ABSTRACT

Title of Dissertation: MULTIFRACTAL INTERNET TRAFFIC MODEL AND ACTIVE QUEUE MANAGEMENT

Jia-Shiang Jou, Doctor of Philosophy, 2003

Dissertation directed by: Professor John S. Baras
Department of Electrical and Computer Engineering

We propose a multilevel (hierarchical) ON/OFF model to simultaneously capture the mono/multifractal behavior of Internet traffic. Parameter estimation methods are developed and applied to estimate the model parameters from real traces. Wavelet analysis and simulation results show that the synthetic traffic (using this new model with estimated parameters) and real traffic share the same statistical properties and queuing behaviors. Based on this model and its statistical properties, as described by the Logscale diagram of traces, we propose an efficient method to predict the queuing behavior of FIFO and RED queues. In order to satisfy a given delay and jitter requirement for real time connections, and to provide high goodput and low packet loss for non-real time connections, we also propose a parallel virtual queue control structure to offer differential quality of services. This new queue control structure is modeled and analyzed as a regular

nonlinear dynamic system. The conditions for system stability and optimization are found (under certain simplifying assumptions) and discussed. The theoretical stationary distribution of queue length is validated by simulation.

MULTIFRACTAL INTERNET TRAFFIC MODEL AND ACTIVE QUEUE MANAGEMENT

by

Jia-Shiang Jou

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2003

Advisory Committee:

Professor John S. Baras, Chairman
Professor Eyad H. Abed
Professor Carlos A. Berenstein
Professor Richard J. La
Professor Armand M. Makowski

DEDICATION

To my parents and wife Jiaying

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincerest gratitude to my advisor, John Baras, for his academic guidance, encouragement and support over the past five years. I especially thank his advice on how to conduct good research. The experience and knowledge I have gained from working with him have been invaluable in my life.

I also wish to thank my committee members, Eyad Abed, Carlos Berenstein, Richard La and Armand Makowski, for their precious comments on my dissertation.

In addition, I appreciate the help from the staff of ISR and CSHCN. Special thanks go to Althia Kirlew, Diane Hicks, Peggy Johnson, Jean Lafonta, Trevor Vaughan, Vijay Bharadwaj. I am also grateful for the financial support of my research and studies from the Defense Advanced Research Projects Agency (DARPA) through SPAWAR under contract No. N66001-00-C-8063.

Also, I owe special thanks to my officemates and colleagues, Xiaobo Tan, Hongjun Li, Sudhir Varma, Chang Zhang, Huigang Chen, Vahid Ramezani, Shah-An Yang and Alvaro Cardenas, who provided me with companionship and academic supports. I am also grateful to the friends of the research group I was involved with, Nelson Liu, Majid Raissi-Dehkordi, Xiaoming Zhou, Karthikeyan Chandrashekar and Georgios Papageorgiou, for many illuminating discussions in shaping my research ideas.

Finally, there is no way I can express how much I owe to my family for their love and spiritual support throughout the many years of my education. My parents have been always providing me with their endless love. My brother, Jia-Chen, has often updated me with news of friends back home. He is not only my dearest brother, but also the best friend. And most importantly, I am especially thankful to my lovely wife, Jiaying Lin, for her constant love and encouragement. She was always there when I needed a sounding board and loads of chicken soups and sympathy.

I thank them all who were instrumental in my completing this study and dissertation.

TABLE OF CONTENTS

List of Tables **vii**

List of Figures **viii**

1 Introduction **1**
1.1 Motivation 1
1.2 Contributions 4
1.3 Organization 8

2 Preliminaries **10**
2.1 Wavelet Analysis 10
2.2 Self-Similarity and Long-Range Dependence 17
2.3 Monofractal and Multifractal Processes 24

3 Multilevel ON/OFF Model **27**
3.1 Motivation 27
3.2 Model Description 29
3.3 Parameter Estimation and Model Fitting 30
3.3.1 Trace Format 30
3.3.2 Parameter Estimation 32
3.4 Second Order Statistics 39
3.5 Higher Order Statistics 43
3.6 Queuing Behavior 47
3.7 Summary 49

4 Performance Analysis of Queuing Behaviors **50**
4.1 Overview 50
4.2 Analytical Solution of Logscale Diagram 51
4.3 Approximation of Steady State Queue Length Distribution 63
4.4 Queuing Delay and Jitter Analysis 80
4.5 Simulation results 94
4.6 Summary 98

5 Parallel Queue Structure for Active Queue Management 100
5.1 Overview . . . 100
5.2 Vulnerability of Adaptive RED to Web-mice . . . 103
5.3 A Parallel Virtual Queues Structure . . . 110
5.4 Simulation and Comparison . . . 113
5.5 Dynamic Thresholds for Adaptive RED . . . 123
5.6 Summary . . . 132

6 Performance Analysis of Active Queue Management in a Parallel Queue Structure 133
6.1 Drop-Tail Queue with Adaptive Service Rate . . . 134
6.2 Adaptive RED Queue with Dynamic Thresholds . . . 138
6.3 Summary . . . 158

7 Conclusions 160

A Appendix 163
A.0.1 . . . 163
A.0.2 . . . 164
A.0.3 . . . 165

Bibliography 167

LIST OF TABLES

3.1 Model Parameters 38

5.1 Delivery delay of small file: mean and standard deviation 108
5.2 Experiment Settings 113
5.3 Performance Metrics 114
5.4 Performance Metrics:Red+Tail with Dynamic threshold scheme . . 127

LIST OF FIGURES

1.1 Methodology 6

2.1 Subspaces of multiresolution analysis 11
2.2 Scaling function $\phi(t)$ and mother wavelet $\psi(t)$ of Haar wavelets . . 16
2.3 Logscale diagram of a real Internet traffic trace 23

3.1 Traffic model for one TCP session 29
3.2 Topology of DirecPC network and the bottleneck gateway 39
3.3 Logscale diagram of real Internet trace and estimated slope in region (4,10) 41
3.4 Logscale diagram of real Internet trace and synthetic trace 42
3.5 Partition Function $T(q)$ of Real Trace, Synthetic Traffic and fGn . 45
3.6 Multifractal Spectra of Real Trace, Synthetic Traffic and fGn . . . 46
3.7 First-come-first-serve queue in satellite gateway 47
3.8 Queue Length Tail distribution of Real Trace and Synthetic Traffic with Utilization $\rho = 0.6$, 0.7, 0.8 and 0.9 48

4.1 Logscale diagram of Real Trace and Synthetic Traffic 55
4.2 Probability density function of Pareto and sum of weighted Exponential distributions with N=1,2,...,30. 58
4.3 Logscale diagram of Real Trace and Synthetic Traffic 61
4.4 Logscale diagram of Real Trace and Synthetic Traffic 62
4.5 The CCDF of workload A_j $j = 1, 2, ..., 7$ and the fitted Normal and Lognormal distribution. 66
4.6 The CCDF of workload A_j $j = 8, 9, ..., 14$ and the fitted Normal and Lognormal distribution. 67
4.7 The upper bound of overflow probability with utilization $\rho = 0.1, 0.2, ..., 0.9$. 69
4.8 The queue length distribution with utilization $\rho = 0.6$, upper bound(dash), real trace(solid) and synthetic traffic(cross). 71
4.9 The queue length distribution with utilization $\rho = 0.7$, upper bound(dash), real trace(solid) and synthetic traffic(cross). 72
4.10 The queue length distribution with utilization $\rho = 0.8$, upper bound(dash), real trace(solid) and synthetic traffic(cross). 73

4.11 The queue length distribution with utilization $\rho = 0.9$, upper bound(dash), real trace(solid) and synthetic traffic(cross). 74
4.12 The $Pr[A_j < B+2^jC\Delta]$ *v.s.* octave j, buffer size $B = 10K, 20K, ..., 400K$ (bytes) with utilization ρ=0.3 76
4.13 The $Pr[A_j < B+2^jC\Delta]$ *v.s.* octave j, buffer size $B = 10K, 20K, ..., 400K$ (bytes) with utilization ρ=0.5 77
4.14 The $Pr[A_j < B+2^jC\Delta]$ *v.s.* octave j, buffer size $B = 10K, 20K, ..., 400K$ (bytes) with utilization ρ=0.7 78
4.15 The $Pr[A_j < B+2^jC\Delta]$ *v.s.* octave j, buffer size $B = 10K, 20K, ..., 400K$ (bytes) with utilization ρ=0.9 79
4.16 The target process and the background traffic 80
4.17 Queue length distribution seen by CBR packet $\rho = 0.191$ 82
4.18 Queue length distribution seen by CBR packet $\rho = 0.293$ 83
4.19 Queue length distribution seen by CBR packet $\rho = 0.383$ 84
4.20 Queue length distribution seen by CBR packet $\rho = 0.468$ 85
4.21 Queue length distribution seen by CBR packet $\rho = 0.573$ 86
4.22 Queue length distribution seen by CBR packet $\rho = 0.661$ 87
4.23 Queue length distribution seen by CBR packet $\rho = 0.771$ 88
4.24 Mean CBR packet delay 89
4.25 The arrival and departure time of CBR traffic 91
4.26 The standard deviation of delay jitter $std(J)$ *v.s.* n with a FIFO queue 96
4.27 The standard deviation of delay jitter $std(J)$ *v.s.* n with a RED queue 97

5.1 The dropping probability function of original RED. 101
5.2 Queue length of the Adaptive RED: 10 FTP starting at t=0 and 800 WEBs and 1 CBR coming in at t=100s. 105
5.3 Congestion window size of TCP Reno: One packet loss in the Slow Start phase (left) and One packet loss in the Congestion Avoidance phace (right). 107
5.4 Mean delivery delay of small file *v.s.* dropping probability P_d with file sizes $30, 60, ..., 210Kbytes$, bandwidth $3Mbps$ and round-trip time $128ms$. 109
5.5 The parallel virtual queue structure for active queue management. . 112
5.6 Queue lengths of RED+Tail virtual queues: 10 FTPs starting at t=0 go to virtual queue 2, and 800 WEBs + 1 CBR starting at t=100 go to virtual queue 1. 117
5.7 Packet losses (packets/sec.) of Adaptive RED and RED+Tail. . . . 118
5.8 Packet delays (sec.) of Adaptive RED and RED+Tail. 119
5.9 Throughputs (KBytes/sec.) of Adaptive RED and RED+Tail. . . . 120
5.10 Accumulated throughputs of Adaptive RED and RED+Tail. 121

5.11 Small file delivery delay: mean and standard deviation. 122
5.12 Average queue length with fixed and dynamic thresholds: 20 FTP starting at t=0, and another 20 FTP starting at t=100s and leaving at t=300s, C=6Mbps, d_k=64ms. 125
5.13 Dropping probability with fixed and dynamic thresholds: 20 FTP starting at t=0, and another FTP 20 starting at t=100s and leaving at t=300s, C=6Mbps, d_k=64ms (Inst. P: instantaneous dropping probability; Avg. P: EWMA average of Inst. P). 126
5.14 Dynamic threshold scheme: Virtual queue lengths of RED+Tail and dropping probability of the Adaptive RED queue, 10 FTPs starting at t=0 and 800 WEBs + 1 CBR starting at t=100. 129
5.15 Dynamic threshold scheme: Packet losses (packets/sec.) of RED+Tail. 130
5.16 Dynamic threshold scheme: Packet delays (sec.) of RED+Tail. . . . 131

6.1 Dynamic bandwidth allocation at the drop-tail queue: $C_1(t)$=max(C_{1min}, min($\frac{q_1(t)C}{maxth_1}$, C_{1max})). 135
6.2 Drop-tail queue length with a time varying bandwidth $C_1(t)$. 136
6.3 Mean delays (sec.) of CBR and WEB packets at the drop-tail queue and mean delay of FTP packets at the Adaptive RED queue (with fixed thresholds) with $maxth_1$= 10, 20,...,80 (KBytes). 137
6.4 Queue length and TCP throughput (of a single connection) with C_2=6Mbps, d_k=64ms, $\bar{W}$=6.02×10^4 bits. Compare with simulation in Fig.5.12. 141
6.5 Mapping functions and equilibrium points when N=20, 40 with S=RTT. 142
6.6 Mapping function and equilibrium point when N=40 with S=0.5RTT, 1RTT and 2RTT. 147
6.7 Queue length with N=40, S=RTT and 2RTT. 148
6.8 Steady state queue length distributions for N=20, S=RTT. 153
6.9 Steady state queue length distributions for N=40, S=RTT. 154
6.10 Mapping function and equilibrium point when N=20, 40 and 8 with S=0.5RTT, 1RTT and 2RTT. 155
6.11 Queue length dynamics $N = 20 \rightarrow 40 \rightarrow 8$. 156
6.12 Queue length with $N = 20 \rightarrow 40 \rightarrow 8$. 157

Chapter 1

Introduction

With the rapid development of communication and networking technologies in the last three decades, the Internet has become the largest artificial system in the world. It is one of the most important and quickest media for delivering information nowadays. One may receive and send various kinds of information such as data, voice and video, in the form of email, web page, Internet phone/conference and online radio/TV. The life style of human beings has been greatly changed by this diversity of information delivered and associated information processing and utilization. However, the current usage of the Internet is far beyond its original design envelope and causes many operational and performance problems. In this dissertation, we focus on traffic modeling and active queue management policies to improve the performance of the Internet.

1.1 Motivation

Recent studies [1] [2] [3] [4] [5] [6] [7] [8] on Internet traffic have shown that the aggregate traffic driven by TCP based protocols such as HTTP is not only monofrac-

tal (self-similar) but also multifractal. Wavelet analysis demonstrates that the Internet traffic is monofractal at large time scales (5-10 minutes and larger), which is mainly due to the heavy-tailed distribution of file sizes transferred over the Internet [9] [10] [11] [12]. However, traffic behavior at small time scales is much more complicated and has been shown to be multifractal [2] [13] [14] [15]. This multifractal behavior is primarily due to protocol dynamics such as TCP flow control, network congestion, packet loss and packet retransmission. Taqqu and Willinger [16] explained the monofractal behavior at large time scales by aggregating a large number of independent ON/OFF flows. The ON and/or OFF duration in their model has a heavy-tailed distribution, which corresponds to the total transmission time of a single file and the user think time respectively. They proved that the aggregate traffic converges to the well-known fractional Brownian motion [17] [18] asymptotically when the number of flows goes to infinity. They also found a simple relationship between the shape parameter of the heavy-tailed distribution and the $Hurst$ parameter of self-similarity. However, the single level ON/OFF model [19] [20] [21] [11] [22] [23] [24] is unable to explain the multifractal behavior found at small time scales with its constant rate assumption in the ON duration. On the other hand, Riedi proposed the Multifractal Wavelet Model to capture the second order statistical behavior at all time scales [25] [26]. Like other cascade models [27] [28] [14], these cascade models are unable to explain the observed traffic behavior by simple operational network mechanisms. Furthermore, these cascade models need typically many parameters to fit the statistical behavior of real Internet traffic. Therefore, it is important to provide a more precise model that can capture traffic behavior at all time scales and better explain the relationship between observed traffic statistical properties and natural and simple network

operational mechanisms. One of the goals of this dissertation is to provide such a model for the Internet traffic.

In addition, we are interested in network control and performance enhancement. Most analyses of queuing behaviors [29] [30] [4] [31] has been concerned with infinite or large buffer size. The effect of short range dependence (SRD) on the traffic is absorbed by such a large buffer size. As a result, the analysis result of queuing behavior is dominated by long range dependence (LRD) phenomena [32] [33] [30] [17] [34] [35] [36]. However, the actual buffer size in a real router/switch is finite and small. In this situation, the queue can not retain the "memory" for a long time. On the other hand, short range dependence in the traffic also plays an important role in queuing behavior. Wavelet analysis provides a convenient way to extract important statistical properties from the traffic. For example, the Logscale diagram of the wavelet analysis provides the second order statistical properties of the traffic at all time scales (SRD and LRD). This background motivated us to propose a new approach to describe traffic and predict the corresponding connection performance such as the delay and delay jitter.

The buffer management policy at the bottleneck routers determines the connection quality directly. The drawbacks of the drop-tail policy, such as long queuing delay and low link utilization, for Internet traffic were well-known [37] [38] [39] [40]. In order to improve network performance, many buffer management policies such as RED (Random Early Detection) [41] [42] [43] [44], BLUE [40], FQ/WFQ (Weighted Fair Queue) [45], CHOKe [46] have been proposed and evaluated. Among them, RED attracted the most attention in the field. According to the TCP protocol, the TCP connection decreases its flow rate dynamically if packets are lost. The basic idea of RED is to maintain the queue length within a given region by randomly

dropping packets among the various connections before the buffer is overflowed. The dropping probability is an increasing function of queue length. A connection with a higher rate has a higher risk to lose packets and reduce its flow rate. Since the queue length is controlled and kept within a desired region, link bandwidth is fully utilized and the packets experience smaller mean delay and delay variation.

However, a great portion of Internet traffic belongs to web traffic (HTTP) and voice/video streams (UDP). Most of web pages contain several small files and create a short but bursty bandwidth requirement. As a consequence, the TCP protocol of web connections is primarily operated in the slow start phase. Dropping packets during this phase cannot effectively control the congestion level at the router, but greatly increases the mean and variance of file delivery delay. Since the connection quality of real time applications, such as web and voice/video traffic is sensitive to the mean and variance of delay, these considerations motivated us to propose a parallel virtual queue control structure to serve real time and non-real time traffic separately, and adaptively.

1.2 Contributions

The contributions of this dissertation are in three areas. First, we developed a new multilevel hierarchical model of Internet traffic traces, using simple operational patterns of packets and sessions. As such the principal model and its derivative models can be used in a variety of situations (open-loop (i.e., without flow control), closed-loop (i.e., with flow control)), with different flow and congestion control schemes (i.e., not just TCP), and in modeling traffic at different layers (network or application layer). Second, we developed queue behavior estimates and predictions of performance for FIFO queues fed by such traffic traces. We also

developed estimates of delay and delay jitter. With these prediction results, one can allocate network resources dynamically to guarantee differential quality of service requirements. For example, in order to provide differential quality of service in the DirecPC system, the bottleneck router (Satellite Gateway) in the network operation center has high and low priority queues for different connections. Our prediction method can help the network manager determine how much bandwidth should be allocated to each queue so that their quality of service requirements can be satisfied respectively. Third, we developed a new parallel queue control scheme that treats bursty and non-bursty (real-time) traffic differently for higher performance of bottleneck nodes. We also investigated analytically the stability of the system.

The first and second areas and contributions are more connected and interrelated than the other. The second and third areas address queue control problems from different perspectives. The connection between the first and third areas is that the qualitative characterization of traffic in the first provided the inspiration for the scheme in the third.

The multifractal behavior of Internet traffic at small time scale is mainly contributed by the TCP flow control mechanism. The corresponding connection parameters such as round-trip time, TCP session lifetime and active time of burst play an important role of traffic behavior. Based on these network parameters estimated from real traces, we propose a key traffic model with a multilevel hierarchical ON/OFF structure for Internet Traffic. The key idea is to simultaneously emulate the packet arrival pattern (operational pattern and characterization) in a typical Internet session at small time scales and the user behavior (operational characterization) at large time scales. We first estimate model parameters such as

round-trip time and active time of each connection from real Internet traces. The synthetic traffic trace is generated by the proposed model with the estimated parameters. The statistical behavior of both synthetic and real traffic are compared by employing wavelet analysis techniques. We demonstrate that this new model precisely captures the statistical behavior of real Internet traffic at all time scales. In addition, many network simulations demonstrated that both real and synthetic traffic traces also display a similar queuing behavior when the First In First Out (FIFO) queue control policy is applied. The steady state distributions of queue length in these FIFO queues by synthetic and real traces match as well within a wide range of utilization conditions.

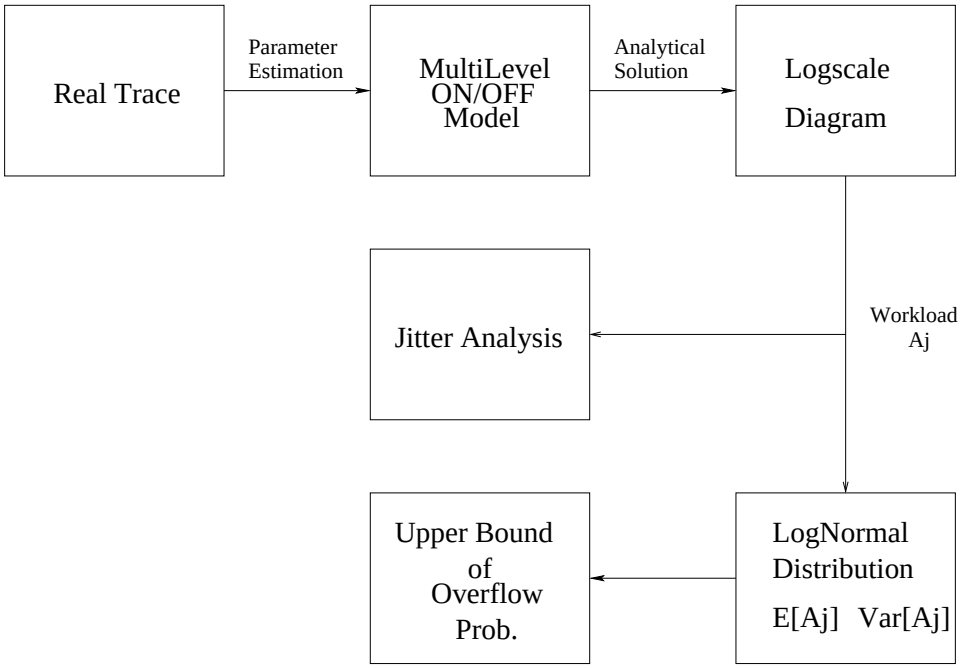

Figure 1.1: Methodology

Since the Logscale diagram of wavelet analysis of traffic traces extracts the second order statistics of traffic traces at all time scales, we provide a numerical approach to obtain the Logscale diagram from the model parameters, instead of

analyzing a long duration trace. Furthermore, we develop a fast algorithm to predict the steady state queue length distribution of such a FIFO queue using parameters computed from the trace Logscale diagram. An upper bound of buffer overflow probability is also derived when the buffer size is finite and given. Figure 1.1 depicts how the trace analysis, traffic models, connection parameters, queue length distribution are interrelated and used. These results are motivated from and apply to network gateways between heterogeneous network domains.

To address the problems of buffer management policy, we first demonstrate that the performance of RED is severely degraded by bursty web traffic. The dropping probability of RED is very sensitive to instantaneous bursts and causes the web connections to see a high instantaneous packet loss rate. This high packet loss rate forces the web TCP connections to enter the congestion avoidance phase prematurely, thus leading to a small TCP window at the sender and low goodput. These effects result in a long delivery delay for small file transmission. In order to prevent the bursty short-life connections (mostly operated in the slow start phase) from disturbing the RED, we propose a parallel queue control structure and apply the RED and drop-tail buffer management policies respectively to the two queues. The proposed policy is validated and evaluated by simulation experiments. This structure preserves the advantages of both RED and drop-tail policies such as high link utilization, low loss rate, small packet delay and low delay jitter, in our new structure/schehe.

Since the real time and non-real time traffic are served at different queues, we have more freedom to control the connection rate. For the non-real time traffic, connection goodput and low packet loss rate are more important than packet delay. Instead of changing the dropping probability to control the flow rate, we propose a

dynamic threshold algorithm to control the flow rate by queuing delay so that the packet loss rate can be limited and kept within a given region. The main advantage of our approach is that it keeps the average packet loss rate within a range of very small values so that the average TCP window size at the senders has a large value in its congestion avoidance phase. Hence, any congestion at the router can not cause another bottleneck at the TCP sender end.

Finally, we model and analyze this queue control system as a regular nonlinear dynamical system. The conditions for system stability are found and stability results are proved. We also develop a numerical approach to obtain the queue length distribution for the dynamic thresholds case. We also provide a linear approximation approach with a small perturbation assumption. Both theoretical results are validated by simulations.

1.3 Organization

The arrangement of this dissertation is as follows. In Chapter 2, we briefly introduce the wavelet analysis method for Internet traffic and show the mono/ multifractal behaviors of real traces. In Chapter 3, we describe the proposed Internet traffic model involving network parameters. We also develop parameter estimation methods to obtain these model parameters from real traces. In Chapter 4, a synthetic traffic trace is generated with the proposed model and estimated model parameters. We compare the statistical properties of these synthetic traces with the corresponding ones from real traces by wavelet analysis (Logscale diagram and multifractal spectrum). Simulation experiments are used to demonstrate the similarity of queuing behaviors in FIFO queues fed by synthetic and real traffic traces. In Chapter 5, we demonstrate the vulnerability of Adaptive RED to bursty web

traffic and propose a parallel virtual queue control structure for buffer management in a router. The system model and performance is given in Chapter 6. Chapter 7 concludes this dissertation and describes directions for future work.

Chapter 2

Preliminaries

In this chapter we provide a brief introduction to wavelet analysis of traffic traces, self-similarity and multifractal processes.

2.1 Wavelet Analysis

Wavelet analysis is a multiresolution analysis (MRA) method and tool, which has been widely used in signal processing and data analysis [47] [48]. It has remarkable advantages in analyzing stochastic processes with long range dependence [49] [50] [51] [52] [53] [54] [55]. For instance, wavelet analysis can eliminate the effect of deterministic trends hidden in random processes if the wavelet function is chosen properly. We now introduce the multiresolution analysis in $L^2(\mathbb{R})$.

Definition 2.1.1 Multiresolution Analysis *[56] [57] A multiresolution analysis consists of a sequence of closed subspaces V_j, $j \in \mathbb{Z}$, of $L^2(\mathbb{R})$ satisfying*

$$V_j \subset V_{j-1} \quad \forall j \in \mathbb{Z} \tag{2.1}$$

$$f \in V_j \leftrightarrow f(2^i(\cdot)) \in V_{j-i} \quad \forall j \in \mathbb{Z} \tag{2.2}$$

$$f \in V_0 \rightarrow f(\cdot - k)) \in V_0 \quad \forall k \in \mathbb{Z} \tag{2.3}$$

$$\bigcap_{j \in \mathbb{Z}} V_j = \{0\}; \tag{2.4}$$

$$\bigcup_{j \in \mathbb{Z}} V_j = L^2(\mathbb{R}); \tag{2.5}$$

There exists a function $\phi \in V_0$, such that

$$\{\phi(\cdot - k) : k \in \mathbb{Z}\} \text{ is an orthonormal basis for } V_0. \tag{2.6}$$

The function ϕ in (2.6) is called the *scaling function* of the given MRA.

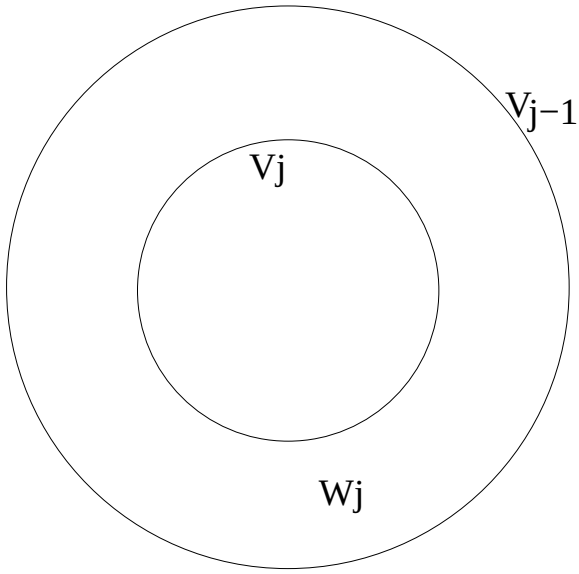

Figure 2.1: Subspaces of multiresolution analysis

Remark 2.1.2 Riesz basis *Eq.(2.6) implies that $\{\phi(\cdot - k) : k \in \mathbb{Z}\}$ is a Riesz basis for V_0. That is, for every $f \in V_0$ there exist a unique sequence $\{\alpha_k\}_{k \in \mathbb{Z}}$ such that*

$$f(t) = \sum_{k \in \mathbb{Z}} \alpha_k \phi(t - k) \tag{2.7}$$

with convergence of the series understood in $L^2(\mathbb{R})$ sense.

Remark 2.1.3 *Let* $\phi_{j,k} = 2^{-j/2}\phi(2^{-j}t - k)$. *According to* (2.2) *and* (2.6), $\{\phi_{j,k} : k \in \mathbb{Z}\}$ *is an orthonormal basis for* V_j.

Now, we introduce the construction of orthonormal wavelets from MRA. Let W_0 be the orthogonal complement of V_0 in V_{-1}; that is, $V_{-1} = V_0 \oplus W_0$. According to Definition 2.1.1, we have

$$V_{j-1} = V_j \oplus W_j, \quad \forall j \in \mathbb{Z}. \tag{2.8}$$

Since $V_j \to \{0\}$ as $j \to \infty$,

$$V_{j-1} = V_j \oplus W_j = \bigoplus_{n=j}^{\infty} W_n, \quad \forall j \in \mathbb{Z}. \tag{2.9}$$

Since $V_j \to L^2(\mathbb{Z})$ as $j \to -\infty$,

$$L^2(\mathbb{R}) = \bigoplus_{n=-\infty}^{\infty} W_n. \tag{2.10}$$

To find an orthonormal wavelet with a given scaling function ϕ, all we have to do is to find a function $\psi \in W_0$ such that $\{\psi(\cdot - k) : k \in \mathbb{Z}\}$ is an orthonormal basis for W_0. Daubechies [57] had shown the existence of orthonormal wavelets with compact support (i.e. finite duration waveforms) in the following theorem:

Theorem 2.1.4 *[56] [57] For any integer* $n = 0, 1, 2, ...$ *there exists an orthonormal wavelet* ψ *with compact support such that* ψ *has bounded derivatives up to order* n. *Moreover,* ψ *can be obtained from an MRA whose scaling function* ϕ *also has compact support and the same smoothness as* ψ.

Many wavelet families such as Haar, Daubechies, Meyer, Morlet, Mexican-hat, Battle-Lemarie, Shannon, etc. [56] [58] [59] [57] have been eveloped for different purposes. We apply these useful results to analyze the statistical properties of Internet traffic traces.

We next give a simple example to explain the procedure. Assume our target function $X(t)$ is defined in the interval [0,1). Let the scaling function be $\phi(t) := 1\{0 \leq t < 1\}$, which forms a basis for the subspace V_0. Since the subspace V_j are all nested, i.e. $V_j \subset V_{j-1}$, and formed by orthonormal bases $\phi_{j,k}(t)$ in Remark 2.1.3, the basis at level $j = 0$ should be expressed in terms of the basis at the finer next level $j = -1$. According to (2.2) we have

$$\begin{aligned} \phi(t) &= \phi(2t) + \phi(2t-1) \\ &= 2^{-1/2}\phi_{-1,0}(t) + 2^{-1/2}\phi_{-1,1}(t). \end{aligned} \tag{2.11}$$

Let

$$a_{j,k} := \int_{-\infty}^{\infty} X(t)\phi_{j,k}(t)dt \tag{2.12}$$

be the coefficient of $\phi_{j,k}(t)$ for all $j, k \in \mathbb{Z}$. We have the projection of $X(t)$ at the subspace V_j:

$$X_j(t) := Proj_j(X(t)) = \sum_{k=-\infty}^{\infty} a_{j,k}\phi_{j,k}(t). \tag{2.13}$$

Now, we define the detail (or residual) process of $X(t)$ at level j as

$$Y_j(t) := X_{j-1}(t) - X_j(t). \tag{2.14}$$

Note that $Y_j(t) \in W_j$ by definition.

Our goal here is to find a basis for the subspace W_j. It is obvious that

$$\begin{aligned} a_{0,0} &= \int_{-\infty}^{\infty} X(t)\phi(t)dt \\ &= \int_{0}^{1} X(t)dt \\ a_{-1,0} &= \int_{-\infty}^{\infty} X(t)\phi_{-1,0}(t)dt \end{aligned} \tag{2.15}$$

$$
\begin{aligned}
&= 2^{1/2}\int_0^{1/2} X(t)\phi(2t)dt \\
&= 2^{1/2}\int_0^{1/2} X(t)dt \qquad (2.16)\\
a_{-1,1} &= \int_{-\infty}^{\infty} X(t)\phi_{-1,1}(t)dt \\
&= 2^{1/2}\int_{1/2}^{1} X(t)\phi(2t-1)dt \\
&= 2^{1/2}\int_{1/2}^{1} X(t)dt \qquad (2.17)
\end{aligned}
$$

and

$$a_{0,0} = \frac{a_{-1,0} + a_{-1,1}}{\sqrt{2}} \qquad (2.18)$$

According to (2.11), (2.14) and (2.18), we have

$$
\begin{aligned}
Y_0(t) &= X_{-1}(t) - X_0(t) \\
&= [a_{-1,0}\phi_{-1,0}(t) + a_{-1,1}\phi_{-1,1}(t)] - a_{0,0}\phi(t) \\
&= \frac{a_{-1,0} - a_{-1,1}}{2}[\phi_{-1,0}(t) - \phi_{-1,1}(t)] \qquad (2.19)
\end{aligned}
$$

We can rewrite $Y_0(t) = d_{0,0}\psi(t)$ for $0 \le t < 1$, where

$$d_{0,0} := \frac{a_{-1,0} - a_{-1,1}}{\sqrt{2}} \qquad (2.20)$$

and

$$
\begin{aligned}
\psi(t) &:= \frac{\phi_{-1,0}(t) - \phi_{-1,1}(t)}{\sqrt{2}} \\
&= \phi(2t) - \phi(2t-1) \qquad (2.21)
\end{aligned}
$$

can form an orthonormal basis for W_0. $\psi(\cdot)$ is called the mother wavelet of the MRA.

Given the scaling function ϕ and the corresponding mother wavelet ψ, the discrete wavelet transform of the continuous time process $X(t)$ is formally defined as follows:

Definition 2.1.5 Discrete Wavelet Transform *[50] Given the scaling function ϕ and the mother wavelet ψ, the approximation coefficients $a_{j,k}$ and detail coefficients $d_{j,k}$ of the discrete wavelet transform of process $X(t)$ are defined as follows*

$$a_{j,k} := \int_{-\infty}^{\infty} X(t)\phi_{j,k}(t)dt \tag{2.22}$$

$$d_{j,k} := \int_{-\infty}^{\infty} X(t)\psi_{j,k}(t)dt, \tag{2.23}$$

where

$$\phi_{j,k}(t) := 2^{-j/2}\phi(2^{-j}t - k) \tag{2.24}$$

$$\psi_{j,k}(t) := 2^{-j/2}\psi(2^{-j}t - k). \tag{2.25}$$

The functions $\phi_{j,k}$ and $\psi_{j,k}$ form an orthonormal basis for V_j and W_j, respectively. The $X(t)$ has the following representation

$$X(t) = \sum_k a_{\infty,k}\phi_{\infty,k}(t) + \sum_{j=-\infty}^{\infty}\sum_k d_{j,k}\psi_{j,k}(t). \tag{2.26}$$

This simple example we just introduced is called 'Haar wavelets'.

Definition 2.1.6 Haar Wavelet *[56] If*

$$\phi(t) = \begin{cases} 1, & \text{if } 0 \le t < 1, \\ 0, & \text{otherwise,} \end{cases}$$

and

$$\psi(t) = \begin{cases} 1, & \text{if } 0 \le t < \frac{1}{2}, \\ -1, & \text{if } \frac{1}{2} \le t < 1, \\ 0, & \text{otherwise,} \end{cases}$$

then ψ is an orthonormal wavelet for $L^2(\mathbb{R})$. This is called the Haar wavelet and $\{\psi_{j,k} : j,k \in \mathbb{Z}\}$ is an orthonormal system in $L^2(\mathbb{R})$ and is shown in Figure 2.2.

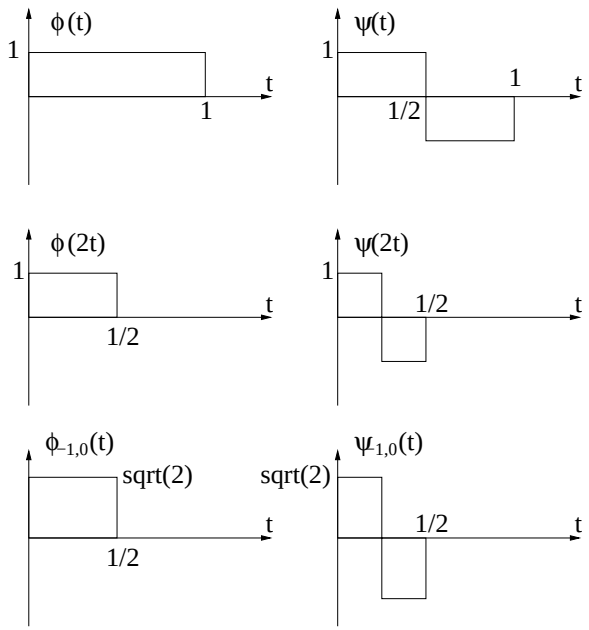

Figure 2.2: Scaling function $\phi(t)$ and mother wavelet $\psi(t)$ of Haar wavelets

Remark 2.1.7 *Since $\phi_{j,k}(t)$ and $\psi_{j,k}(t)$ of Haar wavelets are orthonormal, the coefficients $a_{j,k}$ and $d_{j,k}$ have the following relation:*

$$a_{j,k} = \frac{a_{j-1,2k} + a_{j-1,2k+1}}{\sqrt{2}} \tag{2.27}$$

$$d_{j,k} = \frac{a_{j-1,2k} - a_{j-1,2k+1}}{\sqrt{2}} \tag{2.28}$$

Theoretically, the projection can be performed from $j = -\infty$ to ∞. However, in practice we have a finite index from $j = 0$ to J and we only consider the subspaces $V_J \subset V_{J-1} \subset ... \subset V_0$.

For a discrete time process X_i, $i = 0, 1, 2, ...$, the discrete wavelet transform can be implemented by the fast pyramidal algorithm [60] [61]. To understand the behavior of the traffic X_i, we are more interested in the detail process of the discrete wavelet transform $d_{j,k}$. In the next section, we briefly introduce the long-range dependence property of Internet traffic and its self-similar behavior.

2.2 Self-Similarity and Long-Range Dependence

It is well known that Internet traffic is self-similar and has a long-range dependent property [9] [62] [10]. Many studies [63] [64] have also shown that this long-range dependence property indeed plays an important role in network performance.

Let T be either $\mathbb{R}$, $\mathbb{R}+$ or $\{t : t > 0\}$.

Definition 2.2.1 Self-Simiarity *[65] The real-valued process $X(t)$ is self-similar with index $H > 0$ if for all $a > 0$, the finite-dimensional distributions of $\{X(at), t \in T\}$ are identical to the finite-dimensional distributions of $\{a^H X(t), t \in T\}$; i.e., if for any $k \geq 1$, $t_1, t_2, ..., t_k \in T$ and any $a > 0$,*

$$[X(at_1), X(at_2), ..., X(at_k)] \stackrel{d}{=} [a^H X(t_1), a^H X(t_2), ..., a^H X(t_k)]. \tag{2.29}$$

Where $X \stackrel{d}{=} Y$ denotes that r.v. X and Y have identical distributions.

Definition 2.2.2 Long-Range Dependence *[66] A stationary finite-variance process X_i displays long-range dependence with parameter α if its autocovariance function $R(k) := E[(X_i - EX_i)(X_{i+k} - EX_{i+k})]$ satisfies*

$$R(k) \sim C_r k^{\alpha-1} \ \ as\ k \to \infty, \tag{2.30}$$

where $0 < \alpha < 1$ and C_r is a positive constant.

This also implies that the corresponding spectral density $S(\omega) := \mathscr{F}\{R(k)\}$ satisfies

$$S(\omega) \sim C_f |\omega|^{-\alpha} \ \ as\ \omega \to 0, \tag{2.31}$$

where $C_f = \frac{C_r}{2\Gamma(1-\alpha)sin(\pi\alpha/2)}$ and Γ denotes the Gamma function.

Leland *et al.* [62] have indicated that the Internet traffic is long-range dependent and has a self-similar behavior with Hurst parameter $0.5 < H < 1$. The

Internet traffic observed at different time resolutions has similar statistical properties and this phenomenon cannot be modeled well by traditional traffic models such as Poisson and Markovian processes. A well-known mathematical process for modeling a self-similar process is the so-called fractional Brownian motion process.

Definition 2.2.3 fractional Brownian motion *[29] The fractional Brownian motion* $\{B_H(t), t \in \mathbb{R}\}$ *is a Gaussian process with zero mean and autocovariance function:*

$$Cov(B_H(t_1), B_H(t_2)) = \frac{1}{2}\{|t_1|^{2H} + |t_2|^{2H} - |t_1 - t_2|^{2H}\} Var B_H(1). \qquad (2.32)$$

Remark 2.2.4 *A fractional Brownian motion with* $Var B_H(1) = 1$ *is called standard fractional Brownian motion.*

According to definitions 2.2.1 and 2.2.2, it is obvious that the fractional Brownian motion is self-similar and has a stationary increment. The increment process of fractional Brownian motion is called fractional Gaussian noise. In Chapter 3, we introduce the properties of fractional Gaussian noise process and compare it with our traffic model.

