

國立交通大學

電信工程研究所

博士論文

即時性與非即時性訊務之無線網路

資源分配

Resource Allocation for Real-Time and
Non-Real-Time Traffic in Wireless
Networks

研究生：黃郁文

指導教授：李程輝 博士

中華民國一百零一年一月

即時性與非即時性訊務之無線網路資源分配

Resource Allocation for Real-Time and
Non-Real-Time Traffic in Wireless Network

研究生：黃郁文

Student: Yu-Wen Huang

指導教授：李程輝 博士

Advisor: Dr. Tsern-Huei Lee

國立交通大學

電信工程研究所

博士論文

A Dissertation

Submitted to Institute of Communication Engineering
College of Electrical Engineering
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in
Communication Engineering
January 2012
Hsinchu, Taiwan

中華民國一百零一年一月

即時性與非即時性訊務之無線網路資源分配

學生：黃郁文

指導教授：李程輝 博士

國立交通大學電信工程研究所

摘要

在本篇論文中，我們首先討論無線網路(IEEE 802.11e WLANs、以 OFDMA 技術為基礎的系統)資源分配技術。接著，將相關經驗應用至廣義有線系統即時性訊務多工器。

在 IEEE 802.11e WLANs 中，我們用高效能 TXOP 分配演算法、多工機制與相關的允入控制單元推廣 IEEE 802.11e HCCA 規格標準中的樣本排程器以保證不同變動位元速率訊務的不同服務品質保證需求(延遲限制、封包遺失率)。其中，我們透過定義等效訊務流和集成封包遺失率來得到訊務內與訊務間多工增益。並藉此達到高頻寬使用效率。再者，我們採用加權遺失公平的服務排程演算法將集成 TXOP 分配給各個訊務流。電腦模擬結果顯示我們提出的方法可以達到訊務流之服務品質需求，並且與先前研究比較起來，可以達到較高的頻寬使用效率。

在以 OFDMA 技術為基礎的系統中，我們提出同時處理即時性與非即時性訊務的資源分配演算法。其中，對於即時性訊務而言，假設其服務品質需求為延遲限制與資料遺失率。接著，根據訊務流之延遲限制與資料遺失率計算『最小所需頻寬』，然後將資源分配定義為滿足訊務流之『最小所需頻寬』下，最大化系統吞吐量之最佳化問題。資源分配結束後，若用戶端連結多個訊務流，則採用等比例遺失排程演算法決定訊務流間資源分配。萬一現有資源無

法提供每個訊務流最小所需頻寬，則將資源分配問題轉為最大化即時性訊務傳送量。其中，每個用戶所得資源不得超過其最小所需頻寬。此外，我們也設計『先行處理器』以最大化滿足服務品質需求之訊務流數。在本論文中，我們證明，在任意訊框中，若任意排程演算法可滿足訊務流之服務品質需求，則我們提出之等比例排程演算法亦可。電腦模擬結果亦顯示我們提出之演算法相較於先前的研究，擁有較佳效能。

論文的最後一個部份，我們研究可處理變動封包長度之多工系統。我們提出等比例遺失佇列管理演算法，使其與近期限優先之排程演算法結合提供即時性訊務之不同服務品質需求(延遲限制與資料遺失率)。我們指出，若以等效頻寬為指標，我們所提出之等比例遺失演算法為最佳佇列管理演算法。等比例遺失演算法假設封包可以無限制切割。為了更貼近實際封包交換網路，我們亦提出二個以封包為基本單位之佇列管理演算法。其中一個演算法為 G-QoS 演算法之直接推廣，另外一個則根據等比例遺失演算法的結果設計。電腦模擬結果指出，根據等比例遺失演算法設計的佇列管理演算法(以封包為基本單位)比 G-QoS 演算法之直接推廣，擁有較佳效能。

Resource Allocation for Real-Time and Non-Real-Time Traffic in Wireless Networks

Student: Yu-Wen Huang

advisor: Dr. Tsern-Huei Lee

Institute of Communication Engineering
National Chiao Tung University

Abstract

In this dissertation, we firstly studied resource allocation technique for wireless network such as IEEE 802.11e WLANs and OFDMA-based systems. Then, extend the developed results to a general multiplexer for real-time traffic in wired systems.

In IEEE 802.11e WLANs, we generalize the sample scheduler described in IEEE 802.11e HCCA standard with an efficient TXOP allocation algorithm, a multiplexing mechanism, and the associated admission control unit to guarantee QoS for VBR flows with different delay bound and packet loss probability requirements. We define equivalent flows and aggregate packet loss probability to take advantage of both intra-flow and inter-flow multiplexing gains so that high bandwidth efficiency can be achieved. Moreover, the concept of proportional-loss fair service scheduling is adopted to allocate the aggregate TXOP to individual flows. From numerical results obtained by computer simulations, we found that our proposed scheme meets QoS requirements and results in much higher bandwidth efficiency than previous algorithms.

In OFDMA-based Systems, we present a resource allocation algorithm for OFDMA-based

systems which handles both real-time and non-real-time traffic. For real-time traffic, the QoS requirements are specified with delay bound and loss probability. The resource allocation problem is formulated as one which maximizes system throughput subject to the constraint that the bandwidth allocated to a flow is no less than its minimum requested bandwidth, a value computed based on loss probability requirement and running loss probability. A user-level proportional-loss scheduler is adopted to determine the resource share for flows attached to the same subscriber station (SS). In case the available resource is not sufficient to provide every flow its minimum requested bandwidth, we maximize the amount of real-time traffic transmitted subject to the constraint that the bandwidth allocated to an SS is no greater than the sum of minimum requested bandwidths of all flows attached to it. Moreover, a pre-processor is added to maximize the number of real-time flows attached to each SS that meet their QoS requirements. We show that, in any frame, the proposed proportional-loss scheduler guarantees QoS if there is any scheduler which guarantees QoS. Simulation results reveal that our proposed algorithm performs better than previous works.

Finally, we study a multiplexing system which handles variable-length packets. A proportional loss (PL) queue management algorithm is proposed for packet discarding, which combined with the work-conserving EDF service discipline, can provide QoS guarantee for real-time traffic flows with different delay bound and loss probability requirements. We show that the proposed PL queue management algorithm is optimal because it minimizes the effective bandwidth among all stable and generalized space-conserving schemes. The PL queue management algorithm

is presented for fluid-flow models. Two packet-based algorithms are investigated for real packet switched networks. One of the two algorithms is a direct extension of the G-QoS scheme and the other is derived from the proposed fluid-flow based PL queue management algorithm. Simulation results show that the scheme derived from our proposed PL queue management algorithm performs better than the one directly extended from the G-QoS scheme.



Acknowledgements

First of all, I would like to represent my deeply gratitude to my advisor, Dr. Tsern-Huei Lee, for his constantly teaching, support and encouragement.

Secondly, I would like to represent my sincere appreciation to my parents, my families and my girlfriend for their love, support and encouragement.

Special thank to all the staff of Network Technology Laboratory, Institute of Communication Engineering, National Chiao Tung University for their enthusiastic help.

Finally, this dissertation is dedicated to my dear grandma living in the heaven. I will keep her teaching, love and spirit in my mind forever.

Contents

摘要.....	i
Acknowledgements	vi
Contents	vii
List of Tables.....	ix
List of Figures.....	x
Chapter 1 Introduction.....	1
Chapter 2 Related Works.....	7
2.1. Resource Allocation in IEEE 802.11e HCCA	7
2.2. Resource Allocation in OFDMA-Based Systems.....	10
2.3. Optimal Queue Management Algorithm for ATM Networks.....	18
Chapter 3 Resource Allocation for Real-Time Traffic in IEEE 802.11e WLANs	22
3.1. System Model	23
3.2. Aggregate TXOP Allocation Algorithm	27
3.3. Proportional-loss Service Scheduler	33
3.4. The Associated Admission Control Unit	40
3.5. Simulation Results	41
Chapter 4 Resource Allocation for Real-Time and Non-Real-Time Traffic in OFDMA-Based Systems	52
4.1. System Model	53
4.2. The Proposed Scheme.....	53
4.3. Simulation Results	65
Chapter 5 Optimal Queue Management Algorithm for Real-Time Traffic	78
5.1. System Model	79
5.2. The Proposed PL Queue Management Algorithm.....	81

5.3. Packet-based Systems	95
5.4. Simulation Results	98
Chapter 6 Conclusions	104
Biography	107
Appendix A Derivations of all equations and proofs of all lemmas and theorems	115
Appendix B Pseudo codes of the proposed algorithms	122
Vita	127
Publication List	128



List of Tables

Chapter 3

Table 3.1	Related parameters used in simulations.....	42
Table 3.2	TSPECs of traffic flows attached to Type I, Type II and Type III QSTAs.....	43
Table 3.3	The 99% confidence intervals of packet loss probability of flows attached to Type I, Type II and Type III QSTAs.....	44
Table 3.4	Over-allocation Ratio of Type I, Type II and Type III QSTAs.....	49
Table 3.5	Performance comparison for our proposed scheme and PRO-HCCA.....	51

Chapter 4

Table 4.1	Calculation of $R_{n,k}^*[t]$ and the resulting $P_{n,k}^*[t]$ for four conditions.....	57
Table 4.2	Parameters of simulation environment, traffic characteristics, QoS requirements and adopted modulation and coding scheme.....	67
Table 4.3	Loss probabilities for users attached with one Type I and one Type II real-time flows.....	74
Table 4.4	Number of Type I and Type II flows which meet their QoS requirements in the second scenario.....	74

Chapter 5

Table 5.1	Ω_k and Λ_k , $1 \leq k \leq 5$ for the example illustrated in Fig. 5.2.....	86
Table 5.2	Traffic characteristics and QoS requirements of the five flows generated from video trace files.....	98
Table 5.3	Steady-state (normalized) packet loss probability for flows generated from video trace files.....	100

List of Figures

Chapter 3

Fig. 3.1	Static and periodic schedule for 802.11e HCCA.....	24
Fig. 3.2	The system architecture of our proposed scheme.....	26
Fig. 3.3	The structure of sub-queues for $Queue_{i,j}$	33
Fig. 3.4	Running packet loss probabilities of Jurassic Park I attached to Type I QSTA.	46
Fig. 3.5	Running packet loss probabilities of Lecture Camera attached to Type I QSTA.....	47
Fig. 3.6	Running packet loss probabilities of Lecture Camera attached to Type I QSTA.....	48
Fig. 3.7	Comparison of admissible region.....	51

Chapter 4

Fig. 4.1	Architecture of the proposed scheme.....	54
Fig. 4.2	The relationship between $P_{n,k}[t]$ and $R_{n,k}[t]$	56
Fig. 4.3	Throughputs of various schemes in the first scenario.	69
Fig. 4.4	Loss probabilities of SSs attached with real-time traffic flows in the first scenario.	70
Fig. 4.5	Throughput comparison between proposed:ILP and proposed:Matrix schemes.....	75
Fig. 4.6	Loss probability comparison between proposed:ILP and proposed:Matrix schemes.	76
Fig. 4.7	Throughputs of various schemes in the second scenario.....	77

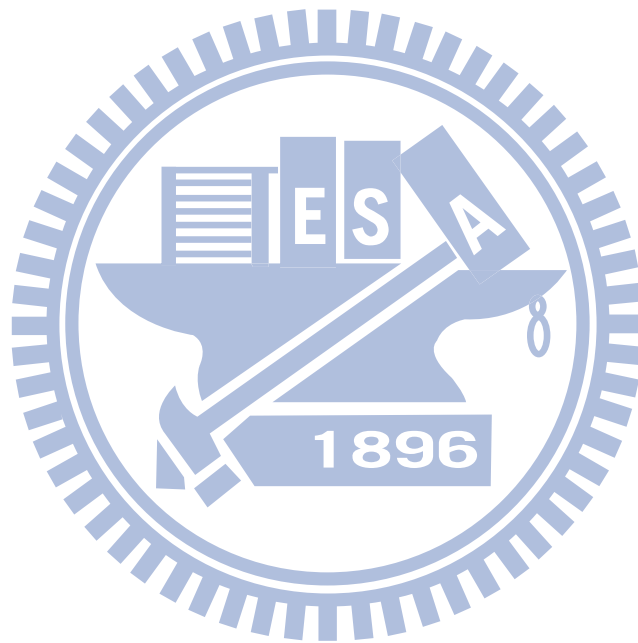
Chapter 5

Fig. 5.1	Architecture of the investigated multiplexer system and the structure of virtual sub-queues, $Queue_k^m$, $1 \leq m \leq \beta_k$, for $Queue_k$	80
----------	--	----

Fig. 5.2 An example for illustrating PL queue management algorithm for traffic flows with identical delay bound requirement.....85

Fig. 5.3 An example for illustrating PL queue management algorithm for traffic flows with different delay bound requirement.....94

Fig. 5.4 Sample Path of packet loss probability for video trace files with (a) our proposed PL queue management algorithm (b) Packet-based algorithm I (c) Packet-based algorithm II adopted.103



Chapter 1

Introduction

Because of the rapid proliferation of real-time multimedia applications such as VoIP and streaming video, providing quality of service (QoS) guarantee for individual traffic flows in current communication networks becomes an important issue. Generally speaking, QoS provisioning includes guarantee of maximum packet delay and packet loss probability.

For a traffic flow, the maximum tolerable delay of all its packets is called the delay bound of the flow. Packet loss probability is normally defined as the ratio of packets which are discarded due to buffer overflow or deadline violation to the total number of packets arrived. Buffer overflow occurs if a packet arrives when buffer is full, and deadline violation means that a packet is placed in the buffer longer than its delay bound. It is often acceptable for a real-time application to lose some packets as long as the packet loss probability is below a desired pre-specified value.

To provide QoS support in WLANs, a new enhancement of WLANs, called IEEE 802.11e [1]

was introduced, and this amendment has been combined into WLAN standard [2]. The QoS-aware coordination function proposed in IEEE 802.11e is called Hybrid Coordination Function (HCF). This function consists of two channel access mechanisms. One is contention-based Enhanced Distributed Channel Access (EDCA) and the other is contention-free HCF Controlled Channel Access (HCCA). The contention-free nature makes HCCA a better choice for QoS support than EDCA [3].

HCCA requires a centralized QoS-aware coordinator, called Hybrid Coordinator (HC), which has a higher priority than normal QoS-aware stations (QSTAs) in gaining channel control. HC can gain control of the channel after sensing the medium idle for a PCF inter-frame space (PIFS) that is shorter than DCF inter-frame space (DIFS) adopted by QSTAs. After gaining control of the transmission medium, HC polls QSTAs according to its polling list. In order to be included in HC's polling list, a QSTA needs to negotiate with HC by sending the Add Traffic Stream (ADDTTS) frame. In this frame, the QSTA describes the traffic characteristics and the QoS requirements in the Traffic Specification (TSPEC) field. Based on the traffic characteristics and the QoS requirements, HC calculates the scheduled service interval (SI) and transmission opportunity (TXOP) duration for each admitted flow. Upon receiving a poll, the polled QSTA either responds with QoS-Data if it has packets to send or a QoS-Null frame otherwise. When the TXOP duration of some QSTA ends, HC gains the control of channel again and either sends a QoS-Poll to the next station on its polling list or releases the medium if there is no more QSTA to be polled.

In this dissertation, we present an efficient scheduling scheme for HCCA to provide QoS guarantee for VBR traffic flows with different delay bound and packet loss probability requirements. The proposed scheme achieves both intra-flow and inter-flow multiplexing gains. In this scheme, HC calculates TXOP duration and performs admission control while every QSTA implements a proportional-loss fair service scheduler to determine how the allocated TXOP is shared by traffic flows attached to it. Numerical results obtained by computer simulations show that our proposed TXOP allocation algorithm results in much better performance than previous works. Moreover, the proposed proportional-loss fair service scheduler successfully manages the TXOP so that different delay bound and packet loss probability requirements of all traffic flows can be fulfilled.

In OFDMA-based wireless systems, such as IEEE 802.16 [4] and the Long Term Evolution (LTE) [5], channel access is partitioned into frames in the time domain and sub-channels in the frequency domain to achieve multi-user and frequency diversities. One obvious performance metric to evaluate resource allocation schemes is system throughput. A simple strategy to achieve high system throughput is to allocate more resources to users with better channel qualities. This strategy, unfortunately, may lead to starvation and cause QoS violation to real-time applications attached to users who have poor channel qualities. A well-designed resource allocation scheme should, therefore, take QoS support into consideration while maximizing system throughput.

In this dissertation, we present a resource allocation algorithm which tries to maximize system

throughput with QoS support for real-time traffic flows. Our contributions include: 1) define and derive the minimum requested bandwidth of each real-time flow based on the loss probability requirement and the running loss probability, 2) formulate the resource allocation problem as one which maximizes system throughput subject to the constraint that the bandwidth allocated to a flow is greater than or equal to its minimum requested value, 3) propose a user-level proportional-loss (PL) scheduler for multiple real-time traffic flows attached to the same subscriber station (SS) to share the allocated resource, and 4) modify the resource allocation problem to maximize the amount of real-time traffic transmitted and add a pre-processor in front of the PL scheduler to maximize the number of real-time flows attached to each SS that meet their QoS requirements, when the available resource is not sufficient to provide each flow its minimum requested bandwidth. We show that, in any frame, the proposed PL scheduler guarantees QoS if there is any scheduler which guarantees QoS. Simulation results reveal that our proposed algorithm performs better than previous works.

Finally, we consider a general multiplexer operated in wired system. In order to provide QoS guarantee for traffic flows with different delay bound and packet loss probability requirements, it is necessary to be equipped with two types of priority schemes: time priority and loss priority. Note that only stable priority schemes are considered in this dissertation, where a priority scheme is said to be stable iff its priority assignment policy does not change over time. A time priority scheme is responsible for service scheduling. It assigns time priority to all the buffered packets so that the multiplexer can select the highest priority packet for service. There are two types of time priority

schemes: static and dynamic. A static time priority scheme assigns priorities to flows while a dynamic scheme does so to packets. Rate monotonic [6] is a famous static time priority scheme, while generalized processor sharing (GPS) [7], [8] and earliest deadline first (EDF) are well-known dynamic time priority schemes. A loss priority scheme is in charge of queue management and normally has two main functions. One determines the necessity to discard packets. When there are some packets needed to be discarded, the other one identifies which packets in the buffer should be discarded. Most of the previous works regarding loss priority assignment can be classified into two categories, namely, push-out [9]-[12] and partial buffer sharing [13]-[15]. In a push-out scheme, when buffer is full upon a packet arrival, the packet with lowest priority is pushed out or discarded. Obviously, tail drop can be considered as a special pushout scheme where loss priorities are assigned based on packet arrival time. In a partial buffer sharing scheme, each traffic flow is assigned a threshold value, and an arriving packet is admitted into the buffer iff the current buffer occupancy does not exceed the threshold assigned to the traffic flow it belongs to. Push-out is more efficient than partial buffer sharing because it minimizes overall packet loss. However, the complexity of push-out is likely to be higher than that of partial buffer sharing.

In this dissertation, we study a multiplexer which provides heterogeneous QoS guarantee, delay bound and loss probability for variable-length packets. In such a multiplexer, it should be more meaningful to consider the amount of data loss rather than the number of packet loss. Consequently, we define loss probability as the ratio of the total amount of data lost to that of data

arrived and then adopt it as the metric for evaluating the performances of schemes handling variable-length packets. A proportional-loss (PL) queue management algorithm for fluid-flow model is proposed for data discarding. The proposed PL queue management algorithm tries to minimize the total amount of data loss and balance the normalized running loss probabilities for all admitted traffic flows. When combined with the EDF service discipline, it is an effective and efficient scheme for both time and loss priority assignments. We show that the combined scheme is optimal because it minimizes the effective bandwidth under the generalized space-conserving constraint. We further investigate and compare two packet-based queue management schemes. One is a direct extension of the G-QoS scheme and the other is a derivative of the proposed PL queue management algorithm. Results show that the scheme derived from the proposed PL queue management algorithm outperforms the one extended from the G-QoS scheme.

The rest of this dissertation is organized as follows. In Chapter 2, we review related works. Then, we present the proposed schemes and evaluate their performances for 1) IEEE 802.11e HCCA, 2) OFDMA-based systems and 3) a general wired systems in Chapter 3, 4 and 5, respectively. Finally, Chapter 6 contains the conclusions drawn for this dissertation.

Chapter 2

Related Works

In this chapter, we review the related works regarding to perform resource allocation in IEEE 802.11e HCCA and OFDMA-based systems, respectively. Before leaving this chapter, we describe the optimal queue management schemes for ATM networks.

2.1. Resource Allocation in IEEE 802.11e HCCA

In IEEE 802.11e HCCA, resource is partitioned and allocated to users in the time domain. As a result, performing resource allocation can be achieved by some scheduling schemes. Scheduling schemes designed for IEEE 802.11e HCCA can be classified into two categories, namely, static and dynamic. In a static scheduling scheme, HC allocates the same TXOP duration to a QSTA every time it is polled. Moreover, the SI is often selected as the minimum of delay bound requirements of all traffic flows. The sample scheduler provided in IEEE 802.11 standard document [2] is a typical example of static scheduling scheme. The HC of the sample scheduler allocates TXOP duration

based on mean data rate and nominal MAC service data unit (MSDU) size. It performs well for constant bit rate (CBR) traffic. For variable bit rate (VBR) traffic, packet loss may occur seriously. In [16], some static scheduling scheme was proposed to generalize the sample scheduler with modified TXOP allocation algorithm and admission control unit so that both delay bound and packet loss probability requirements of admitted traffic flows can be fulfilled. To achieve the same goal, the Rate-Variance envelope based Admission Control (RVAC) algorithm [17] uses token buckets for traffic shaping. With the token buckets, the envelope of traffic arrival can be determined. Using the traffic envelope and the given delay bound requirement, one can compute the packet loss probability for an allocated bandwidth. Although the fact that many real-time VBR applications can tolerate packet loss to certain degree was taken into consideration in these works to improve bandwidth efficiency, it was assumed that all traffic flows have the same delay bound of one SI and the same packet loss probability requirement. Since different real-time applications may require distinct delay bound and packet loss probability requirements, ones can manage the bandwidth more efficiency if each requirement can be considered individually.

In contrast to static ones, a dynamic scheduling scheme allocates TXOP duration to a QSTA dynamically, according to system status, to provide delay bound guarantee and/or fairness. Some dynamic scheduling schemes can be found in [18]-[25]. To achieve delay bound guarantee, a dynamic scheduling scheme requires QSTAs to timely report their queue statuses to HC. As an example, in the prediction and optimization-based HCCA (PRO-HCCA) scheme [20] that was

proposed recently, the SI is set to be smaller than or equal to half of the minimum of delay bounds requested by all traffic flows. As a consequence, compared with static scheduling schemes, QSTAs are polled more frequently, which implies higher overhead for poll frames. Furthermore, static and periodic polling allows QSTAs to easily eliminate overhearing to save energy. Therefore, although dynamic scheduling has the potential to achieve high bandwidth efficiency, it is worthwhile to study static scheduling schemes. In the following paragraphs, we give a detailed description of the sample scheduler.

- The Sample Scheduler [2]

Consider $QSTA_a$ which has n_a flows. Let ρ_l, L_l denote, respectively, the mean data rate and the nominal MSDU size of the l^{th} flow attached to $QSTA_a$. HC calculates $TXOP_a$ as follows. Firstly, it decides, for flow l , the average number of packets \overline{N}_l that arrive at the mean data rate during one SI

$$\overline{N}_l = \left\lceil \frac{\rho_l \times SI}{L_l} \right\rceil \quad (1)$$

Secondly, the TXOP duration for this flow is obtained by

$$TD_l = \max \left\{ \overline{N}_l \times \left(\frac{L_l}{R_a^{\min}} + O \right), \frac{L_{\max}}{R_a^{\min}} + O \right\} \quad (2)$$

where R_a^{\min} is the minimum physical transmission rate of $QSTA_a$, and L_{\max} and O denote, respectively, the maximum allowable size of MSDU and per-packet overhead in time units. The

overhead O includes the transmission time for an ACK frame, inter-frame space, MAC header, CRC field and PHY PLCP preamble and header.

Finally, the total TXOP duration allocated to $QSTA_a$ is given by

$$TXOP_a = \left(\sum_{l=1}^{n_a} TD_l \right) + SIFS + t_{POLL} \quad (3)$$

where $SIFS$ and t_{POLL} are, respectively, the short inter-frame space and the transmission time of a CF-Poll frame.

Admission control is performed as follows. Assume that $QSTA_a$ negotiates with HC for admission of a new traffic flow, i.e., the $(n_a + 1)^{th}$ flow of $QSTA_a$. For simplicity, we further assume that the delay bound of the new flow is not smaller than SI . The process is similar if this assumption is not true. HC updates $TXOP_a$ as $TXOP'_a = TXOP_a + TD_{n_a+1}$. The new flow is admitted iff the following inequality is satisfied

$$\frac{TXOP'_a}{SI} + \sum_{k=1, k \neq a}^K \frac{TXOP_k}{SI} \leq \frac{T_b - T_{cp}}{T_b} \quad (4)$$

where T_{cp} is the time used for EDCA traffic during one beacon interval.

2.2. Resource Allocation in OFDMA-Based Systems

In OFDMA-based systems, resource is partitioned into frames in the time domain and sub-channels in the frequency domain. A well-designed resource allocation algorithm should take system throughput, fairness and QoS support into account.

Several previous works, say, [26], [27], adopted the concept of proportional fairness (PF) to eliminate starvation while maintaining acceptable system throughput. These schemes, although achieve a kind of fairness among users, are not suitable for QoS support. In [28] and [29], the ideas of PF and static minimum bandwidth guarantee were combined to support multiple service classes. This enhanced algorithm, however, does not take delay bound and loss probability requirements of real-time flows into consideration and thus is unlikely to provide QoS support well.

In [30], a power and sub-carrier allocation policy was proposed for system throughput optimization with the constraint that the average delay of each traffic flow is controlled to be lower than its pre-defined level. Guaranteeing average delay, however, is in general not sufficient for real-time applications. The results presented in [31] reveal that dynamic power allocation can only give a small improvement over fixed power allocation with an effective adaptive modulation and coding (AMC) scheme. As a result, to reduce the complexity, it is reasonable to design resource allocation schemes under the assumption that equal power is allocated to each sub-channel.

Some resource allocation algorithms were proposed, assuming equal-power allocation, to assign a user a higher priority for channel access if the deadline of its head-of-line (HOL) packet is smaller [32]-[35]. A simple scheme, called modified largest weighted delay first (M-LWDF), which uses a kind of utility function that is sensitive to loss probability and delay bound requirements as well as delay of HOL packets, was presented in [33]. Obviously, considering only

the deadlines of HOL packets is not optimal. A QoS scheduling and resource allocation algorithm which considers deadlines of all packets was presented in [36]. This scheme requires high computational complexity and thus may not be practical for real systems. To reduce computational complexity, a matrix-based scheduling algorithm was proposed in [27]-[29]. The M-LWDF, the scheme proposed in [36] and the matrix-based scheduling algorithm are related to our work and will be reviewed in the following paragraphs. For ease of presentation, we firstly describe the system model and then depict the details of each scheme.

- System model

We consider a single-cell OFDMA-based system which consists of one base station (BS) and multiple users or subscriber stations (SSs). Time is divided into frames, and the duration of a frame is equal to T_{frame} . In a frame, there are M sub-channels and S time slots. We assume that the sub-channel statuses of different SSs are independent. Moreover, for a given SS, its statuses on the M sub-channels are also independent. The channel quality for a given SS on a specific sub-channel is fixed during one frame. Transmission power is equally allocated to each sub-channel. To improve reliable transmission rate, an effective AMC scheme is adopted to choose a transmission mode based on the reported signal-to-noise ratio (SNR). We only consider downlink transmission.

For ease of description, we assume that no SS is attached with both real-time and non-real-time traffic flows. Let Γ_{RT} and Γ_{NRT} represent, respectively, the sets of SSs that are attached with

real-time and non-real-time traffic flows. Further, let $\Gamma = \Gamma_{RT} \cup \Gamma_{NRT}$. We shall use K_n to denote the number of traffic flows attached to SS n . All non-real-time flows attached to the same SS are aggregated into one so that $K_n = 1$ if SS $n \in \Gamma_{NRT}$. The QoS requirements of real-time traffic flows are specified by delay bound and loss probability. The k^{th} flow attached to SS n is denoted by $f_{n,k}$. If SS $n \in \Gamma_{RT}$, then the delay bound and loss probability requirements of $f_{n,k}$ are represented by $D_{n,k} \cdot T_{frame}$ and $P_{n,k}$, respectively. Data are assumed to arrive at the beginning of frames.

In the BS, a separate queue is maintained for each real-time traffic flow while non-real-time data are stored per SS. Assume that SS $n \in \Gamma_{RT}$. The data of flow $f_{n,k}$ are buffered in $Queue_{n,k}$, which can be partitioned into $D_{n,k}$ disjoint virtual sub-queues, denoted by $Queue_{n,k}^d$, $1 \leq d \leq D_{n,k}$, where $Queue_{n,k}^d$ contains the data in $Queue_{n,k}$ that can be buffered up to $d \cdot T_{frame}$ without violating their delay bounds. We shall use $Q_{n,k}^d[t]$ to represent the size of $Queue_{n,k}^d$ at the beginning of the t^{th} frame (including the newly arrived), $Q_{n,k}[t] = \sum_{d=1}^{D_{n,k}} Q_{n,k}^d[t]$, and $Q_n[t] = \sum_{k=1}^{K_n} Q_{n,k}[t]$. Data which violate their delay bounds are dropped. It is assumed that the size of each queue is sufficiently large so that no data will be dropped due to buffer overflow. To simplify notation, the queue for storing data of SS $n \in \Gamma_{NRT}$ is denoted by $Queue_n$.

Resource allocation is performed at the beginning of each frame and, therefore, it suffices to consider one specific frame, say the t^{th} frame. For SS n , we denote its maximum achievable

transmission rate on the m^{th} sub-channel in the t^{th} frame and its long-term average throughput up to the t^{th} frame by $r_{n,m}[t]$ and $\bar{r}_n[t]$, respectively.

- Scheme of [36]

In [36], resource allocation is formulated as an optimization problem which maximizes some utility function subject to QoS guarantee. It consists of two stages. In the first stage, resources are allocated to real-time traffic flows only. If there are un-allocated resources after the first stage, the second stage is performed to allocate the remaining resources to non-real-time traffic.

In the first stage, called real-time QoS scheduling, the minimum requested bandwidth of each real-time traffic flow is calculated by $R_n^{\min} = \sum_{k=1}^{K_n} \sum_{d=1}^{D_{n,k}} Q_{n,k}^d [t] / d^\beta$. Note that substituting β with 0, 1, or ∞ corresponds, respectively, to strict priority [37], average QoS provisioning [38], or urgent [39] scheduling policy. With the assumption that sub-channel is the smallest resource granularity, the first stage aims to minimize the total number of sub-channels used to serve the sum of calculated minimum requested bandwidths of all real-time flows. This problem can be modeled as maximum weighted bipartite matching (MWBM) and solved by the famous On Kuhn's Hungarian method, whose complexity is $O(M |\Gamma_{RT}| (\min(M, |\Gamma_{RT}|))^2)$ [40], where $|\Gamma_{RT}|$ is the size of Γ_{RT} .

In the second stage, the m^{th} sub-channel, if still available, is allocated to the SS which satisfies $n^* = \arg \max_{n \in \Gamma_{NRT}} U'_n(\bar{r}_n[t]) r_{n,m}[t]$, where $U'_n(x)$, called marginal utility function, is the first derivative of the utility function. For every SS, the utility function, defined by

α -proportional fairness [41], is given by

$$U^\alpha(x) = \begin{cases} (1-\alpha)^{-1} x^{1-\alpha} & \text{if } \alpha \neq 1 \\ \log(x) & \text{otherwise} \end{cases} \quad (5)$$

where x represents the average throughput. Note that the policy corresponds to maximum throughput, proportional fairness, or max-min fairness if α is chosen to be 0, 1, or ∞ , respectively.

It was shown in [36] that the above scheme with $\beta=1$ makes a reasonable trade-off between QoS support and maximization of system utility. However, it has some drawbacks. Firstly, assuming the granularity of resource to be sub-channels can result in waste of bandwidth. In current standards such as IEEE 802.16 and LTE, a sub-channel can be shared by multiple SSs. Secondly, although the number of sub-channels used to serve real-time traffic is minimized in the first stage, the remaining service capability for non-real-time traffic may not be maximized. This is because the qualities of remaining sub-channels could be poor for SSs attached with non-real-time traffic flows. Thirdly, calculation of the minimum requested bandwidth for each real-time traffic flow does not take its loss probability requirement into consideration. Real-time traffic usually can tolerate data loss to certain degree. System throughput can be improved significantly if one takes advantage of this feature in resource allocation. Finally, the complexity of the Hungarian method could make this scheme infeasible for a real system.

- Matrix-based scheduling algorithm [27]

A matrix-based scheduling algorithm which tries to maximize the utility sum of all users with acceptable computational complexity was proposed in [27]. In this scheme, a matrix $U = [u_{n,m}]$ of dimension $|\Gamma| \times M$ is defined for resource allocation, where $u_{n,m} = r_{n,m}[t]/\bar{r}_n[t]$ represents the marginal utility of user n on sub-channel m . For sub-channel m , let s_m represent the number of slots that have not been allocated and $x_{n,m}$ the number of slots allocated to SS n . Initially, we have $s_m = S$ and $x_{n,m} = 0$, $n \in \Gamma$, $1 \leq m \leq M$. The matrix-based scheduling algorithm consists of three steps: 1) Find an (n^*, m^*) which satisfies $u_{n^*, m^*} = \max_{1 \leq n \leq |\Gamma|, 1 \leq m \leq M} \{u_{n,m}\}$. 2) Set $x_{n^*, m^*} = \min(s_{m^*}, \lceil Q_{n^*}[t]/r_{n^*, m^*}[t] \rceil)$ (allocate $\lceil Q_{n^*}[t]/r_{n^*, m^*}[t] \rceil$ or all the remaining slots of sub-channel m^* , whichever is smaller, to user n^*), $Q_{n^*}[t] = \max(0, Q_{n^*}[t] - r_{n^*, m^*}[t] \cdot x_{n^*, m^*})$ (update queue status of user n^*), and $s_{m^*} = s_{m^*} - x_{n^*, m^*}$ (update the remaining number of slots of sub-channel m^*). Replace the $(n^*)^{\text{th}}$ row of U by an all-zero row if $Q_{n^*}[t] = 0$ (user n^* does not need any more resource) and the $(m^*)^{\text{th}}$ column of U by an all-zero column if $s_{m^*} = 0$ (all slots of sub-channel m^* are allocated). 3) Update $\bar{r}_n[t]$. If $Q_{n^*}[t] > 0$, then re-calculate $u_{n^*, m} = r_{n^*, m}[t]/\bar{r}_{n^*}[t]$ for all $m \neq m^*$ (update the marginal utilities of user n^* on various sub-channels before allocating the remaining resources). The above three steps are repeatedly executed until all elements of U are replaced with zeroes. The resulting values of $x_{n,m}$, $n \in \Gamma$, $1 \leq m \leq M$, are the solutions. Assuming that $M \geq |\Gamma|$, the computational complexity of the matrix-based scheduling algorithm in the worst case is $O(M^2|\Gamma| + |\Gamma|^2)$, which happens when

$M - 1$ columns of U are replaced by all-zero columns one by one, followed by replacing the rows by all-zero rows one by one. Its complexity is $O(|\Gamma|^2 M + M^2)$ if $M < |\Gamma|$.

