Fall 1-31-2002

Renegotiation based dynamic bandwidth allocation for selfsimilar VBR traffic

Zafer Sahinoglu
New Jersey Institute of Technology

ABSTRACT

RENEGOTIATION BASED DYNAMIC BANDWIDTH ALLOCATION FOR SELF-SIMILAR VBR TRAFFIC

by
Zafer Sahinoglu

The provision of QoS to applications traffic depends heavily on how different traffic types are categorized and classified, and how the prioritization of these applications are managed. Bandwidth is the most scarce network resource. Therefore, there is a need for a method or system that distributes an available bandwidth in a network among different applications in such a way that each class or type of traffic receives their constraint QoS requirements.

In this dissertation, a new renegotiation based dynamic resource allocation method for variable bit rate (VBR) traffic is presented. First, pros and cons of available off-line methods that are used to estimate selfsimilarity level (represented by Hurst parameter) of a VBR traffic trace are empirically investigated, and criteria to select measurement parameters for online resource management are developed. It is shown that wavelet analysis based methods are the strongest tools in estimation of Hurst parameter with their low computational complexities, compared to the variance-time method and R/S pox plot. Therefore, a temporal energy distribution of a traffic data arrival counting process among different frequency sub-bands is considered as a traffic descriptor, and then a robust traffic rate predictor is developed by using the Haar wavelet analysis. The empirical results show that the new on-line dynamic bandwidth allocation scheme for VBR traffic is superior to traditional dynamic bandwidth allocation methods that are

based on adaptive algorithms such as Least Mean Square, Recursive Least Square, and Mean Square Error etc. in terms of high utilization and low queuing delay. Also a method is developed to minimize the number of bandwidth renegotiations to decrease signaling costs on traffic schedulers (e.g. WFQ) and networks (e.g. ATM). It is also quantified that the introduced renegotiation based bandwidth management scheme decreases heavy-tailedness of queue size distributions, which is an inherent impact of traffic self-similarity.

The new design increases the achieved utilization levels in the literature, provisions given queue size constraints and minimizes the number of renegotiations simultaneously. This renegotiation-based design is online and practically embeddable into QoS management blocks, edge routers and Digital Subscriber Lines Access Multiplexers (DSLAM) and rate adaptive DSL modems.

RENEGOTIATION BASED DYNAMIC BANDWIDTH ALLOCATION FOR SELFSIMILAR VBR TRAFFIC

by
Zafer Sahinoglu

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Electrical Engineering

Department of Electrical and Computer Engineering

January 2002

APPROVAL PAGE

RENEGOTIATION BASED DYNAMIC BANDWIDTH ALLOCATION FOR SELFSIMILAR VBR TRAFFIC

Zafer Sahinoglu

Dr. Sirin Tekinay, Dissertation Advisor — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

Dr. Ali N. Akansu, Committee Member — Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Constantin Manikopulos, Committee Member — Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Jay Jorgenson, Committee Member — Date
Professor of Mathematics, the City College of New York

Dr. Nirwan Ansari, Committee Member — Date
Professor of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Zafer Sahinoglu

Degree: Doctor of Philosophy in Electrical Engineering

Date: January 2002

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ 2002
- Master of Science in Biomedical Engineering
  New Jersey Institute of Technology, Newark, NJ 1998
- Bachelor of Science in Electrical and Electronics Engineering
  Gazi University, Ankara, Turkey, 1994

Major: Electronics Engineer

Presentations, Publications and Patents:

Z. Sahinoglu, S. Tekinay, "*Selfsimilarity and Its Effects on Network Performance,*" IEEE Comm. Magazine, January 1999.

Z. Sahinoglu, S. Tekinay, *"On the Optimal Detection and Measurement of Self-similarity in Network Traffic: A Comparative Study of Proposed Techniques,"* in the Proc of VTC'99, Vancouver, June 1999.

Z. Sahinoglu, S. Tekinay, *"Multi-resolution and Burstiness Analysis of Traffic Traces,"* In the Proc of WCNC'99, New Orleans, September 1999.

Z. Sahinoglu, S. Tekinay *"Efficient Parameter Selection for Use of Self-similarity in Real Time Resource Management,"* submitted to ICC'02, New York City, April 2002.

Z. Sahinoglu, S.Tekinay, *"A Novel Adaptive Bandwidth Allocation: Wavelet Decomposed Signal Energy Approach,"* GLOBECOM'01, San Antonio, November 2001.

