國 立 清 華 大 學
碩 士 論 文

題目　在無線網路中提供服務品質保證的彈性化封包排程機制

Flexible Packet Scheduling for Quality of Service Provisioning in Wireless Networks

系 所 別　通訊工程研究所碩士班
學號姓名　895634 **施坤成** (Kun-Cheng Shih)
指導教授　**陳文村 教授** (Prof. Wen-Tsuen Chen)

中華民國 九十一 年 七 月

摘要

本論文提出一個無線網路的封包排程方法，讓網路服務經營商(network operator)能夠更有彈性地分配網路資源。該方法稱為可變式服務曲線(Adaptive Service Curve, ASC)。除了改進效能外，我們特別專注於提出一個嶄新且完整的架構，希望能為這個領域開創新方向。

在此論文中，我們首先定義出一個好的排程者(scheduler)需要提供三種彈性，然後說明 ASC 如何達到這個要求。第一種彈性，在容易發生傳輸錯誤的無線環境中，ASC 能提供傳輸錯誤時反應方式(error resilience)的差別化(differentiation)。更詳細地說，當傳輸錯誤發生時，ASC 能讓系統選擇是否繼續傳輸，花多少的資源去傳輸；第二種彈性是以上的反應方式能針對個別使用者量身定做；第三種彈性是在資源分配上，我們提出一個有效的方法讓傳統的服務曲線(service curve)可以在無線環境中正常運作，而讓 ASC 具有避免過度配置資源(over-allocation)的情形發生。

在技術面，ASC 所提出的架構融合了傳輸調變(link adaptation)、彈性化(flexibility)、有效的服務定義模式(traffic characterization model)。此論文的結果顯示 ASC 能提供無線通訊資源管理完整的解決方案。

誌謝辭

感謝指導教授陳文村院長兩年來的指導與教誨，並提供一個良好的研究環境。在院長的指導下，我學會了很多做研究的方法。另外，特別感謝院長對我身體情況的關心，在此獻上最崇高的敬意。

感謝中央大學許健平教授及暨南大學陳彥錚教授在口試時的指導，讓我有機會再一次省思自已的研究，找出思考的盲點。

感謝師大資訊蔡榮宗教授五年來的指導，包括通訊網路領域相關課程、網路專題研究和生涯規劃。

感謝實驗室的文宗學長、仁筑學長、政龍學長、一致學長、均家學長、博彬、有明、建良、世賢、松伯、孟鉉、瑞怡、怡蓓和其它學弟妹，感謝師大資訊的益豪、克奇、至德、坤達、柏青學長、明鈿、麗玉、天佐、聖彥、允聖、勇任、泊澄、盟淵、健智等陪我在資訊這領域學習，還有好友木興、建成、惠如、怡正、玉倩、似芳，感謝他們在學業與生活上的扶持砥礪，幫助我渡過挫折和低潮。

最後，謹將此論文獻給我的父母和姐姐，感謝他們一直以來的支持與鼓勵，讓我可以專心無慮的完成學業。

目錄

第一章　簡介

在無線通訊中，提供服務品質保證（QoS）是個非常重要的課題。而要滿足這樣的要求，發展一個支援服務品質差別化和保證的封包排程器是其中很關鍵性的一項工作。相對於有線的環境，無線環境中的高傳輸錯誤率讓封包排程的工作更加困難。

在此論文中，我們首先定義出一個好的排程器(scheduler)需要提供三度空間的彈性，然後說明 ASC 如何達到這個要求。第一度彈性，在容易發生傳輸錯誤的無線環境中，ASC 能提供傳輸錯誤時反應方式(error resilience)的差別化(differentiation)。更詳細地說，當傳輸錯誤發生時，ASC 能讓系統選擇是否繼續傳輸，花多少的資源去傳輸；第二度彈性是以上的反應方式能針對個別使用者量身定做；第三度彈性是在資源分配上的彈性。而讓 ASC 具有避免過度配置資源(over-allocation)的情形發生。為了達到上述的目標，我們提出一個嶄新的排程演算法，稱為可變式服務曲線 (Adaptive Service Curve, ASC)。它利用傳統的服務曲線(Service Curve)，我們提出一個有效的方法讓傳統的服務曲線可以在無線環境中正常運作，而繼承所有服務曲線的優點。另外，為了達到 resilience differentiation，我們讓 ASC 能夠配合傳輸調變（link adaptation）的技術。

第二章 背景知識

本章，我們將描述本論文所適用的環境和一些系統假設。然後我們深入介紹所有解決的問題。最後，簡介服務曲線（service curve）的技術。不同於有線網路，無線網路有兩個獨特的問題：1）突然大量（bursty）的傳輸錯誤；2）隨著地點而不同（location-dependent）的頻寬和傳輸品質。這獨特的問題造成一些獨特的現象，例如，即使伺服器花費相同的傳輸時間和頻寬，但使用者很有可能接受不同的待遇，因為不同的使用者有不同的傳輸品質，這樣就會產生不公平。另外，由系統的角度來看，多花時間在傳輸狀況差的使用者，系統的效能就降低。由這兩點，就可看出的確需要小心地設計系統，以提供服務品質保證。

簡單地說，服務曲線技術就是每個使用者有專屬的一條服務曲線，描述著所有傳輸的要求。系統根據這條服務曲線來服務這位使用者。一般來說，服務曲線技術具有四大優點：1)頻寬和延遲配置的分別化(decoupling)；2)有效地傳輸非即時性的資料（bursty best-effort traffic)；3)機制和決策的區分（the separation of mechanisms and policies)； 4)它是一種系統化的方法（the systematization)。

第三章 所提出的方法

在此章中，我們首先介紹公平化服務曲線（Fair Service Curve）技術，這是我們所選擇原本有線端的服務曲線方法。接下來，我們正式提出我們的解決方法，並接著做一個討論和比較。

我們的方法主要是定義出每個使用者所屬的服務曲線要改變（adaptation）的時機（timing）和如何改變。仔細地說，我們定義一種錯誤反應曲線（resilience curve），做為服務曲線要改變的依據；而在改變的時機方面，我們定義一個參數稱為調變週期（adaptation period)，讓改變的時間能夠在合理的控制下。另外，我們也提出一種簡單的方法讓排程器能和傳輸調變（link adaptation）能夠合作。

接著，我們提出一些選擇錯誤反應曲線的選擇考量。

第四章 模擬實驗與結果

我們使用 OPNET 軟體模擬此篇論文所提出的環境和解決方法。因為本篇論文著重在提出一個原創性的完整架構，所以模擬的重點在於此架構的運作情形和預期的效果展示，在此並不針對細項的效率和其它方法比較。

第五章 結論與未來方向

我們提出一個無線網路的封包排程方法，讓網路服務經營商(network operator)能夠更有彈性地分配網路資源。該方法稱為可變式服務曲線(Adaptive Service Curve, ASC)。除了改進效能外，我們特別專注於提出一個嶄新且完整的架構，希望能為這個領域開創新方向。

我們提出一個原創性的問題：一個好的排程者(scheduler)需要提供三度空間的彈性。接著設計出解決此問題的 ASC 排程器。

ASC 所提出的架構融合了傳輸調變(link adaptation)、彈性化(flexibility)、有效的服務定義模式(traffic characterization model)。此論文的結果顯示 ASC 能提供無線通訊資源管理完整的解決方案。

英　文　附　錄

Flexible Packet Scheduling for Quality of Service Provisioning in Wireless Networks

Student: Kun-Cheng Shih

Advisor: Prof. Wen-Tsuen Chen

July 2002

A Thesis Submitted to the
Institute of Communications Engineering
National Tsing Hua University
Hsinchu, Taiwan 30043
Republic of China

In Partial Fulfillment of Requirements
for the Degree of Master
in Communications Engineering

Abstract

This thesis presents a novel wireless scheduling mechanism, called Adaptive Service Curve (ASC). The proposed mechanism increases the flexibility with which network operators can adjust the allocation of resources. We first point out that a good scheduler should exhibit three kinds of flexibility, and then we show the ASC can achieve this goal. First, ASC should be able to *differentiate* the error resilience requirements due to the impact of location-dependent channel errors in wireless networks. Specifically, ASC can employ link adaptation, enabling a choice to be made between maximizing system throughput and making/wasting more link effort on error-prone channels. Second, the above flexibility is not system-wide. Rather, users can subscribe to different error resilience requirements. ASC utilizes the service curve model that can *best meet* QoS requirements to provide the third kind of flexibility, and prevents an unacceptable *over-allocation* of scarce radio resources. Accordingly, a framework is proposed to make the existing service curve model operate effectively in wireless environments. The ASC scheduler combines in a single-framework the three aspects of scheduling, namely, link adaptation, flexibility and a mature traffic characterization model. This design represents a complete solution for wireless resource management.

Acknowledgements

I am extremely grateful to Prof. Wen-Tsuen Chen, my thesis advisor, for giving me continuous encouragement, support, and technical guidance during my graduate studies.

I am grateful to the members of my thesis committee, Prof. Jang-Ping Sheu And Prof. Yen-Cheng Chen. Their engaging arguments and strong feedback have contributed greatly to this thesis.

I really appreciate Prof. Jung-Tsung Tsai that he have invested enormous amount of time and energy on me over the last five years, including giving me solid course training, helping me start the research career, and solving the technical issues for me.

I am grateful to all the friends and colleagues with whom I spent my time as a graduate student at National Tsing Hua University and as an undergraduate student at National Taiwan Normal University. Special thanks are given to Bo-Bin Jian, Yi-How Lee, Ke-Chi Chen and Po-Ching Lin for their help and patience with my questions about research, careers and life.