Note that the matrix-based scheduling algorithm takes queue occupancy into consideration. However, it does not consider QoS support. The same authors combined the idea of PF with static minimum bandwidth guarantee to support multiple service classes [28], [29]. A user whose channel quality is better than some threshold is guaranteed a pre-defined minimum bandwidth. This enhanced version, still, cannot provide QoS support well because it does not consider delay bound and loss probability requirements of real-time flows.

- Modified-Largest Weighted Delay First (M-LWDF) [33]

The goal of the M-LWDF scheme is to achieve $P(W_{n,k} > D_{n,k}) \leq P_{n,k}$ for all $n \in \Gamma_{RT}$, $1 \leq k \leq K_n$. In M-LWDF, the marginal utility of flow $f_{n,k}$ on sub-channel m is $\gamma_{n,k} \cdot W_{n,k}[t] \cdot r_{n,m}[t]$, where $W_{n,k}[t] \cdot T_{frame}$ is the delay of the HOL packet of $Queue_{n,k}$ at the beginning of frame t and $\gamma_{n,k}$ is an arbitrary positive constant. To transmit data, the flow with the largest marginal utility on some available sub-channel is selected for service. It was shown that M-LWDF is throughput-optimal in the sense that it is able to keep all queues stable if this is at all feasible to do with any scheduling algorithm. Moreover, it was reported that $\gamma_{n,k} = a_{n,k} / \bar{r}_n[t]$, where $a_{n,k} = -(\log P_{n,k}) / D_{n,k}$, performs very well. Clearly, for such a selection of $\gamma_{n,k}$, the marginal utility is sensitive to loss probability and delay bound requirements as well as delay of the

HOL packet. When combined with a token bucket control, M-LWDF can provide QoS support to flows with minimum bandwidth requirements. However, how to serve non-real-time flows with zero minimum bandwidth requirements was not studied. To compare its performance with that of our proposed scheme, we shall assume that the operation of M-LWDF is divided into two stages. In the first stage, only real-time traffic flows are considered. As a consequence, the first stage of M-LWDF is the same as that of the matrix-based scheduling, except for a different marginal utility function. The complexity of the first stage is $\max\{O(M^2|\Gamma_{RT}|+|\Gamma_{RT}|^2), O(|\Gamma_{RT}|^2 M + M^2)\}$. If there are un-allocated resources after the first stage, then the remaining resources are allocated in the second stage to non-real-time flows with zero minimum resource requirements. The goal of the second stage is to maximize system throughput. Assume that the matrix-based scheduling algorithm is adopted in the second stage. As a result, the complexity of the second stage is $\max\{O(M^2|\Gamma_{NRT}|+|\Gamma_{NRT}|^2), O(|\Gamma_{NRT}|^2 M + M^2)\}$.

2.3. Optimal Queue Management Algorithm for ATM Networks

To support heterogeneous QoS differentiation such as delay bound and packet loss probability, it is necessary to jointly design time priority and loss priority schemes. In [42]-[48], relative differentiated service, one approach in DiffServ framework, was proposed trying to provide heterogeneous QoS differentiation. In relative differentiated service, packets are grouped into multiple classes so that a packet belonging to a higher priority class receives better service than a

packet belonging to a lower one. The proportional differentiation model was proposed to refine the relative differentiated service with quantified QoS spacing. In proportional differentiation model, performance metrics such as average delay and/or packet loss probability are controlled to be proportional to the differentiation parameters chosen by network operators. Assume that there are N service classes. The average experienced delay and suffered packet loss probability of the i^{th} service class, denoted by \bar{d}_i and \bar{P}_i , respectively, are spaced from those of the j^{th} service class as $\bar{d}_i/\bar{d}_j = \delta_i/\delta_j$ and $\bar{P}_i/\bar{P}_j = \sigma_i/\sigma_j$, $1 \leq i, j \leq N$. Here, δ_i and σ_i denote, respectively, the delay and packet loss probability differentiation parameters of the i^{th} service class. The work presented for relative differentiated service successfully controls the average delays and packet loss probability in a proportional sense. However, this service model is not practical for real-time traffic. The reasons are stated as follows. 1) For real-time traffic, we believe it is more meaningful for a multiplexer to guarantee delay bounds rather than providing proportional average delays. 2) Since packets of real-time traffic have to be dropped whenever they violate their delay bound, buffer overflow can be eliminated by engineering the buffer space according to the delay bound of all real-time traffic and the service capability of the system. As a result, it is reasonable to assume that packet loss only results from deadline violation for a multiplexer dealing with real-time traffic.

In [49], the authors generalized the QoS scheme [11] and combined it with the earliest deadline first (EDF) service discipline to support multiple delay bound and cell loss probability requirements for real-time traffic flows in ATM networks, assuming cell loss only results from deadline violation.

This generalized version is named G-QoS. It was proved that the G-QoS scheme is optimal in the sense that it minimizes the effective bandwidth among all stable and generalized space-conserving schemes. A scheme is said to be generalized space-conserving if a packet is discarded only when it or some other packets buffered in the system will violate their delay bounds. Moreover, effective bandwidth refers to as the minimum required bandwidth to meet QoS requirements of all traffic flows. Two drawbacks of the G-QoS scheme are 1) it only handles fixed-length packets and 2) when batches of packets arrive, packet-by-packet processing requires high computational complexity. The G-QoS scheme and its original version, the QoS scheme, are related to our work and will be reviewed in the following paragraphs.

It is assumed that there are K traffic flows, namely, f_1, f_2, \dots, f_K , which are multiplexed into a system with transmission capability C and a single queue of size B . Consider f_k . Let P_k represent its packet loss probability requirement. The number of arrived and discarded packets (or cells) by time t are denoted by $A_k(t)$ and $L_k(t)$, respectively. The running packet loss probability $P_k(t)$ is defined as $P_k(t) = L_k(t)/A_k(t)$.

- The QoS scheme [11]

The QoS scheme is operated as follows. Assume that a packet arrives at time t , and the buffer is fully occupied. Define $D(t)$ as the set which contains indices of traffic flows that have at least one packet in the buffer (excluding the one under transmission). Let f_j be the flow in $D(t)$


such that $P_j(t)/P_j \leq P_k(t)/P_k$, $1 \leq k \leq K$. If the arriving packet belongs to f_j , then this packet is discarded. Otherwise, a packet which belongs to f_j is discarded and the arriving packet is admitted to the buffer. As was proved in [12], the QoS scheme is optimal in the sense that it achieves maximum bandwidth utilization among all stable and space-conserving schemes.

- The G-QoS scheme [49]

In the G-QoS scheme, it was assumed that the buffer is sufficiently large so that there is no cell loss due to lack of buffer space. The EDF policy was adopted as its service discipline. Upon arrival, a cell is marked with its deadline, which is equal to its arrival time plus the requested delay bound. Then, the schedulability test of the EDF scheduler is performed according to the deadlines of the newly arrival and all the other existing ones. The newly arrived cell is admitted into the buffer without discarding any cell if no cell will violate its delay bound, assuming that there is no more cell arrival in the future. Otherwise, a cell in the discarding set is lost. The discarding set $S(t)$ is the maximum subset of existing cells at time t , including the newly arrived one, such that the remaining cells in the system are schedulable if cell c is discarded for any $c \in S(t)$. Which cell is to be discarded is determined by the normalized running cell loss probabilities of traffic flows having cells in the discarding set. Among these traffic flows, a cell which belongs to the traffic flow with the smallest normalized running cell loss probability is discarded. It was proved that the G-QoS scheme is optimal in the sense that it minimizes the effective bandwidth among all stable and generalized space-conserving schemes.

Chapter 3

Resource Allocation for Real-Time Traffic in IEEE 802.11e WLANs



The Medium Access Control (MAC) of IEEE 802.11e defines a novel coordination function, namely, Hybrid Coordination Function (HCF), which allocates Transmission Opportunity (TXOP) to stations taking their quality of service (QoS) requirements into account. However, the reference TXOP allocation scheme of HCF Controlled Channel Access (HCCA), a contention-free channel access function of HCF, is only suitable for constant bit rate (CBR) traffic. For variable bit rate (VBR) traffic, packet loss may occur seriously. In this chapter, we generalize the reference design with an efficient TXOP allocation algorithm, a multiplexing mechanism, and the associated admission control unit to guarantee QoS for VBR flows with different delay bound and packet loss probability requirements. We define equivalent flows and aggregate packet loss probability to take advantage of both intra-flow and inter-flow multiplexing gains so that high bandwidth efficiency can be achieved. Moreover, the concept of proportional-loss fair service scheduling is adopted to

allocate the aggregate TXOP to individual flows. From numerical results obtained by computer simulations, we found that our proposed scheme meets QoS requirements and results in much higher bandwidth efficiency than previous algorithms.

3.1. System Model

The studied system consists of K QSTAs, called $QSTA_1$, $QSTA_2$, ..., and $QSTA_K$ such that $QSTA_i$ has n_i existing VBR flows. Transmission over the wireless medium is divided into SIs and the duration of each SI, denoted by SI , is a sub-multiple of the length of a beacon interval T_b . Moreover, an SI is further divided into a contention period and a contention-free period. The HCCA protocol is adopted during contention-free periods.

It is assumed that every QSTA has the capability to measure channel quality to determine a feasible transmission rate which yields a frame error rate sufficiently smaller than the packet loss probability requirements requested by all traffic flows attached to the QSTA. The relationship between measured channel quality and frame error rate can be found in [52].

The QoS requirements of traffic flows are specified with delay bound and packet loss probability. Every QSTA is equipped with sufficiently large buffer so that a packet is dropped if and only if (iff) it violates the delay bound. It is assumed that there are I different packet loss probabilities, represented by P_1 , P_2 , ..., and P_I with $P_1 > P_2 > \dots > P_I$, and J possible delay bounds, denoted by D_1 , D_2 , ..., and D_J with $D_1 < D_2 < \dots < D_J$. We assume that $D_1 = SI$ and

$D_j = \beta_j SI$ for some integer $\beta_j > 1$.

HC allocates TXOPs to QSTAs based on a static and periodic schedule. As illustrated in Fig. 3.1, the TXOP for $QSTA_k$, denoted by $TXOP_k$, is allocated every SI and is of fixed length. The length of scheduled SI is chosen to be the minimum of all requested delay bounds. Note that SI is updated if a new flow with delay bound smaller than those of existing ones is admitted or the existing flow with the smallest delay bound is disconnected and there is no other existing flow with the same delay bound. In this case, the TXOPs allocated to QSTAs have to be recalculated accordingly.

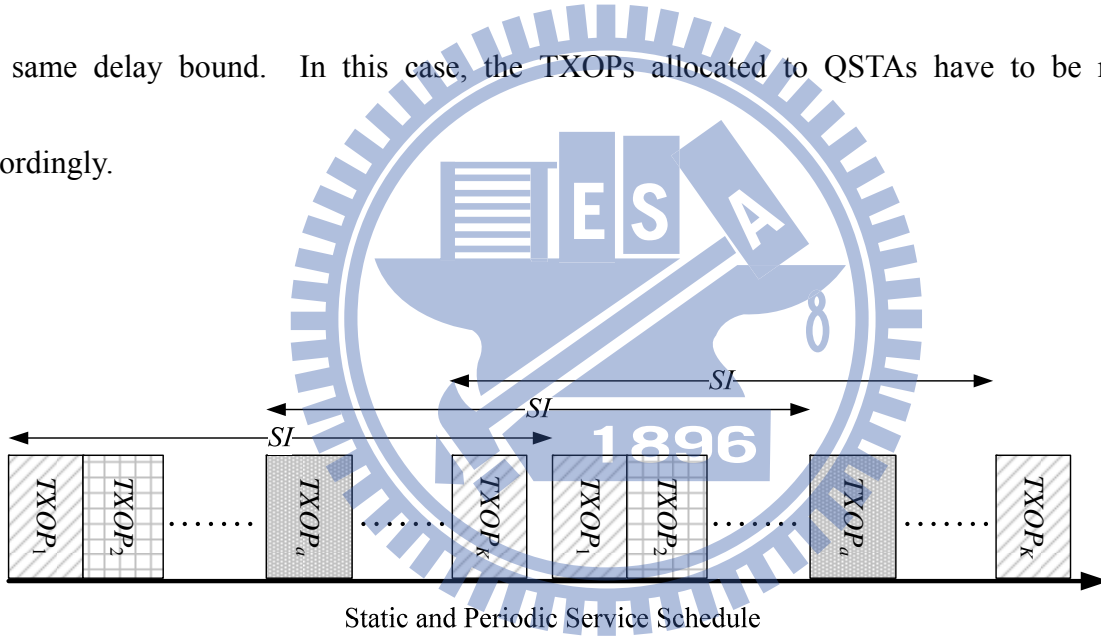


Fig. 3.1 Static and periodic schedule for 802.11e HCCA.

Consider the existing flows of a specific QSTA, say $QSTA_a$. The n_a flows attached to $QSTA_a$ are classified into groups according to their QoS requirements. Let $F_{i,j}$ represent the set which contains all traffic flows with packet loss probability P_i and delay bound D_j . Furthermore, let $F_i = \bigcup_{1 \leq j \leq J} F_{i,j}$ and $F = \bigcup_{1 \leq i \leq I} F_i$. To reduce computational complexity, we assume that the traffic

arrivals of different flows are independent Gaussian processes. Since sum of independent Gaussian random variables remains Gaussian, the aggregated flow of all the flows in set $F_{i,j}$ is Gaussian and will be represented by $f_{i,j}$. For convenience, we shall consider $f_{i,j}$ as a single flow. A separate queue, called $Queue_{i,j}$, is maintained for flow $f_{i,j}$, $1 \leq i \leq I$ and $1 \leq j \leq J$. Let $N(\mu_{i,j}, \sigma_{i,j}^2)$ denote the distribution of traffic arrival for flow $f_{i,j}$ in one SI. Note that the values of $\mu_{i,j}$ and $\sigma_{i,j}^2$ can be calculated by

$$\mu_{i,j} = E(N_{i,j}) \cdot E(X_{i,j}). \quad (6)$$

and

$$\sigma_{i,j}^2 = E(N_{i,j}) \cdot \text{VAR}(X_{i,j}) + E(X_{i,j})^2 \cdot \text{VAR}(N_{i,j}), \quad (7)$$

where $N_{i,j}$ and $X_{i,j}$ represent, respectively, the number of packets belonging to flow $f_{i,j}$ that arrive in one SI and the packet size.

Our proposed scheme consists of an aggregate TXOP allocation algorithm, the proportional-loss fair service scheduler, and the associated admission control unit. As mentioned before, TXOP allocation and admission control are performed in HC and proportional-loss fair service scheduler is implemented in QSTAs. An overview of our proposed scheme is depicted in Fig. 3.2. Once again, let us consider $QSTA_a$ with n_a traffic flows, which are classified into $I \times J$ groups according to their QoS requirements.

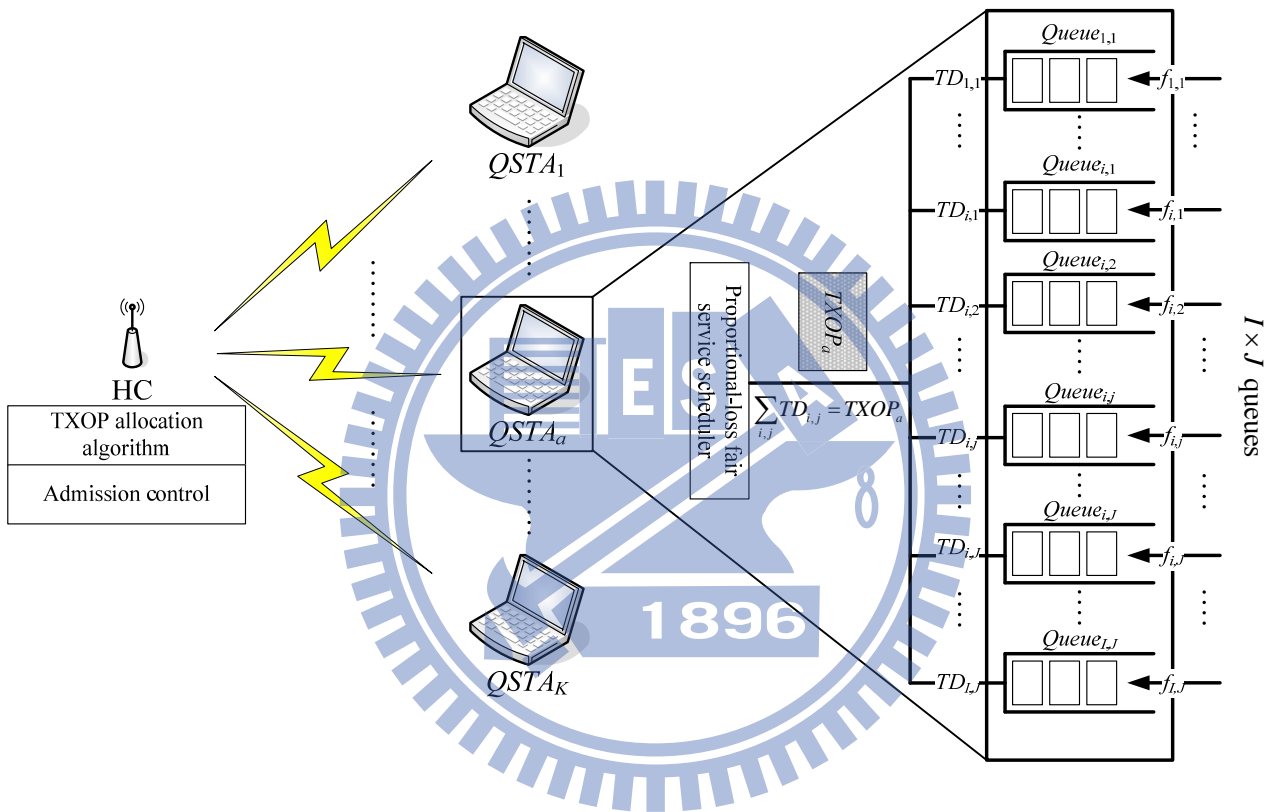


Fig. 3.2 The system architecture of our proposed scheme

3.2. Aggregate TXOP Allocation Algorithm

For ease of presentation, we firstly consider the case that flows are with identical packet loss probability requirement and then, generalize the results to the case that flows are with different packet loss probability requirement.

● Flows with identical packet loss probability requirements

It is assumed that flows requesting different delay bounds but identical packet loss probabilities. Without loss of generality, assume that the packet loss probability requested by all flows is P_1 . As a result, we have $F = F_1$. Further, for ease of description, we assume that there is at least one traffic flow with delay bound D_1 .

Consider $QSTA_a$ which has n_a flows. The n_a flows are classified into J disjoint sets $F_{1,1}$, $F_{1,2}$, ..., and $F_{1,J}$ such that a flow belongs to $F_{1,j}$ iff its delay bound is $\beta_j SI$. Let $f_{1,j}$, $1 \leq j \leq J$, with traffic arrival distribution $N(\mu_{1,j}, \sigma_{1,j}^2)$ denote the aggregated flow of all the flows in set $F_{1,j}$. The first come first serve (FCFS) service discipline was adopted for packet transmission. The effective bandwidth $c_{1,j}$ of flow $f_{1,j}$ is computed to take advantage of intra-flow multiplexing gain. The effective bandwidth $c_{1,j}$ is defined as the minimum TXOP allocated to flow $f_{1,j}$ to guarantee a packet loss probability smaller than or equal to P_1 for flow $f_{1,j}$. Since the delay bound of flow $f_{1,j}$ is $\beta_j SI$, the effective bandwidth $c_{1,j}$ can be determined with a finite-buffer queueing model where the buffer size is $\beta_j c_{1,j}$, the server transmission

capability is $c_{1,j}$, and the desired packet loss probability is P_1 . Given the traffic arrival distribution $N(\mu_{1,j}, \sigma_{1,j}^2)$, the effective bandwidth $c_{1,j}$ can be written as $c_{1,j} = \mu_{1,j} + \alpha_{1,j}\sigma_{1,j}$, where $\alpha_{1,j}$ was called the QoS parameter of flow $f_{1,j}$. Derivation of packet loss probability for a finite-buffer system is complicated. Reference [50] provided a good approximation based on the tail probability of an infinite buffer system and the loss probability of a buffer-less system, as shown in equation (8).

$$P_L(x) \approx \frac{P_L(0)}{P(X > 0)} P(X > x) \quad (8)$$

In the above equation, $P_L(x)$ represents the packet loss probability of a finite-buffer system with buffer size x and $P(X > x)$ denotes the tail probability above level x of an infinite-buffer system. The equation for $P(X > x)$ can be found in [50]. It is pretty complicated and thus is omitted due to space limitation. The equation for $P_L(0)$ can be given by

$$P_L(0) = Q(\alpha_{1,j}) + \left[\frac{\sigma_{1,j}}{\mu_{1,j}\sqrt{2\pi}} e^{-(\alpha_{1,j}^2/2)} - \left(1 + \frac{\alpha_{1,j}\sigma_{1,j}}{\mu_{1,j}} \right) Q(\alpha_{1,j}) \right] \quad (9)$$

where $Q(\alpha_{1,j}) = \int_{\alpha_{1,j}}^{\infty} (1/\sqrt{2\pi}) e^{-(x^2/2)} dx$. Having $P(X > x)$, $P(X > 0)$ and $P_L(0)$, one can obtain the (approximate) packet loss probability of a finite-buffer system with server transmission capability $c_{1,j}$ and buffer size $\beta_j c_{1,j}$ as

$$P_L(\beta_j c_{1,j}) \approx \frac{\sigma_{1,j}}{\mu_{1,j}\sqrt{2\pi}} e^{-(\alpha_{1,j}\beta_j c_{1,j}/\sigma_{1,j})} - \frac{\alpha_{1,j}\sigma_{1,j}}{\mu_{1,j}} e^{(\alpha_{1,j}^2/2) - (\alpha_{1,j}\beta_j c_{1,j}/\sigma_{1,j})} Q(\alpha_{1,j}) \quad (10)$$

Consequently, given mean $\mu_{1,j}$, variance $\sigma_{1,j}^2$, delay bound $\beta_j SI$, and the desired packet loss probability $P_1 = P_L(\beta_j c_{1,j})$, the QoS parameter $\alpha_{1,j}$ can be computed with equation (10) which in

turn can be used to derive the effective bandwidth $c_{1,j} = \mu_{1,j} + \alpha_{1,j}\sigma_{1,j}$ for flow $f_{1,j}$.

Let $L_{1,j}$ represent the nominal packet size of flow $f_{1,j}$. The average number of packets which can be transmitted in one SI, denoted by $\overline{N_{1,j}}$, can be estimated as

$$\overline{N_{1,j}} = \left\lfloor \frac{c_{1,j}}{L_{1,j}} \right\rfloor \quad (11)$$

The allocated TXOP duration for flow $f_{1,j}$ is given by

$$TD_{1,j} = \max \left\{ \frac{c_{1,j}}{R_a} + \overline{N_{1,j}} \times O, \frac{L_{\max}}{R_a} + O \right\} \quad (12)$$

where R_a represents the feasible physical transmission rate of $QSTA_a$.

As mentioned before, using buffer to store packets achieves intra-flow multiplexing gain. To further achieve inter-flow multiplexing gain, an equivalent flow of delay bound D_1 , denoted by $\hat{f}_{1,j}$, is defined for flow $f_{1,j}$, $1 \leq j \leq J$. Let $N(\hat{\mu}_{1,j}, \hat{\sigma}_{1,j}^2)$ be the traffic arrival distribution of $\hat{f}_{1,j}$. We have $\hat{f}_{1,1} = f_{1,1}$. The equivalent flow $\hat{f}_{1,j}$ for $2 \leq j \leq J$ is obtained by letting its mean and effective bandwidth equal to those of flow $f_{1,j}$, i.e., $\hat{\mu}_{1,j} = \mu_{1,j}$ and $\hat{\alpha}_{1,j}\hat{\sigma}_{1,j} = \alpha_{1,j}\sigma_{1,j}$, where $\hat{\alpha}_{1,j}$ is the QoS parameter of the equivalent flow. Since the delay bound of the equivalent flow $\hat{f}_{1,j}$ is equal to $D_1 = SI$, a packet of $\hat{f}_{1,j}$ which arrives in the n^{th} SI will violate its delay bound and be dropped if it is not served in the $(n+1)^{\text{th}}$ SI. As a consequence, the effective bandwidth for $\hat{f}_{1,j}$ can be derived based on a buffer-less system. That is, the QoS parameter $\hat{\alpha}_{1,j}$ can be computed according to equation (9) for $P_L(0) = P_1$. Note that $\hat{\alpha}_{1,j}$ can be well approximated by $Q^{-1}(P_1)$ [51]. With the approximation, we have $\hat{\sigma}_{1,j} = \alpha_{1,j}\sigma_{1,j}/Q^{-1}(P_1)$. After obtaining the equivalent

flows $\hat{f}_{1,j}$, $1 \leq j \leq J$, one can determine the aggregate equivalent flow \hat{f}_1 . Let $N(\hat{\mu}_1, \hat{\sigma}_1^2)$ denote the distribution of traffic arrival in one SI for the aggregate equivalent flow \hat{f}_1 . Since sum of independent Gaussian random variables remains Gaussian, we have $\hat{\mu}_1 = \mu_{1,1} + \sum_{j=2}^J \hat{\mu}_{1,j}$ and $\hat{\sigma}_1^2 = \sigma_{1,1}^2 + \sum_{j=2}^J \hat{\sigma}_{1,j}^2$. Again, given $\hat{\mu}_1$ and $\hat{\sigma}_1^2$, the QoS parameter $\hat{\alpha}_1$ of flow \hat{f}_1 can be derived according to equation (9) for $P_L(0) = P_1$. Having $\hat{\alpha}_1$, one can compute the effective bandwidth \hat{c}_1 for flow \hat{f}_1 . The TXOP duration allocated to $QSTA_a$ is then determined as follows

$$TXOP_a = \max \left\{ \frac{\hat{c}_1}{R_a} + \bar{N}_1 \times O + SIFS + t_{POLL}, n_a \times \left(\frac{L_{\max}}{R_a} + O \right) \right\} \quad (13)$$

where

$$\hat{c}_1 = \hat{\mu}_1 + \hat{\alpha}_1 \hat{\sigma}_1 \quad (14)$$

$$\bar{N}_1 = \left\lceil \frac{\hat{c}_1}{L_1} \right\rceil \quad (15)$$

In equation (15), \bar{L}_1 denotes the weighted average nominal packet size of all the flows in F_1 , and is calculated by

$$\bar{L}_1 = \frac{\sum_{j=1}^J \bar{N}_{1,j} \times L_{1,j}}{\sum_{j=1}^J \bar{N}_{1,j}}. \quad (16)$$

The criterion shown in equation (4) was used for admission control.

Clearly, assuming all traffic flows have identical packet loss probabilities is a big constraint of the above scheme. A straightforward solution to handle flows with different packet loss

probabilities is to assume that all flows have the most stringent requirement. Unfortunately, such a solution increases the effective bandwidths of flows which allow packet loss probabilities greater than the smallest one. Another possible solution is to compute separately the effective bandwidth \hat{c}_i for aggregated equivalent flow \hat{f}_i , $1 \leq i \leq I$, and allocate $TXOP_a = \sum_{i=1}^I \hat{c}_i$. Such a solution, however, does not take advantage of inter-flow multiplexing gain. In the following sub-section, we present our proposed scheme which considers different packet loss probabilities and takes advantage of inter-flow multiplexing gain.

- **Flows with different packet loss probability requirements**

First of all, an aggregate equivalent flow, denoted by \hat{f}_i , is determined using the technique described in the last section for flows $f_{i,1}$, $f_{i,2}$, ..., and $f_{i,J}$, for all i , $1 \leq i \leq I$. Note that the packet loss probability requirement of \hat{f}_i is P_i . Let $N(\hat{\mu}_i, \hat{\sigma}_i^2)$ represent the traffic arrival distribution for flow \hat{f}_i . Define \hat{f} as the ultimate equivalent flow with traffic arrival distribution $N(\sum_{i=1}^I \hat{\mu}_i, \sum_{i=1}^I \hat{\sigma}_i^2)$. The desired packet loss probability of flow \hat{f} , denoted by $P_{ultimate}$, is given by

$$P_{ultimate} = \frac{\sum_{i=1}^I P_i \cdot \hat{\mu}_i}{\sum_{i=1}^I \hat{\mu}_i} \quad (17)$$

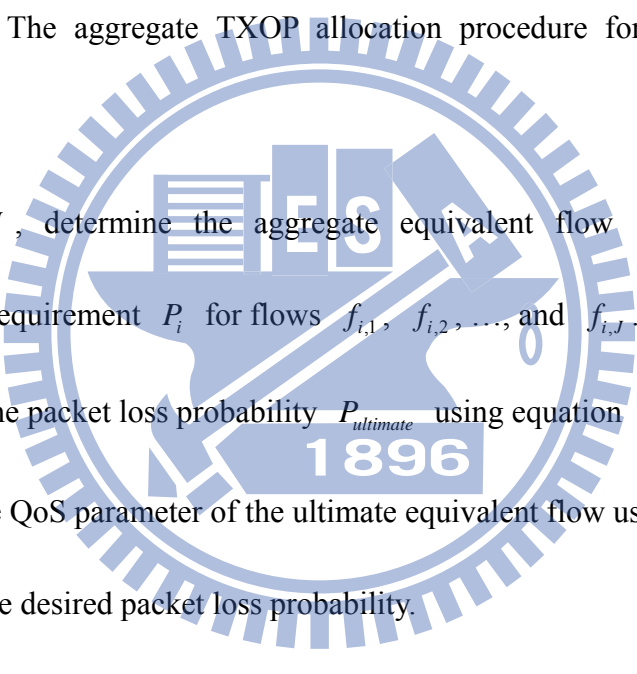
Note that the delay bounds of the aggregate equivalent flows \hat{f}_i , $1 \leq i \leq I$, and the ultimate equivalent flow \hat{f} are equal to SI . Consequently, the QoS parameter $\hat{\alpha}$ of flow \hat{f} can be computed using equation (9) with desired packet loss probability $P_{ultimate}$. The aggregate TXOP

allocated to $QSTA_a$ can be calculated using equation (13), except that the aggregate effective bandwidth and the average number of packets which can be served in one SI are obtained by

$$\hat{c} = \sum_{i=1}^I \hat{\mu}_i + \hat{\alpha} \sqrt{\sum_{i=1}^I \hat{\sigma}_i^2} \quad (18)$$

$$\bar{N} = \left\lceil \frac{\hat{c}}{\bar{L}} \right\rceil \quad (19)$$

In equation (19), \bar{L} denotes the weighted average nominal packet size of all the flows in F and is calculated by $\bar{L} = (\sum_{i=1}^I \bar{N}_i \cdot \bar{L}_i) / (\sum_{i=1}^I \bar{N}_i)$, where \bar{N}_i and \bar{L}_i can be obtained using equations (15) and (16), respectively. The aggregate TXOP allocation procedure for $QSTA_a$ is summarized below.

- 
- Step 1.** For $1 \leq i \leq I$, determine the aggregate equivalent flow \hat{f}_i with packet loss probability requirement P_i for flows $f_{i,1}$, $f_{i,2}$, ..., and $f_{i,J}$.
- Step 2.** Determine the packet loss probability $P_{ultimate}$ using equation (17).
- Step 3.** Compute the QoS parameter of the ultimate equivalent flow using equation (9) with $P_{ultimate}$ as the desired packet loss probability.
- Step 4.** Compute the aggregate transmission duration $TXOP_a$ allocated to $QSTA_a$ using equation (13) with the effective bandwidth and average number of packets served in one SI obtained from equations (18) and (19).

3.3. Proportional-loss Service Scheduler

When polled, $QSTA_a$ needs to determine how the flows attached to it share the allocated TXOP.

Let $Queue_{i,j}$ denote the queue maintained in $QSTA_a$ that is used to save packets of flow $f_{i,j}$. As shown in Fig. 3.3, $Queue_{i,j}$ is divided into β_j virtual sub-queues such that the p^{th} sub-queue, represented by $Queue_{i,j}^p$, $1 \leq p \leq \beta_j$, contains packets which can be kept for up to p SIs before violating the delay bound. How the allocated TXOP is shared is controlled by our proposed proportional-loss fair service scheduler.

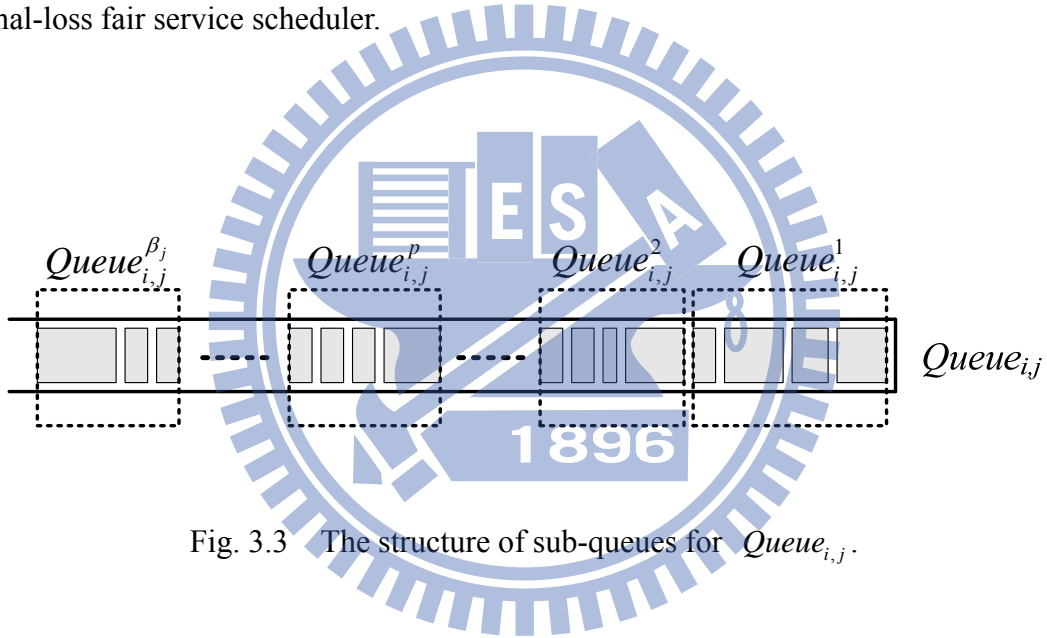


Fig. 3.3 The structure of sub-queues for $Queue_{i,j}$.