Z. Sahinoglu, F. Porikli, *"Minimization of Bandwidth Renegotiations in ATM Networks,"* ICCCN'01, October 2001.

Z. Sahinoglu, F. Porikli, S. Tekinay, *"An Online Renegotiation Based Asynchronous Dynamic Bandwidth Allocation to Provision QoS Constraints for VBR Traffic,"* submitted to IEEE Communications Letters.

Z. Sahinoglu, S. Tekinay, *"Real Time Measurement of Self-similarity in Resource Renegotiable Networks,"* submitted to IEEE Communications Letters.

Z. Sahinoglu, V. Bhagavath, "*A Method for Automated Selection of Architectural Modes for High-Speed xDSL Access IP Network Services", EP1113621A*

S. Abbasi, Z. Sahinoglu, V. Bhagavath, J. Odondi, "*Methods and Systems for Improving Data Transmission Rates Having Adaptive Protocols*" filed in August 99 at AT&T Shannon Labs.

Dedicated to the one who gives me comprehension ,
and to the teacher, then to my parents,
and to all the other precious people whose company I treasure

ACKNOWLEDGMENT

I would like to express my deepest appreciation to Dr. Sirin Tekinay for her invaluable guidance, providing countless resources, insight and intuition since the beginning, and for her patience. I am grateful to Dr. Vijay Bhagavath for his guidance and support, and providing me with an environment to gain practical experience during my internship at AT&T Shannon Labs. I would like to thank Dr. Walter Willinger as well, for his special comments and guidance on traffic selfsimilarity.

Special thanks to Dr. Ali N. Akansu, Dr. Nirwan Ansari, Dr. Constantin Manikopulos, and Dr. Jay Jorgenson for serving as members of the committee.

I would like to thank my friends and colleagues at New Jersey Center for Multimedia Research (NJCMR). Special thanks go to Burak, Sebnem, Taha from NJCMR, and Fatih and Kadir from Mitsubishi Electric Research Labs (MERL).

Finally, I would like to thank my parents who made contributions with their hearts and the Ministry of Education of Turkey to give me an opportunity to pursue my PhD study in the US.

GLOSSARY AND TERMS

AR -	Auto-Regressive
ATM -	Asynchronous Transfer Mode
BDU -	Bandwidth Decision Unit
CLP -	Cell Loss Probability
CLR -	Cell Loss ratio
DBAU -	Dynamic Bandwidth Allocation Unit
DBIND -	Deterministic Bounding Interval Dependent Traffic Descriptor
DSA -	Dynamic Search Algorithm
DSL -	Digital Subscriber Lines
DWT -	Discrete Wavelet Transform
GOP -	Group of Pictures
IRI -	Inter-renegotiation Interval
ITU -	International Telecommunications Union
LAN -	Local Area Network
LRD -	Long Range Dependence
MBAC -	Multi-scale Boundary Admission Control
MSE -	Mean Square Error
MPEG4 -	Motion Pictures Engineering Group-4
MRA -	Multi-resolution Analysis
PLP -	Packet Loss Probability
PT -	Processing Time
R/S -	Rescaled Adjusted Statistics
RADSL -	Rate Adaptive Digital Subscriber Line
RCBR -	Renegotiated Constant Bit Rate
RCU -	Renegotiation Control Unit
RDBA -	Renegotiated Dynamic Bandwidth Allocation
RED-VBR -	Renegotiated Deterministic Variable Bit Rate Service
RSS -	Renegotiation Step Size
SAC -	Selective Aggressiveness Control
SBA -	Static Bandwidth Allocation
S/P -	Serial-to-Parallel Converter
VBR -	Variable Bit Rate
VC -	Virtual Connection
WAN -	Wide Area Network
WFQ -	Weighted Fair Queuing
WRR -	Weighted Round Robin
WS -	Window Size in traffic analysis
WWW -	World Wide Web

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER I

INTRODUCTION

Does any artist paint for the sake of the picture itself, without the hope of offering some good? No, but for the sake of the viewers and the young who will be drawn by it and freed from cares. Or does any potter hastily throw a pot or a bowl without any thought of what it will hold? Does any calligrapher write for the script alone without any regard for the reader?

