QUEUE LENGTH ANALYSIS OF END-TO-END DIFFERENTIATED SERVICE NETWORKS WITH SELF-SIMILAR TRAFFIC

By

YUE WANG

Bachelor of Science

Beijing United University, Art and Science College

Beijing , China

1997

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
July, 2005

QUEUE LENGTH ANALYSIS OF END-TO-END DIFFERENTIATED SERVICE NETWORKS WITH SELF-SIMILAR TRAFFIC

Thesis Approved:

Dr. G. E. Hedrick

Thesis Adviser

Dr. Jong-Moon Chung

Dr. Nohpill Park

Dr. A. Gordon Emslie

Dean of the Graduate College

ACKNOWLEDGEMENTS

I would like to express my sincerely appreciation to Prof. Dr. G. E. Hedrick and Prof. Dr. Jong-Moon Chung, for their guidance, valuable discussions, advice and kindness in completing my thesis and master of science programs.

I would like to express my sincerely appreciation to Prof. Dr. Nohpill Park for his valuable suggestions and comments.

I would like to thank for my family for their supports and understanding during the period I dedicated to achieving this milestone in my life and career.

Last but not the least. I am thankful to all friends and colleagues who made my stay at Oklahoma State University a memorable experience.

TABLE OF CONTENTS

LIST OF TABLES

TABLE PAGE

Table 2-1 Estimated *H* for trace file 1 34

LIST OF FIGURES

FIGURE PAGE

CHAPTER I

INTRODUCTION

Modern computer network research focuses on the issues of performance evaluation and performance optimization. This includes enhancing the performance of an operational network at both the traffic and the resource level. While we are trying to utilize network resources economically and reliably, traffic oriented performance requirements should be satisfied. The traffic oriented performance includes delay, delay variation, packet loss, and throughput.

Recently, many high speed networks were discovered to be self-similar. Self-similarity is a characteristic if the object looks “roughly” the same regardless of scale. It is a powerful mathematics representation of a variety of physical phenomena. They have the long range dependence property.

This thesis focuses on the analysis of queue length boundaries in end-to-end self-similar networks with differentiated service. End-to-end means constant administrations and technical characteristic along the entire path from source to destination. Packets travel from source to destination by basic hop-by-hop forwarding strategy of the Internet. Study shows that Ethernet, TCP, FTP, TELNET and World Wide Web traffic are self-similar.

It is well known that traditional analytical methods of queuing systems are based on Poisson and Poisson-based stochastic processes. Unfortunately, these methods, like Jackson theorem, are not applicable in high speed networks.

In this thesis, a novel analytical model is proposed based on the arrival rate and the service rate for single hop queuing systems. Then the derivations are extended to end-to-end differentiated service networks with self-similar traffic. The upper and lower bound of queue length at each hop is derived. In addition, the analytical model is also applied to the analysis of the traffic flow effects on queue length. The results illustrate the performance gain in queue length at each hop. After the application of *Little's* theorem to queue length, the mean delay at each hop can be obtained. These parameters, queue length and mean delay, are parameters related to the concept of traffic engineering. The rest of this thesis is organized as follows: Chapter II introduces self-similar traffic characteristics in high speed networks. Chapter III discusses Quality of Services (QoS) and queue length analysis in non-preemptive differentiated services networks. Chapter IV is based on the Core-Stateless Fair Queuing (CSFQ) fluid algorithm model, the queuing length analysis extended to multi-hops in high-speed networks. The connection of each hop is connection oriented. Chapter V contains the simulation procedures and results analysis, and Chapter VI concludes this thesis and points out certain directions for future research.

CHAPTER II

SELF-SIMILAR NETWORK

2.1 Introduction

"Self-similarity" describes a phenomenon whose behaviors are same whatever we view either at different scales on space or time dimension. In other words, a self-similar pattern reproduces itself at different scales. It is an important concept that has been applied to data communications traffic analysis as soon as it was observed in early 1990s. The self-similar stochastic process was introduced by Kolmogorov in a theoretical context and brought to the attention of Benoit Mandelbrot. Then Benoit Mandelbrot first presented the mathematical study of self-similar shapes and their relationship to natural shapes. The term "self-similar" was formally defined and applied in [2] and [14]. The discovery of self-similarity has launched a new examination of data traffic performance. In paper [33], the authors have provided a solid proof of self-similar traffic and included a number of useful self-similar traffic generation models. These traffic models are widely used as the input to analysis or simulation.

High speed multiplexed network data traffic can be viewed as a time series of a self-similar stochastic process. As illustrated by Figure 2-1, a sequence of simple plots of the packet counts (i.e., number of packets per time unit) for 4 different time scales are presented. The time unit of (a) is 10 seconds; each subsequent plot is obtained from the previous one by increasing the time resolution by a factor of 10. The time unit

corresponding to (d) is 0.01 seconds. Plots (a)-(d) look very "similar" to one another (in a distributional sense) [14]. The "self-similarity" is related to fractals and chaos theory [29]. Fractals are the images formed by recursively replacing the parts of an image with the entire structural template. Thus, self-similarity is created by fractal. Chaos theory states that a small segment of the fractal is just as detailed as the entire one. It implies that the same principles and patterns hold true for any scale of natural self-similar phenomenon, no mater how large or small [38]. Notice the scaling property (y-axis) and the absence of a natural length of a "burst": at every time scale ranging from 0.01 seconds to 10 seconds, bursts consists of bursty sub-periods separated by less bursty sub-periods. This scale invariant or "self-similar" feature of Ethernet traffic is different from both the traditional telephone traffic and from stochastic models for packet traffic. This "proof" of self-similar nature of Ethernet packet traffic suggests that Ethernet traffic on one time scale is statistically identical (respect to its second-order statistical properties which observing the mutual relationship of two points by autocorrelation function, power spectral density function and correlation function.) to Ethernet traffic on a different time scale and, thus, motivates the use of self-similar stochastic processes for traffic modeling purposes [14].

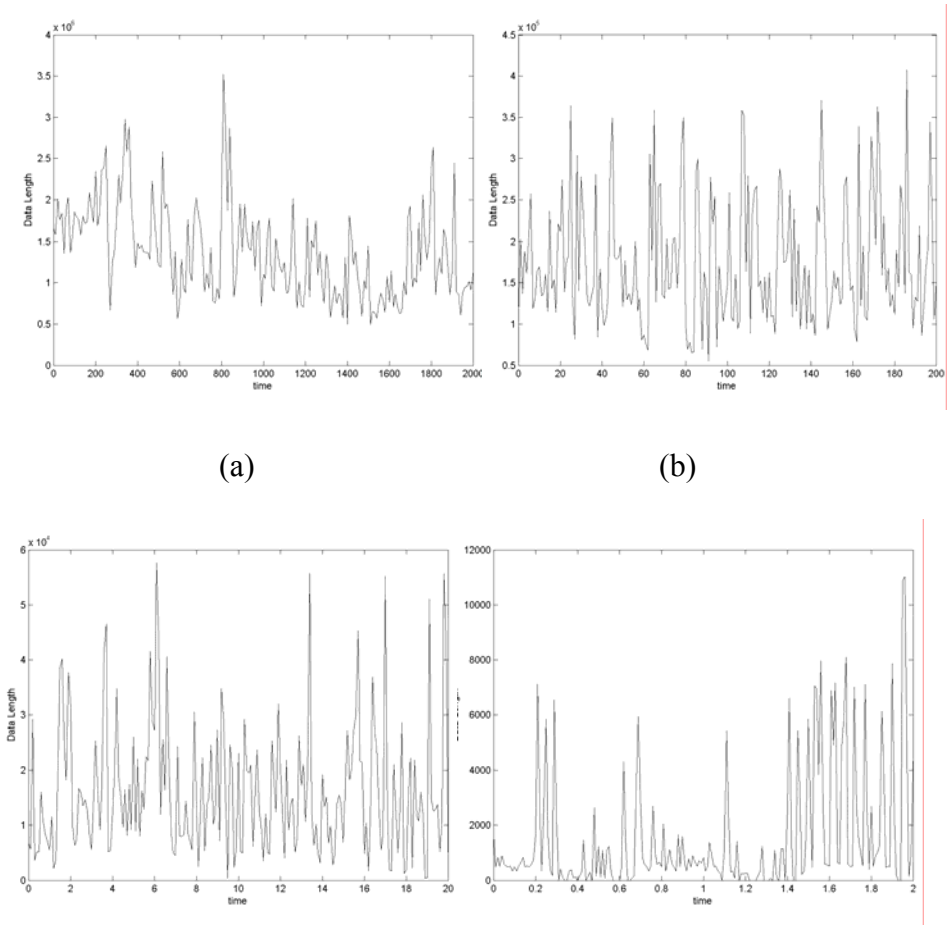


Figure 2-1. Self-similar stochastic processes.

Since self-similarity is believed to have a significant impact on network performance, understanding the causes of self-similarity is important. It has been revealed that the traffic generated and transferred over the World Wide Web (WWW) shows self-similar characteristics. One practical effect of self-similarity is that the buffers needed at switches and multiplexers must be larger than those predicted by traditional queuing analysis and simulations. These larger buffers create greater delay in individual streams than originally anticipated. An exponential trade-off relationship was observed between queuing delay and packet loss rate. Packet loss and retransmission rate decline smoothly

as the self-similarity is increased under reliable flow-controlled packet transport. One important discovery is that the higher the load on the Ethernet, the higher the degree of self-similarity [28].

2.2 Mathematical Definitions of Self-similarity

Continuous-Time Definition: If a stochastic process $x(t)$ is statistically self-similar with parameter H ($0.5 \le H \le 1$), it will satisfy the following three conditions [29]:

$$\text{Mean: } E[x(t)] = \frac{E[x(at)]}{a^H} \tag{2.1}$$

$$\text{Variance: } Var[x(t)] = \frac{Var[x(at)]}{a^{2H}} \tag{2.2}$$

$$\text{Autocorrelation: } R_x(t,s) = \frac{R_x(at,as)}{a^{2H}} \tag{2.3}$$

For a self-similar process, the process $a^{-H}x(at)$ has the same statistical properties (mean, variance, autocorrelation) as $x(t)$ for any real $a > 0$. H is the Hurst parameter, which is a key measure of self-similarity [29].

Discrete-Time Definition: For an original stationary time series $X = \{X_t, t = 0,1,2,...\}$, its m-aggregated time series is $x^{(m)} = \{x_k^{(m)}, x_k^{(m)} = \frac{1}{m}(\sum_{i=km-(m-1)}^{km} x_i), k = 0,1,2,...\}$, where m represents the non-overlapping adjacent block size, and k represents the time in the new series. For example [29],

$$x_k^{(3)} = \frac{x_{3k-2} + x_{3k-1} + x_{3k}}{3} \tag{2.4}$$

The original stationary time series can be considered as $x^{(1)}$ which is the highest resolution the time series has. The process $x^{(m)}$ is the same process reduced in resolution by a factor of m.

Definition of an exactly self-similar process [29]: $\forall m = 1,2,3,...$

$$\text{Variance:} \qquad Var(x^{(m)}) = \frac{Var(x)}{m^{\beta}} \tag{2.5}$$

$$\text{Autocorrelation:} \quad R_{x(m)}(k) = R_x(k) \tag{2.6}$$

$$\text{Hurst Parameter:} \quad H = 1 - (\beta / 2) \tag{2.7}$$

In other words, X is called exactly (second-order) self-similar if the aggregated processes $x^{(m)}$ are indistinguishable from X—at least with respect to their second order statistical properties. An example of an exactly self-similar process parameter H is Fractional Gaussian Noise (FGN) with parameter $0.5 \le H \le 1$ [14].

Definition of asymptotically self-similarity: A stochastic process x is said to be asymptotically self-similar if for all large enough time domain k, it satisfies the two conditions below [29]:

$$\text{Variance:} \qquad Var(x^{(m)}) \approx \frac{Var(x)}{m^{\beta}} \tag{2.8}$$

$$\text{Autocorrelation:} \qquad R_{x(m)}(k) \to R_x(k) \ \text{ as } m \to \infty \tag{2.9}$$

The asymptotic self-similarity describes a process that after m-aggregated, its autocorrelation does not change much when m increases, its sample variance is basically a fixed multiple of its original series variance.

2.3 Self-similarity Properties

For an exactly self-similar process, the statistics of the process (mean, variance, correlation, etc) are equal to its original time series for all block size m. However for non-self-similar processes, they do not have this characteristic. For an exactly self-similar process, after any m-aggregation, the new aggregated time domain series versus its original series will show autocorrelations and a linear relationship among their variances.

Actually in the real world, for self-similar processes, the variance of the time average cannot guarantee any decay, but they commonly decay very slowly. So there is a weaker condition called asymptotically self-similar that is defined to represent these cases.

The most striking feature of both exactly and asymptotically self-similar processes is that their aggregated processes $x^{(m)}$ possess a non-degenerate correlation structure as $m \to \infty$. This behavior is illustrated with the plots in Figure 2-1. If the original time series X represents the number of Ethernet packets per 0.01 seconds (plot (d)), then plots (a) to (c) represent the aggregated time series $X^{(100)}$, $X^{(10)}$, and $X^{(1)}$, respectively. All of the plots look "similar", suggesting a nearly identical autocorrelation function for all of the aggregated processes [14].

2.4 Long Range Dependence

The concept of long range dependence is associated with the auto covariance properties. It defines the relationship between a stationary stochastic process

autocorrelation $C(t)$ with time t increments. For many processes, such as the Poisson increment process, their autocorrelation decrease exponentially or faster [29].

$$C(t) \sim \frac{1}{a^{|t|}} \qquad \text{as} \mid t \mid \to \infty,\ 0 < a < 1 \tag{2.10}$$

They are called short-range dependent.

On the contrary, a long-range dependent process's autocovarance decays more slowly than a short-rang dependence process [29].

$$C(t) \sim \frac{1}{|t|^{\beta}} \qquad 0 < \beta < 1 \tag{2.11}$$

So,
$$\sum_{t=0}^{\infty} C(t) = \infty \tag{2.12}$$

Note that self-similar process is not necessary long-range dependent, and vise versa. For a self-similar process is not necessary a long-rang dependent process, and on the other hand, a long-range dependent process is not necessarily self-similar. However, in the network communication area, these two concepts are interchangeable. $H = 0.5$ implies the process is not long-range dependence. $H = 1$ indicates the process is exactly self-similarity.

2.5 Approaches to Estimate the Hurst Parameter

The Hurst parameter plays an important role in self-similarity. Historically, self-similar processes provide an elegant explanation and interpretation of an empirical law that is commonly referred to as the Hurst's law or the Hurst effect. Briefly speaking, for a given set of random variable $X = \{X_n, n = 0,1,2,...\}$ with sample mean $\bar{x}(n)$ and sample variance $S^2(n)$, the rescaled adjusted range (or the R/S statistic) is given by

$R(n)/S(n) = 1/S(n)[\max(0, W_1, W_2, ..., W_n) - \min(0, W_1, W_2, ..., W_n)]$, where each W term $W_k = (X_1 + X_2 + ... + X_k) - k\overline{X}(n)$, $k = 1,2,...,n.$ Hurst (1955) found that many real world processes appear to be well represented by the relation of $E[R(n)/S(n)] \sim cn^H$, as $n \to \infty$. The variable n is the block size, R is the maximum subtracted minimum value for the block; and S is the standard deviation of the block.

In the real world, the Hurst parameter H "typically" ranges from 0.7 to 0.9. The larger values of H suggest a higher degree of persistent variability in the data. Exactly self-similar process have a Hurst parameter of 1. Network traffic can have a range of H value between 0.5 and 1. On the other hand, if the observation X_k comes from a short-range dependent model, then $E[R(n)/S(n)] \sim cn^{0.5}$, as $n \to \infty$. This discrepancy is generally referred to as the Hurst effect or Hurst phenomenon [14] [29] [37].

There are many ways to estimate the Hurst Parameter H. Three common methods are briefly introduced below.

2.5.1 Variance-Time plot

Recall that for the aggregated time series $x^{(m)}$ of a self-similar process, the variance obeys the following for large m [29]:

$$Var(x^{(m)}) \approx \frac{Var(x)}{m^{\beta}} \tag{2.13}$$

Where the self-similarity parameter $H = 1 - \beta/2$. This can be rewritten as [29]:

$$\log[Var(x^{(m)}] \approx \log[Var(x)] - \beta \log m \tag{2.14}$$

Because $\log[Var(x)]$ is a constant independent of m, the result should be a straight line with a slope of $-\beta$ if we plot $\log[Var(x^{(m)})]$ versus $\log m$ on a graph. Slope values between -1 and 0 suggest self-similarity. Figure 2-2 shows the program flow chart of a variance time plot.

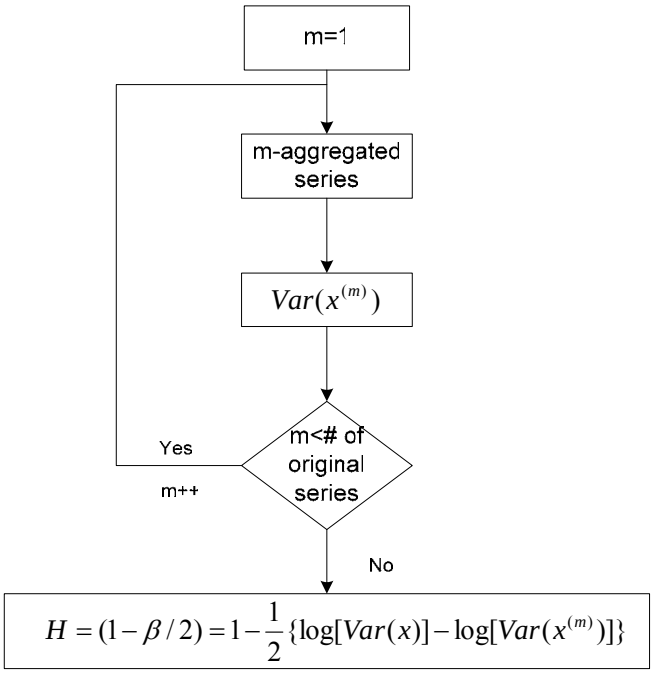


Figure 2-2. Program flow chart for variance time plot.