Finally, I would like to express my earnest gratitude to my parents and my sister for their unconditional love and support, without which any of my achievements would not have been possible.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis proposes an enhanced framework for specifying and providing Quality of Service (QoS) guarantees in packet-switched wireless networks. In this chapter, we present the motivation of this thesis; summarize the contributions; review the literature and finally state the organization of this thesis.

1.1. Motivation and Contributions

A key problem in the emergence of integrated-service wireless networks concerns the arbitration among various traffic sources with different Quality of Service (QoS) requirements over shared channels. A critical step in solving this problem is to develop a wireless packet scheduler that provides QoS differentiation and guarantees. The difficulty is exacerbated by typical wireless channels suffering from severe multipath fading and interference distortion, such that channel errors are bursty, *time-dependent*, and *location-dependent* [4][11].

Wireless packet scheduling has been the subject of intensive study [4][5][6][7][11] [14][16][17][18][25]. Most of their scheduling mechanisms characterize the service in prose rather than in a systematic model, which causes difficulty in meeting multiple QoS requirements quantitatively at different granularities [26][28].

The main contribution of this thesis is to provide a wireless scheduler with three kinds of flexibility, as elaborated in Section 2.2. The first kind of flexibility [17] enables network operators to adjust the behavior in response to channel errors, which is to differentiate error resilience. The second kind of flexibility is that the above flexibility is not system-wide. Rather, users can subscribe to different error resilience requirements [7]. Finally, as the third kind of flexibility, the allocation of scarce radio resources is made as flexible as possible. Basically, the allocation of delay should be decoupled from that of bandwidth. Most WFQ-based scheduling algorithms suffer from the coupling problem [28]. To our knowledge, existing wireless scheduling algorithms do not address these three kinds of flexibility simultaneously.

Accordingly, this thesis presents a novel scheduling algorithm called *Adaptive Service Curve* (ASC). The proposed algorithm utilizes the service curve model [10] to characterize traffic effectively and efficiently. The service curve model can *best match* QoS requirements and prevents unacceptable *over-allocation* of network resources [26], which is especially crucial to sharing scarce radio resources. Consequently, this thesis also proposes a framework to make the existing service curve model operate effectively in wireless environments. Specifically, a *curve adaptation* technique is proposed to tackle location-dependent channel errors that result in a failure to adopt the traditional wireline schedulers in wireless environments. ASC explicitly defines both the behavior and the timing of the adaptation of service curve in response to channel errors, subject to error resilience requirements. The proposed curve adaptation model provides a steppingstone to *resilience differentiation* for wireless services. ASC is made to comply with link-adaptation to realize resilience differentiation; serve traffic with various characteristics, and increase system throughput.

In order to provide a complete solution for wireless resource management, the ASC scheduler combines in a single-framework the three aspects of scheduling, namely link adaptation [16], flexibility [17] and a mature traffic characterization model [10].

1.2. Related Work

In this section, we provide an overview of several recently proposed scheduling schemes that provide QoS guarantees in the context of packet-switched wireless networks. One of the first papers that address the problem of location-dependent and bursty errors in wireless scheduling is [4].

The fairness issue has attracted a lot of research attention in wireless scheduling. For example, Idealized Wireless Fair Queueing (IWFQ) [18], Channel-condition Independent Fair Queuing (CIF-Q) [11], and Server Based Fairness Approach (SBFA) [25], where an ideal error-free reference model is presented and their common goal is to keep the service share of each user in wireless system as close as possible to the error-free model. The error-free model that is usually adapted is WFQ based policies, such PGPS [21]. However, such error-free reference model causes relatively greater computing complexity [5] because there are *two* counting systems maintained at run time. Our design of ASC tries to avoid this overhead, i.e., we maintain only one system at run time.

Using service curve in wireless environments is not new. A pioneered work, Packetized

Service Curve Processor Sharing (PSCPS), is proposed in [9]. It focuses on operating with a time-varying channel capacity while the service curve guarantees are still maintained. However, The PSCPS does not address the issue of location-dependent channel errors. Our additional contributions are to suggest the use of ASC in the context of the wireless networks with location-dependent channel errors and to coordinate the link adaptation and packet layer scheduling in order to serve traffic of variant characteristics effectively and increase system throughput.

The comparisons between related work and our scheme are presented in the following context. We refer readers to [6] and [5], where surveys and classification of the literature are presented.

1.3. Organization

The rest of this thesis is organized as follows. Chapter 2 describes system architecture and related background knowledge. Chapter 3 proposes the ASC scheduling algorithm and discusses the tradeoffs and implementation issues. Chapter 4 presents simulation results, while conclusions are finally drawn in Chapter 5.

Chapter 2

Background Information

This chapter describes the system model and assumptions. The problems of relevant interest are then stated, including those concerning location-dependent channel errors, utilizing link adaptation, and characterizing the service using an effective and efficient model. Finally, the service curve model is introduced, and its special usefulness for wireless scheduling is explained.

2.1 System Environments

Figure 1 illustrates a general network architecture used in this thesis. This thesis considers a wireless cellular network model in which each cell contains a base station (BS) is connected to the wireline networks. Packet scheduling is performed centrally at BS and we focus on the downlink performance. Medium Access Control is integrated with packet scheduling. Traffic is viewed as a set of sessions and multiple sessions (such as, one video session and one FTP session) on an MS are allowed. The scheduler is assumed to have perfect knowledge of the current states of the channels without considering power control, mobility of MSs or handoff.

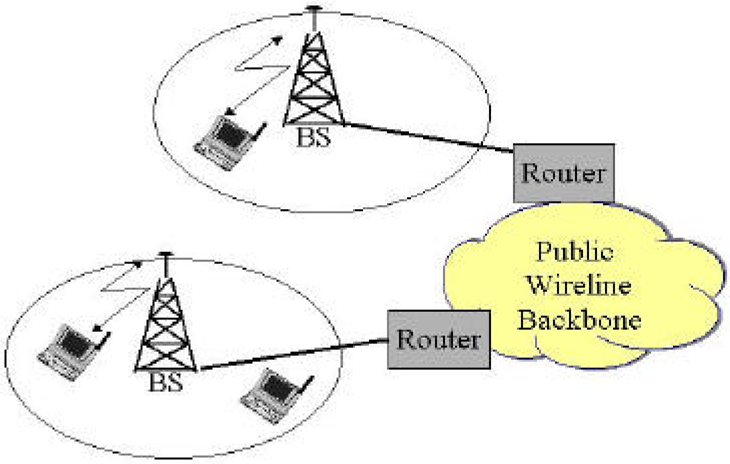

Figure 1: Network architecture

2.2. Problem Statement

In this section, we first explore the complicated relations between channel quality, fairness, and system throughput. Then we point out an operation challenge that a good scheduling algorithm should provide network operators the flexibility of adjusting the way resources are allocated for each individual session. Finally, we explain why service curve model is especially useful to wireless scheduling.

2.2.1. Location-Dependent Channel Errors

TABLE 1: The impact of location-dependent channel errors

Session	Error Rate	Request Rate(kbps)	Scheduler1 (kbps)	Scheduler2 (kbps)	Scheduler3 (kbps)
Audio1	0%	100	100	67	100
FTP1	0%	400 & Greedy	650	267	400
Audio2	50%	100	50	67	50
FTP2	50%	400	200	267	200
System Throughput			100%	67%	75%

This thesis addresses two problems that involve shared wireless channels causing wireline schedulers to fail: 1) bursty channel errors and 2) location-dependent channel capacity and errors. The radio channel is said to be in error (corresponding to the "*Bad*" state) for the receiver if the transmission of the head-of-line packet to a specific receiver repeatedly fails, as may occur if the specific receiver fades away, or another transmitter causes interference at that receiver, for example, due to frequency collisions in Industrial, Scientific and Medical (ISM) band radios. Besides, such channel errors tend to depend on location [4][11][18], implying that some channels may experience errors while other channels may be error-free (corresponding to the "*Good*" state) in the same time.

An example of two MSs, one of which is error-free and the other of which experiences a 50% error rate is considered to elucidate the impact of location-dependent channel errors on fairness and system throughput. Each station has one audio session (100 Kbps) and one FTP session (400 Kbps). The WFQ server has a capacity of 1 Mbps. Table I summarizes the parameters and the results are discussed below.

First, many scheduling algorithms (for example, [4][5][11][14]) assume that transmissions are paused when their channels are bad, i.e., only the sessions with good channels can be served to reduce wastage of radio resource due to erroneous transmissions. In this case, as shown in the column "Scheduler1", the throughput of neither overall system nor good sessions degrades. However, the bad sessions are penalized. Although this approach preserves the system throughput, despite the presence of location-dependent channel errors, it works well only for non-real-time traffic. For session Audio2, transmitting at half of the rate does not make sense. Consider the second scenario for supporting real-time traffic: some other schemes (such as, [24]) compensate for errors that occur during the transmissions of packets. Restated, this kind of scheduler makes much effort to serve bad sessions. In this example, Scheduler2 allocates resources to sessions such that all sessions receive the same ratio as that of their requested rates.

Unfortunately, both schedulers appear to be unfair. On the one hand, Scheduler1 treats bad sessions unfairly, because they are charged the same fee for their services but do not operate at the same rate. On the other hand, Scheduler2 causes good sessions to suffer from performance degradation due to the channel errors of other sessions. As shown in TABLE 1, the system throughput strongly depends on how the resources are assigned in response to channel errors. Given different error rates of sessions, shifting transmission effort to bad sessions reduces the throughput of the overall system.