Rumi

Broadband technology is an always-open gateway to a new world of Internet services delivered at lightning-fast speeds to homes, offices and businesses. High speed connections (e.g. DSL, Cable Modem, T1) open up a new world of multimedia applications each with different traffic characteristics. Accurate modeling of the traffic offered to the network or a component of the network will always be critical to provide high quality of service (QoS) to the applications. The main objective in telecommunications network engineering is to have as many happy users as possible. In other words, a network engineer has to resolve the tradeoff between capacity and quality of service (QoS) requirements. Realistic modeling of the offered traffic characteristics, therefore, is the first step in optimizing resource allocation algorithms such that provision of services complies with the quality of service constraints while maintaining maximum capacity. In order to maintain the QoS provided to applications traffic, it is necessary to regulate network resources dynamically depending on traffic descriptors (e.g. PLP, CBR, delay, delay variation) and network characteristics (e.g. utilization).

In this dissertation, a complete solution is presented on how to dynamically manage available resources for VBR traffic in networks which consist of network units capable of renegotiating resources such as ATM switch complexes, Rate Adaptive Digital

Subscriber Line (RADSL) modems, and QoS routers. First, traffic bit arrival information per unit time and time-variant energy distribution of a bit counting process are defined as traffic descriptors, and utilization as a network characteristic. Second, traffic selfsimilarity is investigated, and drawbacks of selfsimilarity measurements in their suitability for on-line dynamic resource management algorithms are explained. Methods are suggested for proper parameter selection in the estimate of self-similarity level of a traffic to ease linking of self-similarity measurements into on-line resource management systems. Third, a bandwidth predictor that uses wavelet analysis of incoming traffic bit counting process are designed. The output of the bandwidth predictor is optimized by designing a bandwidth renegotiator unit which decides when and at what amount to decrease or increase the bandwidth allocated for an applications traffic. The invented renegotiator unit minimizes the number of bandwidth renegotiations in a network.

The contents of the sections in this chapter are as follows: Section I.1 gives the statement of the problem. It starts with the definition of self-similarity, explains difficulties in allocating network resources for VBR traffic sources exhibiting selfsimilarity, and clearly states that asynchronous renegotiated resource management is a proper solution for QoS provisioning for a selfsimilar VBR traffic. Section I.2 presents prior art studies on selfsimilar traffic, impacts of selfsimilarity on network performance, and on dynamic resource allocation to such traffic sources. These studies include congestion control strategies, bandwidth predictors and resource renegotiation methods. Section I.3 explains the dynamic resource management approach presented in this dissertation to deal with VBR traffic.

1.1. Statement of the Problem

A number of studies have shown that for both local and wide area network traffic, the distribution of packet inter-arrivals clearly differs from exponential [1, 2, 3]. Furthermore, investigations of high-resolution Ethernet Local Area Network (LAN) traces [4, 5], Wide Area Network (WAN) traffic [6] and VBR video traffic [7, 8, 9] have demonstrated that these traffic types exhibit self-similarity. In the case of stochastic objects like time series, self-similarity is used in the distributional sense: when viewed at varying time scales, the object's relational structure remains unchanged. As a result, such a time series exhibits bursts at a wide range of time scales. Self-similar traffic behaves very differently from that predicted by traditional packet traffic models. For example in [4], it is shown that the generally accepted argument for the "Poisson-like" nature of aggregate traffic that aggregate traffic smoothes out as the number of aggregated traffic sources increases does not reflect the reality. In fact, the burstiness of such traffic, defined by the variance-to-mean ratio, peak rate and selfsimilarity level, typically intensifies as the number of active traffic sources increases. There are different studies advocating both the significance of impact of self-similarity on network performance [10-13] and the irrelevance of the need for capturing self-similarity in traffic modeling [14]. The questions that arise here are how prevalent such traffic patterns are and under what conditions performance analysis is critically dependent on taking self-similarity into account.

If sources exhibiting bursts at multiple time scales are allowed only a single service rate to describe their behavior, they are faced with a series of poor choices. If the service rate is chosen close to the long term average rate in order to maximize the

statistical multiplexing gain in the network, during sustained peaks, the source buffer fills up at the peak rate and is drained at the service rate. If the peak rate is much higher than the average rate, either the data buffer has to be very large, or the loss rate will be unacceptably high. Provisioning of large buffers limits the loss rate. On the other hand, it causes long delays. In recent years, as broadband multimedia services became popular they necessitated new traffic models with self-similar characteristics.