There are many estimation methods proposed to estimate the Hurst parameter of a self-similar process such as R/S analysis, variance time-plots, periodogram analysis, Whittle estimator [62], [67]. However, a self-similar process is strongly autocorrelated and displays a long-range dependence. This long-range dependent property results in a severe estimation bias and a difficulty in estimator convergence. Wavelet analysis is able to avoid this problem by choosing the scaling function (and the corresponding wavelet function) properly [68].

According to the wavelet construction [56], the mother wavelet $\psi(t)$ is a band-pass function between ω_1 and ω_2 in the frequency domain, where ω_1 and ω_2 are

the lower and upper cut off frequency of $\psi(t)$. Therefore, the detail coefficient $d_{j,k}$ can be treated as the output process of the corresponding bandpass filter. The square of the detail process $d_{j,k}^2$ roughly measures the amount of energy around the time $t = 2^j k\Delta$ and the frequency $2^{-j}\omega_0$, where Δ is the unit time interval and $\omega_0 := \frac{\omega_1+\omega_2}{2}$.

Proposition 2.2.5 *[66] If a stationary finite-variance process X_i has long range dependence with parameter α, then the corresponding detail coefficients $d_{j,k}$ have the following property:*

$$\begin{aligned} E[d_{j,.}^2] &\approx \int_{2^{-j}\omega_1<|\omega|<2^{-j}\omega_2} S(\omega)|\mathscr{F}\{2^{-j/2}\psi(2^{-j}t)\}|^2 d\omega \\ &= \int_{2^{-j}\omega_1<|\omega|<2^{-j}\omega_2} S(\omega)2^j|\Psi(2^j\omega)|^2 d\omega \\ &= \int_{2^{-j}\omega_1<|\omega|<2^{-j}\omega_2} C_f|\omega|^{-\alpha}2^j|\Psi(2^j\omega)|^2 d\omega \\ &= 2^{j\alpha}C_f \int_{\omega_1<|\omega|<\omega_2} |\omega|^{-\alpha}|\Psi(\omega)|^2 d\omega \\ &= 2^{j\alpha}C_f C(\alpha,\psi). \end{aligned} \tag{2.33}$$

Note that $C_f C(\alpha,\psi)$ is independent of the variable j.

This property suggests that the parameter α can be estimated by the slope of the $log_2 E[d_{j,.}^2]$ *v.s.* j plot. This plot is named the *Logscale* diagram. One advantage of wavelet analysis is that even when the original process X_i has long range dependence, its wavelet transform $d_{j,k}$ still has short range dependence if the number of vanishing moments N of the mother wavelet $\psi(t)$ is chosen large enough ($N > \alpha/2$).

Definition 2.2.6 *[69] The number of vanishing moments N of the mother wavelet $\psi(t)$ is defined as:*

$$\int t^k\psi(t) \equiv 0, \quad k = 0, 1, 2, ..., N-1. \tag{2.34}$$

Proposition 2.2.7 *[57] [66] If the number of vanishing moments $N > \alpha/2$, then $d_{j,k}$ is stationary and no longer exhibits long range dependence but only short range dependence. i.e., $d_{j,k}$ is quasidecorrelated [70] to each other.*

$$E[d_{j,k}d_{j',k'}] \approx |2^j k - 2^{j'}k'|^{\alpha-1-2N}, \tag{2.35}$$

as $|2^j k - 2^{j'}k'| \to \infty$, where $j \neq j'$ and $k \neq k'$. This implies the higher N is, the smaller the correlation.

In order to estimate the parameter α of long range dependence, one may apply linear regression to estimate the slope of the Logscale diagram.

Definition 2.2.8 *Weighted Linear Regression [71]*
Given a sequence of independent variables (x_j, y_j), j=1,2,..., the hypothesis of linear regression is $Ey_j = bx_j + a$. The unbiased estimator $(\widehat{b}, \widehat{a})$ of (b, a) is

$$\begin{aligned} \widehat{b} &= \frac{\sum y_j(Sx_j - S_x)/\sigma_j^2}{SS_{xx} - S_x^2} \\ \widehat{a} &= \frac{\sum y_j(S_{xx} - S_x x_j)/\sigma_j^2}{SS_{xx} - S_x^2} \end{aligned} \tag{2.36}$$

where

$$\begin{aligned} S &= \sum 1/\sigma_j^2 \\ S_x &= \sum x_j/\sigma_j^2 \\ S_{xx} &= \sum x_j^2/\sigma_j^2 \end{aligned} \tag{2.37}$$

and σ_j^2 is an arbitrary weight.

If y_j are mutually independent then the covariance matrix is

$$Cov(b,a) = \begin{bmatrix} \frac{S}{SS_{xx}-S_x^2} & \frac{-S_x}{SS_{xx}-S_x^2} \\ \frac{-S_x}{SS_{xx}-S_x^2} & \frac{S_{xx}}{SS_{xx}-S_x^2} \end{bmatrix}$$

and the correlation coefficient is $r = \frac{-S_x}{\sqrt{SS_{xx}}}$. Note that the minimum variance unbiased estimator ($MVUE$) is achieved if we set σ_j^2=$Var(y_j)$.

Note that the Logscale diagram is a log-log plot and the logarithm is not a linear operation

$$E\log_2(\bar{d_j^2}) \neq \log_2(E\bar{d_j^2}) = j\alpha + \log_2(C_f C(\alpha,\psi)), \tag{2.38}$$

where $\bar{d_j^2} := \frac{1}{n_j}\sum_{k=1}^{n_j} d_{j,k}^2$. We are not able to apply this linear regression directly. Veitch and Abry [68] [51] [55] developed an asymptotically unbiased and efficient joint estimator for the parameter α and $C(\alpha,\psi)$. They also provide a closed-form expression for the covariance matrix of the estimator and show its accuracy. In their work, y_j is rewritten as

$$y_j \;\; := \;\; log_2(\bar{d_j^2}) - g_j,$$

where

$$g_j \;\; := \;\; \psi(n_j/2)/\ln 2 - \log_2(n_j/2)$$

The regression problem becomes $Ey_j = j\alpha + \log_2 C_f C(\alpha,\psi)$. The estimator of slope α is obtained by performing the above weighted linear regression problem with $x_j = j$ and the weight $\sigma_j^2 = Var(y_j)$.

Figure 2.3 is the Logscale diagram of a real traffic trace from the bottleneck router of DirecPC system. The total length of this trace is about one hour with minimum time resolution one millisecond. The slope at large time scales is a constant which demonstrates that the real traffic is self-similar (monofractal) at

large time scales. However, the slope at small time scales has severe different slope. It indicates that the traffic is a multifractal process.

The Logscale diagram not only demonstrates the long range dependence of traffic but also extracts the second statistics at every time scales. We will employ this property of wavelet analysis to provide an efficient algorithm for predicting queuing behaviors.

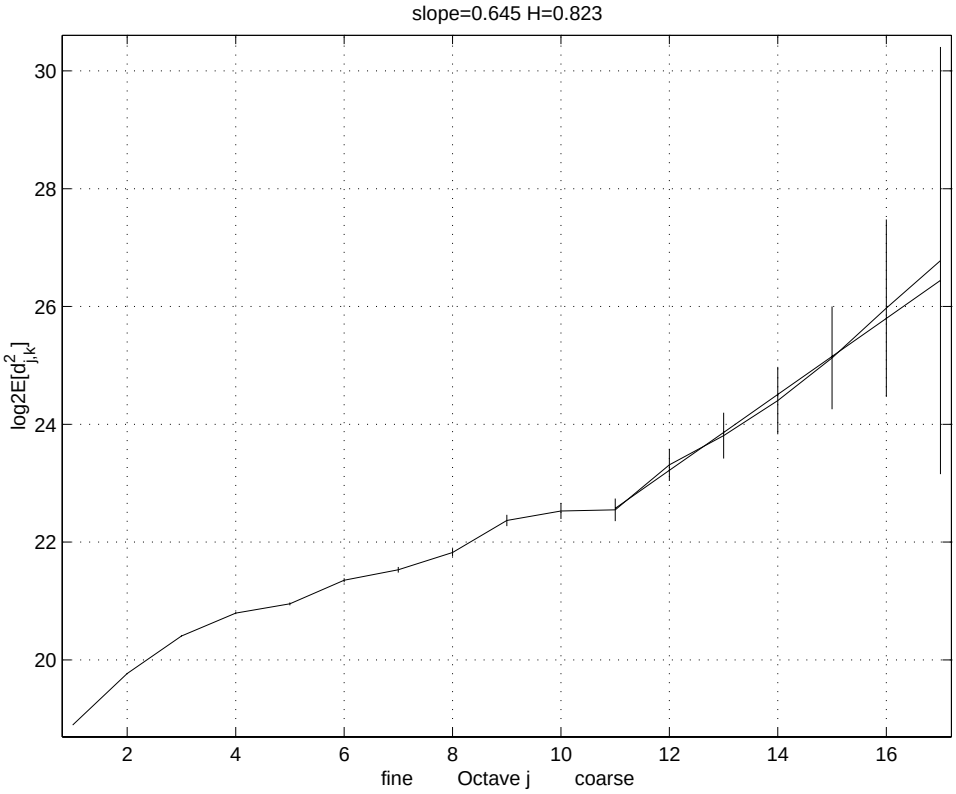

Figure 2.3: Logscale diagram of a real Internet traffic trace

2.3 Monofractal and Multifractal Processes

The basic idea of multifractal analysis is from the large deviation principle (LDP).

Theorem 2.3.1 Large Deviation Principle *[25] [72] Let $Z^{(n)}$ be an arbitrary sequence of random variables on a sequence of probability spaces with probability P_n, and let $a_n \to \infty$. Assume that the following limit exists*

$$-\frac{1}{a_n} log E_n[\exp(qZ^{(n)})] \to c(q) \tag{2.39}$$

and $c(q)$ is finite, concave and differentiable. Then

$$\frac{1}{a_n}\log P_n[\frac{-1}{a_n}Z^{(n)} \in A] \to c^*(\alpha) := \inf_q(q\alpha - c(q)),$$
$$as\ A \to \{\alpha\}$$

where E_n is the expectation w.r.t. P_n.

Consider a normalized cumulative traffic $Y(t)$, $0 \le t \le 1$, with $Y(0) = 0$ and $Y(1) = 1$. Let $X_{k_n} := Y((k_n+1)2^{-n}) - Y(k_n 2^{-n})$, $k_n = 0, 1, ..., 2^n - 1$, denote the normalized traffic increment in the interval $[k_n 2^{-n}, (k_n+1)2^{-n})$. The LDP is employed by defining $Z_{k_n}^{(n)} := \log X_{k_n}$ and $a_n := n \log 2$.

$$\begin{aligned} c(q) &:= \lim_{n\to\infty} \frac{-1}{n\log 2}\log E_n[\exp(qZ^{(n)})] \\ &\approx \frac{-1}{n}\log_2\{\frac{1}{2^n}\sum_{k_n=0}^{2^n-1}(\exp(Z_{k_n}^{(n)})^q\} \\ &= \frac{-1}{n}\log_2\{\frac{1}{2^n}\sum_{k_n=0}^{2^n-1}X_{k_n}^q\}, \end{aligned} \tag{2.40}$$

as $n \to \infty$. If the $c(q)$ exists and is differentiable for all $q \in \mathbb{R}$. According to the LDP, we have

$$\frac{1}{n\log 2}\log\frac{\#\{k_n = 0, 1, .., 2^n - 1 : \alpha - \varepsilon \le \alpha_{k_n} \le \alpha + \varepsilon\}}{2^n}$$

$$\approx \frac{1}{n}\log_2 P[\alpha_i \in [\alpha-\varepsilon, \alpha+\varepsilon]] \to c^*(\alpha)$$
$$as\ n \to \infty,\ \varepsilon \to 0,$$

where

$$\alpha_{k_n} := \frac{-1}{n}\log_2 X_{k_n}. \tag{2.41}$$

In other words, the probability of $X_{k_n} \approx 2^{-n\alpha}$ is approximately equal to $2^{nc^*(\alpha)}$. Note that $c^*(\alpha)$ is a non-positive real function and the multifractal spectrum $f(\alpha)$ is defined as:

$$f(\alpha) := 1 + c^*(\alpha), \tag{2.42}$$

and $2^{nf(\alpha)}$ is interpreted as the "frequency" of a certain value of the *Holder* (or called singularity) exponent α. The idea is easy to understand by the following interpretation [50]. Define the Partition Function

$$\begin{aligned} T(q) &:= \lim_{n\to\infty} \frac{1}{-n}\log_2 E\sum_{k_n=0}^{2^n-1} X_{k_n}^q \\ &= -1 + c(q). \end{aligned} \tag{2.43}$$

We have

$$\begin{aligned} \sum_{k_n=0}^{2^n-1} X_{k_n}^q &\geq \sum_{\alpha_i\approx\alpha} X_{k_n}^q \approx \sum_{\alpha_i\approx\alpha} (2^{-n\alpha})^q \\ &\approx 2^{nf(\alpha)}(2^{-qn\alpha}) = 2^{-n(q\alpha - f(\alpha))}. \end{aligned} \tag{2.44}$$

Take the logarithm on both sides, $T(q) \leq q\alpha - f(\alpha)$ and $f(\alpha) \leq q\alpha - T(q)$. Since this holds for all α and q, we have

$$T(q) \leq \inf_{\alpha}(q\alpha - f(\alpha)) \tag{2.45}$$

$$f(\alpha) \leq \inf_{q}(q\alpha - T(q)). \tag{2.46}$$

Thus the multifractal spectrum $f(\alpha)$ can be obtained by the Legendre transform of $T(q)$. This method is called "increment-based multifractal analysis".

In order to have the numerical advantage of the wavelet analysis [27] [73], one may define the "wavelet-based multifractal analysis" by analyzing the detail process $d_{j,k}$ instead of the increment X_{k_n} in (2.41). Define

$$\widetilde{\alpha}_n(t) := \frac{1}{-n} \log_2(2^{n/2}|d_{-n,k}|). \tag{2.47}$$

It is shown [74] that under some mild conditions, this approach captures the same behavior of the multifractal spectrum. Figure 3.6 shows the wavelet-based multifractal spectrum of a real traffic trace. We will apply this powerful tool to evaluate our model.

Chapter 3

Multilevel ON/OFF Model

3.1 Motivation

According to wavelet analysis of real Internet traffic trace, the Logscale diagram in Fig. 2.1 is shown to be monofractal (self-similar) at large time scales. However, the traffic behavior at small time scales is more complicated and regarded as multifractal. The well known ON/OFF model proposed to model the connection duration and user think time is unable to explain this multifractal behavior at small time scales. Since the traffic behavior at small time scales plays an important role in affecting connection performance such as packet loss and throughput, we are motivated to offer a physical model that can precisely capture the traffic behavior at all time scales and explain the relationship between network parameters and connection performance. However, one has to first understand the network dynamics at small time scales.

There are many protocols designed and deployed on the Internet for various purposes. Nevertheless, the traffic driven by TCP (Transmission Control Protocol) has already dominated the entire network for decades. TCP has a well known

control mechanism for reliable communication and congestion avoidance [75]. In order to avoid congestion at a bottleneck router, the burst size (size of packets) is limited by the current size of the congestion control window. TCP determines the window size according to its current state and packet loss events. The window size update algorithm depends on TCP versions such as Reno and Tahoe. For example, the window size of TCP/Reno has a small initial value (1MSS maximum segment size) and increases its size by one MSS after receiving an acknowledgement from the receiver. This phase is called "slow start". TCP leaves the slow start phase and enters the congestion avoidance phase if TCP encounters a packet loss or its windows size is greater than a parameter called $ssthresh$. The default value of ssthresh is $64KBytes$ and its value is updated by $min(current\_congestion\_window\_size, receiver\_window\_size)/2$. TCP interprets a packet loss event as an indication of network congestion. In the congestion avoidance phase, TCP slowly increases its window size by one packet with every round-trip time and decreases its window size to a half when detecting a new packet loss.

We consider a typical web traffic transported over the Internet. Since most objects in a typical web page are small graphic and text files, the corresponding TCP connection usually spends most of its life in the slow start phase and the packet arrival pattern is much like an ON/OFF process. TCP sends a batch of packets during the ON period. The OFF time is roughly equal to the network round-trip time (RTT). Note that the original ON/OFF process [16] models the TCP session life time and user think time. In order to model the traffic at small time scales, we propose a multilevel ON/OFF model to mimic the operational behavior of a typical Internet connection.

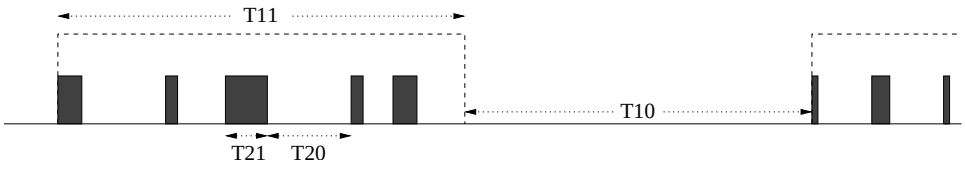

Figure 3.1: Traffic model for one TCP session

3.2 Model Description

We propose a two level ON/OFF model for a single TCP connection as shown in Figure 3.1. The upper level is an ON/OFF process that models the TCP session life time (T_{11}) and user think time (T_{10}). In order to capture the behavior of the TCP mechanism, there is another ON/OFF process inside the ON period T_{11} of the upper level ON/OFF process. It imitates the burst arrival pattern by the active time (duration of a burst T_{21}) and inactive time (T_{20}) within the same TCP session. The packet rate B ($bytes/\Delta$) in T_{21} is assumed to be a constant. T_{11} and T_{21} have Pareto Type I distributions with parameters (K_{11}, a_{11}) and (K_{21}, a_{21}) respectively.[1] The user think time T_{10} and inactive time T_{20} are chosen to be Exponential random variables with mean $1/\lambda_{10}$ and $1/\lambda_{20}$ respectively. All these

[1]The Pareto Type I distribution function is as follows:

$$Pr[T > t] = \begin{cases} (K/t)^a & \text{, if t} \geq \text{K} \\ 1 & \text{, if t} < \text{K} \end{cases}$$

random variables are statistically independent to each other.

$$
\begin{aligned}
T_{10} &:= r.v.\ Exp\ (1/\lambda_{10}) \\
T_{11} &:= r.v.\ Pareto\ (K_{11}, a_{11}) \\
T_{20} &:= r.v.\ Exp\ (1/\lambda_{20}) \\
T_{21} &:= r.v.\ Pareto\ (K_{21}, a_{21}) \\
B &:= \text{Data rate within the active period } (bytes/\Delta) \\
N &:= \text{Number of connections}
\end{aligned}
$$

The synthetic traffic will be generated by aggregating N independent multilevel ON/OFF processes with burst rate B, which will be formulated in (3.1). It is desired to generate synthetic Internet traffic traces that match the real Internet traffic traces as measured by the degree of matching in the Logscale diagram and in the multifractal spectrum.

3.3 Parameter Estimation and Model Fitting

3.3.1 Trace Format

Before discussing parameter estimation and model fitting, we briefly describe the format of a real Internet trace observed in a bottleneck router. The raw trace is recorded by *tcpdump*, which is a network monitoring tool developed by the Department of Energy Lawrence Livermore Lab. To estimate the parameters of the proposed traffic model, we need the following information:

- Timestamp: the arrival time of a packet
- Packet size: the length of a packet
- Source address: source host IP address

- Destination address: destination host IP address
- Source port: source TCP port number
- Destination port: destination TCP port number
- TCP flag: indicate the SYN, SYN-ACK, and FIN packet

One can extract every TCP connection by the source-destination pair (Source IP-address.port, Destination IP-address.port). The aggregate traffic X_i is collected by the time stamp and the corresponding packet size that belongs to the i^{th} time interval. The time interval between the SYN and FIN packets of a certain TCP session is defined as the session life time. Similarly, the time interval between SYN and SYN-ACK packets gives a measure of the round-trip time. From those observations, we are able to estimate the following statistics:

- Mean traffic rate (EX_i)
- Autocorrelation function $R(k)$
- Logscale diagram L_j
- Mean round-trip time (ET_{20})
- Mean session time (ET_{11})

Given a real Internet trace, our goal is to estimate the corresponding model parameters, generate a synthetic traffic trace from the model and demonstrate that the synthetic traffic has similar statistical properties and queuing behaviors with the real traffic trace.

3.3.2 Parameter Estimation

According to the definition of the multilevel model, the aggregate traffic can be written in terms of the sum of $i.i.d$ indicator functions.

Definition 3.3.1 *Assume the multilevel ON/OFF process is already in the steady state before time* 0. *Let* $U_k(t)$ *and* $V_k(t)$ *denote the indicator functions defined as:*

$$U_k(t) := 1\{k^{th} \text{ connection is in } ON \text{ state at } t\}$$
$$V_k(t) := 1\{k^{th} \text{ connection is in Active state at } t\}.$$

The aggregate cumulative byte count of the multilevel ON/OFF process is

$$Y(t) = B \int_0^t \sum_{k=1}^{N} U_k(u) V_k(u) du. \tag{3.1}$$

The single ON/OFF process can be treated as a special case of the multilevel ON/OFF process by setting $B = 1$ and $V_k(t) = 1$ for all t. Hence, the aggregate cumulative byte count of a single level ON/OFF process in the interval [0, Tt) is defined as:

$$\widehat{Y}(Tt) = \int_0^{Tt} \sum_{k=1}^{N} U_k(u) du. \tag{3.2}$$

Taqqu, Willinger and Sherman proved the following theorem:

Theorem 3.3.2 *[16] For large N and T, the aggregate cumulative process* $\{\widehat{Y}(Tt), t \geq 0\}$ *behaves statistically like*

$$TN \frac{ET_{11}}{ET_{11} + ET_{10}} t + T^H \sqrt{N} \sigma_{lim} B_H(t), \tag{3.3}$$

where the Hurst parameter $H = (3 - a_{11})/2$ *and* $B_H(t)$ *is the standard fractional Brownian motion.*

In this theorem, the relationship between the Hurst parameter H (index of self-similarity) and shape parameter a_{11} (index of Pareto distribution) had been proved. This theorem explained that the self-similarity of Internet traffic is mainly due to the heavy-tailed distribution of file sizes which are typically transmitted over the Internet.

Since the lower level ON/OFF process only exists in the ON period of the upper level process, one can easily have the following relationship between $Y(t)$ and $\widehat{Y}(t)$:

Proposition 3.3.3 *If* $max(ET_{21}, ET_{20}) << ET_{11}$, *then*

$$\lim_{t\to\infty} \frac{Y(t)}{\widehat{Y}(t)} = E[V]B. \tag{3.4}$$

As time $t \to \infty$, the aggregate cumulative traffic of the multilevel ON/OFF process $Y(t)$ is statistically like the fractional Brownian motion.

In order to match the second order statistical properties of the real traffic, we have to estimate the model parameters from the real trace. Instead of dealing with the cumulative process $Y(t)$, we define the increment process X_i of the traffic.

Definition 3.3.4 *The increment process of* $Y(t)$ *is defined as*

$$X_i := Y((i+1)\Delta) - Y(i\Delta),\ i = 0, 1, 2, 3, ... \tag{3.5}$$

where Δ *is the minimum time resolution of interest. The increment process* X_i *is interpreted as the total bytes that arrive in the interval* $[i\Delta, (i+1)\Delta)$.

Remark 3.3.5 *If $B_H(t)$ is a fractional Brownian motion (FBM), the increment $G_i := B_H((i+1)\Delta) - B_H(i\Delta)$ is a stationary sequence called fractional Gaussian noise (fGn).*

Proposition 3.3.6 *[65] The fractional Gaussian noise G_i has the autocovariance function*

$$\begin{aligned} R_G(k) &= \frac{\sigma_0^2}{2}(|k+1|^{2H} - 2|k|^{2H} + |k-1|^{2H}),\ k \in \mathbb{Z} \\ &\approx \sigma_0^2 H(2H-1)k^{2H-2},\ as\ k \to \infty \end{aligned} \tag{3.6}$$

where $\sigma_0^2 \equiv Var[G_i]$.

From $(A.1)$ and (3.5), it is obvious that the autocovariance function of X_i has the same form as (3.6) as $k \to \infty$. Recall that the introduction of wavelet analysis and equation (2.30) suggest that the Logscale diagram of the process X_i has a slope $\alpha = 2H - 1 = 2 - a_{11}$ at the large time scales. Veitch and Abry [68] provided an asymptotically unbiased and efficient estimator for the slope α in the Logscale diagram within a certain region. We will apply this method to estimate the model parameter a_{11} through the estimation of slope α at large time scale region.

Similarly, we consider the traffic X_i at small time scale. Under the assumption that $T_{11} >> T_{21}$ and T_{20}, the traffic behavior at small time scale is dominated by the lower lever ON/OFF process $V(t)$. When we observe the upper level ON/OFF process $U_k(t)$ with a small time scale, $U_k(t)$ behaves like a constant. We may rewrite $\lim_{t\to 0} Y(t)$ as:

$$\begin{aligned} \lim_{t\to 0} Y(t) &= \lim_{t\to 0} B \int_0^t \sum_{k=1}^{N} U_k(u)V_k(u)du \\ &\approx B \sum_{k=1}^{N} U_k(0) \lim_{t\to 0} \int_0^t V_k(u)du. \end{aligned} \tag{3.7}$$

It is another single (lower) level ON/OFF process from the point view of small time scale. Based on the structure of the multilevel ON/OFF model, there is another linear region in the Logscale diagram in the small time scale region. It is contributed by the lower level ON/OFF processes. One might use the same technique to estimate the model parameter a_{21} in the small time scale region.

The parameter K_{11} of T_{11} is estimated by matching the first moment of the session life time.

$$\widehat{K}_{11} := \frac{a_{11}-1}{a_{11}} ET_{11} \tag{3.8}$$

Unlike estimating K_{11} by matching the mean of session life time T_{11}, there is no control packet in the real trace indicating the start and end of each active period T_{21}. As mentioned before, we assume that the upper ON/OFF process always keeps its state when we observe the process with a small time scale. The parameter K_{21} is estimated by the normalized autocorrelation function $R_n(t)$ of X_i.

Definition 3.3.7 *Let $R(t)$ be the autocorrelation function of the multilevel ON/OFF process. When $t \to 0$, we assume that the function $U(t)$ is a constant function and have the following approximation:*

$$R(t) \approx N(\Delta B)^2 E[U^2(0)]E[V(0)V(t)], \text{ as } t \to 0.$$

We also define $\pi_{11}(t) := Pr[V(t)=1|V(0)=1]$ and the normalized autocorrelation function $R_n(t)$:

$$\begin{aligned} R_n(t) &:= \frac{R(t)}{R(0)} \approx \frac{E[V(0)V(t)]}{E[V^2(0)]} \\ &= \frac{Pr[V(t)=1, V(0)=1]}{Pr[V(0)=1]} \\ &= \pi_{11}(t), \text{ as } t \to 0. \end{aligned} \tag{3.9}$$

Theorem 3.3.8 *[16] [76] Assume that the ON/OFF process V(t) is stationary, the renewal equation for $\pi_{11}(t)$ is*

$$\pi_{11}(t) = G_{1c}(t) + \int_0^t F_{1c}(t-u)dH_{10}(u), \tag{3.10}$$

where $G_{1c}(t)$:= Pr[residual life of the first ON interval $> t \mid$ at time 0 is ON], $F_{1c}(t) := Pr[T_{21} > t]$ and $H_{10}(u)$ is the renewal function corresponding to the inter-renewal distribution $F_1 \star F_0$. ($H_{10} = \sum_{k=1}^{\infty}(F_1 \star F_0)^{\star k}$, where $\star$ denotes the convolution and $F_i(t)$, $i = 1$ and 0, is the CDF of ON and OFF time respectively.)

Since we are interested in the behavior of $R_n(t)$ around $t = 0$, we have the approximation

$$R_n(t) \approx G_{1c}(t),\ as\ t\ \rightarrow\ 0.$$

Given that the T_{21} has $Pareto(K_{21}, a_{21})$ distribution, we have

$$\begin{aligned} G_{1c}(t) &= \frac{1}{ET_{21}}\int_t^{\infty}(\frac{K_{21}}{u})^{a_{21}}du \\ &= \frac{K^{a_{21}-1}}{a_{21}}t^{-a_{21}+1}. \end{aligned}$$

The parameter K_{21} can be estimated by the following estimator:

$$\widehat{K}_{21} := \Delta(a_{21}\widehat{R}_n(\Delta))^{1/(a_{21}-1)},\ as\ \Delta\ \rightarrow\ 0,$$

where

$$\widehat{R}_n(\Delta) := \sum_i X_i X_{i+1} / \sum_i X_i^2.$$

The empirical results show that $\widehat{R}_n(t)$ and $G_{1c}(t)$ have a good match when t is small.

To estimate the parameter $1/\lambda_{20}$, or equivalently the mean inactive period, we need to measure the network round-trip time from the trace. It can be extracted from the real trace by the duration between the SYN packet and the SYN-ACK packet at the beginning of each TCP session. In our model, the round-trip time is equal to the lower level OFF time T_{20}. We have the mean of T_{20}:

$$1/\lambda_{20} = ET_{20}.$$

The parameter B is the constant data rate in the active period T_{21}. With the assumption of independent connections, we have the following proposition:

Proposition 3.3.9 *Let* $R_1 := \frac{ET_{11}}{ET_{11}+ET_{10}}$ *and* $R_2 := \frac{ET_{21}}{ET_{21}+ET_{20}}$, *we have*

$$EX_i = N\Delta B R_1 R_2 \tag{3.11}$$

$$EX_i^2 = N(\Delta B)^2 R_1 R_2. \tag{3.12}$$

and

$$B = \frac{EX_i^2}{\Delta EX_i}.$$

Thus the estimator $\widehat{B}$ is defined as

$$\widehat{B} := \frac{\sum_i X_i^2}{\Delta \sum_i X_i}.$$

Equation (3.11) implies that there is one degree of freedom to choose N and R_1(or ET_{10} equivalently). In order to satisfy the assumption in the theorem 3.2, one needs to select a large integer for N so that the mean OFF time $1/\lambda_{10}$ can be determined by R_1 in (3.11). Since T_{10} is an Exponential random variable and $ET_{10} >> ET_{11}$, the starting time of each TCP session can also be approximated by a Poisson process as $N \to \infty$.

Figure 3.2 shows the network topology and bottleneck router (Satellite Gateway) of the DirecPC system, of Hughes Network systems. We measure the model parameters from the downlink traffic (from Internet to user) and generate a synthetic traffic trace by our model and the corresponding model parameters. At least ten very long traces (more than a hour) with a high time resolution (one millisecond) are analyzed and compared by our model. For example, one real trace recorded on 17:00-18:00 Oct. 13 1999 is analyzed and compared by the corresponding synthetic traffic trace. According to the previous parameter estimation methods, we have the following results. The mean round-trip time and the mean TCP session time are $0.130sec$ and $4.896sec$ respectively. The aggregate trace has a mean rate $513.98bytes/msec$ and variance $9.8990 \times 10^5 bytes^2/msec$. The normalized autocorrelation function $\widehat{R}_n(1msec)$ is 0.3519. The shape parameters a_{11} and a_{21} are estimated by the slopes of the Logscale diagram in the regions of small and large time scales as shown in Figure 2.3 and 3.3. Table 3.1 provides the corresponding parameters of this model.

With our model and the estimated model parameters, we generate the synthetic trace by the constant bit rate assumption within the lower level ON duration T_{21} of our model (within the period of T_{21}, packets are generated every one millisecond with packet size B bytes), as shown in Figure 3.1. The aggregate synthetic trace are generated by aggregating N independent such connections.

Table 3.1: Model Parameters

Para.	K_{11}	a_{11}	$1/\lambda_{10}$	B
Value	1.27sec	1.35	167.55sec	1926bytes/ms
Para.	K_{21}	a_{21}	$1/\lambda_{20}$	N
Value	0.54ms	1.77	128.75ms	1000

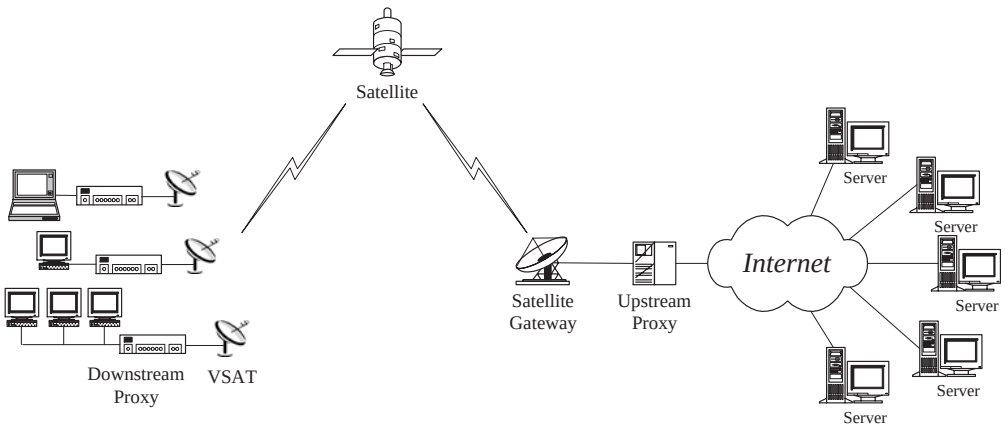

Figure 3.2: Topology of DirecPC network and the bottleneck gateway

3.4 Second Order Statistics

We employ the discrete wavelet transform to analyze the real and synthetic traffic. The second order analysis of the traffic is obtained by studying the detail process of wavelet transforms $d_{j,k}$. As mentioned previously, the X_i, $i = 0, 1, 2, ..$ is the time series of total bytes transmitted during the interval $[i\Delta, (i+1)\Delta)$. In order to avoid the estimation error from the deterministic trend, the mother wavelet of the discrete wavelet decomposition is chosen to be the *Daubechies* wavelet with the number of vanishing moments $N = 3$. The Logscale diagram in Figure 3.4 is the energy of the detail process $log_2 E[d_{j,k}^2]$ *v.s.* the octave j of the real traffic and synthetic traffic. It is easily seen that the second order statistics of these two traffic traces have almost the same values on every scale. This match also implies their similar autocorrelation structures in time.

There is a breaking point around the scale $j = 11$ ($2^{11}\Delta = 2.048sec$) related to the minimum value of T_{11} ($=K_{11}$ in the model). When the observing time scale is smaller than 2 sec. in this trace, the traffic behavior is dominated by the lower level ON/OFF process or equivalently by the TCP congestion control mechanism.

The figure 3.4 shows that TCP dynamics can be modeled well in the second order behavior by a multilevel ON/OFF process. On the right-hand side of the breaking point, the behaviors of the real and synthetic traces are both monofractal with the same Hurst parameter ($H \approx 0.823$). Note that the slope α and the $Hurst$ parameter [66] have the relation $\alpha = 2H - 1$.

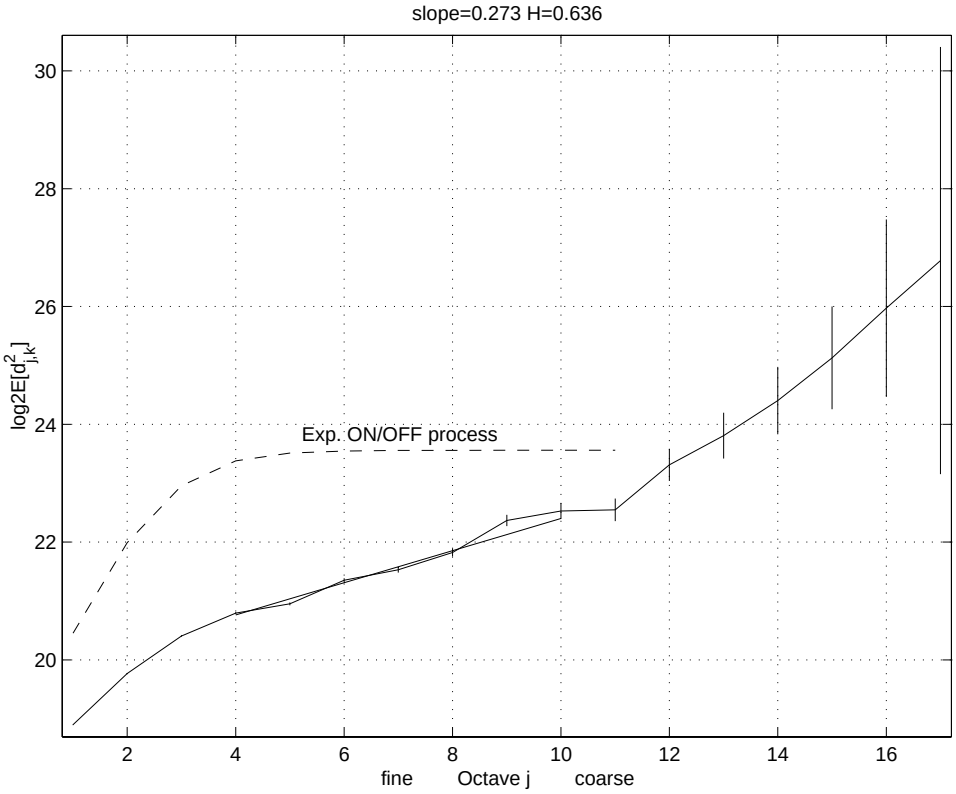

Figure 3.3: Logscale diagram of real Internet trace and estimated slope in region (4,10)

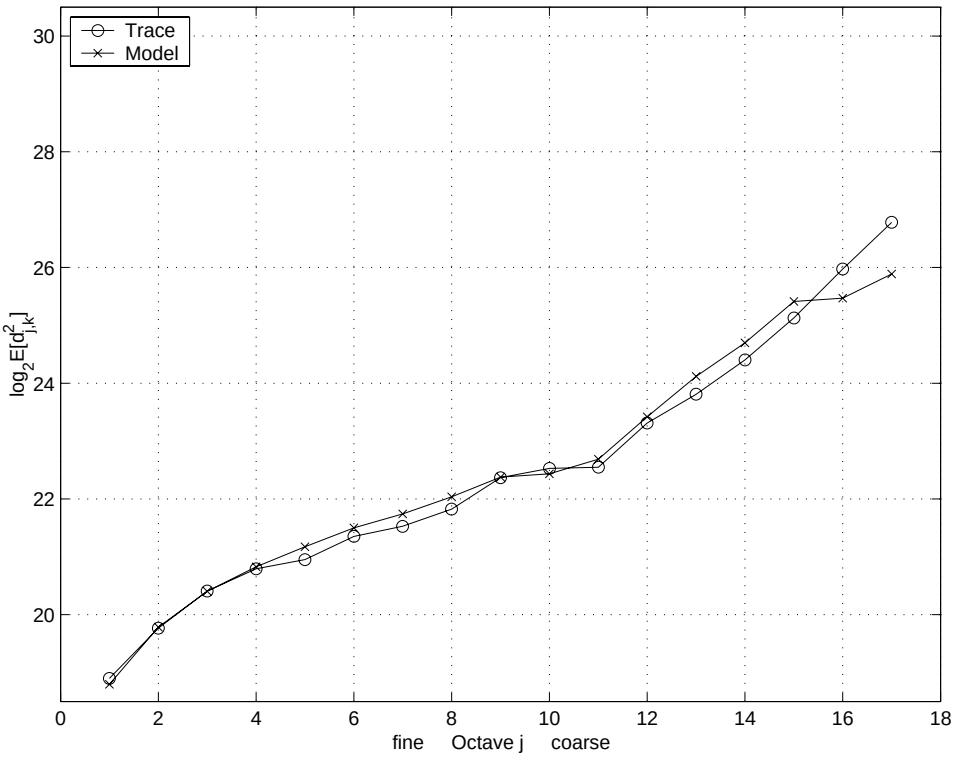

Figure 3.4: Logscale diagram of real Internet trace and synthetic trace

3.5 Higher Order Statistics

We employ wavelet-based multifractal analysis to analyze real and synthetic traffic traces. Unlike the *Logscale* diagram which only considers the second order statistics, this multifractal analysis method considers all moments of the stochastic process in the wavelet domain. The higher order statistics are extracted by the structure function $S(q, j)$ and the partition function $T(q)$ defined in [25] [77],

$$S(q, j) := \sum_{k=1}^{2^{(L-j)}} \|2^{-(L-j)/2} d_{j,k}\|^q$$

where $L := log_2$(Data Length). $T(q)$ is approximated by the slope of $log_2 S(q, j)$ when j is small. The multifractal spectrum $f(\alpha)$ is the Legendre Transform of $T(q)$:

$$f(\alpha) := \inf_q (q\alpha - T(q)).$$

The multifractal spectrum $f(\alpha)$ provides a measure of "frequency" of the singularity exponent $\alpha(t)$ at time t. It indicates the probability of a certain value of the singularity exponent:

$$Pr[\alpha(t) = \alpha] \approx 2^{-L(1-f(\alpha))}$$

For a monofractal process, like the fractional Gaussian noise (fGn, the increment of fractional Brownian motion), its singularity exponent $\alpha(t)$ is a constant H for every t. This might be considered as a degenerate case of multifractality. The corresponding partition function $T(q) = qH - 1$ of fGn is a linear function of q. Since the $\alpha(t)$ is equal to H for every t in fGn, its multifractal spectrum should be a single point at (H,1). We will employ the fGn as a pilot process and compare the multifractal spectrum with the real and synthetic traffic. For

a multifractal process, the partition function is a concave function of q and the singularity exponent $\alpha(t)$ has a wide range of values. In other words, there is a non-negligible probability that $\alpha(t)$ is equal to other values. Figure 3.5 shows the partition functions of the real trace, of the synthetic trace and of the fGn. The concave curves of the partition function show that the real traffic and synthetic traffic are multifractal processes and the partition function of fGn is a linear function due to its monofractal behavior. It is much clearer to see the difference in their multifractal spectra in Figure 3.6. The spectrum of fGn shows the probability $Pr[\alpha(t) = H] \approx 1$. For the real and synthetic traffic, their spectra show a rich variety of singularity exponents with a non-negligible probability. Moreover, the spectrum of our model is not only multifractal but also has the same shape as that of the real traffic.