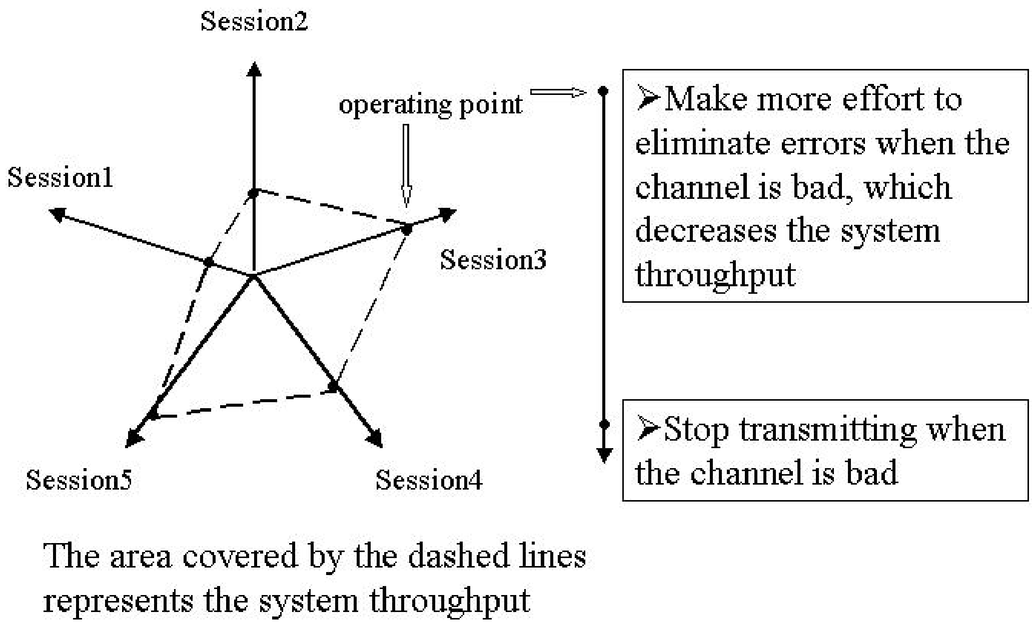

Figure 2: Two dimensional flexibility for wireless scheduling

Consequently, location-dependent channel errors give the scheduler two dimensions of policy options, as shown in Figure 2. This idea is extended from the one-dimensional scheduling axis in [17]. The first dimension of flexibility [17] is that the scheduler can be configured for various positions of the point of operation along the axis. The middle point of the axis corresponds to Schedule3 in the previous example. Moving to the center of the axis results in the operation of Scheduler1 where the session receives more effort when the channel is bad, degrading the system throughput, as in Session1. The second dimension of flexibility is that the decision concerning the location of operating point is not system-wide. Rather, it is customized to an individual session [7]. According to

Figure 2, if all sessions request the outmost operating points, the total area covered by the dashed lines would be maximized, implying that the scheduler only serves the good sessions and that the system throughput is maximized.

We believe that an effective scheduling algorithm should provide network operators the flexibility to adjust the operating point for each individual session since no single consistently optimum point is suitable for all traffic classes. Moreover, as stated in Section 2.3, by utilizing the service curve model, ASC provides the third kind of flexibility, namely decoupling the delay and bandwidth allocation. The ASC scheduler will be shown to achieve such three kinds of flexibility, representing a significantly original contribution of this thesis. To the best of our knowledge, existing wireless scheduling algorithms do not address such three kinds of flexibility.

2.2.2. Link Adaptation

Link adaptation selects appropriate *modulation* and *coding schemes* for transmission based on link-quality measurements on characteristics such as channel conditions, air link efficiency, and others [24]. For example, General Packet Radio Service (GPRS) supports four channel coding schemes, and Enhanced Data rates for GSM Evolution (EDGE) defines eight different combinations of modulation and coding schemes.

Packet/network layer retransmissions are insufficient because transmission is still required when channel errors occur. Link layer adaptation must be applied. In reality, a complete solution for wireless packet scheduling depends on coordination between the link layer adaptation and network layer compensation to support diverse QoS requirements in error-prone wireless environments [16]. Therefore, a novel contribution of this thesis is to enable the ASC scheduler to employ link adaptation.

2.2.3. An Effective and Efficient Model

The WFQ-based schemes may be suitable for backbone routers where large bandwidth is available and traffic is usually backlogged continuously [16]. However, wireless environments do not exhibit these two characteristics. As a result, developing good wireless scheduler depends on a model that can best utilize radio resources and is suitable for both real-time traffic and bursty data traffic. As elaborated in Subsection 2.3.2, the service curve is an effective and efficient model that can be used to achieve these two goals. Therefore, the ASC scheduler uses the service curve model for traffic characterization. Consequently, this thesis also contributes to the field by proposing a framework to make the existing service curve model operate well in wireless environments.

2.3. Review of the Service Curve

In this section, we first intuitively introduce the service curve model and formally define it. We then discuss and illustrate the advantages of using service curves. Finally, we provide an overview of what has been done in the literature and possible applications of service curve model.

2.3.1. Definition of Service Curve

The service curve model is a systematic and analytical tool for designing high-speed, integrated services data networks, as first introduced by Parekh and Gallager [21][22] and later generalized by Cruz [10]. A service curve is a function that specifies the minimum amount of service received by the corresponding session within a specific period. Intuitively, a service curve can be interpreted as a service contract or a QoS profile between the session and the scheduler (i.e., network operator). Once such service contract is agreed (usually by admission control), the scheduler guarantees the session that the amount of the service received by that session during each busy period (s,t) is at least $S(t-s)$, where $S(t)$ represents the service curve.

Formally, a session i is said to be guaranteed a service curve $S_i(t)$, where $S_i(t)$ is a *non-decreasing* function, if for any time t_2 when session i is backlogged, there exists a time $t_1 < t_2$, which is the beginning of one of the backlogged periods of session i (not necessarily including t_2), such that the following holds:

$$S_i(t_2 - t_1) \leq w_i(t_1, t_2), \tag{1}$$

where $w_i(t_1, t_2)$ is the amount of service received by session i during period (t_1, t_2).

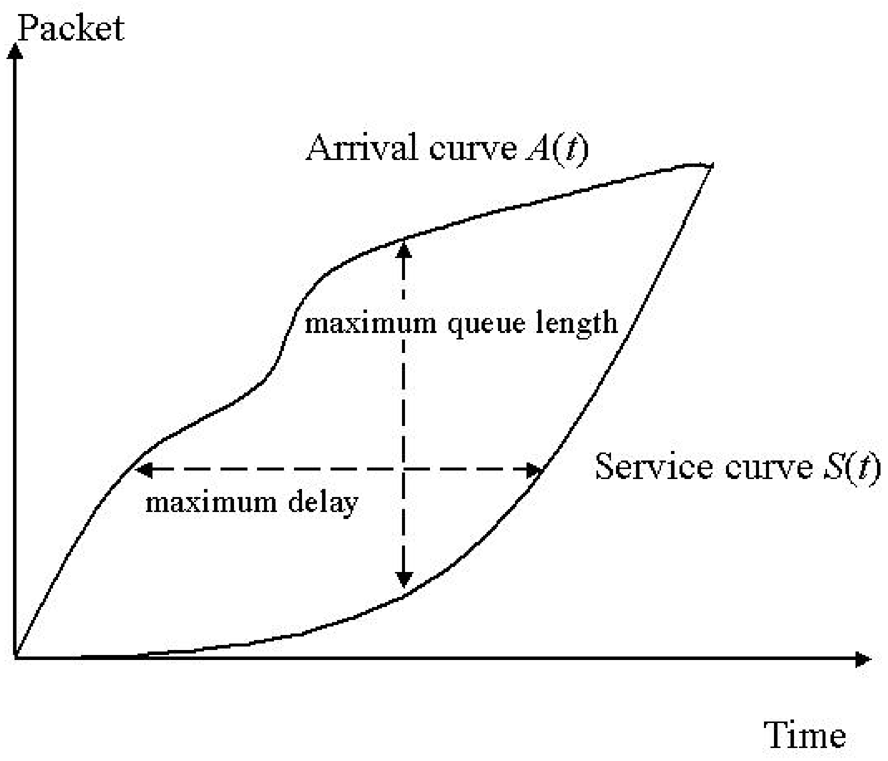

Figure 3: An illustration for the performance bounds

For packetized systems, t_2 is restricted to times at which packets depart.

To see how such framework provides performance bounds, consider the illustration as shown in Figure 3. Let the arrival curve $A(t)$ be the arrival process of packets and the scheduler guarantees the service curve $S(t)$. As interpreted graphically, the maximum queue length is bounded above by the maximum vertical distance between the curves $A(t)$ and $S(t)$, while the maximum delay is bounded above by the maximum horizontal distance between $A(t)$ and $S(t)$. A more formal derivation of performance bounds on backlog and delay can be found in [8].