A classical control approach for dynamic resource management exists in monitoring the level of unused resources and in feeding back the measured levels to a controller that adjusts the input traffic rates. The input rate control approach is known in literature as rate-based [15]. Many rate-based algorithms can be found in literature. However, none of these are completely satisfactory either for their complexity or for lack of stability properties, as reported in [16]. In fact, due to transmission and propagation delay most algorithms exhibit persistent oscillations. A static resource allocation method for a VBR source or a self-similar traffic is away from the maintaining the required QoS by those applications traffic, but a dynamic resource allocation. However, allocating resources dynamically for a real-time VBR stream is a challenging task. First, it necessitates a design of a proper traffic predictor. A one-step-ahead predictor produces as many outputs as the number of traffic information samples are. Therefore, an additional mechanism is required to process the predictor output, and to decide if a bandwidth renegotiation is needed, and at what amount it can be provisioned. A renegotiation control unit takes some network or application dependent cost constraints such as a cost for each bandwidth unit and a cost for each renegotiation as inputs and returns a decision. There are various bandwidth predictors and

renegotiation methods available in the literature. However, either most them are designed for off-line systems, or they have high complexity and computational overload that are not proper for on-line QoS provisioning. Therefore, there is a need for a robust bandwidth predictor and a renegotiator especially in provisioning of QoS for bursty VBR streams.

1.2. Prior Art and Motivation

1.2.1. Self-Similarity and Its Impacts on Network Performance

This sub-section presents a summary of the survey study about self-similarity and network performance, and possible solutions offered to deal with this phenomenon. Understanding the nature of network traffic is critical in order to properly design and implement both wired and wireless networks. Since the paper published in 1993 by W. Leland et. al.[4], there appeared many number of studies which reported a bursty traffic pattern on different time scales of multimedia networks [11-13]. What exactly burstiness means in terms of measurable traffic statistics has been a critical problem. In the literature, traffic that is bursty on many or all time scales is described as self-similar. When these traffics are viewed at varying scales, the data's correlation structure remains unchanged. Also, it is possible to see studies pointing out the impact of traffic self-similarity on network performance that is basically expressed in terms of quality of service (QoS) parameters like packet loss ratio (PLR), and delay and network resources such as bandwidth and buffer sizes.

Research done by M. E. Crovella et.al. [17] revealed that the traffic generated by the worldwide web (WWW) transfers showed self-similar characteristics. Comparing the

distributions of ON and OFF times, they found that the ON time distribution was heavier tailed than the OFF time distribution. The distribution of file sizes in the WEB might be the primary determiner of Web traffic self-similarity. In fact, the work presented by K. Park etc. al. [10] showed that the transfer of files whose sizes were drawn from a heavy-tailed distribution was sufficient to generate self-similarity in network traffic. The ON and OFF periods do not need to have the same distribution. These results suggest that the self-similarity of Web traffic is not a machine-induced artifact; in particular, changes in protocol processing and document display are not likely to remove the self-similarity of Web traffic [17].

In a realistic client/server network environment, the degree to which file sizes are heavy tailed can directly determine the degree of traffic self-similarity at the link level [10-12]. This causal relation is proven to be robust with respect to changes in network resources (bottleneck bandwidth and buffer capacity), network topology, the influence of cross traffic, and the distribution of packet inter-arrival times. Specifically, measuring self-similarity via the Hurst parameter H and the file size distribution by its power law exponent α, it has been shown that there is a linear relationship between H and α over a wide range of network conditions. One practical effect of self-similarity is that the buffers needed at switches and multiplexers must be bigger than those predicted by traditional queuing analysis and simulations. These larger buffers create greater delays in individual streams that are originally anticipated [12, 13]. The delay-bandwidth product problem arising out of high-bandwidth networks and QoS issues stemming from support of real-time multimedia communication have added further complexities to the problem of optimizing the performance. How much the self-similarity affects the network

performance is modulated by the protocols acting at the transport/network layer. An exponential trade-off relationship is observed between queuing delay & packet loss rate [11].

It is assured that linear increase in buffer sizes will produce nearly exponential decreases in packet loss and that an increase in buffer size will result in a proportional increase in the effective use of transmission capacity. With self-similar traffic, these assumptions do not hold. As can be seen from Fig.1.1, as H approaches 1.0, the queue size diverges to infinity at lower utilization levels than when H is closer to 0.5. The decrease in packet loss with buffer size is far less than expected. In other words, the buffer requirements begin to explode at lower levels of utilization for higher degrees of long-range dependence (Higher values of H).