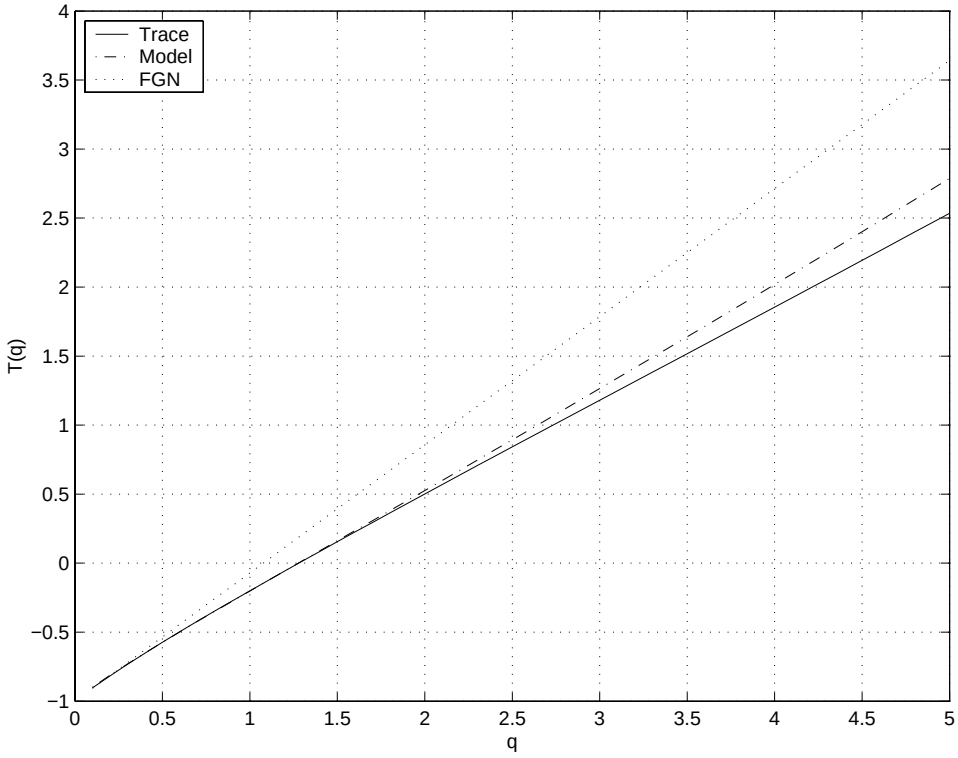

Figure 3.5: Partition Function $T(q)$ of Real Trace, Synthetic Traffic and fGn

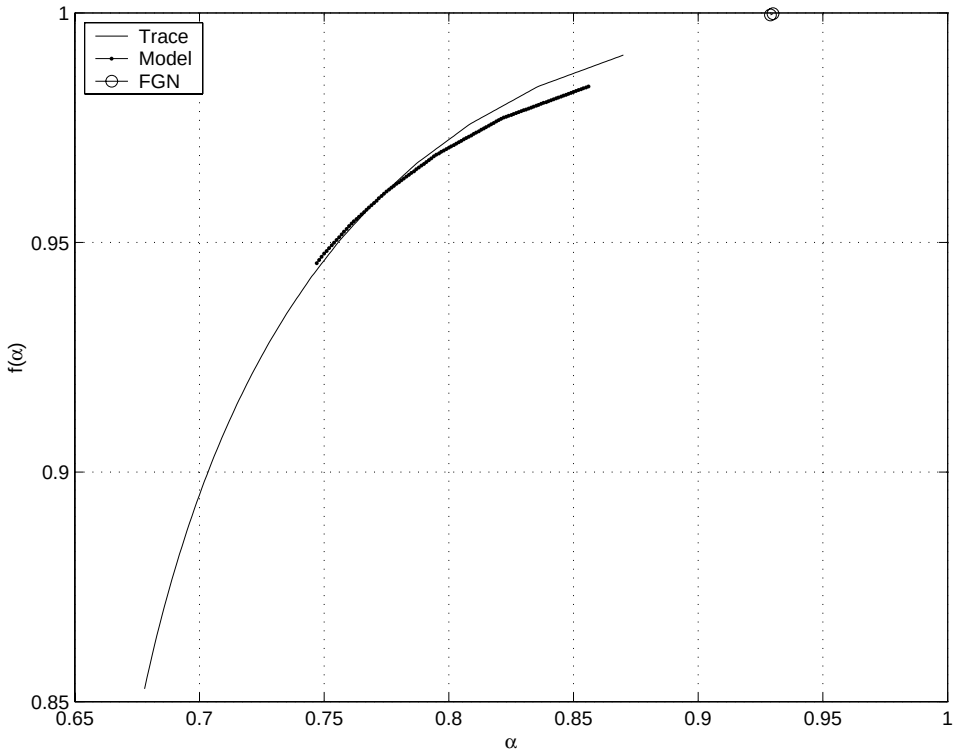

Figure 3.6: Multifractal Spectra of Real Trace, Synthetic Traffic and fGn

3.6 Queuing Behavior

After comparing the statistical properties of the real and synthetic traffic, we are also interested in their queuing behaviors. Figure 3.2 shows network topology and bottleneck router (Satellite Gateway) of the real DirecPC system. The arrival traffic of bottleneck router is aggregated downlink traffic (from Internet to user), which is modeled by our traffic model. In order to predict the queuing behavior of the bottleneck router, we consider a simple first-come-first-serve queuing system (FIFO) with a fixed service rate and an infinite buffer size as shown in Figure 3.7. Given the mean rate of arrival traffic, the queue length distributions under different utilization are obtained by properly adjusting the service rate. Figure 3.8 shows the steady state queue length tail distributions $Pr[Q > x]$ with various values of utilization $\rho = 0.6, 0.7, 0.8$ and 0.9. When the traffic load is heavy (large utilization), the real and synthetic traffic have almost the same distributions. With the light traffic load, the synthetic traffic also provides a good prediction for the queue length distribution when the queue length is less than 50K bytes. In the region of large queue lengths, the tail queue length distribution is overestimated. However, this event happens with a small probability due to the light traffic load.

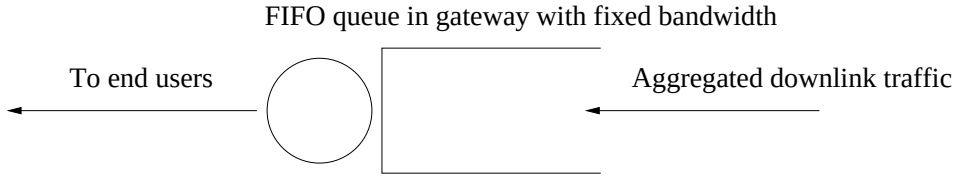

Figure 3.7: First-come-first-serve queue in satellite gateway

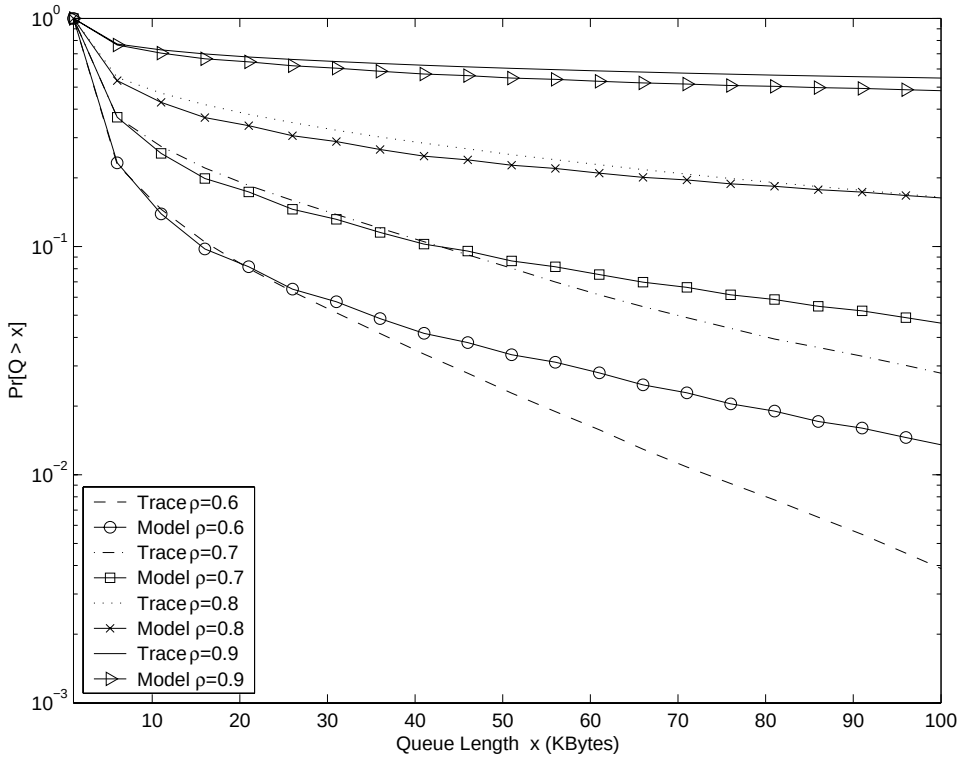

Figure 3.8: Queue Length Tail distribution of Real Trace and Synthetic Traffic with Utilization $\rho = 0.6$, 0.7, 0.8 and 0.9

3.7 Summary

In this chapter, we propose a multilevel ON/OFF model to capture the multifractal behavior of Internet traffic. The idea of this model is to mimic the arrival pattern of packets within a TCP section. The network parameters such as round-trip time are involved in the model. In the next chapter, we address that this model can be a useful tool to understand the impact of network parameters on connection performance. We develop a simple algorithm to estimate the parameter from a real traffic trace and generate a synthetic traffic. The second order and higher order statistics comparisons were performed by Logscale diagram and multifractal spectrum. Finally, we compare their queuing behaviors such as steady state distribution by simulations.

Chapter 4

Performance Analysis of Queuing Behaviors

4.1 Overview

In Chapter 3, we propose a multilevel ON/OFF model for Internet traffic. In this chapter, we focus on theoretic results of traffic statistical properties and its corresponding queuing behavior such as steady state queue length distribution, mean packet delay and jitter.

Since the Logscale diagram comprises important statistical properties of the traffic, we are looking for an explicit expression of Logscale diagram as a function of model parameters. Section 4.2 contains an approximation of the Logscale diagram by considering the partial integral with power spectrum density and the autocorrelation function of the traffic model. This explicit expression of Logscale diagram will help us understand the relationship between network parameters and performance of TCP connections.

Given the Logscale diagram and mean rate of traffic, we develop a recursive

method in Section 4.3. In this method we apply the properties of wavelet analysis to compute the variances of aggregate traffic workload. This simple transformation allows us to obtain the workload variance at every time scale from Logscale diagram. Furthermore, assuming that the traffic workload has a Lognormal distribution, one can easily predict the steady state queue length distribution by using the same technique provided in [78].

For most of real time applications such as voice and video conference, packet delay and jitter are important factors for connection performance. In Section 4.4, we discuss the delay and jitter of a CBR (Constant Bit Rate) connection in the bottleneck queue with background traffic. The CBR connection has a fixed packet interarrival time and fixed packet size. It is employed here as a probing process. According to our prediction of steady state queue length distribution, the mean CBR packet delay of a FIFO queue is obtained straightforward. We also develop a method to provide a tight upper bound of jitter according to the Logscale diagram of the background traffic. Both predictions of mean delay and jitter are validated by ns2 simulations. This fast algorithm could provide useful information for network management such as routing decision and resource allocation.

4.2 Analytical Solution of Logscale Diagram

We already demonstrated that the multilevel ON/OFF model has a good match in their statistical properties and queuing behaviors in Chapter 3. A question that arises naturally is whether one can find the relationship between network parameters and connection performance. The first step of our approach is to find an explicit expression of the Logscale diagram as a function of network parameters.

Recall the model definition in Chapter 3, the aggregate cumulative byte count

of the multilevel ON/OFF process is given in Definition 3.3.1 as

$$Y(t) = B \int_0^t \sum_{k=1}^{N} U_k(u) V_k(u) du. \tag{4.1}$$

The real traffic is modeled by the increment process X_i of $Y(t)$, which is defined in 3.2.4:

$$\begin{aligned} X_i &:= Y((i+1)\Delta) - Y((i)\Delta) \\ &= B \int_{i\Delta}^{(i+1)\Delta} \sum_{k=1}^{N} U_k(u) V_k(u) du, \quad i = 0, 1, 2, 3, ... \end{aligned} \tag{4.2}$$

Since X_i is the sum of N $i.i.d$ process $Z_{i,k}$,

$$Z_{i,k} := B \int_{i\Delta}^{(i+1)\Delta} U_k(u) V_k(u) du, \tag{4.3}$$

where $k = 1, 2, ..., N$. It is obvious that X_i and Z_i have the same autocovariance structure:

$$Cov(X_i, X_{i+n}) = NCov(Z_i, Z_{i+n}). \tag{4.4}$$

Our goal here is to calculate the Logscale diagram of X_i, which can be treated as the signal energy in a certain frequency range. In order to obtain the power spectrum density of the process X_i, we have to calculate the corresponding autocovariance $Cov(X_i, X_{i+n})$.

We consider the lower level ON/OFF process $V(t)$. Let the random variables T_{21} and T_{20} be the duration of ON and OFF with mean μ_1 and μ_0 respectively. The ON/OFF process starts at $t = -\infty$ and reaches the steady state before $t = 0$.

According to the Definition 3.3.7 and Theorem 3.3.8, the autocovariance function of $V(t)$ in the steady state is

$$r(t) := E[V(0)V(t)] - (E[V(0)])^2 = \frac{\mu_1}{\mu_1 + \mu_0}[\pi_{11}(t) - \frac{\mu_1}{\mu_1 + \mu_0}] \tag{4.5}$$

and

$$\pi_{11}(t) = G_{1c}(t) + \int_0^t F_{1c}(t-u)dH_{10}(u). \tag{4.6}$$

In order to obtain the power spectrum density function, we first consider its Laplace transform $\widehat{r}(s)$.

Proposition 4.2.1 *[16] [76] The Laplace transform of $\pi_{11}(t)$ is :*

$$\widehat{\pi}_{11}(s) = \widehat{G}_{1c}(s) + \widehat{h}_{10}(s)(1-\widehat{f}_1(s))/s \tag{4.7}$$

where

$$\begin{aligned} \widehat{f}_i(s) &:= \mathscr{L}\{f_i(t)\} && (4.8)\\ \widehat{G}_{1c}(s) &:= \mathscr{L}\{\int_t^\infty \frac{F_{1c}(u)}{\mu_1}du\} \\ &= \frac{1}{\mu_1}\mathscr{L}\{\mu_1 - \int_0^t F_{1c}(u)du\} \\ &= \frac{1}{\mu_1}\mathscr{L}\{\mu_1 - \int_0^t [1-\int_0^u f_1(x)dx]du\} \\ &= \frac{1}{s} - \frac{1-\widehat{f}_1(s)}{\mu_1 s^2} && (4.9) \end{aligned}$$

and

$$\widehat{h}_{10}(s) = \frac{(1-\widehat{f}_1(s))\widehat{f}_0(s)}{\mu_1 s(1-\widehat{f}_1(s)\widehat{f}_0(s))}. \tag{4.10}$$

Note that the $f_i(t)$, $i = 0, 1$ is the pdf of the ON and OFF duration respectively. The Laplace transform of $r(t)$ is

$$\widehat{r}_v(s) = \frac{\mu_1\mu_0}{(\mu_1+\mu_0)^2 s} - \frac{(1-\widehat{f}_1(s))(1-\widehat{f}_0(s))}{(\mu_1+\mu_0)s^2(1-\widehat{f}_1(s)\widehat{f}_0(s))}. \tag{4.11}$$

Case 1: Exponential distribution at ON duratoin

If the ON/OFF periods T_{21}/T_{20} have the Exponential distributions with mean

$1/\lambda_1 = \mu_1$ and $1/\lambda_0 = \mu_0$ respectively, we have the density function and the Laplace transform of the ON and OFF duration as follows:

$$f_i(t) = \lambda_i \exp(-\lambda_i t) \tag{4.12}$$

$$\widehat{f}_i(s) = \frac{\lambda_i}{s+\lambda_i}, \tag{4.13}$$

where $i = 1$ and 0. The corresponding $\widehat{r}(s)$, $r(t)$ and the power spectrum density $S(\omega)$ of this Exponential ON/OFF process are

$$\widehat{r}(s) = \frac{\mu_1\mu_0}{(\mu_1+\mu_0)^2}\frac{1}{s+\lambda} \tag{4.14}$$

$$r(t) = \frac{1}{\lambda(\mu_1+\mu_0)}e^{-\lambda t} \tag{4.15}$$

$$S(\omega) = \int_{-\infty}^{\infty} r(t)e^{-j\omega t}dt = \frac{2}{(1/\lambda_1+1/\lambda_0)(\omega^2+\lambda^2)}, \tag{4.16}$$

where $\lambda = \lambda_1 + \lambda_0$. With the Proposition 2.2.5, we have the approximation of $E[d_{j,k}^2]$ by assuming that $\Psi(\omega)$ is an ideal bandpass function:

$$E[d_{j,k}^2] \approx 2\int_{\pi/2^j}^{\pi/2^{j-1}} S(\omega)2^j\|\Psi(2^j\omega)\|^2 d\omega \tag{4.17}$$

$$\approx \frac{2^{j+2}}{\lambda(1/\lambda_1+1/\lambda_0)}(arctan\frac{\pi/\lambda}{2^{j-1}} - arctan\frac{\pi/\lambda}{2^j}). \tag{4.18}$$

Take the logarithm on both sides, the Logscale diagram of single ON/OFF process is

$$\widetilde{L}_j := log_2 E[d_{j,k}^2] \approx log_2\{\frac{2^{j+2}}{\lambda(1/\lambda_1+1/\lambda_0)}(arctan\frac{\pi/\lambda}{2^{j-1}} - arctan\frac{\pi/\lambda}{2^j})\} \tag{4.19}$$

Recall that there are N independent connection with Burst rate B and the upper level has a ON/OFF ratio R_1. The Logscale diagram of the aggregate traffic at small time scale region is approximated by

$$L_j \approx \widetilde{L}_j + log_2(N(BR_1)^2). \tag{4.20}$$

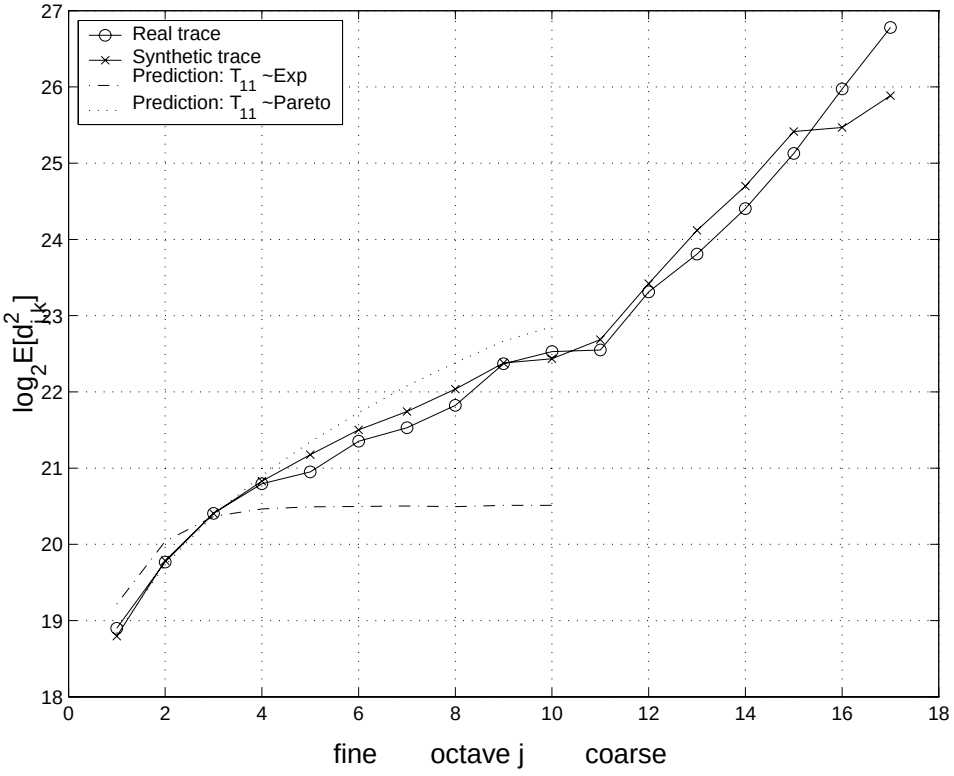

Figure 4.1: Logscale diagram of Real Trace and Synthetic Traffic

Since both ON and OFF duration have Exponential distributions, the autocovariance function $r(t)$ at (4.15) also has an exponential form which indicates a short-range dependence property. The Logscale diagram of this short range dependent ON/OFF process is shown in Figure 4.1 with the same mean ON time and OFF time estimated in Chapter 3. Note that the zero slope of the Logscale diagram at large time scales demonstrates the absence of long range dependence and obviously the Exponential distribution assumption of ON/OFF duration is not valid for modeling a real traffic. The heavy-tailed type distribution such as Pareto distribution is suggested as a candidate for modeling the long-range dependence traffic [16] [1] [66]. However, there is no closed-form for the autocovariance function of such ON/OFF process with Pareto distribution. We provide an approximation for the Logscale diagram of the ON/OFF model.

Case 2: Pareto distribution at ON duration

The density function of $Pareto(K, a)$ distribution $f_P(t) = aK^a t^{-a-1}$ can be approximated (truncated tail) [79] by the weighted sum of N density functions of Exponential distribution with mean γ^n/ν, $n = 0, 1, ..., N-1$. Define $r.v.$ Y_N:=$\sum_{n=0}^{N-1} X_n$, where $r.v.$ X_n has an Exponential density function:

$$f_{X_n} := \frac{\nu}{\gamma^n} \exp(-\frac{\nu}{\gamma^n} t). \tag{4.21}$$

Then the pdf of Y_N is:

$$f_{Y_N}(t) \quad = \quad G(N) \sum_{n=0}^{N-1} \theta^n \frac{\nu}{\gamma^n} \exp(-\frac{\nu}{\gamma^n} t), \tag{4.22}$$

where parameter θ is the weighting factor. Note that$\int_0^\infty f_{Y_N}(t) = 1$ for probability density function, we have the normalization factor $G(N) = \frac{1-\theta}{1-\theta^N}$.

Recall that the α^{th} moments of Pareto distribution is diverging when α is

greater or equal to the shaping parameter a. i.e.

$$\begin{aligned} E[P^{\alpha}] &= \int_{K}^{\infty} t^{\alpha} f_P(t) dt \\ &= aK^a \int_{K}^{\infty} t^{-a-1+\alpha} dt \end{aligned} \tag{4.23}$$

diverges if $\alpha \geq a$. Hence, we consider the a^{th} moments of $r.v.$ Y_N:

$$E[Y_N^a] = G(N) \frac{1-(\theta\gamma^a)^N}{1-\theta\gamma^a} E[X_0^a]. \tag{4.24}$$

Let $N \to \infty$, we have

$$\lim_{N\to\infty} E[Y_N^a] = \frac{1-\theta}{1-\theta\gamma^a} E[X_0^a]. \tag{4.25}$$

Since X_0 has Exponential distribution, we have $E[X_0^a] < \infty$ for all $a < \infty$. However, if $\theta\gamma^a \geq 1$, the limit in (4.25) diverges. We are free to choose the scaling factor to be $\gamma = 2$, then the weighting factor $\theta = 2^{-a}$. The mean of the mixed density function is $E[Y_N] = G(N) \sum_{n=0}^{N-1} \theta^n \frac{\gamma^n}{\nu}$. In order to match the mean of the Pareto distribution $aK/(a-1)$, we have

$$\nu = G(N) \frac{1-(\theta\gamma)^N}{1-\theta\gamma} \frac{a-1}{aK}. \tag{4.26}$$

The Pareto distribution approximation by sum of weighted Exponential distributions are shown in Figure 4.2.

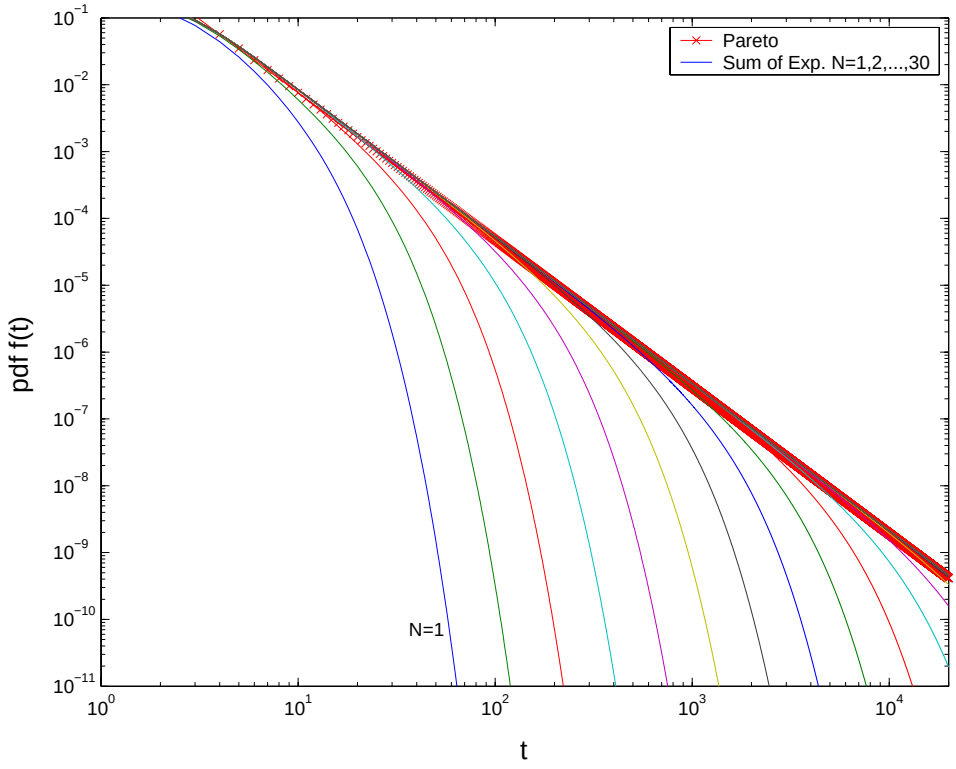

Figure 4.2: Probability density function of Pareto and sum of weighted Exponential distributions with N=1,2,...,30.

The Laplace transform of $\widehat{f}_{Y_N}(s)$ is:

$$\widehat{f}_{Y_N}(s) = G(N)\sum_{n=0}^{N-1}\theta^n\frac{\nu/\gamma^n}{s+\nu/\gamma^n}. \tag{4.27}$$

Substitute $\widehat{f}_0(s) := \frac{\lambda_0}{s+\lambda_0}$ and $\widehat{f}_1(s) := \widehat{f}_{Y_N}(s)$ into equation (4.11), the Laplace transform of the autocovariance function is in (4.28).

$$\widehat{r}_v(s) = \frac{\mu_1\mu_0}{(\mu_1+\mu_0)s} - \frac{1 - G(N)\sum_{n=0}^{N-1}\theta^n\frac{\nu/\gamma^n}{s+\nu/\gamma^n} - \frac{\lambda_0}{s+\lambda_0} + G(N)\sum_{n=0}^{N-1}\theta^n\frac{\lambda_0\nu/\gamma^n}{\lambda_0-\nu/\gamma^n}(\frac{1}{s+\nu/\gamma^n} - \frac{1}{s+\lambda_0})}{(\mu_1+\mu_0)s^2[1 - G(N)\sum_{n=0}^{N-1}\theta^n\frac{\lambda_0\nu/\gamma^n}{\lambda_0-\nu/\gamma^n}(\frac{1}{s+\nu/\gamma^n} - \frac{1}{s+\lambda_0})]} \tag{4.28}$$

The power spectrum density of $r_v(t)$ can be calculated by the following relation:

$$S_v(\omega) = \widehat{r}_v(s)|_{s=-j\omega} + \widehat{r}_v(s)|_{s=j\omega}. \tag{4.29}$$

Assuming that the mother wavelet ψ is an idea bandpass function, the Logscale diagram $\widetilde{L}_j$ of a single connection at small time scales can be approximated as follows:

$$\begin{aligned}\widehat{L}_j &= log_2E[d_{j,k}^2] \\ &\approx log_2\{\int_{-\pi}^{\pi} S_v(\omega)2^j\|\Psi(2^j\omega)\|^2d\omega\}\}.\end{aligned} \tag{4.30}$$

Again, the Logscale diagram of the aggregate traffic is:

$$L_j \approx \widehat{L}_j + log_2(N(BR_1)^2). \tag{4.31}$$

The dash and dot lines at Figure 4.1 compare the theoretical approximations of Logscale diagram with Exponential (Case 1) and Pareto distribution (Case 2) respectively. It shows that the equation 4.31 provides a good approximation for

the Logscale diagram at small time scales. With this formula, we are able to understand the relationship between network parameters and the traffic statistical properties. Figure 4.3 and 4.4 also show the Logscale diagrams of numerical approximation and synthetic traffic with different network parameters. Figure 4.3 shows the effect of increasing the burst size K_{21} by two times and four times. In Figure 4.4 both ON(T_{21}) and OFF (T_{20}) time periods are increased in order to keep the same mean rate. In those two figures both synthetic and theoretical results predict the statistical properties of real traffic with different network parameters.

The second step of our approach is to find the queuing behaviors according to the Logscale diagram of aggregate traffic. Given the Logscale diagram (numerically estimated from the trace or theoretically computed from the model), we develop an efficient algorithm in the next section to predict the queue length distribution and the corresponding connection performance such as delay and jitter. With this technique, one can employ the network parameters to predict the queuing behavior of a bottleneck router. It builds up a relationship between the network parameter and the connection performance, which is very helpful in protocol design. Furthermore, based on the prediction of the queuing behavior, one may adjust control parameters in a router such as the maximum threshold and the dropping probability of the RED (Random Early Detection) scheme. We will discuss these problems in the following chapters.

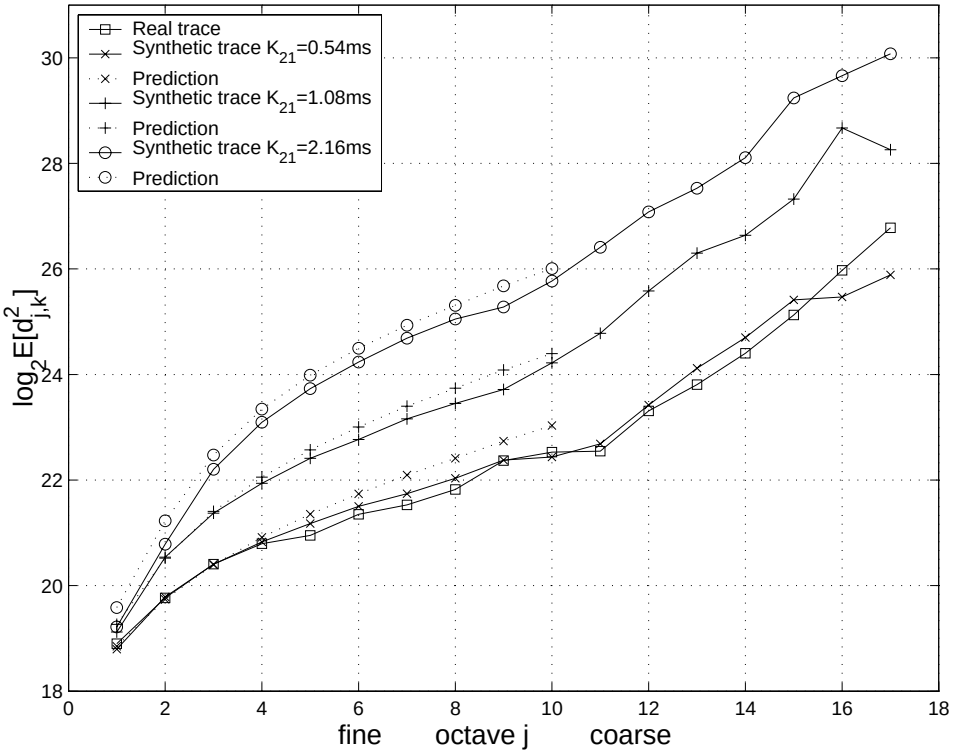

Figure 4.3: Logscale diagram of Real Trace and Synthetic Traffic

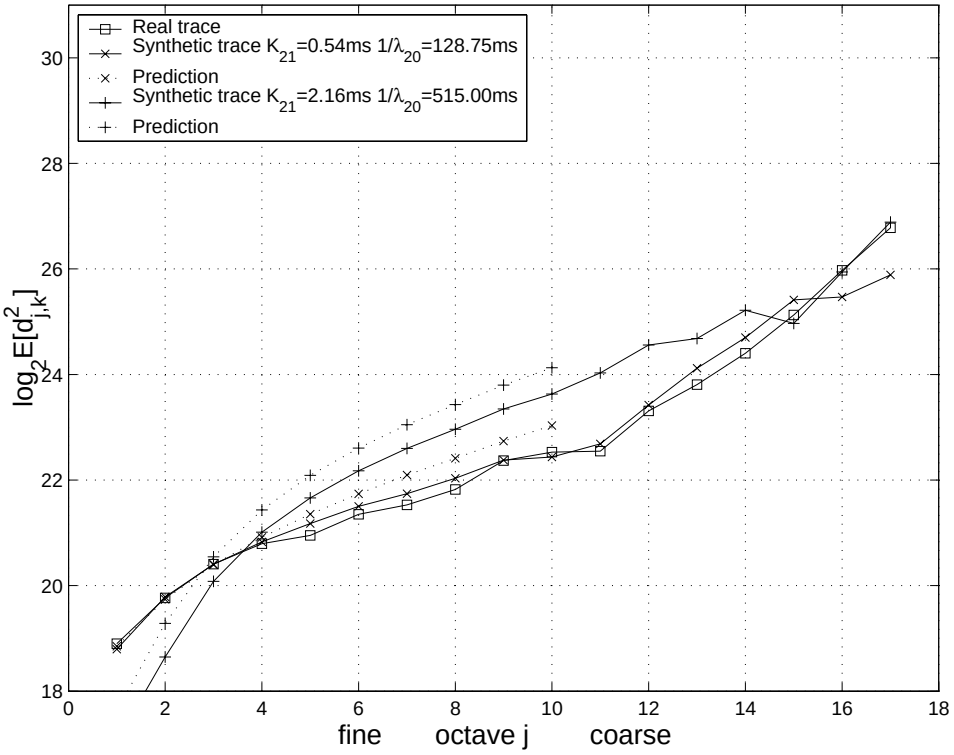

Figure 4.4: Logscale diagram of Real Trace and Synthetic Traffic

4.3 Approximation of Steady State Queue Length Distribution

As shown in the Chapter 2, the wavelet method has many advantages in parameter estimation and traffic analysis. Given the Logscale diagram (numerically or theoretically) of the arrival traffic, we develop an efficient method to obtain an upper bound of the overflow probability by using the properties of wavelet.

Proposition 4.3.1 *Let $a_{j,k}$ and $d_{j,k}$ be the approximation coefficient and the detail coefficient of the Haar wavelet, we have the following relations according to Remark 2.1.7*

$$a_{j,2k} = \frac{a_{j+1,k} + d_{j+1,k}}{\sqrt{2}} \tag{4.32}$$

$$a_{j,2k+1} = \frac{a_{j+1,k} - d_{j+1,k}}{\sqrt{2}}. \tag{4.33}$$

With the uncorrelated assumption between $a_{j+1,k}$ and $d_{j+1,k}$ for each j, we have

$$Var[a_j] = \frac{Var[a_{j+1}] + Var[d_{j+1}]}{2}. \tag{4.34}$$

Note that the plot $\log_2 Var[d_j]$ $v.s$ j is the *Logscale* diagram. On the other hand, let the workload A_j to be the total arrival bytes in the interval $[0, 2^j\Delta)$. From the definition of Haar wavelet, we have

$$A_j = a_j 2^{j/2}. \tag{4.35}$$

Thus, by using the *Logscale* diagram and the variance of workload at the finest scale $Var[A_0]$ as the initial values, the variance of workload $Var[A_j]$ can be computed recursively for all j.

Proposition 4.3.2 *Given the Logscale diagram* $L_j := log_2 Var[d_j]$ *and the variance of* A_0*, the variance of* A_j*,* $j = 1, 2, ...$ *is*

$$\begin{aligned} Var[A_j] &= 2^j Var[a_j] \\ Var[a_j] &= 2Var[a_{j-1}] - Var[d_j]. \end{aligned}$$

Since the workload A_j is contributed by many independent connections which are active in the the interval $[0, 2^j\Delta)$, The Central Limit Theory suggests that the candidate distribution of workload should be Normal distribution with parameters $(EA_j, Var[A_j])$. Figure 4.6 shows that the Normal distribution works well at large time scales in fitting the empirical distribution. However, the packet arrival pattern is very bursty and spare at small time scales. There are few connections have packets arriving in a small interval. The assumption of large number aggregation of independent connections is invalid. On the other hand, the Normal distribution always has a positive probability for negative values. This property is not suitable for modeling a positive random variable such as traffic workload. We are looking for a distribution of positive random variable which can best fit the empirical distribution of real traffic at all time scale. We applyied the *Kolmogorov-Smirnov* Goodness-of-Fit test [80] [81] [82] to most well-known distributions of positive random variables and found the Lognormal distribution is a good choice.

Assuming that the workload A_j has the Lognormal distribution for all j with mean M_j and variance V_j, the probability density function of the Lognormal distribution is:

$$f_{A_j}(x) := \frac{1}{x\sigma_j\sqrt{2\pi}} \exp[-\frac{(\ln x - \mu_j)^2}{2\sigma_j^2}], \quad x > 0 \tag{4.36}$$

Since the r^{th} moment of the Lognormal distribution has a closed-form:

$$EA_j^r = \exp(r\mu_j + \frac{r^2\sigma_j^2}{2}). \tag{4.37}$$

The parameters μ_j and σ_j can be easily obtained:

$$\sigma_j^2 = \ln(\frac{M_j^2 + V_j}{M_j^2}) \tag{4.38}$$

$$\mu_j = \ln(M_j) - \frac{\sigma_j^2}{2}. \tag{4.39}$$

Figure 4.5 and Figure 4.6 are the complement CDF of the real workload with scale $j = 1, 2, .., 14$ and the corresponding Normal and Lognormal distribution fitted by the estimated mean and variance. It is obvious that the prediction of Normal distribution is severely underestimated at small time scales.

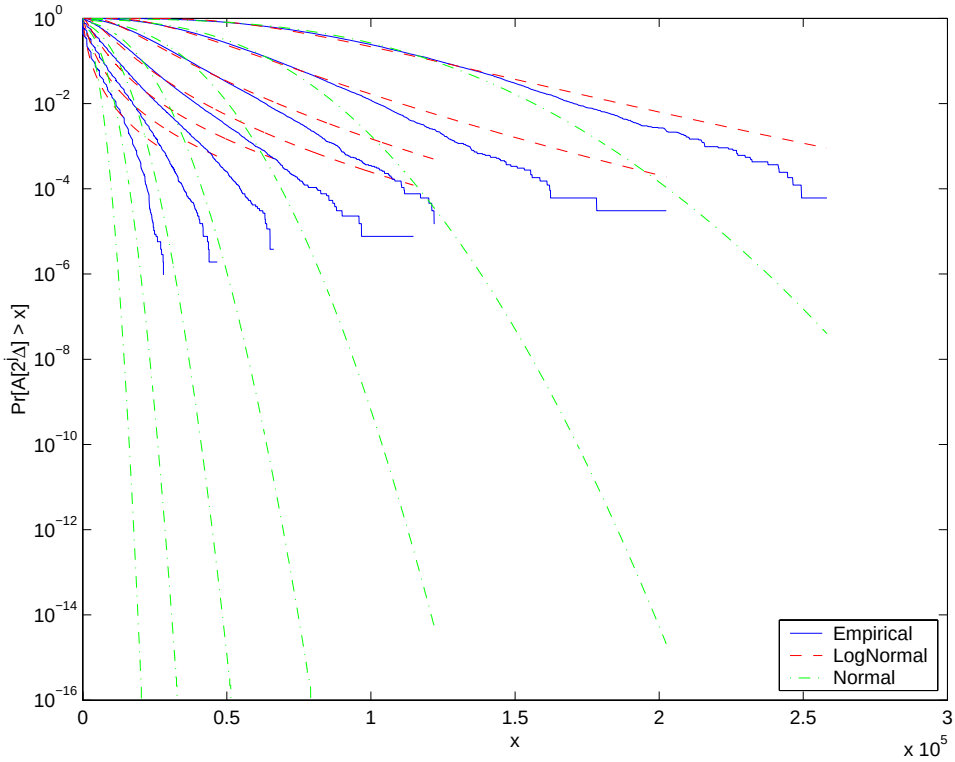

Figure 4.5: The CCDF of workload A_j $j = 1, 2, ..., 7$ and the fitted Normal and Lognormal distribution.