2.5.2 R/S Plot

This is one of the oldest methods. For details, see [10]. For a stochastic discrete time process $X_{(t)} = \{x_t, t = 0,1,2,....\}$, the rescaled range of $X_{(t)}$ over a time interval N is defined as the ratio R/S [29]:

$$\frac{R}{S} = \frac{\max\limits_{1\le j\le N}[\sum_{k=1}^{j}(X_k - M(N))] - \min\limits_{1\le j\le N}[\sum_{k=1}^{j}(X_k - M(N))]}{\sqrt{\frac{1}{N}\sum_{k=1}^{N}(X_k - M(N))^2}} \tag{2.15}$$

$$M(N) = \frac{1}{N}\sum_{j=1}^{N} X_j \tag{2.16}$$

where $M(N)$ is the sample mean over the time period N.

For a self-similar process, the ratio has the following characteristic for large N [29]:

$$R/S \sim (N/2)^H \quad \text{with} \quad H > 0.5 \tag{2.17}$$

This can be rewritten as [29]:

$$\log[R/S] \sim H\log(N) - H\log(2) \tag{2.18}$$

A log-log graph plot of [R/S] versus N shows the result fits a straight line with slope H. Therefore, Hurst parameter H equals the slope of the line on a log-log graph [29]. Figure 2-3 shows the program flow chart of the R/S plot.

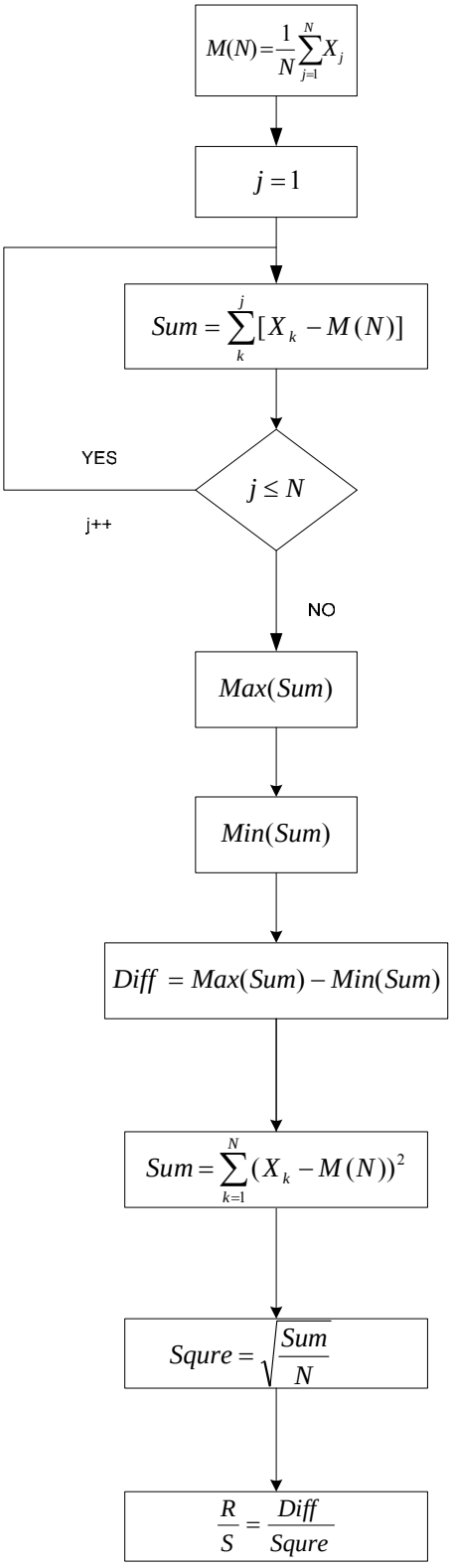


Figure 2-3. Program flow chart for *R/S* plot.

2.5.3 Wavelet plot

Since network performances are highly influenced by the level of burstiness of the input data, which is strongly related to the value of *H*, it becomes crucial to have a reliable tool for the estimation of the Hurst parameter. An innovative approach involves exploiting the scale-invariance prosperity of self-similar processes by means of a multi-resolution decomposition technique, namely the Wavelet Transform [12].

The wavelet plot uses the wavelet transform from the original signal to a *m*-level series. The Hurst parameter *H* is estimated by using the self-similar energy property.

Most signals are time-domain, i.e. measured signal is a function of time. In other words, when we plot the signal one of the axes is time (independent variable), and the other (dependent variable) is usually the amplitude". However, the frequency domain is more useful. That is, "the frequency spectrum of a signal shows what frequencies exist in the signal" [36]. The Fourier transform only applies to stationary stochastic process, because the frequencies are stable regardless of time. However, the non-stationary process keeps changing at their frequencies in the time domain. Therefore the wavelet transformation is used to keep all the information containing both time and frequency information. Because continous time wavelet transform is considerably difficulty and unrelated to our work, only the discrete time wavelet transform id discussed here. Figure 2-4 illustrates the basic wavelet transformation concept.

Assume the original series $x(t)$ (t = 0, 1, 2, …16) consists of 16 random variables. When this series passes through high pass wavelet filters and is sub-sampled by 2, it becomes a level 1 DWT (Discrete Wavelet Transform) series. The number of original series is reduced by half. At the same time, the original series passes through the low pass

wavelet filter and sub-sampled by 2, again reducing by half. This new series becomes the original series for the next high pass and low pass wavelet filters, giving rise to the level 2 DWT series. Then we do the same process to the level 2 product and finally get a level 4 DWT. In level 4,we have just one variable, whick indicates termination of wavelet transformation [36].

Therefore, the time domain resolution has decreased by 4 and the value resolution increased by 4 due to the DWT. The sum of all the level coefficients will give the original series coefficient.

Due to the wavelets ability to discriminate the signal energy at different frequency levels, it is easy to estimate the spectral behavior of the analysis trace, which in case of self-similar processes is of the form $\frac{1}{f^{\gamma}}$. Recall the equation (2.14), this form is related to Hurst parameter [12].

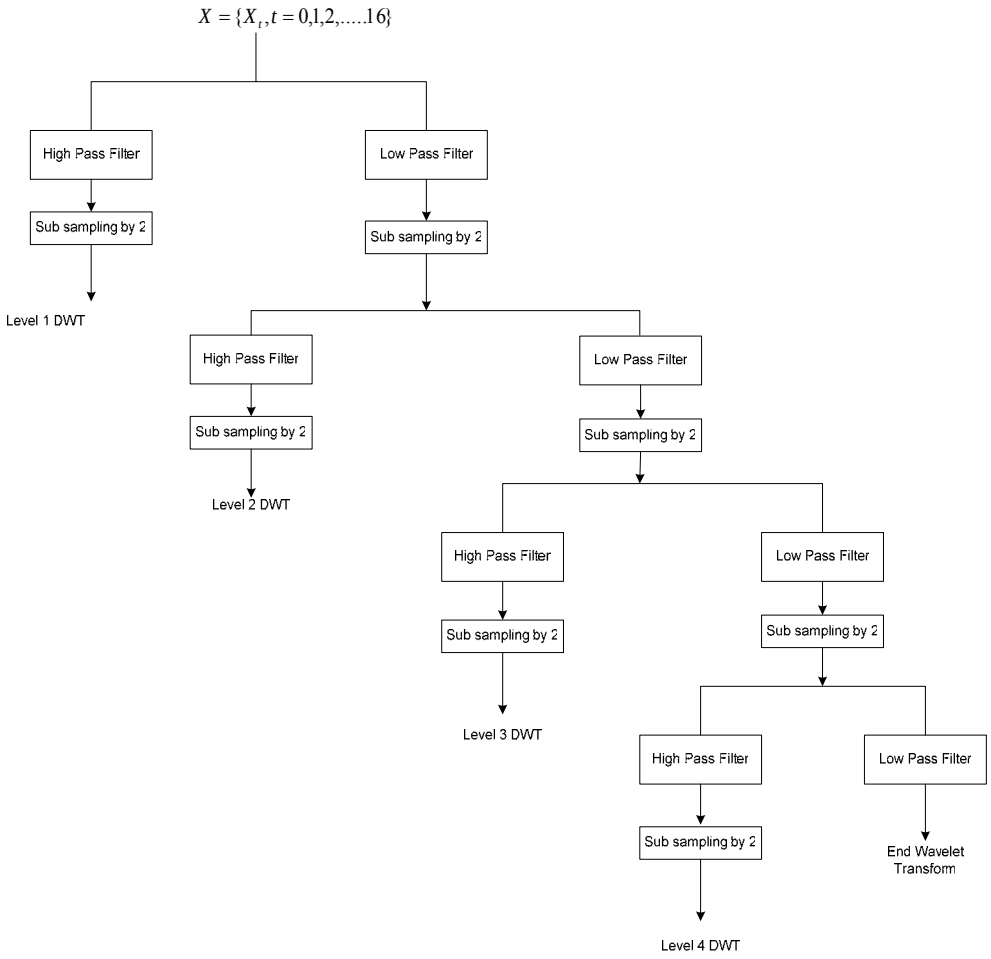


Figure 2-4. Wavelet transform.

The wavelet plot is based on the analysis of FGN traces generated by means of the well-known Random Midpoint Displacement (RMD) algorithm. The authors of paper [12] start from the assumption that the generation is asymptotically correct. In order to verify the relationship between the estimation efficiency and the length of the analyses sequence, they first tested the wavelet estimator with long traces and then cut them into sub-traces [12].

Since the Wavelet transform implies the decomposition of a signal at different resolutions and the mother wavelet itself is defined recursively, it appear as the natural

tool for the analysis of self-similar processes, which present the same statistical characteristics at every scale [12].

The wavelet plot for estimation of the Hurst parameter is derived directly from the following two statements [29], [32]:

(1) The wavelet decomposition discriminates the different frequency contributions: the wavelet coefficients at a given level *m* are associated to the mean amount of energy Γ_m of the analyzed signal around the frequency $v_0 a_0^{-m}$ where v_0 depends upon the choice of the mother wavelet and α_0=2 in order to obtain orthonormal decomposition.

(2) Self-similar processes are characterized by a $\frac{1}{f^{\gamma}}$ decay of the power spectral density and $\gamma = 2H - 1$ in the FGN case [12].

Therefore use (28) to evaluate Γ_m as the sample mean [32]:

$$\Gamma_m = \frac{x_{m1}{}^2 + x_{m2}{}^2 + x_{m3}{}^2 .... + x_{mn}{}^2}{n_m} \tag{2.19}$$

m is the resolution level of wavelet transform, n_m is the number of level *m* wavelet coefficients, and Γ_m is *m* 's energy at a given level. For a self-similar process, it satisfies [32]:

$$\Gamma_m = 2^{m\gamma}\Gamma_0 \tag{2.20}$$

Therefore [29], [32],

$$\log_2 \Gamma_m \sim m\gamma + \log_2 \Gamma_0 \tag{2.21}$$

$$H = \frac{1+\gamma}{2} \tag{2.22}$$

We use $\log_2 \Gamma_m$ versus *m* to obtain a straight line. Then the slope will be γ. Based on the relationship with *H* in (31), we can obtain an estimated Hurst Parameter *H* [12]. Figure 2-5 shows the program flow chart of the wavelet plot.

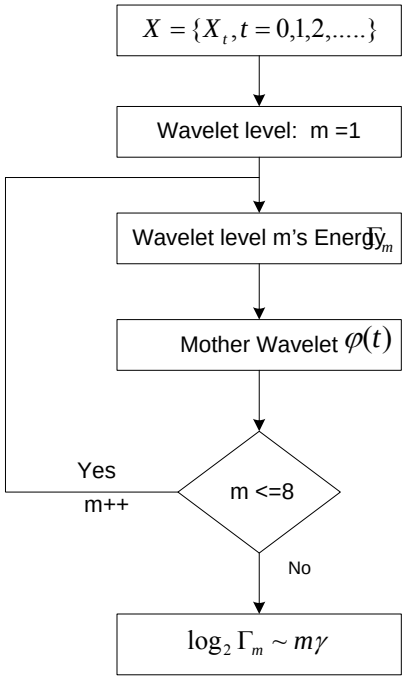


Figure 2-5. Program flow chart for wavelet plot.

2.7 Simulation Results and Analysis

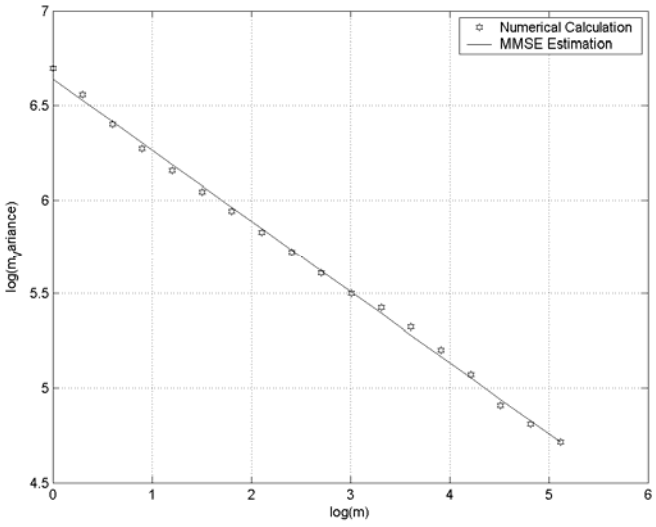
2.7.1 Trace files

Two trace files from [14] and [32] are used in the analysis. In the first trace file BC-pAug89.TL, there are a total of two columns. The left column represents time in seconds. The right column represents the new Ethernet data length in bytes that have arrived based on the time reference in the left column. Since the time interval in the original file is random, the first thing that needs to be done is to aggregate the traffic data to fixed time intervals. The aggregation levels are set to 100, 10, 1, 0.1, 0.01 seconds respectively. The resulting series are ready for analysis.

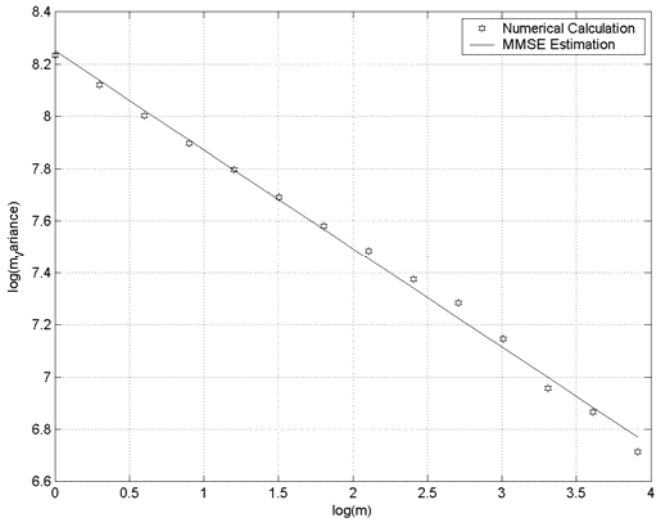
For the second trace file the high quality Star Wars IV trace file was used [32]. The file name is Terse_StarWarsIV.dat. This file has just one column. It represents the frame length at different time intervals. Actually, there are three frame types. They are I type, B type, and P type. For the high quality trace, they are equal [32]. Since the frame length in the original file is random, the first thing that needs to be done is to aggregate the traffic data to a fixed frame length per group. The aggregation level is set to be 800, 400,100, 50, 12 frames per group. The result series are ready for analysis.

2.7.2 Variance Time Plot

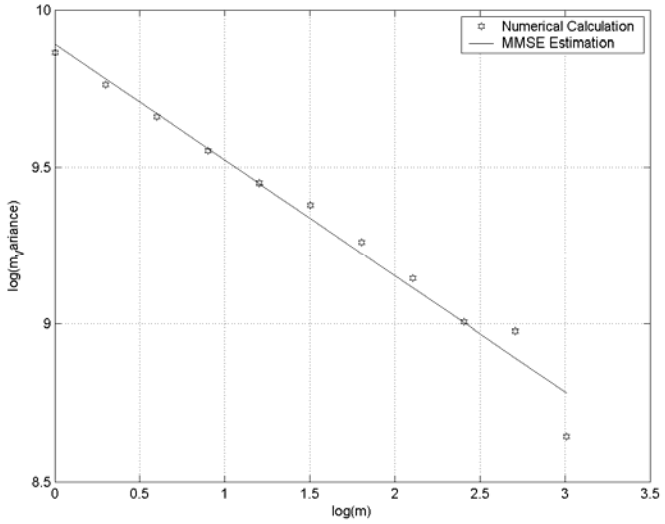
Figure 2-6 shows the simulation results of using trace file 1. Figure 2-7 shows the simulation results when using the trace file StarWarsIV.



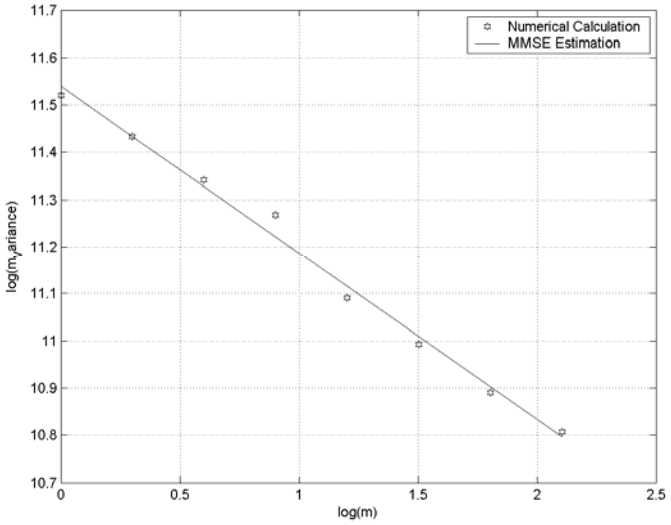
(a) Time interval =0.01 second.