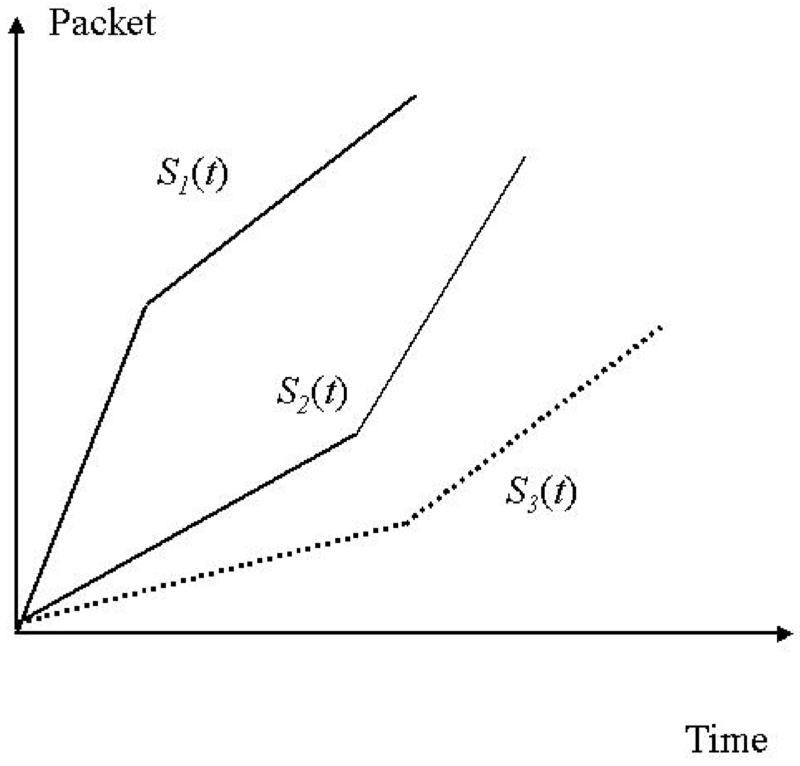

Figure 4: Three examples of service curves

Figure 4 illustrates three examples of service curves. A service curve $S(t)$ is said to be *convex* if for any two times t_1 and t_2, and for any $a \in (0,1)$ we have $S(at_1 + (1-a)t_2) \le aS(t_1) + (1-a)S(t_2)$, e.g., the $S_2(t)$ in Figure 4. Similarly, a service curve is said to be *concave* if $S(at_1 + (1-a)t_2) \ge aS(t_1) + (1-a)S(t_2)$, e.g., the $S_1(t)$ in Figure 4. A convex service curve specifies larger latency but larger throughput, whereas a concave service curve allows smaller latency but specifies smaller throughput. Thus, for example, an audio session that is delay-sensitive should be characterized by a concave service curve, whereas an FTP session that is throughput-oriented is more suitable for a convex one. Even though any non-decreasing function can be used as a service curve, for simplicity of exposition and without loss of generality (enough to decouple bandwidth and delay allocation, as elaborated below), only piecewise linear service curves are considered in this thesis. In practice, a two-piece linear service curve is characterized by four parameters [12] - $m1$, the slope of the first segment; $m2$, the slope of the second segment; b, the y-projection of the point intersection of the two segments; d, the x-projection of the point of intersection of the two segments. Notice that the service curves are *discrete* functions for packetized systems, e.g., the $S_3(t)$ in Figure 4. For idealized fluid systems that assume packets are infinitely dividable, *continuous* functions are used for service curves.

2.3.2. The Advantages of the Service Curve Model

Approaches (such as [9][12][26][28]) based on service curves have several important advantages - 1) decoupling of delay and bandwidth allocation; 2) effectively supporting bursty best-effort traffic; 3) separating mechanisms from policies; 4) systematizing the traffic characterization.

First, as mentioned in [26] and [28], instead of characterizing service by a single parameter, such as minimum bandwidth, weight (under WFQ) or maximum delay, service curves provide a broad spectrum of service characterization, to *best match* the QoS requirements and prevent an unacceptable *over-allocation* of network resources to

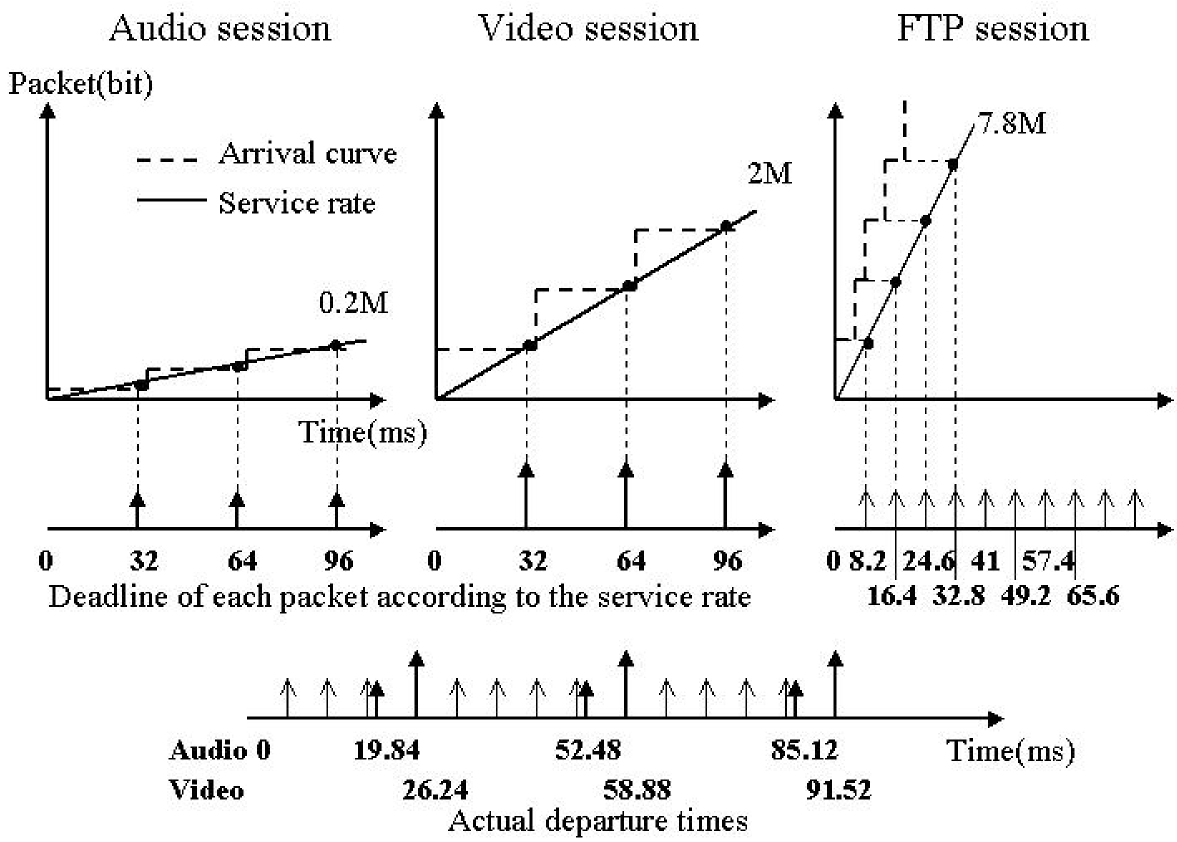

Figure 5: WFQ

users, increasing the flexibility of resource management and utilizing resource. In practice, delay and bandwidth allocation can be decoupled by using non-linear service curves.

To illustrate the advantage of such decoupling property, consider the example (modified and extended from that in [27]) in Figures 5, 6 and 7 where the bandwidth of the server is 10 Mbps. Without any loss of generality, we assume that the server serves the packets in a cut-through manner, which means that a packet may arrive and depart in the same time. There are three sessions: a real-time audio session, a real-time video