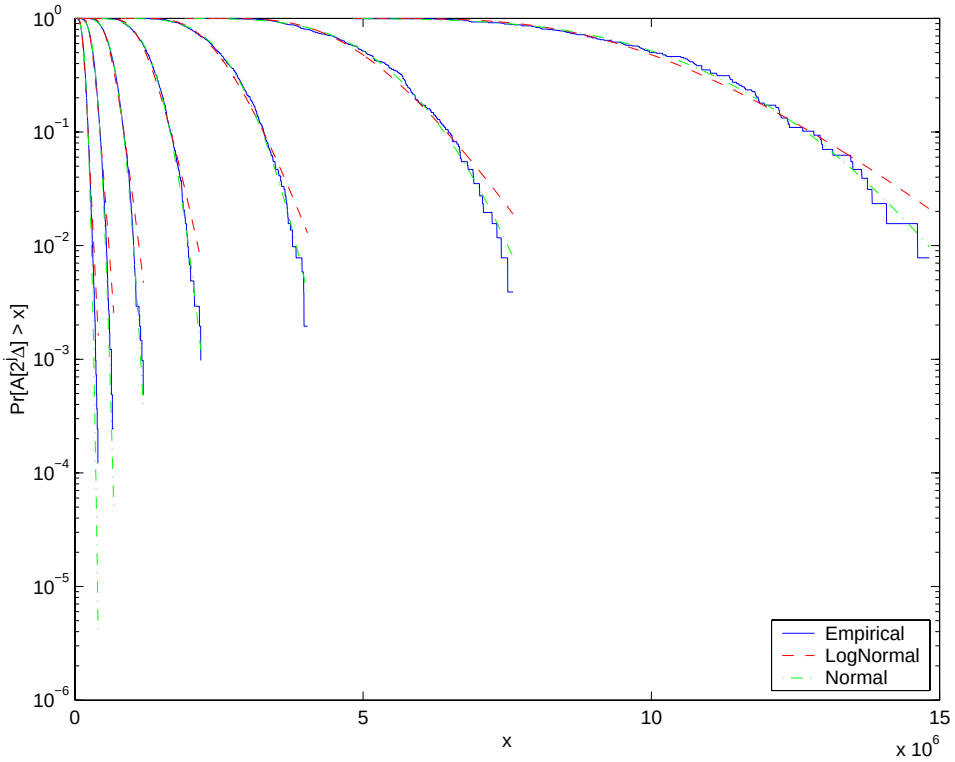

Figure 4.6: The CCDF of workload A_j $j = 8, 9, ..., 14$ and the fitted Normal and Lognormal distribution.

Consider a simple FIFO queue with a finite buffer size B (bytes) and capacity C (bytes/Δ). Assuming that the distribution of workload A_j is known for all j, Riedi [78] proposes a simple upper bound for the overflow probability $P[Q > B]$, where $r.v.$ Q is the queue length in the steady state. Let the event $E_j := \{A_j < B+C2^j\Delta\}$ be assumed to be independent of each other. According to the lemma proved by Riedi:

Lemma 4.3.3 *[78] Assume that the event $E_j = \{S_j < b_j\}$, where $S_j = R_0 + ... + R_{j-1}$ for $1 \leq j \leq n$ and where R_0,...,R_n are independent, otherwise arbitrary random variables. Then, for $1 \leq j \leq n$, we have*

$$P[E_j|E_{j-1}, ..., E_0] \geq P[E_j]. \tag{4.40}$$

With Lemma 4.3.3, we have the upper bound of the overflow probability:

Proposition 4.3.4 *Let $E_j := \{A_j < B + C2^j\Delta\}$. An upper bound of the overflow probability of a FIFO queue is*

$$\begin{aligned} p[Q > B] &= 1 - P[Q \leq B] \approx 1 - P[\cap_{j=1}^{K} E_j] \\ &= 1 - P[E_0] \prod_{j=1}^{K} P[E_j|E_{j-1}, ..., E_0] \\ &\leq 1 - \prod_{j=0}^{K} P[E_j] = 1 - \prod_{j=0}^{K} P[A_j < B + C2^j\Delta], \end{aligned} \tag{4.41}$$

where K is the maximum octave and $2^K\Delta$ is the maximum time scale.

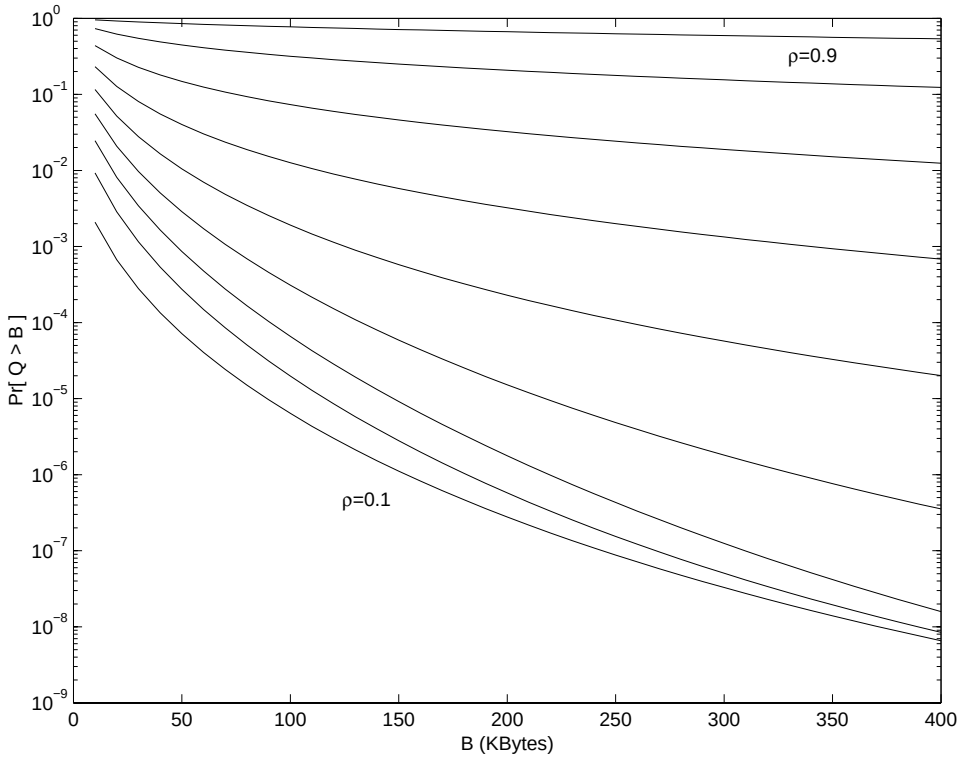

Figure 4.7: The upper bound of overflow probability with utilization $\rho = 0.1, 0.2, ..., 0.9$.

Figure 4.7 shows this upper bound with different bandwidth utilization $\rho = 0.1, 0.2, ..., 0.9$. Figure 4.8 to 4.11 are the comparisons of this upper bound with the queue length distribution of the real network traffic and the synthetic traffic. These results suggest that the upper bound of overflow probability may provide a good approximation of the queue length distribution. In the next section, we will use this approximation to predict the mean queuing delay and choose the RED parameter properly to satisfy the given mean delay requirement.

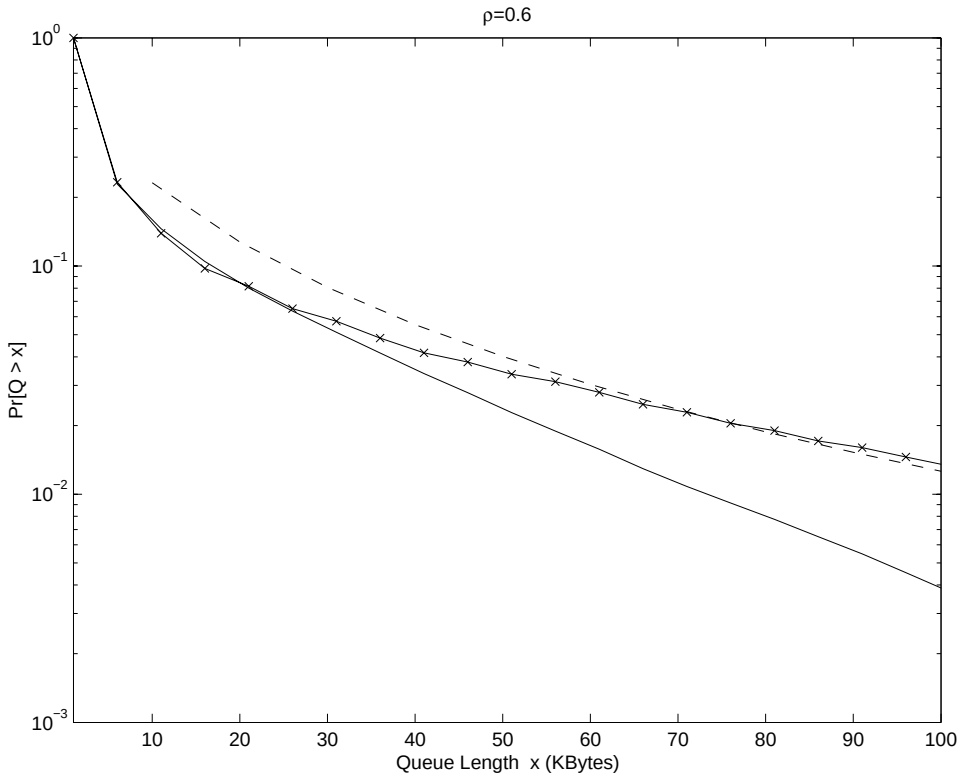

Figure 4.8: The queue length distribution with utilization $\rho = 0.6$, upper bound(dash), real trace(solid) and synthetic traffic(cross).

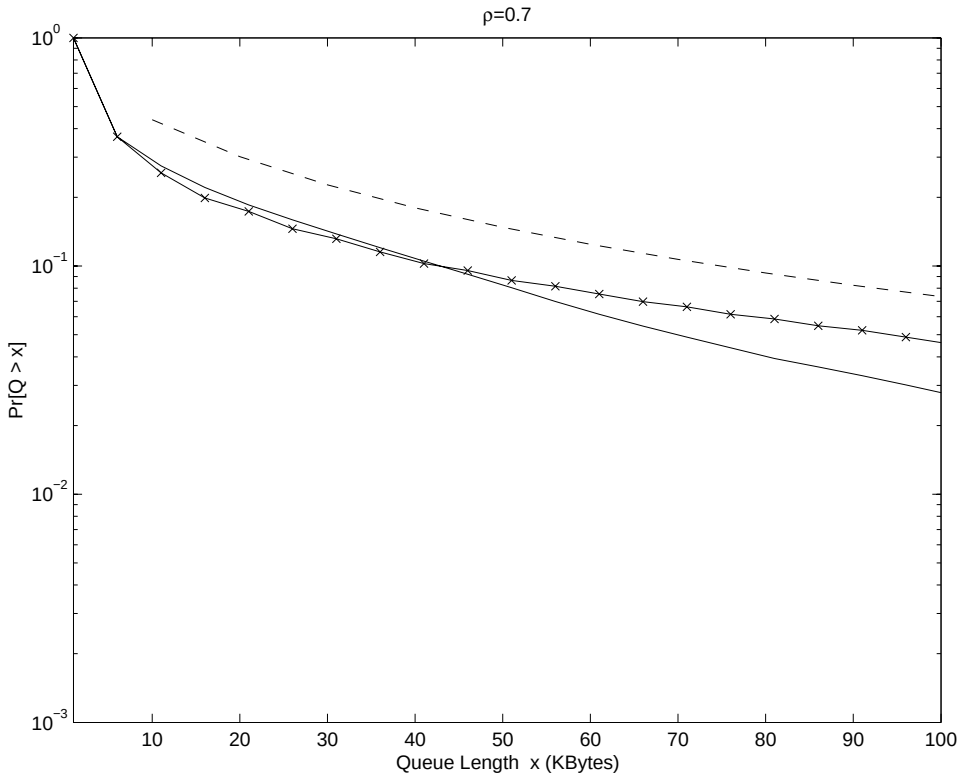

Figure 4.9: The queue length distribution with utilization $\rho = 0.7$, upper bound(dash), real trace(solid) and synthetic traffic(cross).

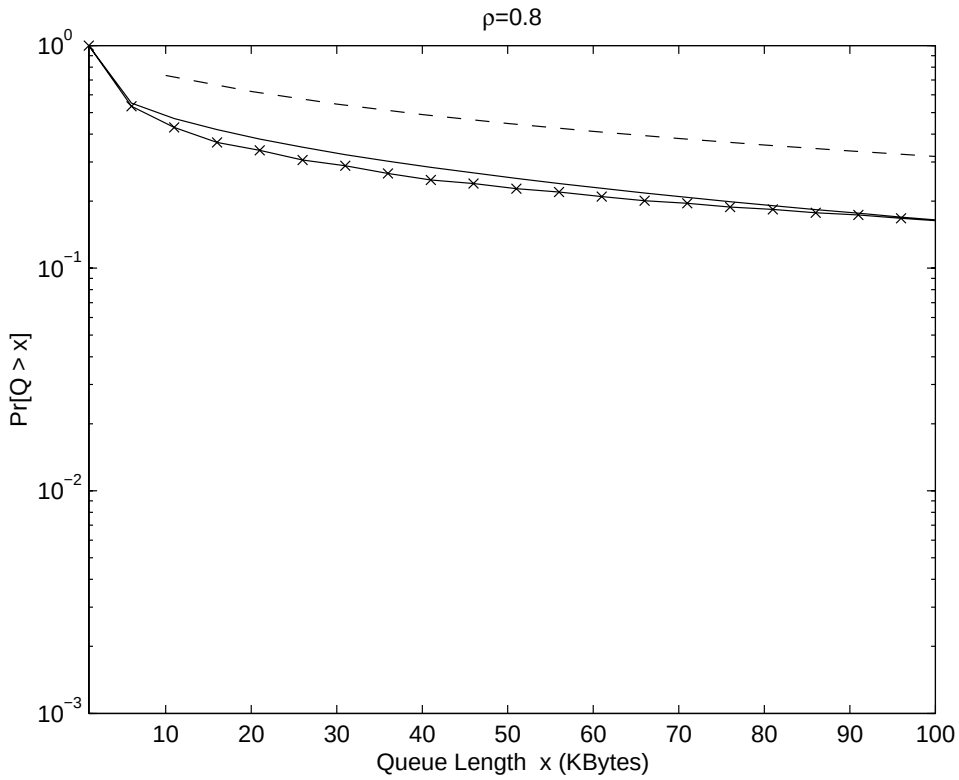

Figure 4.10: The queue length distribution with utilization $\rho = 0.8$, upper bound(dash), real trace(solid) and synthetic traffic(cross).

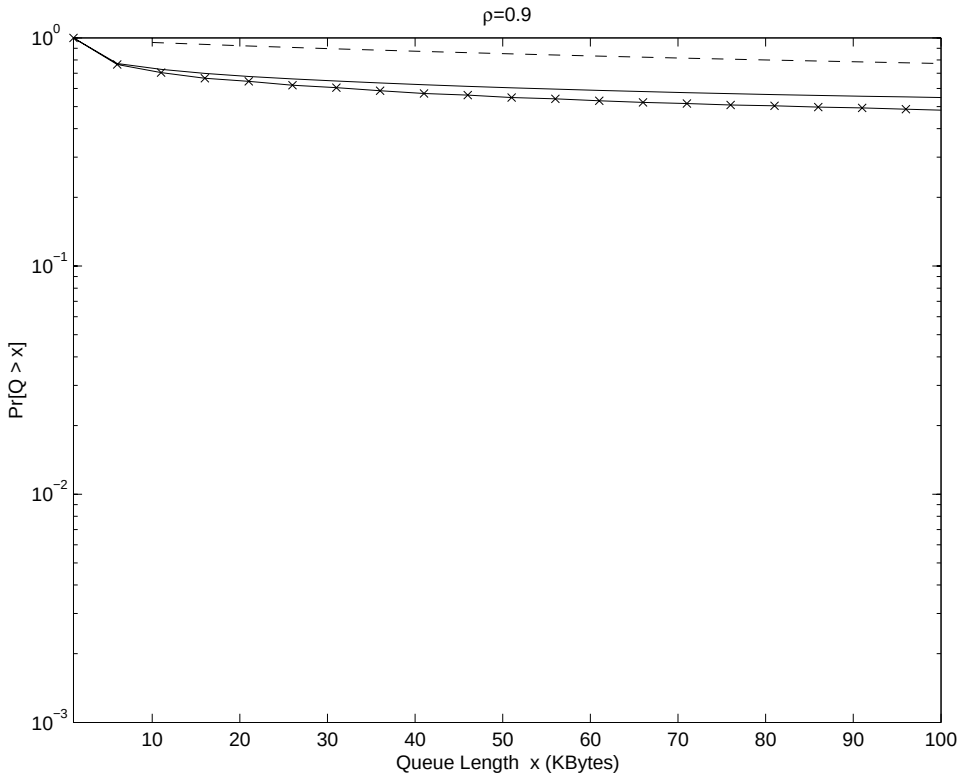

Figure 4.11: The queue length distribution with utilization $\rho = 0.9$, upper bound(dash), real trace(solid) and synthetic traffic(cross).

When we calculate the upper bound of the overflow probability, we find that only one workload A_j at certain time scale j dominates the overflow probability. Figure 4.12 shows that in the light traffic condition ($\rho = 0.3$), the traffic perturbation at small time scales mainly causes the overflow events. Figure 4.13, 4.14 and 4.15 show that when the traffic load is increasing, the overflow probability of the queue will be mainly dominated by traffic behavior at larger time scales. On the other hand, with a fixed utilization, the dominating time scale is related to the buffer size. The smaller buffer size has a smaller dominating time scale. It implies that the short range dependence in the traffic dominates the queuing behavior when the buffer size is small. Note that most of dominating time scales are less than 1 *sec*. The statistical properties of the traffic at small time scale is directly related to the protocol behavior. It suggests that, for the small buffer queue, the protocol behavior plays a more important role than the file size distribution does at the upper level. It also suggests that the monofractal or single level ON/OFF model is not quite suitable for predicting the behavior of a small buffer queue.

The selection of maximum octave K also determines the upper bound. Riedi [78] suggested that the maximum time scale $2^K\Delta$ should be large enough to contain an empty event. Figure 4.12 to 4.15 show that the upper bound will converge as K goes to infinite. In practice, the maximum octave K can be chosen large enough so that the probability $P[E_j]$ converges to 1.

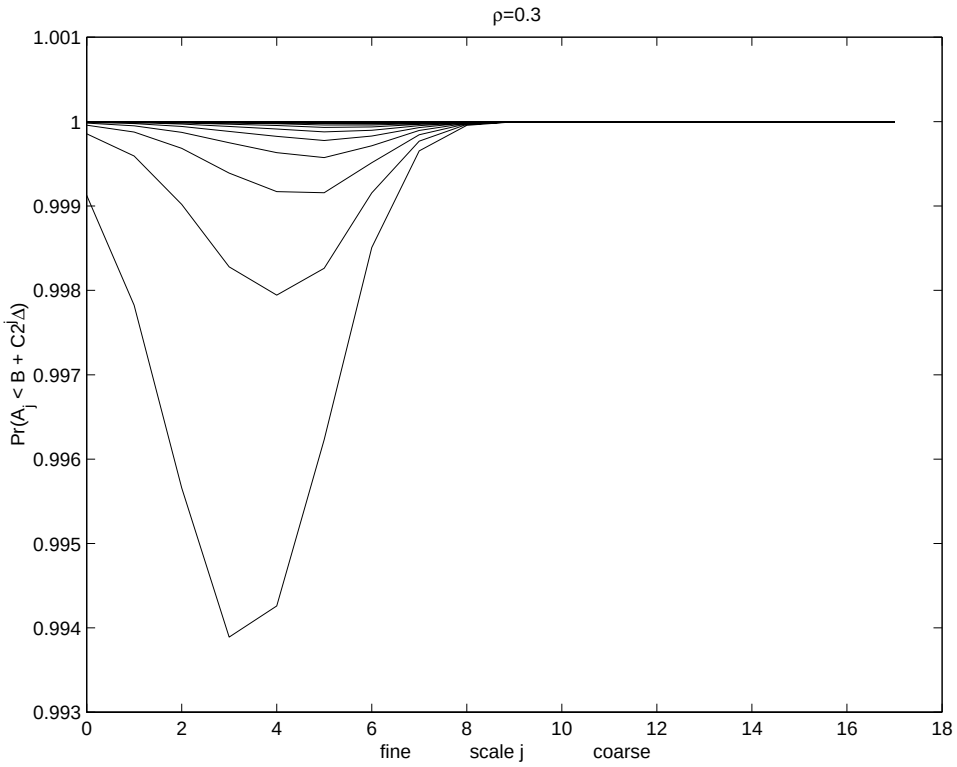

Figure 4.12: The $Pr[A_j < B + 2^j C\Delta]$ *v.s.* octave j, buffer size $B = 10K, 20K, ..., 400K$ (bytes) with utilization ρ=0.3

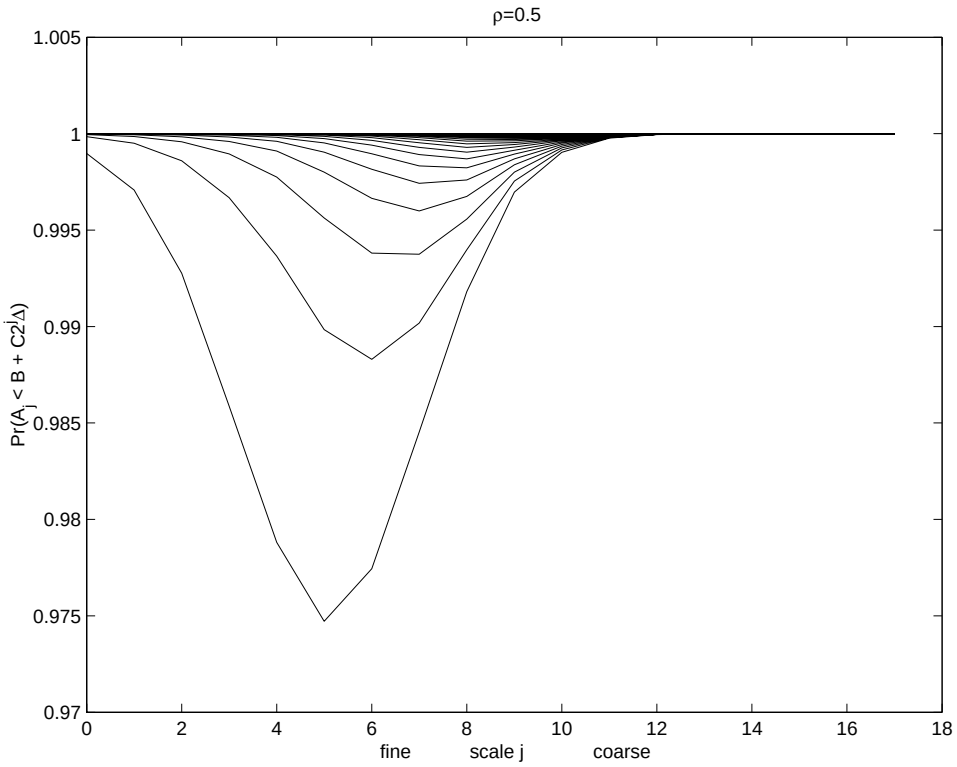

Figure 4.13: The $Pr[A_j < B + 2^j C\Delta]$ *v.s.* octave j, buffer size $B = 10K, 20K, ..., 400K$ (bytes) with utilization ρ=0.5

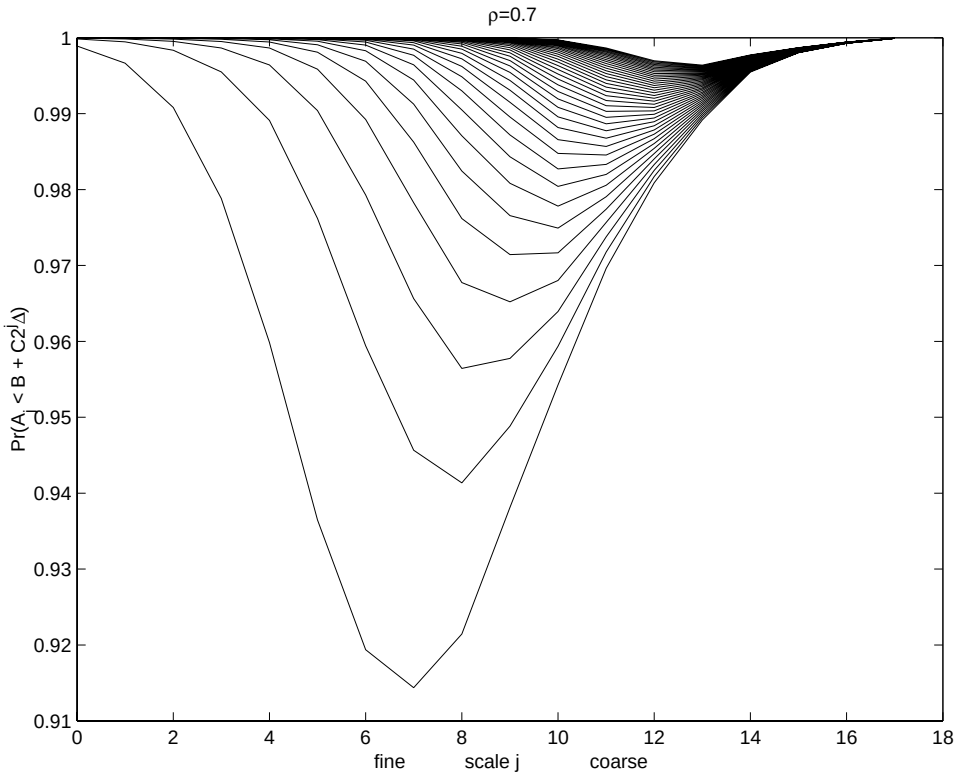

Figure 4.14: The $Pr[A_j < B + 2^j C\Delta]$ $v.s.$ octave j, buffer size $B = 10K, 20K, ..., 400K$ (bytes) with utilization ρ=0.7

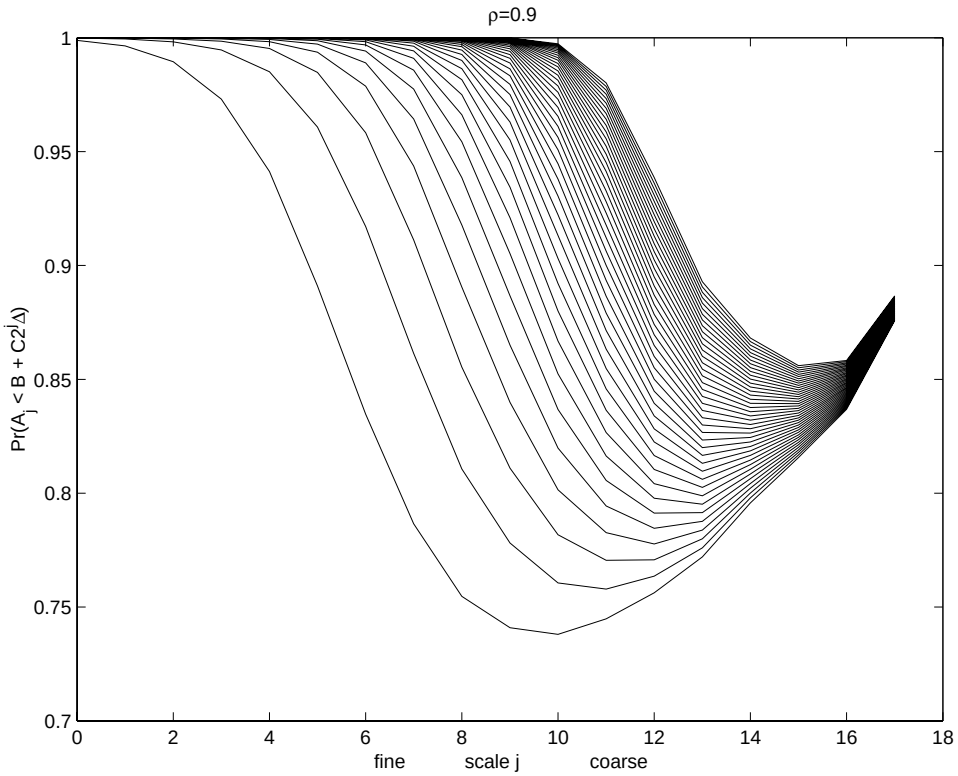

Figure 4.15: The $Pr[A_j < B + 2^j C\Delta]$ $v.s.$ octave j, buffer size $B = 10K, 20K, ..., 400K$ (bytes) with utilization ρ=0.9

4.4 Queuing Delay and Jitter Analysis

With the prediction of steady state queue length distribution, it is straightforward to predict the average queuing delay by calculating the ratio of mean queue length and service rate. We consider a constant-bit rate traffic as our probing process. As shown in Figure 4.16, the CBR process sends out a small packet every $2^n \Delta sec$ persistently and shares the bandwidth with the background traffic. Assume that the packet size of the CBR process is set to be small enough such that mean rate of CBR traffic is negligible. According to the approximation of steady state queue length distribution at , the average queuing delay $\bar{D}$ of CBR packet can be predicted by the following proposition.

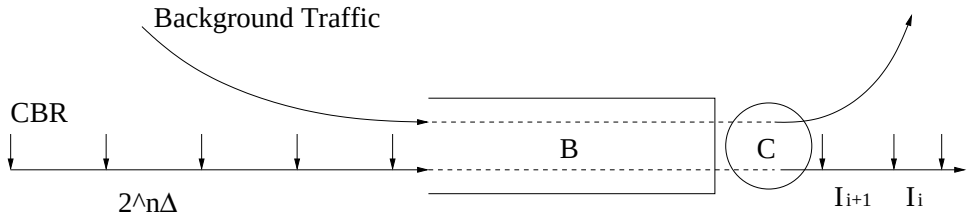

Figure 4.16: The target process and the background traffic

Proposition 4.4.1 *Assume a FIFO queue is in the steady state with (time average) distribution $Pr[Q > b]$. The average queuing delay of CBR packets is:*

$$\bar{D} = \frac{1}{C} \sum_{b>0} Pr[Q > b], \tag{4.42}$$

where $\sum_{b>0} Pr[Q > b]$ is equal to the mean (time average) queue length of the FIFO queue with infinite buffer size.

Sketch of Proof. Since the CBR packets arrive at the queue every $2^n \Delta sec$, the average queue length seen by the CBR packets is equal to the time average of

queue length. The average packet delay at a FIFO queue:

$$\bar{D} = \bar{S} + \bar{W} \tag{4.43}$$

$$= \frac{1}{C} \sum_{b>0} Pr[Q > b], \tag{4.44}$$

where $\bar{W}$ and $\bar{S}$ mean waiting time and mean service time respectively. Note that the CBR packet size is assumed to be small and negligible, we assume $\bar{S} = 0$.

□

Figure 4.17 to 4.23 demonstrate the queue length distribution which is seen by CBR packets. These figures also show that the analysis approximation indeed provide an accurate prediction for delay distribution of CBR packets. Figure 4.24 shows the prediction of mean packet delay with different utilization.