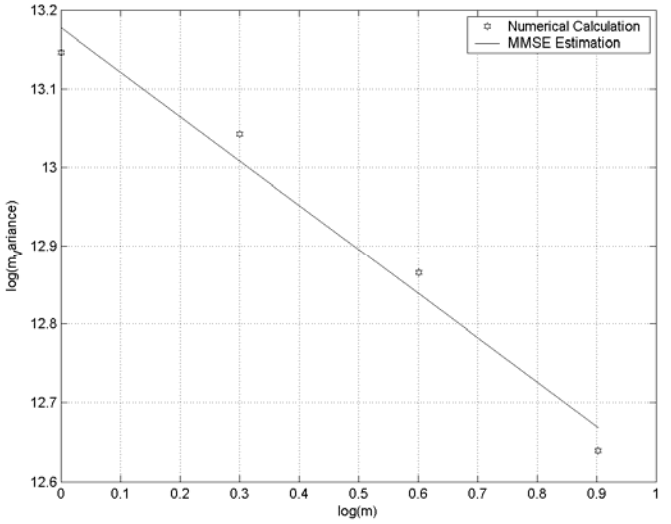
(b) Time interval = 0. 1 second.



(c) Time interval =1 second.

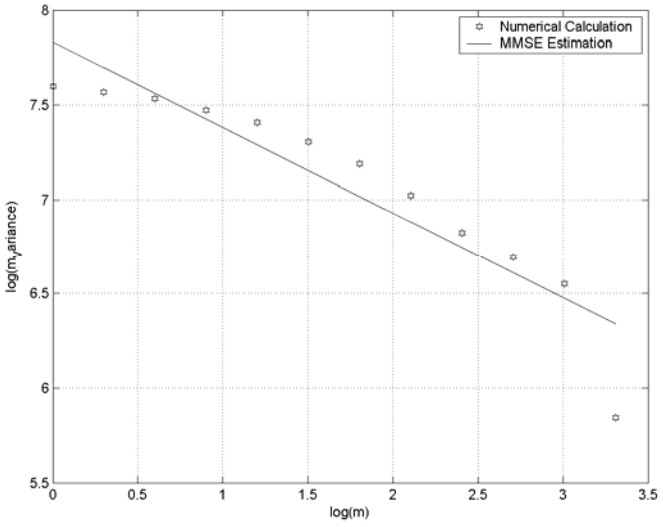


(d) Time interval =10 seconds.

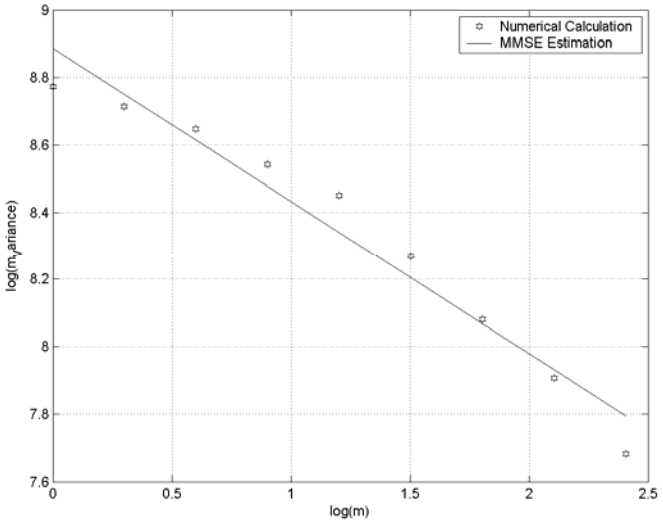


(e) Time interval =100 seconds.

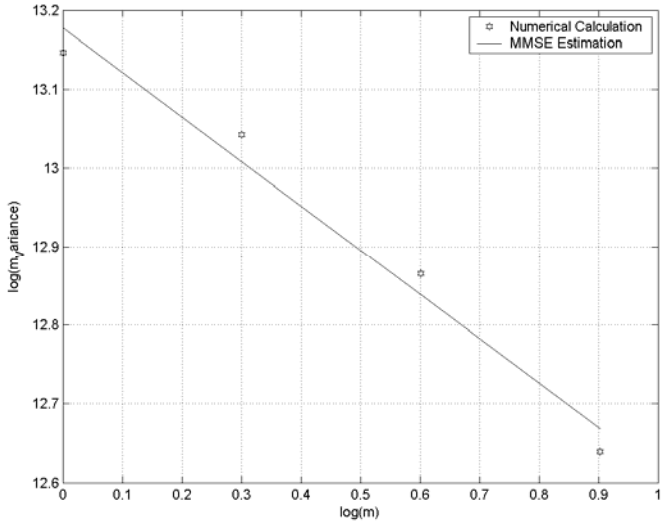
Figure 2-6. Variance time plot experimental results of trace file 1.



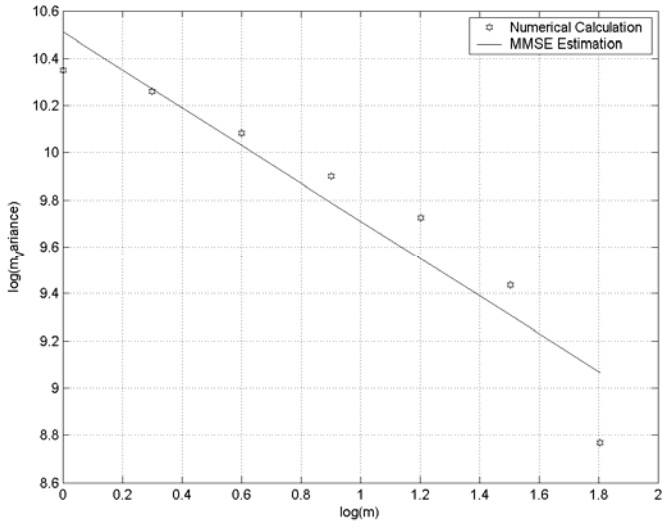
(a) 12 frames per group.



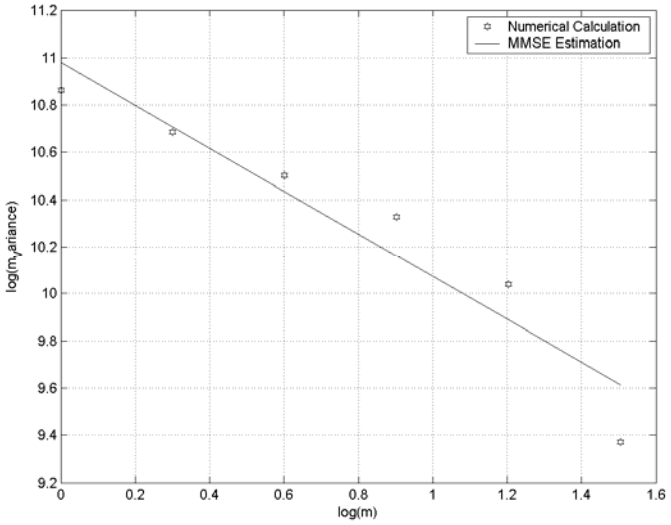
(b) 50 frames per group.



(c) 100 frames per group.



(d) 400 frames per group.

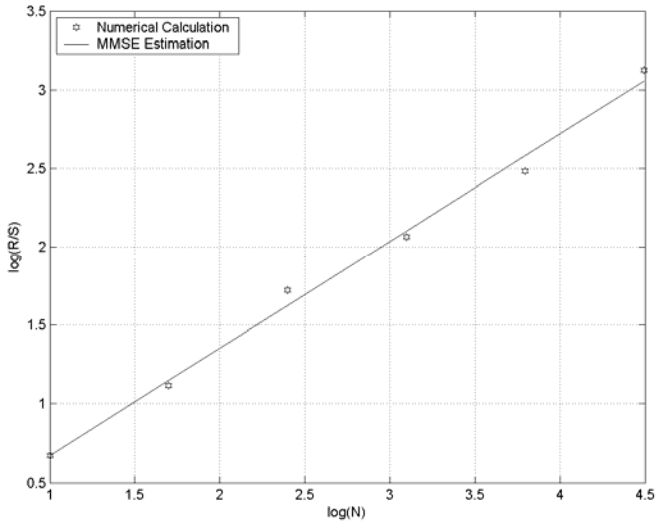


(e) 800 frames per group.

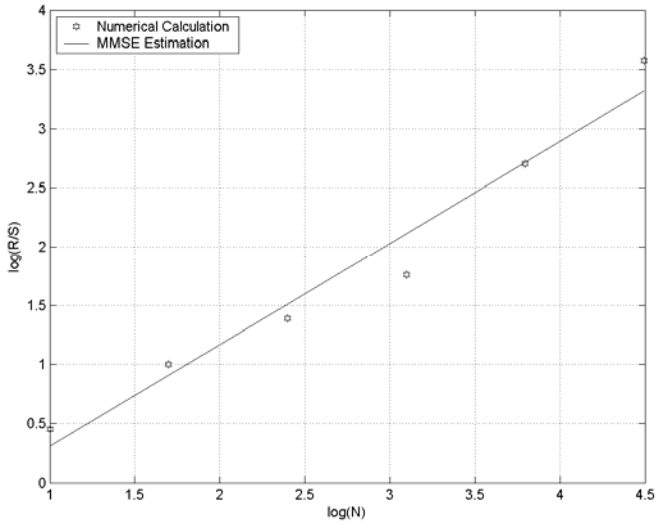
Figure 2-7. Variance time plot experimental results of trace file StarWarsIV.

2.7.3 R/S Plot

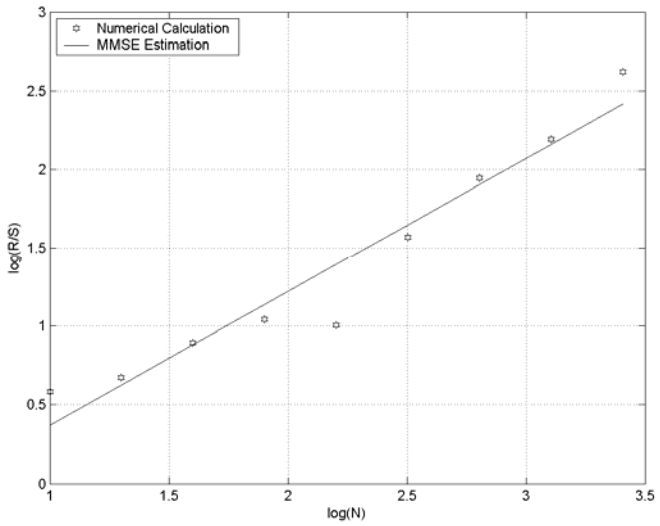
Figure 2-8 and 2-9 are *R/S* plot experimental results of the trace file 1 and StarWarsIV.



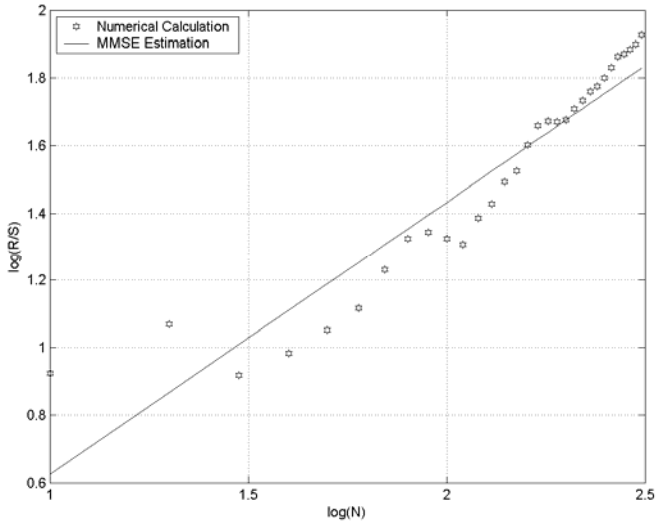
(a) Time interval =0.01 second.



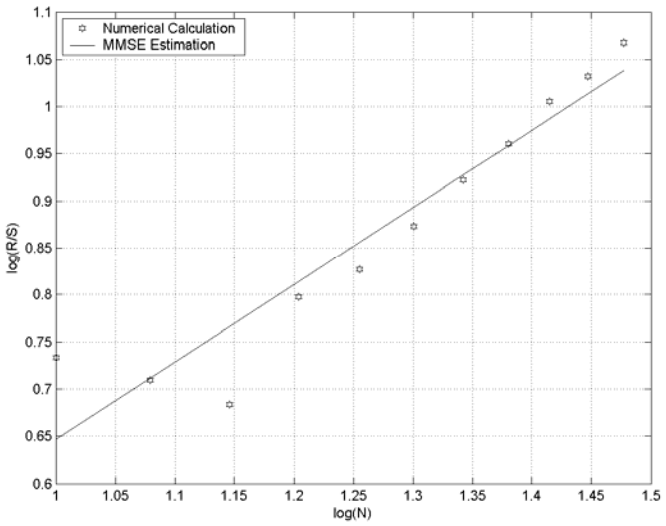
(b) Time interval =0. 1 second.



(c) Time interval =1 second.

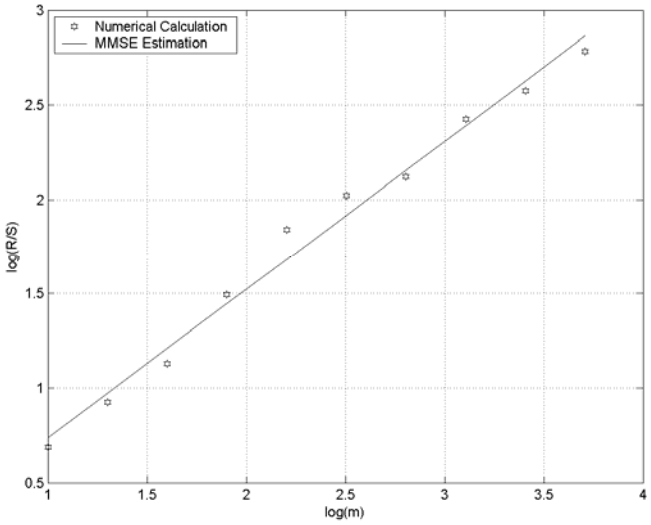


(d) Time interval =10 second.

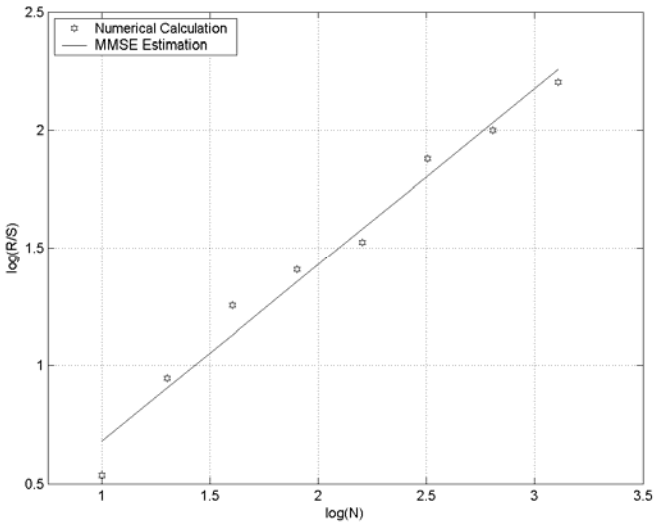


(e) Time interval =100 second.

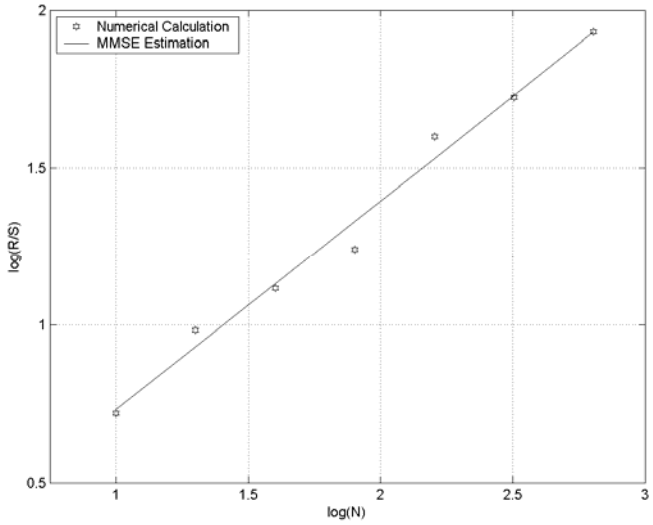
Figure 2-8. *R/S* plot experimental results of trace file 1.



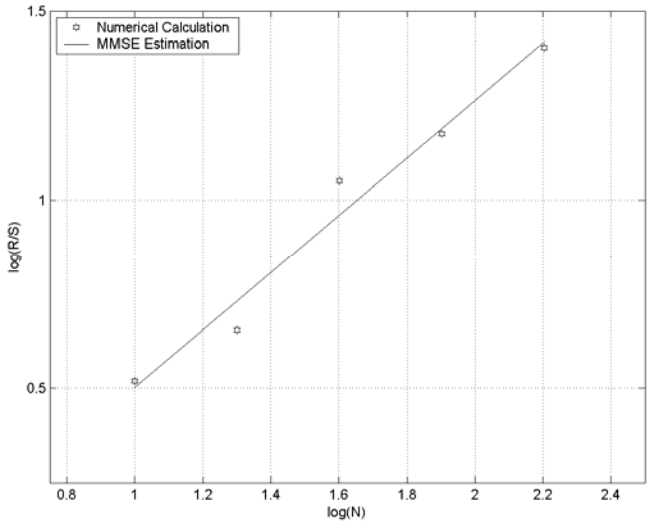
(a) 12 frames per group.



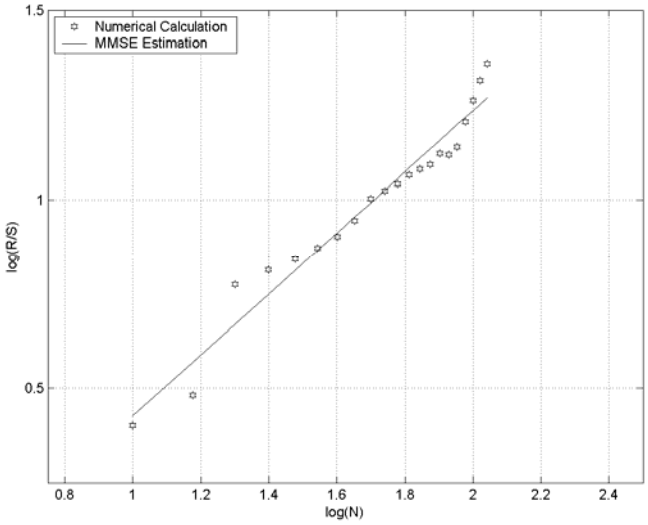
(b) 50 frames per group.