Heyman etc. al. [14] showed that for sources with large Hurst parameter, Markov chain models estimated the buffer occupancy well when the buffer sizes were not too

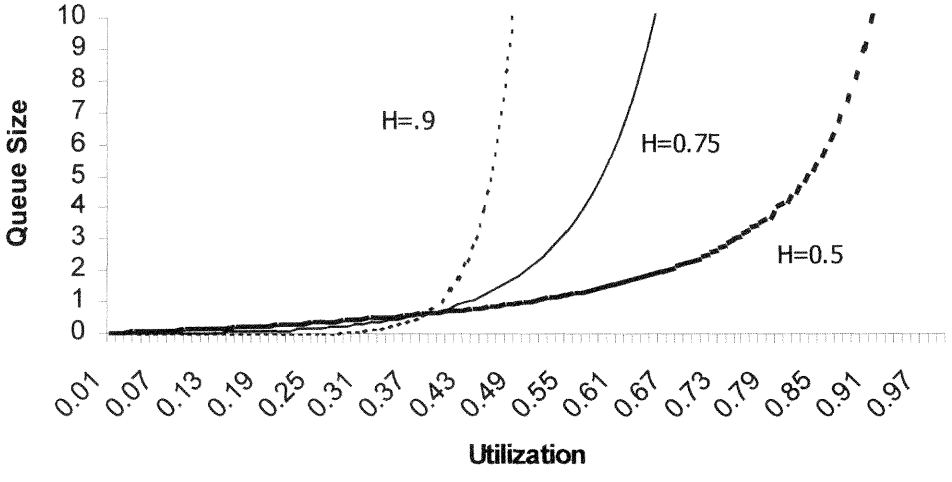


Figure 1.1 Queue size- utilization trade-off as the self-similarity level, which is measured by Hurst parameter H, changes [18].

large (no larger than 10ms for a single source) but these models might not estimate the cell-loss rate and mean buffer size accurately for larger buffers. Also, [11] has shown that queuing delay exhibited a super-linear dependence on self-similarity when buffer capacity was large. The queue length distribution decayed more slowly for long-range dependent sources than short-range dependent sources. Moreover, scale invariant burstiness implies the existence of concentrated periods of high activity at a wide range of time scales that adversely affects congestion control and it is an important correlation structure that may be exploitable for congestion control purposes [13]. Network performance as captured by throughput, packet loss rate and packet retransmission rate degrades gradually with increasing heavy-tailedness. The degree to which heavy-tailedness affects self-similarity is determined by how well congestion control is able to shape its source traffic into an on-average constant output stream while conserving flow [11].

A dynamic congestion control strategy is difficult to implement. Such a strategy is based on measurement of recent traffic and can fail utterly to adapt to rapidly changing conditions. Also, congestion prevention by appropriate sizing of switches and multiplexers is difficult because data network traffic doesn't exhibit a predictable level of busy traffic period; patterns can change over a period of days, weeks or months and congestion occur unexpectedly with dramatic intensity. On the other hand, a predictive congestion control was studied for improving network performance by Tsunyi Tuan et. al. [12]. In their algorithm, information about the future is utilized to make traffic control decisions. They called this Selective Aggressiveness Control (SAC) and it is aimed to be robust, efficient and portable such that it can be easily incorporated into existing

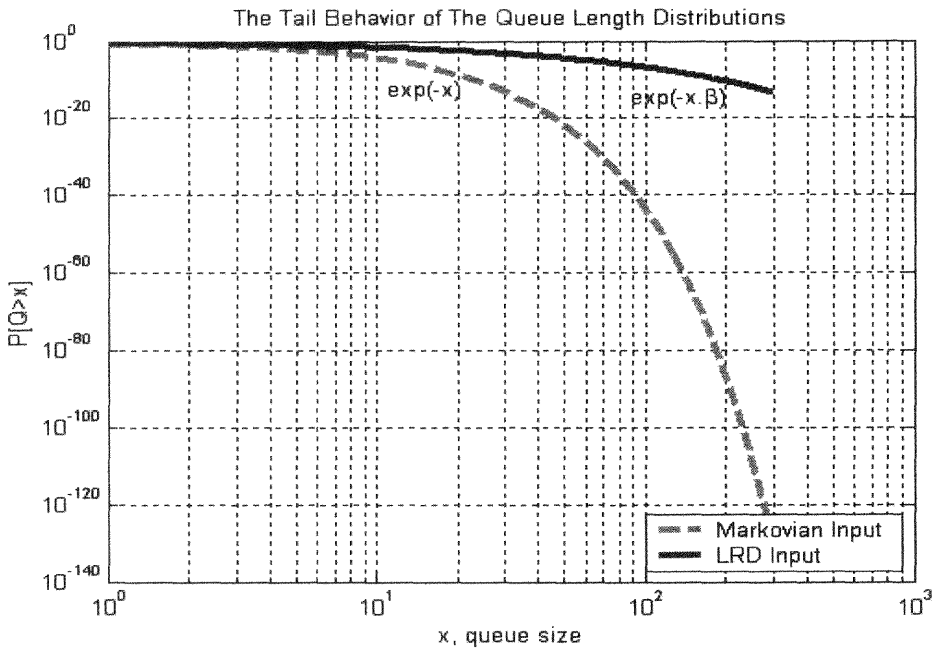


Figure 1.2 Tail behavior of the queue length distributions for Markovian and LRD inputs to an infinite capacity queue in a single server Note: $\beta=2-2H\in(0,1)$