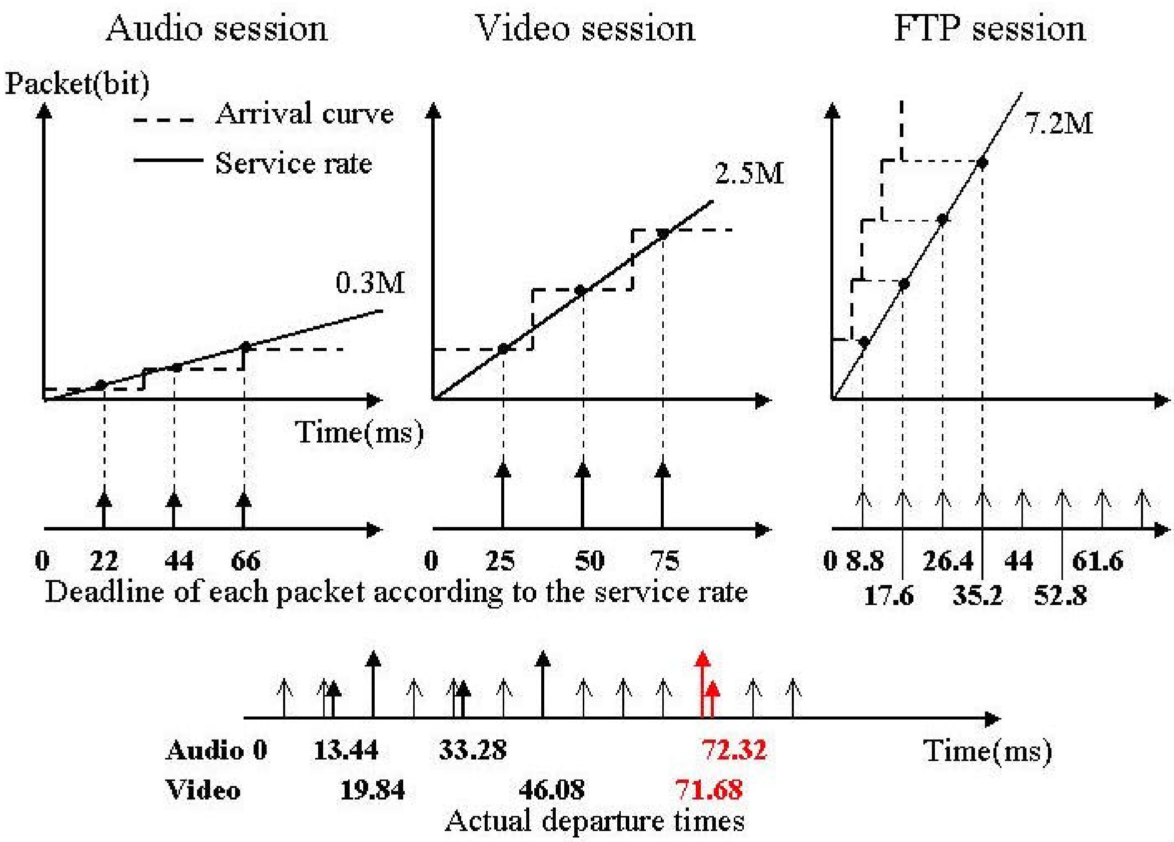

Figure 6: Coupling and over-allocation

session, and a continuously backlogged FTP session. The audio and video sessions are constant-bit-rate sources sending 30 packets of size 0.8 KB and 8 KB per second respectively. In other words, the required bandwidths are 192 Kbps and 1.92 Mbps respectively. The remaining bandwidth is reserved by the FTP session that all packets are of equal size 8 KB. Therefore, it takes 0.64 ms, 6.4 ms and 6.4 ms to transmit a packet of the audio, video, and FTP respectively. The arrival curve represents the cumulative number of bits received by the server. Notice that there is an arrival for each session at time 0, and both audio and video sessions announce arrivals every 33 ms.

When a packet arrives, it is stamped with a deadline corresponding to its service rate or service curve. The server is work-conserving and serves packets in an increasing order of timestamp. If there is a tie, choose the packet with the smallest size.

First, consider the scenario in Figure 5, where a simplified and packetized WFQ scheduler is used for providing the audio, video and FTP sessions with the service rate 0.2 Mbps, 2 Mbps and 7.8 Mbps, which corresponds to one packet per 32 ms, 32 ms, and 8.2 ms respectively. The results show that the average delays are approximately 20 ms for an audio packet and 26 ms for a video packet.

In the second scenario, a delay-bandwidth coupling phenomenon due to the WFQ-like scheduler is illustrated in Figure 6. This is, if a smaller delay is required, the only thing can be done is to increase the guaranteed rate (weight) [21]. The service rates become 0.3 Mbps and 2.5 Mbps for the audio and video sessions. The results show that the delays of the audio and video sessions indeed become shorter. However, the audio session only consumes 0.2 Mbps and the remaining capacity guaranteed for the FTP session is only 7.2 Mbps. This results in the *over-allocation* of network resource. The reason for such coupling phenomenon is that, under WFQ scheme, the delay bound is a function of the guaranteed weight [21].

In Figure 7, a service curve based scheduler like SCED [26] is used. Corresponding to the parameters of a service curve described in Section 2.3.1, $m1$, $m2$ and d are 1.6 Mbps (resp. 6.4 Mbps), 0.2 Mbps (resp. 2 Mbps) and 4 ms (resp. 10 ms) for the audio (resp. video) session respectively. Both service curves are concave. The service curve of the FTP session is chosen to fulfill the remaining capacity, which results in a three-piece linear service curve that two inflection points occur at 4 ms and 10 ms. The results show

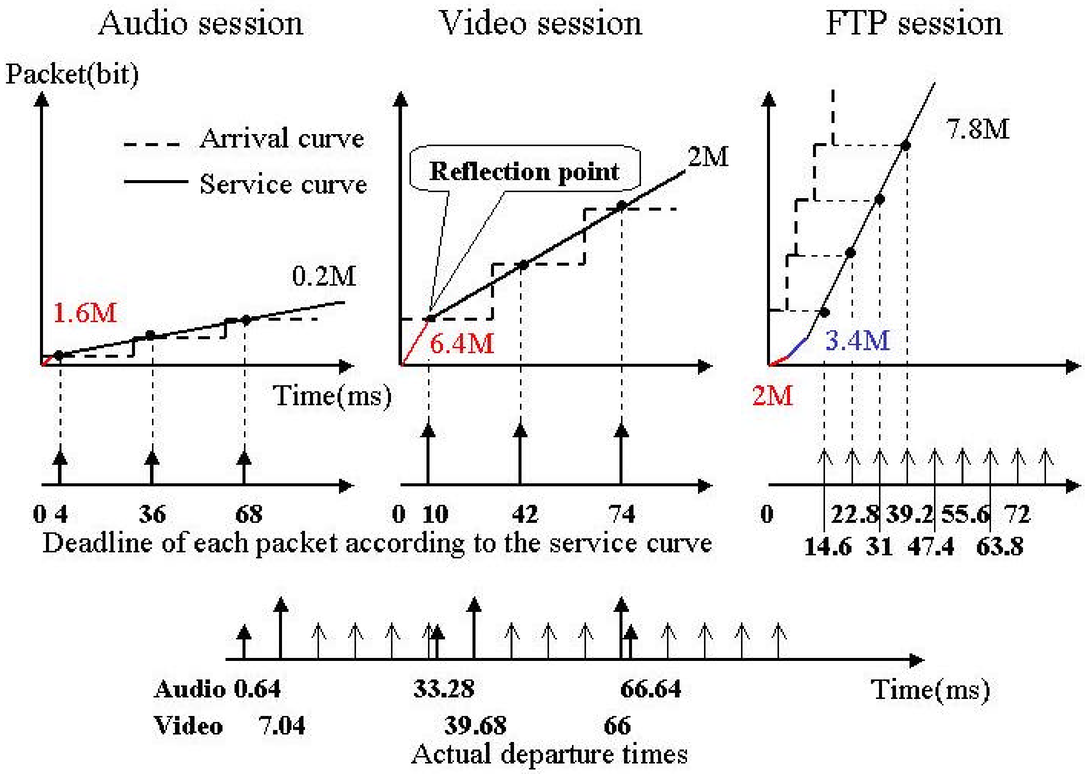

Figure 7: The decoupling of service curve model

that the delays of the video and audio sessions significantly reduce. The overall throughputs remain the same as those in Figure 5. The delay-bandwidth allocation is decoupled and the over-allocation is avoided. The only thing traded is that the average delay of the FTP session increases, which is acceptable due to the throughput-oriented nature of FTP applications.

The second advantage is that a service curve-based approach can effectively and efficiently support bursty best-effort traffic [12]. Most packet fair queuing (PFQ) algorithms favor throughput-oriented applications (such as FTP) over delay-sensitive bursty applications (such as WWW and telnet) [12]. Consider Fair Service Curve (FSC)

[12] as an example. Used for best-effort traffic, FSC can improve the performance of delay-sensitive bursty applications without negatively impacting the performance of throughput-oriented applications, by properly choosing service curves. Besides, the service curve is shown to be suitable for real-time services [28].

These two desirable characteristics of the service curve model exactly meet the two requirements of a good wireless scheduler, namely to best utilize radio resources and to be suitable for both real-time traffic and bursty data traffic.

The third advantage of service curve model is then to separate mechanisms and policies. For example, most existing packet scheduling algorithms are proved to be effective based on the assumption that only the traffic of video and voice is crucial and that data traffic is served with best-effort. However, it is relevant that the contemporary networks will need to be able to provide QoS-support data service [16]. Such a change in policy would possibly require a change in the underlying mechanisms of those packet schedulers, while a service curve based scheduler can keep its mechanism unchanged and regularly serves another different service curve which integrates all of the modified policies. This is especially important to support customized services. Besides, such an abstraction also contributes to keeping the scheduler from distinguishing between different types of traffic.

In addition to the three advantages mentioned above, finally, systematizing the traffic characterization is another important property of service curve [28]. Under such systematization, the situations where the performance goals cannot be simultaneously satisfied can be quantifiably defined. Therefore the fundamental tradeoffs can be explicitly exposed. The situation of difficulty in supporting diverse QoS is exacerbated by the fact that there are no precise definitions that specify all the requirements in an *integrated* fashion, and that most of the scheduling mechanisms are described in prose rather than in a formal model. Furthermore, a systematic mechanism might be more suitable for VLSI implementation. From such a point of view, the service curve based approaches are accordingly more attractive. For example, admission control is made easy and accurate. Specifically, take SCED algorithm as an example. Consider a server with time-varying capacity $C(t)$ that serves M sessions simultaneously. With no assumptions on the arriving traffic from each session, it is straightforward that the sufficient condition that the SCED server is capable of guaranteeing service curve $S_i(t)$ to session i for i = 1, . . ., M is $\sum_{i=1}^{M} S_i(t) \le C(t)$ for all positive real numbers t.

More discussion about the service curve approach and its usefulness in the analysis and design of data networks can be found in [23], and the generalized and systematic explanations by a filtering theory under the min-plus algebra about it can be found in [8].

Chapter 3

Adaptive Service Curve Scheduler

This chapter first explains the central concepts of the Fair Service Curve (FSC) algorithm. The new Adaptive Service Curve (ASC) scheduling algorithm for systems with location-dependent channel errors is then presented. Finally, the approach is objectively assessed.