(c)100 frames per group.



(d) 400 frames per group.

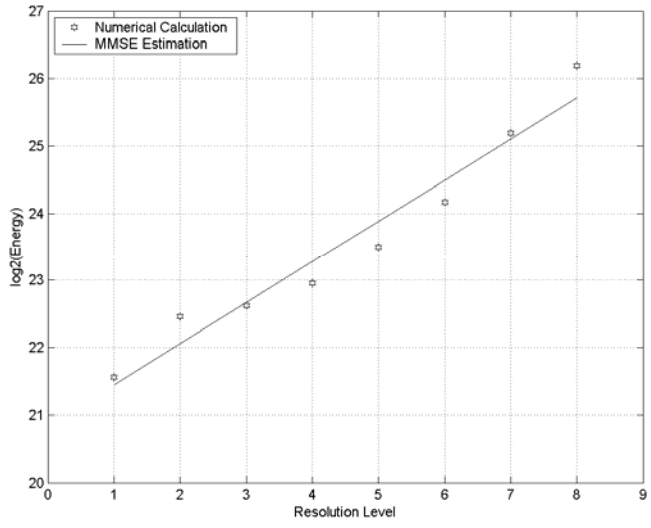


(e) 800 frames per group.

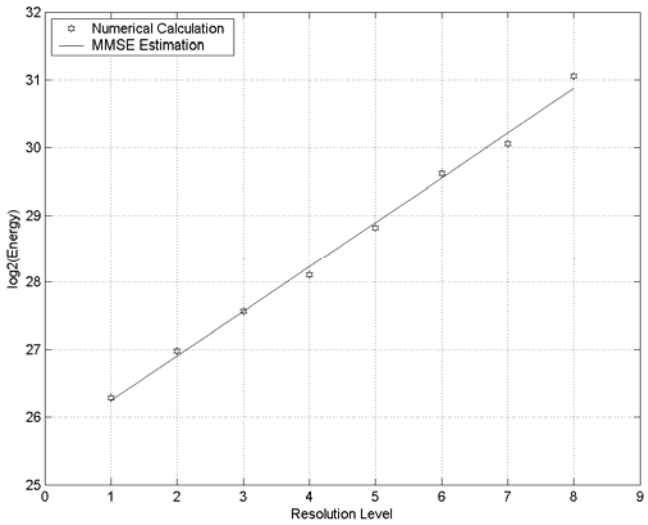
Figure 2-9. *R/S* plot experimental results of trace file StarWarsIV.

2.7.4 Wavelet plot result

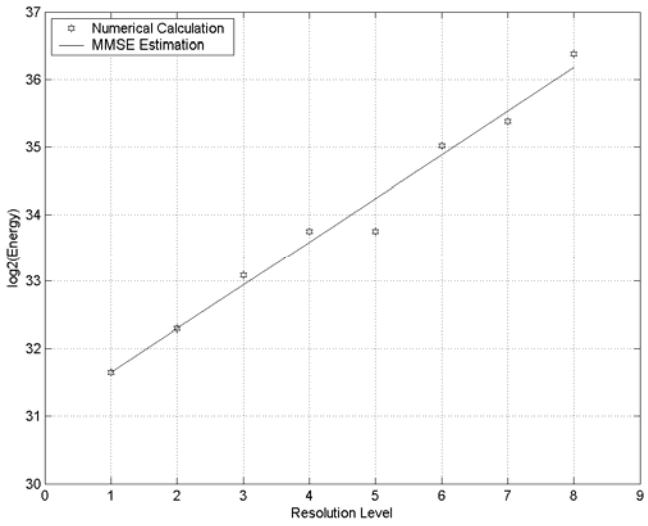
Figure 2-10 and 2-11 are the wavelet plot experimental results of the trace file 1 and StarWarsIV.



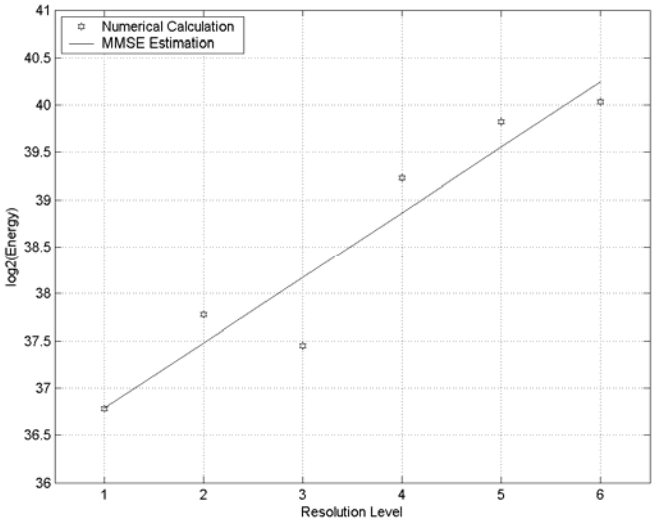
(a) Time interval =0.01 second.



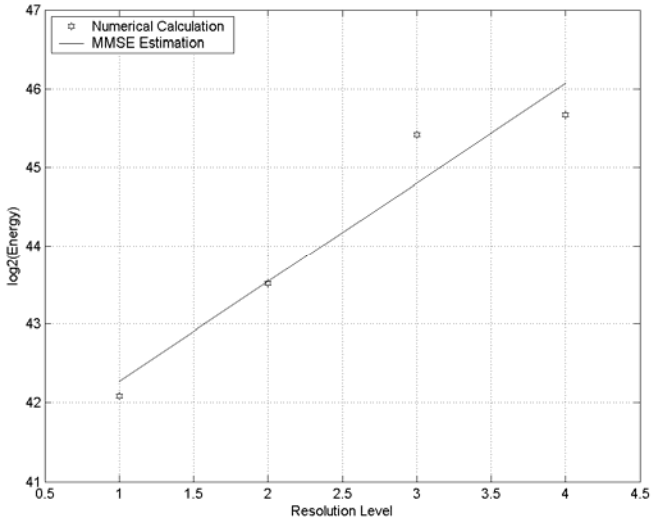
(b) Time interval =0. 1 second.



(c) Time interval =1 second.

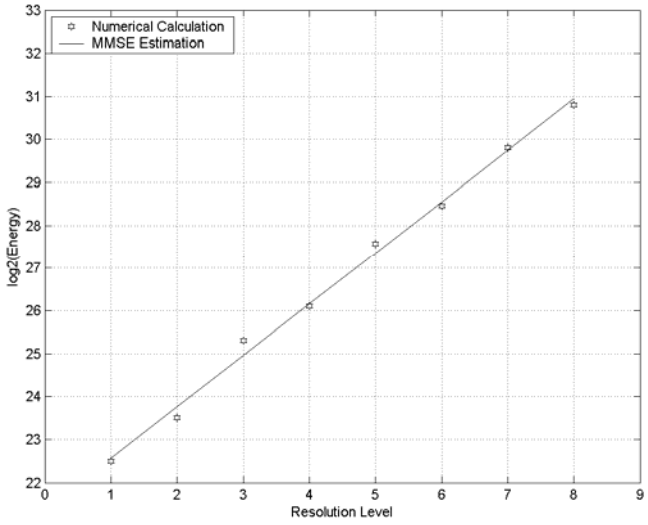


(d) Time interval =10 seconds.

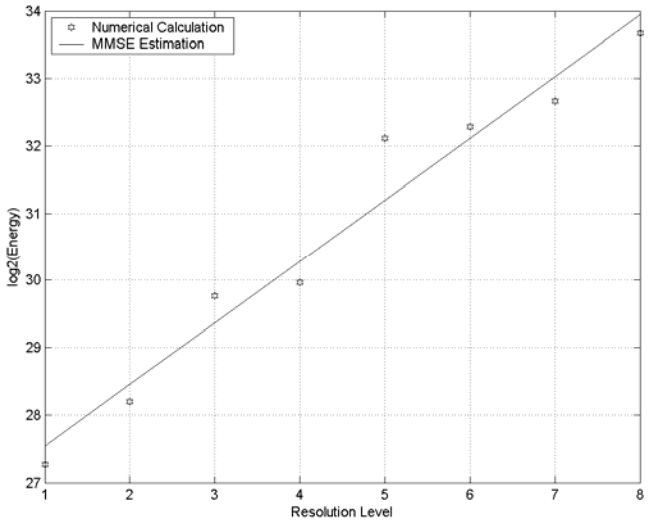


(e) Time interval =100 seconds.

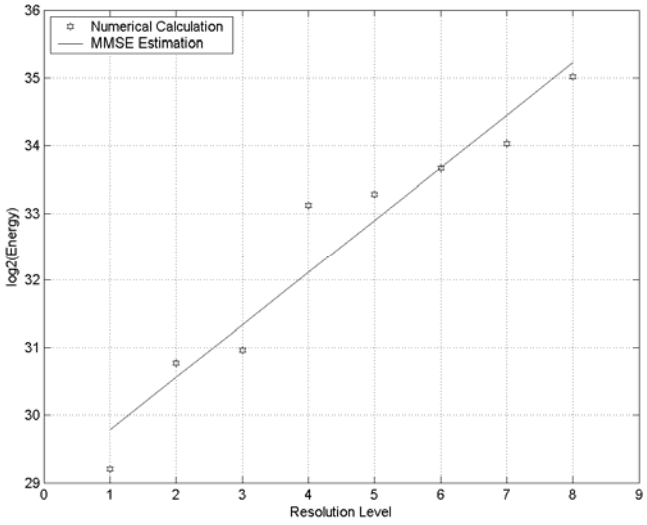
Figure 2-10. Wavelet plot experimental results of trace file 1.



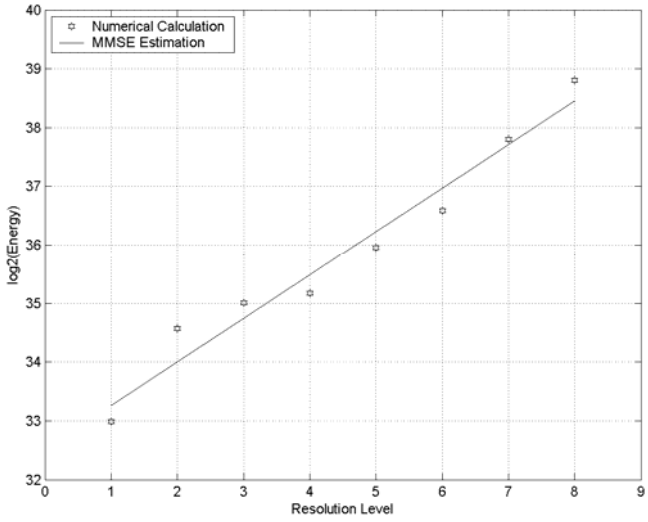
(a) 12 frames per group.



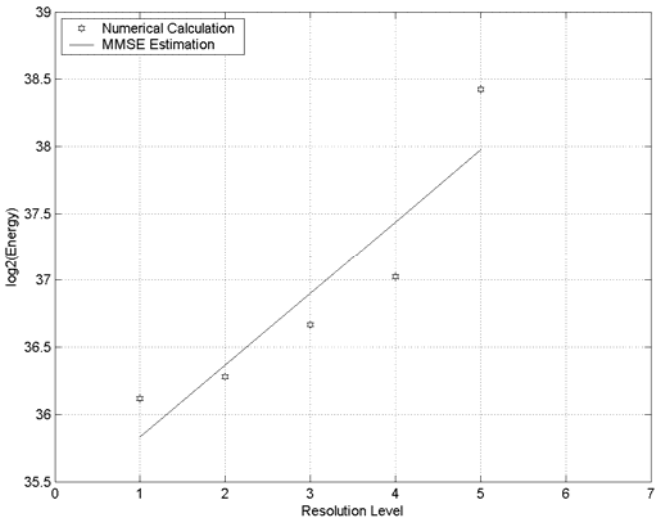
(b) 50 frames per group.



(c)100 frames per group.



(d) 400 frames per group.



(e) 800 frames per group.

Figure 2-11. Wavelet plot experimental results of trace file StarWarsIV. The corresponding estimate Hurst parameters are summarized in Table 2-1 and Table 2-2.

Aggregation Level / Plots	0.01 Sec	0.1 Sec	1 Sec	10 Sec	100 Sec
Variance	0.6884	0.6893	0.6847	0.6766	0.7821
R/S	0.6832	0.8625	0.8497	0.8092	0.8213
Wavelet	0.8048	0.8312	0.8238	0.8458	1.1325

Table 2-1. Estimated *H* for trace file 1.

Aggregation Level / Plot	12Fr	50Fr	100Fr	400Fr	800Fr
Variance	0.7257	0.7271	0.8199	0.9005	0.9544
R/S	0.785	0.7493	0.6638	0.7618	0.8097
Wavelet	1.0715	0.9575	0.889	0.8714	0.7678

Table 2-2. Estimated *H* for trace file StarWarsIV.

Compared to the results of papers [14] and [32], the first trace file's $H \approx 0.8$, and StarWarsIV trace file's H is decreased along with an increase in the block size per group. The *R/S* plot's results are good and very close to the papers' results either using trace file 1 or trace file 2. The variance plot's result is also close to the papers' results. The wavelet plot's results are not as good as above two.

Comparing the above three plots, the *R/S* plot and variance time plot are similar. Although they not be used to obtain an estimate of H, they give us a rough idea of whether a given data set is consistent with self-similar features ($H > 0.5$) or whether it falls within the realm of rational short-range dependent models ($H \approx 0.5$). However the wavelet method is different from the above plots, it assumes the data set satisfies the self-similar properties. Hence, the wavelet plot serves a different purpose than the variance-time plot and the *R/*S plot. Those two techniques are used to test whether a time series is self-similar and if so to obtain a rough estimate of H. The wavelet plot assumes that the time series is a self-similar process of a particular form and provides an estimate of H with a confidence interval [29].

2.8 Summary

This summary introduces the mathematical definitions and properties of self-similarity, and its application to network traffic analysis. Determining if the traffic is self-similar or not and determining the degree of self-similarity are usually done by estimating the Hurst parameter. Three estimation methods, i.e., variance time, *R/S*, and wavelet plot are introduced in detail. These three approaches are then used to estimate the Hurst parameters of two data traffic sets obtained by other researchers. Estimations are

performed at different aggregation levels. It is shown that the results agree with each other. In addition, the differences among these three approaches are also discussed.

CHAPTER III

DIFFERENTIATED SERVICE

3.1 Introduction

The Internet Protocol (IP) is a network layer protocol providing a lowest common denominator for network interconnection. IP can be implemented over almost any network layer. An IP packet consists of a header and payload. The header contains the source and destination addresses, and the type of service (ToS) for the packet, along with other information relevant to the transport of the packet. Figure 3-1 shows the IP protocol header structure. In classical IP networks, a packet traverses the network based solely on its destination address. A decision is taken at each router as to where the packet should next be sent based on the destination address contained in the packet header and the current contents of the router's routing tables.

<table>
<tr><td>Version
(4 bits)</td><td>IHL
(4 bits)</td><td>Type of
Service
(8 bits)</td><td colspan="4">Total length (16 bits)</td></tr>
<tr><td colspan="3">Identification Sequence number
(16 bits)</td><td>(1
bit)</td><td>DF
(1
bit)</td><td>MF
(1
bit)</td><td>Fragment offset (13 bits)</td></tr>
<tr><td colspan="2">Time to live
(8 bits)</td><td>Protocol
(8 bits)</td><td colspan="4">Header checksum (16 bits)</td></tr>
<tr><td colspan="7">Source address (32 bits)</td></tr>
<tr><td colspan="7">Destination address (32 bits)</td></tr>
<tr><td colspan="7">Options (0 or more words)</td></tr>
</table>

Figure 3-1. The IP protocol [7].

Until recently, IP networks supported one service class: best effort. With the development of networking technology, modern networks are desired to support a wide variety of applications, such as interactive TV, IP telephone, on-line gamming, VPNs, etc. However, best-effort cannot satisfy these. Then quality-of-service (QoS) over IP-based networks become an important issue. In 1995, the Internet community began to define an Integrated Services Architecture (ISA) that supports two traffic classes in addition to best-effort service: (1) guaranteed service supports real-time traffic flows that require a quantifiable bound on delay; and (2) controlled load approximates a best-effort service over an un-congested network. Though Integrated Services (IntServ) can support all of the above applications, the actual implementation may be too complicated and not scalable. Differentiated Services (DiffServ) is the current approach for supporting IP QoS.

DiffServ is a relatively simple and coarse method to provide differentiated types of services. Rather than being based on the idea of per-flow resource reservation, DiffSServ assumes that much coarser service differentiation will be satisfied given the plentiful nature of bandwidth. DiffServ make the network's support guaranteed, predictive, and best-effort. There are two important mechanisms that are still used today. They are token bucket filter and weighted fair queuing. Packet flows are classified at the network ingress and receive a certain forwarding treatment in the network based on their priority class. Multiple queuing mechanisms offer differentiated forwarding treatments. Figure 3-2 shows the DiffServ (DS) field of IP packets. DS field reuses the first 6 bits from the former ToS byte. The other two bits are proposed to be used by Explicit

Congestion Notification (ECN). The DS field is used to indicate the forwarding treatment that a packet should receive at a node.

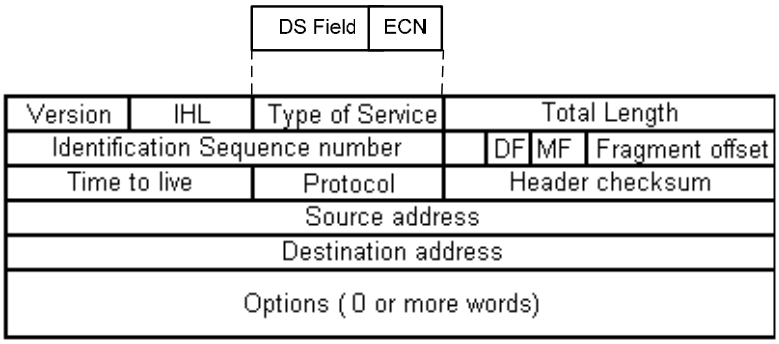


Figure 3-2. Differentiated service field of IP packet [18].

