi, Y.C. KIM and D.K. Sung, "An analysis of M, MMPP/G/1 queues with QLT scheduling policity and Bernoullli schedule," *IEICE Trans. Commun.*, vol. E81-B, no. 1, pp. 13–21, Jan. 1998.

[27] D.I. Choi and Y. Lee, "Performance analysis of a dynamic priority queue for traffic control of bursty traffic in ATM networks," *IEE Proc.-Commun.*, vol. 148, no. 3, pp. 181–187, Jun. 2001.

[28] B.T. Hokstad, "A supplementary variable technique applied to the M/G/1 queue," *Scand. J. Stat.*, vol. 2, pp. 95–103, 1976.

[29] G.U. Hwang B.D. Choi and D.H. Han, "Supplementary variable method applied to the MAP/G/1 queueing system," *J. Austral. Math. Soc. Ser.B*, vol. 40, pp. 86–96, 1998.

[30] E. Cinlar, "Markov renewal theory," *Adv. Appl. Prob.*, vol. 12, pp. 222–261, 1969.

[31] *Supplement to IEEE Standard, Part 11: Wireless Medium Access Control and Physical Layer Specifications: High-speed Physical Layer in the 5 GHz Band*, P802.1a. Sep. 1999.

[32] E. Gilbert, "Capacity of burst noise channel," *Bell Syst. Tech. J.*, vol. 39, pp. 1253–1266, Sep. 1960.

[33] J.G. Proakis, *Digital Communications*, McGraw-Hill, 3rd edition, 1995.

[34] L. Kleinrock and F. Tobagi, "Packet switching in radio channels, Part II: the hidden terminal problem in carrier sense multiple access and the busy tone solution," *IEEE Trans. Commun.*, vol. 23, no. 12, pp. 1417–1433, Dec. 1975.