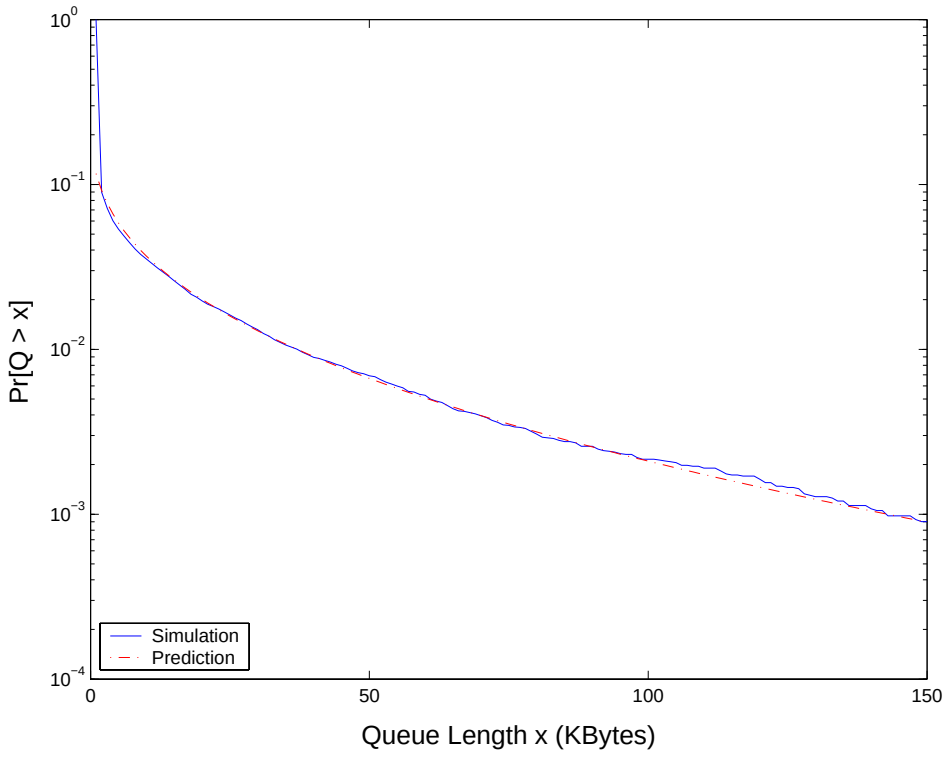

Figure 4.17: Queue length distribution seen by CBR packet $\rho = 0.191$

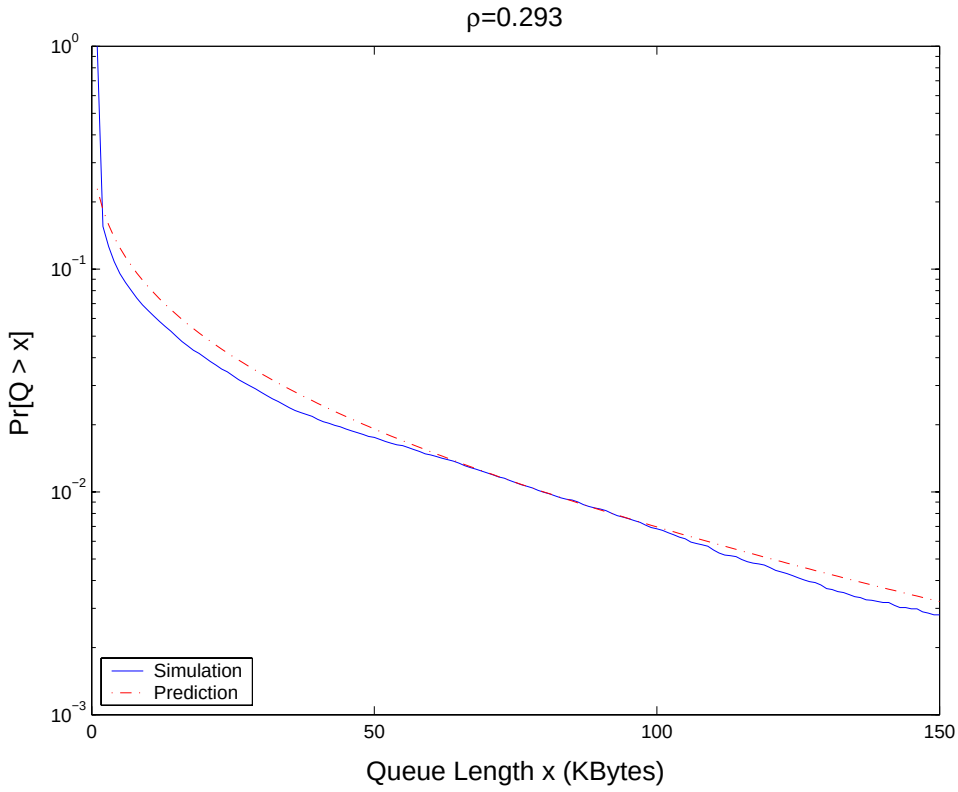

Figure 4.18: Queue length distribution seen by CBR packet $\rho = 0.293$

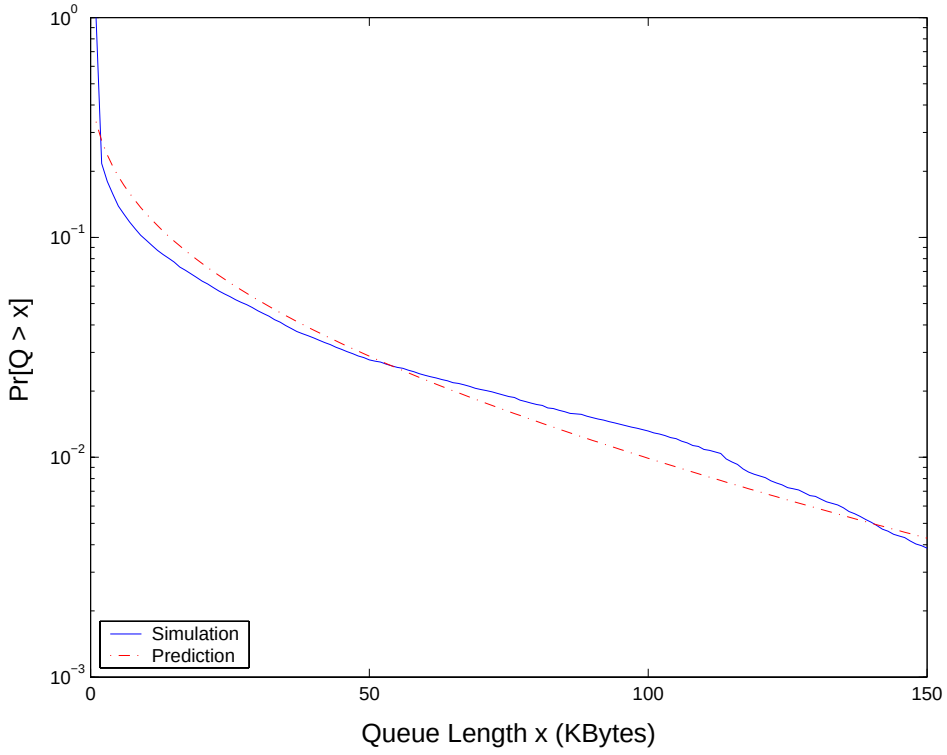

Figure 4.19: Queue length distribution seen by CBR packet $\rho = 0.383$

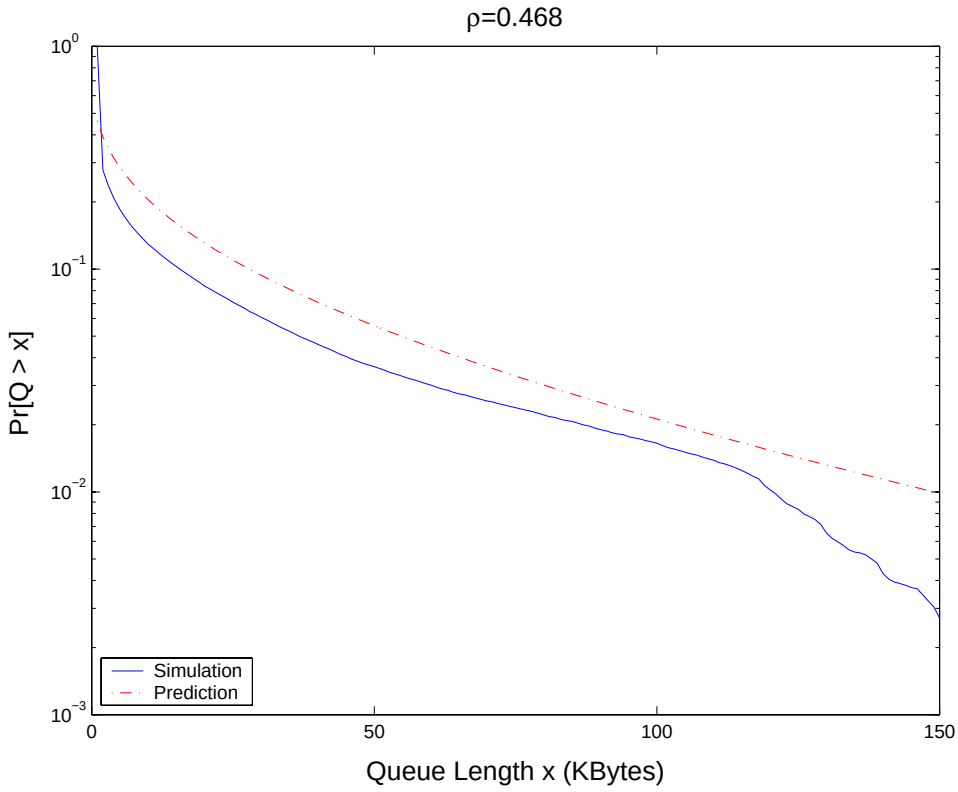

Figure 4.20: Queue length distribution seen by CBR packet $\rho = 0.468$

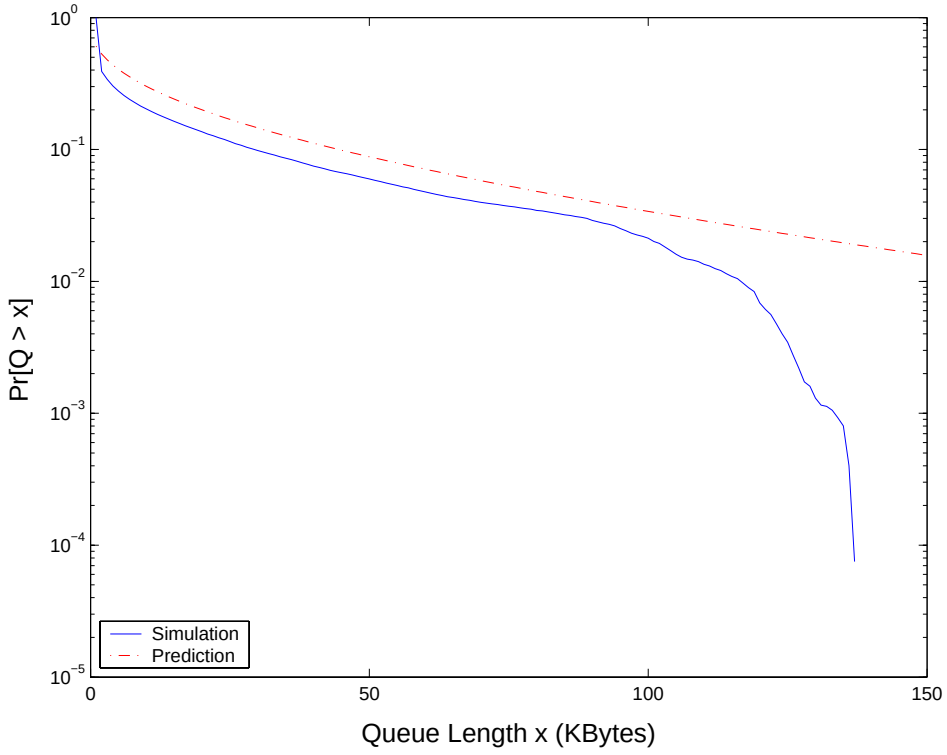

Figure 4.21: Queue length distribution seen by CBR packet $\rho = 0.573$

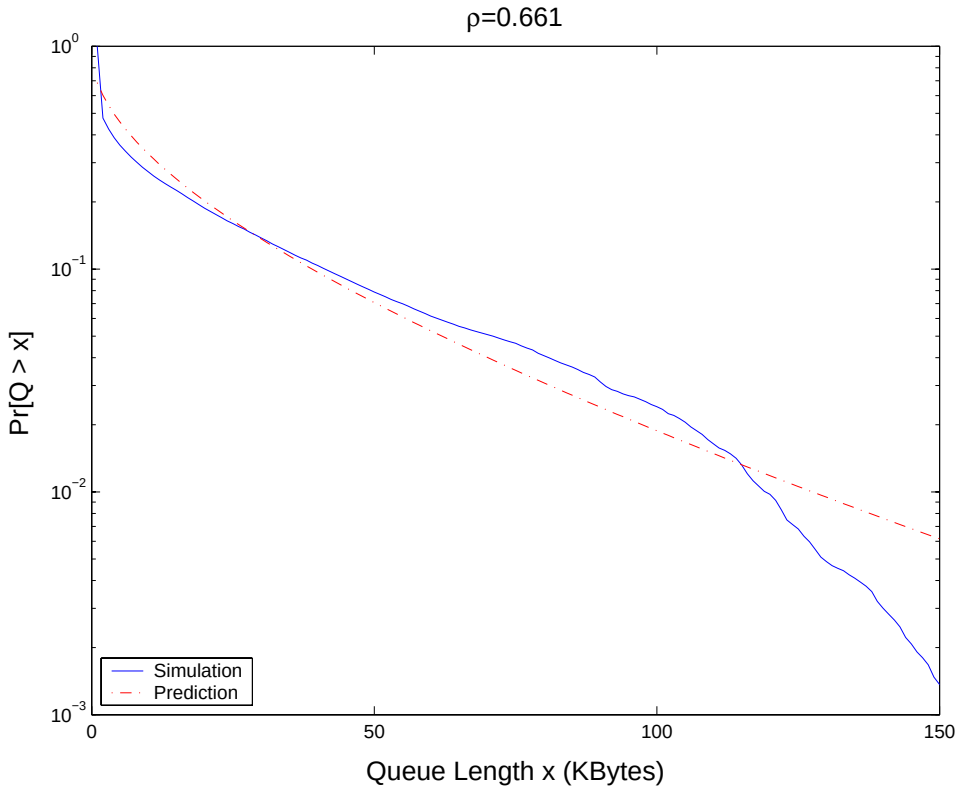

Figure 4.22: Queue length distribution seen by CBR packet $\rho = 0.661$

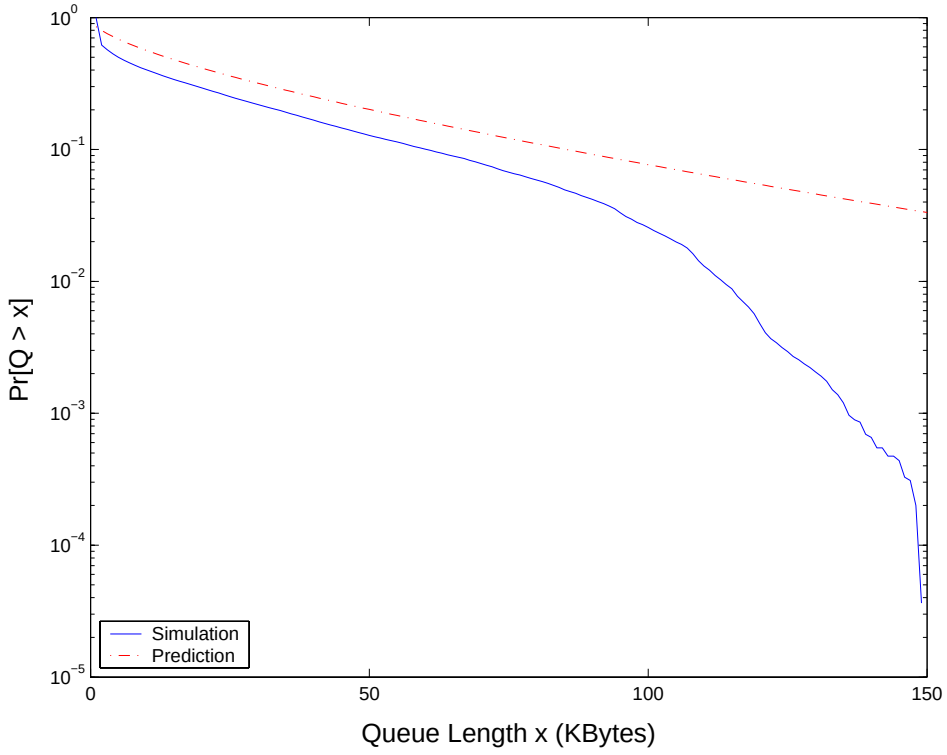

Figure 4.23: Queue length distribution seen by CBR packet $\rho = 0.771$

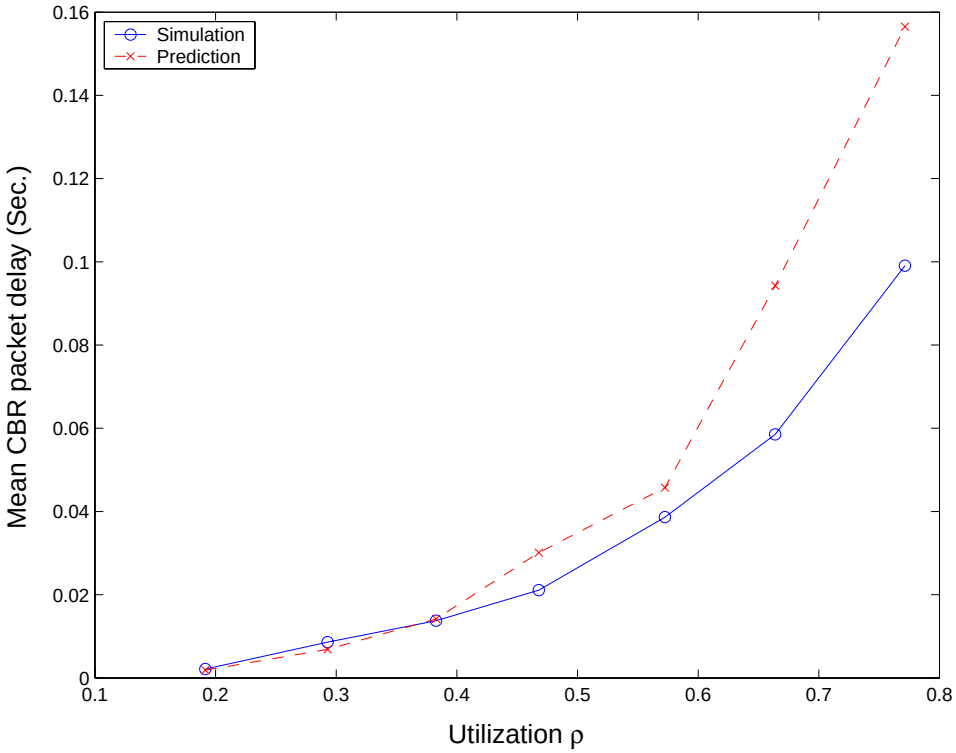

Figure 4.24: Mean CBR packet delay

For real time applications such as video/voice conference/stream and voice over IP, the delay jitter plays an more important role in the connection quality. The jitter is mainly caused by the perturbation of background traffic in the bottleneck queue. Fulton and Li [83] provided analytical approximations for the first-order and second-order statistics of the delay jitter. However, like other the jitter analysis methods [84] [85] [86], their works are based on the Markovian model of background traffic, which is not suitable for the long range dependent process such as Internet traffic. The Markovian approach also requires a lot of computational efforts and memory space for conditional probabilities and matrices . We develop an efficient method to predict the variance of jitter based on the the Logscale diagram of the traffic.

We consider the CBR traffic in the previous scenario of Figure 4.16. The CBR process is employed here to estimate the delay jitter of the bottleneck queue. The Logscale diagram of the background traffic is given as L_j. The delay jitter is defined as follows:

Definition 4.4.2 *Let the random sequence I_i be the interdeparture time of the target process, the jitter is defined as the difference of two consecutive interdeparture times:*

$$J_i := I_{i+1} - I_i. \tag{4.45}$$

We also define $A_{n,i}$ as the total arrival bytes of the background traffic in the i^{th} time slots. The duration of each time slot is $T := 2^n\Delta$ sec.

Proposition 4.4.3 *Let the current time be t=iT and b=$2^{n+1}\Delta(C-m)$. Assuming*

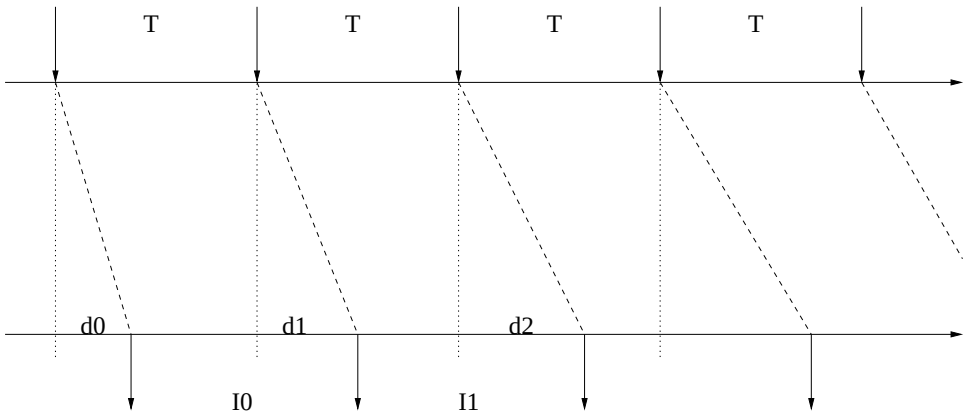

Figure 4.25: The arrival and departure time of CBR traffic

that the current queue length is $q(t) \geq b$. *The conditional variance of jitter is*

$$Var[J|q(t) \geq b] = \frac{1}{C^2} Var[(A_{n,i+1} - A_{n,i})]. \tag{4.46}$$

Proof Without loss of generality, let the CBR packets arrive at time 0, T, and $2T$, which have queuing delay d_0, d_1 and d_2, respectively. As shown in Figure 4.25, the total arrival bytes of the background traffic in the i^{th} time slots ($t \in [iT, (i+1)T)$) is $A_{n,i}$. Since the current length $q(0) \geq b$ is quite large, it is reasonable to say that the output link is always busy during the 0^{th} and 1^{th} time slots. Moreover, the buffer size is infinite so that there is no packet loss event. The Lindley equation:

$$q(t) = max_{0 \leq s \leq t}[A(t) - A(s) - C(t-s)], \forall t \geq 0 \tag{4.47}$$

can be simplified as

$$q((i+1)T) = q(iT) + A_{n,i} - CT. \tag{4.48}$$

Hence, the packet delay $d_i = q(iT)/C$ and the interdeparture time is

$$I_i \;\; = \;\; T + d_{i+1} - d_i$$

$$
\begin{aligned}
&= T + \frac{q((i+1)T) - q(iT)}{C} \\
&= \frac{A_{n,i}}{C}.
\end{aligned} \tag{4.49}
$$

The jitter variance under this condition is

$$
\begin{aligned}
Var[J|q(t) \geq b] &= Var(I_{i+1} - I_i) \\
&= \frac{Var[(A_{n,i+1} - A_{n,i})]}{C^2}.
\end{aligned} \tag{4.50}
$$

□

According to the definition of Haar wavelet, one may easily obtain the value of $Var[(A_{n,1} - A_{n,0})]$ from the Logscale diagram L_j

$$
2^{L_{j+1}} = E[d_{j+1,k}^2] = \frac{E[(A_{j,2k+1} - A_{j,2k})^2]}{2^{j+1}}, \tag{4.51}
$$

for every j and k.

Hence, the conditional jitter variance of the CBR process is

$$
Var[J|q(t) \geq b] = \frac{2^{n+1+L_{n+1}}}{C^2}. \tag{4.52}
$$

On the other hand, if the current queue length is small ($q(t) < b$), we assume that there is at least one idle server event happening in the next two time slots. The simple relationship of equation (4.48) and (4.49) does not hold. Since there is at least one idle event (queue empty event) in this period, the sequence of interdeparture times I_i and the sequence of packet delays d_i can be treated as uncorrelated random sequences respectively. We have the following upper bound:

$$
\begin{aligned}
Var[J|q(t) < b] &= Var[I_{i+1} - I_i] \approx 2Var[I_i] \\
&= 2Var[d_{i+1} - d_i] \approx 4Var[d_i] \\
&\leq \frac{4}{C^2} \max_{0 \leq j \leq n} Var[(A_j - C2^j\Delta)^+].
\end{aligned} \tag{4.53}
$$

Proposition 4.4.4 *Let A be the Lognormal random variable with parameter (μ, σ) and $d > 0$ be any real number, we have*

$$\begin{aligned} E[(A-d)^+] &= \frac{e^{\mu+\sigma^2/2}}{2} erfc(\frac{\ln d - \mu - \sigma^2}{\sqrt{2}\sigma}) - d\bar{F}(d) \qquad (4.54) \\ E[((A-d)^+)^2] &= \frac{e^{2\mu+2\sigma^2}}{2} erfc(\frac{\ln d - \mu - 2\sigma^2}{\sqrt{2}\sigma}) \\ &- de^{\mu+\sigma^2/2} erfc(\frac{\ln d - \mu - \sigma^2}{\sqrt{2}\sigma}) \\ &+ d^2\bar{F}(d), \qquad (4.55) \end{aligned}$$

where $\bar{F}(d) := Pr[A > d]$.

The probability of $Pr[q(t) \geq b]$ is based on the prediction of the steady state queue length distribution. We apply the previous results of queue length distribution in section 4.3 to estimate the probability $Pr[Q < b]$.

Proposition 4.4.5 *Let the r.v. Q be the queue length in steady state. From (4.52) (4.53) and (4.4), there is an upper bound of the jitter variance of the CBR traffic at the bottleneck router:*

$$\begin{aligned} Var(J) \leq\ & Var(J|Q \geq b)Pr[Q \geq b] \\ & + Var(J|Q < b))Pr[Q < b]. \qquad (4.56) \end{aligned}$$

Given the Logscale diagram of the traffic, we are able to predict the jitter variance immediately via (4.56). We validate this efficient approach by ns2 simulation in the next section.

4.5 Simulation results

The network topology in this experiment is the simple dumbbell network with a single bottleneck link. One side of the bottleneck link consists of 800 web clients. The clients send request to web serve, wait response, receive data and close the session. The interval between the end of session and next web request is called user think time. In this simulation the user think time of clients has an Exponential distribution with mean $50sec$, which is related to the human behavior in the Internet [9].

The other side has 800 web servers. The server is running HTTP 1.1 protocol and has a *Pareto* file size distribution with parameters (K=$2.3Kbytes$, α=1.3). The propagation delay of each server link is uniformly distributed in the interval ($16ms$, $240ms$) and the mean round-trip time is about $128ms$. The aggregated traffic requested by web client and generated by web servers has a mean rate about $1.2Mbps$. It is treated as the background traffic in the previous analysis. The unique CBR source at the server side has a permanent connection cross the bottleneck link. The CBR packets are sent periodically every T *sec.* and received at the client end. Since the link propagation delay of CBR connection is known, the queuing delay and jitter of CBR packet is estimated straightforwardly. We consider both FIFO and RED as the buffer management policy at the bottleneck link respectively.

Case 1: Jitter at a FIFO Queue

Figure 4.26 compares the predicted standard deviation of jitter with the simulation results. The target CBR process has fixed interarrival time $T = 2^n ms$ and $n = 3, 4, .., 8$. As a probing connection, the CBR has a small packet size so that the average mean rate of CBR traffic is negligible. Since the mean rate of the

background traffic is about $1.2Mbps$, the link utilization is about $0.4, 0.6$ and 0.8 with the corresponding bandwidth $C = 3.0, 2.0$ and $1.5Mbps$ respectively.

Case 2: Jitter at a RED Queue

We replace the FIFO queue by an adaptive RED queue at the bottleneck router. The adaptive RED queue [87] [88] will keep the average queue length located in a desired region by randomly dropping the TCP packets. Since the queue length is in the desired region, the link has a 100% utilization and no idle event happened. Hence, the jitter variance is bounded by equation (4.52). Figure 4.27 shows that the prediction method also provides tight bound for the jitter variance with different queuing policy such as RED is employed.

Since the interarrival time of most time-sensitive applications is located in the range of ($8ms$, $256ms$). Both simulation results show that our prediction method indeed provides a tight upper bound of jitter within this range.

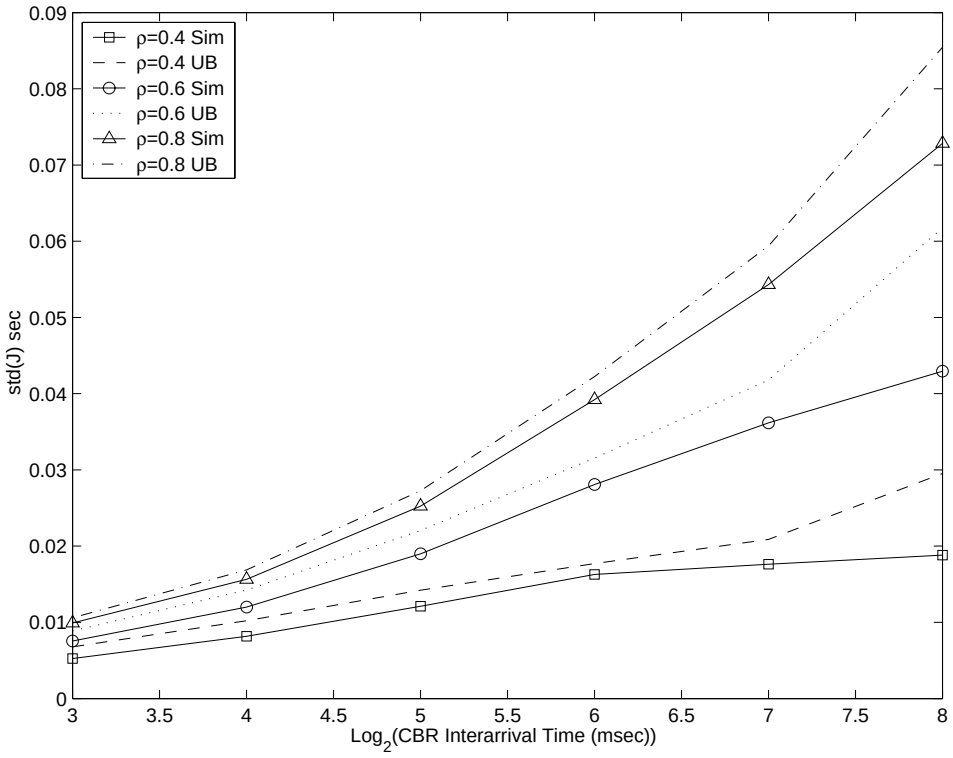

Figure 4.26: The standard deviation of delay jitter $std(J)$ $v.s.$ n with a FIFO queue

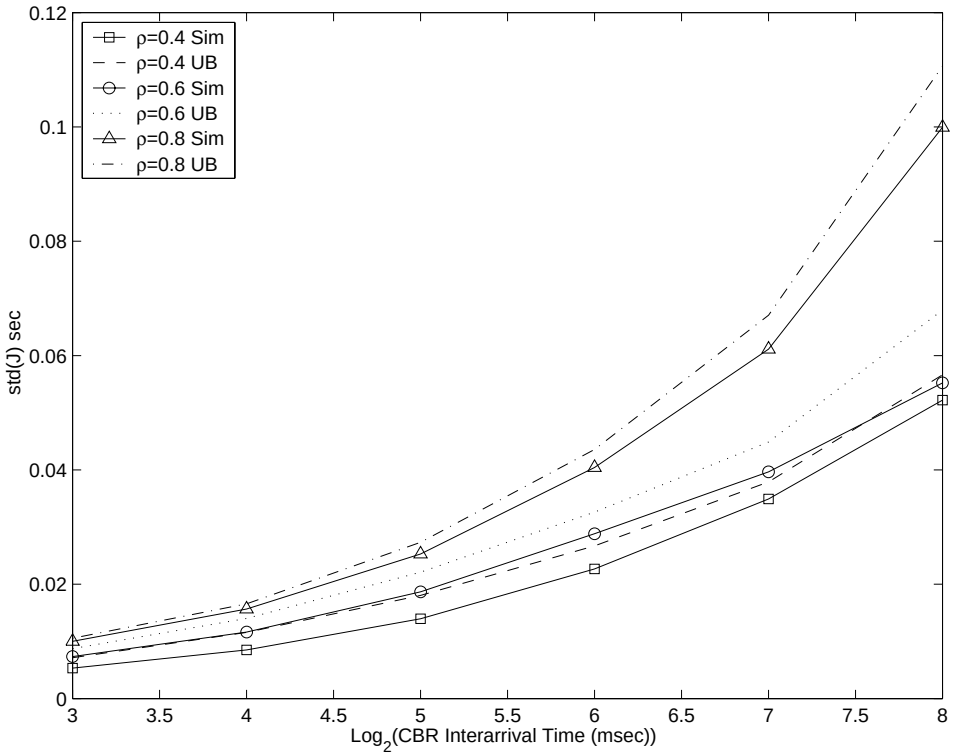

Figure 4.27: The standard deviation of delay jitter $std(J)$ $v.s.$ n with a RED queue

4.6 Summary

In this chapter we first analyze the multilevel ON/OFF model and provide a theoretical approximation of Logscale diagram. According to the property of wavelet analysis, the second order statistics of traffic can be immediately extracted from the corresponding Logscale diagram. Given the mean and variance of traffic workload at every time scale, we are looking for a mathematical distribution which can best fit the empirical distribution of real workload. After we employed the *Kolmogorov-Smirnov* goodness-of-fit test, the Lognormal distribution is chosen as the best candidate. The steady state queue length distribution is approximated according to the workload distribution at every time scale.

The queuing delay and jitter are also investigated on the basis of the approximation of queue length distribution. The mean delay is directly predicted by the mean queue length and the link bandwidth. An upper bound of jitter variance is developed according to the property of wavelet analysis and a simple condition probability. All theoretical results such as queue length distribution, mean delay and jitter variance are validated by simulation.

The proposed traffic model and the analysis in statistical property and queuing behavior provide a simple but efficient platform to understand the network engineering. It explains how the network parameters such as the active time, round-trip time, session life time and user think time affect the traffic statistic at different time scales. Furthermore, it also helps us understand the impacts of network parameters on connection performance such as packet loss, delay and jitter.

In the next chapter we focus on the active queue management (AQM) policy at the bottleneck router. A parallel structure and a new idea of flow control are

proposed for providing a better quality of service.

Chapter 5

Parallel Queue Structure for Active Queue Management

5.1 Overview

For small queuing delay, the buffer size in a router is not large. However, a router with small buffer size often has a high packet dropping rate since the Internet traffic is bursty. When packets are lost, the TCP protocol dramatically reduces the flow rate during the congestion avoidance phase [89]. Therefore, after a buffer overflow event in a drop-tail queue, all connections sense packet loss and slow down the transfer rate simultaneously. In order to prevent this global synchronization phenomenon and increase link utilization, many active queue management schemes such as RED (Random Early Detection) [41] have been proposed and received increasing attention.

The basic idea of RED is to randomly drop packets to prevent buffer overflow and the global synchronization problem. The dropping probability is a non-decreasing function of the queue length. A TCP connection with a higher flow

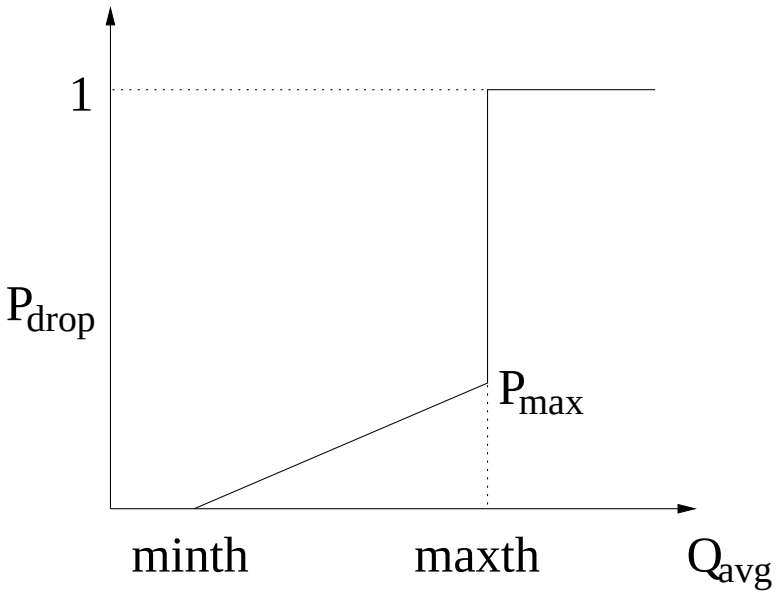

Figure 5.1: The dropping probability function of original RED.

rate has a better chance to get packets dropped and reduce its rate more rapidly. By dropping packets actively, RED keeps the queue length within a desired region. However, some simulation and analysis results [90] [39] [91] [92] have demonstrated that the performance of RED is very sensitive to parameter settings. Based on the original idea of RED, there have been some modifications such as Stabilized RED (SRED) [93], Flow RED (FRED) [94], Weighted RED [95], Random Early Marking (REM) [38], BLUE [40] and Adaptive RED [88] [87]. The Adaptive RED scheme dynamically updates the maximum dropping probability according to the exponentially weighted moving average (EWMA) of the queue length, and makes itself more robust with respect to the congestion level.

The Adaptive RED policy provides good rate control for TCP connections operating in the congestion avoidance phase [75] [87]. However, a great portion of Internet traffic is web and UDP traffic. Since most web connections involve transfer of several small files, these connections have a short life and are mostly operated

in the TCP slow start phase with a small congestion window. Dropping web packets in this phase is not an effective way to control the traffic rate and alleviate the congestion at the bottleneck router. Furthermore, from the viewpoint of a web user, one or several packet losses in the slow start phase would lead to extra delay for retransmission or even TCP timeout. It would also force TCP to enter the congestion avoidance phase prematurely with a small congestion window and result in a low throughput. The delay and low throughput would severely degrade the performance of delivering short messages such as web pages, and web browsers experience long waiting times even with a high speed network. On the other hand, the Adaptive RED fails to maintain the queue length within the desired region due to the bursty nature of web traffic.

To address these problems, we propose a parallel virtual queue structure for active queue management in this paper. In this structure, real time (web, UDP) and non-real time (FTP) traffic are separated into two different virtual queues which share the same physical buffer memory. The drop-tail policy is applied at the first virtual queue to serve real time applications. In order to have a small mean delay, the service rate of this drop-tail queue is dynamically determined by its virtual queue length. The remaining non-real time traffic is directed to an Adaptive RED virtual queue. Simulation shows that this parallel virtual queue structure not only has the advantages of Adaptive RED such as high link utilization and small delay, but also greatly reduces the total packet loss rate at the router. Despite that the bandwidth is shared with the bursty drop-tail virtual queue, the Adaptive RED queue has a small length variation.

The original Adaptive RED dynamically changes the maximum dropping probability P_{max} to keep the queue length within the thresholds. However, for some

non-real time applications, high goodput (low packet dropping rate) is more important than short packet delay. Hence we explore a modified Adaptive RED policy for the non-real time applications at the second virtual queue, where the queue length thresholds are dynamically adjusted to maintain the dropping probability of Adaptive RED algorithm in a desired range.

The remainder of the dissertation is organized as follows. In Section 5.2, we demonstrate the vulnerability of the Adaptive RED in the presence of web and UDP traffic. The parallel virtual queue structure is described in Section 5.3. Comparison of this approach with the original Adaptive RED scheme is given through simulation in Section 5.4. In Section 5.5, we present the modified Adaptive RED policy with dynamic queue length thresholds. Performance and stability analysis is provided for this parallel structure with the drop-tail and the modified Adaptive RED queue in Chapter 6. Finally, we conclude our work in Chapter 7.

5.2 Vulnerability of Adaptive RED to Web-mice

In this section we consider a scenario containing short-life TCP (WEB), UDP (CBR) and long-life TCP (FTP) traffic. The purpose is to demonstrate that the performance of the Adaptive RED scheme is severely degraded by the short-life web traffic. The network in our ns2 experiment is the same scenario of delay and jitter analysis in Chapter 4 except the FTP traffic. It has a simple dumbbell topology with the bottleneck link bandwidth C=3.0$Mbps$. One side of the bottleneck consists of 800 web clients. Each client sends a web request and has a think time of Exponential distribution with mean 50s after the end of each session. The other side contains 800 web servers, running HTTP 1.1 protocol and having a $Pareto$ [16] file size distribution with parameters (K_p=2.3$Kbytes$, α=1.3) (mean

$10Kbytes$). The round-trip propagation delay of HTTP connections is uniformly distributed in (16, 240)ms. Note that the mean rate of the aggregate web traffic is around $1.2Mbps$. There is one CBR traffic source which periodically generates a $1Kbytes$ UDP packet every $50ms$. Besides these short web connections and UDP traffic, there are 10 persistent FTP connections sharing the bottleneck link with round-trip propagation delay of $64ms$. Figure 5.2 shows that the Adaptive RED works well with those FTP connections before the web traffic comes in. However, after the CBR source and web servers begin to share the bandwidth at time t=100s, the queue length of Adaptive RED deviates dramatically from the desired region. Since the Adaptive RED scheme relies on average queue length to determine the dropping probability and control the TCP flow rate, the extra queue length perturbation contributed by the bursty web traffic makes the Adaptive RED increase/decrease its dropping probability rapidly. This over-reaction causes a great queue length variation and poor performance in packet delay and loss.

Since most web pages contain one or several very small files, these TCP connections are mostly operated in their slow start phase during the session life. According to the TCP protocol, the congestion control window is just beginning to increase its size from the initial value and the flow rate is low. Dropping packets in the slow start phase cannot efficiently alleviate the congestion level at the bottleneck router. In other words, any random dropping/marking policy such as RED is unable to effectively control the congestion level without considering short-life TCP (and UDP) traffic. Furthermore, losing one or two packets in the slow start phase not only causes a very low throughput and extra delay, but also leads to a high probability of connection timeout. This is further illustrated below.

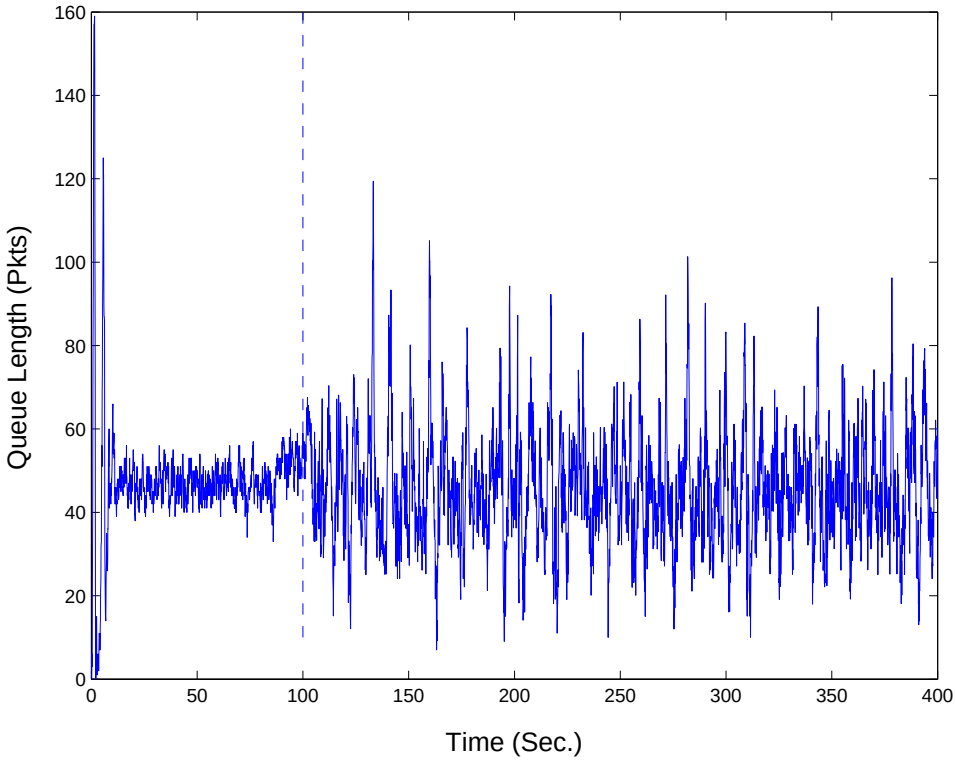

Figure 5.2: Queue length of the Adaptive RED: 10 FTP starting at t=0 and 800 WEBs and 1 CBR coming in at t=100s.

From the viewpoint of a web browser, a short-life TCP session may only need several round-trip times (RTT) to finish the whole transmission. When the sender senses a packet loss, the slow start threshold $ssthresh$ will be reduced to $min(cwnd, rcv\_window)/2$ [89] and the new congestion window size $cwnd$ is also decreased depending on different TCP versions (For TCP Reno, the new $cwnd = ssthresh$ and TCP enters the fast recovery phase. For TCP Tahoe, $cwnd = MSS$ (maximum segment size) and TCP begins a new slow start phase). Since original $cwnd$ is just beginning to increase its size from its initial value MSS in the first slow start phase, one packet loss during the initial several round-trip times leads TCP to enter the congestion avoidance phase with a very small $ssthresh$ and $cwnd$. In the congestion avoidance phase, TCP slowly increases $cwnd$ (the increment is about one MSS per round-trip time) from the current $ssthresh$. Therefore, losing one packet in the slow start phase (as shown in Figure 5.3) takes TCP a long time to complete a short message. In addition, since the web traffic is short but bursty, these web connections usually experience a higher packet loss rate (see the web packet loss rates of the Adaptive RED and the drop-tail policies in Table 5.3).

The default initial value of $ssthresh$ is $64KB$ and the packet size is $1KB$ in this paper. Assuming a typical packet dropping probability P_d=0.04 when using the Adaptive RED, the probability of losing one or more packets in the slow start phase is equal to $1-(1-P_d)^{64} = 0.9267$ (assuming that packets are dropped independently). Therefore, most TCP connections have at least one packet dropped in their first slow start phase. For example, assuming that the 15^{th} packet is dropped by the Adaptive RED, $ssthresh$ decreases from $64KB$ to $4KB$ and the new congestion window $cwnd$ is decreased from $8KB$ to $1KB$ (Tahoe). The situation gets worse if one packet is dropped earlier (in the first 3 round-trip times). The con-

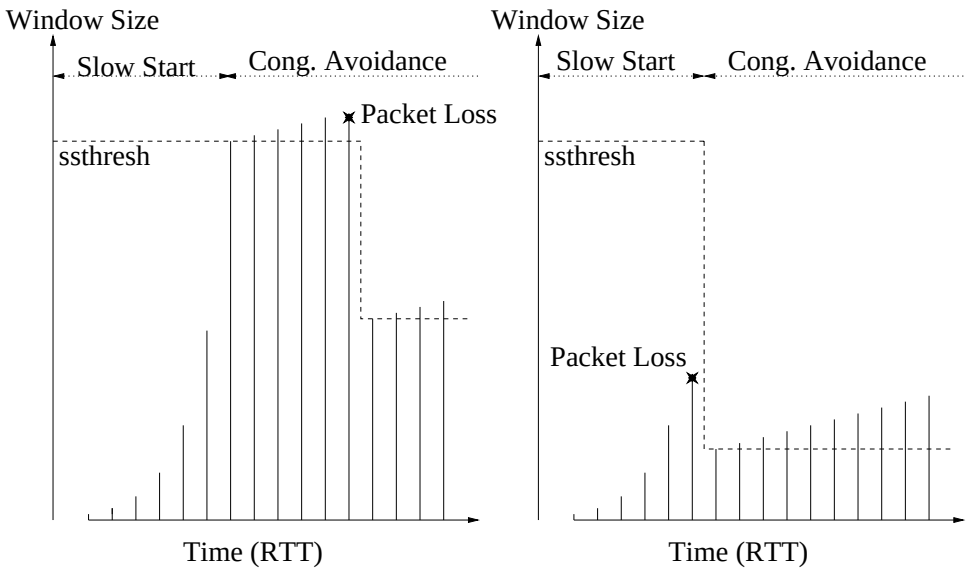

Figure 5.3: Congestion window size of TCP Reno: One packet loss in the Slow Start phase (left) and One packet loss in the Congestion Avoidance phace (right).

gestion window at this moment is so small that the sender may not have enough data packets to trigger the receiver to generate three duplicate acknowledgements. If packets cannot be recovered by this fast recovery scheme, TCP has to depend on the protocol timer for error recovery. The default value of the protocol timer is usually large and the delivery delay could be increased dramatically by timeout events. Moreover, the probability of losing two or more packets of the same congestion window in the slow start phase also cannot be ignored. These events lead to a high probability of TCP timeout and connection reset.

For illustration we conduct simulation of transferring a small web file in a stand alone and one hop environment. There is no other traffic sharing the bandwidth and packets are dropped intentionally. Figure 5.4 shows the mean delivery delay *v.s.* the dropping probability for file sizes $30KB$-$210KB$, and Table 5.1 lists the mean and standard deviation of the delay. For example, TCP takes about $4.81s$

to complete transmission of a $90KB$ file if $P_d = 0.04$; in comparison, in the loss free case, the file can be delivered in $1.18s$. Since most web pages have sizes in the above range, a web browser will experience a long response time when the dropping probability of the Adaptive RED is high.