3.1. Fair Service Curve Model

Stoica et al. have proposed a new service discipline called Fair Service Curve (FSC) [27][28], which serves a wireline part of the ASC scheduler in this thesis. FSC is shown to be effective in both real-time traffic [28] and bursty data traffic [12].

Packets are practically indivisible; therefore, FSC uses an efficient concept called *virtual time*, proposed originally to implement the Packetized Generalized Processor Sharing (PGPS) [21] scheduler that is a packet-by-packet transmission scheme to approximate Generalized Processor Sharing (GPS) scheduler [21]. This concept is extensively used in the literature (e.g., [8][9][11][12]). Specifically, FSC uses the smallest virtual start time $v_i(t)$ as the selection criterion, and $v^s(t) = (v_{i,\min}(t) + v_{i,\max}(t))/2$ as the system virtual time function, where $v_{i,\min}(t)$ and $v_{i,\max}(t)$ are the minimum and maximum virtual start times among all backlogged sessions at time t. When computating the virtual time stamps, FSC calls the remainder of the service curve (also known as the run-time service curve) the virtual curve $V(\cdot)$. When a session i first becomes backlogged, $V_i(v)$ is initialized to the service curve $S_i(t)$. To keep the session i from reclaiming the service allocated to its unbacklogged period, every time the session becomes backlogged, the $V_i(v)$ is updated as follows:

$$V_i(v) = \min(V_i(v), S_i(v - v^s(t)) + w_i(t)), \forall v \geq v^s(t) \qquad (2)$$

This step is just used for eliminating the punishment effect [21]. The virtual time of each session i is defined such that $v_i(t) = V_i^{-1}(w_i(t))$, where $w_i(t)$ is the total amount of service received by session i at time t. FSC is detailed in [13].

However, like other wireline scheduling algorithms, FSC does not work well in wireless environments, as mentioned in Section 2.2. In the rest of this chapter we propose a series of modifications to adapt FSC to wireless networks.

3.2. Curve Adaptation for Error Resilience

This section presents the Adaptive Service Curve (ASC) scheduling algorithm for systems with location-dependent channel errors. Notably, the notation follows that in Section 3.1.

During error-free periods, the ASC scheduler behaves similar to the FSC scheduler and serves sessions according to their requested service curves normally. Things go wrong when channel errors occur, as mentioned in Section 2.2.1. This problem is solved by a new strategy of *curve adaptation*, in which the associated service curve is *adapted* to track the service that is changed in response to channel errors, subject to error resilience requirements.

A *resilience curve* $R(v)$ that specifies error resilience requirements is first defined. A resilience curve is mathematically equivalent to a service curve $S(t)$. It is also a non-decreasing function associated with each session. A combination of the commonly discussed QoS and the resilience requirements is desirable since the error resilience requirements are basically orthogonal to the classical QoS requirements. An analogous proposal, namely that of Resilience-Differentiated QoS (RD-QoS) [2][3], exists for wireline IP services. The authors believe that the resilience curve model provides a steppingstone to *resilience differentiation* for wireless services.

Given the resilience curve $R_i(v)$ and the virtual curve $V_i(v)$ (that is the run-time service curve), the curve adaptation of session i is performed as follows:

$$V_i(v) = \min(V_i(v), R_i(v - v_i(t)) + w_i(t)), \forall v \ge v_i(t), \tag{3}$$

where $R_i(0) = 0$,and $w_i(t)$ and $v_i(t)$ are same as those in Section 3.1. To give an intuition of how curve adaptation works, consider the example shown in Figure 8. The resilience curve is convex and specified by parameters *m1*, *m2*, and d (as defined in Section 2.3.1), and the virtual curve at this moment is linear with slope *m3*. When the session i is scheduled and its channel is bad at time t, curve adaptation is performed once.

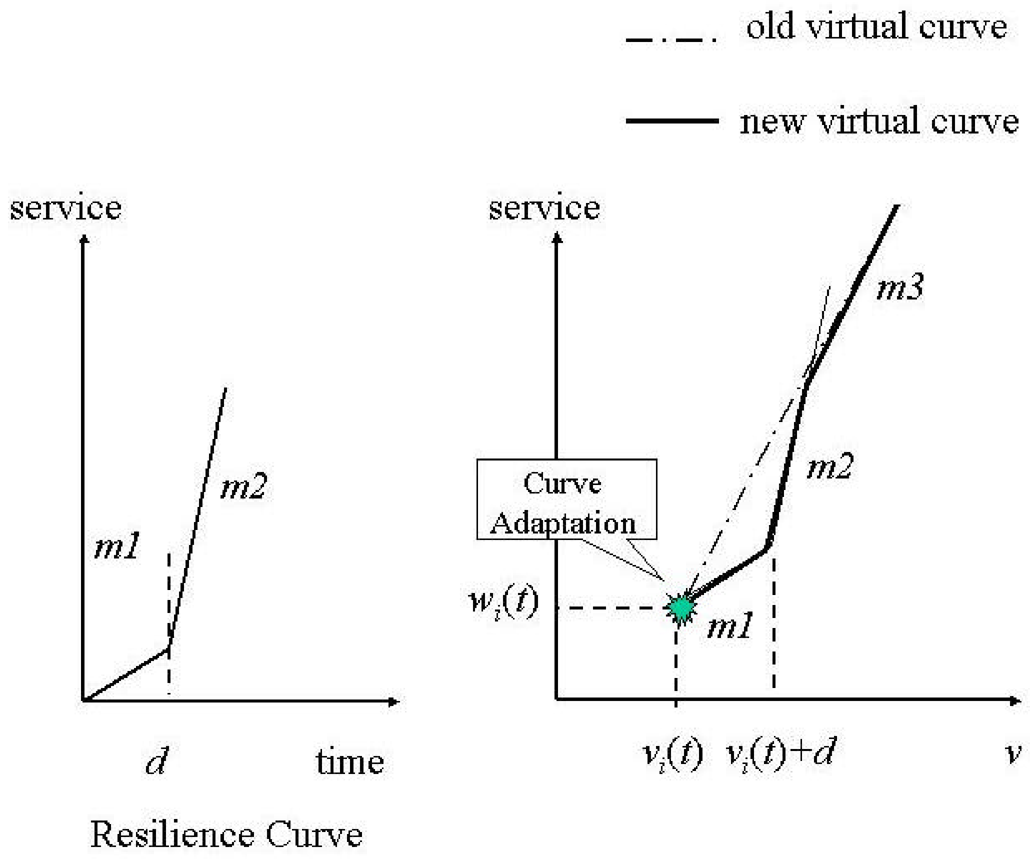

Figure 8: An illustration of the process of curve adaptation.

After the curve adaptation, the virtual curve becomes a three-piece linear curve, implying that, according to the resilience requirements of session i, the transmission rate changes from *m3* to a lower rate *m1* during error burst. When the channel becomes good, a higher rate *m2* is used for compensation. Finally, the service returns to the original scale. Notably, both the behavior in response to error burst and the subsequent compensation depend on the resilience curve, as elaborated in Section 3.3.

Equation (3) has one rationale. The ASC scheduler maintains only one curve, the virtual curve, at run time to prevent associating an independent error-free reference system (for example, [5][11][18][25]), which increasing computing complexity [5]. Thus, the ASC scheduler changes the original virtual curve into a new one according to the corresponding resilience curve, and subsequently guarantees the adapted virtual curve.

The *adaptation period* (AP) of each session is defined such that the minimum interval between adjacent curve adaptations equals the corresponding AP. Restated, curve adaptation is executed only at the moment when session *i* is selected; the state of the corresponding channel is measured to be bad, and the elapsed time from the previous curve adaptation of session *i* exceeds AP_i.

Setting a minimum interval, namely AP, between adjacent curve adaptations is desirable for implementation; otherwise the curve adaptation may be executed whenever the session is selected and the channel is bad. From this perspective, the concept of AP is similar to the so-called *backoff* scheme. Thus, AP can be designed to change dynamically. However, this thesis assumes that the APs are deterministic and their enhancement is regarded as future work. Besides, the underlying concept behind AP is also to avoid continuous polling of the state of error-prone channels because transmitting an unsuccessful RTS-CTS, which acts as a *channel predictor*, to a channel in a bad state causes the server to spend time on a channel without increasing the throughput [14].

Figure 9 shows the pseudocode of the three key procedures in ASC. The first procedure, *receive_packet* (the same as that in [12]), is invoked whenever a packet arrives for a session. The second procedure, *schedule_packet*, is invoked whenever the transmission of a packet completes or when a packet arrives in an idle system, to select

```
receive_packet(i, p)            /* session i has received packet p */
  enqueue(i, p);
  if (i is not active)          /* session i is originally unbacklogged */
    update_v(i, p);             /* update the virtual time and virtual curve
                                   to avoid the punishment effect */
    set i be active;            /* mark session i backlogged */

schedule_packet()               /* select next packet to transmit */
  i = (active sessions);        /* select the minimum virtual time */
  if (v_i > v_i^AP and channel(i) is bad)         /* perform curve adaptaion */
    curve_adaptation(i);
  else
    p = dequeue(i);
    if (v_i ≦ v_i^AP and channel(i) is bad)       /* conduct link adaptation */
      send_packet_more_link_effort(i, p);        /* to preserve service */
    else
      send_packet(i, p);
    update_v(i, p);             /* update the virtual time according to service received */

if (queue(i) is empty)
    set i be non-active;


update_v(i, p) {
  if (not active(i))            /* an unbacklogged session becomes backlogged */
    v_i = max(v_i, v^s);        /* update the virtual time to avoid reclaiming service
                                   from the previous idle period of this session */
    min_VC(i, v^s);             /* compute the remainder of service curve */
  else
    w_i = w_i + length(p);      /* update the virtual time according to service received */
    v_i = V_i^-1(w_i);
}

min_VC(i, v^s) {                /* compute the remainder of service curve */
  V_i(v) = min [V_i(v), S_i(v - v^s) + w_i], all v ≧ v^s
}


curve_adaptation(i) {           /* perform curve adaptation */
  v_i^AP = v_i + AP_i;          /* set adaptation period */
                                /* update virtual curve subject to the resilience curve */
  V_i(v) = min [V_i(v), R_i(v - v_i) + w_i],  v ≧ v_i
  v_i = V_i^-1(w_i);            /* required if V() is not an injection */
}
```

Figure 9: The pseudocode of the ASC scheduler.

the next packet to be served. The third procedure, called *curve_adaptation* was described above.