Consider the n^{th} SI. The proposed proportional-loss fair service scheduler is similar to the earliest deadline first (EDF) scheduler [53]. Let $Q_{i,j}^p[n]$, $1 \leq p \leq \beta_j$, represent the buffer occupancy in terms of transmission time for $Queue_{i,j}^p$ and $Q_{i,j}[n] = \sum_{p=1}^{\beta_j} Q_{i,j}^p[n]$. If the aggregate TXOP allocated to $QSTA_a$ satisfies $TXOP_a \geq \sum_{i,j} Q_{i,j}[n]$, then all packets in $Queue_{i,j}$ can be served and, therefore, no traffic is lost in the n^{th} SI. In this case, our proposed proportional-loss fair service

scheduler is the same as the EDF scheduler.

Assume that $TXOP_a < \sum_{i,j} Q_{i,j}[n]$. Under this assumption, there exists a minimum m such that $\sum_{i,j} \sum_{p=1}^m Q_{i,j}^p[n] > TXOP_a$. Packets with deadlines smaller than $m \cdot SI$ are served in this SI according to the EDF scheduler. Any packet which can be kept for longer than $m \cdot SI$ stays in queue. Packets in $Queue_{i,j}^m$, $1 \leq i \leq I$, $\beta_j \geq m$, are handled differently by our proposed proportional-loss fair service scheduler and the EDF scheduler. In the proposed proportional-loss fair service scheduler, which packets should stay in queue (if $m > 1$) or be dropped (if $m = 1$) is decided based on running packet loss probabilities. Once the decision is made, the service order of those packets to be transmitted is determined by the EDF scheduler.

Define $Loss[n] = \sum_{i,j} \sum_{p=1}^m Q_{i,j}^p[n] - TXOP_a$. For $Queue_{i,j}$, let $A_{i,j}[n]$ and $L_{i,j}[n]$ denote, respectively, the accumulated amount of traffic arrived and lost up to the n^{th} SI. Define $l_{i,j}[n]$ as the amount of lost traffic (if $m = 1$) or the amount of traffic with deadline $m \cdot SI$ that stays in $Queue_{i,j}$ (if $m > 1$). Also, define $TD_{i,j}[n]$ as the TXOP duration shared by $Queue_{i,j}$. It holds that $\sum_{i,j} TD_{i,j}[n] = TXOP_a$. Finally, let $P_{i,j}[n] = (L_{i,j}[n-1] + l_{i,j}[n]) / A_{i,j}[n]$. We call $P_{i,j}[n]$ the running packet loss probability for $Queue_{i,j}$ up to the n^{th} SI if $m = 1$, or a pseudo one if $m > 1$.

Our proposed proportional-loss fair service scheduler tries to minimize the total amount of packet loss while maintaining a kind of fairness in the sense that the (pseudo) running packet loss

probabilities of traffic flows are proportional to their packet loss probability requirements. To achieve the goal, we let $l_{i,j}[n]=0$ if $\beta_j < m$ or $\beta_j \geq m$ and $Q_{i,j}^m[n]=0$. For $Queue_{i,j}$ with $\beta_j \geq m$ and $Q_{i,j}^m[n]>0$, the following equations are solved for $l_{i,j}[n]$.

$$\frac{P_{i,j}[n]}{P_i} = \frac{P_{r,s}[n]}{P_r} \quad \forall (i,j),(r,s) \in U_{active} \quad (20)$$

$$Loss[n] = \sum_{(i,j) \in U_{active}} l_{i,j}[n] \quad (21)$$

In equations (20) and (21), U_{active} is a set which contains (i,j) such that $Q_{i,j}^m[n]>0$. For ease of description, we assume that every $Queue_{i,j}$ is in U_{active} if $\beta_j \geq m$. After some derivations (shown in the Appendix), we get

$$l_{i,j}[n] = \frac{1}{\sum_{(r,s) \in U_{active}} P_r \cdot A_{r,s}[n]} \left[P_i \cdot A_{i,j}[n] \cdot \left(Loss[n] + \sum_{(r,s) \neq (i,j), (r,s) \in U_{active}} L_{r,s}[n-1] \right) - L_{i,j}[n-1] \cdot \sum_{(r,s) \neq (i,j), (r,s) \in U_{active}} P_r \cdot A_{r,s}[n] \right] \quad (22)$$

If the solution satisfies $0 \leq l_{i,j}[n] \leq Q_{i,j}^m[n]$ for all $(i,j) \in U_{active}$, then a feasible solution is obtained.

The TXOP duration for $Queue_{i,j}$, i.e., $TD_{i,j}[n]$, is given by

$$TD_{i,j}[n] = \left(\sum_{p=1}^{m-1} Q_{i,j}^p[n] \right) + Q_{i,j}^m[n] - l_{i,j}[n] \quad (23)$$

However, the solution obtained by equation (22) may be infeasible, i.e., it is possible to have $l_{i,j}[n] > Q_{i,j}^m[n]$ or $l_{i,j}[n] < 0$ for some $(i,j) \in U_{active}$. If it happens, then adjustment is necessary to make the solution feasible. The adjustment is accomplished by the loss computation algorithm shown in Appendix B. Its basic idea is described below. There are four possible cases for the solution obtained by equation (22).

Case 1 $0 \leq l_{i,j}[n] \leq Q_{i,j}^m[n]$ for all $(i,j) \in U_{active}$.

If $0 \leq l_{i,j}[n] \leq Q_{i,j}^m[n]$ for all $(i,j) \in U_{active}$, then a feasible solution is found.

Case 2 $l_{i,j}[n] \geq 0$ for all $(i,j) \in U_{active}$ and $l_{r,s}[n] > Q_{r,s}^m[n]$ for some (r,s) .

In this case, let $Loss'[n] = Loss[n]$. For every (i,j) such that $l_{i,j}[n] \geq Q_{i,j}^m[n]$, assign $l_{i,j}[n] = Q_{i,j}^m[n]$, remove (i,j) from U_{active} , and set $Loss'[n] = Loss'[n] - Q_{i,j}^m[n]$. Use equation (22) again to compute $l_{i,j}[n]$ for the updated U_{active} and $Loss'[n]$. Note that, as proved in Theorem 3.1 below, the updated solution should fall in either Case 1 or Case 2. If it falls in Case 1, then a feasible solution is obtained. Otherwise, the same process is repeated. Eventually, a feasible solution will be obtained because it holds that $\sum_{i,j} Q_{i,j}^m[n] > Loss[n]$.

Theorem 3.1 Given U_{active} and $Loss[n]$. Assume that the solution shown in equation (22) satisfies $l_{i,j}[n] \geq 0$ for all $(i,j) \in U_{active}$ and $l_{r,s}[n] > Q_{r,s}^m[n]$ for some (r,s) . Let $U = U_{active} - \{(r,s)\}$ and $Loss'[n] = Loss[n] - Q_{r,s}^m[n]$. Further, let $l'_{i,j}[n]$, $(i,j) \in U$, be the solution of equations (20) and (21) for U and $Loss'[n]$. It holds that $l'_{i,j}[n] > l_{i,j}[n] > 0$.

Note that proof of all Lemmas and Theorems are provided in Appendix A. Theorem 3.1 says that if we set $l_{r,s}[n] = Q_{r,s}^m[n]$ when $l_{r,s}[n] > Q_{r,s}^m[n]$, then $l_{i,j}[n]$ has to be increased for all $(i,j) \in U$ in order to satisfy equation (20) for queues in U and equation (21). In fact, the amount $l_{r,s}[n] - Q_{r,s}^m[n]$ is proportionally shared by queues in U , i.e., it holds that $(l'_{a,b}[n] - l_{a,b}[n]) / A_{a,b}[n] P_a = (l'_{c,d}[n] - l_{c,d}[n]) / A_{c,d}[n] P_c$ for all $(a,b), (c,d) \in U$. It is worth to

point out that although Theorem 3.1 is stated for one (r, s) which satisfies $l_{r,s}[n] > Q_{r,s}^m[n]$, it actually implies the same conclusion if multiple queues satisfy the condition.

Case 3 $l_{i,j}[n] < Q_{i,j}^m[n]$ for all $(i, j) \in U_{active}$ and $l_{r,s}[n] < 0$ for some (r, s) .

In this case, we assign $l_{i,j}[n] = 0$ for every (i, j) such that $l_{i,j}[n] \leq 0$, remove (i, j) from U_{active} , and solve for new $l_{i,j}[n]$ with equation (22) for the updated U_{active} and $Loss[n]$. The updated solution will fall in either Case 1 or Case 3. This is implied by Theorem 3.2 stated below. Similarly, a feasible solution is found if the updated solution falls in Case 1. Otherwise, the same process is repeated till a feasible solution appears. The proof for Theorem 3.2 is similar to that for Theorem 3.1 and is omitted.

Theorem 3.2 Given U_{active} and $Loss[n]$. Assume that the solution shown in equation (22) satisfies $l_{i,j}[n] \leq Q_{i,j}^m[n]$ for all $(i, j) \in U_{active}$ and $l_{r,s}[n] < 0$ for some (r, s) . Let $U = U_{active} - \{(r, s)\}$ and $l'_{i,j}[n]$, $(i, j) \in U$, be the solution of equations (20) and (21) for U and $Loss[n]$. It holds that $l'_{i,j}[n] < l_{i,j}[n] < Q_{i,j}^m[n]$.

Theorem 3.2 states that if we set $l_{r,s}[n] = 0$ when $l_{r,s}[n] < 0$, then $l_{i,j}[n]$ has to be decreased for all $(i, j) \in U$ in order to satisfy equation (20) for queues in U and equation (21). Again, although we state Theorem 3.2 for one (r, s) which satisfies $l_{r,s}[n] < 0$, it implies the same conclusion if multiple queues satisfy the condition. Therefore, for Case 3, we can repeatedly set $l_{i,j}[n] = 0$ for all (i, j) such that $l_{i,j}[n] \leq 0$ and solve equations (20) and (21) for the updated

U_{active} and $Loss[n]$ until a feasible solution is found.

Case 4 $l_{r,s}[n] > Q_{r,s}^m[n]$ for some (r,s) and $l_{r',s'}[n] < 0$ for some (r',s') .

Let U be the set which contains all $(i,j) \in U_{active}$, such that $l_{i,j}[n] \geq 0$. Case 4 is further divided into two sub-cases.

Sub-case 1 $\sum_{(i,j) \in U} Q_{i,j}^m[n] < Loss[n]$

For this sub-case, define $V_1 = \{(i,j) \in U_{active} : l_{i,j}[n] \geq Q_{i,j}^m[n]\}$ and $V_2 = U_{active} - V_1$. We set $l_{i,j}[n] = Q_{i,j}^m[n]$ for all $(i,j) \in V_1$ and $Loss'[n] = Loss[n] - \sum_{(i,j) \in V_1} l_{i,j}[n]$. Then, solve equations (20) and (21) for V_2 and $Loss'[n]$. Let $l'_{i,j}[n]$, $(i,j) \in V_2$, be the solution. No further adjustment is necessary if the solution falls in Case 1. If the solution falls in Case 2, then Case 2 is performed repeatedly until a feasible solution is found. Similarly, if the solution falls in Case 3, then Case 3 will be repeatedly executed until a feasible solution is obtained. Finally, if the solution falls in Case 4, then either Sub-case 1 or Sub-case 2 is performed again.

Sub-case 2 $\sum_{(i,j) \in U} Q_{i,j}^m[n] \geq Loss[n]$

For this sub-case, let $V_1 = U$ and $V_2 = U_{active} - V_1$. Equations (20) and (21) are solved for V_1 and $Loss[n]$. If the solution falls in Case 1, then no further processing is required. Assume that the solution falls in Case 2. Let $l'_{i,j}[n]$, $(i,j) \in V_1$, be the solutions and W_1 and W_2 be two sub-sets of V_1 such that $W_1 = \{(i,j) \in V_1 : l'_{i,j}[n] < Q_{i,j}^m[n]\}$ and $W_2 = V_1 - W_1$. We set $l_{i,j}[n] = Q_{i,j}^m[n]$ for all $(i,j) \in W_2$. Let $V_2 = V_2 \cup W_1$ and $Loss'[n] = Loss[n] - \sum_{(i,j) \in W_2} l_{i,j}[n]$. Equations (20) and (21) are solved for V_2 and $Loss'[n]$. Note that this step is necessary to

achieve the equality described in equation (20) for queues in the updated V_2 . If the solution falls in Case 3, then Case 3 will be repeatedly executed until a feasible solution is obtained. Finally, if the solution falls in Case 4, then either Sub-case 1 or Sub-case 2 is performed again.

The computational complexity of the loss computation algorithm is stated in the following

Theorem 3.3.

Theorem 3.3 The loss computation algorithm takes at most $2(N-1)$ iterations to find the feasible solution, where $N = |U_{active}|$, the size of U_{active} .

After the feasible solution is found, $TD_{i,j}[n]$ can be obtained according to equation (23). If data are dropped (i.e., $m=1$), $L_{i,j}[n]$ is updated as follows

$$L_{i,j}[n] = L_{i,j}[n-1] + l_{i,j}[n] \quad (24)$$

Since the number of real-time flows attached to each QSTA is normally small, the complexity of the loss computation algorithm should be acceptable. Furthermore, because of static and periodic TXOP allocation, each QSTA has time one SI to compute the solution. Therefore, the proposed proportional-loss fair service scheduler should be feasible for real systems.

3.4. The Associated Admission Control Unit

Assume that $QSTA_a$ is negotiating with HC for a new traffic flow, i.e., the $(n_a + 1)^{th}$ flow of $QSTA_a$, that requires packet loss probability P_i and delay bound D_j . Define available bandwidth BW_{ava} as

$$BW_{ava} = SI \left(1 - \frac{T_{cp}}{T_b} \right) - \sum_{i=1}^K TXOP_i \quad (25)$$

Let θ and ρ^2 denote, respectively, the mean and variance of traffic arrival in one SI for the new traffic flow. The new flow, if admitted, will become part of flow $f_{i,j}$. As a result, we need only update the parameters related to flows $f_{i,j}$, \hat{f}_i and \hat{f} . Let $N(\mu'_{i,j}, \sigma'^2_{i,j})$, $N(\hat{\mu}'_{i,j}, \hat{\sigma}'^2_{i,j})$ and $N(\hat{\mu}'_i, \hat{\sigma}'^2_i)$ denote, respectively, the traffic arrival distributions for flows $f_{i,j}$, $\hat{f}_{i,j}$ and \hat{f}_i before the new flow is admitted. Assume that this new flow is admitted. The parameters of $f_{i,j}$ are updated as $\mu_{i,j} = \mu'_{i,j} + \theta$ and $\sigma^2_{i,j} = \sigma'^2_{i,j} + \rho^2$. Moreover, the traffic arrival distribution of the aggregate equivalent flow \hat{f}_i is updated as $N(\hat{\mu}_i, \hat{\sigma}_i^2)$, where $\hat{\mu}_i = \mu_{i,j} + \sum_{s \neq j, s=1}^J \hat{\mu}'_{i,s}$ and $\hat{\sigma}_i^2 = (\alpha_{i,j} \sigma_{i,j} / \hat{\alpha}_{i,j})^2 + \sigma_{i,1}^2 + \sum_{s \neq j, s=2}^J \hat{\sigma}_{i,s}^2$ (if $j \neq 1$) or $\hat{\sigma}_i^2 = \sigma_{i,1}^2 + \sum_{s=2}^J \hat{\sigma}_{i,s}^2$ (if $j=1$). The traffic arrival distribution of the ultimate equivalent flow \hat{f} is updated as $N(\hat{\mu}, \hat{\sigma}^2)$, where $\hat{\mu} = \hat{\mu}_i + \sum_{r \neq i, r=1}^I \hat{\mu}'_r$ and $\hat{\sigma}^2 = \hat{\sigma}_i^2 + \sum_{r \neq i, r=1}^I \hat{\sigma}_r'^2$. The ultimate packet loss probability has to be recalculated using equation (17) with the above updated parameters as input. Finally, the effective bandwidth and the required TXOP, denoted by $TXOP_a^*$, can be computed, respectively, by equations (9) and (13). Define $\Delta TXOP = TXOP_a^* - TXOP_a$. The new flow is admitted iff the following inequality is satisfied.

$$BW_{ava} \geq \Delta TXOP \quad (26)$$

If the new flow is admitted, we update BW_{ava} by $BW_{ava} = BW_{ava} - \Delta TXOP$.

Note that, if an existing flow of $QSTA_a$ is disconnected, a process similar to that shown above is conducted to obtain $\Delta TXOP = TXOP_a - TXOP_a^*$, and BW_{ava} is updated by $BW_{ava} = BW_{ava} + \Delta TXOP$.

Note that if admission or disconnection of a flow leads to change of SI, then the TXOPs for all QSTAs should be recalculated.

3.5. Simulation Results

The PHY and MAC parameters and all related information used in simulations are shown in Table

3.1. Note that the sizes of QoS-ACK and QoS-Poll in the table only include the sizes of MAC header and CRC overhead. The simulations are performed using Matlab on a PC with an Intel (R) Core (TM) 2 Quad CPU Q9550 operated at 2.83GHz with 3072 MB of RAM.

Traffic is delivered from QSTAs to AP and the contention-free period occupies the whole SI. We investigate three types of QSTA in the simulations. Each type of QSTA is assumed to be attached with two real-time traffic flows. Real traffic traces, developed by [54], are used for Type I and Type II QSTAs in our simulations. A Type III QSTA is attached with two flows, one with constant packet size and the other with variable packet size. The arrival processes are assumed to be Poisson. For flows which generate variable-size packets, the packet size varies according to exponential distribution. The length of each traffic flow lasts for one hour. The detailed information of traffic flows, including QoS

requirements and traffic parameters, are described in Table 3.3. For each flow, the mean μ and the variance σ^2 of traffic arrivals in one SI can be calculated from the mean data rate ρ and the variance of frame size v^2 provided in the trace file or derived using the technique described in Chapter 3.1. The calculated μ and σ^2 of each flow are shown in the last two rows of Table 3.2. Note that Type III QSTA is included to study the effect of aggregating flows with identical QoS requirements

Table 3.1 Related parameters used in simulations.

PHY and MAC parameters	
SIFS	10 us
MAC Header size	32 bytes
CRC size	4 bytes
QoS-ACK frame size	16 bytes
QoS CF-Poll frame size	36 bytes
PLCP Header Length	4 bytes
PLCP Preamble length	20 bytes
PHY rate(R)	11 Mbps
Minimum PHY rate (R_{min})	2 Mbps
Transmission time for different header and per-packet overhead	
PLCP Preamble and Header (t_{PLCP})	96 μ s
Data MAC Header (t_{HDR})	23.2727 μ s
Data CRC (t_{CRC})	2.90909 μ s
ACK frame (t_{ACK})	107.63636 μ s
QoS-CFPoll (t_{POLL})	122.1818 μ s
Per-packet overhead (O)	249.81818 μ s

Table 3.2 TSPECs of traffic flows attached to Type I, Type II and Type III QSTAs.

Type of QSTA	Type I		Type II		Type III	
Attached Traffic Model	Jurassic Park I	Lecture Camera	Mr. Bean	Office Camera	Poisson (Constant)	Poisson/EXP (Variable)
Packet Loss Rate Requirement (P_L)	0.01	0.001	0.01	0.001	0.01	0.01
Maximum Service Interval (SI_{max})	80(ms)	160(ms)	80(ms)	160(ms)	80(ms)	80(ms)
Mean Data Rate (ρ)	268k(bps)	210k(bps)	184k(bps)	112k(bps)	500k(bps)	500k(bps)
Nominal MSDU size (L)	1339 (bytes)	1048 (bytes)	920(bytes)	558(bytes)	1000 (bytes)	1000 (bytes)
Variance of Frame Size (ν^2)	1273237	828990	801216	1604797	1000000	1000000
Frame inter-arrival time		40 (ms)			Exponential($L/(\rho \cdot SI)$)	
Scheduled Service Interval (SI)				80 (ms)		
Calculated Mean per SI (μ)	2680 (bytes)	2100 (bytes)	1840 (bytes)	1120 (bytes)	4000 (bytes)	4000 (bytes)
Calculated Variance per SI (σ^2)	2546474	1657980	1602432	3209594	5000000	10000000

In Table 3.3, we compare packet loss probabilities after all data are delivered. Since there is only one trace for each video, we conducted simulations with 1,000 different starting positions to collect the 99% confidence intervals. The symbol $a \pm b$ in Table 3.3 means the 99% confidence interval is given by $(a-b, a+b)$. Transmission error is also considered for our proposed scheme. The frame transmission error probability is set to be 0.5×10^{-3} . The packet loss probability considering transmission error is marked with * and shown in the last row of In Table 3.3. According to the results, our proposed scheme can meet the individual QoS requirements requested by traffic flows whether or not there is aggregation of flows with identical QoS requirements. Moreover, no matter which TXOP allocation scheme is adopted, our proposed proportional-loss fair service scheduler can achieve the goal of maintaining the ratio of actual packet loss probabilities as that of the requested values. For example, the ratios of the actual packet loss probabilities of

Jurassic Park I and Lecture Camera for the sample scheduler, the RVAC scheme, the scheme proposed by Lee and Huang (2008), our proposed scheme, and our proposed scheme with transmission error are, respectively, 0.1857:0.0186, 0.0008:0.0001, 0.0052:0.0005, 0.0099:0.0010, and 0.0100:0.0010, which are all very close to the ratio of the requested packet loss probabilities, i.e., 0.01:0.001. Another important observation is that the results of our proposed scheme are satisfactory even for a frame error rate of 0.5×10^{-3} . This implies that, to cope with transmission errors, one need only select an appropriate feasible physical transmission rate so that the probability of transmission error is sufficiently smaller than the requested packet loss probability. The average execution times of the proposed proportional-loss fair service scheduler are 0.31, 0.32, and 0.52 (ms) for Types I, II, and III QSTA, respectively. These numbers are much smaller than SI (80ms) and, therefore, the scheduler is feasible for real systems.

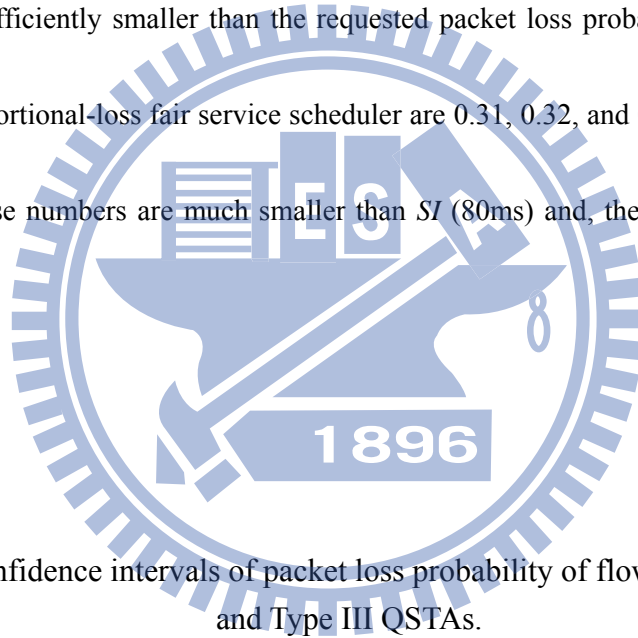


Table 3.3 The 99% confidence intervals of packet loss probability of flows attached to Type I, Type II and Type III QSTAs.

	Packet Loss Probability (P_L)					
	Type I QSTA		Type II QSTA		Type III QSTA	
	Jurassic Park I	Lecture Camera	Mr. Bean	Office Camera	Poisson (Constant)	Poisson/EXP (Variable)
Sample scheduler	$0.1857 \pm 4 \times 10^{-3}$	$0.0186 \pm 3 \times 10^{-3}$	$0.2323 \pm 3 \times 10^{-5}$	$0.0232 \pm 6 \times 10^{-6}$	$0.0446 \pm 6 \times 10^{-3}$	$0.0446 \pm 6 \times 10^{-3}$
RVAC	$0.0008 \pm 4 \times 10^{-6}$	$0.0001 \pm 4 \times 10^{-6}$	$0.0025 \pm 9 \times 10^{-6}$	$0.0003 \pm 9 \times 10^{-6}$	$0.0003 \pm 2 \times 10^{-4}$	$0.0003 \pm 2 \times 10^{-4}$
Scheme of Lee and Huang (2008)	$0.0052 \pm 2 \times 10^{-5}$	$0.0005 \pm 2 \times 10^{-5}$	$0.0032 \pm 1 \times 10^{-5}$	$0.0003 \pm 1 \times 10^{-5}$	$0.0030 \pm 8 \times 10^{-4}$	$0.0030 \pm 8 \times 10^{-4}$
Our proposed scheme	$0.0099 \pm 3 \times 10^{-5}$	$0.0010 \pm 3 \times 10^{-5}$	$0.0072 \pm 2 \times 10^{-5}$	$0.0007 \pm 2 \times 10^{-5}$	$0.0030 \pm 8 \times 10^{-4}$	$0.0030 \pm 8 \times 10^{-4}$
Our proposed scheme*	$0.0100 \pm 1 \times 10^{-4}$	$0.0010 \pm 3 \times 10^{-5}$	$0.0073 \pm 1 \times 10^{-4}$	$0.0007 \pm 2 \times 10^{-5}$	$0.0031 \pm 2 \times 10^{-3}$	$0.0031 \pm 2 \times 10^{-3}$

We also record the running packet loss probabilities of traffic flows attached to Type I QSTA for all investigated schemes. Here, the running packet loss probability for flow $f_{i,j}$ up to the n^{th} SI is given by $L_{i,j}[n]/A_{i,j}[n]$. For the sample scheduler, as shown in Fig. 3.4, the running packet loss probabilities of all simulated traffic flows are more than 10 times larger than their requested levels for most of the time. For TXOP allocation schemes which consider packet loss probability, we compare the sample paths of each traffic flow attached to Type I QSTA. The results are illustrated in Fig. 3.4 and Fig. 3.5. It can be seen that the long-term packet loss probability meets the requirement for all the investigated schemes. However, our proposed scheme is the most efficient one because it allocates the smallest TXOP durations to QSTAs. To compare the bandwidth efficiency of the investigated schemes, we list the over-allocation ratios in Table 3.4. Here, the over-allocation ratio is defined as the ratio of unused TXOP duration to the allocated TXOP duration. As one can see, our proposed scheme has the least over-allocation ratio among the investigated schemes which meet QoS requirements. In other words, compared with other static TXOP allocation algorithms, our proposed scheme reduces over-allocation ratio and hence improves bandwidth utilization without sacrificing QoS guarantee.

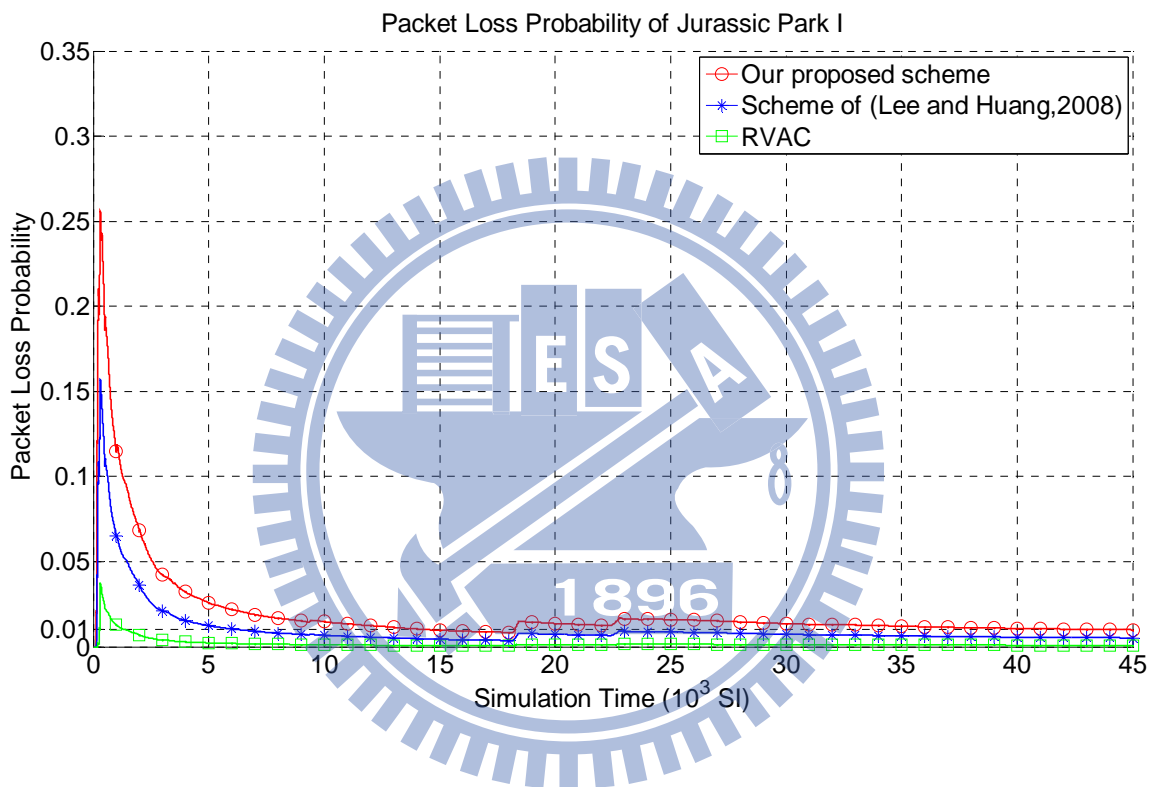


Fig. 3.4 Running packet loss probabilities of Jurassic Park I attached to Type I QSTA.

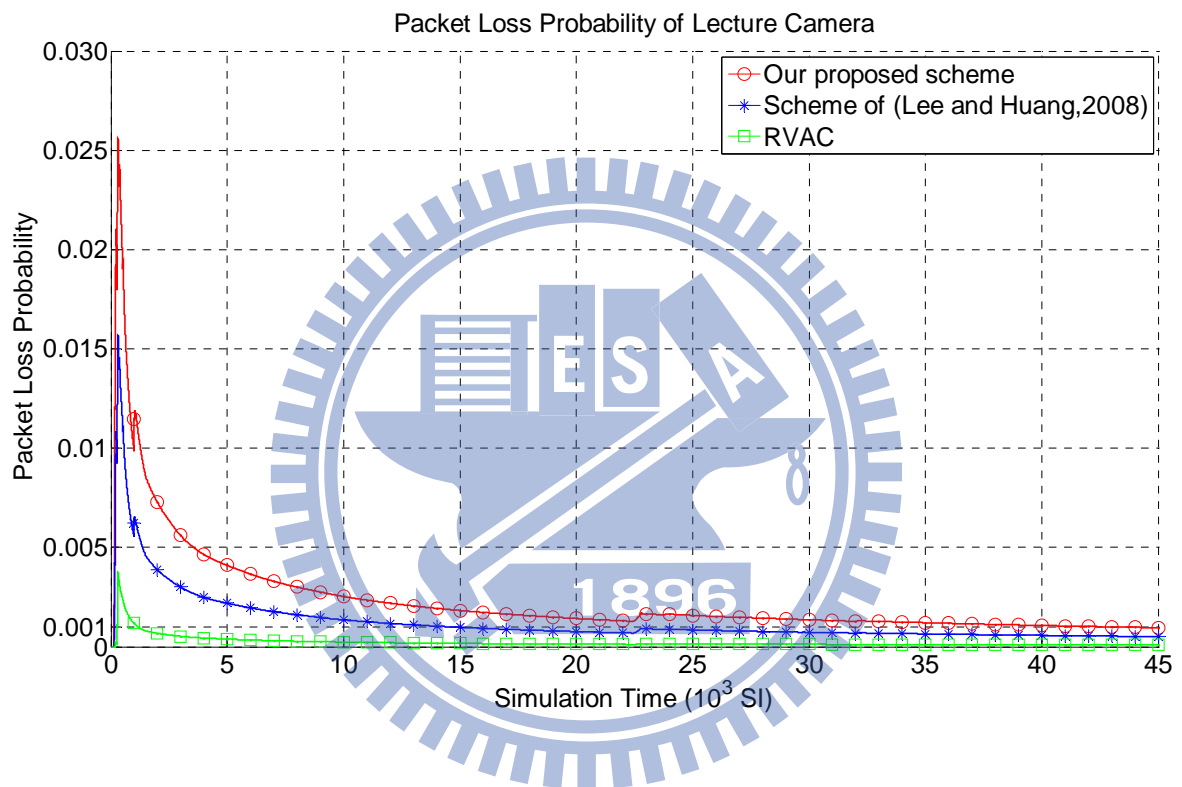


Fig. 3.5 Running packet loss probabilities of Lecture Camera attached to Type I QSTA.

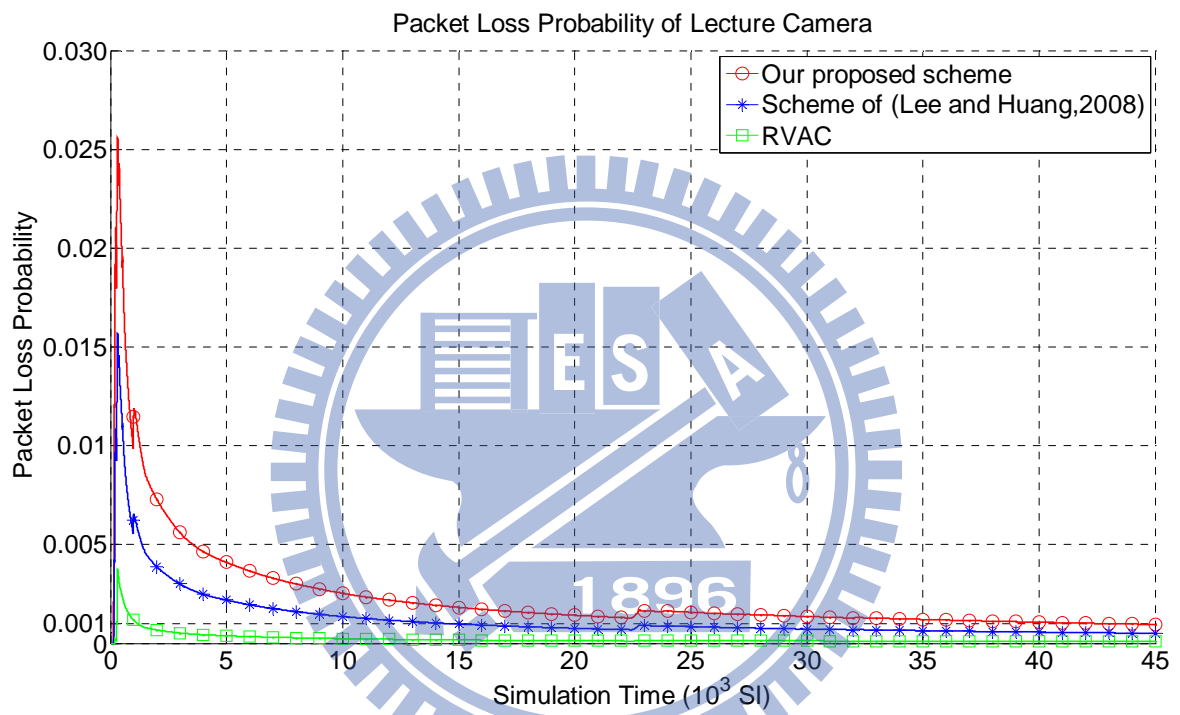


Fig. 3.6 Running packet loss probabilities of Lecture Camera attached to Type I QSTA.