Table 5.1: Delivery delay of small file: mean and standard deviation

P_d	0.00	0.02	0.04	0.08
$30KB$	0.88(.0006)	1.60(1.88)	2.74(4.27)	5.88(7.79)
$90KB$	1.18(.0008)	2.79(2.39)	4.81(3.91)	9.24(6.30)
$150KB$	1.34 (0.0008)	3.51(1.90)	6.51(4.60)	13.38(8.87)

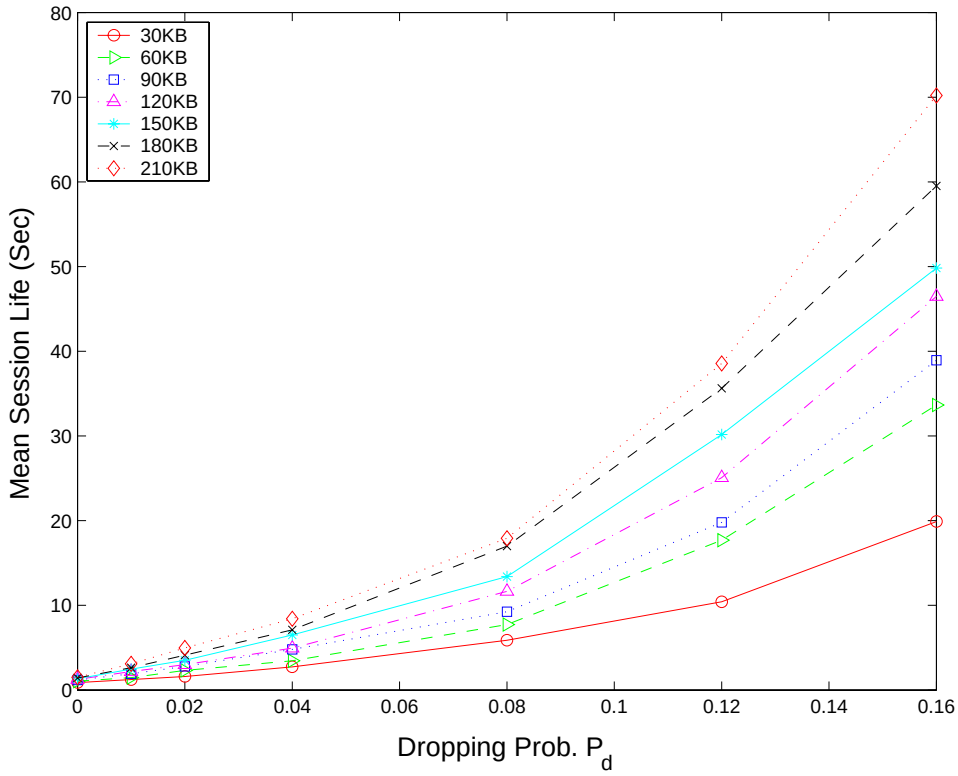

Figure 5.4: Mean delivery delay of small file $v.s.$ dropping probability P_d with file sizes $30, 60, ..., 210 Kbytes$, bandwidth $3 Mbps$ and round-trip time $128 ms$.

5.3 A Parallel Virtual Queues Structure

To solve the problem discussed in Section II, we propose a parallel virtual queue structure in the router. The first virtual queue deals with the short-life real-time traffic (web, UDP). Since dropping these packets cannot effectively alleviate the congestion level, but severely increases delivery delay, it would be good to keep them in the queue unless the total buffer (shared with the other queue) has overflowed. Hence, the queuing policy of the first virtual queue is chosen to be drop-tail to minimize the packet loss rate. In order to have a short delivery delay for web browsers and UDP connections, the service rate $C_1(t)$ is changed dynamically according to its virtual queue length $q_1(t)$.

The second virtual queue serves long-life TCP connections such as FTP with large file sizes, where the Adaptive RED is used. Although the available bandwidth of this queue is determined by $C_2(t)$=C-$C_1(t)$, the Adaptive RED scheme is expected (and will be verified by simulation in Section 5.4 to keep its virtual queue length $q_2(t)$ in a desired region for the following reason. When there is a heavy workload at the drop-tail queue, $C_2(t)$ decreases quickly. FTP receivers experience slower packet arrival rates and send acknowledgement packets (ACK) back more slowly. Without increasing the dropping probability at the Adaptive RED queue, the slower ACK arrival rates from the receivers make FTP senders reduce flow rates automatically without shrinking their congestion window sizes. On the other hand, when the congestion level is alleviated, the Adaptive RED queue receives more bandwidth. Since the congestion window sizes

are still large in the FTP servers, the throughputs of FTP is quickly recovered by faster arrival rates of ACK packets from the receivers.

With this parallel virtual queue structure (which will be called RED+Tail pol-

icy in this paper), we can keep the benefits of Adaptive RED such as high (100%) link utilization. Furthermore, the packet loss rate of the short-life TCP and UDP connections is greatly reduced by the drop-tail policy and a shared buffer. The packet loss rate of long-life TCP traffic is also reduced due to the suppressed bandwidth, larger thresholds (longer RTT) and more stable average virtual queue length for the Adaptive RED queue.

We now discuss how to implement the RED+Tail policy. The first problem is how to split the long-life traffic from other short-life web traffic at the router. To this end, the router has to know the age or elapsed time of each TCP connection. Unfortunately, this information is hidden in the TCP header which is not available to the IP router. However, one may roughly estimate the elapsed time by using the following approach:

- When a packet arrives with a *new* source-destination pair which has not been seen by the router in the past T sec, we treat it as a new TCP connection and identify this connection as a short-life connection;
- Send the new connection packets to the drop-tail queue;
- Set a counter for the number of packets of this connection;
- If the cumulative packets number is greater than a threshold N, we assume that the file size is large enough and this TCP connection has already left its slow start phase. We redirect the subsequent packets of this connection to the Adaptive RED queue;
- Remove the counter if there is no packet arrival in the last T sec.

Preliminary simulation results show that this approach successfully prevents small web traffic from entering the RED queue. The probability of false alarm is less

than 0.02 in our scenario. Since the web traffic has small file sizes and short session times, there is no harm if the short-life connection packets are misdirected to the RED queue after time T.

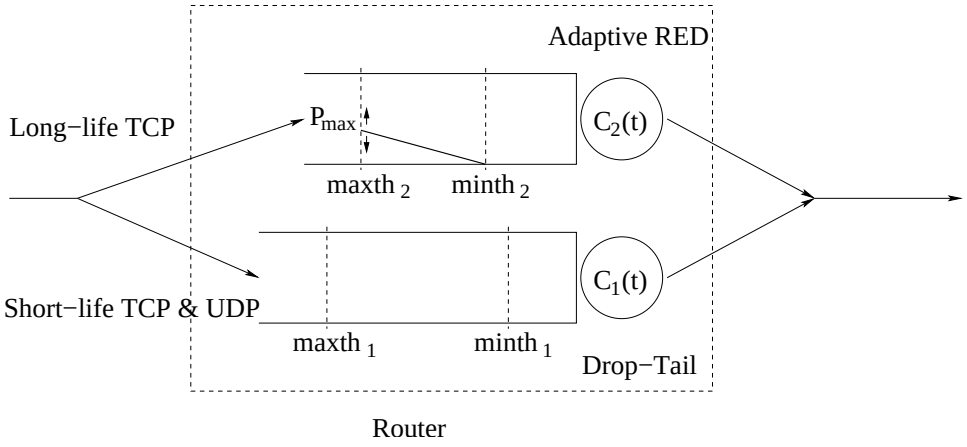

Figure 5.5: The parallel virtual queue structure for active queue management.

Figure 5.5 shows the RED+Tail parallel queue structure in the router. Recall that $C_1(t)$ and $C_2(t)$ denote the service rates of the drop-tail queue and the Adaptive RED queue at time t respectively. In order to allocate bandwidth dynamically to both queues and assign a desired region of queue length for the Adaptive RED queue, we define the maximum threshold $maxth_i$ and minimum threshold $minth_i$ for $i = 1, 2$. The service rates $C_1(t)$ and $C_2(t)$ are given by the following algorithm:

- if $q_1 = 0$, then $C_1(t) := 0$.
- if $0 < q_1 < minth_1$, then $C_1(t) := C_{1min}$.
- if $minth_1 \leq q_1$, then $C_1(t) := min(C\frac{q_1}{maxth_1}, C_{1max})$.
- $C_2(t) := C - C_1(t)$,

where C is the link bandwidth. The variable q_1 denotes the queue length of the

drop-tail queue. The constant C_{1max} preserves the minimum available bandwidth $C - C_{1max}$ for the RED queue to prevent FTP connections from timeout.

5.4 Simulation and Comparison

In this section, we compare the RED+Tail scheme with the Adaptive RED on typical TCP performance metrics. For the Adaptive RED, we use the parameter settings suggested by Floyd *et al* [87] (α and β of the AIMD algorithm). Both schemes were implemented in the ns2 simulator. The network topology and scenario are as described in Section II. Table 5.2 lists the parameters for the RED+Tail policy and the Adaptive RED policy. Note that the virtual queues of the RED+Tail scheme share the total physical buffer size, *i.e.*, the packets in the drop-tail virtual queue will not be dropped unless the physical memory is full. The Adaptive RED is set in a "*gentle*" mode meaning that the dropping probability between ($maxth_2$, $2maxth_2$) is linear in (P_{max}, 1).

Table 5.2: Experiment Settings

Virtual Qu. i	Buffer Size	$minth_i$	$maxth_i$	α	β
$i = 1$	160KB	2KB	30KB	-	-
$i = 2$	shared	20KB	80KB	0.01	0.9
Adapt. RED	160KB	20KB	80KB	0.01	0.9

The performance for a connection is evaluated by the packet loss rate, delay and throughput. However, we are more concerned about packet loss rate and delay for web (short-TCP) and CBR (UDP) connections, and more concerned about

Table 5.3: Performance Metrics

Policy	Loss %	Delay Sec.	Rate KB/s
RED+Tail:FTP	2.747	0.184	209.465
RED+Tail:WEB	1.278	0.114	144.455
RED+Tail:CBR	0.300	0.109	19.867
AdaptRED:FTP	4.149	0.143	217.531
AdaptRED:WEB	4.514	0.143	137.124
AdaptRED:CBR	3.950	0.141	19.140
DropTail:FTP	1.916	0.349	215.243
DropTail:WEB	4.234	0.340	138.983
DropTail:CBR	1.550	0.342	19.601

throughput for FTP (long-TCP). We replaced the Adaptive RED with RED+Tail scheme at the router and repeated the experiment of Section II. For comparison, an experiment with the drop-tail policy was also conducted. The random seed of the simulator was fixed so that the processes of web requests and file sizes had the same sample paths in all experiments. Table 5.3 lists the performance metrics under RED+Tail, the Adaptive RED and the traditional drop-tail scheme respectively.

Figure 5.6 shows the queue lengths of the RED+Tail scheme, which demonstrates that the virtual queue length q_2 is quite stable and stays in the desired region even after the web and CBR traffic begins to share the bandwidth at time t=100s. The actual dropping probability for the FTP traffic is reduced from 4.15% to 2.75% by a longer queuing delay (184ms, see Table III). This scheme prevents the over-reaction behavior of RED in the original Adaptive RED case and keeps

the mean queue length q_2 in a desired region (Compare to Figure 5.2).

Figure 5.7 shows the packet loss rates of FTP, web and CBR connections under the Adaptive RED and RED+Tail schemes. We see that RED+Tail provides great improvement in packet loss for web and CBR connections. The web packet loss rate is reduced from 4.51% to 1.28% and CBR packet loss rate is reduced from 3.95% to 0.30%.

Figure 5.8 compares the packet delays. The mean queuing delay of web and CBR packets in the RED+Tail scheme is shortened at the cost of the FTP packets delay. The web and CBR packet delay depends on how much bandwidth is allocated to the drop-tail queue. One can satisfy a mean delay requirement for the web and CBR connections by properly adjusting the parameter $maxth_1$. For example, the $maxth_1$ of the RED+Tail scheme is set to be $30Kbytes$ so that the estimate of mean delay at the drop-tail queue is about $80ms$. However, the service rate C_1 reaches its maximum C_{1max} when $q_1 > maxth_1$. The actual mean delay should be larger than expected. For our simulation the mean delay of web and CBR traffic is around $110ms$ (refer to analysis in Section 6.1).

Figures 5.9 and 5.10 show the throughputs of FTP, web and CBR traffic. Both schemes achieve 100% utilization of the link bandwidth. Due to the bandwidth allocation scheme in the RED+Tail scheme, FTP has a slightly smaller throughput. However, the saved bandwidth allows web burst to pass through the bottleneck link faster.

Figure 5.11 compares the small web file delivery time under different schemes. Since the RED+Tail policy has a small packet loss rate, its delivery time is almost equal to the loss free case in Table 5.1. On the other hand, the Adaptive RED has a loss rate 4.5%, its delivery time is three times longer. Note that the drop-tail queue

has a similar loss rate (4.2%) as Adaptive RED for web packets. However, the file delivery time of the drop-tail scheme is about 2.5 times longer than Adaptive RED's. This is mainly due to the long queuing delay ($0.340sec$) of the drop-tail policy.

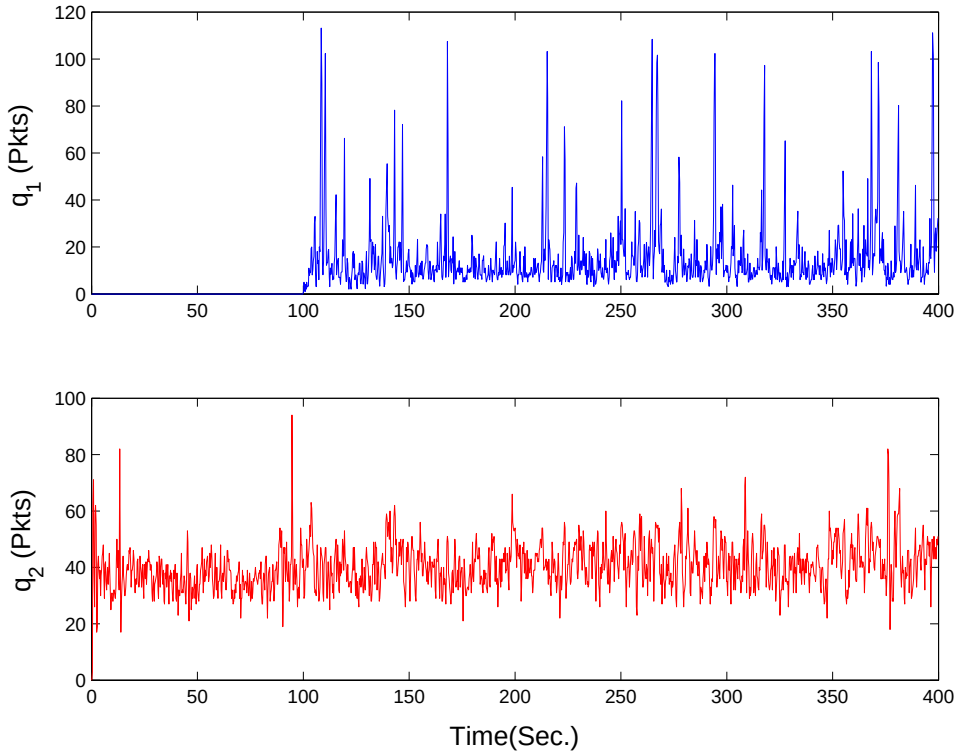

Figure 5.6: Queue lengths of RED+Tail virtual queues: 10 FTPs starting at t=0 go to virtual queue 2, and 800 WEBs + 1 CBR starting at t=100 go to virtual queue 1.

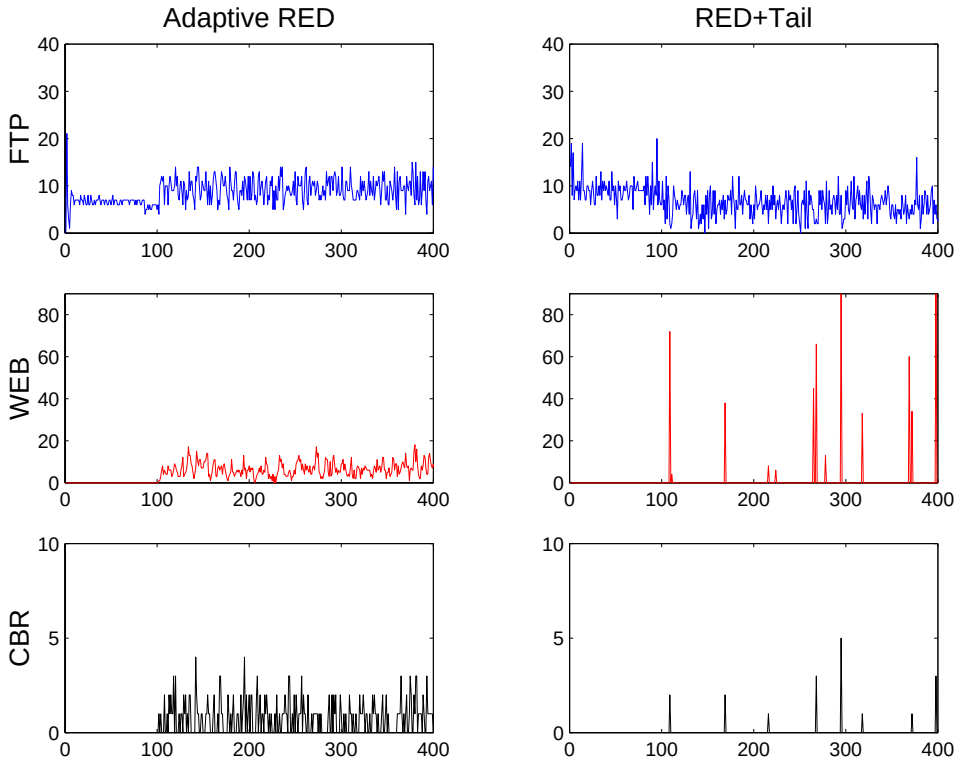

Figure 5.7: Packet losses (packets/sec.) of Adaptive RED and RED+Tail.

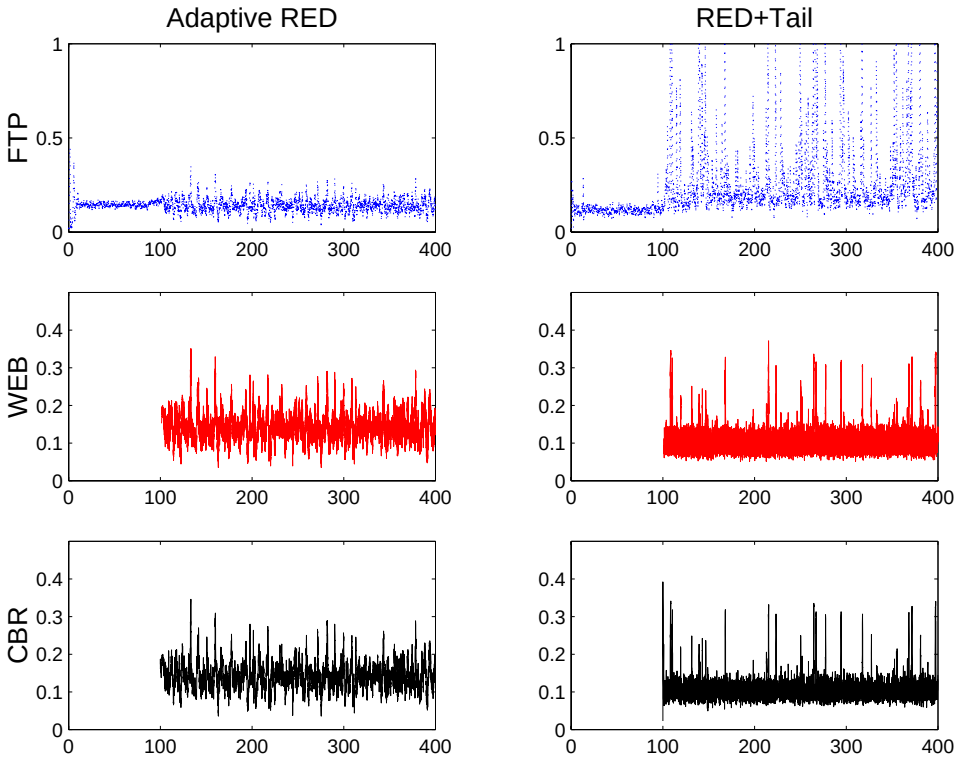

Figure 5.8: Packet delays (sec.) of Adaptive RED and RED+Tail.

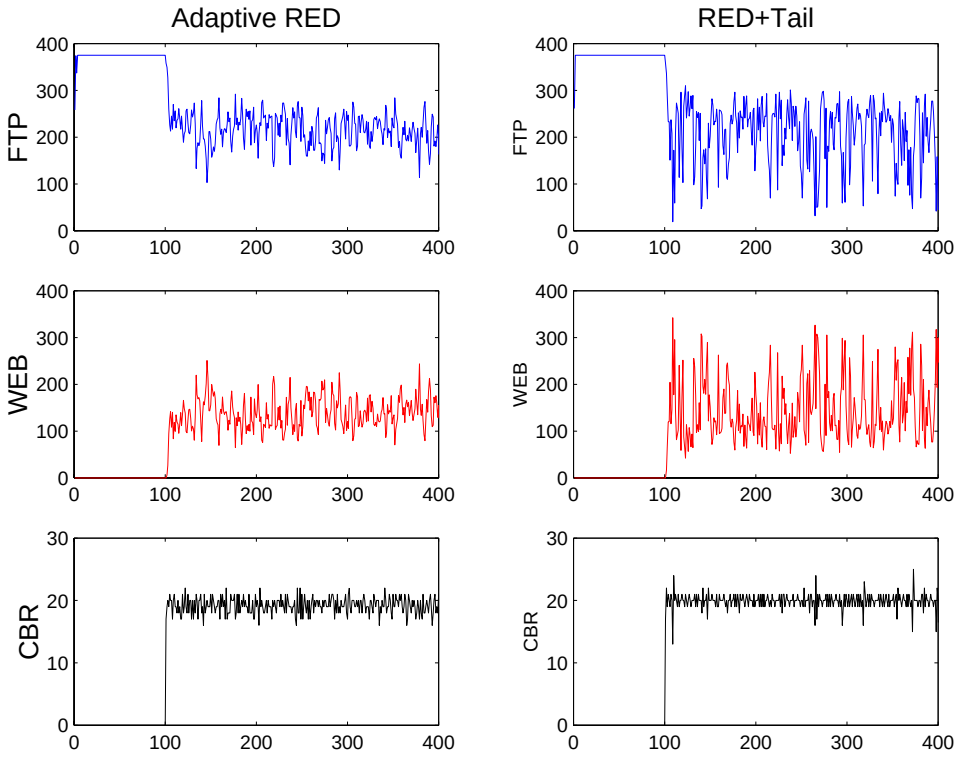

Figure 5.9: Throughputs (KBytes/sec.) of Adaptive RED and RED+Tail.

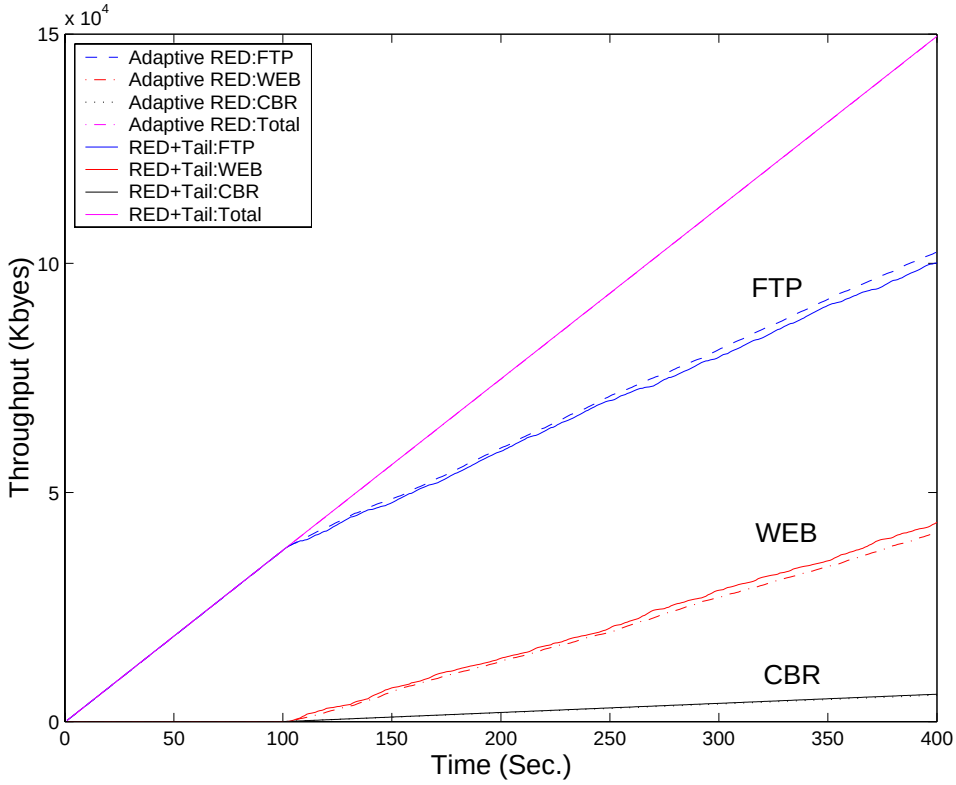

Figure 5.10: Accumulated throughputs of Adaptive RED and RED+Tail.

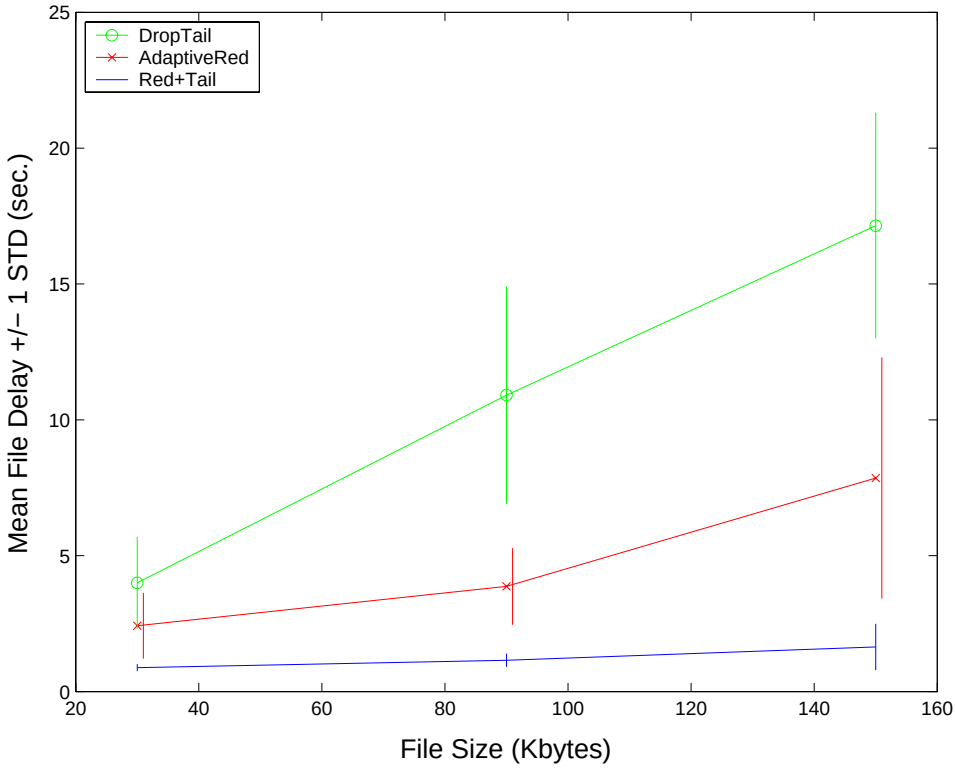

Figure 5.11: Small file delivery delay: mean and standard deviation.

5.5 Dynamic Thresholds for Adaptive RED

The original Adaptive RED dynamically adjust the maximum dropping probability P_{max} (or equivalently, the slope of dropping function) to control the flow rates of TCP connections and keep the average queue length in a desired region. However, for those applications with large file sizes, the goodput is more important than the packet delay. The packet loss rate is a key factor in determining the connection goodput. Since the minimum and maximum thresholds of the original Adaptive RED scheme are fixed, the dropping probability of Adaptive RED could be very high when a congestion happens. This high dropping probability causes frequent re-transmissions, small average congestion window size and low goodput. In other words, the congestion in bottleneck router causes another bottleneck at the TCP sender end. Considering that the Adaptive RED queue is designed for serving time insensitive connections, we propose to control the TCP flow rate by adjusting its queuing delay instead of dropping packets.

To maintain a low packet loss rate (and a large average congestion window size at the TCP sender), the following modified Adaptive RED scheme for the Adaptive RED queue is proposed, where $minth_2$ and $maxth_2$ are dynamically adjusted while D=$maxth_2 - minth_2$ is maintained constant:

- Pick $0 < \gamma < 1$ (γ=0.05 in this paper).
- If $\bar{P}_d > P_U$, then $minth_2 := minth_2(1+\gamma)$, $maxth_2 := minth_2 + D$.
- If $\bar{P}_d < P_L$, then $minth_2 := minth_2(1-\gamma)$, $maxth_2 := minth_2 + D$,

where $\bar{P}_d$ is the average dropping probability obtained by the EWMA algorithm and (P_L, P_U) is the desired region of dropping probability. Note that if we set $P_U < P_{max}$, the floating thresholds do not change the current slope of dropping

probability function dramatically, since the distance between the thresholds is fixed.

The rationale behind the above scheme is that, by increasing the thresholds (when $\bar{P}_d > P_U$), the queuing delay is increased and the flow rates are reduced. Since the average TCP throughput [96] [97] is proportional to $\frac{1}{RTT\sqrt{P_d}}$, we achieve the same throughput without raising the packet loss rate. Figures 5.12 and 5.13 compare the Adaptive RED schemes with fixed and dynamic thresholds respectively. There are 20 persistent FTP servers sharing a $6Mbps$ bottleneck link. Another 20 FTP servers arrive at time $100s$ and leave at time $300s$. It can be seen that the fixed threshold scheme has a small queue length variation and a large dropping probability (0.05). In contrast, the dynamic threshold scheme has a much lower average dropping probability (0.014 with P_L=0.01, P_U=0.02), but a higher packet delay. Note that both schemes achieve 100% link utilization so that each FTP connection has the same throughput. However, with a much lower packet loss rate, the dynamic threshold scheme achieves a higher goodput. This dynamic threshold scheme allows us to consider the trade-off between packet loss and queuing delay in an Adaptive RED queue.

Dynamically varying the thresholds may also have implications in achieving throughput fairness among multiple Adaptive RED queues. Since the flow rates of TCP connections are determined by the corresponding dropping probabilities and queuing delays at different queues, connections with shorter link propagation delays and higher throughputs can be suppressed by raising the queue length thresholds at the router.

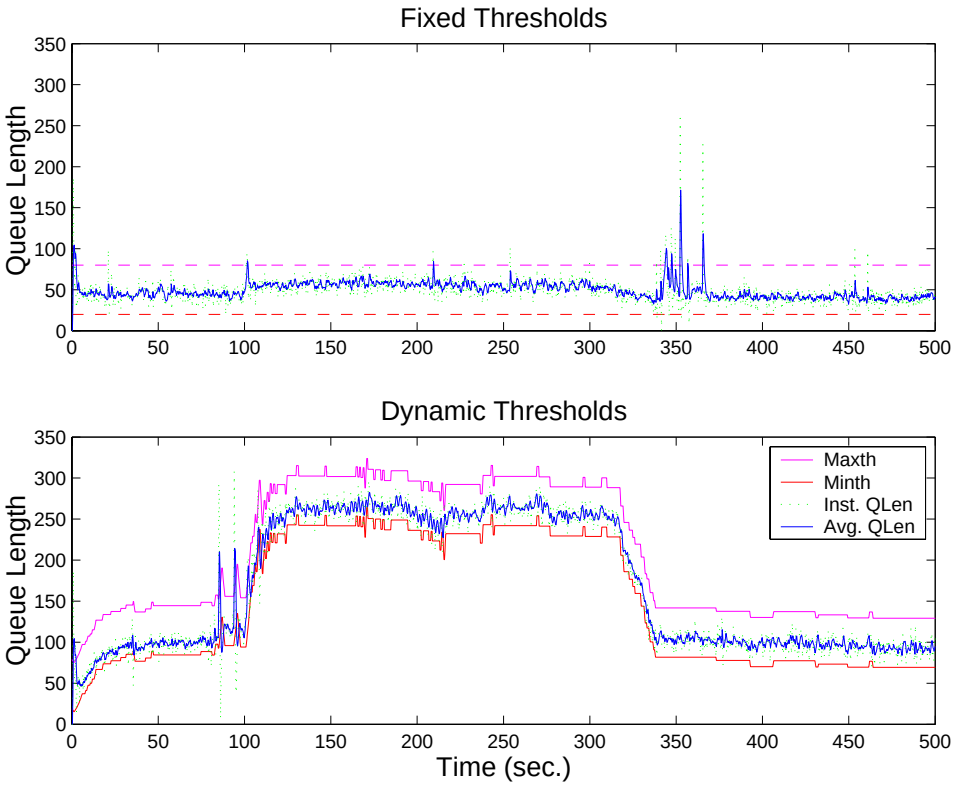

Figure 5.12: Average queue length with fixed and dynamic thresholds: 20 FTP starting at t=0, and another 20 FTP starting at t=100s and leaving at t=300s, C=6Mbps, d_k=64ms.

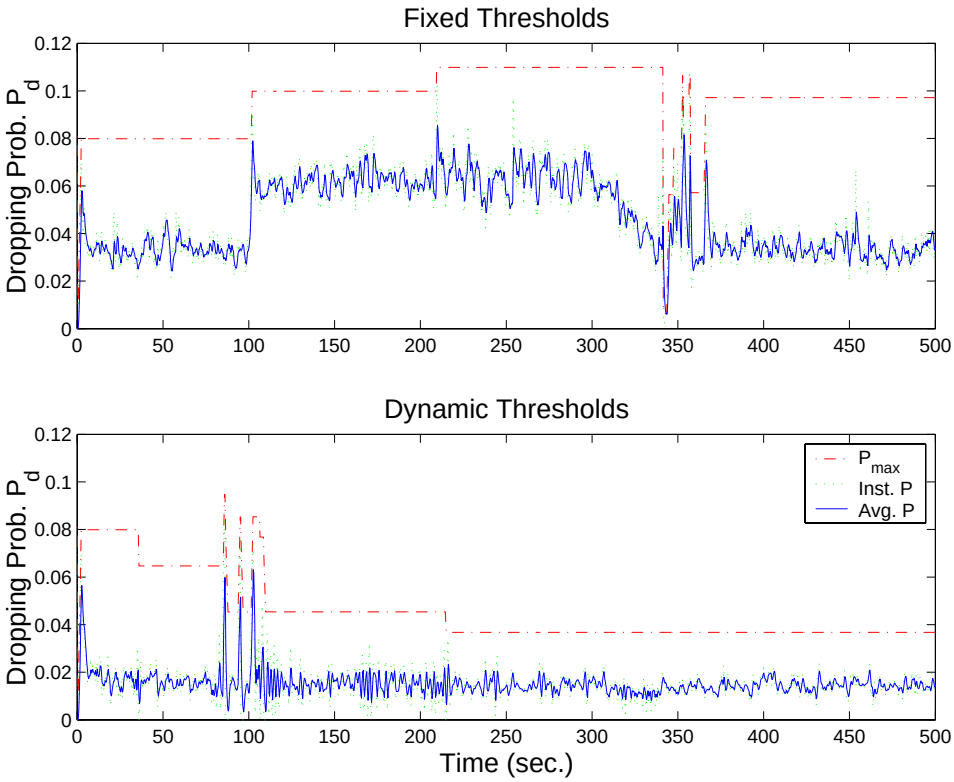

Figure 5.13: Dropping probability with fixed and dynamic thresholds: 20 FTP starting at t=0, and another FTP 20 starting at t=100s and leaving at t=300s, C=6Mbps, d_k=64ms (Inst. P: instantaneous dropping probability; Avg. P: EWMA average of Inst. P).

Table 5.4: Performance Metrics:Red+Tail with Dynamic threshold scheme

Policy	Loss %	Delay Sec.	Rate KB/s
Dyn. Thres.:FTP	0.899	0.318	209.455
Dyn. Thres.:WEB	2.306	0.093	144.505
Dyn. Thres.:CBR	0.519	0.091	19.827

The ns2 simulation in Section 5.4 is conducted again with the *modified* Adaptive RED serving the second virtual queue. Parameters (except $minth_2$ and $maxth_2$, which are dynamically adjusted) used are as listed in Table 5.2. The desired region of dropping probability for the Adaptive RED queue is set to be (P_L,P_U)=(0.005,0.010). Figure 5.14 shows the lengths of both virtual queues and the dropping probability at the Adaptive RED queue. The dropping probability stays in the desired region most of the time as expected. Note that the flow rate of FTP connections are reduced without increasing the queue length $q_2(t)$ and the dropping probability dramatically when the bursty web traffic arrives at t=100. This is because that the available bandwidth for FTP connections is reduced and FTP senders see a longer round-trip time (longer packet delay at q_2, see Figure 5.16).

Figures 5.15 and 5.16 show the packet losses and delays for FTP, web and CBR connections respectively. Table 5.4 collects the corresponding performance metrics. Comparing to Figure 5.7, 5.8 and Table 5.3, the packet loss rate of FTP connection is reduced from 2.747% to 0.988% at the cost of packet delay (increased from $0.184s$ to $0.318s$). Since the average queue length at the Adaptive RED queue is about $80KBytes$ instead of $60KBytes$ in the fixed threshold scheme, web and UDP packets see a smaller shared buffer at the drop-tail queue and experience a higher loss rate from 1.278% to 2.306% and from 0.300% to 0.519%, respectively.

This situation can be improved by increasing the total buffer size despite that this approach is general useless or even harmful in a single RED queue or a single drop-tail queue. On the other hand, the average delays of web and UDP packets are slightly shorter for a smaller shared buffer space at the drop-tail queue. Finally, Tables 5.3 and 5.4 also show that the throughputs for the fixed threshold scheme and the dynamic threshold scheme are almost the same.

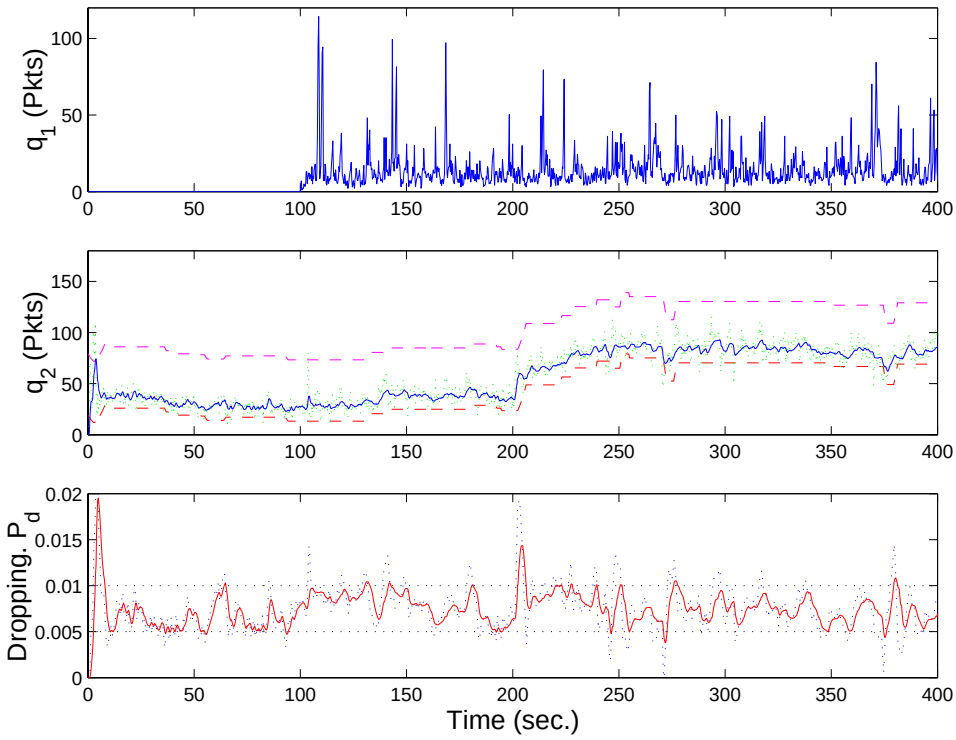

Figure 5.14: Dynamic threshold scheme: Virtual queue lengths of RED+Tail and dropping probability of the Adaptive RED queue, 10 FTPs starting at t=0 and 800 WEBs + 1 CBR starting at t=100.