The scheduler may need to make more link effort by link adaptation, which is governed by the procedure *send_packet_more_link_effort* to preserve the service, subject to the resilience curve. In ASC, the transmissions with more link effort are restricted only during the AP. If one AP expires and the channel remains bad, the curve adaptation will be performed again and a new AP commences, to enable link adaptation.

This thesis does not propose how to implement link adaptation. Consider a scheme as described in [24] for example, to develop an intuition of how coordination works. Given a channel quality measure, Signal-to-Interference Ratio (SIR), different link adaptation modes yield different Block Error Rates (BLER). Let m represent to the slope of the resilience curve guaranteed during AP; R represent the radio interface rate and $BLER(SIR)$ indicate a mapping from the SIR to the BLER. A session i is said to be guaranteed if the following holds:

$$m_i \leqq R(1 - BLER(SIR)). \qquad (4)$$

Notably, this solution is not mature because the error rate is unbounded. That is, in wireless environments, all service guarantees are feasible only if the channel errors are constrained.

Since $V(v)$ may not be an *injection*, its inverse function may not be uniquely defined. In ASC, we define $V_i^{-1}(x)$ to be the largest value y such that $V_i(y) = x$. For generalization, a finite number of simple discontinuities are allowed. To see the demand for this, consider a high-charged session with a three-piece linear virtual curve that the second segment of the curve is vertical after the curve adaptation. This means the session will receive a continuous compensation service of size equal to the *cavity* at the discontinuous point without increasing its virtual time.

3.3. Resilience Curve

Until now, the admission controller and the ASC scheduler have been assumed to be able to set appropriate per-flow resilience curves, although the means by which they might be chosen have not been specified. Some criteria for choosing resilience curves are listed below.

1) Ideally, if d is set to accommodate the most common period of burst errors, a significantly higher throughput can be provided. Besides, the natural characteristics and the behavior in response to channel errors of each traffic type are quite different and must be taken into account for determining the length, d. For example, the regression analysis in [20] indicates that the packet error rate doubles for every 300-byte increment of the packet size, and therefore a large d and small $m1$ are chosen when the mean packet size of the session is large.

2) Another criterion that governs the allocation of resilience curves is based on the result in [20] that almost 90 % of all error bursts are shorter than four packets (the transmission rate and packet size are 1.5 Mbps and 1400 bytes respectively); that is, the duration is about 0.03 second.

3) In practice, the resilience curve should be coupled with the service curve, such that $D(t)$ is a function of $S(t)$. When fast compensation is required, $m1$ should be set as close

as possible to $dV(v(t))/dt$, given that it is differentiable.

An admission controller would need to avoid over-allocating the resources, and would need some policy to determine how many resources could be assigned to each item of traffic and resilience class.

3.4. Discussion and Applicability

The proposed ASC scheduler meets the requirements stated in Section 2.2. We first explain how three kinds of flexibility are achieved. Under ASC, the first kind of flexibility is achieved by choosing different shapes of resilience curves, as discussed in Section 3.3. For second one, users can subscribe to different resilience curves while the operators can determine how many resources could be assigned to each kind of resilience curves according to the distribution of users, system load and the current quality of channels.

The computation of inverse mapping, i.e., $v_i(t) = V_i^{-1}(w_i(t))$ in Figure 9, may be costly. However, the complexity will be decreased tremendously if piecewise linear service curves are used [28].

So far we have not mentioned the coordination between admission control and scheduling. Based on the server curve, admission is made easy and accurate. One policy is proposed in [26] that the sufficient condition that a server with capacity C is capable of guaranteeing service curve $S_i(\cdot)$ to session i for $i = 1, \ldots, M$ is that the summation of all $S_i(t)$ is less than $C \cdot t$, for all positive real numbers t.

A deadline-based scheduler or any scheduler attempting to meet delay or jitter guarantees will clearly need to track outcomes for individual packets rather than entire flows [7].

In general, the service curve model is especially suitable for streaming traffic, when the user is looking at (or listening to) real time video (or audio) and real time streams are involved, because the characteristics of the traffic sources can be predicted. Besides, the proportionality makes ASC a work-conserving mechanism and effectively able to serve best-effort traffic. Reference [12] shows that FSC can significantly decrease the delay experienced by bursty data traffic. As a sequel, we believe ASC is very suitable for wireless Internet of the future and for accessing today's Internet wirelessly.

This thesis uses FSC as the core of the error-free part of the proposed scheduler. However, the concept of curve adaptation can be applied to other mature service curve-based schemes (such as, SCED [26] and H-FSC [28]) to make them applicable to wireless networks.

Chapter 4

Simulation

4.1. Simulation Model

TABLE 2: Parameters used in the four sessions

	Pkt. Size	Src. Model	Service Curve	Resilience Curve
FTP1	3000 bits	Greedy	120 Kbps	Error-free
FTP2	8000 bits	Greedy	700 Kbps	*m1* = 0, *m2* = 900 Kbps, *d* = 100 ms
FTP3	3000 bits	Greedy	80 Kbps	Error-free
FTP4	2000 bits	Greedy	1 Mbps	*m1* = 200 Kbps, *m2* = 1.2 Mbps, *d* = 30 ms

Simulations based on a continuous time and discrete event model equivalent to that discussed here are run using OPNET. The maximum bandwidth C is about 2 Mbps, the same as that of WaveLAN. The primary goal is to illustrate curve adaptation; thus all the service curves are linear for simplicity. Four distinct traffic sessions are simulated. Table I details the parameters used. The resilience curves allocated to the two sessions are both two-piece linear convex. For simplicity, each adaptation period is set to the corresponding

d. Therefore, the virtual curves are at most three-piece linear. Notably, the summation of all service curves remains less than $C \cdot t$. In this case, an *absolute* guarantee is provided. However, the summation of the second segments of all virtual curves possibly exceed $C \cdot t$, yielding a *relative* guarantee. This is still accepted because of the proportionality of ASC. With reference to channel errors, FTP4 experiences the error burst from time 0.1s to time 0.7s; FTP2 suffers from error burst at time 0.32s and returns error-free at time 0.4s; the channels of FTP1 and FTP3 are error-free during their lifetimes.

4.2. Simulation Results

Figures 10 and 11 show the average shared capacity and the service received by the four sessions, respectively. The bandwidth share of FTP4 at time 0.1s drops sharply because its channel is bad. At time 0.32s, it begins to rise because the FTP2 is paused. After time 0.7s, the service of FTP4 is compensated according to the *m2* of its resilience curve and gradually converges to its normal share of bandwidth, 1 Mbps. With reference to FTP2, during the error period, i.e., from 0.32s to 0.4s, the slope of Figure 11 is almost horizontal and few points are logged (served) in Figure 10, because the *m1* of FTP2 is zero. The curves of FTP2 and FTP4 in Figure 11 have a meaningful shape, and closely resemble that in Figure 8. The curves for FTP1 and FTP3 are *parallel* to each other in Figure 10, implying the short-term fairness guarantee provided by ASC, that states the sessions receive their share of service at all time according to their normalized service curves, independently of the behavior of other erroneous sessions.

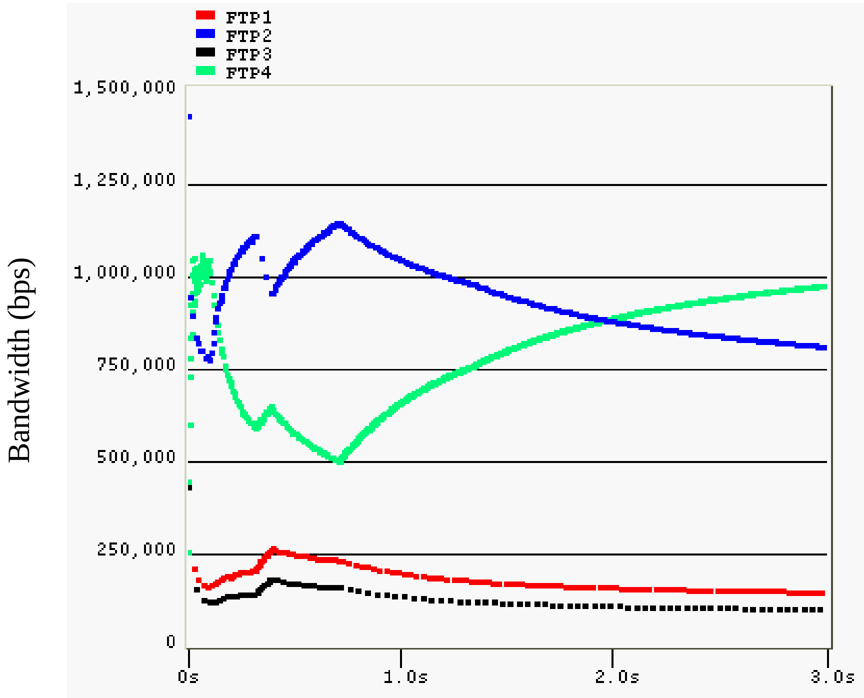

Figure 10: Average bandwidth share of the four sessions.