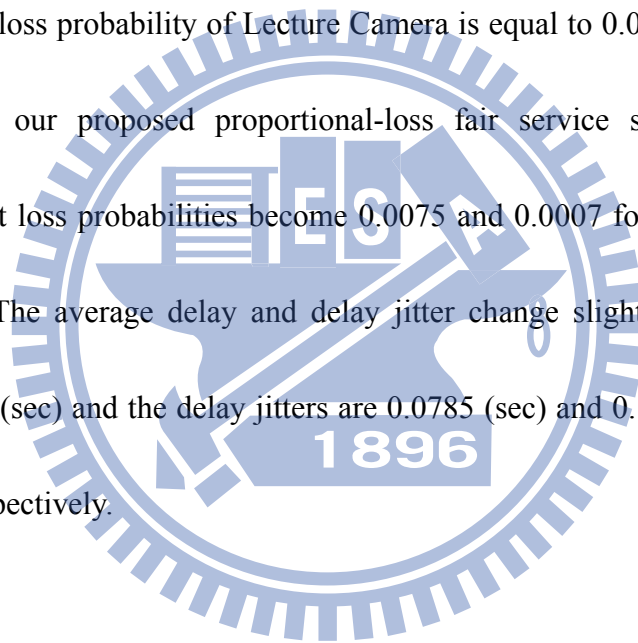
Table 3.4 Over-allocation Ratio of Type I, Type II and Type III QSTAs.

	Over-allocation Ratio		
	Type I QSTA	Type II QSTA	Type III QSTA
Sample scheduler	11.58%	15.51%	4.56%
RVAC	52.46%	52.11%	24.75%
Scheme of Lee and Huang (2008)	45.64%	48.49%	16.54%
Our proposed scheme	41.52%	44.87%	16.54%
Our proposed scheme*	41.50%	44.86%	16.03%

Fig. 3.7 compares the admissible regions of the investigated TXOP allocation schemes. For a particular scheme, the system can accommodate x Type I QSTAs and y Type II QSTAs with QoS guarantee if (x, y) falls in the triangle formed by the x -axis, y -axis, and the curve labeled for the scheme. Our proposed scheme allows 8% and 18% more QSTAs to be admitted than the scheme proposed by Lee and Huang (2008) and RVAC, respectively.

In the second part of simulations, the circular round robin is adopted as the polling scheme so that all QSTAs are treated equally. In other words, the polling order in the i^{th} SI is QSTA $i \pmod{10}$, QSTA $i+1 \pmod{10}$..., and QSTA $i+9 \pmod{10}$. As a consequence, it suffices to consider the performance of one specific QSTA. The results are shown in Table 3.5. Note that, being a dynamic scheme, the PRO-HCCA has to calculate TXOP allocations at the beginning of each SI which is an overhead to the HC. According to our simulation results, the PRO-HCCA achieves smaller average transmission delay than our proposed scheme because it allocates TXOPs to QSTAs dynamically based on the queue status. However, compared with PRO-HCCA, our proposed scheme has smaller delay

jitter, which is defined as the difference between maximum and minimum delays in this chapter. The reason is that the TXOP duration allocated by our proposed scheme is a constant which equals 7.6 ms while that allocated by PRO-HCCA is dynamic and can be larger than 7.6 ms. As a result, the maximum delay of PRO-HCCA is larger than that of our proposed scheme, which implies the delay jitter of our proposed scheme is smaller because the minimum delays are roughly the same. Moreover, our proposed scheme guarantees packet loss probability requirements while PRO-HCCA does not. For PRO-HCCA, the packet loss probability of Lecture Camera is equal to 0.0028, which is greater than its requirement 0.001. If our proposed proportional-loss fair service scheduler is combined with PRO-HCCA, then packet loss probabilities become 0.0075 and 0.0007 for Jurassic Park I and Lecture Camera, respectively. The average delay and delay jitter change slightly. The average delays are 0.0261 (sec) and 0.0289 (sec) and the delay jitters are 0.0785 (sec) and 0.1591 (sec) for Jurassic Park I and Lecture Camera, respectively.



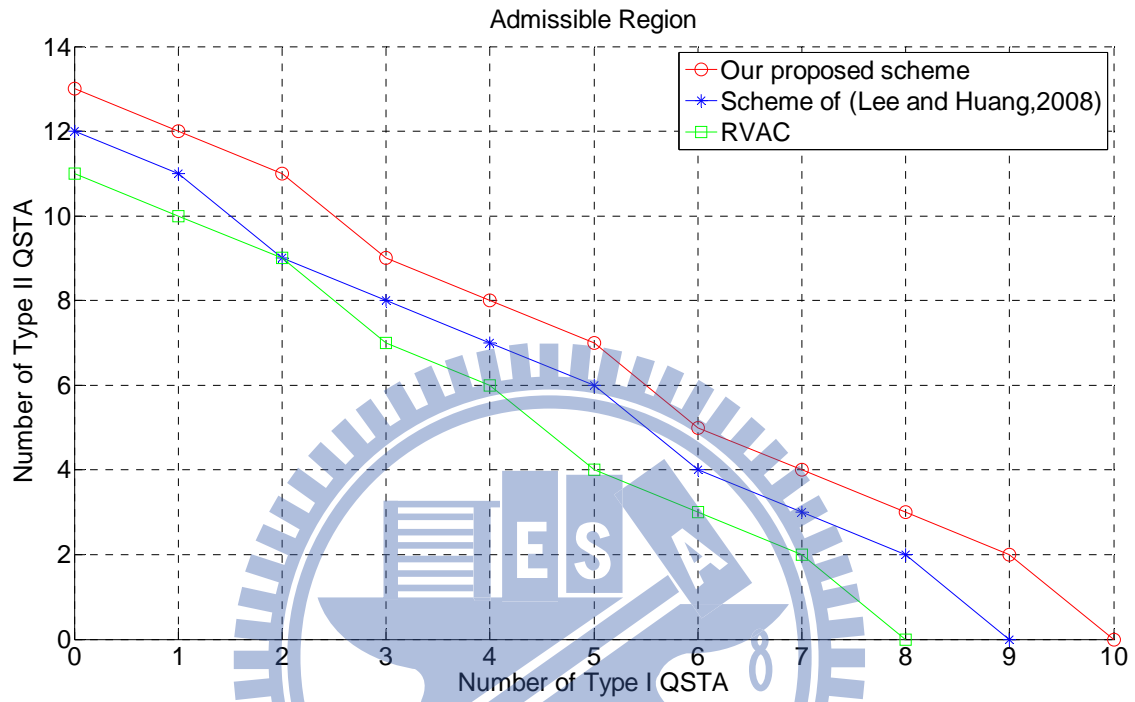


Fig. 3.7 Comparison of admissible region

Table 3.5 Performance comparison for our proposed scheme and PRO-HCCA.

	Average Transmission delay (sec)		Delay Jitter (sec)		Packet loss probability	
	Jurassic Park I	Lecture Camera	Jurassic Park I	Lecture Camera	Jurassic Park I	Lecture Camera
PRO-HCCA	0.0262	0.0287	0.0786	0.1589	0.0046	0.0028
Our proposed scheme	0.0274	0.0327	0.0757	0.1562	0.0099	0.0010

Chapter 4

Resource Allocation for Real-Time and Non-Real-Time Traffic in OFDMA-Based Systems

In this chapter, we present a resource allocation algorithm for OFDMA-based systems which handles both real-time and non-real-time traffic. For real-time traffic, the QoS requirements are specified with delay bound and loss probability. The resource allocation problem is formulated as one which maximizes system throughput subject to the constraint that the bandwidth allocated to a flow is no less than its minimum requested bandwidth, a value computed based on loss probability requirement and running loss probability. A user-level proportional-loss scheduler is adopted to determine the resource share for flows attached to the same subscriber station (SS). In case the available resource is not sufficient to provide every flow its minimum requested bandwidth, we maximize the amount of real-time traffic transmitted subject to the constraint that the bandwidth

allocated to an SS is no greater than the sum of minimum requested bandwidths of all flows attached to it. Moreover, a pre-processor is added to maximize the number of real-time flows attached to each SS that meet their QoS requirements. We show that, in any frame, the proposed proportional-loss scheduler guarantees QoS if there is any scheduler which guarantees QoS. Simulation results reveal that our proposed algorithm performs better than previous works.

4.1. System Model

The system model investigated in this chapter is the same as that presented in Chapter 2.2 and thus is not repeated here.

4.2. The Proposed Scheme

In this chapter, we present a resource allocation scheme which considers both delay bound and loss probability requirements requested by real-time traffic flows. As shown in Fig. 4.1, the minimum requested bandwidths of real-time flows are computed, summed for each SS, and then used together with queue occupancy as constraints in resource allocation. After the solution is obtained, a PL scheduler is adopted to determine how multiple real-time traffic flows attached to the same SS share the allocated bandwidth. In case the available resource is not sufficient to provide each flow its minimum requested bandwidth, a pre-processor is required to maximize the number of real-time flows attached to each SS that meet their QoS requirements. We describe calculation of minimum requested bandwidth, resource allocation, PL scheduler, and pre-processor separately

below.

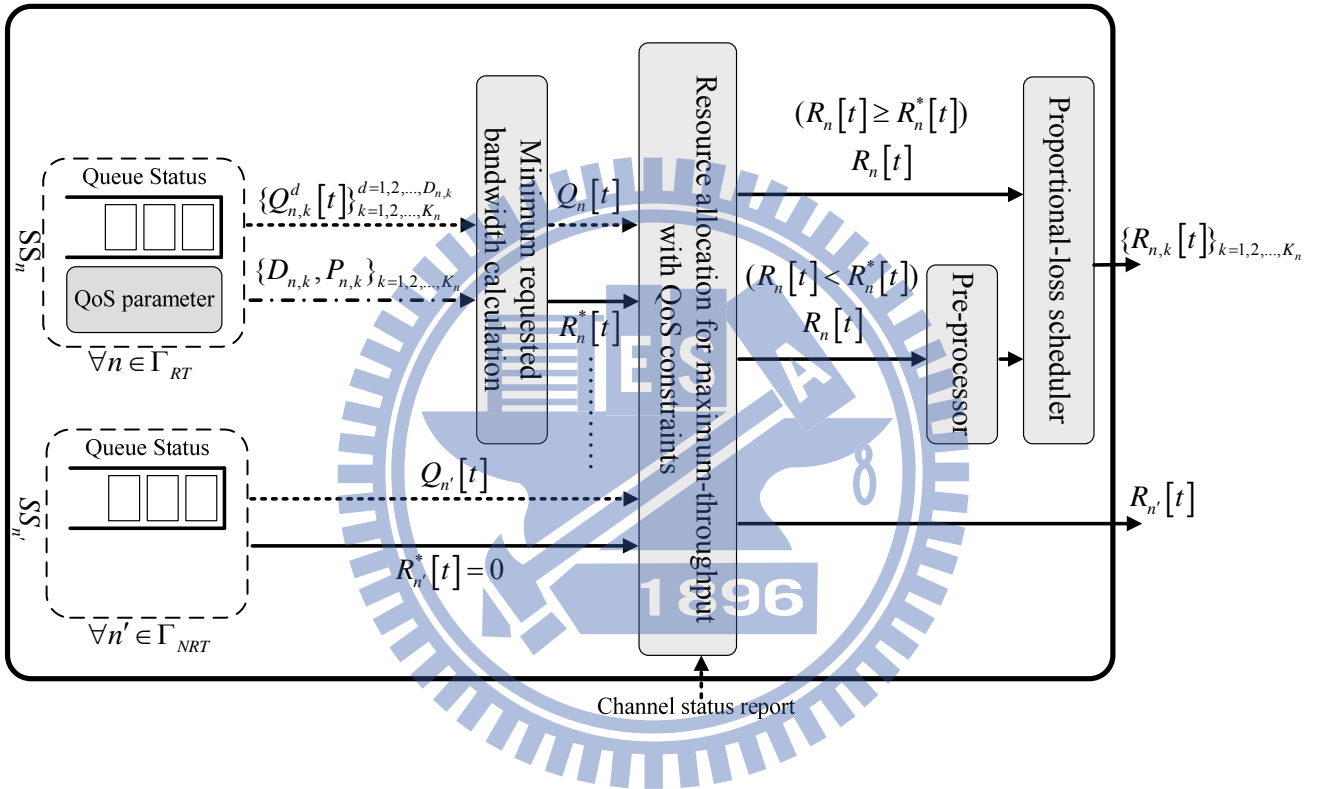


Fig. 4.1 Architecture of the proposed scheme.

- The minimum requested bandwidth

For flow $f_{n,k}$ attached to SS $n \in \Gamma_{RT}$, define $P_{n,k}[x]$, the running loss probability up to frame x , as $L_{n,k}[x]/(S_{n,k}[x]+L_{n,k}[x])$, where $S_{n,k}[x]$ and $L_{n,k}[x]$ represent, respectively, the accumulated amount of data served and lost up to the end of the x^{th} frame. Consider the t^{th} frame. Let $R_{n,k}[t]$ be the bandwidth allocated to flow $f_{n,k}$. For convenience, $R_{n,k}[t]$ is expressed in terms of the amount of data served. As a result, we have $0 \leq R_{n,k}[t] \leq Q_{n,k}[t]$. Let $(x)^+ = \max(x, 0)$. Since data are lost only due to violation of their delay bounds, we have

$$P_{n,k}[t] = \frac{L_{n,k}[t-1] + (Q_{n,k}^1[t] - R_{n,k}[t])^+}{S_{n,k}[t-1] + L_{n,k}[t-1] + \max(R_{n,k}[t], Q_{n,k}^1[t])}. \quad (27)$$

It is not hard to see that $P_{n,k}[t]$ is a continuous, strictly decreasing function of $R_{n,k}[t]$ in the range $0 \leq R_{n,k}[t] \leq Q_{n,k}[t]$. The curve of $P_{n,k}[t]$ as a function of $R_{n,k}[t]$ is illustrated in Fig. 4.2. In this figure, there are three special points on the y-axis, namely, $P_{n,k}^{\max}[t]$, $P_{n,k}^{\text{knee}}[t]$, and $P_{n,k}^{\min}[t]$, which can be obtained by substituting $R_{n,k}[t]$ with 0, $Q_{n,k}^1[t]$, and $Q_{n,k}[t]$ into equation (27), respectively. Note that if $Q_{n,k}[t] = 0$, we have $P_{n,k}[t] = P_{n,k}[t-1] = P_{n,k}^{\max}[t] = P_{n,k}^{\text{knee}}[t] = P_{n,k}^{\min}[t]$.

The minimum requested bandwidth of $f_{n,k}$, denoted by $R_{n,k}^*[t]$, is determined as follows. If $P_{n,k} \geq P_{n,k}^{\max}[t]$, then we set $R_{n,k}^*[t] = 0$ because there is no loss probability violation even if zero resource is allocated to $f_{n,k}$. Assume that $P_{n,k}^{\max}[t] > P_{n,k} > P_{n,k}^{\min}[t]$. In this case, $R_{n,k}^*[t]$ is obtained by solving $P_{n,k} = P_{n,k}[t]$, where $P_{n,k}[t]$ is described by equation (27). Finally, if $P_{n,k} \leq P_{n,k}^{\min}[t]$, then the running loss probability is still greater than or equal to the pre-defined level

$P_{n,k}$ even if all buffered data of $f_{n,k}$ are served. Therefore, we assign $R_{n,k}^*[t] = Q_{n,k}[t]$ to minimize the difference between $P_{n,k}[t]$ and $P_{n,k}$. For convenience, we use $P_{n,k}^*[t]$ to denote the running loss probability of $f_{n,k}$ at the end of the t^{th} frame if the bandwidth allocated to $f_{n,k}$ is $R_{n,k}^*[t]$. Clearly, $P_{n,k}^*[t]$ equals $P_{n,k}^{\max}[t]$ if $P_{n,k} > P_{n,k}^{\max}[t]$ or $P_{n,k}^{\min}[t]$ if $P_{n,k} < P_{n,k}^{\min}[t]$. The following lemma states that $P_{n,k}^*[t]$ is closer to $P_{n,k}$ than any other $P_{n,k}[t]$.

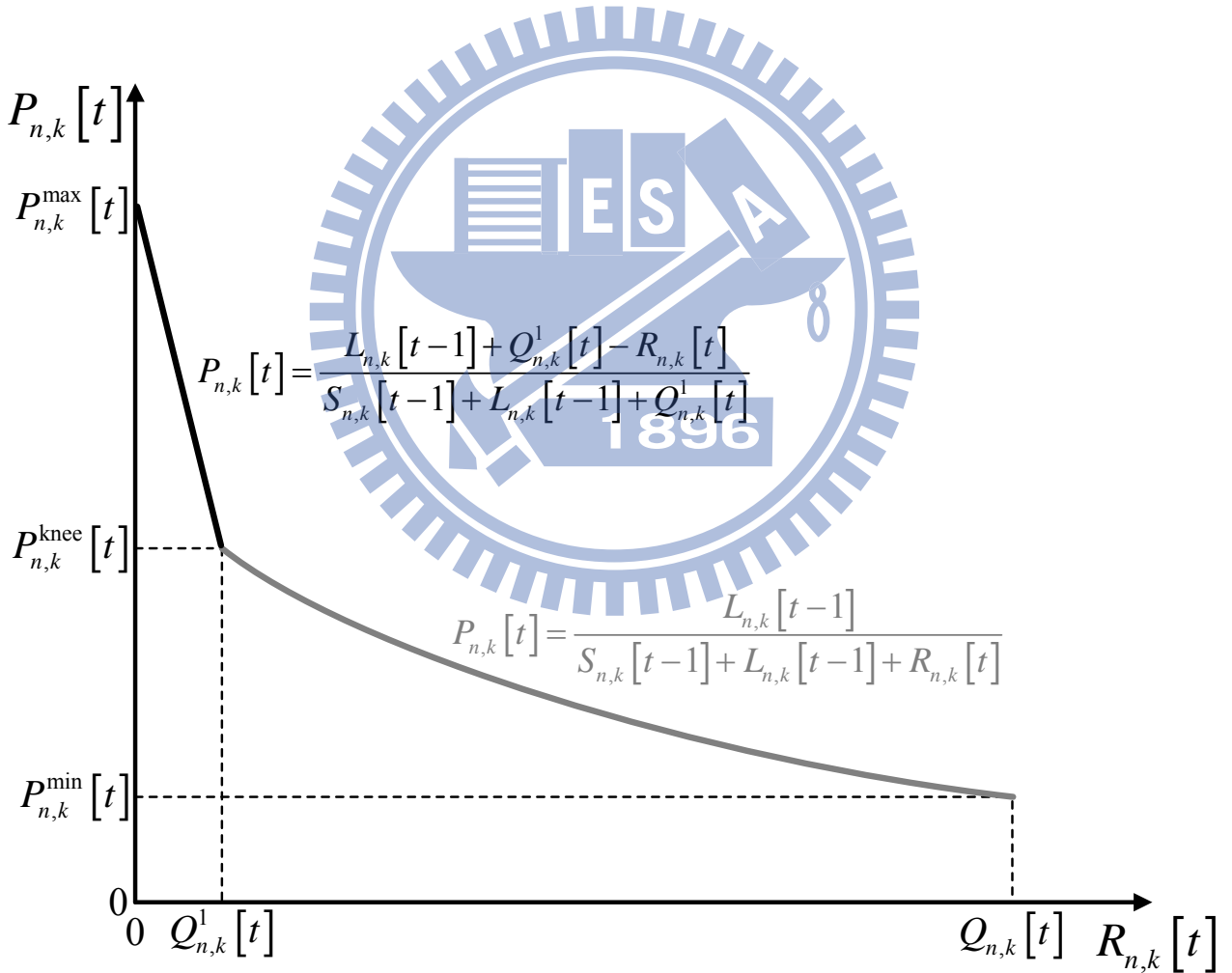


Fig. 4.2 The relationship between $P_{n,k}[t]$ and $R_{n,k}[t]$.

Lemma 4.1. It holds that $\min_{0 \leq R_{n,k}[t] \leq Q_{n,k}[t]} |P_{n,k}[t] - P_{n,k}| = |P_{n,k}^*[t] - P_{n,k}|$.

The minimum requested bandwidth for all cases is summarized in Table 4.1. Note that the actual allocated bandwidth could be different from $R_{n,k}^*[t]$. After obtaining $R_{n,k}^*[t]$ for all k , $1 \leq k \leq K_n$, one can compute $R_n^*[t]$, the aggregate minimum requested bandwidth for SS n , as $\sum_{k=1}^{K_n} R_{n,k}^*[t]$. The values of $R_n^*[t]$, $n \in \Gamma_{RT}$ are used in the resource allocation algorithm described in the next sub-section.

Table 4.1 Calculation of $R_{n,k}^*[t]$ and the resulting $P_{n,k}^*[t]$ for four conditions.

Condition	$R_{n,k}^*[t]$	$P_{n,k}^*[t]$
$P_{n,k} \geq P_{n,k}^{\max}[t]$	0	$P_{n,k}^{\max}[t]$
$P_{n,k}^{\max}[t] > P_{n,k} \geq P_{n,k}^{\text{knee}}[t]$	$R_{n,k}^*[t] = (1 - P_{n,k})(L_{n,k}[t-1] + Q_{n,k}^1[t]) - P_{n,k} \cdot S_{n,k}[t-1]$	$P_{n,k}$
$P_{n,k}^{\text{knee}}[t] > P_{n,k} > P_{n,k}^{\min}[t]$	$R_{n,k}^*[t] = \frac{L_{n,k}[t-1]}{P_{n,k}} - (S_{n,k}[t-1] + L_{n,k}[t-1])$	$P_{n,k}$
$P_{n,k} \leq P_{n,k}^{\min}[t]$	$Q_{n,k}[t]$	$P_{n,k}^{\min}[t]$

- Resource allocation for maximum-throughput with QoS constraints

As described in Problem **P1**, the proposed resource allocation algorithm maximizes system throughput while providing QoS guarantee to real-time traffic flows. In problem **P1**, we let $R_n^*[t] = 0$ for all SS $n \in \Gamma_{NRT}$. As in previous section, we use $r_{n,m}[t]$ to denote the maximum

achievable transmission rate on the m^{th} sub-channel for SS n in the t^{th} frame. The variable $x_{n,m}[t]$ represents the number of time slots allocated to SS n on the m^{th} sub-channel, in the t^{th} frame.

P1

$$\max \sum_{n \in \Gamma} \sum_{m=1}^M x_{n,m}[t] \cdot r_{n,m}[t] \quad (28)$$

subject to

$$\sum_{n \in \Gamma} x_{n,m}[t] \leq S, \quad \forall m, 1 \leq m \leq M, \quad (29)$$

$$R_n^*[t] \leq \sum_{m=1}^M x_{n,m}[t] \cdot r_{n,m}[t] \leq Q_n[t], \quad \forall n \in \Gamma, \quad (30)$$

and

$$x_{n,m}[t] \in \{0, 1, 2, \dots, S\} \quad \forall n \in \Gamma, 1 \leq m \leq M. \quad (31)$$

Problem **P1** can be solved by some integer linear programming algorithm [55]. If there is no feasible solution, meaning that the available resource is smaller than the summation of all minimum requested bandwidths, we set $x_{n,m}[t] = 0$, for all $n \in \Gamma_{NRT}$, $1 \leq m \leq M$, and solve a modified problem, called problem **P2**, which is basically the same as problem **P1** except that the constraint shown in equation (30) is replaced by $0 \leq \sum_{m=1}^M x_{n,m}[t] \cdot r_{n,m}[t] \leq R_n^*[t]$, $\forall n \in \Gamma$. Note that the solution of Problem **P2** always exists because $x_{n,m}[t] = 0$, for all $n \in \Gamma$, $1 \leq m \leq M$ is one feasible solution. Unfortunately, the complexity of integer linear programming is NP-complete [56]. One possible strategy to mitigate the computational complexity is to set $u_{n,m} = r_{n,m}[t]$ for all $n \in \Gamma$, $1 \leq m \leq M$, and conduct the matrix-based scheduling algorithm for one or two rounds. In the first

round, we only consider SSs contained in Γ_{RT} , assuming that the queue occupancy of SS n is equal to $R_n^*[t]$. The algorithm ends if the resource is exhausted in the first round. Otherwise, the second round is performed to allocate the remaining resource to all SSs, assuming the queue occupancy of SS n is equal to $Q_n[t] - R_n^*[t]$. According to the analysis provided in Chapter 2.2, the computational complexity of the modified matrix-based scheduling algorithm is $O(\max(M^2|\Gamma| + |\Gamma|^2, |\Gamma|^2 M + M^2))$.

Let $y_{n,m}[t]$ be the solution obtained either from integer linear programming or matrix-based scheduling algorithm. We have $R_n[t] = \sum_{m=1}^M y_{n,m}[t] \cdot r_{n,m}[t]$. If $R_n[t] = R_n^*[t]$, then the bandwidth allocated to the k^{th} attached flow, i.e., $R_{n,k}[t]$, is equal to $R_{n,k}^*[t]$. Assume that $R_n[t] \neq R_n^*[t]$. In this case, we need a user-level resource allocation algorithm for the attached flows to share the allocated bandwidth. In the following sub-section, we define the PL scheduler to solve this problem.

- Proportional-loss (PL) scheduler

Consider SS n and assume that it is attached with multiple real-time traffic flows. Define three disjoint sets U_Z , U_P , and U_A such that flow $f_{n,k}$ is contained in U_Z , U_P , or U_A iff $R_{n,k}[t] = 0$, $0 < R_{n,k}[t] < Q_{n,k}[t]$, or $R_{n,k}[t] = Q_{n,k}[t]$, respectively. Given $R_{n,k}[t]$, the proposed PL scheduler is a scheduler which achieves, for any $f_{n,z} \in U_Z$, $f_{n,p}, f_{n,p'} \in U_P$, and $f_{n,a} \in U_A$,

$$\frac{P_{n,z}[t]}{P_{n,z}} \leq \frac{P_{n,p}[t]}{P_{n,p}} = \frac{P_{n,p'}[t]}{P_{n,p'}} \leq \frac{P_{n,a}[t]}{P_{n,a}}, \quad (32)$$

subject to

$$R_n[t] = \sum_{k=1}^{K_n} R_{n,k}[t]. \quad (33)$$

Define $P_{n,k}[t]/P_{n,k}$ as the normalized running loss probability of $f_{n,k}$ up to frame t . The proposed PL scheduler achieves min-max optimality, as stated in Lemma 4.2. In Theorem 4.3, we show that if there exists a scheduler which guarantees the loss probability requirements, so does the PL scheduler.

Lemma 4.2. Given $R_n[t] > 0$, $S_{n,k}[t-1]$, $L_{n,k}[t-1]$, and $\{Q_{n,k}^m[t]\}_{m=1}^{D_{n,k}}$, $1 \leq k \leq K_n$, the proposed PL scheduler minimizes the maximum normalized running loss probability of all the traffic flows attached to SS n .

Theorem 4.3. Given $R_n[t] > 0$, $S_{n,k}[t-1]$, $L_{n,k}[t-1]$, and $\{Q_{n,k}^m[t]\}_{m=1}^{D_{n,k}}$, $1 \leq k \leq K_n$, if there exists a scheduler which can guarantee the loss probability requirements of all the K_n traffic flows, so can the PL scheduler.

Theorem 4.3 provides the answer why the PL scheduler is proposed as the user-level resource allocation algorithm. Define $[R_n[t], S_{n,k}[t-1], L_{n,k}[t-1], \{Q_{n,k}^m[t]\}_{m=1}^{D_{n,k}} (1 \leq k \leq K_n)]$ as the state of SS n at the beginning of the t^{th} frame. Given the state at the beginning of the first frame, the PL scheduler is preferred over other schedulers in the first frame, according to Theorem 4.3. Assume that the PL scheduler is adopted in the first frame. The state at the beginning of the second frame is determined once traffic arrivals at the beginning of the second frame is known and $R_n[2]$ is provided. Based on Theorem 4.3 again, the PL scheduler is still the preferred scheduler in the

second frame. The arguments can be applied to all frames.

In the rest of this sub-section, we present a realization of the PL scheduler. Again, consider SS n in the t^{th} frame and assume that $R_n[t]$ is given. We need to determine $R_{n,k}[t]$, $1 \leq k \leq K_n$, so that equations (32) and (33) are satisfied.

Lemma 4.4. If $R_n[t] = R_n^*[t]$, equations (32) and (33) are satisfied for $R_{n,k}[t] = R_{n,k}^*[t]$, $1 \leq k \leq K_n$.

Assume that $R_n[t] \neq R_n^*[t]$. We have the following Theorem 4.5.

Theorem 4.5. Define $\Delta R_n[t] = R_n[t] - R_n^*[t]$ and $\Delta R_{n,k}[t] = R_{n,k}[t] - R_{n,k}^*[t]$, $1 \leq k \leq K_n$. Under the PL scheduler, it holds that $\Delta R_{n,k}[t] \geq 0$ ($1 \leq k \leq K_n$) if $\Delta R_n[t] \geq 0$ or $\Delta R_{n,k}[t] \leq 0$ otherwise.

A consequence of Theorem 4.5 is that $R_{n,k}^*[t] = Q_{n,k}[t]$ implies $R_{n,k}[t] = Q_{n,k}[t]$ if $R_n[t] \geq R_n^*[t]$; and $R_{n,k}^*[t] = 0$ implies $R_{n,k}[t] = 0$ if $R_n[t] \leq R_n^*[t]$. To realize the PL scheduler, we start with $R_{n,k}[t] = R_{n,k}^*[t]$, $1 \leq k \leq K_n$. If $R_n[t] = R_n^*[t]$, then the solution is found. Adjustment is necessary if $R_n[t] \neq R_n^*[t]$. To do the adjustment, flows are classified into four sets U_Z , U_{P_1} , U_{P_2} , and U_A such that $f_{n,k}$ is in U_Z , U_{P_1} , U_{P_2} , or U_A iff $R_{n,k}^*[t] = 0$, $0 < R_{n,k}^*[t] \leq Q_{n,k}^1[t]$, $Q_{n,k}^1[t] < R_{n,k}^*[t] < Q_{n,k}[t]$, or $R_{n,k}^*[t] = Q_{n,k}[t]$, respectively. Two cases are considered separately.

Case 1 $R_n[t] > R_n^*[t]$

According to Theorem 4.5, $R_n[t] > R_n^*[t]$ implies $R_{n,k}[t] \geq R_{n,k}^*[t]$. Therefore, we should increase the value of $R_{n,k}[t]$ for $f_{n,k} \in U_{P_1} \cup U_{P_2} \cup U_Z$. Our idea is to increase $R_{n,k}[t]$

gradually, keeping equations (32) satisfied, until $R_n[t] = \sum_{k=1}^{K_n} R_{n,k}[t]$ is true. During the process of increasing $R_{n,k}[t]$, we shall either find a solution or have to move a flow from U_Z to U_{P1} , from U_{P1} to U_{P2} , or from U_{P2} to U_A . For example, assume that $f_{n,i} \in U_{P1}$ and the first event, called Event 1, we encountered is to move $f_{n,i}$ from U_{P1} to U_{P2} . For Event 1 to happen, the conditions to be met are 1) $P_{n,i}^{\text{knee}}[t]/P_{n,i} = \max_{f_{n,k} \in U_{P1}} P_{n,k}^{\text{knee}}[t]/P_{n,k}$ (no flow is moved from U_{P1} to U_{P2} earlier than Event 1), 2) $P_{n,i}^{\text{knee}}[t]/P_{n,i} \geq \max_{f_{n,k} \in U_{P2}} P_{n,k}^{\text{min}}[t]/P_{n,k}$ (no flow is moved from U_{P2} to U_A earlier than Event 1), 3) $P_{n,i}^{\text{knee}}[t]/P_{n,i} \geq \max_{f_{n,k} \in U_Z} P_{n,k}^{\text{max}}[t]/P_{n,k}$ (no flow is moved from U_Z to U_{P1} earlier than Event 1), and 4) $\sum_{f_{n,k} \in U_{P1} \cup U_{P2}} h_{n,k} \left(\left(\frac{P_{n,i}^{\text{knee}}[t]}{P_{n,i}} \right) \cdot P_{n,k}; t \right) + \sum_{f_{n,k} \in U_A} Q_{n,k}[t] < R_n[t]$ (no solution is found earlier than Event 1), where

$$h_{n,k}(x; t) = \begin{cases} \frac{1}{x} \cdot L_{n,k}[t-1] - S_{n,k}[t-1] - L_{n,k}[t-1] & \text{if } P_{n,k}^{\text{min}}[t] \leq x < P_{n,k}^{\text{knee}}[t] \\ L_{n,k}[t-1] + Q_{n,k}^1[t] - x \cdot (S_{n,k}[t-1] + L_{n,k}[t-1] + Q_{n,k}^1[t]) & \text{if } P_{n,k}^{\text{knee}}[t] \leq x \leq P_{n,k}^{\text{max}}[t] \end{cases} \quad (34)$$

Note that $h_{n,k}(x; t)$ is the inverse function of $P_{n,k}[t]$ shown in equation (27). The conditions for other events to happen can be similarly determined. After all flows are placed in the correct sets, the solution can be obtained by solving equations (32) and (33). To summarize, we repeatedly check the inequality shown in equation (35). If it holds, flow f_{n,k^*} is moved from one set to another.

$$\sum_{f_{n,k} \in U_{P1} \cup U_{P2}} h_{n,k}(p \cdot P_{n,k}; t) + \sum_{f_{n,k} \in U_A} Q_{n,k}[t] < R_n[t], \quad (35)$$

where

$$p = \max \left(\max_{f_{n,k} \in U_Z} \frac{P_{n,k}^{\text{max}}[t]}{P_{n,k}}, \max_{f_{n,k} \in U_{P1}} \frac{P_{n,k}^{\text{knee}}[t]}{P_{n,k}}, \max_{f_{n,k} \in U_{P2}} \frac{P_{n,k}^{\text{min}}[t]}{P_{n,k}} \right), \quad (36)$$

and

$$k^* = \arg \max_{f_{n,k} \in U_Z \cup U_{P1} \cup U_{P2}} \left(\max_{f_{n,k} \in U_Z} \frac{P_{n,k}^{\max} [t]}{P_{n,k}}, \max_{f_{n,k} \in U_{P1}} \frac{P_{n,k}^{\text{knee}} [t]}{P_{n,k}}, \max_{f_{n,k} \in U_{P2}} \frac{P_{n,k}^{\min} [t]}{P_{n,k}} \right). \quad (37)$$

All flows are placed in their correct sets once the inequality shown in (35) becomes false. The solution can then be obtained as follows. Set $R_{n,k} [t] = 0$ if $f_{n,k} \in U_Z$ or $Q_{n,k} [t]$ if $f_{n,k} \in U_A$. For $f_{n,k} \in U_{P1} \cup U_{P2}$, $R_{n,k} [t]$ can be obtained by $R_{n,k} [t] = h_{n,k} (P_n^F [t] \cdot P_{n,k}; t)$, where $P_n^F [t]$ represents the normalized running loss probability for any $f_{n,k} \in U_{P1} \cup U_{P2}$ at the end of the t^{th} frame and is derived in the Appendix A.