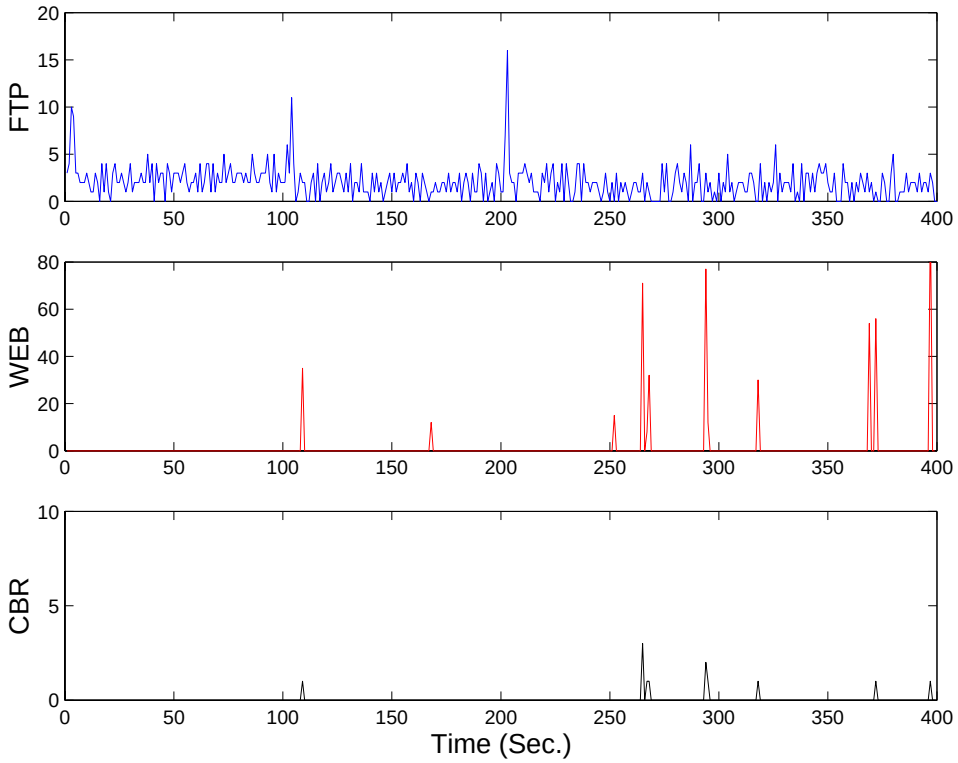

Figure 5.15: Dynamic threshold scheme: Packet losses (packets/sec.) of RED+Tail.

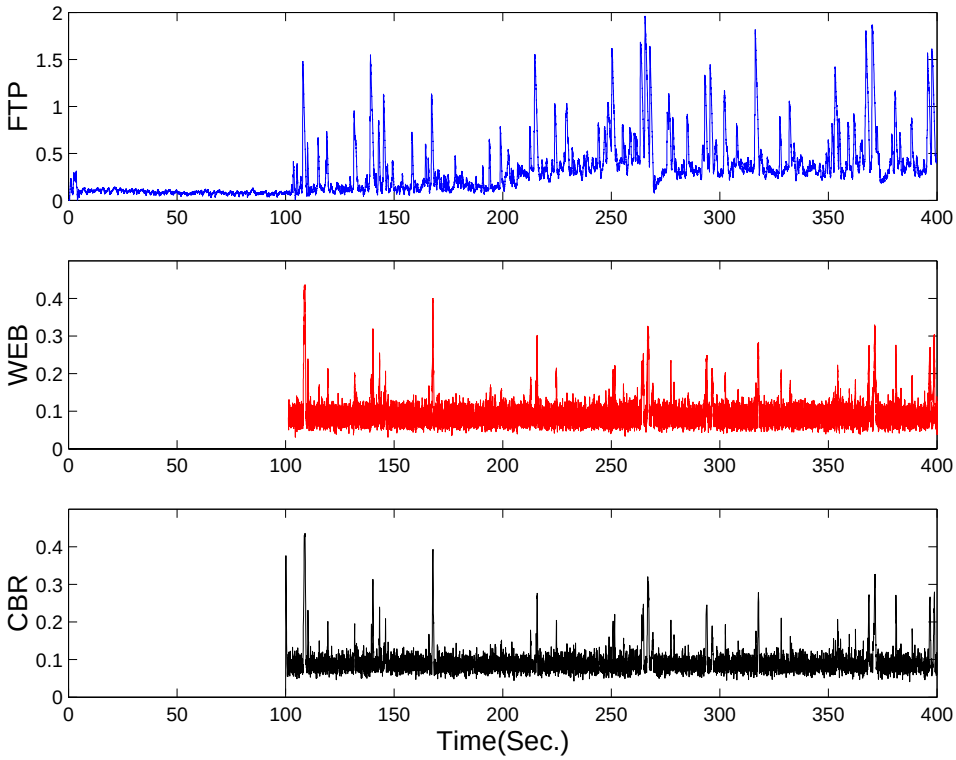

Figure 5.16: Dynamic threshold scheme: Packet delays (sec.) of RED+Tail.

5.6 Summary

In this chapter, we first demonstrated the vulnerability of Adaptive RED scheme to bursty traffic and then proposed a parallel virtual queue structure to eliminate unnecessary packet loss. A simple detection algorithm is employed to separate the short-life and long-life TCP connections into different virtual queues. The packet loss rate and mean delay can be greatly reduced by dynamic bandwidth allocation and active queue management with a parallel queue structure. This scheme combines the advantages of drop-tail and Adaptive RED policies. The simulation results in the study show that this scheme achieves a shorter mean delay for real time applications and keeps a high throughput for the best effort connections as well as greatly reduces the packet loss rate in both queues.

This parallel queue structure also provides more degree of freedom to control the router by considering different bandwidth allocation policies and dynamic thresholds for Adaptive RED. Here, the bandwidth allocation policy is a simple function of the current virtual queue length. However, it is well-known that web traffic is strongly correlated and has a long range dependency property. Based on observations of the "recent past" traffic, the future bandwidth demand of the web traffic was shown to be predictable. In future work, we will consider the optimal bandwidth allocation policy based on the prediction of congestion level.

Chapter 6

Performance Analysis of Active Queue Management in a Parallel Queue Structure

Joint analysis of the parallel queue is involved. Instead the queues are analyzed separately in Section 6.1 and 6.2, respectively. For the drop-tail queue, the emphasis is put on packet delay; for the Adaptive RED queue (with dynamic thresholds), the stability of queue length is the primary concern.

6.1 Drop-Tail Queue with Adaptive Service Rate

First, we investigate the queuing delay of CBR and web traffic at the drop-tail queue. Note that the service rate $C_1(t)$ of this queue is a function of $minth_1$, $maxth_1$ and the current length of drop-tail queue $q_1(t)$ (Figure 6.1). For ease of presentation, we let $minth_1 = 0$ and $C_{1max} = C$. When a packet enters the drop-tail queue at time t, it sees an instant queue length $q_1(t)$ and a service rate

$$C_1(t) \stackrel{\triangle}{=} \min(\frac{q_1(t)C}{maxth_1}, C). \tag{6.1}$$

General analysis of mean packet delay is difficult. However, the dynamic bandwidth $C_1(t)$ is updated periodically in both practice and simulation. If we assume that the update period is $S = \frac{maxth_1}{C}$ and the packets belonging to the same slot have uniformly distributed arriving times, there exists a simple lower bound for the mean packet delay. The impulse at $t = iS$ in Figure 6.2 denotes the new packets that arrive uniformly in the time period $[(i-1), iS)$. The absolute value of the slope indicates the bandwidth $C_1(iS)$, which is dependent on $q_1(iS)$. If $q_1(iS)$ is less than $maxth_1$, $C_1(iS) = \frac{q_1(iS)C}{maxth_1}$. It is evident that each packet in this slot has a constant delay $maxth_1/C$. If $q_1(iS) \geq maxth_1$, $C_1(iS) = C$ and the average packet delay in this slot is larger than $maxth_1/C$. Although the bound $\frac{maxth_1}{C}$ is derived based on special assumptions, it appears to provide a good approximation to the lower bound of mean packet delay for general cases (refer to Figure 6.3).

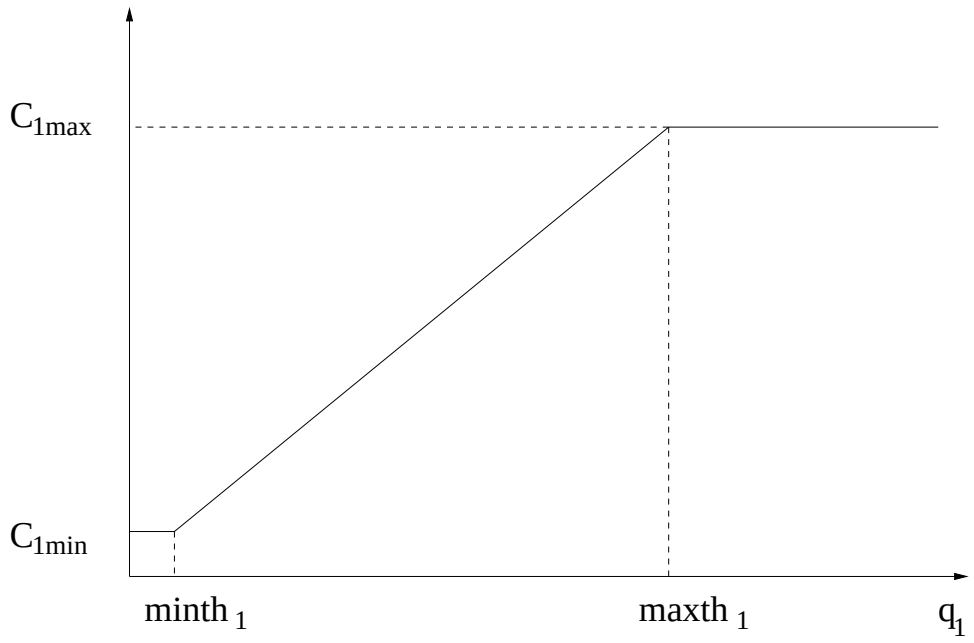

Figure 6.1: Dynamic bandwidth allocation at the drop-tail queue: $C_1(t)$=max(C_{1min}, min($\frac{q_1(t)C}{maxth_1}$, C_{1max})).

The experiment in Section 5.4 is re-conducted with parameters listed in Table 5.2 except $maxth_1$ being varied from $10KB$ to $80KB$. Figure 6.3 shows the mean packet delay of CBR and web traffic at the drop-tail queue and that of FTP traffic at the Adaptive RED queue as $maxth_1$ is varied. Note that the packet delay at the Adaptive RED queue with fixed thresholds is almost a constant even when $maxth_1$ is decreasing. That is because the Adaptive RED queue has fixed thresholds ($minth_2$, $maxth_2$) and the average queue length of RED queue is around $\bar{q}_2$=($minth_2$+$maxth_2$)/2. Let $\bar{C}_1$ denote the average bandwidth for the first queue. Then the mean delay at the Adaptive RED queue is around $\frac{\bar{q}_2}{C-\bar{C}_1}$.

For some real time applications such as video conference and voice, small delay jitter is very important for the connection quality. Figure 6.3 also shows that the drop-tail queue has a very small delay variance. Note that the delay variance at

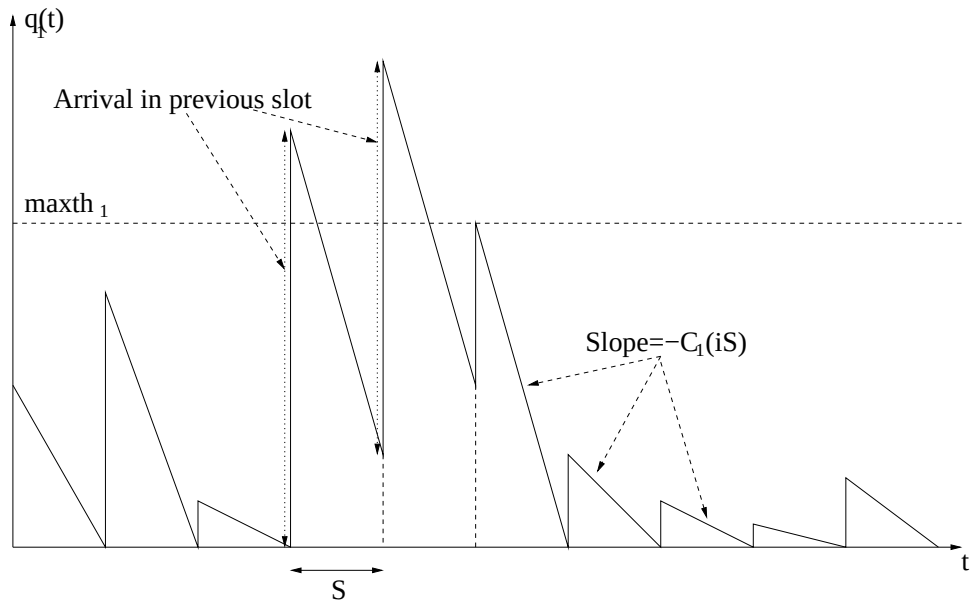

Figure 6.2: Drop-tail queue length with a time varying bandwidth $C_1(t)$.

the Adaptive RED queue is slightly increased when a smaller value of $maxth_1$ at the drop-tail queue is applied. According to these results, the mean packet delay requirements at both queues can be satisfied by properly designing the values of ($minth_1$, $maxth_1$) and ($minth_2$, $maxth_2$).

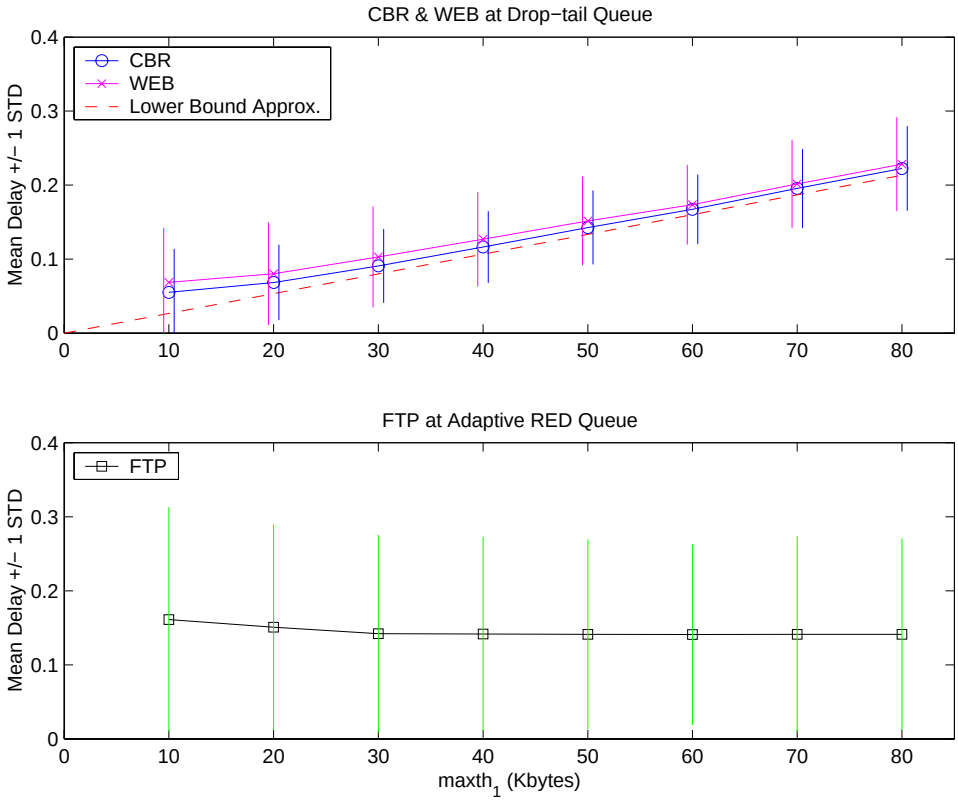

Figure 6.3: Mean delays (sec.) of CBR and WEB packets at the drop-tail queue and mean delay of FTP packets at the Adaptive RED queue (with fixed thresholds) with $maxth_1$= 10, 20,...,80 (KBytes).

6.2 Adaptive RED Queue with Dynamic Thresholds

In Chapter 5 we proposed the modified Adaptive RED scheme with dynamic thresholds in the parallel queue structure for controlling the flow rate of non-real time applications. The maximum threshold $maxth_2$ and minimum threshold $minth_2$ are changed dynamically to keep the packet dropping probability P_d within a desired small region (P_L, P_U) at the cost of packet delay variation. In this section we analyze issues related to the stability of this virtual queue. For ease of analysis, it is assumed that the dropping probability P_d of the Adaptive RED at the bottleneck router is fixed so that the average flow rate of each TCP connection can be approximated by a simple function of its round-trip time. Note that this assumption is not very restrictive considering the interval $(P_L,\ P_U)$ is small.

Consider N persistent TCP flows. To simplify analysis, it is assumed that the service rate C_2 of the Adaptive RED queue is constant, $C_2 = C - \bar{C}_1$, where $\bar{C}_1$ is the average bandwidth for the drop-tail queue. Define T_k^n as the average flow rate of the k^{th} TCP connection during time slot n. Let d'_k be the link round-trip propagation delay of connection k. At the beginning of time slot n the k^{th} connection sees a round-trip time R_k^n, which is equal to the sum of link propagation delay and the average queuing delay in the forward direction q^n/C_2 and in the backward direction q_b^n/C_2:

$$R_k^n = d'_k + \frac{q^n}{C_2} + \frac{q_b^n}{C_2}, \tag{6.2}$$

where q^n and q_b^n are the forward queue length and the backward queue length (of this Adaptive RED queue) at the beginning of time slot n, respectively. We assume that congestion only happens in the forward direction and the queuing delay q_b^n/C_2

in the backward direction is a constant. Hence we can write $R_k^n = d_k + q^n/C_2$ with $d_k = d'_k + q_b^n/C_2$.

Based on the assumption of fixed dropping probability at the router, each TCP connection experiences a fixed packet loss rate P_d and the corresponding average congestion window size is assumed to be a constant $\bar{W}$. Hence, the average flow rate T_k^n of the k^{th} TCP connection at slot n is

$$T_k^n = \frac{\bar{W}}{R_k^n} + E_k^n \tag{6.3}$$

where E_k^n is a white Gaussian process with zero mean and variance σ^2 modeling the flow rate perturbation of the k^{th} connection at slot n.

Given the arrival rate of each TCP connection, the dynamics of queue length q^n follows the *Lindley* equation:

$$q^{n+1} = min\{B, max[0, q^n + (\sum_{k=1}^{N} T_k^n - C_2)S]\}, \tag{6.4}$$

where B is the buffer size and S is the duration of one time slot. We list the parameter definitions as follows:

- T_k^n : average flow rate of TCP connection k at time slot n
- E_k^n : perturbation of flow rate (modeled by a white Gaussian process $\mathcal{N}(0, \sigma^2)$
- $\bar{W}$: average congestion window size
- C_2 : link bandwidth
- d_k : link round-trip propagation delay d'_k + backward queuing delay q_b/C_2
- q^n: forward link queue length at the beginning of time slot n
- S : duration of one time slot

- B : buffer size
- N : number of TCP connections

Since the queue length of Adaptive RED is mostly operated in a region far from the boundary, we first ignore the *max* and *min* operations in (6.4) and have a simplified nonlinear dynamic system:

$$q^{n+1} \quad = \quad f(q^n) + \xi^n, \tag{6.5}$$

where

$$f(q^n) \stackrel{\triangle}{=} q^n + S\{(\sum_{k=1}^{N} \frac{\bar{W}C_2}{q^n + d_k C_2}) - C_2\}, \tag{6.6}$$

and

$$\xi^n \stackrel{\triangle}{=} S \sum_{k=1}^{N} E_k^n. \tag{6.7}$$

To avoid the trivial case $q \equiv 0$, we assume that the sum of possible peak rates of all connections is greater than the link bandwidth at the bottleneck router:

$$\sum_{k=1}^{N} \frac{\bar{W}}{d_k} \geq C_2. \tag{6.8}$$

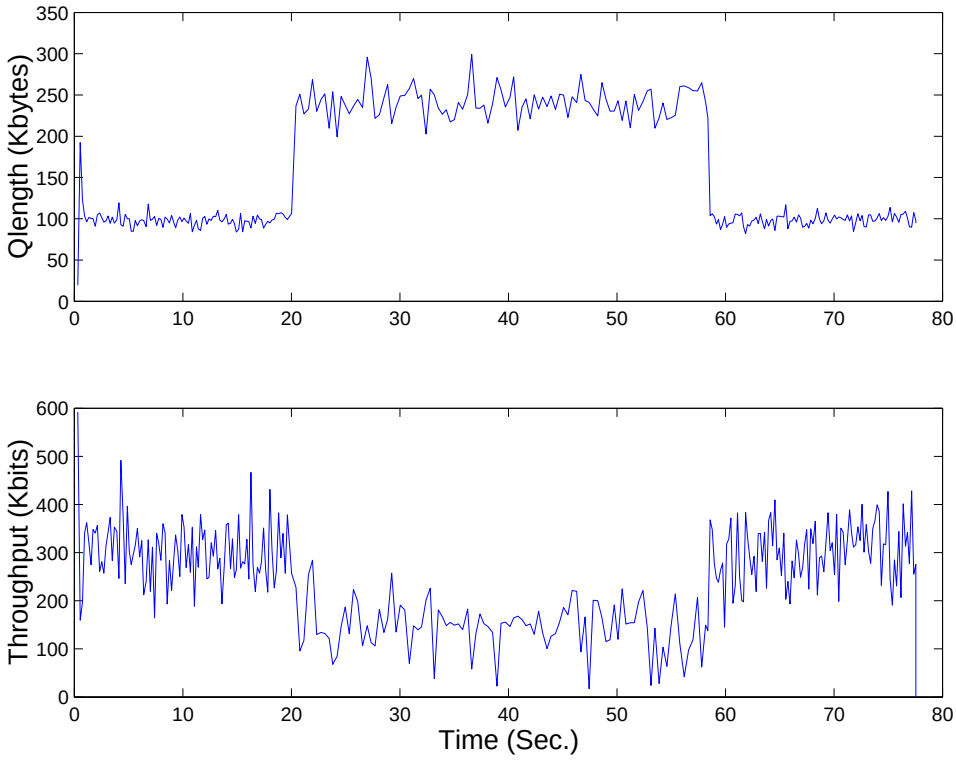

Figure 6.4: Queue length and TCP throughput (of a single connection) with C_2=6Mbps, d_k=64ms, $\bar{W}$=6.02×10^4 bits. Compare with simulation in Fig.5.12.

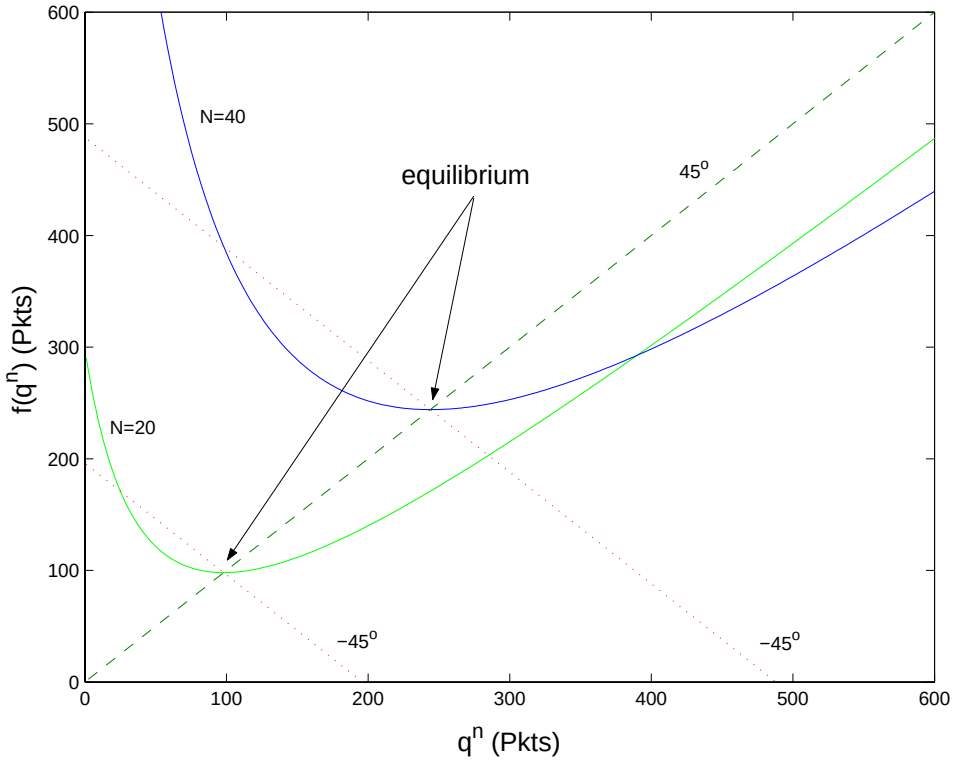

Figure 6.5: Mapping functions and equilibrium points when N=20, 40 with S=RTT.

Figure 6.4 shows the queue length dynamics (and the throughput of a persistent TCP connection) based on the model (6.5), where the flow rate deviations $\sigma =$ 77021, 128490 (bits/s) for N=20, 40 are measured from the simulation in Chapter 5, respectively. For both $N = 20$ and $N = 40$, Figure 6.4 shows consistent steady state behavior with simulation results in Figure 5.12. The mapping $f(\cdot)$ is plotted in Figure 6.5 for $N = 20$ and $N = 40$.

We first analyze the stability of the equilibrium of the model (6.5) when there is no flow disturbance., *i.e.*, $E_k^n = 0$. An equilibrium q_e of $q^{n+1} = f(q^n)$ should satisfy

$$\sum_{k=1}^{N} \frac{\bar{W}}{d_k + q_e^n/C_2} = C_2. \tag{6.9}$$

Now, we show the mapping function f is strictly convex and q_e is the unique solution. According to (6.6), we have the first and second derivatives of f:

$$f^{'}(q) = \quad 1 + S\bar{W}C_2 \textstyle\sum_{k=1}^{N} \frac{-1}{(q+d_kC_2)^2} < 1, \ \forall\ 0 < q < \infty \tag{6.10}$$

$$f^{''}(q) = \quad 2S\bar{W}C_2 \textstyle\sum_{k=1}^{N} \frac{1}{(q+d_kC_2)^3} > 0, \ \forall\ 0 < q < \infty. \tag{6.11}$$

Note that f, $f^{'}$ and $f^{''}$ are continuous and differentiable for all $0 < q < \infty$.

Lemma 6.2.1 *f is strictly convex for all $q > 0$.*

Proof *Since $d_k > 0$, according to (6.11) we have $f^{''} > 0$ for all $0 < q < \infty$, which indicates that f is a strictly convex function.*

Equation (6.10) indicates that the slope of f is less than 1 for all $0 < q < \infty$.

Lemma 6.2.2 *q_e is unique in the region $(0, \infty)$.*

Proof *Since $\sum_{k=1}^{N} \frac{\bar{W}}{d_k} \geq C_2$ by assumption, (6.9) has a solution q_e in $[0, \infty)$. q_e is located at the intersection of the graph of f with the 45^o line (see Figure 6.5).*

Furthermore, $f' < 1$ implies there is no other intersection in (q_e, ∞) and shows q_e is unique in the region $(0, \infty)$.

It is well known that q_e is locally asymptotically stable if $|f'(q_e)| < 1$. In the following we give conditions for q_e to be *globally* asymptotically stable.

Proposition 6.2.3 *If the rate update interval S satisfies*

$$S < \frac{2C_2}{\bar{W}(\sum_{k=1}^{N} d_k^{-2})}, \tag{6.12}$$

the equilibrium q_e is globally asymptotically stable. Furthermore, $|q^n - q_e| < \rho^n |q^0 - q_e|$ for some $\rho \in (0,1)$ dependent on q^0.

Proof First we observe that the function f is convex since

$$f''(q) = \sum_{k=1}^{N} \frac{2S\bar{W}C_2}{(q + d_k C_2)^3} > 0, \quad \forall q \in [0, \infty). \tag{6.13}$$

For any B_0 such that $B_0 > q_e$ and

$$B_0 \geq f(0) = (\sum_{k=1}^{N} \frac{\bar{W}}{d_k} - C_2)S, \tag{6.14}$$

one can verify that f maps $[0, B_0]$ to $[0, B_0]$ due to convexity of f and

$$f'(q) = 1 - S\sum_{k=1}^{N} \frac{\bar{W}C_2}{(q + d_k C_2)^2} < 1, \quad \forall q \in [0, \infty). \tag{6.15}$$

When restricted to $[0, B_0]$, $f'(q) \leq \rho_1$ with $\rho_1 \in (0,1)$. If (6.12) is satisfied, $f'(q) > -1$, $\forall q \in [0, \infty]$, and $f'(q) \geq -\rho_2$, $\forall q \in [0, B_0]$, with $\rho_2 \in (0,1)$.

Hence $|f'(q)| \leq \rho \stackrel{\triangle}{=} \max(\rho_1, \rho_2) < 1$ $\forall q \in [0, B_0]$, which implies that f is a contraction mapping on $[0, B_0]$. By the Contraction Mapping Principle [98],

$$|q^n - q_e| < \rho^n |q^0 - q_e|, \quad \text{if } q^0 \in [0, B_0]. \tag{6.16}$$

Since B_0 can be arbitrarily large (as long as $B_0 < \infty$), q_e is globally asymptotically stable. Note that the contraction constant ρ depends on B_0 and thus on q^0. □

From Proposition 6.2.3, when rate update is frequent enough, the equilibrium will be asymptotically stable (the equilibrium itself does not depend on S). Another sufficient condition for asymptotic stability is the following:

Proposition 6.2.4 *If $f'(q_e) \geq 0$, then q_e is a globally asymptotically stable.*

Proof *As shown in the proof of Lemma 6.2.1, f is strictly convex. If $f'(q_e) \geq 0$, graphical analysis revels that*

$$|q^{n+1} - q_e| \leq |q^n - q_e|,$$

where the equality holds if and only if $q^n = q_e$. The claim thus follows. □

For the homogeneous case $d_k = d$, we have $q_e = N\bar{W} - dC_2$. And the condition $f'(q_e) \geq 0$ is equivalent to $S \leq \frac{N\bar{W}}{C_2} = q_e/C_2 + d$. In other words, q_e is asymptotically stable if the rate update interval S is no larger than the round-trip time (RTT). Figure 6.10 shows the mapping f and the equilibrium q_e for different S. Figure 6.7 shows the queue length dynamics (noise is included) for $S{=}RTT$ and $2RTT$, respectively. We can see that in the case $S{=}RTT$, the queue length stays around q_e with small variation, while in the case $S{=}2RTT$, the queue length dynamics is much more chaotic.

For the heterogeneous case, a sufficient condition $S \leq (\frac{C_2}{W} - \frac{N(N-1)}{(q_e+D_m)^2})^{-1}$ for stability can be derived as follows:

Proposition 6.2.5 *For any $d_k > 0$, the system is stable if $S \leq (\frac{C_2}{W} - \frac{N(N-1)}{(q_e+D_m)^2})^{-1}$.*

Proof *According to Proposition 6.2.4, we want to find the condition of S so that*

$f^{'}(q_e) \geq 0$ *is satisfied. Let* $D_k := d_k C_2$ *and* $G := \bar{W} C_2$*, from (6.9) we have:*

$$\sum_{k=1}^{N} \frac{1}{q_e + D_k} = \frac{C_2}{G}. \tag{6.17}$$

from (6.10), we want to show $f^{'}(q_e) \geq 0$ *by showing that*

$$\sum_{k=1}^{N} \frac{1}{(q_e + D_k)^2} \leq \frac{1}{SG}. \tag{6.18}$$

Substitute (6.17) in (6.18), we have

$$\frac{C_2^2}{G^2} - \sum_{i \neq j} \frac{1}{q_e + D_i} \frac{1}{q_e + D_j} \leq \frac{1}{SG}. \tag{6.19}$$

Since $q_e > 0$ *and* $D_k > 0$*, a sufficient condition of the above inequality is:*

$$\frac{C_2^2}{G^2} - \frac{N(N-1)}{(q_e + D_m)^2} \leq \frac{1}{SG} \tag{6.20}$$

$$\Leftrightarrow S \leq (\frac{C_2}{\bar{W}} - \frac{N(N-1)}{(q_e + D_m)^2})^{-1}, \tag{6.21}$$

where $D_m := \min D_k$*. Hence* $S < (\frac{C_2}{\bar{W}} - \frac{N(N-1)}{(q_e + D_m)^2})^{-1}$ *is a sufficient condition for stability.* $\square$

Similar to the homogeneous case, if the rate update interval S is short enough, we can guarantee the stability in the heterogeneous case.

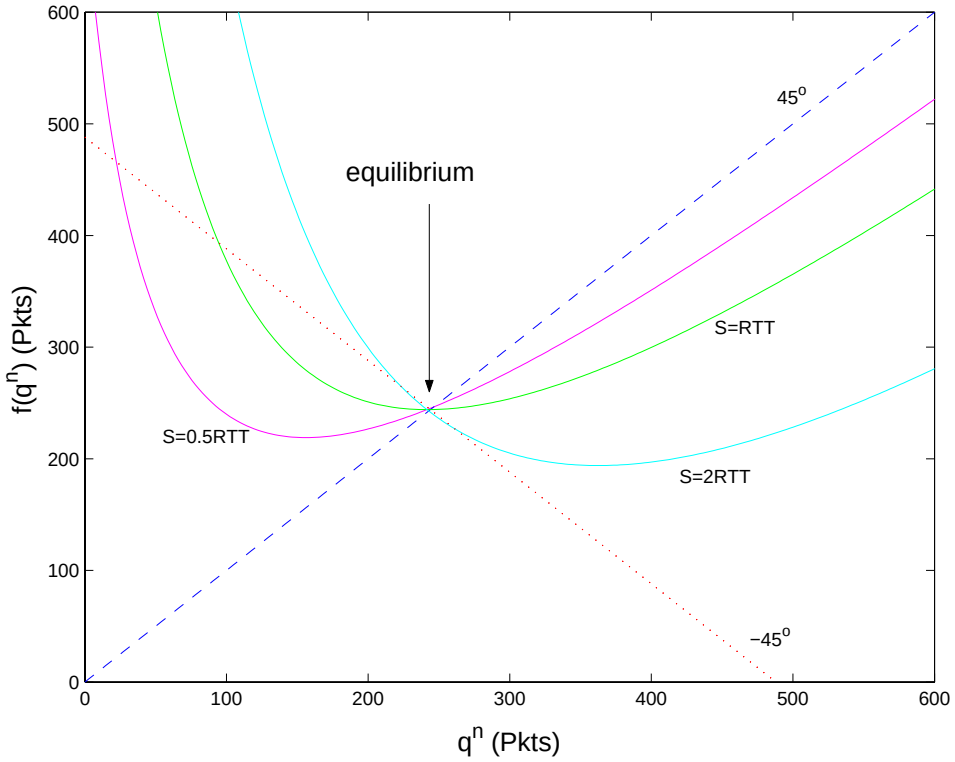

Figure 6.6: Mapping function and equilibrium point when N=40 with S=0.5RTT, 1RTT and 2RTT.

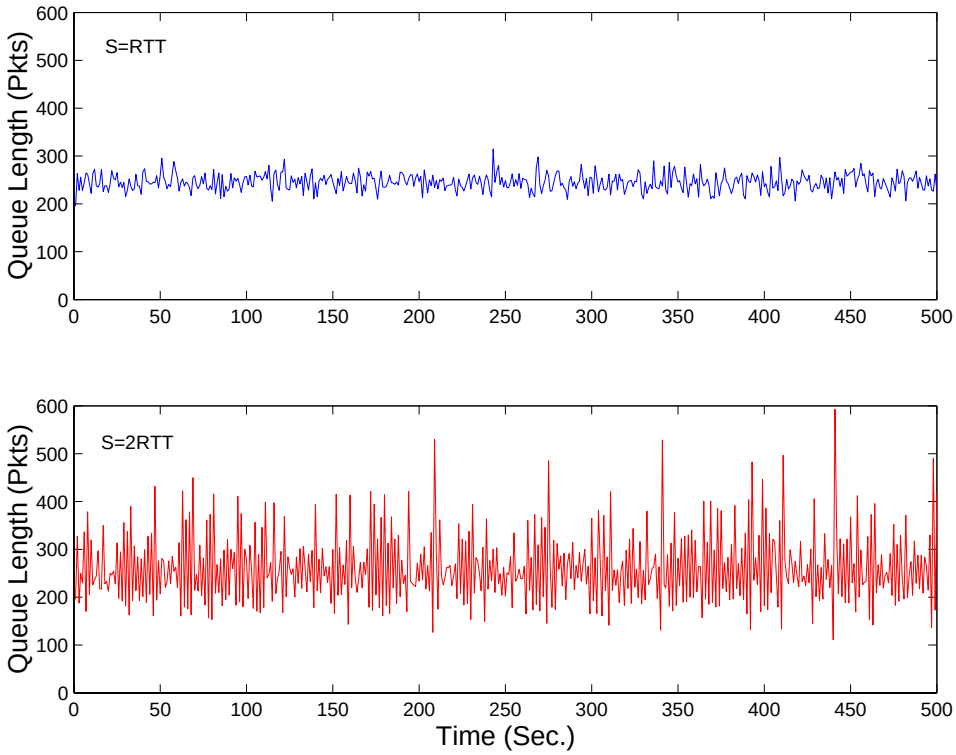

Figure 6.7: Queue length with N=40, S=RTT and 2RTT.

Next we consider the *Lindley* equation with finite random perturbation $\xi^n = S\sum_{k=1}^{N} E_k^n$:

$$q^{n+1} \stackrel{\triangle}{=} g(q^n, \xi^n) = \min\{B, \max[0, f(q^n) + \xi^n]\} \tag{6.22}$$

Note that since $\{E_k^n\}$ is white and stationary, so is $\{\xi^n\}$. It turns out that stability of the equilibrium of the deterministic system $q^{n+1} = f(q^n)$ is closely related to stochastic stability of the system (6.22).

Define a compact set $X{:=}[0, B]$ and the transformation $g{:}X \times W \to X$ of (6.22) is called *regular stochastic dynamic system* in [99], if the following assumptions hold.

(a) The random vectors $\xi^0, \xi^1, ...$, have value in W and have the same distribution.

(b) g is defined on the subset $X \times W$ of $R^d \times R^k$. The set $X \subset R^d$ is closed and $W \subset R^k$ is Borel measurable. For every fixed $y \in W$ the function $g(x, y)$ is continuous in x and for every fixed $x \in X$ it is measurable in y.

(c) The initial random vector x^0 has initial value in $X$and the vectors x^0, $\xi^0, \xi^1, ...$, are independent.

Definition 6.2.6 *[99] (***Foias operator***) Let a function $g{:}X \times W$ satisfies condition (b) and a probabilistic measure (supported on W) be given. Then the operator P: $M_{fin} \to M_{fin}$ given by*

$$P\mu(A) = \int_X \{\int_W 1_A(g(x, y))\nu(dy)\}\mu(dx) \tag{6.23}$$

will be called the **Foias operator** *corresponding to the dynamic system. M_{fin} denotes the subspace of finite measures.* □

Theorem 6.2.7 *[99] Krylov-Bogolubov Theorem Let P be the Faias operator to a regular stochastic dynamical system. Assume that there is a $\mu_0 \in M_1$ having the following property. For every $\varepsilon > 0$ there is a bounded set $B \in \boldsymbol{B}(X)$ such that*

$$\mu_n(B) = P^n\mu_0(B) \geq 1 - \varepsilon \quad for\ n = 0, 1, 2, ... \tag{6.24}$$

Then P has an invariant distribution. □

Proposition 6.2.8 *The stochastic system (6.22) admits an invariant probability measure μ^* for the queue length q^n. Furthermore, if the condition (6.12) on Proposition 6.2.3 is satisfied, this system is weakly asymptotically stable, i.e., the queue length distribution μ^n for q^n converges to μ^* weakly.*

Sketch of Proof. Since f is continuous and $\{\xi^n\}$ is identically and independently distributed, the system (6.22) is a regular stochastic dynamic system.

Since $[0, B]$ is compact, the system admits an invariant probability measure μ^* by the Krylov-Bogolubov Theorem. When condition (6.12) is satisfied, g is a contraction mapping with respect to its first argument, i.e.,

$$|g(x, \xi) - g(y, \xi)| < \rho|x - y|, \quad \forall x, y \in [0, B], \forall \xi, \tag{6.25}$$

where $\rho \in (0, 1)$. Hence the system is weakly asymptotically stable by Theorem 12.6.1 of [99]. □

The invariant probability measure μ^* has probability masses at $q = 0$ and $q = B$, and has probability density on $(0, B)$. An approximation to μ^* can be obtained by numerically advancing the probability distribution μ^n for the queue length q^n. We have discretized the queue length and consequently obtained a Markov chain for the dynamics of the queue length distribution.