When a packet traverses the boundary between different DS domains, the DS field of the packet may be re-marked according to existing agreements between the domains. DiffServ allows only a finite number of service classes to be indicated by the DS field. Resources are allocated on a per-class basis and the amount of state information is proportional to the number of classes rather than to the number of application flows. The main advantage of the DiffServ approach relative to the IS model is scalability [29], [28].

Providing different levels of service requires two major concerns: control path and data path. The data path mainly includes two basic mechanisms, queue management and scheduling algorithm. In traditional queue management packets are dropped only when the queue is overflowed. The two shortcomings of this queue management are: (1) urgent packets can not be served in time, (2) the queue length tends to be unlimited, which results in longer delay. Buffer management scheme needs to control traffic fairly and efficiently especially under congestion periods. The priority problem can be used to

solve the first problem. It permits the high priority packets to be served earlier than the low priority ones. The queue size problem is solved with the Random Early Detect (RED) to provide a feedback mechanism. In order to present high-priority traffic from starving low-priority traffic, a token bucket filter is used to reserve some resource for other network flows.

In this thesis, we focus on how to apply the queuing discipline to estimate the queuing boundaries in high speed networks. Next, the three common queuing disciplines are introduced.

3.2 Queuing Disciplines

3.2.1 FIFO Queuing

FIFO is a traditional queuing discipline used by most of the routers around the world. It is very simply to implement. Figure 3-3 shows the working theory of FIFO queuing. Packets arriving from different flows are treated to be fair. Packets in the queue are dispatched in the same order strictly corresponding to their arriving order. This means, first packet that comes in is the first packet that goes out [31]. The shortcoming of this method is when the queue is full, the upcoming packets will be discarded, and then packets loss happens. For self-similar networks, burstness is a common phenomenon. How to reduce the packets loss ratio becomes a problem.

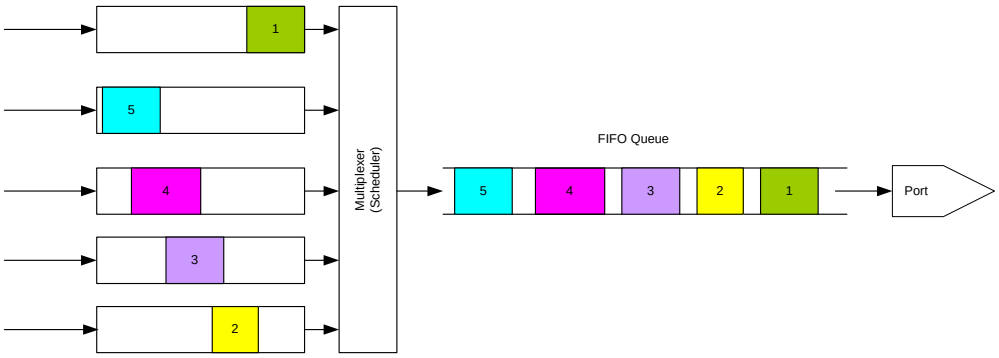


Figure 3-3. FIFO queuing.

3.2.2 Weighted Fair Queuing (WFQ)

In Fair Queuing (FQ) algorithms, each queue is treated equally. Weighted Fair Queuing (WFQ) belongs to the family of Fair Queuing algorithms. It was designed to ensure that each flow is treated equally. Every queue is fairly accessed to network resources in order to prevent bursty flows from consuming more than its shared output bandwidth. Figure 3-4 shows the working theory of WFQ. WFQ has a limited number of queues which are selected by the user or fixed by default. WFQ uses a hashing algorithm to divide the entering packets. When a packet arrives, it is classified by the classifier and assigned to one of the queues. The entry to each queue is served in a weighted round-robin order. Thus the service is 'fair' for every queue. The classifier distributes the packets to corresponding queues based on information taken from the packet header (source address, source port, destination address, protocol, IP precedence, etc.). See Figure 3-1 for the IP protocol structure. In this method, increasing the number of queues as large as possible helps the fairness of the algorithm. It avoids the problems in Priority queuing; however, this might result in large complexity [31].

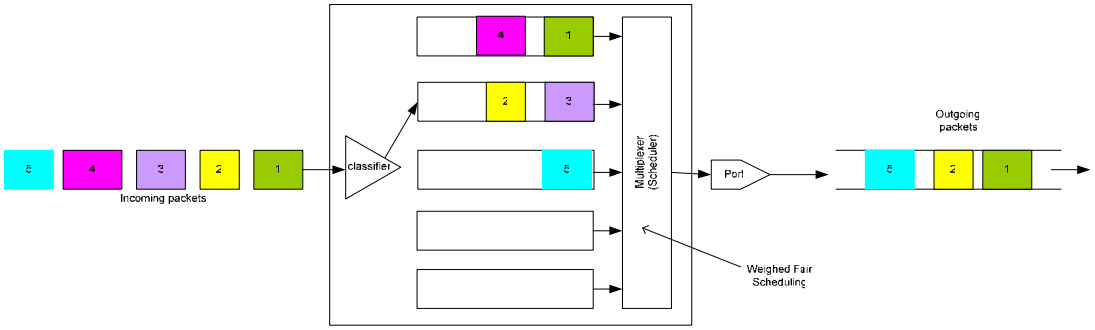


Figure 3-4. Weighted fair queuing.

3.2.3 Priority Queuing

The priority queuing discipline is a relatively simple algorithm where it is possible to implement DiffServ classes. At least two FIFO queues are needed, where having three or four can help improve the performance. Each of these queues has a different priority. For example, having three queues, their priorities were assigned to be high, medium, and low priority. In addition, there are two other mechanisms. One is a classifier, which is in charge of deciding in which of the queues to place the packet based on the information taken from the packet header. The other is a scheduler, which is in charge of emptying the queues by selecting the packets with the highest priority queue until it is empty, next the medium ones, and finally the lowest one. As soon as there are packets in the high priority queue it must be served first until there are no more packets in the queue [31].

Figure 3-5 depicts the priority queuing mechanism. Based on [29] and [28], the structure of DiffServ can be defined as follows. The packets are marked with a Differentiated Service Code Point (DSCP) in the IP header, using the six most significant bits of the IPv4 header. A “behavioral aggregate” (BA) is a collection of flows that should receive the same service and are marked in the same manner. A “per-hop

behavior" (PHB) specifies the treatment that a BA should receive at a DiffServ router. Based on the information in IP, traffic metering information, edge router classifies each packet into a BA. Each BA is mapped to a PHB, which determines its treatment at each node. If traffic in a particular BA exceeds its allocated bandwidth, that BA will result in congestion and packet loss.

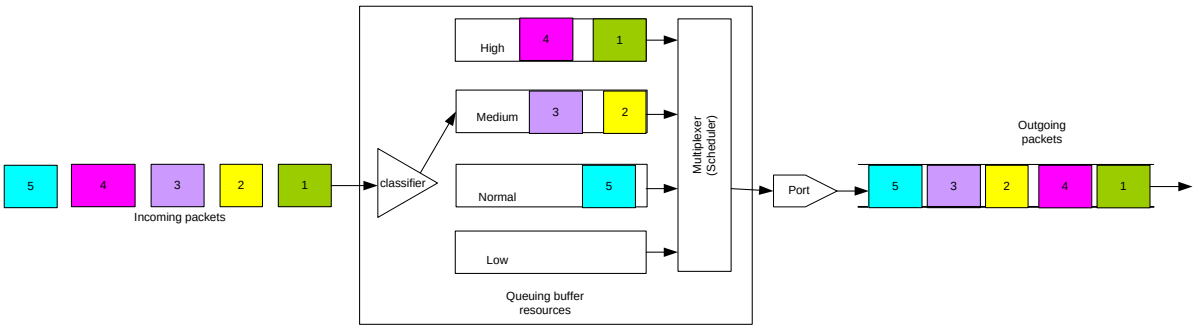


Figure 3-5. Priority queuing.

Based on above classification discipline, packets are assigned priority values according to their QoS requirements, and then are transmitted out in a priority-based order. As a consequence, the packets of highest priority are influenced only by the packets of their equivalent priority. On the other hand, the traffic of the lower priority will be under the influence of the higher and equivalent priority traffic. Hence the packets of a certain priority can be served only when there are no packets available in the queues of higher priority. The problem is the waiting time of lower priority queue is much longer than that of the higher priority. When burstiness happens, the lower priority queue will easily become full, and the packet loss ratio will increase correspondingly.

3.3 Expected Queuing Length

The following simulation is to regenerate the work of paper [25]. The main purpose of the simulation is to get the queue length upper bound and lower bound for FIFO single queue and non-preemptive priority queues in DiffServ networks.

3.3.1 Simple Queuing System

Figure 3-6 shows the node model of a simple queueing model. The data packets from the traffic traces file pass through the Souce Generator and then goes into the FIFO queue. We assumed that the server has a deterministic service rate for each packet. In this model, we assume all the packets have the same priority. The buffer size is infinitive. The first packet that arrives at a router (or server) is the first packet to be transmitted.

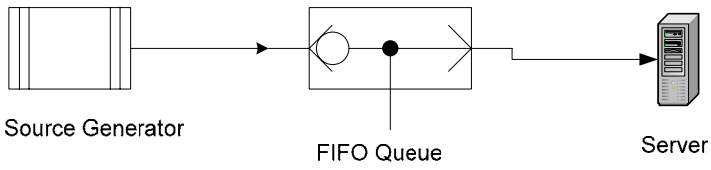


Figure 3-6. Simple queuing model [25].

Let Q_n denote the queue length, X_n represent the number of packets arriving, and C_n be the server capacity at the *n*th time interval. Based on Lindley's equation [25], we have [25]:

$$Q_n = \max(0, Q_{n-1} + X_n - C_n) \tag{3.1}$$

We define a stochastic process Y_n to be the number of the packets that the server can service at time interval *n* [25]:

$$Y_n = -\min(0, Q_{n-1} + X_n - C_n) \tag{3.2}$$

$$Y_n = \max(0, C_n - Q_{n-1} - X_n) \tag{3.3}$$

which denotes the leftover server capacity at the *n*th time interval. Thus, we have [25]:

$$Q_n = Q_{n-1} + X_n - C_n + Y_n \tag{3.4}$$

Because Q_{n-1} and $(X_n - C_n)$ are independent, we get the equation of expected queue length of a singe server [25]:

$$\overline{Q} = \frac{E[(X-C)^2 - E(Y^2)}{2[E(C) - E(X)]} = \frac{Var(C) + Var(X) - Var(Y)}{2[E(C) - E(X)]} \quad . \tag{3.5}$$

Based on $Var(Y) \geq 0$, the upper bound of the queue length is [25]:

$$\overline{Q} = \frac{Var(C) + Var(X) - Var(Y)}{2[E(C\} - E(X)]} \leq \frac{Var(C) + Var(X)}{2[E(C) - E(X)]} \quad . \tag{3.6}$$

Based on $E(Y^2) \leq E(C^2)$, the lower bound of the expected queue length is [25]:

$$\overline{Q} = \frac{E[(X-C)^2] - E(Y^2)}{2[E(C) - E(X)]} \geq \frac{E[(X-C)^2] - E(C^2)}{2[E(C) - E(X)]} \tag{3.7}$$

$$\overline{Q} = \max\left(0, \frac{Var(X) + E^2(X) - 2E(C)E(X)}{2[E(C) - E(X)]}\right) \quad . \tag{3.8}$$

3.3.2 Non-preemptive (Head-of-Line) Priority System

In DiffServ networks, it is necessary to recognize the priority and then provide control support for different QoS requests. Based on a certain classification discipline, packets are assigned priority values according to their QoS requirement, and then are

transmitted out in a priority-based order. Figure 3-7 is the HOL priority queuing model. In this system, we assume that there are only two queues. These queues are independent from each other. With a head-of-line priority scheme, once a packet is in the transmission of the server (or router) it will not be interrupted by a packet of higher priority that arrives later. We assume class 1 has the highest priority; class 2 has the lower priority. The packet arriving rates of different classes are the same.

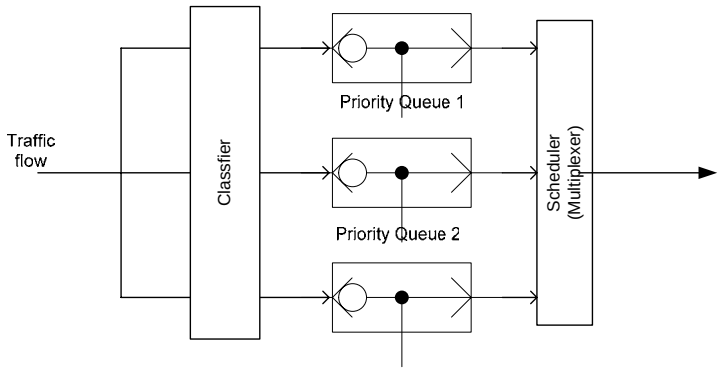


Figure 3-7. HOL priority queuing model of one hop in networks [25].

Let $Q_n^{(j)}$ denote the queue length of class j, X_n^j represents the number of packets arriving of class j, $Y_n^{(j-1)}$ be the available service capacity for packets of class j, $Y_n^{(j)}$ be the left over server capacity seen by class j, $Y_n^{(0)}$ represents the total capacity of the server (same as C_n in a simple queuing system). We define that:

$$Y_n^{(j)} = -\min(0, Q_{n-1}^{(j)} + X_n^{(j)} - Y_n^{(j-1)}) = \max(0, Y_n^{(j-1)} - Q_{n-1}^{(j)} - X_n^{(j)}). \quad (3.9)$$

Then, we have

$$Q_n^{(j)} = Q_{n-1}^{(j)} + X_n^{(j)} - Y_n^{(j-1)} + Y_n^{(j)} \quad (3.10)$$

The expected queue length of class j is [25]:

$$\overline{Q^{(j)}} = \frac{Var(C) + \sum_{i=1}^{j} Var(X^{(j)}) - 2\sum_{i=1}^{j-1} E(Y^{(j)})\overline{Q^{(i)}} - Var(Y^{(j)})}{2E(Y^{(j)})} \tag{3.11}$$

where [25],

$$E(Y^{(j)}) = E(Y^{(0)}) - \sum_{i=1}^{j} E(X^{(i)}) \ . \tag{3.12}$$

Based on $E(Y^{(j)^2}) \le E(Y^{(j-1)^2})$, the lower bound of the expected queue length of class *j* is [25]:

$$\overline{Q^{(j)}} \ge \left[\frac{Var(X^{(j)}) + E(X^{(j)}\left[E(X^{(j)}) - 2K^{(j-1)}\right]}{2K^{(j)}} \right]^{+} \tag{3.13}$$

$$\overline{Q^{(j)}} = \max\left(0, \frac{Var(X^{(j)}) + E(X^{(j)}\left[E(X^{(j)}) - 2K^{(j-1)}\right]}{2K^{(j)}}\right) \equiv \overline{Q^{(j)}} \ . \tag{3.14}$$

The upper bound of expected queue length of class *j* is [25]:

$$\overline{Q^{(j)}} \le \frac{Var(Y^{(0)}) + \sum_{i=1}^{j} Var(X^{(i)}) - 2\sum_{i=1}^{j-1} K^{(j)}\overline{Q^{(j)}}}{2K^{(j)}} \ . \tag{3.15}$$

3.3.3 Multi-class DiffServ Queuing System under Self-similar Traffic

The number of packets arriving at the *n*th time interval has the standard deviation of [25]:

$$STD(X) = \Delta t^{H}\sqrt{ma} \tag{3.16}$$

Here, Δt denotes time interval, H represents Hurst Parameter, m is the mean input rate, a is the variance coefficient [24]. We obtain

$$m = \frac{\sum_{i=1}^{n} X_i}{n \times \Delta t} \ . \tag{3.17}$$

$$a = \frac{\sqrt{Var(X)}}{E(X)} \times 100 \quad . \tag{3.18}$$

Then, the lower bound of the expected queue length of class *j* can be represented as equation (3.19) or (3.20) [25]:

$$\overline{Q^{(j)}} \geq \left[\frac{m^{(j)} a^{(j)} \Delta t^{2H^{(j)}} + m^{(j)} \left[m^{(j)} - 2K^{(j-1)} \right]}{2K^{(j)}} \right]^{+} \tag{3.19}$$

$$\overline{Q^{(j)}} = \max\left(0, \frac{m^{(j)} a^{(j)} \Delta t^{2H^{(j)}} + m^{(j)} \left[m^{(j)} - 2K^{(j-1)} \right]}{2K^{(j)}} \right) \equiv \overline{Q^{(j)}} \, . \tag{3.20}$$

The upper bound of the expected queue length of class *j* can be represented as equation (3.21) [25]:

$$\overline{Q^{(j)}} \leq \frac{Var(Y^{(0)}) + \sum_{i=1}^{j} m^{(i)} a^{(j)} \Delta t^{2H^{(j)}} - 2\sum_{i=1}^{j-1} K^{(j)} \overline{Q^{(j)}}}{2K^{(j)}} \quad . \tag{3.21}$$

3.4 Simulation and Results analysis

3.4.1 Simulation 1

In the simulation, the trace file is the same as that in paper [14]. We use equation (3.16), (3.17), and (3.18) to calculate the STD(*X*) of the self-similar traffic. The time interval between the *n*th and *(n+1)*th packets is 0.01 second. The unit of the packet arrival rate is packet numbers per 0.01 seconds.