Case 2 $R_n [t] < R_n^* [t]$

Case 2 is similar to Case 1, except that we need to decrease $R_{n,k} [t]$ for $f_{n,k} \in U_{P1} \cup U_{P2} \cup U_A$.

For this case, we repeatedly check the inequality shown in (38) until it becomes false. If it is true, flow f_{n,k^*} is moved from U_A to U_{P2} , from U_{P2} to U_{P1} , or from U_{P1} to U_Z .

$$\sum_{f_{n,k} \in U_{P1} \cup U_{P2}} h_{n,k} (p \cdot P_{n,k}; t) + \sum_{f_{n,k} \in U_A} Q_{n,k} [t] > R_n [t], \quad (38)$$

where

$$p = \min \left(\min_{f_{n,k} \in U_{P1}} \frac{P_{n,k}^{\max} [t]}{P_{n,k}}, \min_{f_{n,k} \in U_{P2}} \frac{P_{n,k}^{\text{knee}} [t]}{P_{n,k}}, \min_{f_{n,k} \in U_A} \frac{P_{n,k}^{\min} [t]}{P_{n,k}} \right), \quad (39)$$

and

$$k^* = \arg \min_{f_{n,k} \in U_{P1} \cup U_{P2} \cup U_A} \left(\min_{f_{n,k} \in U_{P1}} \frac{P_{n,k}^{\max} [t]}{P_{n,k}}, \min_{f_{n,k} \in U_{P2}} \frac{P_{n,k}^{\text{knee}} [t]}{P_{n,k}}, \min_{f_{n,k} \in U_A} \frac{P_{n,k}^{\min} [t]}{P_{n,k}} \right). \quad (40)$$

After the inequality shown in (38) becomes false, the solution can be obtained as follows. Set

$R_{n,k} [t] = 0$ if $f_{n,k} \in U_Z$ or $Q_{n,k} [t]$ if $f_{n,k} \in U_A$. For $f_{n,k} \in U_{P1} \cup U_{P2}$, $R_{n,k} [t]$ can be obtained

by $R_{n,k}[t] = h_{n,k}(P_n^F[t] \cdot P_{n,k}; t)$. The pseudo code of the above realization of the PL scheduler is provided in the Appendix B.

Note that, for Case 1, the maximum number of iterations needed for the PL scheduler is $3K_n$, which happens when each flow is moved from U_Z to U_{P_1} , from U_{P_1} to U_{P_2} , and then from U_{P_2} to U_A . In each iteration, the computational complexity is $O(K_n)$. Therefore, the total computational complexity is $O(K_n^2)$. Obviously, the complexity for Case 2 is the same.

- Pre-processor

Assume that $R_n[t] < R_n^*[t]$ (i.e., Case 2 occurs) and $R_{n,k}^*[t] > 0$. In this case, flow $f_{n,k}$ will violate its loss probability requirement if the PL scheduler is adopted. As a consequence, all flows attached to SS n violate their loss probability requirements if $R_{n,k}^*[t] > 0$ for all k . This is clearly not desirable. One possible remedy is to place a pre-processor in front of the PL scheduler to maximize the number of flows which meet their loss probability requirements. Let $\Omega = U_{P_1} \cup U_{P_2} \cup \{f_{n,k} \mid f_{n,k} \in U_A, P_{n,k}^*[t] = P_{n,k}\}$. The operation of the pre-processor is as follows. 1) Select flow $f_{n,k}$ which satisfies $R_{n,k}^*[t] = \min_{f_{n,i} \in \Omega} \{R_{n,i}^*[t]\}$, 2) End the pre-processor operation if $R_{n,k}^*[t] > R_n[t]$. Otherwise, set $R_{n,k}[t] = R_{n,k}^*[t]$ and remove $f_{n,k}$ from the set it originally belongs to, 3) Update $R_n[t] = R_n[t] - R_{n,k}^*[t]$ and $\Omega = \Omega - \{f_{n,k}\}$, 4) End the pre-processor operation if $\Omega = \emptyset$. Otherwise, repeat the process. After the operation of the pre-process ends, the remaining resource is allocated to the remaining flows belonging to $U_{P_1} \cup U_{P_2} \cup U_A$ by the PL scheduler. Clearly, the computational complexity of the pre-processor is $O(K'_n \log K'_n)$, where

$K'_n = |U_{P1} \cup U_{P2} \cup \{f_{n,k} \mid f_{n,k} \in U_A, P_{n,k}^*[t] = P_{n,k}\}| \leq K_n$. As will be seen in the next section, adoption of the pre-processor can significantly increase the number of real-time flows which meet their QoS requirements.

4.3. Simulation Results

In our simulations, SSs are uniformly distributed in a circular area of radius 2Km and the BS is located at the center. Two types of real-time traffic flows are studied. Parameters of the simulation environment, AMC schemes, traffic specifications and QoS requirements of real-time flows are summarized in 0. A frame is decomposed into downlink and uplink sub-frame. We only consider downlink transmission, which is assumed to occupy 30 time slots in a frame. The other time slots are used for uplink transmission and signaling overhead. For non-real-time traffic, we assume that its queue is always non-empty. Two scenarios are investigated. In both scenarios, we assume that $|\Gamma_{NRT}| = 40$ and the minimum requested bandwidth of every non-real-time flow is zero.

In the first scenario, in addition to the 40 non-real-time flows, there are various number of SSs each attached with one Type I real-time flow. The second scenario has 13 SSs each attached with two real-time flows, one of Type I and another of Type II. Simulations are performed for 10,000 frames using Matlab on a PC with an Intel Core 2 Quad CPU operated at 2.83GHz with 3072 MB of RAM.

For the first scenario, we compare our proposed scheme with the pure maximum-throughput

algorithm, the three scheduling policies proposed in [36], and the M-LWDF scheme. To maximize system throughput, the minimum requested bandwidth of any real-time traffic flow is zero for the pure maximum-throughput algorithm. For fair comparison, we change the resource granularity from sub-channel to time slot for the three policies proposed in [36]. With such a change, their performances are better than the original versions. We label our proposed scheme by “proposed:ILP” or “proposed:Matrix” if the resource allocation problem is solved by integer linear programming or matrix-based scheduling algorithm, respectively. Both the PL scheduler and the pre-processor are adopted in Scenario 2 for all investigated schemes, except the M-LWDF scheme.

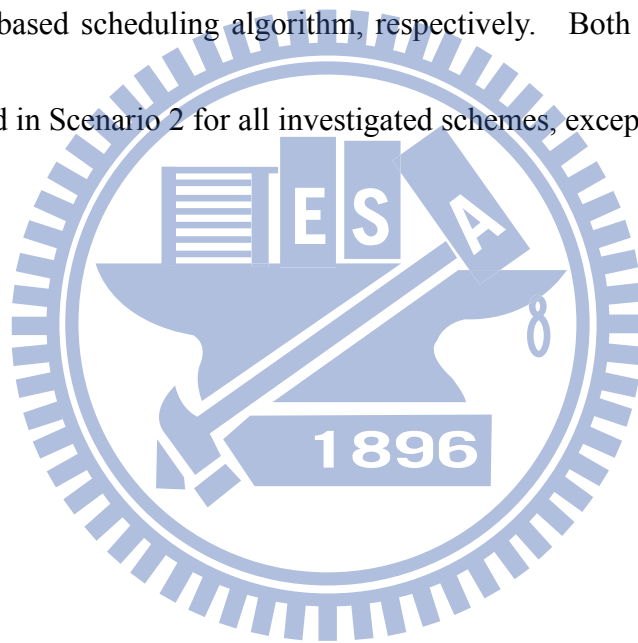


Table 4. 2 Parameters of simulation environment, traffic characteristics, QoS requirements and adopted modulation and coding scheme.

Simulation environment			
Radius of cell	2km		
User distribution	Uniform		
Bandwidth	10 MHz		
Channel model	Rayleigh fading channel		
Doppler frequency	4.6 Hz (speed: 2Km/hr)		
Pass loss exponent	4		
Frame duration	5ms		
Time slot duration	0.1ms		
Number of sub-channels	16		
Number of sub-carriers per sub-channel	64		
Traffic characteristics and QoS requirements			
Traffic Type	Type I	Type II [54]	
Content	Voice	video streaming (Star War II)	
Codec format	G.711	MPEG 4	
Mean inter-arrival time	20ms	40ms	
Mean packet size	200 bytes	267bytes	
Delay bound	80ms	160ms	
Loss probability requirement	10(%)	5, 10, 15, 20, 25(%)	
The adopted modulation and coding scheme [35].			
Mode	Modulation	Coding rate	Receiver SNR (dB)
1	QPSK	1/2	5
2	QPSK	3/4	8
3	16QAM	1/2	10.5
4	16QAM	3/4	14
5	64QAM	1/2	16
6	64QAM	2/3	18
7	64QAM	3/4	20

In Fig. 4.3 and Fig. 4.4, we compare, respectively, total system throughput and loss probability of the investigated schemes for SSs attached with Type I real-time traffic flows in the first scenario. Compared with the schemes presented in [36] for $\beta = 0$ and $\beta = 1$, our proposed scheme achieves better system throughput. The maximum improvement is about 28% (6.018Mbps versus 4.696Mbps), which occurs when $|\Gamma_{RT}| = 60$. Although the pure maximum-throughput algorithm and the scheme presented in [36] for $\beta = \infty$ have better throughput performance than our proposed scheme, their loss probabilities are higher than the specified value. In fact, a large proportion (about 80%) of real-time data is lost for the pure maximum-throughput algorithm. The reason is that there are many SSs attached with non-real-time traffic flows that are assumed to always have data for transmission. The improvement of our proposed scheme stops when $|\Gamma_{RT}| \geq 70$. The reason is that, for $|\Gamma_{RT}| \geq 70$, the average running loss probability is greater than the loss probability requirement and, therefore, the resource is allocated to users with good channel qualities by our proposed scheme and the scheme presented in [36] for $\beta = 0$ and $\beta = 1$. Compared with the M-LWDF scheme, our proposed algorithm achieves higher throughput without sacrificing QoS guarantee.

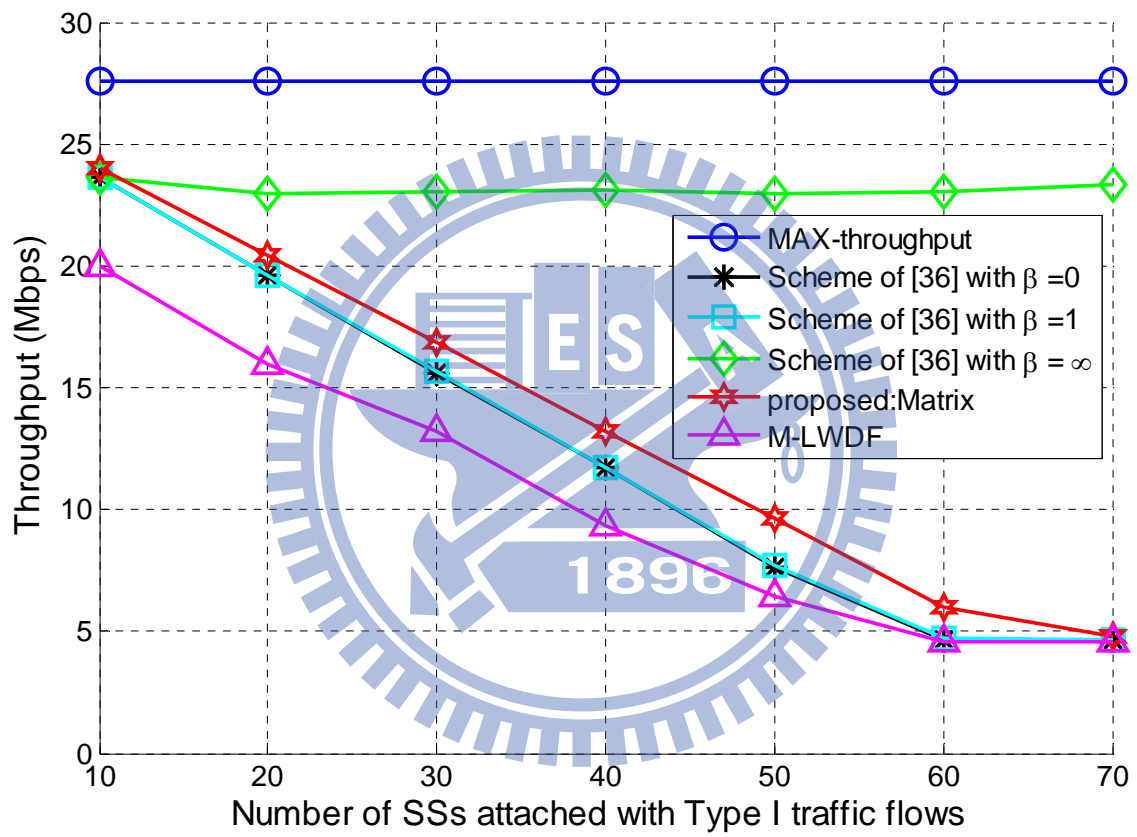


Fig. 4.3 Throughputs of various schemes in the first scenario.

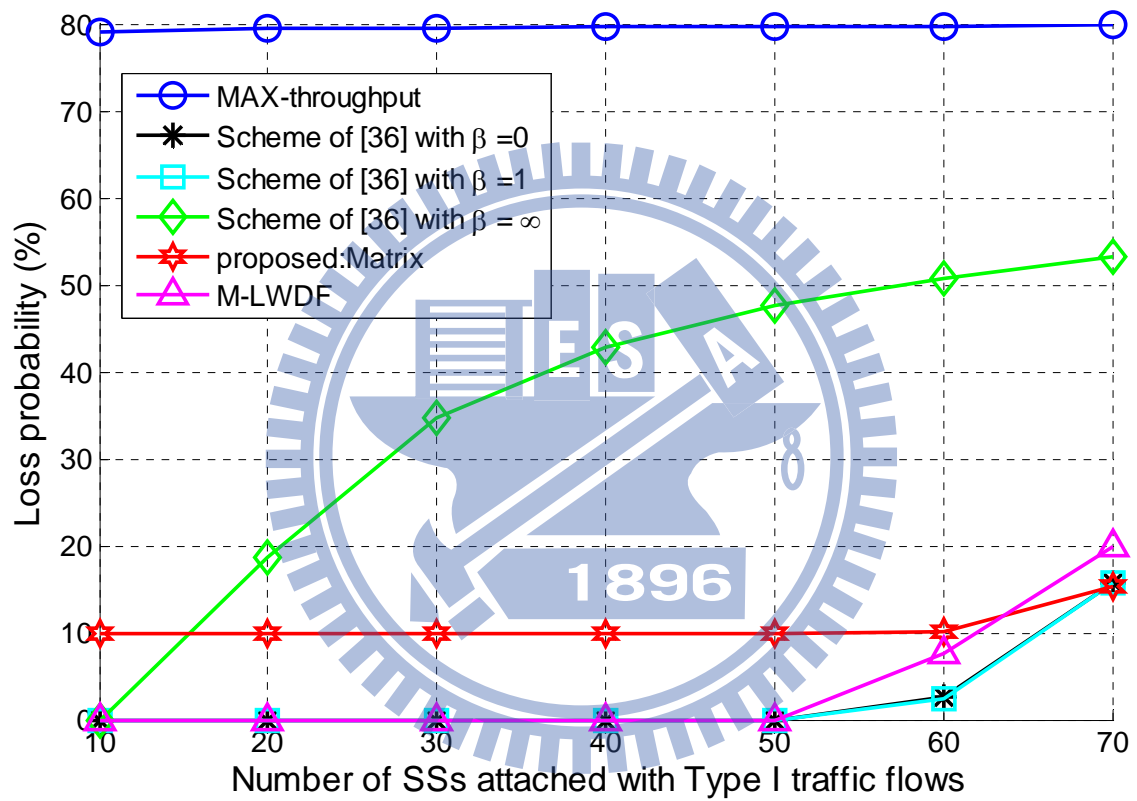


Fig. 4.4 Loss probabilities of SSs attached with real-time traffic flows in the first scenario.

In Fig. 4.5 and Fig. 4.6, we compare the performances of our proposed:ILP and proposed:Matrix schemes. Results show that the difference is not significant. For $|\Gamma_{RT}| = 30$, the execution time of the proposed:Matrix scheme is 0.9 ms, which is much smaller than 47.4 ms, the execution time of the proposed:ILP scheme.

Fig. 4.7 shows the comparison of throughput performances of the investigated schemes which guarantee QoS of all the real-time flows in the second scenario. As one can see, our proposed:Matrix scheme outperforms M-LWDF and the scheme of [13] with $\beta = 0$ or 1. The improvement increases as the loss probability requirement increases. The reason is simply because our proposed:Matrix scheme takes loss probability requirements into consideration in calculating the minimum requested bandwidth of every real-time flow. As shown in Table 4.3, both M-LWDF and the scheme of [13] (with $\beta = 0$ or 1) do not take full advantage of the tolerance of data loss feature of real-time flows. By controlling the actual loss probabilities close to requirements, our proposed scheme improves system throughput.

To study the effect of pre-processor, we conduct simulations for our proposed:Matrix scheme with and without pre-processor. The results are shown in Table 4.4. For comparison, we also include simulation results of the M-LWDF scheme. In this table, the loss probability requirement of Type II real-time flows is chosen to be 10%. As one can see, the number of Type II flows which meet their QoS requirements with pre-processor is much larger than that without pre-processor when

$|\Gamma_{RT}|$ is large. The reason is that, under the PL scheduler, the denominator of the running loss probability, i.e., $S_{n,k}[t] + L_{n,k}[t]$, is often smaller for a real-time flow with a smaller data arrival rate. As a result, a flow with a smaller data arrival rate tends to have a smaller minimum requested bandwidth and is more likely to be selected by the pre-processor. In our simulations, a flow of Type II has a smaller data arrival rate than a flow of Type I. When compared with M-LWDF, the proposed:Matrix scheme with pre-processor yields more flows which meet their QoS requirements. One interesting observation is that M-LWDF favors Type I flows. This is because Type I flows require more stringent delay bounds than Type II flows, which implies Type I flows are assigned higher priority than Type II flows when loss probability requirements are identical. We also conducted simulations for a scenario where all SSs are attached with two Type II flows. The loss probability requirement is 10% for one flow and 20% for the other. Results show that the pre-processor favors flows with 20% loss probability requirement. This is intuitively true because, under the same data arrival distribution, a flow with a larger loss probability requirement tends to have a smaller minimum requested bandwidth than one which has a smaller loss probability requirement. Owing to space limitation, we do not show these results.

We have presented in this chapter an efficient resource allocation scheme which tries to maximize system throughput while providing QoS support to real-time traffic flows. The basic idea of our proposed scheme is to calculate a dynamic minimum requested bandwidth for each traffic flow and use it as a constraint in an optimization problem which maximizes system throughput.

The minimum requested bandwidth is a function of the pre-defined loss probability and the running loss probability. In addition, a user-level PL scheduler is proposed to determine the bandwidth share for multiple real-time flows attached to the same SS. A pre-processor is adopted to maximize the number of real-time flows attached to each SS which meet their QoS requirements, when the resource is not sufficient to provide every flow its minimum requested bandwidth. Computer simulations were conducted to evaluate the performance of our proposed scheme. Results show that the running loss probabilities of traffic flows attached to the same SS are effectively controlled to be proportional to their loss probability requirements. Besides, compared with previous designs, our proposed scheme achieves higher throughput while providing QoS support. Although we present our designs for long time average of loss probabilities, the idea can be applied to other measurements such as exponentially weighted moving average. How to design a pre-processor which meets user's need is an interesting topic which can be further studied. Evaluation of the impact to user perception of satisfaction for various performance measurements is another potential further research topic.

Table 4. 3 Loss probabilities for users attached with one Type I and one Type II real-time flows.

Loss probability requirement	M-LWDF		Scheme of [36] with $\beta=0$		Scheme of [36] with $\beta=1$		proposed: Matrix	
	$P_{L,I}$	$P_{L,II}$	$P_{L,I}$	$P_{L,II}$	$P_{L,I}$	$P_{L,II}$	$P_{L,I}$	$P_{L,II}$
5%	0.0025	0.0013	0.0182	0.0091	0.0671	0.0336	0.1000	0.0502
10%	0	0.0035	0.0122	0.0122	0.0448	0.0448	0.1000	0.1000
15%	0	0.0036	0.0094	0.0141	0.0342	0.0513	0.1002	0.1505
20%	0	0.0037	0.0079	0.0158	0.0280	0.0561	0.1000	0.2000
25%	0	0.0039	0.0066	0.0165	0.0238	0.0594	0.1001	0.2503

Table 4. 4 Number of Type I and Type II flows which meet their QoS requirements in the second scenario.

Number of SSs	proposed: Matrix		proposed: Matrix without pre-processor		M-LWDF	
	Type I	Type II	Type I	Type II	Type I	Type II
10	10	10	10	10	10	10
20	20	20	20	20	19	13
30	12	30	12	12	28	14
40	16	40	16	16	30	16
50	20	50	20	20	32	20

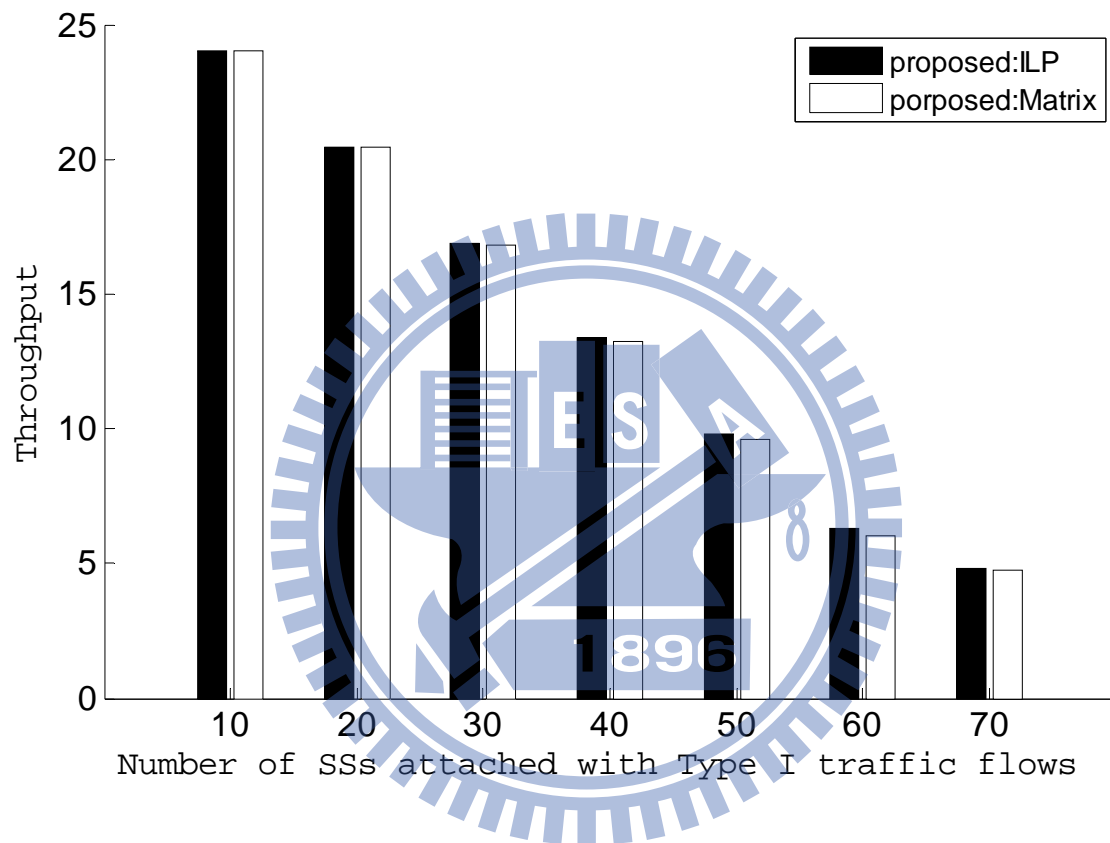


Fig. 4.5 Throughput comparison between proposed:ILP and proposed:Matrix schemes.

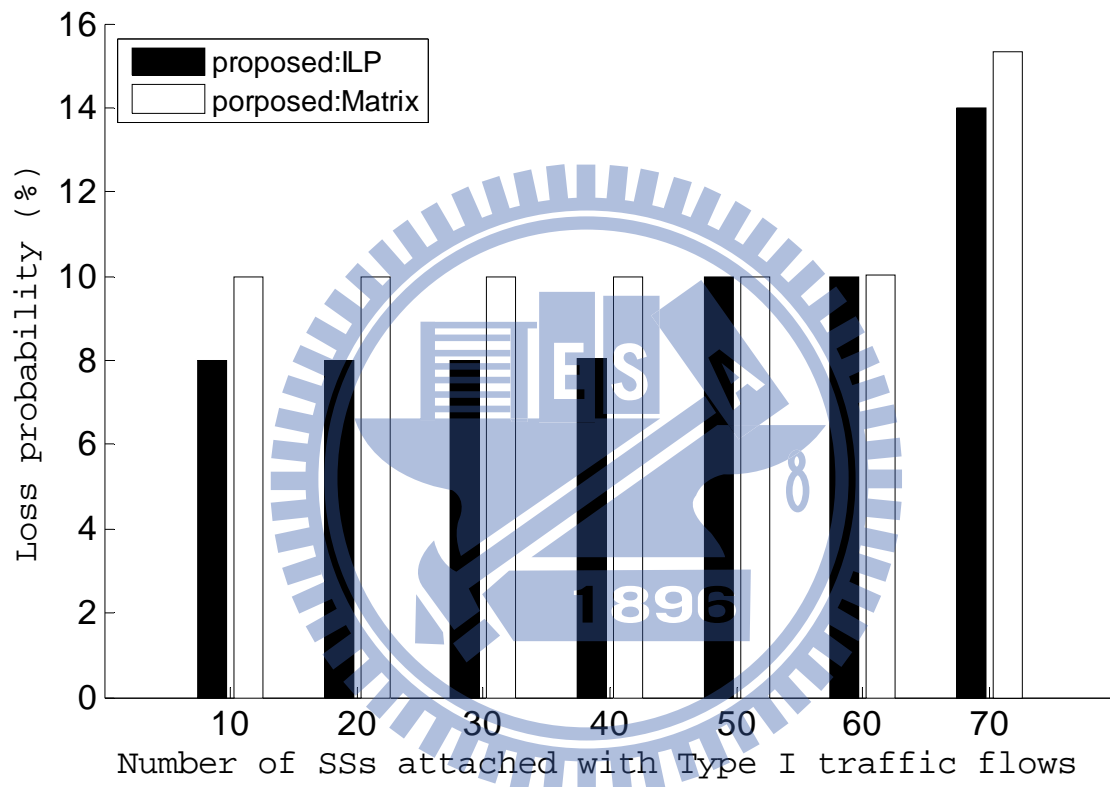


Fig. 4.6 Loss probability comparison between proposed:ILP and proposed:Matrix schemes.

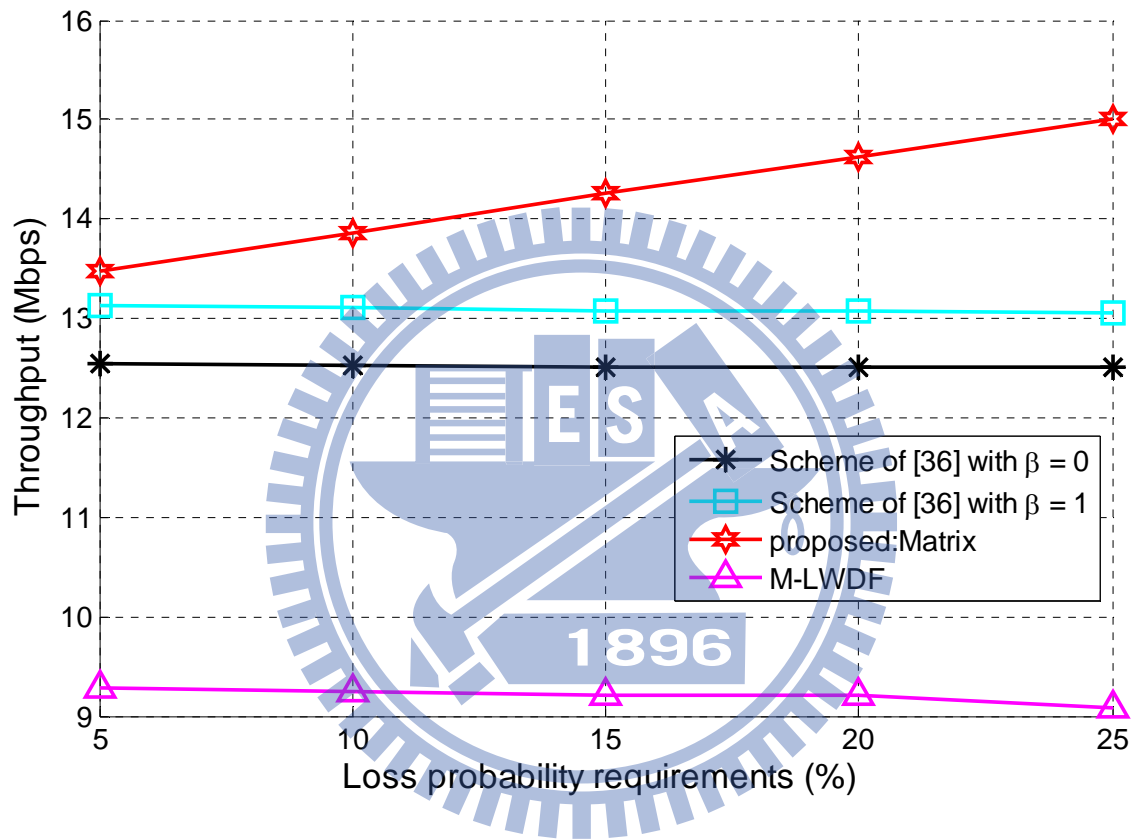


Fig. 4.7 Throughputs of various schemes in the second scenario.

Chapter 5

Optimal Queue Management

Algorithm for Real-Time Traffic

As real-time applications are proliferating rapidly, QoS guarantee for traffic flows becomes an important issue. A generalized quality of service (G-QoS) scheme coupled with the earliest deadline first (EDF) service discipline was proposed to support multiple delay bounds and cell loss probabilities in ATM networks. The G-QoS scheme, however, is only suitable for ATM networks which transport fixed-length packets. In this chapter we study a multiplexing system which handles variable-length packets. A proportional loss (PL) queue management algorithm is proposed for packet discarding, which combined with the work-conserving EDF service discipline, can provide QoS guarantee for real-time traffic flows with different delay bound and loss probability requirements. We show that the proposed PL queue management algorithm is optimal because it minimizes the effective bandwidth among all stable and generalized space-conserving schemes. The PL queue management algorithm is presented for fluid-flow models. Two packet-based

algorithms are investigated for real packet switched networks. One of the two algorithms is a direct extension of the G-QoS scheme and the other is derived from the proposed fluid-flow based PL queue management algorithm. Simulation results show that the scheme derived from our proposed PL queue management algorithm performs better than the one directly extended from the G-QoS scheme.

5.1. System Model

As illustrated in Fig 5.1, the system investigated in this paper is a multiplexer handling variable-length packets. Assume that there are K traffic flows, namely, $f_1, f_2, \dots,$ and f_K . In the investigated multiplexer, each traffic flow is allocated with a separate queue, denoted by $Queue_1, Queue_2, \dots,$ and $Queue_K$. Time is divided into slots of same duration T . In each time slot, the service capability of the multiplexer for each flow is identical and equal to C . The service scheduler arranges data of each flow for service according to the work-conserving EDF. It is assumed that data always arrives in the beginning of each time slot. Upon data arrivals, the queue management algorithm will decide if it is schedulable. If yes, no further action will be taken. Otherwise, some data are discarded so that the remaining data can be transmitted before their own deadlines.

QoS are specified by delay bound and loss probability. Consider $f_k, 1 \leq k \leq K$. Its delay bound and loss probability requirement are denoted by D_k and P_k , respectively, where

$D_k = \beta_k \cdot T$ and β_k is a positive integer. Assume that data arrive in order so that $Queue_k$, $1 \leq k \leq K$, can be virtually divided into sub-queues, $Queue_k^m$, $1 \leq m \leq \beta_k$, as shown in the bottom of 0, where $Queue_k^m$ stores the data of f_k which can be kept up to m time slots without violating their delay bounds.