Let the packet size have a fixed length L (bits), z^n:=$ceil(q^n/L)$ be the number of packets in the queue at time n and $\pi^n = [Pr(z^n = 0), ..., Pr(z^n = B)]$ denote

the corresponding probability vector. We have

$$\pi^{n+1} = \pi^n \mathcal{T}, \tag{6.26}$$

$$\pi^* = \pi^* \mathcal{T}, \tag{6.27}$$

where $\pi^* = \lim_{n\to\infty} \pi^n$ is the steady state distribution and

$$\mathcal{T}(i,j) := Pr[z^{n+1} = j | z^n = i] \tag{6.28}$$

is the corresponding transition matrix of the Markov chain. The conditional probability $Pr[z^{n+1} = j|z^n = i]$ is obtained as

$$Pr[j \leq (min\{B, max[0, f(iL) + \xi]\})/L < (j+1)]. \tag{6.29}$$

On the other hand, when the buffer size B is far greater than the equilibrium queue length and the perturbation magnitude is small, the transformation $g(q, \xi)$ can be linearized around the equilibrium point q_e. Let $Q^n \stackrel{\triangle}{=} q^n - q_e$. Then

$$Q^{n+1} \stackrel{\triangle}{=} \quad f^{'}(q_e)Q^n + \xi^n. \tag{6.30}$$

Since $\{\xi^n\}$ is white Gaussian process with zero mean and variance $NS\sigma^2$, $\{Q^n\}$ will be a Gaussian process with zero mean and normalized variance

$$Var[Q^n/S] = \frac{N\sigma^2}{1 - |f^{'}(q_e)|^2}. \tag{6.31}$$

From (6.31) the normalized queue length variation will be minimal if $f^{'}(q_e) = 0$, which corresponds to $S = RTT$ for the homogeneous case.

Figure 6.8 and 6.9 show the queue length distributions obtained through empirical estimation from ns2 simulation, numerical computation based on (6.26), and linear approximation based on (6.31), repectively. Three distributions agree well,

which verifies that our nonlinear model (6.22) captures the queue length dynamics under the Adaptive RED scheme with dynamic thresholds.

Figure 6.10 6.11 and 6.12 illustrate the Mapping functions, equilibrium points, queue length dynamics from $N = 20$, 40 to 8.

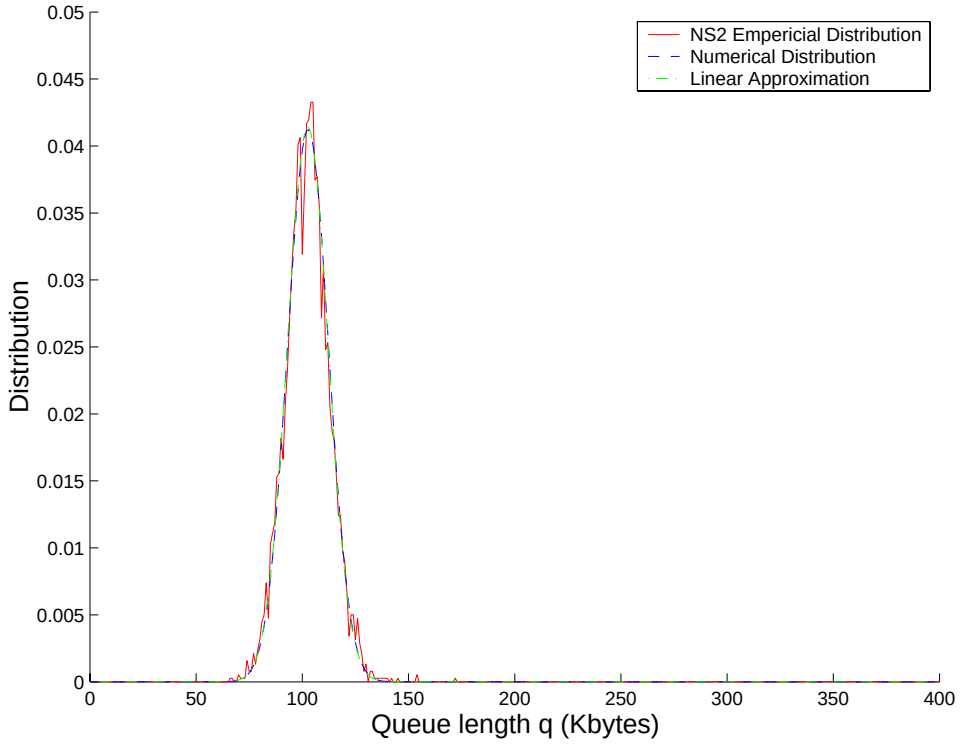

Figure 6.8: Steady state queue length distributions for N=20, S=RTT.

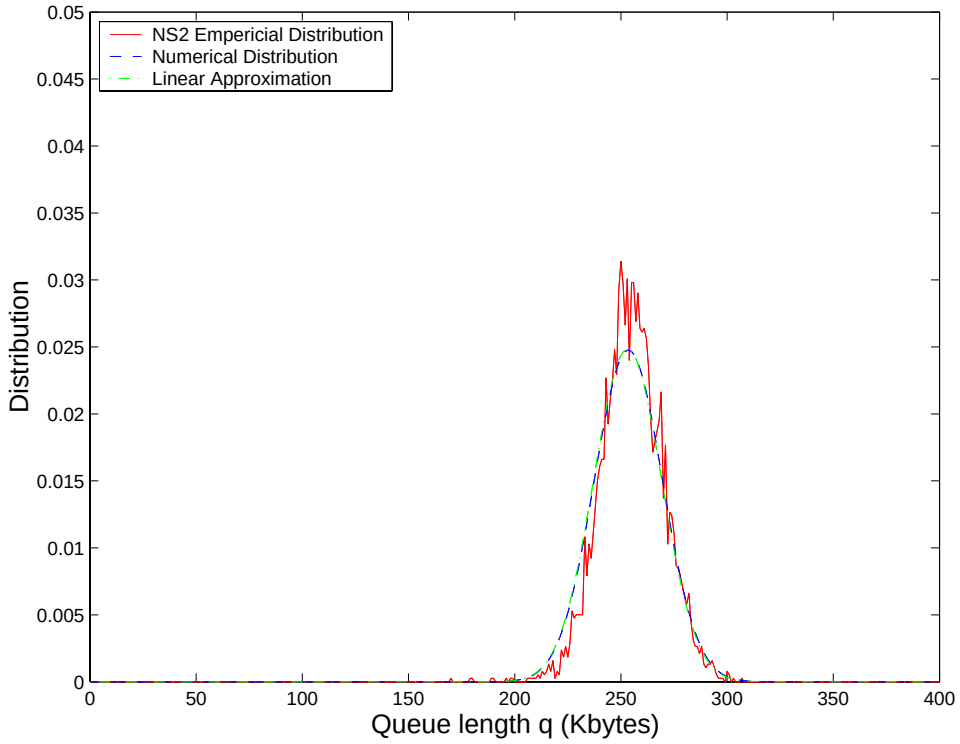

Figure 6.9: Steady state queue length distributions for N=40, S=RTT.

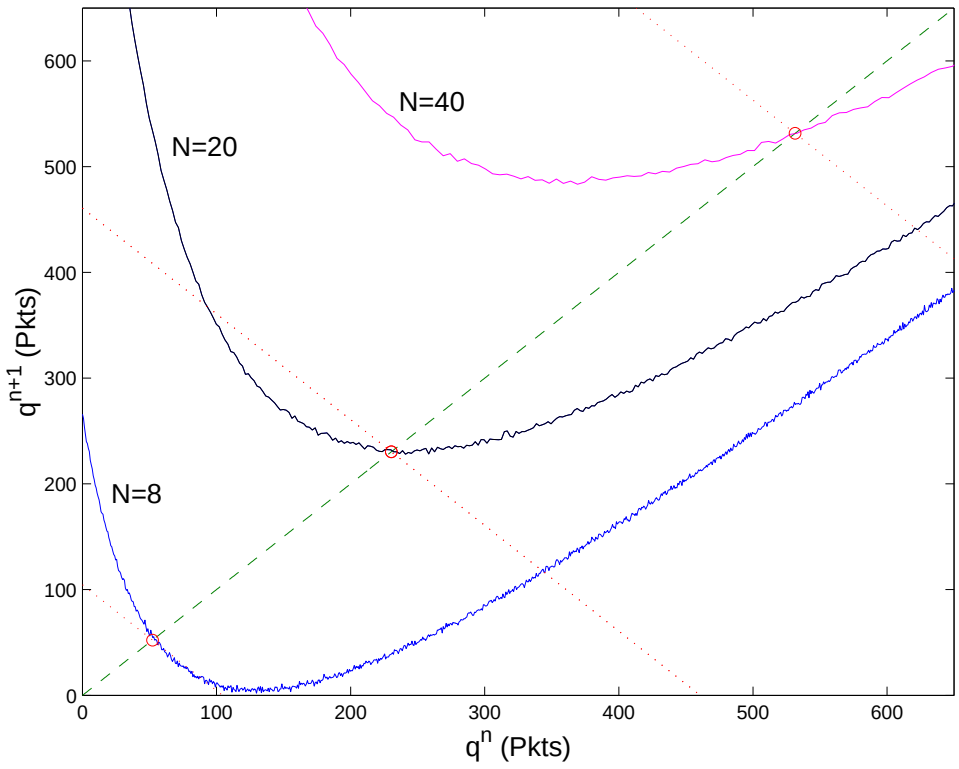

Figure 6.10: Mapping function and equilibrium point when N=20, 40 and 8 with S=0.5RTT, 1RTT and 2RTT.

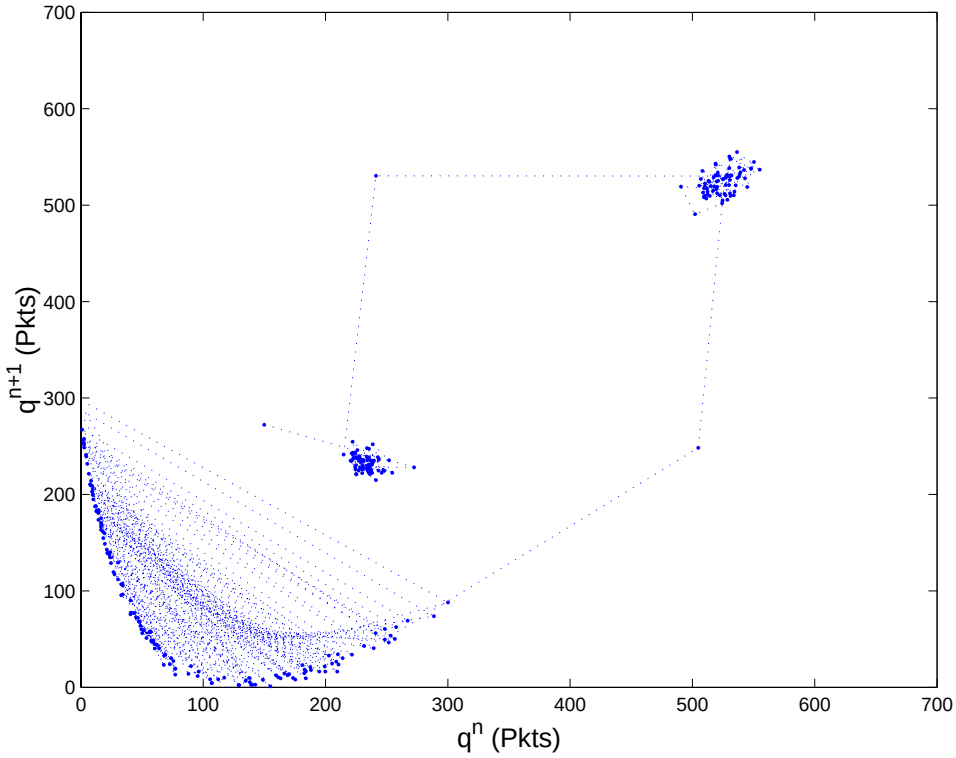

Figure 6.11: Queue length dynamics $N = 20 \rightarrow 40 \rightarrow 8$.

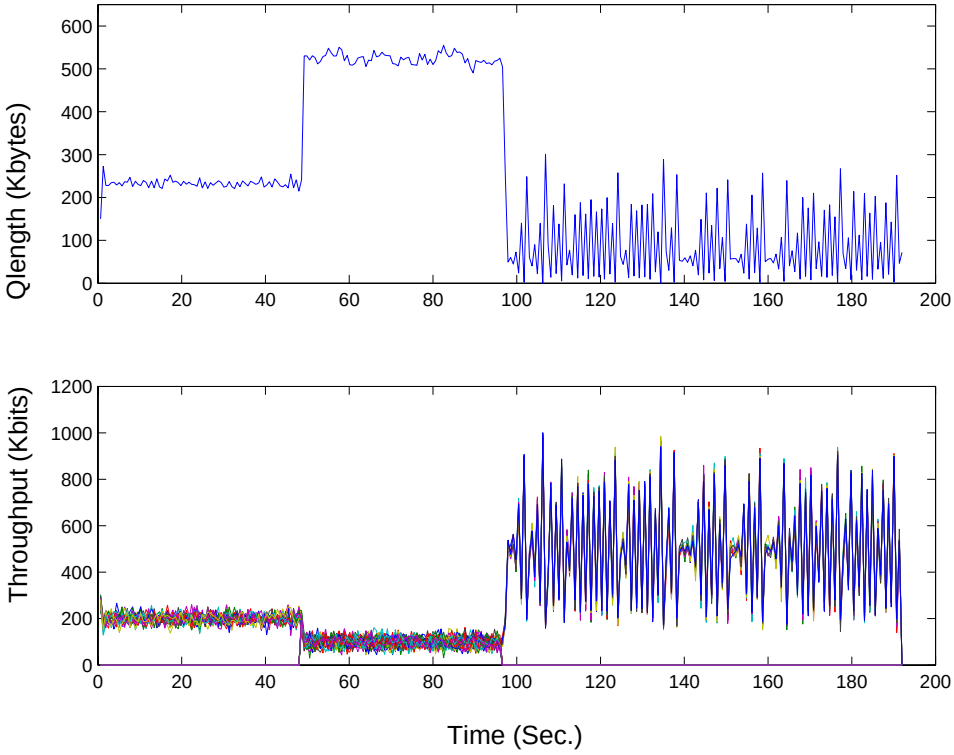

Figure 6.12: Queue length with $N = 20 \rightarrow 40 \rightarrow 8$.

6.3 Summary

In Chapter 5 and 6 we have first demonstrated the vulnerability of Adaptive RED scheme to bursty web traffic, and then proposed a parallel virtual queue structure for active queue management at the router. A simple detection algorithm is employed to separate the short-life and long-life TCP connections into different virtual queues. The packet loss rate and mean delay for short-life traffic can be greatly reduced by dynamic bandwidth allocation with this parallel queue structure. This scheme combines the advantages of drop-tail and Adaptive RED policies. The simulation results in the study show that this scheme achieves a shorter mean delay for real time applications and keeps a high throughput for the best effort connections as well as greatly reduces the packet loss rate in both queues.

This parallel virtual queue structure also offers more degrees of freedom for AQM due to its flexibility in accommodating variants of the Adaptive RED scheme and different dynamic bandwidth allocation algorithms. We have explored a modified Adaptive RED scheme with sliding queue length thresholds. This scheme is able to maintain the dropping probability within a small interval and improve the goodput of non-real time connections. The queue length variation under this policy has been analyzed and conditions for its stability have been given. The dynamic threshold Adaptive RED might also be useful for achieving throughput fairness among multiple RED queues.

As to the dynamic bandwidth allocation policy for the drop-tail queue, we only used the current virtual queue length information. However, it is well-known that web traffic is strongly correlated and has a long range dependency property. Based on observations of the "recent past" traffic, the future bandwidth demand of the web traffic is predictable. In future work optimal bandwidth allocation based on

prediction of the congestion level will be explored.

Chapter 7

Conclusions

In this dissertation, we first reviewed the theory of wavelet analysis for network traffic. We also discussed the monofractal and multifractal behaviors of Internet traffic at large and small time scales. In order to capture these mono/multi-fractal properties at all time scales, we proposed a multilevel ON/OFF model for the Internet traffic and developed an algorithm to estimate the model parameters from a real trace. The idea of this model is to imitate the TCP packet arrival pattern at the lower level and the connection arrival pattern at the upper level. A synthetic traffic was generated by the proposed model with the parameters estimated from a real trace. The wavelet analysis showed that this new model can successfully capture the statistical properties in second order (Logscale diagram) and higher orders (multifractal spectrum). The simulation results also showed that these two traffic share the same queuing behavior.

Since the Logscale diagram carries important statistical properties of the traffic at all time scales, we developed an approximation of Logscale diagram of the Internet traffic. The goal is to compute the Logscale diagram directly from the network parameters of the model instead of analyzing a long trace. Given the

Logscale diagram of the input traffic, one can immediately transfer the Logscale diagram to the traffic workload distribution at all time scales and predict the corresponding queuing behavior and network performance. Furthermore, our analysis results show that the queuing behavior with a small buffer size is dominated by the traffic behaviors at small time scales. It indicates that the traditional monofractal (self-similar) traffic model is not suitable for predicting the performance in real networks.

For some real time applications such as Constant Bit Rate (CBR) connections are sensitive to the mean and variance of packet delay. By applying our predicted results of queue length distribution and the properties of wavelet analysis, we developed a fast algorithm to estimate the mean and variance of packet delay in real time. This tool could help the network resource providers decide how much bandwidth should be allocated to guarantee the quality of service for certain applications.

In the second part of this thesis, we are focusing on the policy of buffer management. Many Active Queue management schemes such as RED were proposed to improve the TCP throughput and link utilization. The basic assumption of these policies is assuming that the TCP is operated in the Congestion Avoidance phase. However, the current Internet traffic is dominated by web traffic and most web connections are operated in the slow start phase. We indicated that these AQM policies implemented in a shared queue is not suitable for the current Internet traffic. The simulation results demonstrated that the web traffic severely degrades the performance of those AQM schemes. Thus, we proposed a parallel queue structure for a better solution of buffer management. Since TCP connection is untamed in the slow start phase, we invented a detection algorithm to prevent the untamed

TCP from entering into the AQM queue. In addition, the parallel structure also gives us more freedom to control the AQM scheme. A typical AQM scheme such as RED relies on dropping packets actively to control the TCP flow rate. This method wastes bandwidth resources and caused unnecessary re-transmissions. For non-real time application such as transmitting a large data file by FTP, packet delay is not an important issue. Under this structure, we have the freedom to control the flow rate of non-real time application by increasing the queuing delay instead of dropping the packet.

Appendix A

Appendix

A.0.1

Proposition:3.3.3

$$\lim_{t\to\infty}\frac{Y(t)}{\widehat{Y}(t)}=E[V]B.$$

Proof: *Since there are N independent user connections, we let $N=1$ without loss of generality.*

$$\begin{aligned}
\lim_{t\to\infty}\frac{Y(t)}{\widehat{Y}(t)} &= \lim_{t\to\infty}\frac{B\int_0^t U_k(u)V_k(u)du}{\int_0^t U_k(u)du} \\
&= B\lim_{t\to\infty}\frac{\sum_{i=1}^m\int_{t_{i-1}}^{t_i}U_k(u)V_k(u)du+\int_{t_m}^t U_k(u)V_k(u)du}{\sum_{i=1}^m\int_{t_{i-1}}^{t_i}U_k(u)du+\int_{t_m}^t U_k(u)du} \\
&= B\lim_{t\to\infty}\frac{\sum_{i=1}^m\int_{t_{i-1}}^{t_i}V_k(u)du+\int_{t_m}^t V_k(u)du}{\sum_{i=1}^m(t_i-t_{i-1})+t-t_m} \\
&= B\lim_{t'\to\infty}\frac{\int_0^{t'}V_k'(u)du}{t'} \\
&= B\frac{ET_{21}}{ET_{21}+ET_{20}} \\
&= BE[V] \qquad \text{(A.1)}
\end{aligned}$$

where t_{i-1} and t_i are the beginning and end points of the ON state, t_m is the maximal end point which is less than t, and $V'(t)$ is a shifted version of $V(t)$.

A.0.2

Proposition 3.3.6:

Without loss of generality, let $\Delta = 1$ and $\sigma_0^2 = EG_i^2 = 1$. From (2.32),

$$\begin{aligned} R_G(k) &= EG_0EG_k \\ &= E(B_H(1) - B_H(0))(B_H(K+1) - B_H(k)) \\ &= Cov(B_H(1)B_H(k+1)) - Cov(B_H(1)B_H(k)) \\ &= \frac{1}{2}(|k+1|^{2H} - 2|k|^{2H} + |k-1|^{2H}) \end{aligned} \tag{A.2}$$

and

$$\begin{aligned} &\lim_{k\to\infty} R_G(k) \\ &\qquad = \lim_{k\to\infty} \frac{1}{2}(|k+1|^{2H} - 2|k|^{2H} + |k-1|^{2H}) \\ &\qquad = \lim_{k\to\infty} \frac{1}{2}k^{2H-2}[k^2(|1+\frac{1}{k}|^{2H} - 2 + |1-\frac{1}{k}|^{2H})] \end{aligned}$$

Note that

$$\begin{aligned} &\lim_{x\to 0} \frac{(1+x)^{2H} - 2 + (1-x)^{2H}}{x^2} \\ = &\lim_{x\to 0} \frac{2H(1+x)^{2H-1} - 2H(1-x)^{2H-1}}{2x} \\ = &\lim_{x\to 0} \frac{2H(2H-1)(1+x)^{2H-2} + 2H(2H-1)(1-x)^{2H-2}}{2} \\ = &\ 2H(2H-1) \end{aligned}$$

A.0.3

Show equations (4.54) and (4.55) with a Lognormal distribution $f_A(x)$:

$$f_A(x) := \frac{1}{x\sigma\sqrt{2\pi}} \exp[-\frac{(\ln x - \mu)^2}{2\sigma^2}], \quad x > 0 \tag{A.3}$$

and

$$erfc(x) := \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt \tag{A.4}$$

$$\begin{aligned} E[(A-d)^+] &= \int_d^\infty (x-d) f(x) dx \\ &= \int_d^\infty x f(x) dx - d\bar{F}(d) \end{aligned} \tag{A.5}$$

where

$$\begin{aligned} \int_d^\infty x f(x) dx &= \frac{1}{\sigma\sqrt{2\pi}} \int_d^\infty \exp(-\frac{(\ln x - \mu)^2}{2\sigma^2}) dx \\ &= \frac{1}{\sigma\sqrt{2\pi}} \int_{\frac{\ln d - \mu}{\sigma\sqrt{2}}}^\infty \exp(-y^2) \exp(\mu) \sqrt{2}\sigma \exp(\sqrt{2}\sigma y) dy \\ &= \frac{\exp(\mu + \sigma^2/2)}{\sqrt{\pi}} \int_{\frac{\ln d - \mu}{\sigma\sqrt{2}}}^\infty \exp(-y^2 - \sqrt{2}\sigma y + \frac{\sigma^2}{2}) dy \\ &= \frac{\exp(\mu + \sigma^2/2)}{\sqrt{\pi}} \int_{\frac{\ln d - \mu}{\sigma\sqrt{2}}}^\infty \exp(-(y - \frac{\sigma}{\sqrt{2}})^2) dy \\ &= \frac{\exp(\mu + \sigma^2/2)}{\sqrt{\pi}} \int_{\frac{\ln d - \mu - \sigma^2}{\sigma\sqrt{2}}}^\infty e^{-t^2} dt \\ &= \frac{e^{\mu + \sigma^2/2}}{2} erfc(\frac{\ln d - \mu - \sigma^2}{\sigma\sqrt{2}}) \end{aligned} \tag{A.6}$$

$$\begin{aligned} E[\{(A-d)^+\}^2] &= \int_d^\infty (x-d)^2 f(x) dx \\ &= \int_d^\infty x^2 f(x) dx - 2d \int_d^\infty x f(x) dx + d^2 \bar{F}(x). \end{aligned} \tag{A.7}$$

where

$$\begin{aligned}
\int_d^\infty x^2 f(x)dx &= \frac{1}{\sigma\sqrt{2\pi}}\int_d^\infty x\exp(-\frac{(\ln x-\mu)^2}{2\sigma^2})dx \\
&= \frac{e^{2\mu+2\sigma^2}}{\sqrt{\pi}}\int_{\frac{\ln d-\mu}{\sigma\sqrt{2}}}^\infty \exp(-(y-\sqrt{2}\sigma)^2)dy \\
&= \frac{e^{2\mu+2\sigma^2}}{\sqrt{\pi}}\int_{\frac{\ln d-\mu-2\sigma^2}{\sigma\sqrt{2}}}^\infty e^{-t^2}dt \\
&= \frac{e^{2\mu+2\sigma^2}}{2}erfc(\frac{\ln d-\mu-2\sigma^2}{\sigma\sqrt{2}}) \qquad \text{(A.8)}
\end{aligned}$$

BIBLIOGRAPHY

[1] M. Taqqu and V. Teverosky, "Is network traffic self-similar or multifractal?," *Fractals*, vol. 5, no. 1, pp. 63–73, 1997.

[2] R. Riedi and J.L. V'ehel, "Multifractal properties of tcp traffic: A numerical study," Tech. Rep. 3129, INRIA, Feb. 1997.

[3] V. Paxson and S. Floyd, "Why we don't know how to simulate the internet," in *Winter Simulation Conference*, 1997, pp. 1037–1044.

[4] A. Erramilli, O. Narayan, and W. Willinger, "Experimental queueing analysis with long-range dependent packet traffic," *IEEE/ACM Transactions on Networking*, vol. 4, no. 2, pp. 209–223, 1996.

[5] A. Veres and M. Boda, "The chaotic nature of TCP congestion control," in *Proceedings of INFOCOM*, 2000, pp. 1715–1723.

[6] M. Crovella and A. Bestavros, "Explaining world wide web traffic self-similarity," Tech. Rep. 1995-015, CS Dept. Boston University, 1995.

[7] L. Guo, M. Crovella, and I. Matta, "TCP congestion control and heavy tails," Tech. Rep. 2000-017, CS Dept. Boston University, Mar. 2000.

[8] L. Guo, M. Crovella, and I. Matta, "How does TCP generate pseudo-self-similarity?," Tech. Rep. 2001-014, CS Dept. Boston University, Dec. 2001.

[9] M.E. Crovella and A. Bestavros, "Self-similar in world wide web traffic: Evidence and possible causes," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 835–846, 1997.

[10] M. Grossglauser and J. Bolot, "On the relevance of long-range dependence in network traffic," *IEEA/ACM Transactions on Networking*, vol. 7, no. 5, pp. 629–640, Oct. 1999.

[11] K. Park, G. Kim, and M. Crovella, "On the relationship between file sizes, transport protocols, and self-similar network traffic," in *Proceedongs of IEEE International Conference on Network Protocols*, 1996, pp. 171–180.

[12] M.M. Krunz and A.M. Makowski, "Modeling video traffic using M/G/∞ input processes: A compromise between Markovian and LRD models," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 5, pp. 733–748, 1998.

[13] A. Feldmann, A.C. Gilbert, P. Huang, and W. Willinger, "Dynamics of IP traffic: A study of the role of variability and the impact of control," in *Proceedings of SIGCOMM*, 1999, pp. 301–313.

[14] D. Veitch, J. Bäckar, J. Wall, J. Yates, and M. Roughan, "On-line generation of fractal and multi-fractal traffic," in *PAM2000 Workshop on Passive and Active Networking, Hamilton, New Zealand.*, 2000.

[15] J. Gao and I. Rubin, "Multiplicative multifractal modeling of long-range-dependent traffic," in *Proceedings of ICC*, 1999.

[16] M. Taqqu, W. Willinger, and R. Sherman, "Proof of a fundamental result in self-similar traffic modeling," *Computer Communication Review*, vol. 27, no. 2, pp. 5–23, 1997.

[17] I. Norros, "On the use of fractional Brownian motion in the theory of connectionless networks," *IEEE Journal of Selected Areas in Communications*, vol. 13, no. 6, pp. 953–962, 1995.

[18] F. Kelly, "Notes on effective bandwidths," in *Stochastic Networks: Theory and Applications, Oxford University Press*, 1996, pp. 141–168.

[19] W. Willinger, M.S. Taqqu, R. Sherman, and D.V. Wilson, "Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level," *IEEE/ACM Transactions on Networking*, vol. 5, no. 1, pp. 71–86, 1997.

[20] P. Barford and M. Crovella, "Generating representative web workloads for network and server performance evaluation," in *Measurement and Modeling of Computer Systems*, 1998, pp. 151–160.

[21] K. Park, "On the effect and control of self-similar network traffic: A simulation perspective," in *Winter Simulation Conference*, 1997, pp. 989–996.

[22] A.M. Makowski, "Bounding on-off sources - variability ordering and majorization to the rescue," Tech. Rep. ISR TR2001-13, ISR Univ. of Maryland, 2001.

[23] S. Resnick, "Heavy tail modeling and teletraffic data," in *Annals of Statistics 25:1805-1869*, 1997.

[24] D. Heath, S. Resnick, and G. Samorodnitsky, "Patterns of buffer overflow in a class of queues with long memory in the input stream," in *Annals of Applied Probability.*, 1997.

[25] R. Riedi, "An improved multifractal formalism and self-similar measures," *Math. Anal. Appl.*, vol. 189, pp. 462–490, 1995.

[26] V.J. Ribeiro, R.H. Riedi, M.S. Crouse, and R.G. Baraniuk, "Simulation of nonGaussian long-range-dependent traffic using wavelets," in *Measurement and Modeling of Computer Systems*, 1999, pp. 1–12.

[27] A.C. Gilbert, W. Willinger, and A. Feldman, "Scaling analysis of conservative cascades with application to network draffic," *IEEE Transactions on Information Theory*, vol. 45, no. 3, pp. 971–991, 1999.

[28] K. Daoudi, A.B. Frakt, and A.S. Willsky, "Multiscale autoregressive models and wavelets," *IEEE Transactions on Information Theory*, vol. 45, no. 3, pp. 828–845, Apr. 1999.

[29] I. Norros, "A storage model with self-similar input," *Queueing Systems*, vol. 16, pp. 387–396, 1994.

[30] N. Duffield and N. Connell, "Large deviations and overflow probabilities for the general single server queue, with applications," in *Proceedings of Cam. Phil. Soc. available as: ftp:stp01.stp.dias.ie/DAPG/dapg9330.ps.*, 1994.

[31] N.G. Duffield and S.H. Low, "The cost of quality in networks of aggregate traffic," in *Proceedings of INFOCOM 98*, 1998, pp. 525–532.

[32] M.W. Garrett and W. Willinger, "Analysis, modeling and generation of self-similar VBR video traffic," in *Proceedings of SIGUOMM*, 1994, pp. 269–280.

[33] V. Almeida, A. Bestavros, M. Crovella, and A. Oliveira, "Characterizing reference locality in the WWW," in *Proceedings of the IEEE Conference on Parallel and Distributed Information Systems (PDIS)*, Miami Beach, FL, 1996.

[34] O.J. Boxma and V. Dumas, "Fluid queues with long-tailed activity period distributions," in *196*, p. 42. Centrum voor Wiskunde en Informatica (CWI), ISSN 1386-3711, 1997.

[35] P. Jelenkovic and A. Lazar, "Asymptotic results for multiplexing subexponential on-off sources," *Advances in Applied Probability*, vol. 31, no. 2, pp. 394–421, 1999.

[36] R. Agrawal, A.M. Makowski, and P. Nain, "On a reduced load equivalence for fluid queues under subexponentiality," *Queueing Systems*, vol. 33, no. 1-3, pp. 5–41, 1999.

[37] B. Braden et al, "Recommendations on queue management and congestion avoidance in the internet," *RFC 2309*, p. 16, 1998.

[38] S. Athuraliya, S.H. Low, V. Li, and Q. Yin, "REM: Active queue management," *IEEE Network*, vol. 15, no. 3, pp. 48–53, 2001.

[39] M. Christiansen, K. Jaffay, D. Ott, and F.D. Smith, "Tuning RED for web traffic," in *Proceedings of SIGCOMM*, 2000, pp. 139–150.

[40] W. Feng, D. Kandlur, D. Saha, and K. G. Shin, "BLUE: A new class of active queue management algorithms," Tech. Rep. U. Michigan EECS CSE-TR-387-99, 1999.

[41] S. Floyd and V. Jacobson, "Random etrly detection gateways for congection avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–493, 1993.

[42] C. Hollot, V. Misra, D. Towsley, and W. Gong, "A control theoretic analysis of RED," Tech. Rep. TR 00-41, CMPSCI Univ. of Massachusett, July 2000.

[43] V. Firoiu and M. Borden, "A study of active queue management for congestion control," in *Proceedings of INFOCOM*, 2000, pp. 1435–1444.

[44] P. Tinnakornsrisuphap and A.M. Makowski, "Queue dynamics of RED gateways under large number of TCP flows," Tech. Rep. ISR TR2001-14, ISR Univ. of Maryland, 2001.

[45] O. Ndiaye, "An efficient implementation of a hierarchical weighted fair queue packet scheduler," Tech. Rep. MIT/LCS/TM-509, 1994.

[46] Rong Pan, Balaji Prabhakar, and Konstantios Psounis, "CHOKe - A stateless queue management scheme for approximating fair bandwidth allocation," Tech. Rep. CSL-TR-99-787, 1999.

[47] R.R. Coifman, "Wavelet analysis and signal processing," in *Signal Processing, Part I: Signal Processing Theory*, Louis Auslander, Tom Kailath, and Sanjoy K. Mitter, Eds., pp. 59–68. Springer-Verlag, New York, NY, 1990.

[48] A. Antoniadis, "Wavelets in statistics: a review," in *J. It. Statist. Soc.*, 1999.

[49] P. Abry, L. Delbeke, and P. Flandrin, "Wavelet-based estimator for the self-similarity parameter of α stable process," *ICASSP*, 1999.

[50] R. Riedi, M.S. Crouse, V.J. Ribeiro, and R.G. Baraniuk, "A multifractal wavelet model with application to network traffic," *IEEE Transactions on Information Theory*, vol. 45, no. 3, pp. 992–1018, Apr. 1999.

[51] P. Abry and D. Veitch, "Wavelet analysis of long-range-dependent traffic," *IEEE Transactions on Information Theory*, vol. 44, no. 1, pp. 2–15, 1998.

[52] Y. Wang, "Function estimation via wavelet shrinkage for long-memory data," *Annals of Statistics*, vol. 24, no. 2, pp. 466–484, 1996.

[53] I.M. Johnstone and B.W. Silverman, "Wavelet threshold estimators for data with correlated noise," *Journal of the Royal Statistical Society B*, vol. 59, no. 2, pp. 319–351, 1997.

[54] I.M. Johnstone, "Wavelet shrinkage for correlated data and inverse problems: Adaptivity results," *Statistica Sinica*, vol. 9, no. 1, pp. 51–83, 1999.

[55] P. Abry, D. Veitch, and P. Flandrin, "Long range dependence: Revisiting aggregation with wavelets," *Journal of Time Series Analysis*, vol. 19, no. 3, pp. 253–266, 1998.

[56] E. Hernandez and G. Weiss, *A First Course on Wavelets*, CRC Press, 1996.

[57] I. Daubechies, *Ten Lecture on Wavelets*, SIAM, Philadelphia, 1992.

[58] R. M. Rao and A. D Bopardikar, *Wavelet Transforms: Introduction to Theory and Applications*, Addison Wesley, 1998.

[59] P. Maab A. K. Louis and A. Rieder, *Wavelets Theory and Applications*, John Wiley & Sons, 1997.

[60] S.G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE/ACM Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, July 1989.

[61] P.P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice-Hall, 1993.

[62] W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Wilson, "On the self-similar nature of ethernet traffic," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1–15, Feb. 1994.

[63] V. Paxson and S. Floyd, "Wide-area traffic: The failure of poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 226–244, June 1995.

[64] J. Beran, R. Sherman, W. Willinger, and M.S. Taqqu, "Variable-bit-rate video traffic and long-range dependence," *IEEE Transactions on Communications*, vol. 43, no. 2, pp. 1566–1579, 1995.

[65] G. Samorodnitsky and M.S. Taqqu, *Stable-Non-Gaussian Random Processes: Stochastic Models with Infinite Variance*, Chapman and Hall, 1994.

[66] K. Park and W. Willinger, *Self-Similar Network Traffic and Performance Evaluation*, John Wiley & Sons, 2000.

[67] M. Taqqu and V. Teverovsky, "On estimating the intensity of long-range dependence in finite and infinite variance time series," 1996.

[68] D. Veitch and P. Abry, "A wavelet based joint estimator for the parameters of long-range dependence," *IEEE Transactions on Information Theory*, vol. 45, no. 3, pp. 878–897, 1999.

[69] D. Veitch and P. Abry, "A statistical test for the time constancy of scaling exponents," *IEEE Transactions on Signal Processing*, vol. 49, no. 10, pp. 2325–2334, Oct. 2001.

[70] P. Flandrin, "Wavelet analysis and synthesis of fractional Brownian motion," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 910–917, Mar. 1992.

[71] S.M. Kay, *Fundamentals of Statistical Signal Processing*, Prentice-Hall, 1993.

[72] C.S. Chang, *Performance Guarantees in Communication Netowrks*, Springer-Verlag, 2000.

[73] R. Riedi and W. Willinger, "Toward an improved understanding of network traffic dynamics," *Self-Similar Network Traffic and Performance Evaluation*, pp. 507–530, 2000.

[74] S. Jaffard, "Local behavior of riemann's function," *Contemp. Math.*, vol. 189, pp. 287–307, 1995.

[75] Van Jacobson, "Congestion avoidance and control," in *Proceedings of SIGCOMM*, 1988, pp. 314–329.

[76] D.R. Cox, *Renewal Theory*, Methuen & Co. LTD Science Paperback edition, 1967.

[77] R. Riedi, "Multifractal processes," Tech. Rep. TR99-06, ECE Dept. Rice University, 1999.

[78] V. Ribeiro, R. Riedi, M. Crouse, and R. Baraniuk, "Multiscale queuing analysis of long-range dependence network traffic," *Submitted to IEEE Transactions on Networking*.

[79] M. Greiner, M. Jobmann, and L. Lipsky, ," *Operations Research*, vol. 47, no. 2, pp. 313–326, 1999.

[80] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in C*, Cambridge University Press, 1995.

[81] W.T. Eadie, *Statistical Methods in Experimental Physics*, North Holland, 1971.

[82] D'Agostino and Stephens, *Goodness-Of-Fit Techniques*, Marcel-Dekker, New York, 1986.

[83] C. Fulton and S. Li, "Delay jitter first-order and second-order statistical functions of general traffic on high-speed multimedia networks," *IEEE/ACM Transactions on Networking*, vol. 6, no. 2, pp. 150–163, Apr. 1998.

[84] Randall Landry, "End-to-end delay and jitter analysis in atm networks," .

[85] W. Matragi, C. Bisdikian, and K. Sohraby, "Jitter calculus in ATM networks: single node case," in *Proceedings of INFOCOM*, 1994, pp. 232–241.

[86] W. Matragi, K. Sohraby, and C. Bisdikian, "Jitter calculus in ATM networks: multiple node case," in *Proceedings of INFOCOM*, 1994, pp. 242–251.

[87] S. Floyd, R. Gummadi, and S. Shenker, "Adaptive RED: An algorithm for increasing the robustness of RED," available at http://www.icir.org/floyd/papers/adaptiveRed.pdf, Aug. 2001.

[88] W. Feng, D.D. Kandlur, D. Saha, and K.G. Shin, "A self-configuring RED gateway," in *Proceedings of INFOCOM 99*, 1999, pp. 1320–1328.

[89] W. R. Stevens, *TCP/IP Illustrated (Volume 1)*, Addison Wesley, 1994.

[90] S.H. Low, F. Paganini, J. Wang, S. Adlakha, and J.C. Doyle, "Dynamics of TCP/RED and a scalable control," in *Proceedings of INFOCOM*, 2002, pp. 239–248.

[91] V. Misra, W. Gong, and D. F. Towsley, "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," in *Proceedings of SIGCOMM*, 2000, pp. 151–160.

[92] P. Ranjan, E. Abed, and R. La, "Nonlinear instabilities in TCP-RED," in *Proceedings of INFOCOM*, 2002, pp. 249 –258.

[93] T.J. Ott, T.V. Lakshman, and L.H. Wong, "SRED: Stabilized RED," in *Proceedings of INFOCOM 99*, 1999, pp. 1346–1355.

[94] D. Lin and R. Morris, "Dynamics of random early detection," in *Proceedings of SIGCOMM*, 1997, pp. 127–137.

[95] D. Clark and W. Feng, "Explicit allocation of best-effort packet delivery service," *IEEE/ACM Transactions on Networking*, vol. 6, no. 4, pp. 362–373, Aug. 1998.

[96] T. Ott, J. Kemperman, and M. Mathis, "The stationary behavior of ideal TCP congestion avoidance," http://citeseer.nj.nec.com/ott96stationary.html.

[97] M. Mathis, J. Semke, and Mahdavi, "The macroscopic behavior of the TCP congestion avoidance algorithm," *Computer Communication Review*, vol. 27, no. 3, pp. 953–962, 1997.

[98] D. R. Smart, *Fixed Point Theorems*, Combridge University Press, London, New York, 1974.

[99] A. Lasota and M.C. Mackey, *Chaos, Fractals, and Noise: Stochastic Aspects of Dynamics*, Springer-Verlag, second edition, 1994.