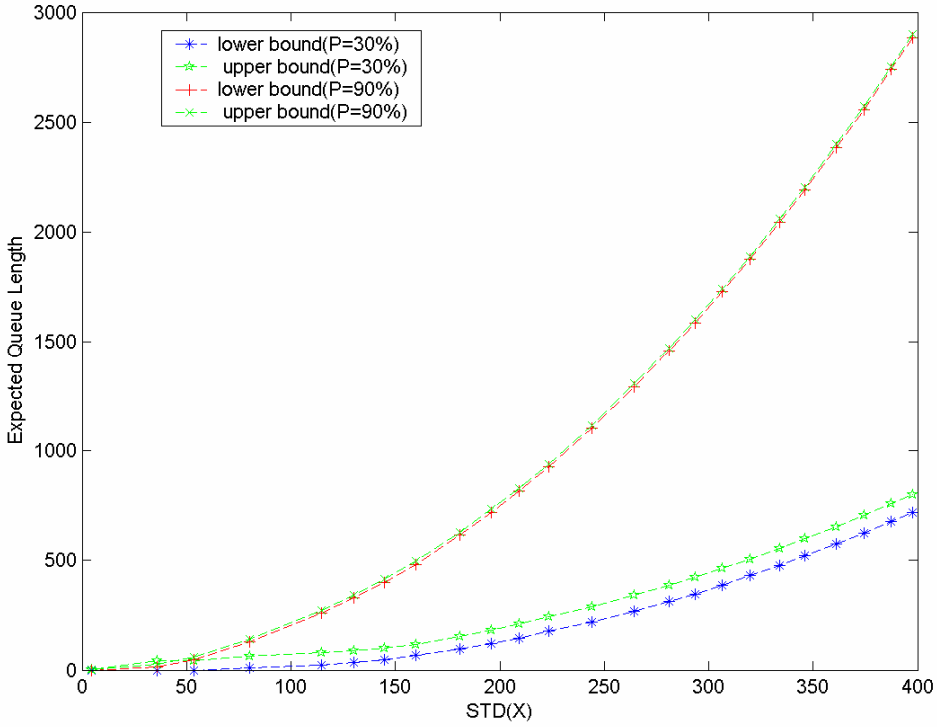
To calculate the lower and upper bound of expected queue length of a single server, we use equation (3.6), (3.7), and (3.8). The packet arrival rate is obtained from the number of packets per 0.01 second. In the presented examples, we consider the utilization of single queue be 90% and 30%. We have

$$\rho = \frac{E(X)}{E(C)} \tag{3.22}$$

where ρ is the server utilization. We assume that the service rate is always 1.2 times of the packets arrival rate. Based on equation (3.22), we have the mean service rate. We assume the service rate is fixed for simulation 1. We will generate the stochastic process of *Cn*. We use the self-generated *Cn* process to calculate corresponding *E*(*C*) and *Var*(*C*). Besides, we use equation (3.19), (3.20), and (3.21) to calculate the expected queue length.

The results are shown in Figure 3-8.

All others assumptions are keep the same as the simulation 1.



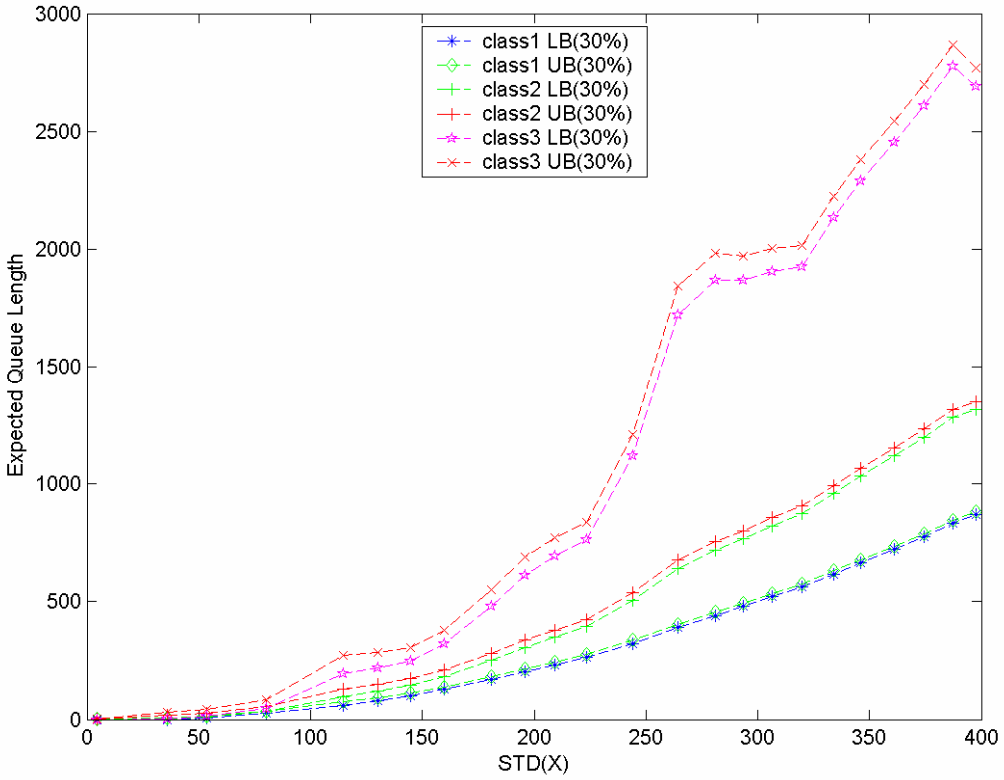
single queue system (with utilization of 90% and 30%).

Figure 3-9. Expected queue length of simulation 1.

3.4.1 Simulation 2

In simulation 2, we consider there are three classes of priority queues. In the priority model, the percentages of class 1, class2, and class 3 are 33%, 33.3%, 33.3%, respectively. All others assumptions and units are similar to simulation 2.

The results are shown in Figure 3-9.



priority queuing system.

Figure 3-9. Expected queue length of simulation 2.

3.4.4 Analysis of Simulation 1-2's results

Figure 3-8 illustrates the 30% and 90% utilization boundary conditions for a single queue traffic steam. The results show that the lower the utilization of the server, the smaller is the queue length.

Figure 3-9 provides an insight on how the priority assignment and the class traffic affects the queue length, and the tightness of the upper and lower bounds. The highest priority class only needs to consider the variance of its own class because no other classes will have any influence over it.

Comparing the single queue 90% utilization in Figure 3-8 to the combined 90% utilization of class 1 through 3 in Figure 3-9, we can obtain the advantages of differentiated services. The queuing length bounds of class 1 and class 2 in Figure 3-9 are significantly lower compared to the single queue 90% utilization of Figure 3-8. In addition, class 3 of Figure 3-9 is only slightly worse than the single queue 90% utilization of Figure 3-8.

Overall, for the above examples, the HOL priority was observed. It greatly reduced the queuing length (i.e. waiting time) of the class 1 and class 2. And it only shows a little disadvantage in class 3 compared to the FIFO single queue (e.g. 90% utilization case in Figure 3-9).

3.5 Summary

The upper and lower bounds of the expected queue length are estimated in previous paper [25]. It is based on the arrival rate and deterministic server capacity for both single class and multi-class steady-state queuing systems. “The derivations here provide a practical and effective method to analyze the effects of self-similar traffic in DiffServ networks based on the utilization and priority assignment of the traffic”[25]. Equations (3.19) through (3.21) can be used to provide the boundary of the queue length. They also show that the Hurst parameter, which indicates the degree of burstness, affects

the queuing length in high speed networks. The mean waiting time ($\overline{W}$) can also be obtained by applying the *Little*'s theorem. The difference between lower and upper bounds is the most significant in the lowest priority class. On the other hand, there is not much difference between lower and upper bounds in the highest priority class [25].

CHAPTER IV

QUEUE LENGTH ANALYSIS IN END-TO-END DIFFERENTIATED NETWORKS WITH SELF-SIMILAR TRAFFIC

4.1 Limitation of Previous Work and Advanced Approach

In the previous chapter, the upper and lower bounds of the expected queue length in steady-state queuing systems were observed. The study was focused on priority queuing of 2 point-to-point nodes in self-similar networks. The queuing model in each node was based on GI/G/1 queuing system. For simplicity, the previous works consider a bufferless fluid model of a router. So, the following directions are used to improve the previous work of [25], [15], and [30].

(1) First, this approach can be extended to the situation in real routers where transmission is packetized. The drop-on-input scheme should be considered. Then queue length boundaries will no longer be a problem. The new concern issue of focus will be the packet loss probability.

(2) Second, GI/G/1 queue model can be changed to GI/G/*m* queue model. The upper and lower bounds of the queue length are still the concern issue. The only difference is the model has 2 servers. It falls into the multiple priority queuing domains. A lot of research has been done in this area.

(3) Third, pervious work has been based on the one hop model [24] and [25]. However, realistic networks are composed of more than two directly-connected nodes. In self-similar traffic, cross traffics will influence each other. Then their queue lengths will change respectively. So, the next step is to setup the structure and extend the network to multiple networks nodes. To extend this to multiple nodes with self-similar traffic, should begin with the simplest case of 3 nodes and then extend to more nodes.

4.2 Core-Stateless Fair Queuing

Many researchers observed that routers with fair bandwidth allocation mechanisms greatly benefit the end-to-end congestion-control. Until now, the fair allocations were achieved by using per-flow queuing mechanisms which are more complex to implement. In fact, fair allocation mechanisms require the routers to maintain the states and perform per-flow basis operations. Therefore, a number of scheduler designs have been proposed to reduce the complexity of the packet classification and the per-flow management.

4.2.1 Definition of Core-Stateless Fair Queuing

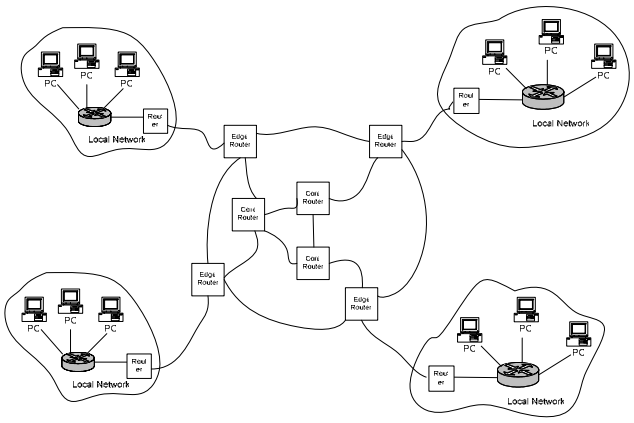
In high-speed networks, the routers in the network can be divided as edge routers and core routers as depicted in Figure 4-1 (a). There is an observation regarding the multihop schedulers. The edge routers still maintain per flow operation, but the core routers can be core-stateless as depicted in Figure 4-2 (b). In other words, the core routers do not maintain per-flow state in high speed networks [15]-[16].

In this thesis, two assumptions are made, similar to [30]:

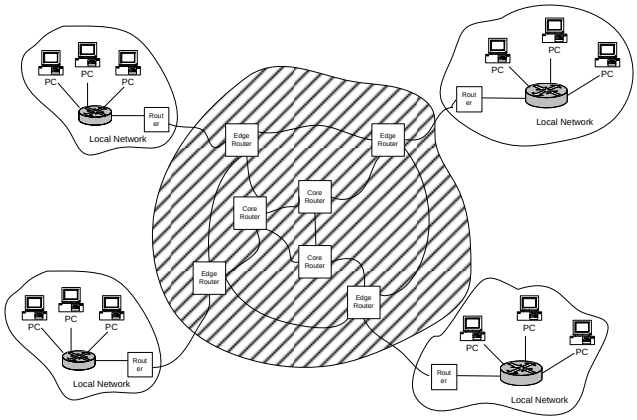
(1) Fair allocation mechanisms play an important role in congestion control. The focus on the implications if indeed they were true, not claiming whether they are true or not.

(2) The complexity of existing fairs allocation mechanisms is a substantial hindrance to their adoption.

Then, queue length analysis uses the architecture proposed in [30]. The approach identifies a contiguous region of the network to distinguish between the edge and the core of the routers. Edge routers region keep using per-flow operation to estimate the per-flow rate and label the packets passing through them. The core router region allows FIFO queuing and do not maintain per-flow state. The bandwidth allocation within the core routers is fair. That is, the bandwidth of each core router is same. Thus, if this approach were adopted in the high-speed networks, and fail allocation mechanisms were adopted for the slower links outside of these high-speed interiors, then fail allocations could be achieved everywhere. Assume each FIFO queue is using a HOL queuing discipline. This approach is called Core-Stateless Fair Queuing (CSFQ) since the core routers keep no per-flow state [30]. In this thesis, only the multi-node networks in the core router region is conserdered.



(a) Network architecture



(b) Core-stateless fair queuing domain

Figure 4-1. Overview of network architecture.

4.2.2. Advantages and Disadvantages of Applying CSFQ

Advantages:

(1) CSFQ uses a distributed algorithm to avoid maintaining the per-flow state at each router so that there will be a much lower complexity scheduler in the core routers.

(2) CSFQ uses tradition simple queuing discipline FIFO queue to avoid per-flow buffering and scheduling.

(3) CSFQ is approximately fair queuing.

Disadvantages:

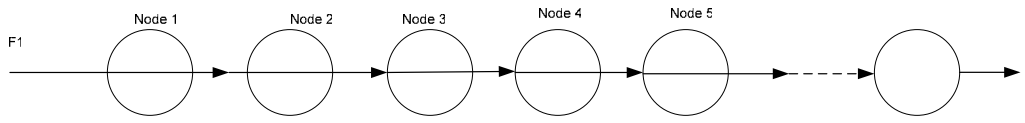
(1) There is no guarantee on the average per-flow delay, and no guarantee of a feasible delay class allocation.

(2) The CSFQ approach requires some configuration, with edge routers distinguished from the core routers. Moreover, CSFQ must be adopted one island at a time rather than router by router.

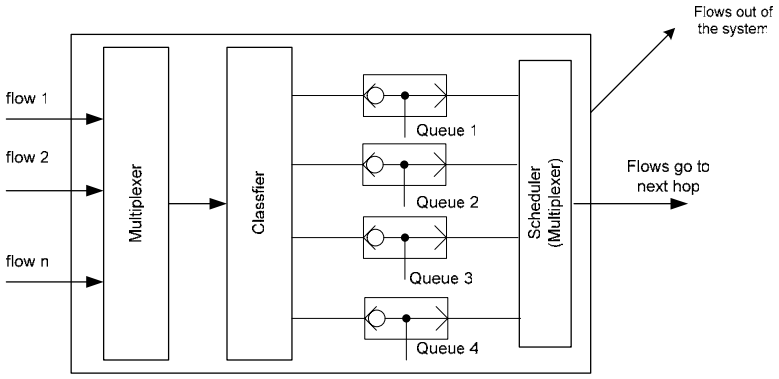
4.3 Fluid Model

The fluid analysis model enables us to accurately investigate the queue length of each hop in end-to-end differentiated service networks. The derivation of single hop differentiated service networks can be extended to multi-hop differentiated service networks.

First, a simple queuing model is considered. It is assumed that the server at each hop has a deterministic service rate. The buffer size is infinite. Figure 4-3 (a) shows the queuing model.



(a) Fluid Model in End-to-End High Speed Networks.



(b) Flows at Each Hop.

Figure 4-3. Fluid Model.

The flows are continuous streams of bytes. We precisely calculate each flow's mean arrival rate $m_i(t)$. $m_i(t)$ denotes the mean arrival rate of flow *i* at time *t*. The fair share rate is the output rate of each node. Each hop has the same output rate. Let $o(t)$ denote the fair share rate at time *t*, and *C* represent the output link speed. In general, when the total arrival rate $M(t) = \sum_{i=1}^{n} m_i(t)$ is less than the output link speed of a node, all the flows will be sent to the downward hops. If $M(t) \le C$, the queue length of this node is 0. If $M(t) > C$, a fraction, $\left(\frac{m_i(t) - o(t)}{m_i(t)} \right)$ of the flows will be queued. Based on the above queued statistics, we will calculate each hop's queue length [30].

The above algorithm can be extended to support flows with different priorities. Let p_i denote the weight of flow i. If $M(t) > C$, a fraction, $\left(\frac{m_i(t)/p_i - o(t)}{m_i(t)/p_i} \right)$ of the flows will be queued [30].

4.4 Single Queue Length in end-to-end DiffServ networks

Let $Q_{n,h}$ denote the queue length at the h^{th} hop, $X_{n,h}$ represent the number of arrival packets at the h^{th} hop, $C_{n,h}$ be the number of packets served or available server capacity at the h^{th} hop, and $Y_{n,h}$ denote the left over server capacity at the h^{th} hop.

The left over server capacity of class j at the h^{th} hop is defined as (4.1) or (4.2):

$$Y_{n,h} = -\min(0, Q_{n-1,h} + X_{n,h} - Y_{n,h}) \tag{4.1}$$

$$Y_{n,h} = \max(0, Y_{n,h} - Q_{n-1,h} - X_{n,h}) \tag{4.2}$$

The queue is stationary, and hence have:

$$E(Q_{n,h}) = E(Q_{n-1,h}) \tag{4.3}$$

Since $Q_{n-1,h}$ and $(X_{n,h} - C_{n,h})$ are independent, (4.3) can be squared on both sides to get the equation of the expected queue length of a single server at the h^{th} hop as shown in (4.9)

$$\overline{Q} = \max\left(0, \frac{Var(X_h) + E^2(X_h) - 2E(C_h)E(X_h)}{2[E(C_h) - E(X_h)]}\right) \quad . \tag{4.4}$$

Proof: First, have:

$$(Q_{n,h} - Y_{n,h})^2 = [(Q_{n-1,h} + (X_{n,h} - C_{n,h})]^2 . \tag{4.5}$$

Equation (4.5) is equivalents to (4.6):

$$Q_{n,h}{}^2 - 2Q_{n,h}Y_{n,h} + Y_{n,h}{}^2 = Q_{n-1,h}{}^2 + 2Q_{n-1,h}(X_{n,h} - C_{n,h}) + (X_{n,h} - C_{n,h})^2 \tag{4.6}$$

Equation (4.7) is derived from (4.6)

$$E(Q_{n,h}{}^2) + E(Y_{n,h}{}^2) = E(Q_{n-1,h}{}^2) + 2E(Q_{n-1,h}(X_{n,h} - C_{n,h})) + E((X_{n,h} - C_{n,h})^2) \tag{4.7}$$

Then, based on (4.3), we can obtain (4.8)

$$E(Q_{n-1,h}) = \frac{E((X_{n,h} - C_{n,h})^2) - E(Y_{n,h}{}^2)}{2(C_{n,h} - X_{n,h})} \quad . \tag{4.8}$$

When $n \to \infty$,

$$\overline{Q_h} = \frac{E[(X_h - C_h)^2] - E(Y_h{}^2)}{2[E(C_h) - E(X_h)]} \tag{4.9}$$

Since

$$E[(X_h - C_h)^2] = Var(X_h - C_h) + E^2(X_h - C_h) \tag{4.10}$$

Based on $Var(Y_h) \geq 0$, the upper bound of the queue length is:

$$\overline{Q}_h = \frac{Var(C_h) + Var(X_h) - Var(Y_h)}{2[E(C_h) - E(X_h)]} \leq \frac{Var(C_h) + Var(X_h)}{2[E(C_h) - E(X_h)]} \quad . \tag{4.11}$$

Based on $E(Y_h^2) \leq E(C_h^2)$, the lower bound of the expected queue length is:

$$\overline{Q} = \frac{E[(X_h - C_h)^2] - E(Y_h^2)}{2[E(C_h) - E(X_h)]} \geq \frac{E[(X_h - C_h)^2] - E(C_h^2)}{2[E(C_h) - E(X_h)]} \tag{4.12}$$

4.5 HOL Priority in end-to-end DiffServ networks

Let $Q_{n,h}^{(j)}$ denote the queue length of class j at the h^{th} hop, $X_{n,h}^{j}$ represent the number of arrival packets of class j at the h^{th} hop, $Y_{n,h}^{(j-1)}$ be the number of packets served or available service capacity of class j at the h^{th} hop, $Y_{n,h}^{(j)}$ denote the left over server capacity of class j at the h^{th} hop, and $Y_{n,h}^{(0)}$ be the total servers capacity at the h^{th} hop (same as C_n in the simple queuing system).