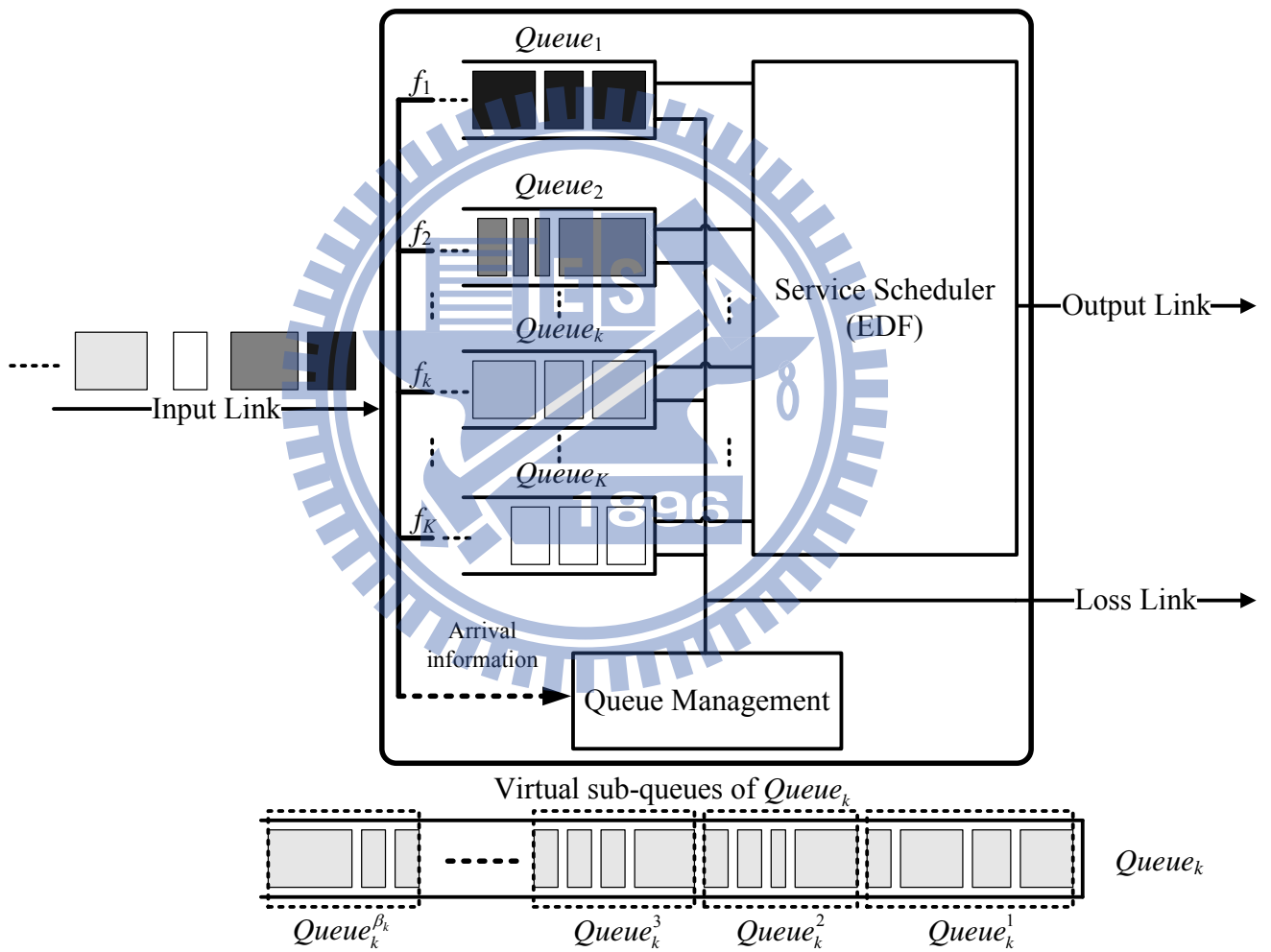


Fig. 5.1 Architecture of the investigated multiplexer system and the structure of virtual sub-queues, $Queue_k^m$, $1 \leq m \leq \beta_k$, for $Queue_k$.

5.2. The Proposed PL Queue Management Algorithm

It is assumed that packets are infinitesimally dividable, which is referred to as fluid-flow model in this dissertation. A more realistic system which manages the queues packet by packet, namely, packet-based system, will be studied in the next section. Consider f_k , $1 \leq k \leq K$, in the n^{th} time slot. Let $A_k[n]$ and $L_k[n]$ denote, respectively, the accumulated amount of data belonging to f_k arrived and lost up to the n^{th} time slot. For convenience, we set $A_k[0] = L_k[0] = 0$. Let $l_k[n]$ denote the amount of data lost from $Queue_k$ in the n^{th} time slot. The size of $Queue_k$ and $Queue_k^m$, $1 \leq m \leq \beta_k$, in the n^{th} time slot are denoted by $Q_k[n]$ and $Q_k^m[n]$, respectively. Obviously, it holds that $\sum_{m=1}^{\beta_k} Q_k^m[n] = Q_k[n]$. For convenience, we let $Q_k^m[n] = 0$ for $m > \beta_k$.

For better comprehension, we firstly investigate the case that all traffic flows request identical delay bound and then extend the results to a more general case that traffic flows request different delay bounds, which completes the description of the proposed PL queue management algorithm.

- Flows with identical delay bound requirement

Assume that $\beta_k = \beta$, $1 \leq k \leq K$, where β is a positive integer. Consider the n^{th} time slot. Define the running loss probability of f_k as $P_k[n] = L_k[n]/A_k[n]$. Upon traffic arrivals, all data buffered in the multiplexer is schedulable iff

$$\sum_{f_k \in U} Q_k[n] \leq \beta \cdot C, \quad (41)$$

where U denotes the set containing traffic flows with non-empty queues. If equation (41) holds,

we have $l_k[n]=0$ and thus $L_k[n]$ can be updated by $L_k[n]=L_k[n-1]$, $1 \leq k \leq K$. Assume that $\sum_{f_k \in U} Q_k[n] > \beta \cdot C$. Let $Loss[n]$ denote the total amount of data lost in the n^{th} time slot, which can be calculated by

$$Loss[n] = \left(\sum_{f_k \in U} Q_k[n] - \beta \cdot C \right)^+, \quad (42)$$

where $a^+ = \max(a, 0)$. It is not hard to see that a queue management scheme is generalized space-conserving if equation (42) holds for each time slot, assuming that all traffic flows request identical delay bound.

Obviously, it holds that $l_k[n]=0$ for f_k with $Q_k[n]=0$, $1 \leq k \leq K$. Given $Loss[n] > 0$, $Q_k[n]$, $A_k[n]$, $L_k[n-1]$ and P_k , the remaining task of the proposed PL queue management algorithm is to calculate $l_k[n]$ for all $f_k \in U$. Divide U into three disjoint subsets, U_c (Complete loss), U_p (Partial loss) and U_z (Zero loss) so that f_k is contained in U_c , U_p and U_z iff $l_k[n]=Q_k[n]$, $0 < l_k[n] < Q_k[n]$ and $l_k[n]=0$, respectively. For any $f_c \in U_c$, $f_p, f_{p'} \in U_p$ and $f_z \in U_z$, the proposed PL queue management algorithm achieves

$$\frac{P_c[n]}{P_c} \leq \frac{P_p[n]}{P_p} = \frac{P_{p'}[n]}{P_{p'}} \leq \frac{P_z[n]}{P_z}, \quad (43)$$

and

$$\sum_{f_k \in U} l_k[n] = Loss[n]. \quad (44)$$

To satisfy both equations (43) and (44), we found that one simple interpretation, called water-filling interpretation, can be adopted to facilitate the development of the proposed PL queue

management algorithm. As an example, assume that $U = \{f_1, f_2, f_3, f_4, f_5\}$, $\beta = 1$, all flows belonging to U have packets arrived in the n^{th} time slot and $Loss[n] > 0$. Before performing queue management, we interpret the state of each flow by the picture portrayed in Fig. 5.2 (a). In this figure, notice the following observations. 1) For each flow belonging to U , there is a rectangular vessel, which consists of two parts, solid (gray) and hollow (white) part. Note that the bottom lengths of all rectangular vessels in the x axis are assumed to be the same and thus it suffices to consider a 2-D picture. 2) Consider f_1 . The volumes of its solid and hollow part are equal to $L_1[n-1]$ and $Q_1[n]$, respectively, while the common bottom areas of them are the same and equal to $P_1 \cdot A_1[n]$. In other words, if we fill some water with amount x , $0 \leq x \leq Q_1[n]$, into this vessel, the level of water will be increased up to $(L_1[n-1] + x) / (A_1[n] \cdot P_1)$. The same idea can also be applied to all the other flows belonging to U . Based on these two observations, we found that managing queues so that both equations (43) and (44) are satisfied is just identical to fill water with amount $Loss[n]$ into the super vessel, the combination of all the vessels of flows belonging to U . The results are shown in Fig. 5.2 (b). In this figure, it is not hard to see that the level and volume of water contained in the vessel of each traffic flow represent, respectively, the corresponding running loss probability and loss amount in the n^{th} time slot.

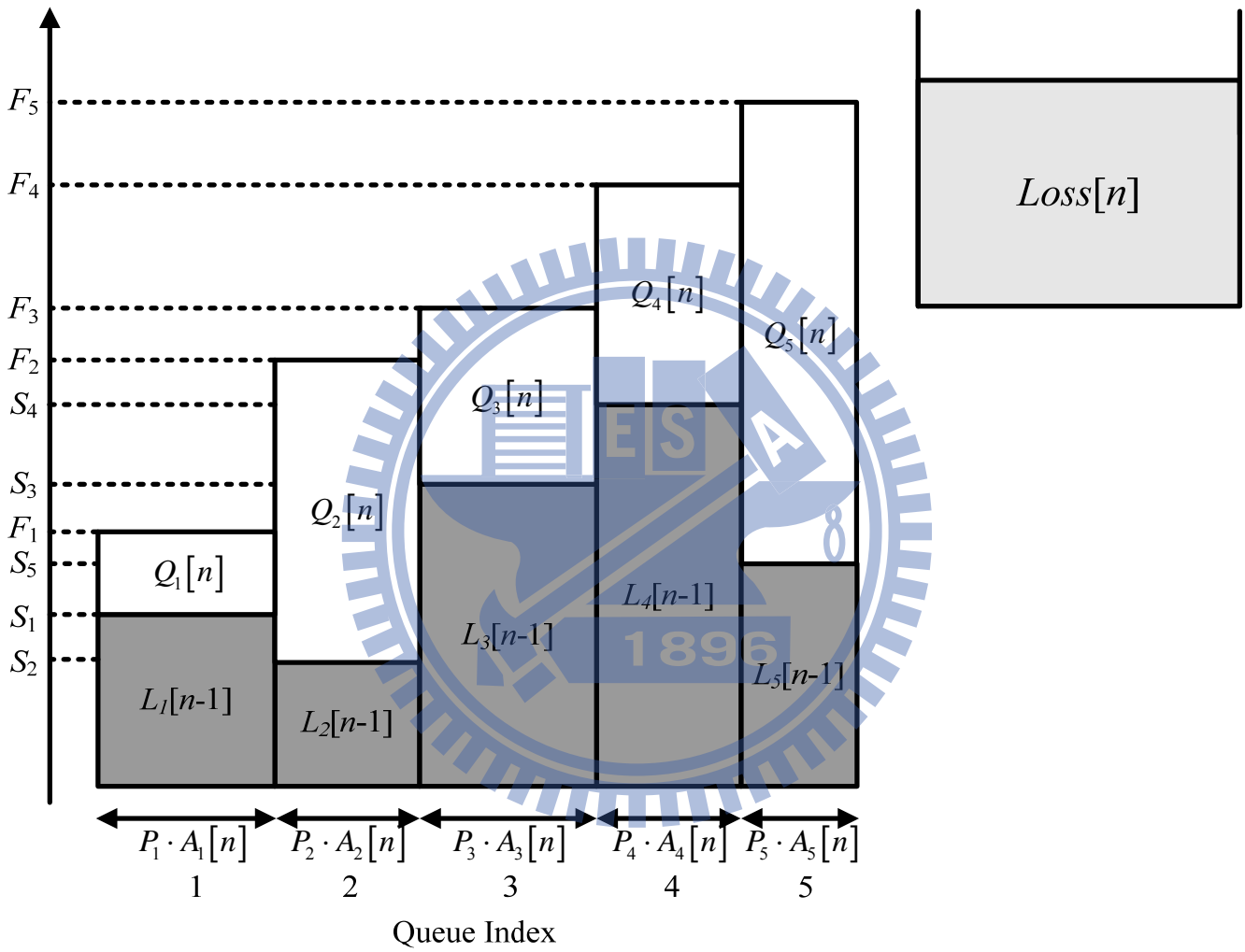


Fig. 5.2(a) Initial state

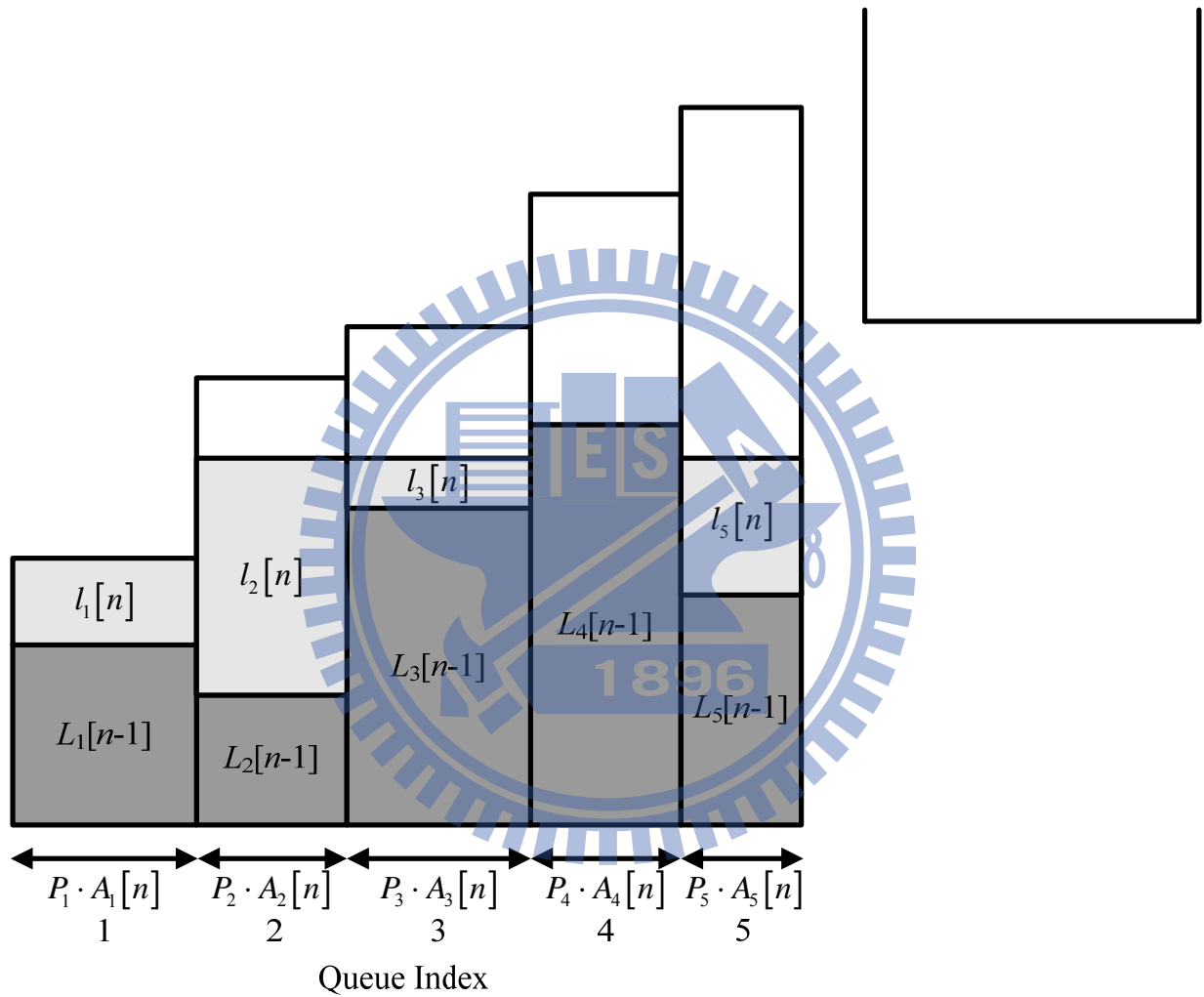


Fig. 5.2(b) The result of water-filling

Fig. 5.2 An example for illustrating PL queue management algorithm for traffic flows with identical delay bound requirement.

Inspired by the water-filling interpretation, we develop the proposed PL queue management algorithm by placing all flows belonging to U into their appropriate subsets (U_C , U_P , or U_Z) and then calculating their individual loss amount. Define

$$S_k = \frac{L_k[n-1]}{P_k \cdot A_k[n]}, \quad (45)$$

and

$$F_k = \frac{L_k[n-1] + Q_k[n]}{P_k \cdot A_k[n]}, \quad (46)$$

for all $f_k \in U$. Obviously, $P_k[n]$ equals S_k or F_k if “no” or “all” data is discarded in the n^{th} time slot. Without loss of generality, we assume that $U = \{f_1, f_2, \dots, f_K\}$ such that $F_k \leq F_{k+1}$, $1 \leq k \leq K-1$, and $F_0 = 0$. Let Ω_k , $1 \leq k \leq K$, be sub-sets of U such that $f_j \in \Omega_k$ iff $j \geq k$ and $S_j < F_k$. Further, define Λ_k , $1 \leq k \leq K$, as a sub-set of U_k such that $f_j \in \Lambda_k$ iff $S_j < F_{k-1}$.

For the example shown in Fig. 5.2, Ω_k and Λ_k , $1 \leq k \leq 5$, are listed in Table 5.1.

Table 5.1 Ω_k and Λ_k , $1 \leq k \leq 5$ for the example illustrated in Fig. 5.2.

	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
Ω_k	$\{f_1, f_2, f_5\}$	$\{f_2, f_3, f_4, f_5\}$	$\{f_3, f_4, f_5\}$	$\{f_4, f_5\}$	$\{f_5\}$
Λ_k	\emptyset	$\{f_2, f_5\}$	$\{f_3, f_4, f_5\}$	$\{f_4, f_5\}$	$\{f_5\}$

Initially, we set $U_C = U_P = U_Z = \emptyset$. To avoid the trivial case, assume that $K \geq 2$. The first phase of the proposed PL queue management algorithm decides which flows should be placed in

U_C . Define H_k , $1 \leq k \leq K$, as

$$H_k = \begin{cases} \sum_{f_r \in \Omega_k} (A_r[n]P_rF_k - L_r[n-1]) & \text{if } k=1 \\ \sum_{r=1}^{k-1} Q_r[n] + \sum_{f_r \in \Omega_k} (A_r[n]P_rF_k - L_r[n-1]) & \text{if } k > 1 \end{cases} \quad (47)$$

Note that H_k represents the capacity if the super vessel is filled with water (lost data), where the level is up to F_k . Therefore, we have $f_k \in U_C$ iff $H_k \leq \text{Loss}[n]$. Since, by assumption, $F_k \leq F_{k+1}$, we know that $f_j \in U_C$ implies $f_k \in U_C$ for all $k \leq j$. Consequently, to determine U_C , we only need to find the minimum k such that $H_k > \text{Loss}[n]$. Let e be the solution, and U_C can be obtained as

$$U_C = \begin{cases} \emptyset & \text{if } e=1 \\ \{1, 2, \dots, e-1\} & \text{if } e > 1 \end{cases} \quad (48)$$

The second phase of the proposed PL queue management algorithm decides which flow should be placed in U_P . It is not hard to see that $U_P \subseteq \Omega_e$ and $\Lambda_e \subseteq U_P$. As a result, the remaining work is to determine whether or not $f_j \in U_P$ for every $f_j \in (\Omega_e - \Lambda_e)$. Compute, for each $f_j \in (\Omega_e - \Lambda_e)$,

$$H_e^j = \sum_{r=1}^{e-1} Q_r[n] + \sum_{f_l \in \Omega_e} (A_l[n]P_lS_j - L_l[n-1])^+ \quad (49)$$

Note that H_e^j represents the capacity if the super vessel is filled with water (lost data) up to the level S_j . We have $f_j \in U_P$ iff 1) $f_j \in \Lambda_e$ or 2) $f_j \in (\Omega_e - \Lambda_e)$ and $H_e^j < \text{Loss}[n]$. After U_C and U_P are determined, one can obtain $U_Z = U - U_C - U_P$.

Once all flows are placed in U_C , U_P and U_Z appropriately, we have

$$l_k[n] = \begin{cases} 0 & \text{if } f_k \in U_Z \\ Q_k[n] & \text{if } f_k \in U_C \end{cases}. \quad (50)$$

For flows belonging to U_P , we can directly solve equations (43) and (44) to obtain $l_k[n]$. The solution is given by

$$l_k[n] = \frac{P_k \cdot A_k[n] \cdot \left(Loss[n] - \sum_{f_r \in U_C} Q_r[n] + \sum_{f_j \in U_P, j \neq k} L_j[n-1] \right) - L_k[n-1] \cdot \left(\sum_{f_j \in U_P, j \neq k} P_j \cdot A_j[n] \right)}{\sum_{f_j \in U_P} P_j \cdot A_j[n]}. \quad (51)$$

Note that the derivations of equation (51) is basically the same as those of equation (22) and thus are not repeated. Finally, discard $l_k[n]$ from the head of $Queue_k$ and update $L_k[n]$ by $L_k[n] = L_k[n-1] + l_k[n]$ for each $f_k \in U$, which completes the proposed PL queue management algorithm. To facilitate the presentation in the next sub-section, the above procedure based on the water-filling interpretation is represented as

$$\{l_k[n]\}_{k=1}^K = WF \left(Loss[n], \{L_k[n-1]\}_{k=1}^K, \{A_k[n]\}_{k=1}^K, \{Q_k[n]\}_{k=1}^K, \{P_k\}_{k=1}^K \right). \quad (52)$$

For the example shown in Fig. 5.2, we have $e = 2$, $U_C = \{f_1\}$, $U_P = \{f_2, f_3, f_5\}$ and $U_Z = \{f_4\}$. Note that our algorithm guarantees $P_j[n]/P_j = P_k[n]/P_k$ for all $f_j, f_k \in U_P$. For the considered example, we have $P_2[n]/P_2 = P_3[n]/P_3 = P_5[n]/P_5$.

● Flows with Different Delay Bound Requirements

Without loss of generality, let $U = \{f_1, f_2, \dots, f_K\}$ and $\beta_1 \leq \beta_2 \leq \dots \leq \beta_K$. Assume that all data buffered in the multiplexer is schedulable in the $(n-1)^{th}$ time slot and some data of f_k arrives in the n^{th} time slot. In the n^{th} time slot, it is not hard to see that all data buffered in the multiplexer

is schedulable iff

$$\sum_{m=1}^j \sum_{f_k \in U} Q_k^m [n] \leq j \cdot C, \quad (53)$$

holds for $j = \beta_k, \beta_k + 1, \dots, \beta_K$. Note that equation (53) must be true for $j = 1, 2, \dots, \beta_k - 1$ because

all data which can be buffered for less than or equal to $(\beta_k - 1)$ time slot without violating their

delay bound has higher priorities than the newly-arrived one of f_k . Similarly, if multiple flows

have data newly arrived, all data buffered in the multiplexer is schedulable iff equation (53) holds for

$j = \beta_{\min}, \beta_{\min} + 1, \dots, \beta_K$, where $\beta_{\min} = \min_{f_k \in U_{\text{new}}} \{\beta_k\}$ and U_{new} is a set which contains traffic flows

with newly-arrived data. Again, if all data in the multiplexer can be transmitted before their own

deadlines, we have $l_k [n] = 0$ and thus $L_k [n]$ can be updated by $L_k [n] = L_k [n-1]$, $1 \leq k \leq K$.

Assume that the schedulability test shown above fails. Define $Loss_0 [n] = 0$ and

$$Loss_i [n] = \left(\sum_{m=1}^i \sum_{f_k \in U} Q_k^m [n] - \sum_{m=0}^{i-1} Loss_m [n] - i \cdot C \right)^+, \quad (54)$$

for $i = 1, 2, \dots, \beta_K$. It is not hard to see that $Loss_i [n]$ is the total amount of data which will

violate their delay bound in the $(n+i-1)^{\text{th}}$ time slot and thus should be discarded, assuming that no

data arrives to the multiplexer in the future. For each $Loss_i [n] > 0$, we need to discard data which

can be buffered in the multiplexer without violating their delay bound for no longer than i time slots.

Denote $l_k^i [n]$, $1 \leq k \leq K$, as the corresponding loss amount of data belonging to f_k , and we have

$$0 \leq l_k^i [n] \leq \sum_{m=1}^i Q_k^m [n] - \sum_{m=0}^{i-1} l_k^m [n]. \quad (55)$$

Note that, for convenience, we set $l_k^i [n] = 0$, $1 \leq k \leq K$, if $Loss_i [n] = 0$.

Assume that $Loss_i[n] > 0$. Obviously, $l_k^i[n] = 0$ if $\sum_{m=1}^i Q_k^m[n] - \sum_{m=0}^{i-1} l_k^m[n] = 0$. Define U_i to be the set containing traffic flows, which satisfies $\sum_{m=1}^i Q_k^m[n] - \sum_{m=0}^{i-1} l_k^m[n] > 0$. Again, U_i can be divided into three disjoint sets, U_C^i , U_P^i and U_Z^i so that f_k is contained in U_C^i , U_P^i and U_Z^i iff $l_k^i[n] = \sum_{m=1}^i Q_k^m[n] - \sum_{m=0}^{i-1} l_k^m[n]$, $0 < l_k^i[n] < \sum_{m=1}^i Q_k^m[n] - \sum_{m=0}^{i-1} l_k^m[n]$ and $l_k^i[n] = 0$, respectively. For each $Loss_i[n] > 0$, the proposed PL queue management algorithm manages the queues so that, for any $f_c \in U_C^i$, $f_p, f_{p'} \in U_P^i$ and $f_z \in U_Z^i$,

$$\frac{P_c^i[n]}{P_c} \leq \frac{P_p^i[n]}{P_p} = \frac{P_{p'}^i[n]}{P_{p'}} \leq \frac{P_z^i[n]}{P_z}, \quad (56)$$

and

$$\sum_{f_k \in U_i} l_k^i[n] = Loss_i[n]. \quad (57)$$

Note that $P_k^i[n]$ is defined as $P_k^i[n] = (L_k[n-1] + \sum_{m=1}^i l_k^m[n]) / A_k[n]$.

Let Γ contain i such that $Loss_i[n] > 0$. To simultaneously meet equations (56) and (57) for each $i \in \Gamma$, we only have to execute WF iteratively for all $i \in \Gamma$ in an increasing order, the corresponding inputs and outputs of which are described as follows

$$\{l_k^i[n]\}_{k=1}^K = WF \left(Loss_i[n], \{L_k[n-1] + \sum_{m=1}^{i-1} l_k^m[n]\}_{k=1}^K, \{A_k[n]\}_{k=1}^K, \{\sum_{m=1}^i Q_k^m[n] - \sum_{m=1}^{i-1} l_k^m[n]\}_{k=1}^K, \{P_k\}_{k=1}^K \right) \quad (58)$$

Finally, we can have $l_k[n] = \sum_{i \in \Gamma} l_k^i[n]$, $1 \leq k \leq K$. Again, discarding $l_k[n]$ from the head of $Queue_k$ and updating $L_k[n]$ by $L_k[n] = L_k[n-1] + l_k[n]$, $1 \leq k \leq K$ completes the proposed PL queue management algorithm.

Fig. 5.3 illustrates an example considering flows with different delay bound requirements.

Assume that there are five traffic flows in the multiplexer, where $(\beta_1, \beta_2, \beta_3, \beta_4, \beta_5) = (1, 1, 2, 3, 3)$.

Each flow has data arrived in the n^{th} time slot and it holds that $Q_3^1[n] = 0$, and $Q_k^m > 0$ for

$(k, m) = (4, 1), (4, 2), (5, 1), (5, 2)$. After plugging the related information into equation (54), we

have $Loss_i[n] > 0$ for $i = 1, 3$ and $Loss_2[n] = 0$, meaning that *WF* needs to be performed for two

rounds. Fig. 5.3(a) shows the initial state of each flow in the first round. The results of the first

round are presented in Fig. 5.3 (b), which in turn to be the initial state of each flow in the second

round. In the first round, it holds that $U = \{f_1, f_2, f_4, f_5\}$, $U_C = \{f_4, f_5\}$, $U_P = \{f_1\}$ and

$U_Z = \{f_2\}$. Similarly, Fig. 5.3 (c) demonstrates the results of the second round. In the second

round, we have $U = \{f_1, f_2, f_3, f_4, f_5\}$, $U_C = \emptyset$, $U_P = \{f_3, f_4\}$ and $U_Z = \{f_1, f_2, f_5\}$.

It is clear that the theoretical results developed in [11] can also be applied to prove that the proposed PL queue management algorithm, coupling with EDF service scheduler, is optimal in the sense that the effective bandwidth is minimized under generalized space-conserving constraint. In

fact, we have shown the proposed PL queue management algorithm is generalized space-conserving

and the criterion shown in equation (43) is identical to that of G-QoS scheme, assuming the size of

cell is infinitesimally small.

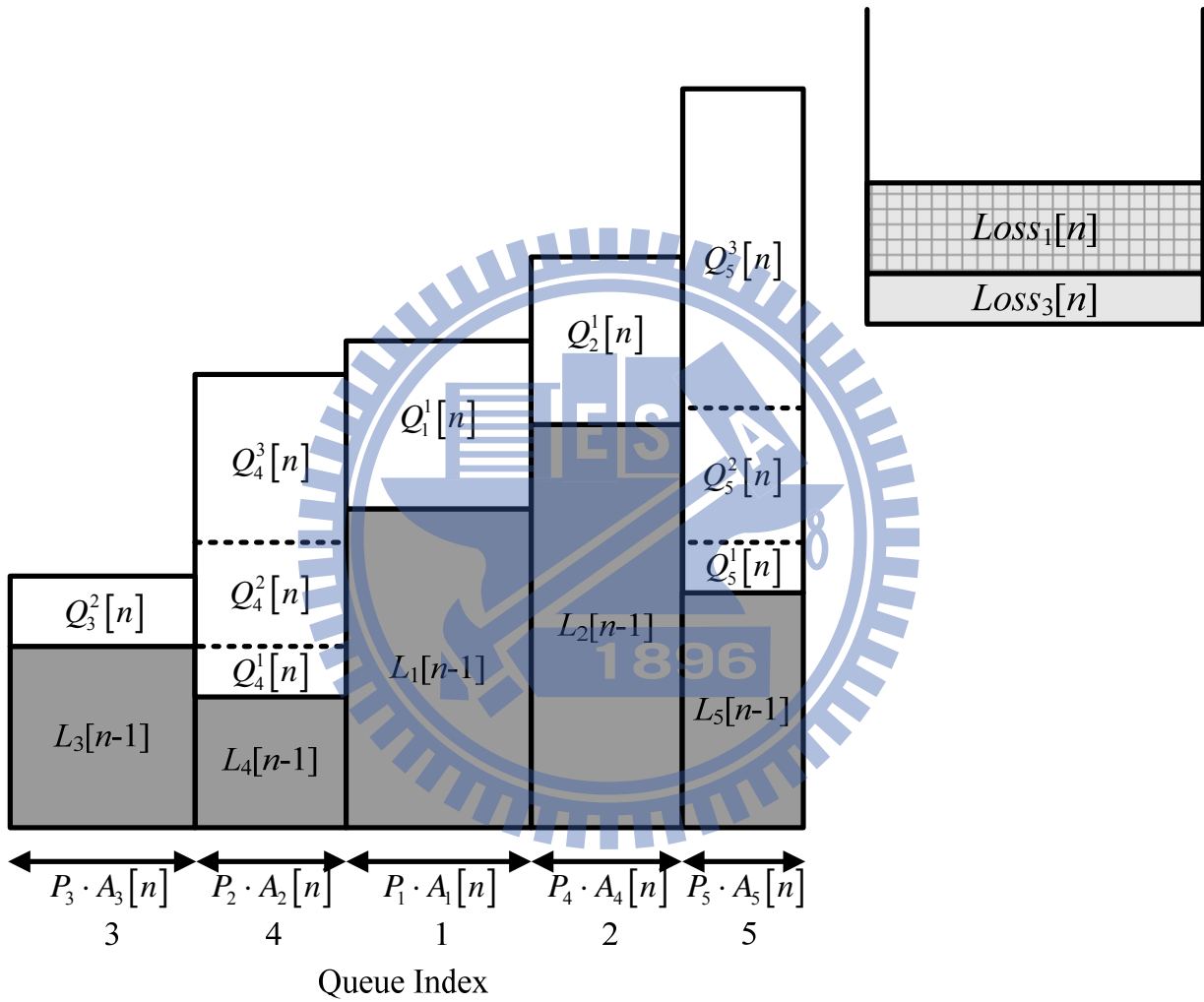


Fig. 5.3 (a) Initial state

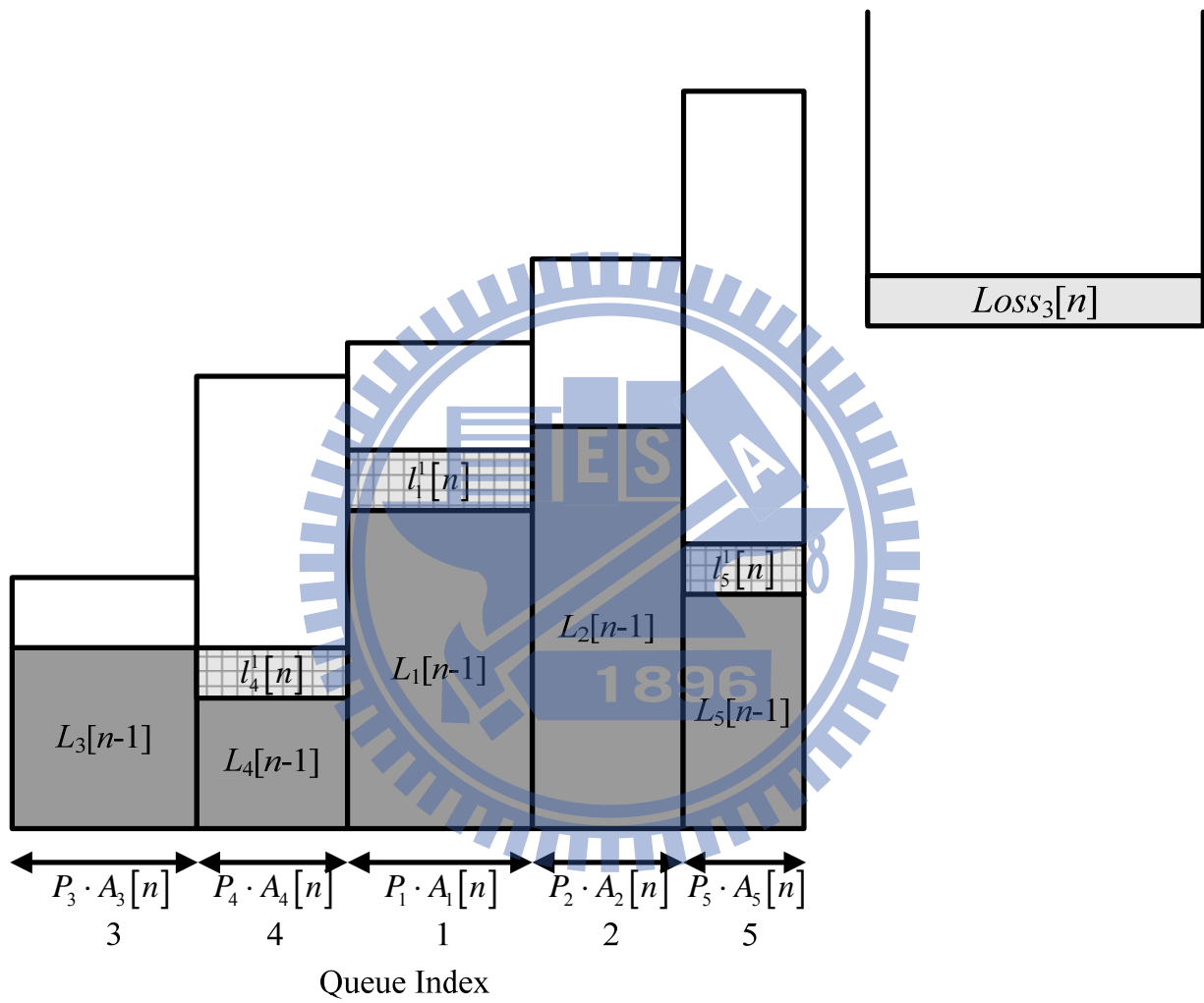


Fig. 5.3 (b) The result of the first water-filling

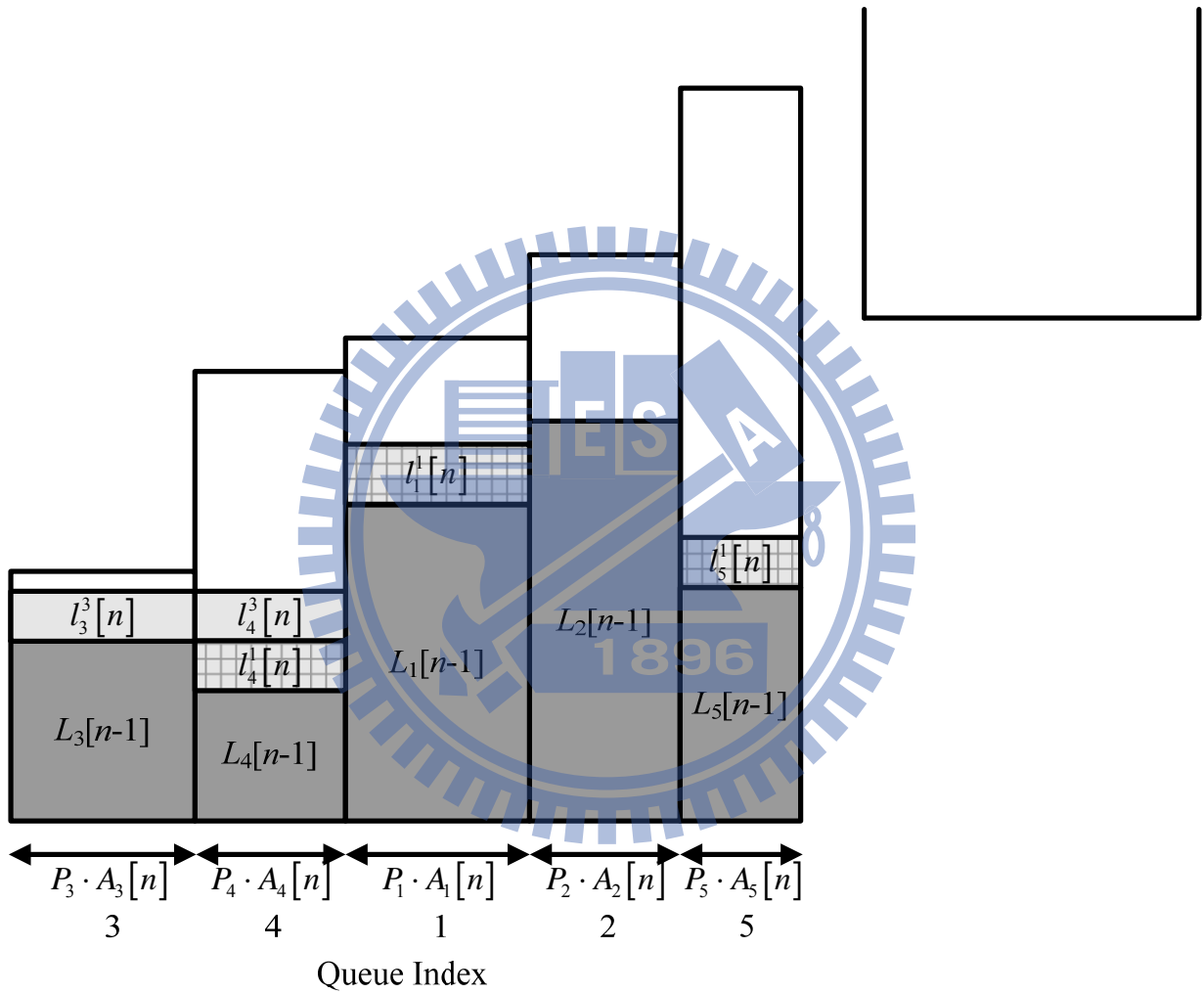


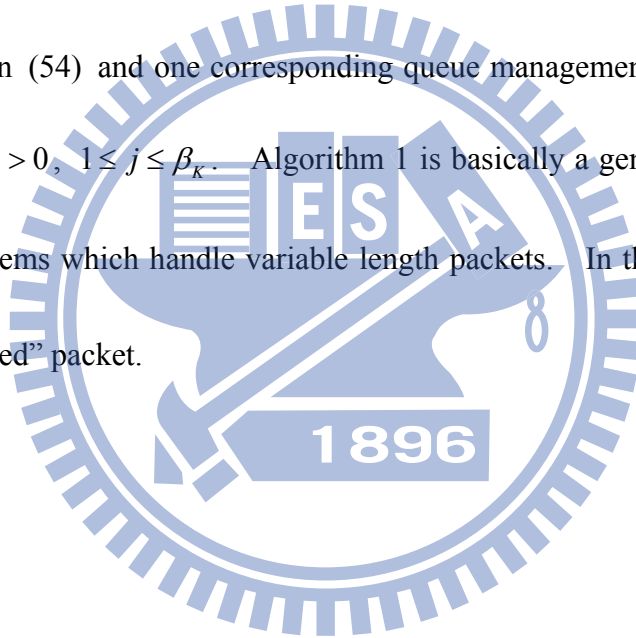
Fig. 5.3 (c) The result of the second water filling