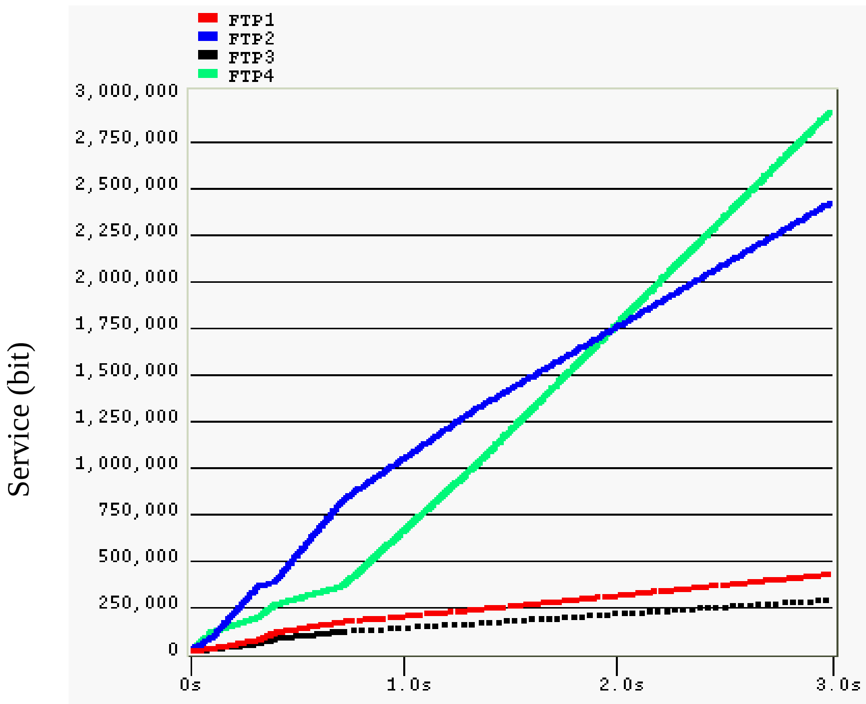

Figure 11: Service received of the four sessions.

Chapter 5

Conclusions

In this chapter, we conclude the thesis by summarizing our contributions and proposing several directions of future research.

5.1. Summary of Contributions

An effective ASC scheduler is proposed in wireless networks with location-dependent channel errors. As a significant originality, we point out that a good wireless scheduler should provide the three kinds of flexibility and the ASC scheduler is shown to exhibit such a property. The proposed curve adaptation scheme explicitly defines both the behavior and the timing of the adaptation of service curve in response to location-dependent channel errors. Besides, for providing error resilience differentiation and serving traffic of various characteristics effectively, ASC is made to comply with link-adaptation. An in-depth discussion on implementation and applicability is also presented. The ASC scheduler has been shown to combine in a single-framework the three aspects of scheduling, namely, link adaptation, flexibility and a mature traffic characterization model. The results in this study provide a complete solution for wireless resource management and the three kinds of flexibility is an invaluable feature for service operators in their daily operations.

5.2. Future Research

Some possible future work can proceed along the following several directions. First, we should provide a more sophisticated modeling between link adaptation and curve adaptation. The ASC algorithm should be extended to support the dynamic adaptation period. As a second possible improvement, we should take medium access control into consideration in order to gain accurate knowledge of channel states. Besides, we believe that the resilience classes should be defined and standardized like the traffic classes in [1]. To best of our knowledge, there is no work on it. Finally, discussions in this thesis have been focused on FSC system. The concept of curve adaptation applies in other service curve based system as well.

REFERENCES

[1] 3G TS 23.107, "Universal Mobile Telecommunications System (UMTS): QoS Concept and Architecture", Jan. 2000.

[2] A. Autenrieth and A. Kirstädter, "Provisioning of Differentiated IP Resilience and QoS: An Integrated Approach," *ITG Wksp. "IP in Telekommunikationsnetzen,"* Bremen, Germany, Apr. 2001.

[3] A. Autenrieth and A. Kirstädter, "Engineering End-to-End IP Resilience Using Resilience-Differentiated QoS," *IEEE Communications Magazine*, *Vol. 40, Issue 1*, Jan. 2002.

[4] P. Bhagwat, P. Bhttacharya, A. Krishna, and S. Tripathi, "Enhancing Throughput over Wireless LANs Using Channel State Dependent Scheduling," in *Proc. IEEE INFOCOM*, Mar. 1996.

[5] S. Bucheli, J. R. Moorman, J. W. Lockwood and S. M. Kang, "Compensation Modeling for QoS Support on a Wireless Network," in *Proc. IEEE GLOBECOM*, Dec. 2000.

[6] Y. Cao and O. K. Li, "Scheduling Algorithm in Broad-Band Wireless Networks," in *Proc. The IEEE*, 89(1): 76-86, Jan 2001.

[7] D. A. Cekhardt and P. Steenkiste, "Effort-Limited Fair (ELF) Scheduling for Wireless Networks," in *Proc. IEEE INFOCOM*, 2000.

[8] Cheng S. Chang, "Performance Guarantees in Communication Networks,"

Springer-Verlag, 2000.

[9] Chia S. Chang and K. C. Chen, "Service Curve Proportional Sharing Algorithm for Service-Guaranteed Multiaccess in Integrated-Service Wireless Networks," in *Proc. IEEE VTC*, 1999.

[10] R. L. Cruz, "Quality of Service Guarantees in Virtual Circuit Switched Networks," *IEEE J. Select. Areas in Commun.*, 13: 1048-1056, 1995.

[11] T. S. Eugene Ng, I. Stoica and H. Zhang, "Packet fair Queueing Algorithms for Wireless Networks with Location-Dependent Errors," in *Proc. IEEE INFOCOM*, Mar. 1998.

[12] T. S. Eugene Ng, D. C. Stephens, I. Stoica and H. Zhang, "Supporting Best-Effort Traffic with Fair Service Curve," in *Proc. IEEE GLOBECOM*, 1999.

[13] T. S. Eugene Ng, D. C. Stephens, I. Stoica and H. Zhang, "Supporting Best-Effort Traffic with Fair Service Curve," *Technical Report, CMU-CS-99169*, Feb. 2000.

[14] C. Fragouli, V. Sivaraman, and M. Srivastava, "Controlled Multimedia Wireless Link Sharing via Enhanced Class-Based Queuing with Channel-State-Dependent Packet Scheduling," in *Proc. IEEE INFOCOM*, 1998.

[15] IEEE Local and Metropolitan Area Network Standards Committee, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-1997, The Institute of Electrical and Electronics Engineers*, 1997.

[16] Z. Jiang, L. F. Chang, and N. K. Shankaranarayanan, "Providing Multiple Service Classes for Bursty Data Traffic in Cellular Networks," in *Proc. IEEE INFOCOM*, Mar. 2000.

[17] Z. Jiang and N. K. Shankaranarayanan, "Channel Quality Dependent Scheduling for

Flexible Wireless Resource Management," in *Proc. IEEE ICC*, June 2001.

[18] S. Lu, V. Bharghavan and R. Srikant, "Fair Scheduling in Wireless Packet Networks," *IEEE/ACM Trans. Networking*, 7: 473-488, 1999.

[19] S. Lu, T. Nandagopal, and V. Bharghavan, "A Wireless Fair Service Algorithm for Packet Cellular Networks," in *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking* (*MOBICOM 98*), Oct. 1998.

[20] G. T. Nguyen, R. H. Katz, B. D. Noble, and M. Satyanarayanan, "A Trace-Based Approach for Modeling Wireless Channel Behavior," in *Proc. Winter Simulation Conf.*, Dec 1996.

[21] 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. Networking*, 1: 344-357, June 1993.

[22] 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. Networking*, 2: 137-150, April 1994.

[23] S. D. Patek, "Service Curves in Data Networks Analysis," *Area Exam Report for the Department of EECS, MIT, Cambridge, MA* 1997.

[24] X. Qiu and J. Chuang, "Link Adaptation in Wireless Data Networks for Throughput Maximization under Retransmissions," in *Proc. IEEE ICC*, 1999.

[25] P. Ramanathan and P. Agrawal, "Adapting Packet Fair Queueing Algorithms to Wireless Networks," in *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking* (*MOBICOM 98*), Oct. 1998.

[26] H. Sariowan, R. L. Cruz, and G C. Polyzos, “SCED: A Generalized Scheduling Policy for Guaranteeing Quality-of-Service,” *IEEE/ACM Trans. Networking*, 7(5): 669-684, Oct. 1999.

[27] I. Stoica, H. Zhang, and T. S. Eugene Ng, "A Hierarchical Fair Service Curve Algorithm for Link-Sharing, Real-Time and Priority Services," in Proc. ACM SIGCOMM, 1997.

[28] I. Stoica, H. Zhang, and T. S. Eugene Ng, “A Hierarchical Fair Service Curve Algorithm for Link-Sharing, Real-Time and Priority Services,” *IEEE/ACM Trans. Networking*, 8(2): 185-199, Apr. 2000.