The left over server capacity of class j at the h^{th} hop is defined as (4.1) or (4.2):

$$Y_{n,h}^{(j)} = -\min(0, Q_{n-1,h}^{(j)} + X_{n,h}^{(j)} - Y_{n,h}^{(j-1)}) \tag{4.13}$$

$$Y_{n,h}^{(j)} = \max(0, Y_{n,h}^{(j-1)} - Q_{n-1,h}^{(j)} - X_{n,h}^{(j)}) \tag{4.14}$$

Then, the queue length of class j at the h^{th} hop at the n^{th} time interval is:

$$Q_{n,h}^{(j)} = Q_{n-1,h}^{(j)} + X_{n,h}^{(j)} - Y_{n,h}^{(j-1)} + Y_{n,h}^{(j)} \tag{4.15}$$

According to (3.12), the expected queue length of class j at the h^{th} hop is:

$$\overline{Q_h^{(j)}} = \frac{Var(C_h) + \sum_{i=1}^{j} Var(X_h^{(j)}) - 2\sum_{i=1}^{j-1} E(Y_h^{(j)})\overline{Q_h^{(j)}} - Var(Y_h^{(j)})}{2E(Y_h^{(j)})} \quad . \tag{4.16}$$

where,

$$K_h^{(j)} = E(Y_h^{(j)}) = E(Y_h^{(0)}) - \sum_{i=1}^{j} E(X_h^{(j)}) \quad . \tag{4.17}$$

According to $E(Y_h^{(j)^2}) \le E(Y_h^{(j-1)^2})$, the lower bound of the expected queue length of class j at the h^{th} hop is:

$$\overline{Q_h^{(j)}} \ge \left[\frac{Var(X_h^{(j)}) + E(X_h^{(j)})\left[E(X_h^{(j)}) - 2K_h^{(j-1)}\right]}{2K_h^{(j)}} \right]^+ \tag{4.18}$$

$$\overline{Q_h^{(j)}} = \max\left(0, \frac{Var(X_h^{(j)}) + E(X_h^{(j)})\left[E(X_h^{(j)}) - 2K_h^{(j-1)}\right]}{2K_h^{(j)}}\right) \equiv \overline{Q_h^{(j)}} \quad . \tag{4.19}$$

The upper bound of expected queue length of class j at the h^{th} hop is:

$$\overline{Q_h^{(j)}} \le \frac{Var(Y_h^{(0)}) + \sum_{i=1}^{j} Var(X_h^{(j)}) - 2\sum_{i=1}^{j-1} K_h^{(j)}\overline{Q_h^{(j)}}}{2K_h^{(j)}} \tag{4.20}$$

4.6 End-to-End DiffServ under self-similar traffic

Based on the self-similar model in [32], the variance of the arrival rate of class j at the h^{th} hop can be expressed by equation (4.21):

$$Var(X_h^j) = m_h^{(j)} a_h^{(j)} \Delta t^{2H_h^{(j)}} \tag{4.21}$$

Then, the lower bound of the expected queue length of class j at the h^{th} hop can be represented as shown in equation (4.22) or (4.23):

$$\overline{Q_h^{(j)}} \ge \left[\frac{m_h^{(j)} a_h^{(j)} \Delta t^{2H_h^{(j)}} + m_h^{(j)}\left[m_h^{(j)} - 2K_h^{(j-1)}\right]}{2K_h^{(j)}} \right]^+ \tag{4.22}$$

$$\overline{Q_h^{(j)}} = \max\left(0, \frac{m_h^{(j)} a_h^{(j)} \Delta t^{2H_h^{(j)}} + m_h^{(j)}\left[m_h^{(j)} - 2K_h^{(j-1)}\right]}{2K_h^{(j)}}\right) \equiv \overline{Q^{(j)}} \quad . \tag{4.23}$$

The upper bound of the expected queue length of class j at the h^{th} hop can be represented as shown in equation (4.24):

$$\overline{Q_h^{(j)}} \le \frac{Var(Y_h^{(0)}) + \sum_{i=1}^{j} m_h^{(j)} a_h^{(j)} \Delta t^{2H_h^{(j)}} - 2\sum_{i=1}^{j-1} K_h^{(j)} \overline{Q_h^{(j)}}}{2K_h^{(j)}} \quad . \tag{4.24}$$

4.7 Simulation Results and Analysis

In the simulation, two types of flows will be considered. There is one target flow, flow F1, entering the network through the first hop and exiting through the last hop. There are one cross traffic flows. They are flow F2 which traverse a single node and then exits the network. F1 flow is the main flow, F2 flow is considered to be interference flow.

We assume the traffic flows, F1, F2, and F3 have the same characteristics. They have the same mean input rate, and Hurst Parameter. The Hurst parameter will be between 0.6 and 0.9 for each traffic flow. Each flow contains four classes, class 1, class 2, class 3 and class 4. Class 1 has the highest priority; class 2 has the second highest priority, and so on. The combined server's utilization is 90%. They have the same utilization at each hop. The upper and lower bound of the queuing length of each class will be investigated.

In simulation 1, Figure 4-3 shows the result of four priority queue length at five hops. The arrival traffic is classified into 4 classes at each hop. The combined utilization is 90%. Each class occupies 25%. It depicts the queuing length for each class at a certain time slot, t = 410 and t = 710, at corresponding hop. It can be seen that the queue length of each class at each hop is a constant.

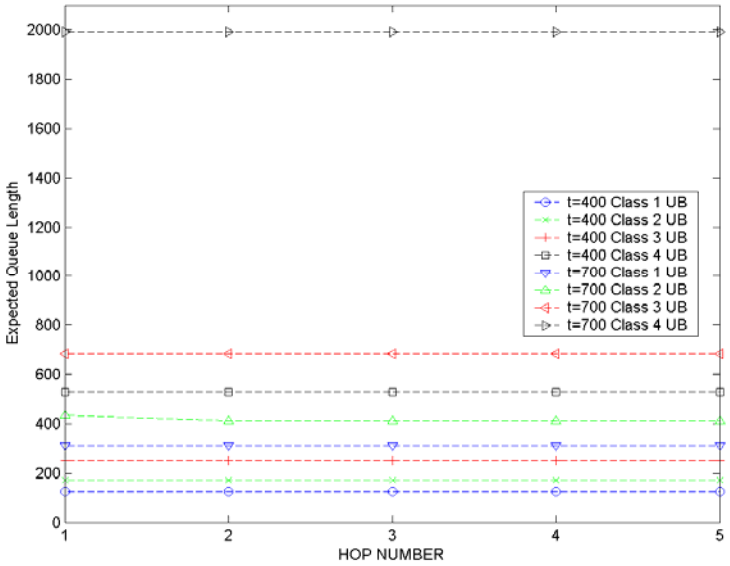
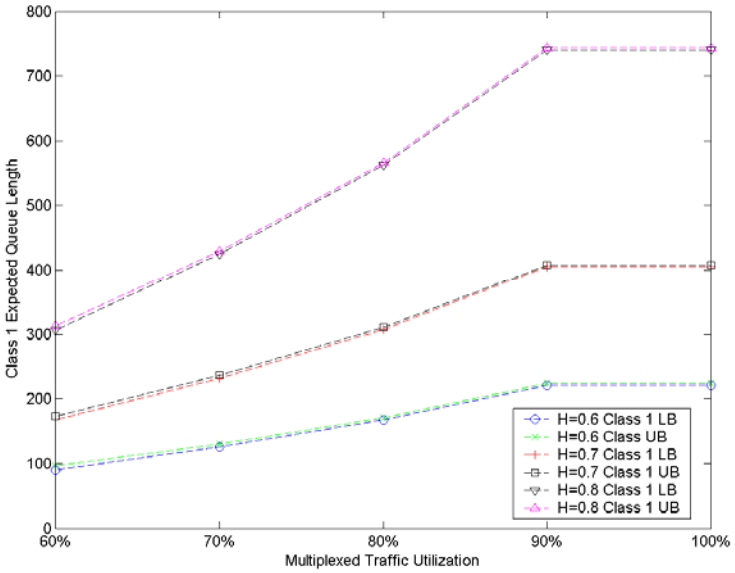
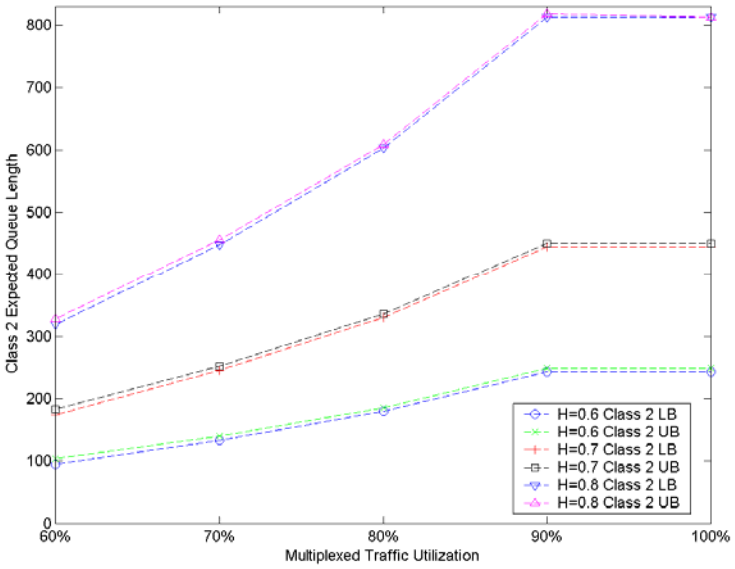


Figure 4-3. Priority Class Queue Length at Each Hop.

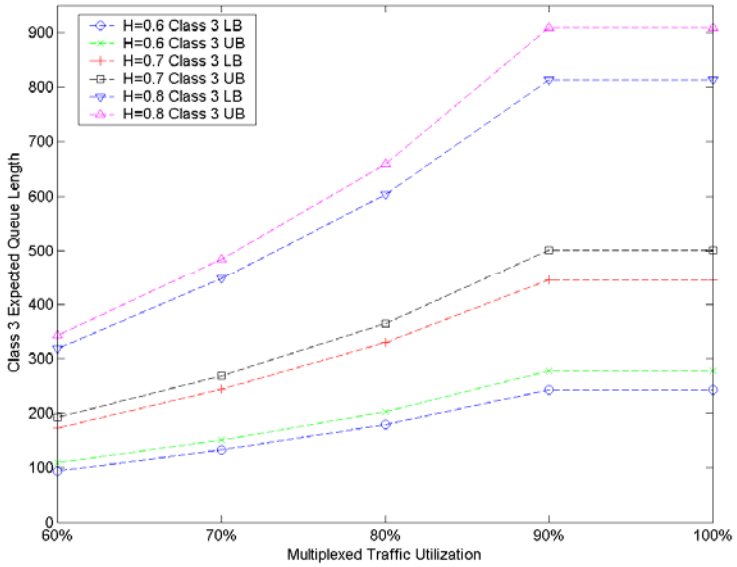
Figure 4-4 (a)-(d) show the class 1 to class 4 expected queue length relationship to multiplexed traffic Utilization and Hurst Parameter. They show that higher the utilization of the multiplexed traffic, the larger the expected queue length is. When multiplexed traffic utilization is fixed, for example at 80%, class 1 always has the smallest queue length among the four classes at each hop. Class 2 always has a smaller queue length than class 1, but larger than Class 3 and Class 4 at each hop. In addition, the upper and lower bound of class 1 and class 2 are tighter (i.e., closer) than the other 2 classes. Hurst parameter also affects expected queue length of each class. For example, in Figure 4-4(a), the combined utilization of all the traffic is 70%, Class 1's expected queue length is larger at $H = 0.8$ than at $H = 0.6$ or at $H = 0.7$. Hurst parameter indicates the burstiness of the data traffic. The degree of burstiness increases with the Hurst parameter. So, larger H causes longer queuing delay or waiting time at each hop.



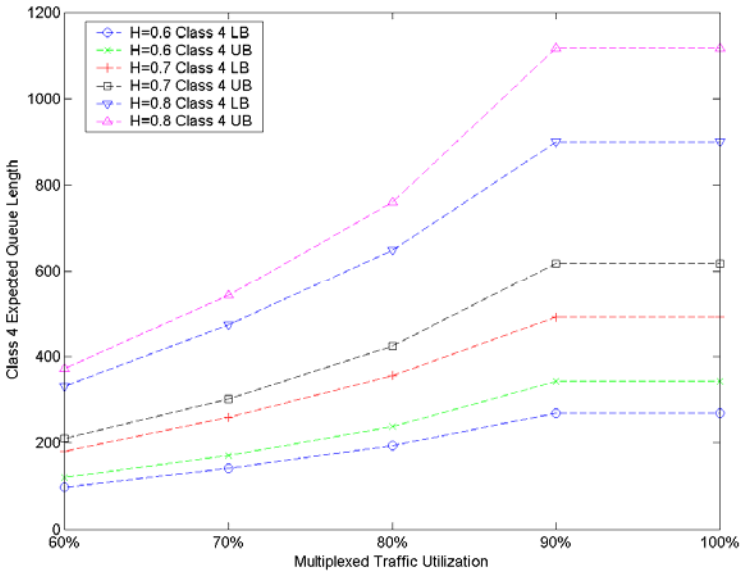
(a) Class 1 Expected Queue Length



(b) Class 2 Expected Queue Length



(c) Class 3 Expected Queue Length

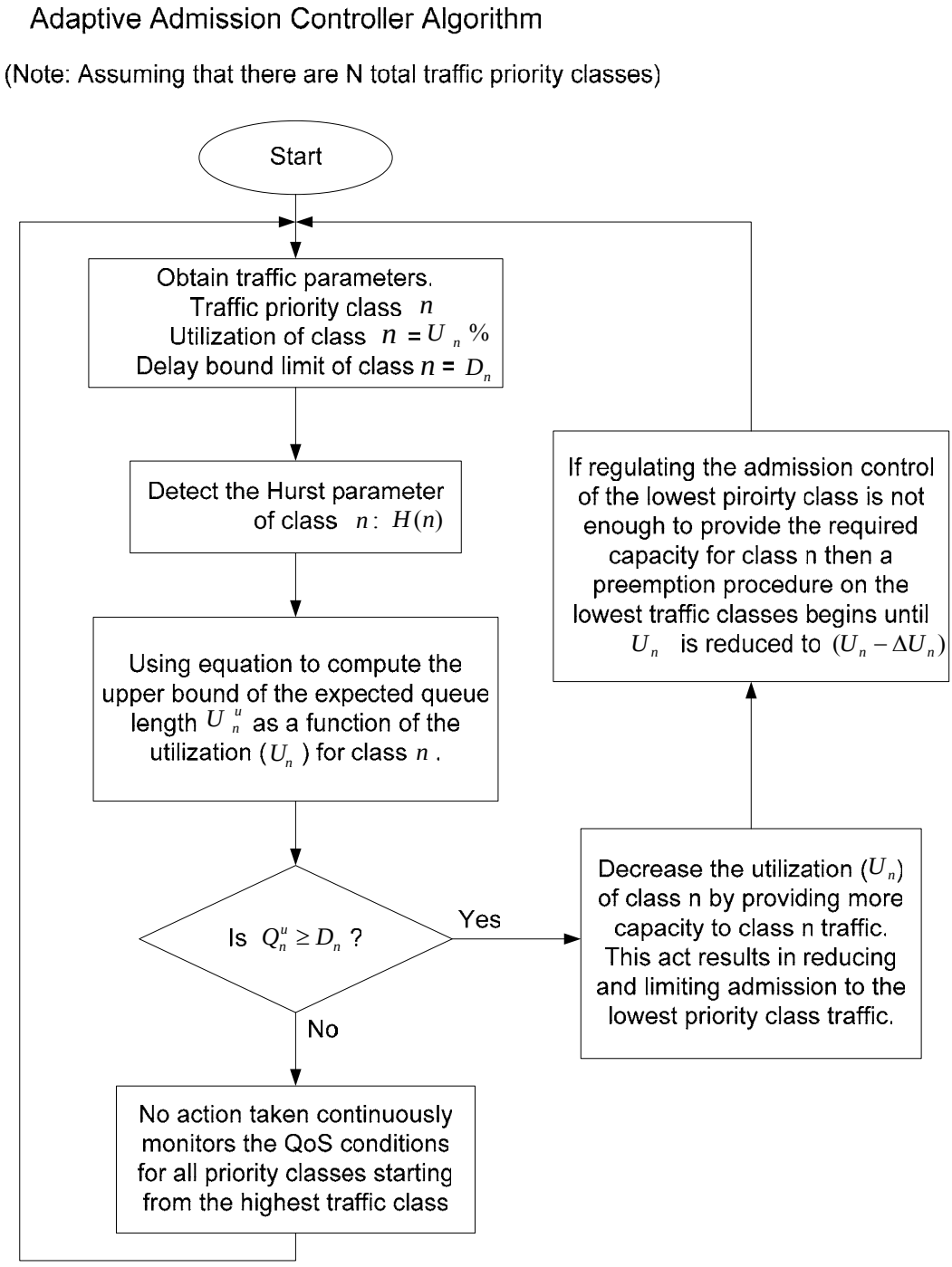


(d) Class 4 Expected Queue Length

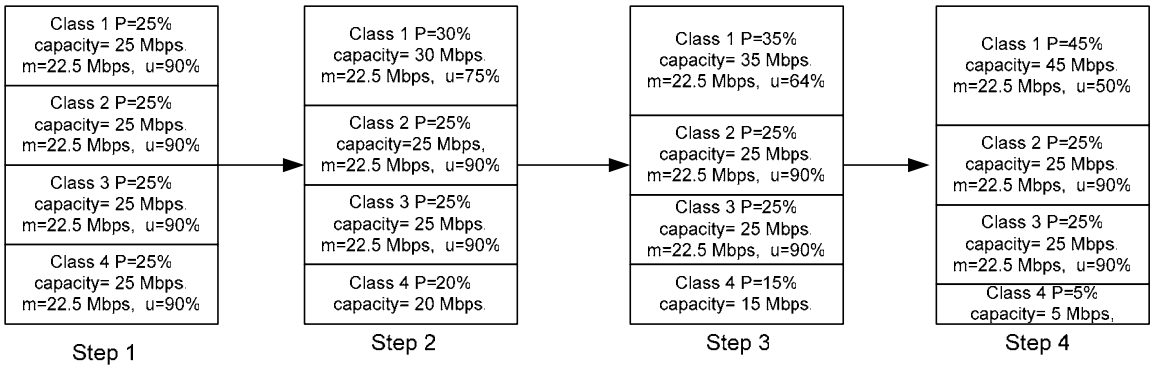
Figure 4-4. Expected Queue Length from Class 1 to Class 4.