Fig. 5.3 An example for illustrating PL queue management algorithm for traffic flows with different delay bound requirement.

5.3. Packet-based Systems

In this section, we present two algorithms for packet-based systems. In these algorithms, we break a tie, if exists, arbitrarily. Without loss of generality, we assume that $U = \{f_1, f_2, \dots, f_K\}$ and $\beta_1 \leq \beta_2 \leq \dots \leq \beta_K$.

Again, upon packets arrived, the way to decide if the data buffered in the multiplexer is schedulable is the same as that presented in the previous section. If not, we can obtain the loss amount by using equation (54) and one corresponding queue management is needed if there exists one j such that $Loss_j[n] > 0$, $1 \leq j \leq \beta_K$. Algorithm 1 is basically a generalization of the scheme proposed in [11] for systems which handle variable length packets. In this algorithm, t represents the length of the “discarded” packet.



Algorithm 1

Initialization:

$$L_k[n] = L_k[n-1], \quad 1 \leq k \leq K.$$

$$l_k^j[n] = 0, \quad 1 \leq j \leq \beta_k, \quad 1 \leq k \leq K.$$

Begin:

1. **Calculate** $Loss_j[n]$ for $\beta_1 \leq j \leq \beta_K$ according to equation (54)
 2. $\Gamma = \{j \mid \beta_1 \leq j \leq \beta_K, Loss_j[n] > 0\}$
 3. **For each** $j \in \Gamma$ in an increasing order
 4. **While** $Loss_j[n] > 0$
 5. $U = \{f_k \mid \sum_{m=1}^j Q_k^m[n] - \sum_{m=1}^{j-1} l_k^m[n] > 0\}$
 6. $k^* = \arg \min_{k \in U} \{L_k[n] / A_k[n] P_k\}$
 7. **Discard the packet on the head of** $Queue_{k^*}$
 8. $l_{k^*}^j[n] = l_{k^*}^j[n] + t$
 9. $L_{k^*}[n] = L_{k^*}[n] + t$
 10. $Loss_j[n] = Loss_j[n] - t$
 11. **End While**
 12. **Calculate** $Loss_{j'}[n]$ for $j < j' \leq \beta_K$ according to equation (54)
 13. $\Gamma = \{j' \mid j < j' \leq \beta_K, Loss_{j'}[n] > 0\}$
 14. **End For**
 15. $l_k[n] = \sum_{j=\beta_1}^{\beta_k} l_k^j[n], \quad 1 \leq k \leq K.$
 16. $L_k[n] = L_k[n-1] + l_k[n], \quad 1 \leq k \leq K.$
-

The second algorithm slightly differs from the first one in the content to be minimized. In this algorithm, t_k represents the length of the oldest packet of $Queue_k$. Note that it selects, packet by packet, the queue that minimizes the maximum of normalized running packet loss probability.

Algorithm 2

Initialization:

$$L_k[n] = L_k[n-1], \quad 1 \leq k \leq K.$$

$$l_k^j[n] = 0, \quad 1 \leq j \leq \beta_k, \quad 1 \leq k \leq K.$$

Begin:

1. **Calculate** $Loss_j[n]$ for $\beta_1 \leq j \leq \beta_K$ according to equation (54)
 2. $\Gamma = \{j \mid \beta_1 \leq j \leq \beta_K, Loss_j[n] > 0\}$
 3. **For each** $j \in \Gamma$ in an increasing order
 4. **While** $Loss_j[n] > 0$
 5. $U = \{f_k \mid \sum_{m=1}^j Q_k^m[n] - \sum_{m=1}^{j-1} l_k^m[n] > 0\}$
 6. $k^* = \arg \min_{k \in U} \{(L_k[n] + t_k) / A_k[n] P_k\}$
 7. **Discard the packet on the head of** $Queue_{k^*}$.
 8. $l_{k^*}^j[n] = l_{k^*}^j[n] + t_{k^*}$
 9. $L_{k^*}[n] = L_{k^*}[n] + t$
 10. $Loss_j[n] = Loss_j[n] - t_{k^*}$
 11. **End While**
 12. **Calculate** $Loss_{j'}[n]$ for $j < j' \leq \beta_K$ according to equation (54)
 13. $\Gamma = \{j' \mid j < j' \leq \beta_K, Loss_{j'}[n] > 0\}$
 14. **End For**
 15. $l_k[n] = \sum_{j=\beta_1}^{\beta_k} l_k^j[n], \quad 1 \leq k \leq K.$
 16. $L_k[n] = L_k[n-1] + l_k[n], \quad 1 \leq k \leq K.$
-

5.4. Simulation Results

In this section, we evaluate the transient and steady-state performance of our proposed PL queue management algorithm and the two packet-based algorithms that can be implemented for real systems. We assume that there exist five traffic flows which can be generated by video trace files [54].

We adopt both interactive (parking and lecture camera) and non-interactive (Die Hard III, Mr. Bean, and Starship Troopers) videos in the simulation. The traffic characteristics and the QoS requirements (including the required delay bound and packet loss probability) are summarized in Tables 5.2.

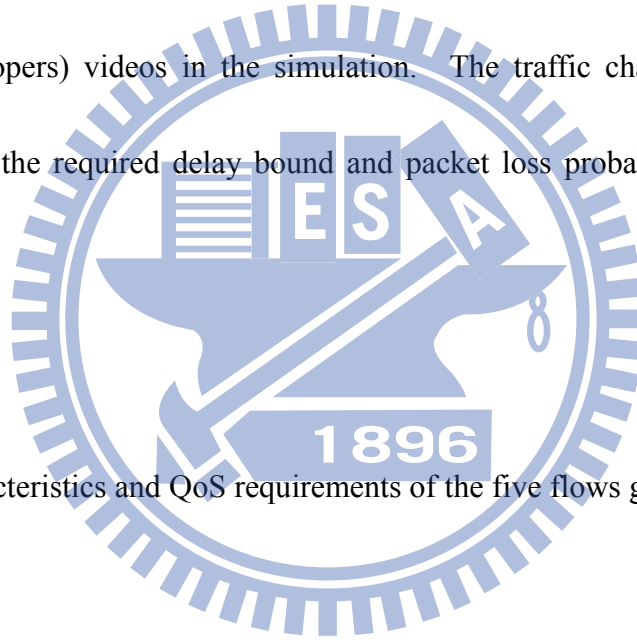


Table 5.2 Traffic characteristics and QoS requirements of the five flows generated from video trace files.

Traffic flow No.	1	2	3	4	5
Video name	Parking Cam	Lecture Cam	Die Hard III	Mr. Bean	Starship Troopers
Video type	Interactive	Interactive	Non-interactive	Non-interactive	Non-interactive
Mean data rate (Kbps)	236	58	246	184	202
Peak data rate (Kbps)	1551	686	1632	1513	1453
Mean packet size (bytes)	1182	288	1232	919	1008
Delay bound (ms)	160	160	80	80	80
Packet loss probability	0.01	0.008	0.006	0.004	0.002

To investigate the steady-state performance, the simulation for flows generated video trace files is conducted by repeating the files for 20 times. The length of each time slot is assumed to be 80 ms. The proposed algorithm presented in Chapter 5.3 is referred to as the fluid-flow based algorithm. For performance comparisons, we let the system capacity equal the effective bandwidth under the fluid-flow based algorithm. Note that the effective bandwidth, which is defined as the minimum bandwidth to meet the QoS requirements of all traffic flows, can be found in advance by, say, the bisection method.

Fig. 5.4 shows the evolution of running packet loss probabilities of the five traffic flows. As one can see in Figs. 5.4(a), the steady-state loss probabilities meet the requirements for the fluid-flow based algorithm. The reason is simply because we used the effective bandwidths in both experiments. The loss probabilities are about 2.6 and 2.5 times of the desired upper bounds under the packet based algorithms I and II, respectively. Note that the fluid-flow based algorithm achieves the goal of maintaining the ratios of steady-state packet loss probabilities equal to those of the requested values, which can be seen from the results shown in Table 5.3. In this table, $P_{L,mean}$ the steady-state loss probability, which becomes $P_{L,norm}$ after the normalizing it by the loss probability requirement. For the two packet-based algorithms, the $P_{L,norm}$ values of different flows fluctuate slightly due to the constraint of handling packets as data units.

Table 5.3 Steady-state (normalized) packet loss probability for flows generated from video trace files.

Traffic flow No.	1	2	3	4	5
PL algorithm (P_L)	0.0100	0.0080	0.0060	0.0040	0.0020
Packet-based algorithm I (P_L)	0.0266	0.0213	0.0159	0.0106	0.0053
Packet-based algorithm II (P_L)	0.0252	0.0202	0.0151	0.0101	0.0050
PL algorithm ($P_{L,norm}$)	1.0000	1.0000	1.0000	1.0000	1.0000
Packet-based algorithm I ($P_{L,norm}$)	2.6582	2.6571	2.6569	2.6572	2.6569
Packet-based algorithm II ($P_{L,norm}$)	2.5225	2.5205	2.5204	2.5206	2.5204

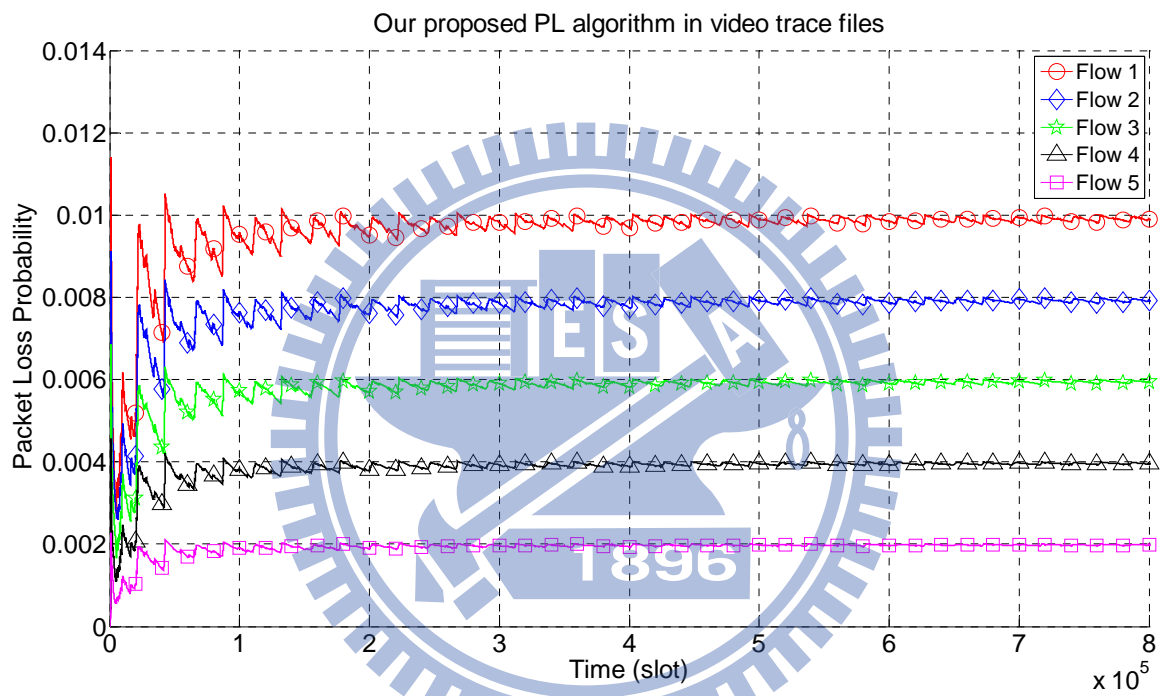


Fig. 5.4 (a)

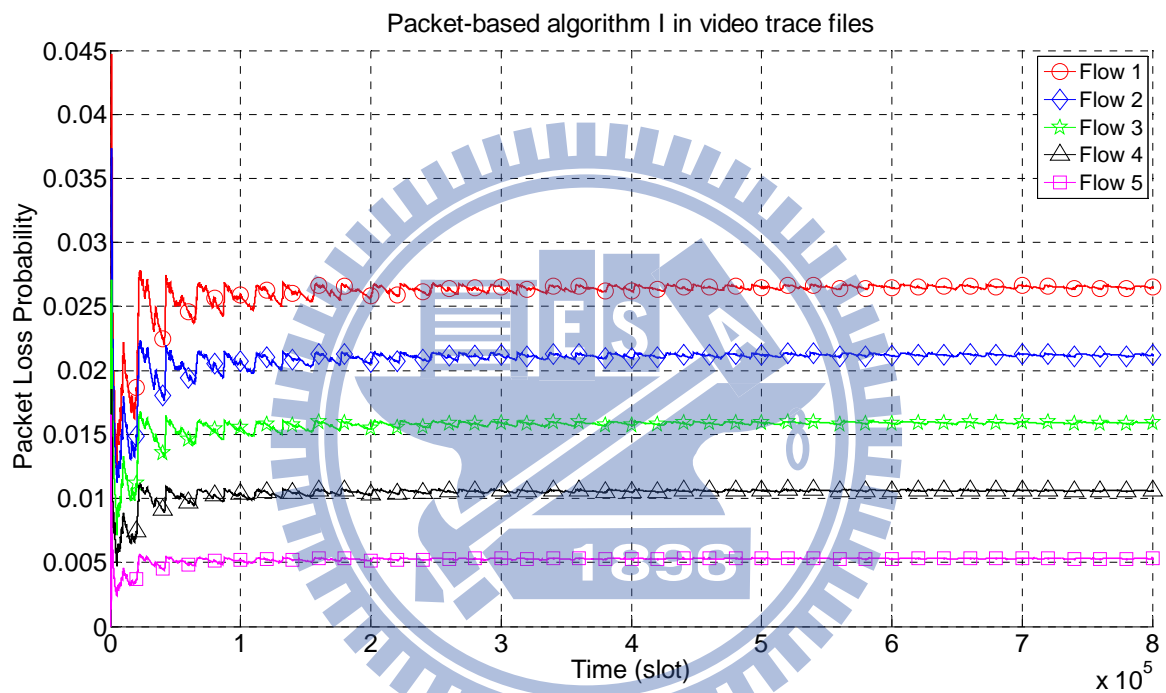


Fig. 5.4 (b)

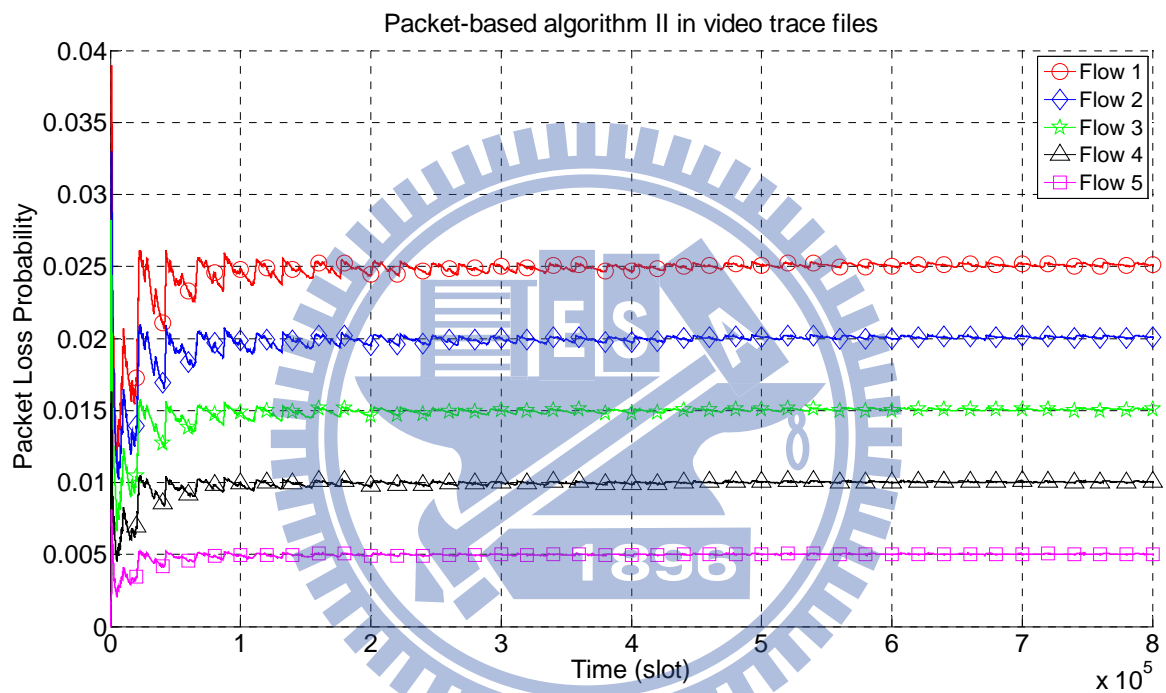


Fig. 5.4 (c)

Fig. 5 4 Sample Path of packet loss probability for video trace files with (a) our proposed PL queue management algorithm (b) Packet-based algorithm I (c) Packet-based algorithm II adopted.

Chapter 6

Conclusions

In this dissertation, we have studied the resource allocation technique for IEEE 802.11e HCCA, OFDMA-based systems and finally extended the results for real-time traffic to a general multiplexing system. The conclusions and future works are drawn below.

In IEEE 802.11e HCCA, we have presented an efficient static TXOP allocation algorithm, a proportional-loss fair service scheduler, and the associated admission control unit to provide QoS guarantee for VBR traffic flows with different packet loss probability and delay bound requirements.

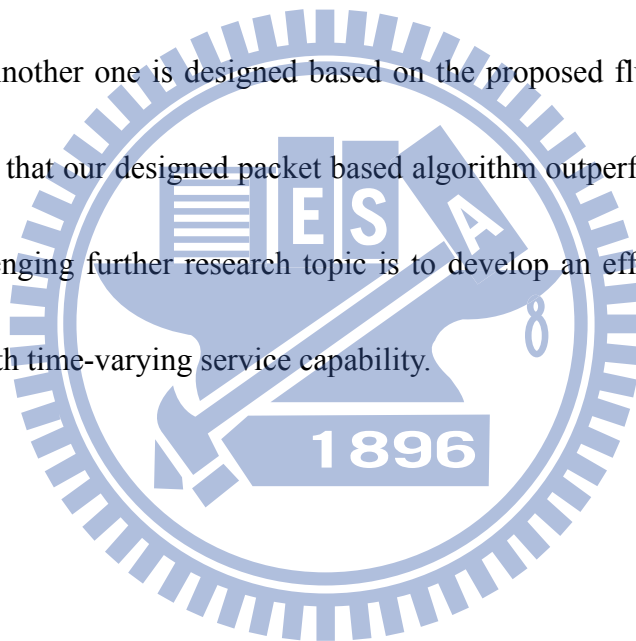
Computer simulations are conducted to evaluate the performance of our proposed scheme. Results show that our proposed scheme is effective in QoS guarantee and, moreover, performs much better than previous works. Our proposed proportional-loss fair service scheduler can also be combined with dynamic TXOP allocation algorithms to provide better QoS support. In real systems, it is likely that there are only a limited number of possible applications. Therefore, one can pre-compute the QoS parameter of each type of application so that admission control can be

performed in real time. An interesting further research topic is to extend the results to different traffic models and other types of wireless networks.

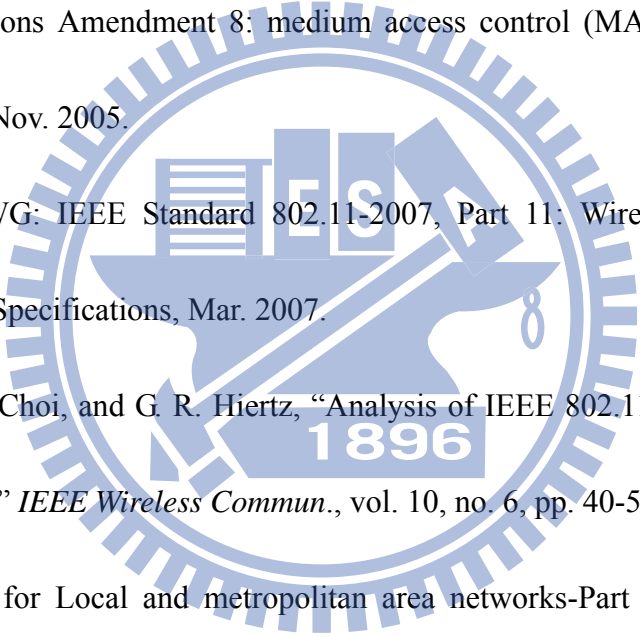
In OFDMA-based systems, we have presented an efficient resource allocation scheme which tries to maximize system throughput while providing QoS support to real-time traffic flows. The basic idea of our proposed scheme is to calculate a dynamic minimum requested bandwidth for each traffic flow and use it as a constraint in an optimization problem which maximizes system throughput. The minimum requested bandwidth is a function of the pre-defined loss probability and the running loss probability. In addition, a user-level PL scheduler is proposed to determine the bandwidth share for multiple real-time flows attached to the same SS. A pre-processor is adopted to maximize the number of real-time flows attached to each SS which meet their QoS requirements, when the resource is not sufficient to provide every flow its minimum requested bandwidth. Computer simulations were conducted to evaluate the performance of our proposed scheme. Results show that the running loss probabilities of traffic flows attached to the same SS are effectively controlled to be proportional to their loss probability requirements. Besides, compared with previous designs, our proposed scheme achieves higher throughput while providing QoS support. Although we present our designs for long time average of loss probabilities, the idea can be applied to other measurements such as exponentially weighted moving average. How to design a pre-processor which meets user's need is an interesting topic which can be further studied. Evaluation of the impact to user perception of satisfaction for various performance measurements is

another potential further research topic.

Finally, we consider a general multiplexing system. We proposed a PL queue management algorithm for packet discarding. With combined with EDF service scheduler, we show that our proposed queue management algorithm is optimal in the sense that it minimizes the effective bandwidth under generalized space-conserving constraint. Two packet based algorithms were studied for real systems. One of them is a direct extension of a previous scheme which handles fixed-length packets. Another one is designed based on the proposed fluid-flow based algorithm. Simulations results show that our designed packet based algorithm outperforms the direct extension. An interesting but challenging further research topic is to develop an efficient queue management algorithm for systems with time-varying service capability.



Biography

- 
- [1] IEEE Std. 802.11e-2005, Part 11: Wireless LAN medium access control and physical layer specifications Amendment 8: medium access control (MAC) quality of service enhancements, Nov. 2005.
- [2] IEEE 802.11 WG: IEEE Standard 802.11-2007, Part 11: Wireless LAN MAC and Physical Layer Specifications, Mar. 2007.
- [3] S. Mangold, S. Choi, and G. R. Hiertz, "Analysis of IEEE 802.11e for QoS support in Wireless LANs," *IEEE Wireless Commun.*, vol. 10, no. 6, pp. 40-50, Dec. 2003.
- [4] IEEE Standard for Local and metropolitan area networks-Part 16: Air Interface for Fixed Broadband Wireless Access Systems, IEEE Std. 802.16-2009, May 2009.
- [5] E. Dahlman, S. Parkvall, J. Sköld, and P. Beming, "3G HSPA and LTE for Mobile Broadband," *New York: Academic*, 2007.
- [6] C.L. Liu and J.W. Layland, "Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment," *J. ACM*, vol. 20, no. 1, pp. 46-61, 1973.
- [7] A. K. Parekh and R.G. Gallager, "A Generalized Processor Sharing Approach to flow

- control in Integrated Services Networks - The Single Node Case," *IEEE/ACM Trans. on Networking*, vol. 1, no.3, pp. 344-357, Jun. 1993.
- [8] A. K. Parekh and R.G. Gallager, "A Generalized Processor Sharing Approach to flow control in Integrated Services Networks - The Multiple Node Case," *IEEE/ACM Trans. on Networking*, vol. 2, no.1, pp. 137-150, Apr. 1994.
- [9] G. Hebuterne and A. Gravey, "A Space Priority Queueing Mechanism for Multiplexing ATM Channels", *Computer Networks and ISDN system*, vol. 20, pp. 37-43,1990.
- [10] S. Sumita and T. Ozawa, "Achievability of Performance Objectives in ATM Switching Nodes", in *Proc. International Seminar on Performance of Distributed and Parallel Systems*, pp.45-56, Dec. 1988.
- [11] T. Yang, D. Tsang, and P. McCabe, "Cell scheduling and bandwidth allocation for heterogeneous VBR video conferencing traffic", in *Proc. IEEE GLOBECOM'95*, vol.1, pp. 371-377, Nov. 1995.
- [12] T. Yang and J. Pan, "A Measurement-Based Loss Scheduling Scheme," in *Proc. IEEE INFOCOM'96*, vol. 3, pp. 1062-1071, Mar. 1996.
- [13] H. Kroner, "Comparative Performance Study of Space Priority Mechanisms for ATM networks," in *Proc. IEEE INFOCOM'90*, vol.3, pp. 1136-1143, Jun. 1990.
- [14] H. Kroner, G. Hebuterne, P. Boyer and A Gravey, "Priority Management in ATM Switching Nodes", *IEEE J. Select. Areas Commun.* vol. 9, no.3 pp. 418-427, 1991.

- [15] N. Lin, S. Li and T. Stern, "Congestion Control for Packet Voice by Selective Packet Discarding", *IEEE Trans. on Commun.* vol. 38, no. 5, pp. 674-683, May 1990.
- [16] W. F. Fan, D. Y. Gao, D. H. K. Tsang and B. Bensaou, "Admission Control for variable bit rate traffic in IEEE 802.11e WLANs," in *Proc. IEEE LANMAN'04*, pp. 61-66, Apr. 2004.
- [17] Gao, D., Cai, J., and Chen, C. W., " Admission control based on rate-variance envelop for VBR traffic over IEEE 802.11e HCCA WLANs", *IEEE Trans. on Vehicular Tech.*, vol. 57, no. 3, pp. 1778-1788, May 2008.
- [18] Cicconetti, C., Lenzini, L., Mingozi, E., and Stea, G., "An efficient cross layer scheduler for multimedia Traffic in Wireless Local Area Networks with IEEE 802.11e HCCA", *ACM Mobile Computing and Commun. Review*, vol. 11, no. 3, pp. 31-46, Jul. 2007.
- [19] Higuchi, Y., Foronda, A., Ohta, C., Yoshimoto M., and Okada, Y., "Delay guarantee and service interval optimization for HCCA in IEEE 802.11e WLANs," in *Proc. of IEEE WCNC'07*, pp. 2080-2085, Mar. 2007.
- [20] Rashid, M. M., Hossain, E., and Bharggava, V. K., "Controlled channel access scheduling for guaranteed QoS in IEEE 802.11e-based WLANs," *IEEE Trans. Wireless Commun.* vol. 7, no. 4, pp.1287-1297, Apr. 2008.
- [21] Bourawy, A. A., AbuAli, N. A., and Hassanein, H. S., "A selectivity function scheduler

- for IEEE 802.11e”, in *Proc. of ISCC’09*, pp. 950-955, Jul. 2009.
- [22] Huang, J. J., Chen, Y. H., and Chang, C.Y., “An MSI-based scheduler for IEEE 802.11e HCCA,” in *Proc. of IEEE VTC-Fall’09*, pp. 1-5, Sep. 2009.
- [23] Luo H., and Shyu, M. L., “An optimized scheduling scheme to provide quality of service in 802.11e Wireless LANs”, in *Proc. of IEEE ISM’09*, pp. 651-656, Dec. 2009.
- [24] Huang J. J., Chen Y. H. and Shiung D., “ A Four-Way–Polling QoS scheduler for IEEE 802.11e HCCA,” in *Proc. of IEEE TENCON’10*, pp. 1986-1991, Nov. 2009.
- [25] Hantrakoon S. and Phonphoem A., 2010, “Priority based HCCA for IEEE 802.11e,” in *Proc. of CMC’10*, pp.485-489, Apr. 2010
- [26] M. Kaneko, P. Popovski and J. Dahl, “Proportional fairness in multi-carrier system with multi-slot frames: upper bound and user multiplexing algorithms,” *IEEE Trans. on Wireless Commun.* vol. 7, no. 1, pp. 22-26, Jan. 2008.
- [27] N. Ruangchaijatupon and Y. Ji, “Simple Proportional Fairness Scheduling for OFDMA-based Wireless Systems,” in *Proc. IEEE WCNC’08*, pp.1593-1597, Mar. 2008.
- [28] N. Ruangchaijatupon and Y. Ji, “OFDMA Resource Allocation Based on Traffic Class-Oriented Optimization,” *IEICE Trans. on Commun.*, vol. E92-B, no.1, pp. 93-101, Jan, 2009.
- [29] N. Ruangchaijatupon and Y. Ji, “Integrated approach to proportional fair resource

- allocation for multiclass services in an OFDMA system,” in *Proc. IEEE GLOBECOM'09*, Dec. 2009.
- [30] D. S. W. Hui, V. K. N. Lau and W. H. Lam, “Cross-layer design for OFDMA wireless systems with heterogeneous delay requirements,” *IEEE Trans. on Wireless Commun.* vol. 6, no. 8, pp. 2872-2880, Aug. 2007.
- [31] J. Jang and K. B. Lee, “Transmit power adaptation for multiuser OFDM system,” *IEEE J. Select. Areas in Commun.*, vol. 21, no. 12, pp. 171-178, Feb. 2003.
- [32] S. Shakkottai and A. L. Stolyar, “A study of scheduling algorithms for a mixture of real and non-real time data in hdr,” Bell Laboratories, Lucent Technologies, Oct. 2000.
- [33] M. Andrews, K. Kumaran, K. Ramanan, A. L. Stolyar, P. Whiting, and R. Vijayakumar, “Providing quality of service over a shared wireless link,” *IEEE Commun. Mag.*, vol. 39, no. 2, pp. 150–154, Feb. 2001.
- [34] A. K. F. Khattab and K. M. F. Elsayed, “Opportunistic scheduling of delay sensitive traffic in OFDMA-based networks,” in *Proc. IEEE WOWMOM'06*, pp.109-114, Jun. 2006.
- [35] X. Zhu, J. Huo, C. Xu and W. Ding, “QoS-guaranteed scheduling and resource allocation algorithm for IEEE 802.16 OFDMA system,” in *Proc. IEEE ICC'08*, pp. 3463-3468, May 2008.
- [36] Y. Kim, K. Son and S. Chong, “QoS scheduling for heterogeneous traffic in

- OFDMA-based wireless systems,” in *Proc. IEEE GLOBECOM'09*, Dec. 2009.
- [37] R. Chipalkatti, J. Jurose, and D. Towsley, “Scheduling policies for real-time and non-real-time traffic in a statistical multiplexer,” in *Proc. IEEE INFOCOM'89*, pp. 774–783, Apr. 1989.
- [38] R. Yang, C. Yuan, and K. Yang, “Cross Layer Resource Allocation of Delay Sensitive Service in OFDMA Wireless Systems,” in *Proc. IEEE ICCSC'08*, pp. 862–866, May 2008.
- [39] V. Huang and W. Zhuang, “QoS-Oriented Packet Scheduling for Wireless Multimedia CDMA Communications,” *IEEE Trans. Mobile Computing*, pp. 73–85, Jan. 2004.
- [40] A. Frank, “On Kuhn’s Hungarian Method - A tribute from Hungary,” *Naval Research Logistics*, vol. 52, no. 1, pp. 2–5, Dec. 2005.
- [41] J. Mo and J. Walrand, “Fair end-to-end window-based congestion control,” *IEEE/ACM Trans. Networking*, vol. 8, no. 5, pp. 556–567, Oct. 2000.
- [42] C. Dovrolis and P. Ramanathann, “A Case for Relative Differentiated Services and the Proportional Differentiation Model”, *IEEE Network*, vol. 13, no. 5, pp. 26-34, Oct. 1999.
- [43] C. Dovrolis, D. Stiliadis and P. Ramanathann, “Proportional Proportional differentiated services: delay differentiation and packet scheduling”, *ACM SIGCOMM Computer Commun. Review*, vol. 29, no. 4, pp. 109-120, Oct. 1999.