In order to improve the performance of class 1 and class 2 at burstiness time, we can increase class 4's utilization at the server and decrease the utilization to class 1

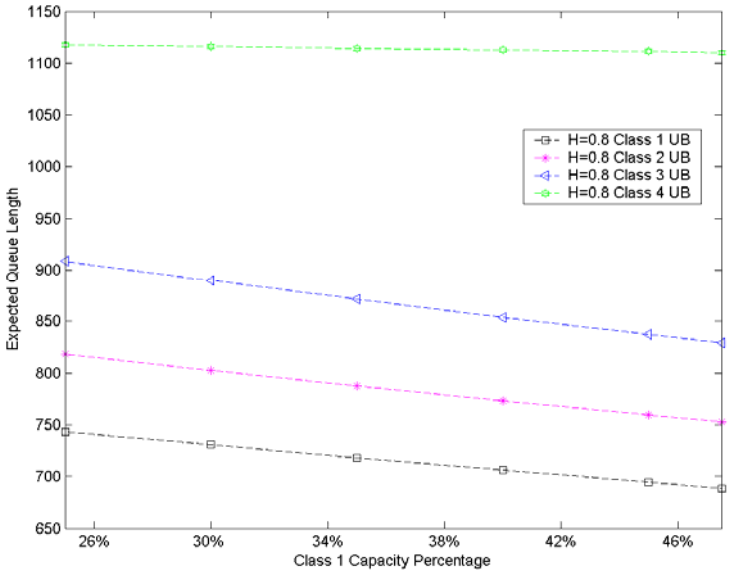
as well. For example, in Figure 4-4 (a), the expected queue length is around 750 for class 1 at 90% multiplexed traffic utilization. Suppose the safety queue length is 700. That is, the packet is queued larger than 750, it will be discarded or else it will violate the local deadline. Larger queue length will cause larger packet loss rate and longer waiting time at each hop. A novel adaptive admission controller algorithm is proposed as shown in Figure 4-5 (a). Figure 4-5 (b) shows how to reduce and limit admission to the lowest priority class traffic step by step. Initially, the four classes have the same capacity at the server. In order to achieve a queue length of less than 700 and keep the combined traffic utilization at 90%, the admission control just decreases the class 1 utilization by increasing server capacity percentage of class 1 at each hop, and increases the utilization of class 4 at the same time. The total server capacity is fixed. And class 2 and 3 keep the same utilization. The class 1's queue length at the hop will be decreased. If the queue length is still larger than 700, the admission controller will continue to decreasing class 1's utilization, and increasing class 4 utilization. Figure 4-5 (c) shows the performance improvement. The queue length of class 2 and class 3 decreased along with the class 1's. The adaptive admission controller algorithm decreases all the queue length of class 1, class 2, and class 3. In addition, class 4's queue length still keeps the same as before. The adaptive controller algorithm can improve the performance around 8 percent.



(a) Flow Chart of the Adaptive Admission Controller Algorithm.



(b) Each Step for Adaptive Admission Controller



(c) Performance Improvement.

Figure 4-5. An Adaptive Admission Controller Algorithm.

The benefit of using the novel adaptive admission controller algorithms is: It improves the performance of class 1, class 2 and class3 about 8%. It is shorter the queue length and waiting time of class 1, class 2 and class 3.

CHAPTER V

CONCLUSION AND FUTURE WORK

In this thesis, a novel adaptive admission controller is proposed based on the arrival rate and the service rate for multiple hop queuing systems with self-similar. The derivations begin with the mathematical modeling of a single queue based on [25], and then the derivations are extended to end-to-end differentiated service networks with self-similar traffic. The upper and lower bound of queue length at each hop is derived. In addition, the analytical model is also applied to the analysis of traffic flow effects on queue length. The results give an insight into the performance gained in queue length at each hop. After the application of *Little's* theorem to queue length, the mean delay at each hop can be obtained. These parameters, queue length and mean delay, are parameters related to the concept of traffic engineering. In addition, a novel admission control mechanism is provided to improve QoS reliability performance for the higher classes in DiffServ networks.

Based on the observations of the simulation results, the following conclusion can be drawn:

(1) Each hop's queue length is related to the arrival flow's mean, the variance of input, the utilization at each server, the utilization of multiplexed traffic, and

the Hurst parameter. The downstream hops' queuing length has the similar property and value as the first hop. It is also dependent on the above parameters.

(2) The class 1 and class 2 have a smaller expected queuing length and waiting time compared to class 3 and class 4. At fixed multiplexed traffic utilization, larger Hurst parameter will increase the expected queue length and waiting time at each hop.

(3) Each class' expected queue length will increase with the increasing of combined traffic utilization at each hop. The larger the utilization of the total traffic, the larger the queue length and the longer the waiting time.

(4) The upper and lower bound of class 1 and class 2 are tighter closer than the bounds of class 3 and class 4. This demonstrates that the QoS services that will be provided by the DiffServ schedulers are more consistent and stable for the higher priority classes. This is a very desirable feature for DiffServ networks.

(5) A novel adaptive admission controller algorithm is developed to provide the upper classes of the DiffServ networks the required QoS. The adaptive admission controller algorithm can provide a guaranteed QoS performance for the higher priority classes, as long as the class 1 QoS requests do not exceed the comprehensive network resources.

In the future, the fluid algorithm model can be extended to the substantial buffering model [30], as the fluid algorithm model is used when the arrival rate is exactly known.

REFERENCES

[1] P. Abry and D. Veitch, "Wavelet analysis of long-range dependent traffic," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2-15, Jan. 1998.

[2] J. Adibi, W-M. Shen, E. Noorbakhsh, "Self-similarity for data mining and predictive modeling: a case study for network data," *6th Pacific-Asia Conference*, PAKDD 2002, pp. 210-217, May 2002.

[3] S. Basu, A. Mukherjee, "Time series models for internet traffic," *Proceedings, IEEE Infocom Conf.* vol.2, pp. 611-620, Mar. 1996.

[4] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and W. Weiss, " An Architecture for Differentiated Services," RFC 2475, Dec 1998

[5] D. A. Burn, "Simulation of stationary time series," *Proceedings of the 19th Simulation Conference*, pp. 289-294, Dec. 1987.

[6] Z. Cao, Z. Wang, and E.Zegura, "Rainbow fair queueing: Fair band-width sharing without per-flow state," i*n Proc, IEEE INFOCOM 2000*, Tel Aviv, Isreal, pp.922-931, Mar. 2000.

[7] J.-M.Chung, Lecture Notes of ECEN 5553 Telecommunication System I .

[8] M. E. Crovella, "Self-similarity in WWW traffic: evidence and possible causes," *IEEE Trans.Networking*, vol. 5, pp. 835-845, Dec. 1997.

[9] A. Erramilli, M. Roughan, D. Veitch, W. Willinger, "Self-similar traffic and network dynamics," *Proc. of IEEE*, vol. 90, pp. 800-819, May 2002.

[10] J. Feder, *Fractals*. New York: Plenum Press, 1989.

[11] F. H. P. Fitzek, M. Reisslien, "MPEG-4 and H.263 video traces for network performance evaluation," *IEEE Network*, vol. 15, pp. 40-54, Nov. 2001.

[12] S. Giordano, S. Miduri, M. Pagano, F. Russo, S. Tartarelli, "A wavelet-based approach to the estimation of the Hurst parameter for self-similar data," *Digital Signal Processing Proceedings, 1997 13th International Conference on,* vol. 2, pp. 479 – 482, July 2-4 1997.

[13] E.Knightly, “Enforceable quality of service guarantees for bursty traffic streams,” in *Proc*. *IEEE INFOCOM’98,* San Francisco, CA, pp635-642, Mar.1998.

[14] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, “On the self-similar nature of Ethernet traffic (extended version),” *IEEE/ACM Trans*. *Networking,* vol. 2, pp. 1-15, Feb. 1994.

[15] C. Li and E. Knightly, "Coordinated Multihop Scheduling: A Framework for End-to-End Services," *IEEE/ACM Transactions on Networking,* vol. 10, no.6, pp. 776-789, Dec. 2002.

[16] C.Li, E.W.Knightly, “Schedulability Criterion and performance analysis of coordinated schedulers,” *Trans. on IEEE/ACM Networkings,* vol. 13, no.2, Apr. 2005.

[17] C. Metz, “IP QoS: Traveling in First Class on the Internet”, *IEEE Internet Computing*, vol. 3, no. 2, pp. 84-88, March-April 1999.

[18] P. R. Morin, “The impact of self-similarity on network performance analysis,” Ph.D. dissertation, Carleton Univ., Dec. 1995.

[19] T.Nandagopal, N. Venkitaraman, R. Sivakumar, and V.Bharghavan, “Delay differentiation and adaptation in core-stateless networks,” in *Proc. IEEE INFOCOM 2000*, Tel Aviv, Israel, pp. 792-799, Mar. 2000.

[20] Nichols, K., Blake, S., Baker, F. and D. Black, “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers,” RFC 2474, Dec. 1998

[21] I. Norros, “On the Use of Fractional Brownian Motion in the Theory of Connectionless Networks,” *IEEE J. Sel. Areas Commun*., vol. 13, no.6, pp. 953-962, Aug. 1995.

[22] H.Ohsaki, M.Murata, “Steady State Analysis of the RED Gateway: Stability, Transient Behavior, and Parameter Setting, H.Ohsaki, M. Murata,” *IEICE Trans. Comm*., vol. E85-B, no. 1, Jan. 2002.

[23] V. Paxson, S. Floyd, “Wide area traffic, the failure of Poisson modeling,” *IEEE/ACM Trans. Networking*, vol. 3, pp. 226-244, June 1995.

[24] Z. Quan, J.-M. Chung, “A Novel Analysis of Queue Length in Differentiated Services Networks with Self-similar Arrival Processes,” *Proceedings of the 45th IEEE International Midwest Symposium on Circuits and Systems IEEE MWSCAS'02* , vol. 2, pp. III-85-88, Aug. 2002.

[25] Z. Quan, J.-M. Chung, "Queue Length Analysis of Non-Preemptive DiffServ Networks," AEÜ, *International Journal of Electronics and Communications*, vol.57, is. 5, pp. 338-340, Sept. 2003.

[26] Z. Quan, J.-M. Chung, "Impact of Self-similarity on Performance Evaluation in Differential Service Networks," *Proceedings of the 45th IEEE International Midwest Symposium on Circuits and Systems IEEE MWSCAS'02* , vol. 2, pp. 326-329, Aug. 2002.

[27] Z. Quan, J.-M. Chung, "Priority Queueing Anaysis of Self-similar Trafficin High-Speed Networks," *Proceedings of the IEEE International Conference on Communications IEEE ICC'03,* vol. 3, pp. 1606-1610, May 2003.

[28] Z. Sahinoglu, S. Tekinay, "On multimedia networks: self-similar traffic and network performance," *IEEE Comm*, pp. 48-52, Jan. 1999.

[29] W. Stallings, *High-Speed Network and Internets*, 2nd ed., New Jersey: Prentice Hall, 2001.

[30] I. Stoica, S. Shenker, and H. Zhang, "Core-stateless fair queueing: A scalable architecture to approximate fair bandwidth allocations in high speed networks," in *Proc. ACM SIGCOMM '98,* Vancouver, BC, Canada, pp.118-130, Sept.1998.

[31] J. Wen, X. Liu, "The Design of QoS Guarantee Network Subsystem," *ACM SIGOPS Operating System Review,* vol. 1, no. 1, pp. 81-87, Jan. 2002.

[32] G. W. Wornell, "Wavelet-based representations for the 1/f family of fractal processes," *Proceedings of the IEEE*, vol. 81, pp. 1428-1450, Oct. 1993.

[33] "Estimation of the Hurst Parameter of Long-range dependent time series", Feb. 1996.

[34] I. Yeom, A.L.N.Reddy, "Modeling TCP behavior in a Differentiated Services Network," *IEEE/ACM Trans. on Networking*, vol. 9, no.1, Feb. 2001.

[35] C. You, "Time series models for internet data traffic," Thesis, University of Massachusetts Lowell, 1999.

[36] http://users.rowan.edu/~polikar/WAVELETS/WTpart4.html Date updated: Jan. 12, 2001. Data accessed: April 12, 2005.

[37] http://www.cs.kent.ac.uk/people/staff/pfl/presentations/longrange/sld017.htm Date updated: Mar. 9, 2001. Data accessed: Apr 12, 2005.

[38] http://cin.ufpe.br/~cak/publications/thesis_proposal.pdf. Date updated: Data accessed: Apr. 12, 2005.

APPENDIX ACRONYMS

BA	Behavioral Aggregate
CSFQ	Core Stateless Fair Queuing
DiffServ(DS)	Differentiated Service
DSCP	Differentiated Service Code Point
DWT	Discrete Wavelet Transform
ECN	Explicit Congestion Notification
FGN	Fractional Gaussian Noise
FIFO	First in first out
FQ	Fair Queuing
HOL	Head of Line
IP	Internet Protocol
LRD	Long Range Dependence
MDS	Mean Delay Scheduler
PHB	Per-Hop Behavior
QoS	Quality of Service
RED	Random Early Detect
RMD	Random Midpoint Displacement
ToS	Type of Service
WFQ	Weighted Fair Queuing

VITA

YUE WANG

Candidate for the Degree of

Master of Science

Thesis: QUEUE LENGTH ANALYSIS OF END-TO-END DIFFERENTIATED SERVICE NETWORKS WITH SELF-SIMILAR TRAFFIC

Major Field: Computer Science

Biographical:

Personal Data: Born in Beijing, China, July 1, 1974, daughter of Mr. Wenjun Wang and Mrs. Xueshuang Song.

Education: Received Bachelor of Science degree in Chemistry from Beijing United University Art and Science College, China, in July 1997; completed the requirements for Master of Science degree in Computer Science at the Computer Science Department at Oklahoma State University in July 2005.

Experience: Teaching Assistant, Computer Science Department, Oklahoma State University, January 2004 to May 2005; Research Assistant, Advanced Communications System Engineering Laboratory (ACSEL) and Oklahoma Communication Laboratory for Networking and Bioengineering (OCLNB), Oklahoma State University, January 2004 to July 2005.

ABSTRACT

Name: YUE WANG Date of Degree: July, 2005

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: QUEUE LENGTH ANALYSIS OF END-TO-END DIFFERENTIATED SERVICE NETWORKS WITH SELF-SIMILAR TRAFFIC

Pages in Study: 74 Candidate for the Degree of Master of Science

Major Field: Computer Science

Scope and Method of Study: This thesis focuses on the analysis of queue length boundaries in end-to-end self-similar networks with differentiated service. First, the Hurst parameter was calculated by three methods. Secondly, queue length boundaries were estimated at a single hop with differentiated service under self-similar traffic. Finally, the derivations are extended to end-to-end differentiated service networks with self-similar traffic.

Findings and Conclusions: Quality-of-Service (QoS) is a key issue in networks. Improving the performance with guaranteed QoS is one of the major problems. It is well known that traditional analytical methods of queuing systems are based on Poisson and Poisson-based stochastic processes. Unfortunately, these methods, like Jackson theorem, are not applicable in high speed broadband networks. In this thesis, a novel analytical model is proposed based on the arrival rate and the service rate for multiple hops queuing systems. Then the mathematical derivations are extended to end-to-end differentiated service networks with self-similar traffic. The upper and lower bound of the queue length at each hop is derived. The results illustrate the performance gain in queue length at each hop. Finally, a novel adaptive admission controller algorithm is proposed based on the arrival rate and the service rate for multiple hop queuing systems with self-similar network traffic. The adaptive admission controller algorithm can provide a guaranteed QoS performance for the higher priority classes, as long as the highest class QoS requests do not exceed the comprehensive network resources.

ADVISER'S APPROVAL: Dr. G. E. Hedrick