- [44] C. Dovrolis, D. Stiliadis and P. Ramanathann, "Proportional differentiated services: delay differentiation and packet scheduling", *IEEE/ACM Trans. on Networking*, vol. 10, no. 1, pp. 12-26, Feb. 2002.
- [45] C. Dovrolis and P. Ramanathann, "Proportional Differentiated Services, Part II: Loss Rate Differentiation and Packet Dropping," in *Proc. IEEE IWQoS'00*, pp.53-61, Jun. 2000.
- [46] U. Bobin, A. Jonsson, O. Schelen, "On creating proportional loss-rate differentiation: predictability and performance", *Lecture Notes in Computer Science* 2092 (2001) 372-379.
- [47] J. Zeng and N. Ansari, "An Enhanced Dropping Scheme for Proportional Differentiated Services," in *Proc. IEEE ICC'03*, vol.3, pp.1897-1901, May 2003.
- [48] Y. C. Lai and Y. C. Szu, "Achieving Proportional Loss Rate Differentiation in A Wireless Network with A Multi-State Link," *Computer Commun.*, vol. 31 no. 10, Jun. 2008.
- [49] Y. Xie and T. Yang, "Cell Discarding Policies Supporting Multiple Delay and Loss Requirements in ATM Networks," in *Proc. IEEE GLOBECOM'97*, vol.2, pp. 1075-1080, Nov. 1997.
- [50] H. S. Kim and N. B. Shroff, "Loss probability calculations and asymptotic analysis for finite buffer multiplexers," *IEEE/ACM Trans. Networking*, vol. 9, no. 6, pp. 755-768,

Dec. 2001.

- [51] Y. W. Huang, T. H. Lee and J. R. Hsieh, "Gaussian approximation based admission control for variable bit rate traffic in IEEE 802.11e WLANs," in *Proc. IEEE WCNC'07*, pp. 3768-3773, Mar. 2007.
- [52] B. S. Kim, S. Kim, Y. Fang, and T.F. Wong, "Two-step multipolling MAC protocol for wireless LANs," *IEEE J. Select. Areas in Commun.*, vol. 23, no. 6, pp. 1276-1286, June 2005.
- [53] L. Georgiadis, R. Guerin, A. Parekh, "Optimal multiplexing on a single link: delay and buffer requirements", *IEEE Trans. Inform. Theory*, vol. 43, no. 5, pp.1518-1535, Sep. 1997.
- [54] MPEG-4 and H.263 video traces for network performance evaluation, <http://www.tkn.tu-berlin.de/research/trace/trace.html>, Oct. 2006.
- [55] J. E. Beasley, "Advances in linear and integer Programming," *Oxford Science*, 1996.
- [56] A. Schrijver, "Theory of linear and integer programming", *Wiley*, 1986.

Appendix A

Derivations of all equations and proofs of all lemmas and theorems

Proof of Theorem 3.1

Assume that $l'_{i,j}[n] \leq l_{i,j}[n]$ for some (i, j) . According to equation (20), we have $l'_{a,b}[n] \leq l_{a,b}[n]$ for any $(a, b) \in U$. As a result, it holds that $\sum_{(a,b) \in U} l'_{a,b}[n] \leq \sum_{(a,b) \in U} l_{a,b}[n] = \sum_{(a,b) \in U_{active}} l_{a,b}[n] - l_{r,s}[n] < \sum_{(a,b) \in U_{active}} l_{a,b}[n] - Q_{r,s}^m[n] = Loss'[n]$.

This contradicts equation (21). Therefore, Theorem 3.1 is true.

Proof of Theorem 3.3

It is clear that the solution of the last iteration falls in Case 1. Let M denote the size of U in that iteration. We shall prove that the loss computation algorithm takes at most $2(N - M)$ iterations to find the feasible solution if $M < N$ or one iteration if $M = N$. The case of $M = N$ is obviously true. We prove the case of $M < N$ by mathematical induction. For simplicity, we

use Sub-case i ($i = 1, 2$) to represent Sub-case i of Case 4 in this proof.

For $N = 2$, we have $M = 1$. Since $M < N$, we know that the solution of the first iteration cannot fall in Case 1. By tracing the algorithm, one can see that the number of iterations required to find the feasible solution is equal to $2 = 2(N - M)$. Assume that the statement is true for $N = H$ and $M = 1, 2, \dots, H - 1$ (Hypothesis I). Consider the case of $N = H + 1$. If Sub-case 2 is never visited, then the number of iterations required is at most $N - M + 1 \leq 2(N - M)$ because at least one queue is removed from U_{active} in each iteration before the last one. Assume that Sub-case 2 was visited before the feasible solution is found. If the solution of the first iteration does not fall in Sub-case 2, then the size of U in the second iteration is at most H . According to Hypothesis I, the maximum number of iterations required to find the feasible solution, starting from iteration 2, is equal to $2(H - M)$. As a result, the total number of iterations is upper bounded by $2(H - M) + 1 < 2(N - M)$.

Assume that the solution of the first iteration falls in Sub-case 2. Let $|V_1| = i$ and $|V_2| = j$ with $i + j = N$. Further, let k represent the number of queues added to V_2 when iteration 1 resumes its execution. The total number of iterations required is at most $1 + B(i, k) + 2(j + k - M)$, where $B(i, k)$ represents the maximum number of iterations required before iteration 1 resumes its execution and $2(j + k - M)$ denotes the upper bound of the number of iterations required to find the feasible solution for the updated V_2 , according to Hypothesis I. Theorem 3.3 is true if $B(i, k) \leq 2(i - k) - 1$. We shall prove this by mathematical induction.

By tracing the algorithm one can see that it is true for $i = 2$ and $k = 0$ or 1 . Assume that it is true for $i = p$ and $k = 0, 1, \dots, p-1$ (Hypothesis II). Consider the case of $i = p+1$. If Sub-case 2 is not visited again before iteration 1 resumes its execution, then we have $B(i, k) \leq i - k$. Note that if $k = 0$, then Case 2 is not visited. If $k > 0$, then there are 0 to $(i - k - 1)$ times of Sub-case 1 followed by a Case 2. Since $k \leq i - 1$, we have $B(i, k) \leq 2(i - k) - 1$. Assume that, before Sub-case 1 resumes its execution, Sub-case 2 is visited for the second time in iteration r . This implies the solutions of iterations 2, ..., and $r-1$ all fall in Sub-case 1 and, therefore, at least $r-2$ queues are removed from U_{active} . Let x , y , and z represent, respectively, the size of V_1 , the size of V_2 , and the number of queues added to V_2 when iteration r resumes its execution. It is clear that $x + y \leq i - r + 2$. After iteration r resumes its execution, the situation is the same as iteration 1 except that the size of V_1 (of iteration 1) is changed from i to $y + z$. As a result, we have $B(i, k) \leq (r-1) + B(x, z) + B(y + z, k)$. According to Hypothesis II, it holds that $B(i, k) \leq (r-1) + 2(x - z) - 1 + 2(y + z - k) - 1 \leq 2(i - k) - 1$. This completes the proof of Theorem 3.3.

Derivation of equation (22)

As defined in Chapter 3.3, the running packet loss probability of $f_{i,j}$, namely, $P_{i,j}[n]$, can be written as

$$P_{i,j}[n] = \frac{L_{i,j}[n-1] + l_{i,j}[n]}{P_i \cdot A_{i,j}[n]}$$

After substituting the above equation into equation (17), we get

$$\frac{L_{i,j}[n-1] + l_{i,j}[n]}{P_i \cdot A_{i,j}[n]} = \frac{L_{r,s}[n-1] + l_{r,s}[n]}{P_r \cdot A_{r,s}[n]},$$

which implies

$$l_{r,s}[n] = -L_{r,s}[n-1] + \left(\frac{P_r \cdot A_{r,s}[n]}{P_i \cdot A_{i,j}[n]} \right) (L_{i,j}[n-1] + l_{i,j}[n]).$$

Summing over all $(r,s) \in U_{active}$ except for $(r,s) = (i,j)$, we have

$$\sum_{(r,s) \neq (i,j), (r,s) \in U_{active}} l_{r,s}[n] = \sum_{(r,s) \neq (i,j), (r,s) \in U_{active}} \left[-L_{r,s}[n-1] + \left(\frac{P_r \cdot A_{r,s}[n]}{P_i \cdot A_{i,j}[n]} \right) (L_{i,j}[n-1] + l_{i,j}[n]) \right].$$

According to equation (18), it holds that

$$Loss[n] - l_{i,j}[n] = \sum_{(r,s) \neq (i,j), (r,s) \in U_{active}} \left[-L_{r,s}[n-1] + \left(\frac{P_r \cdot A_{r,s}[n]}{P_i \cdot A_{i,j}[n]} \right) (L_{i,j}[n-1] + l_{i,j}[n]) \right].$$

After some manipulations, we get

$$l_{i,j}[n] = \frac{1}{\sum_{(r,s) \in U_{active}} P_r \cdot A_{r,s}[n]} \cdot \left[P_i \cdot A_{i,j}[n] \cdot \left(Loss[n] + \sum_{(r,s) \neq (i,j), (r,s) \in U_{active}} L_{r,s}[n-1] \right) - L_{i,j}[n-1] \cdot \left(\sum_{(r,s) \neq (i,j), (r,s) \in U_{active}} P_r \cdot A_{r,s}[n] \right) \right]$$

Proof of Lemma 4.1

Lemma 4.1 is obviously true for $P_{n,k}^{\min}[t] \leq P_{n,k} \leq P_{n,k}^{\max}[t]$ because, in this case, we have

$P_{n,k}^* [t] - P_{n,k} = 0$. For $P_{n,k} > P_{n,k}^{\max}[t]$, it holds that

$$|P_{n,k}^* [t] - P_{n,k}| = P_{n,k} - \frac{L_{n,k}[t-1] + Q_{n,k}^1[t]}{S_{n,k}[t-1] + L_{n,k}[t-1] + Q_{n,k}^1[t]} \leq P_{n,k} - \frac{L_{n,k}[t-1] + (Q_{n,k}^1[t] - R_{n,k}[t])^+}{S_{n,k}[t-1] + L_{n,k}[t-1] + \max(R_{n,k}[t], Q_{n,k}^1[t])}$$

since $R_{n,k}[t] \geq 0$. Therefore, Lemma 4.1 is true for $P_{n,k} > P_{n,k}^{\max}[t]$. For $P_{n,k} < P_{n,k}^{\min}[t]$, we have

$$|P_{n,k}^* [t] - P_{n,k}| = \frac{L_{n,k}[t-1]}{S_{n,k}[t-1] + L_{n,k}[t-1] + Q_{n,k}^1[t]} - P_{n,k} \leq \frac{L_{n,k}[t-1] + (Q_{n,k}^1[t] - R_{n,k}[t])^+}{S_{n,k}[t-1] + L_{n,k}[t-1] + \max(R_{n,k}[t], Q_{n,k}^1[t])} - P_{n,k}$$

since $R_{n,k}[t] \leq Q_{n,k}^1[t]$. This completes the proof of Lemma 4.1.

Proof of Lemma 4.2

Let $R_{n,k}[t]$ and $P_{n,k}[t]$ be, respectively, the bandwidth allocated to and the resulting running loss probability of $f_{n,k}$ under our proposed PL scheduler. Further, let $R'_{n,k}[t]$ and $P'_{n,k}[t]$ be the same variables under some other scheduler. Assume that $\phi = \arg \max_{1 \leq k \leq K_n} P_{n,k}[t]/P_{n,k}$. We shall prove $P_{n,\phi}[t]/P_{n,\phi} \leq \max_{1 \leq k \leq K_n} P'_{n,k}[t]/P_{n,k}$.

Let U_Z , U_P , and U_A be the three sets such that flow $f_{n,k}$ is contained in U_Z , U_P , or U_A iff $R_{n,k}[t] = 0$, $0 < R_{n,k}[t] < Q_{n,k}[t]$, or $R_{n,k}[t] = Q_{n,k}[t]$, under the proposed PL scheduler. Assume that $U_A = \emptyset$. Since $R_n[t] > 0$, it must hold that $\phi \in U_P$. If $P_{n,\phi}[t]/P_{n,\phi} > P'_{n,\phi}[t]/P_{n,\phi}$, meaning that $R_{n,\phi}[t] < R'_{n,\phi}[t]$, there must exist $f_{n,k} \in U_P$ such that $R_{n,k}[t] > R'_{n,k}[t]$. Otherwise, equation (33) is violated. Since $P'_{n,k}[t]/P_{n,k} > P_{n,k}[t]/P_{n,k} = P_{n,\phi}[t]/P_{n,\phi}$, Lemma 4.2 is true for this case. Consider the case $U_A \neq \emptyset$. The proposed PL scheduler allocates $R_{n,i}[t] = Q_{n,i}[t]$ to all $f_{n,i} \in U_A$, which implies $f_{n,\phi}$ is in U_A or can be selected from U_A , according to equation (32). Consequently, Lemma 4.2 is true because $R_{n,\phi}[t] \geq R'_{n,\phi}[t]$, which implies $P_{n,\phi}[t]/P_{n,\phi} \leq P'_{n,\phi}[t]/P_{n,\phi}$.

Proof of Theorem 4.3

Assume that there exists a scheduler which can guarantee the loss probability requirements of all the K_n traffic flows. In other words, it holds that $P'_{n,k}[t]/P_{n,k} \leq 1$, $1 \leq k \leq K_n$, where $P'_{n,k}[t]$ is the loss probability of flow $f_{n,k}$ at the end of the t^{th} frame, under the considered scheduler. Let

$P_{n,k}[t]$ be the loss probability of flow $f_{n,k}$ at the end of the t^{th} frame, under the PL scheduler.

According to Lemma 4.2, we have $P_{n,k}[t]/P_{n,k} \leq \max_{1 \leq i \leq K_n} P'_{n,i}[t]/P_{n,i} \leq 1$, $1 \leq k \leq K_n$, and, therefore,

Theorem 4.3 is true.

Proof of Lemma 4.4

Lemma 4.4 can be easily verified with the calculation results shown in Table 4.1.

Proof of Theorem 4.5

We prove Theorem 4.5 for $\Delta R_n[t] \geq 0$. The other case can be proved similarly. Let V_Z , V_P and V_A be three sets such that $f_{n,k}$ is in V_Z , V_P , or V_A iff $R_{n,k}^*[t] = 0$, $0 < R_{n,k}^*[t] < Q_{n,k}[t]$, or $R_{n,k}^*[t] = Q_{n,k}[t]$, respectively. Similarly, $f_{n,k}$ is in U_Z , U_P , or U_A iff $R_{n,k}[t] = 0$, $0 < R_{n,k}[t] < Q_{n,k}[t]$, or $R_{n,k}[t] = Q_{n,k}[t]$, respectively. Recall that equations (32) and (33) are satisfied under the PL scheduler.

Assume that $\Delta R_{n,i}[t] < 0$ for some flow $f_{n,i}$. Since $\Delta R_n[t] \geq 0$, there must be some other $f_{n,j}$ with $\Delta R_{n,j}[t] > 0$. The assumption $\Delta R_{n,i}[t] < 0$ implies $f_{n,i} \in V_P \cup V_A$ and $\Delta R_{n,j}[t] > 0$ implies $f_{n,j} \in V_Z \cup V_P$. From Lemma 4, we have $P_{n,i}^*[t]/P_{n,i} \geq P_{n,j}^*[t]/P_{n,j}$. The assumption $\Delta R_{n,i}[t] < 0$ also implies $f_{n,i} \in U_Z \cup U_P$ and $\Delta R_{n,j}[t] > 0$ implies $f_{n,j} \in U_P \cup U_A$. According to equation (32), we have $P_{n,i}[t]/P_{n,i} \leq P_{n,j}[t]/P_{n,j}$, a contradiction, because $P_{n,k}[t]$ is a strictly decreasing function of $R_{n,k}[t]$ for $0 \leq R_{n,k}[t] \leq Q_{n,k}[t]$, which together with $P_{n,i}^*[t]/P_{n,i} \geq P_{n,j}^*[t]/P_{n,j}$, $\Delta R_{n,i}[t] < 0$, and $\Delta R_{n,j}[t] > 0$ imply $P_{n,i}[t]/P_{n,i} > P_{n,j}[t]/P_{n,j}$. This

proves Theorem 4.5.

Derivation of $P_n^F[t]$

Given $P_n^F[t]$, one can compute $h_{n,k}(P_n^F[t] \cdot P_{n,k}; t)$ based on equation (34) for any $f_{n,k} \in U_{P_1} \cup U_{P_2}$. Substituting $h_{n,k}(P_n^F[t] \cdot P_{n,k}; t)$ into $\sum_{f_{n,k} \in U_{P_1} \cup U_{P_2}} h_{n,k}(P_n^F[t] \cdot P_{n,k}; t) = R_n[t] - \sum_{f_{n,k} \in U_A} Q_{n,k}[t]$, we get $A \cdot (P_n^F[t])^2 + B \cdot (P_n^F[t]) + C = 0$, where

$$A = \sum_{f_{n,k} \in U_{P_1}} P_{n,k} \cdot (S_{n,k}[t-1] + L_{n,k}[t-1] + Q_{n,k}^1[t]),$$

$$B = R_n[t] - \sum_{f_{n,k} \in U_A} Q_{n,k}[t] + \sum_{f_{n,k} \in U_{P_2}} (S_{n,k}[t-1] + L_{n,k}[t-1]) - \sum_{f_{n,k} \in U_{P_1}} (L_{n,k}[t-1] + Q_{n,k}^1[t])$$

and

$$C = -\sum_{f_{n,k} \in U_{P_2}} (1/P_{n,k}) L_{n,k}[t-1].$$

If $U_{P_1} = \emptyset$, which implies $A = 0$, $P_n^F[t]$ can be obtained by $P_n^F[t] = -C/B$. Assume that $A \neq 0$. In this case, we have $P_n^F[t] = (-B + \sqrt{B^2 - 4AC})/(2A)$ because $B^2 - 4AC \geq B^2$ and $P_n^F[t]$ must be non-negative.

Appendix B

Pseudo codes of the proposed algorithms

- Loss computation of the proportional-loss service scheduler

Algorithm: Loss computation

Initialization

1. $U_{temp} = U_{active}$
2. $Loss_{temp} = Loss[n]$
3. $Flag = 0$

Begin

4. $[l_{i,j}[n], \forall (i, j) \in U_{active}]_{1 \times |U_{active}|} = LossComputation(Loss_{temp}, U_{temp})$

End**/*Loss computation module*/**

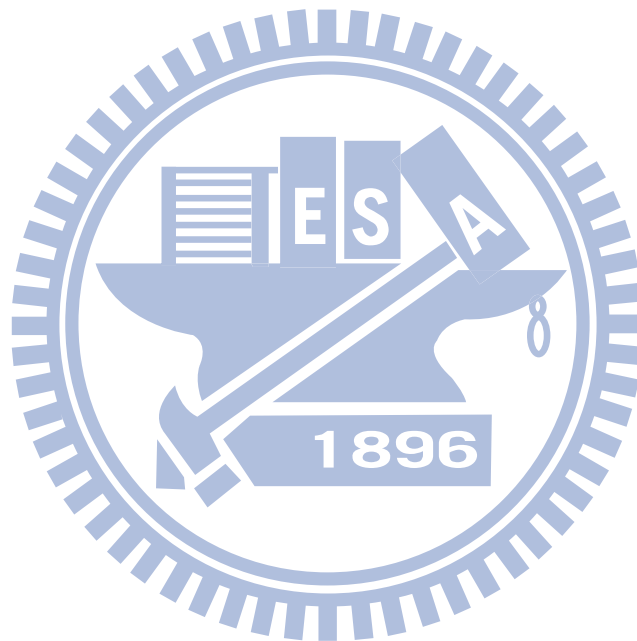
1. $LossComputation(Loss, U)$
 2. $WeightedLossCalculation(Loss, U)$ /*Compute $l_{i,j}[n]$ with eqn. (22)*/
 3. **if** $0 \leq l_{i,j}[n] \leq Q_{i,j}^m[n] \forall (i, j) \in U$ /*Case 1*/
 4. **exit**
 5. **elseif** $0 \leq l_{i,j}[n] \forall (i, j) \in U$ and $\exists (i, j) \in U, s.t. l_{i,j}[n] > Q_{i,j}^m[n]$ /*Case 2*/
 6. **for all** $(i, j) \in U$
 7. **if** $l_{i,j}[n] \geq Q_{i,j}^m[n]$
 8. $l_{i,j}[n] = Q_{i,j}^m[n]$
 9. $U = U - \{(i, j)\}$
-

```

10.          $Loss = Loss - l_{i,j}[n]$ 
11.     end if
12. end for
13. if  $Flag = 1$ 
14.      $Flag = 0$ 
15.     exit
16. else
17.      $LossComputation(Loss, U)$ 
18. end if
19. elseif  $l_{i,j}[n] \leq Q_{i,j}^m[n] \forall (i, j) \in U$  and  $\exists (i, j) \in U, s.t. l_{i,j}[n] < 0$  /*Case 3*/
20.     for all  $(i, j) \in U$ 
21.         if  $l_{i,j}[n] \leq 0$ 
22.              $l_{i,j}[n] = 0$ 
23.              $U = U - \{(i, j)\}$ 
24.         end if
25.     end for
26.      $LossComputation(Loss, U)$ 
27. else /*Case 4:  $\exists (i, j)$  and  $(r, s) \in U, s.t. l_{i,j}[n] > Q_{i,j}^m[n]$  and  $l_{r,s}[n] < 0$  */
28.      $V_1 = \{(i, j) \in U: l_{i,j}[n] \geq 0\}$ 
29.      $V_2 = U - V_1$ 
30.     if  $\sum_{(i,j) \in V_1} Q_{i,j}^m[n] < Loss[n]$  /*Sub-case 1*/
31.         for all  $(i, j) \in U$ 
32.             if  $l_{i,j}[n] \geq Q_{i,j}^m[n]$ 
33.                  $l_{i,j}[n] = Q_{i,j}^m[n]$ 
34.                  $U = U - \{(i, j)\}$ 
35.                  $Loss = Loss - l_{i,j}[n]$ 
36.             end if
37.         end for
38.          $LossComputation(Loss, U)$ 
39.     else /*Sub-case 2:  $\sum_{(i,j) \in V_1} Q_{i,j}^m[n] \geq Loss[n]$  */
40.          $Flag = 1$ 
41.          $LossComputation(Loss, V_1)$ 
42.         if  $Flag = 0$  and  $\exists (i, j) \in V_1, s.t. l_{i,j}[n] < Q_{i,j}^m[n]$ 
43.             for all  $(i, j) \in V_1$ 
44.                 if  $l_{i,j}[n] < Q_{i,j}^m[n]$ 
45.                      $V_2 = V_2 \cup \{(i, j)\}$ 
46.                 else
47.                      $Loss = Loss - Q_{i,j}^m[n]$ 

```

```
48.         end if
49.     end for
50.     LossComputation(Loss, V2)
51. else
52.     for all  $(i, j) \in V_2$ 
53.          $l_{i,j}[n] = 0$ 
54.     end for
55.     exit
56. end if
57. end if
58. end if
```



● PL scheduler

Algorithm: PL scheduler

Initialization

- 1 $U_Z = \{f_{n,k} : R_{n,k}^* [t] = 0\}$
- 2 $U_{P1} = \{f_{n,k} : 0 < R_{n,k}^* [t] \leq Q_{n,k}^1 [t]\}$
- 3 $U_{P2} = \{f_{n,k} : Q_{n,k}^1 [t] < R_{n,k}^* [t] < Q_{n,k} [t]\}$
- 4 $U_A = \{f_{n,k} : R_{n,k}^* [t] = Q_{n,k} [t]\}$

Begin

- 1 **If** $R_n [t] = R_n^* [t]$
 - 2 $R_{n,k} [t] = R_{n,k}^* [t], 1 \leq k \leq K_n$
 - 3 **elseif** $R_n [t] > R_n^* [t]$
 - 4 **while** (1)
 - 5
$$p = \max \left(\max_{f_{n,k} \in U_Z} \frac{P_{n,k}^{\max} [t]}{P_{n,k}}, \max_{f_{n,k} \in U_{P1}} \frac{P_{n,k}^{\text{knee}} [t]}{P_{n,k}}, \max_{f_{n,k} \in U_{P2}} \frac{P_{n,k}^{\min} [t]}{P_{n,k}} \right)$$
 - 6 **If** $\sum_{f_{n,k} \in U_{P1} \cup U_{P2}} h(p \cdot P_{n,k}; t) + \sum_{f_{n,k} \in U_A} Q_{n,k} [t] \geq R_n [t]$
 - 7 $R_{n,k} [t] = 0$ **for all** $f_{n,k} \in U_Z$
 - 8 $R_{n,k} [t] = Q_{n,k} [t]$ **for all** $f_{n,k} \in U_A$
 - 9 $R_{n,k} [t] = h_{n,k} (P_n^F [t] \cdot P_{n,k}; t)$ **for** $f_{n,k} \in U_{P1} \cup U_{P2}$
 - 10 (Flow $f_{n,k}$ is moved from U_{P2} to U_A if $R_{n,k} [t] = Q_{n,k} [t]$.)
 - 11 **exit**
 - 12 **else**
 - 13
$$k^* = \arg \max_{f_{n,k} \in U_Z \cup U_{P1} \cup U_{P2}} \left(\max_{f_{n,k} \in U_Z} \frac{P_{n,k}^{\max} [t]}{P_{n,k}}, \max_{f_{n,k} \in U_{P1}} \frac{P_{n,k}^{\text{knee}} [t]}{P_{n,k}}, \max_{f_{n,k} \in U_{P2}} \frac{P_{n,k}^{\min} [t]}{P_{n,k}} \right)$$
 - 14 **If** $f_{n,k^*} \in U_Z$
 - 15 $U_Z = U_Z - f_{n,k^*}$
 - 16 $U_{P1} = U_{P1} \cup f_{n,k^*}$
 - 17 **elseif** $f_{n,k^*} \in U_{P1}$
 - 18 $U_{P1} = U_{P1} - f_{n,k^*}$
 - 19 $U_{P2} = U_{P2} \cup f_{n,k^*}$
 - 20 **else**
 - 21 $U_{P2} = U_{P2} - f_{n,k^*}$
 - 22 $U_A = U_A \cup f_{n,k^*}$
 - 23 **endif**
 - 24 **endif**
-

```

25   endwhile
26   else
27   while(1)
28       
$$p = \min \left( \min_{f_{n,k} \in U_{P1}} \frac{P_{n,k}^{\max} [t]}{P_{n,k}}, \min_{f_{n,k} \in U_{P2}} \frac{P_{n,k}^{\text{knee}} [t]}{P_{n,k}}, \min_{f_{n,k} \in U_A} \frac{P_{n,k}^{\min} [t]}{P_{n,k}} \right)$$

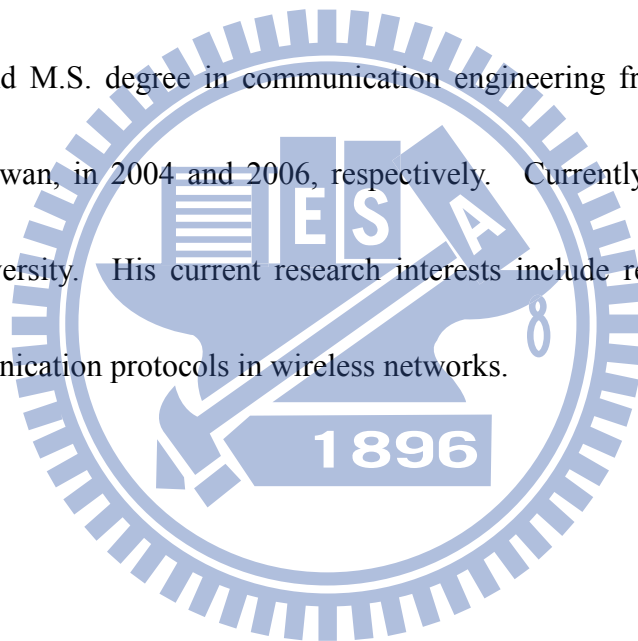
29   If  $\sum_{f_{n,k} \in U_{P1} \cup U_{P2}} h(p \cdot P_{n,k}; t) + \sum_{f_{n,k} \in U_A} Q_{n,k} [t] \leq R_n [t]$ 
30        $R_{n,k} [t] = 0$  for all  $f_{n,k} \in U_Z$ 
31        $R_{n,k} [t] = Q_{n,k} [t]$  for all  $f_{n,k} \in U_A$ 
32        $R_{n,k} [t] = h_{n,k} (P_n^F [t] \cdot P_{n,k}; t)$  for  $f_{n,k} \in U_{P1} \cup U_{P2}$ 
33       (Flow  $f_{n,k}$  is moved from  $U_{P2}$  to  $U_{P1}$  if  $R_{n,k} [t] = Q_{n,k}^1 [t]$  or from  $U_{P1}$ 
to  $U_Z$  if  $R_{n,k} [t] = 0$ .)
34   exit
35   else
36       
$$k^* = \arg \min_{f_{n,k} \in U_{P1} \cup U_{P2} \cup U_A} \left( \min_{f_{n,k} \in U_{P1}} \frac{P_{n,k}^{\max} [t]}{P_{n,k}}, \min_{f_{n,k} \in U_{P2}} \frac{P_{n,k}^{\text{knee}} [t]}{P_{n,k}}, \min_{f_{n,k} \in U_A} \frac{P_{n,k}^{\min} [t]}{P_{n,k}} \right)$$

37   If  $f_{n,k^*} \in U_{P1}$ 
38        $U_{P1} = U_{P1} - f_{n,k^*}$ 
39        $U_Z = U_Z \cup f_{n,k^*}$ 
40   elseif  $f_{n,k^*} \in U_{P2}$ 
41        $U_{P2} = U_{P2} - f_{n,k^*}$ 
42        $U_{P1} = U_{P1} \cup f_{n,k^*}$ 
43   else
44        $U_A = U_A - f_{n,k^*}$ 
45        $U_{P2} = U_{P2} \cup f_{n,k^*}$ 
46   endif
47   endif
48   endwhile
49   endif

```

Vita

Yu-Wen Huang (黃郁文) was born in Gangshan District, Kaohsiung City, Taiwan, in 1982. He received the B.S. and M.S. degree in communication engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2004 and 2006, respectively. Currently, he is pursuing his Ph.D. degree at the same university. His current research interests include resource allocation, power management and communication protocols in wireless networks.



Publication List

Journal Paper (Published or accept)

- [J1] T. H. Lee and **Y. W. Huang**, "Effective Transmission Opportunity Allocation Scheme for Real-Time VBR Traffic Flows with Different Delay Bounds," *IET Commun.*, 2008, Vol. 2, No. 4, pp.598-608
- [J2] T. H. Lee and **Y. W. Huang**, "Resource Allocation Achieving High System Throughput with QoS Support in OFDMA-based Systems," *IEEE Trans. on Commun.* (accepted on Nov. 4, 2011)
- [J3] T. H. Lee and **Y. W. Huang**, "Quality of Service Guarantee for VBR Traffic Flows with Different Delay Bound and Loss Probability in WLANs", *Journal of the Chinese Institute of Engineers*. (accepted on Dec. 31, 2011)
- [J4] T. H. Lee and **Y. W. Huang**, Chien-Nan Chen" Optimal Resource Allocation for Increasing Strictly Concave Utility Functions in Wireless Networks ", *IEEE Trans. on Vehicular. Tech.* (accepted on Dec. 28, 2011)

Journal Paper (To be submitted)

- [J5] **Y. W. Huang** and T. H. Lee, "An Optimal Queue Management Algorithm for Real-Time Traffic }} with Different Delay Bound and Loss Probability Requirements," to be submitted to *IEEE Trans. on Commun.*.

Conference Paper (Published or accept)

- [C1] **Y.W. Huang**, T. H. Lee and J. R. Hsieh, "Gaussian Approximation Based Admission Control for Variable Bit Rate Traffic in IEEE 802.11e WLANs ," in *Proc . IEEE WCNC 2007*, Hong Kong
- [C2] T. H. Lee, J. R. Hsieh, M. C. Huang, and **Y. W. Huang**, " A Bandwidth Efficient Pairing Strategy for the MIMO-OFDM Based WLANs," in *Proc. IEEE VTC-Spring 2009*, Barcelona.
- [C3] T. H. Lee, **Y. W. Huang** and C. C. Yang , " A Low Overhead Queue Status Report Scheme for QoS Support in WLANs,"in *Proc. IEEE TENCON 2009*, Singapore.
- [C4] Y.W. Kuo, T. H. Lee, **Y. W. Huang** and J. R. Hsieh,"Design and Evaluation of a High Throughput MAC with QoS Guaranteefor Wireless LANs," in *Proc. IEEE MICC 2009*, Kuala Lumpur.
- [C5] T. H. Lee, Y W. Kuo, **Y. W. Huang** and Y. H. Liu " To Piggyback Or Not to Piggyback Acknowledgments ? ," in *Proc. IEEE VTC 2010-Spring*, Taipei.