congestion control schemes. SAC tries to aggressively soak up bandwidth if it predicts the future network state to be "idle", adjusting the level of aggressiveness as a function of the predicted idleness and its confidence. They showed that the performance gain due to SAC is higher as the more self-similar the network traffic is. Although, in real life, the perfect prediction of the future traffic congestion is not possible, SAC achieves the highest throughput with perfect future information among other congestion control algorithms such as the generic feedback congestion control.

A crucial performance indicator exhibiting a more sensitive dependence on self-similarity is mean queue length and this concurs with the observation that queue length distribution under self-similar traffic decays more slowly than with Poisson sources (Fig. 1.2). Increasing network resources such as link bandwidth & buffer space results in a

super linear improvement in performance. Increasing link bandwidth, given a large buffer capacity, has the effect of decreasing queuing delay much more drastically. Therefore, high-bandwidth communication links (for multimedia network applications) should be employed to alleviate the exponential trade-off relation between queuing delay & packet loss (throughput) for supporting QoS sensitive traffic.

Based on the survey results, it is asserted that dynamic bandwidth allocation with minimum number of renegotiations is a key solution to deal with self-similar traffic since static resource allocation preserves the heavy-tailed queue sizes. This dissertation clearly demonstrates the improvement in the queue size performance in a single server environment after deployment of the invented dynamic bandwidth allocation method compared with the static ones.

1.2.2. Bandwidth Prediction and Renegotiation

An efficient resource allocation comprises determining optimal buffer sizes, assigning bandwidth and other resources either statically or dynamically in order to get the desired QoS expressed in terms of parameters such as queuing delay, retransmission time, packet loss probability, and bit error rate. In a static resource allocation, available resources are set or assigned to the source traffic at service initialization and kept the same throughout the life of the connection. A dynamic approach is classified into two groups as synchronous and asynchronous (Fig. 1.3). In a synchronous method, resources are modified periodically, at fixed time intervals, unlike the asynchronous in which the allocated resources to the traffic are updated whenever a need is detected [22, 23, 26, 27, 29-31].

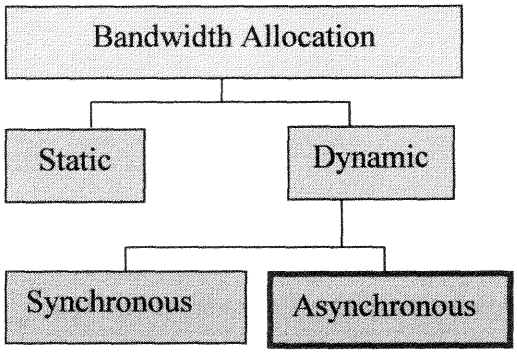


Figure 1.3 Bandwidth allocation methods

The admission control and resource allocation problem is complicated when a VBR multimedia source or communications device seeks access to the network and requests a virtual circuit for streaming data. The complication is due to that the features, which describe the variations in content of the multimedia, are generally imprecise. Thus, it is difficult to predict what the requirements for network resources are. If the network resource requirements are overestimated, the network will underutilize its capacity. On the other hand, if the network resource requirements are underestimated, then the network may become congested and packets traversing the network may be lost [19-21].

Bandwidth allocation and management for applications bit stream traffic is generally done at the edge routers in a network. This alleviates the computational load on the network switches. Off-line systems can determine the exact bandwidth characteristics of a stream a-priori. However, on-line algorithms are needed in many real time applications [22-28]. Online methods periodically renegotiate resources based on predicted traffic behavior [29-33]. Conventional methods typically renegotiate resources according to changes in bit stream level statistics as in [32]. Online methods have the advantage of adjusting resource allocations with respect to a desired QoS. On the other hand, most methods suffer from large number of renegotiations, and rely on very

complex measurements and allocation algorithms. If renegotiations are too frequent, signaling overhead increases. On the other hand, an infrequent number of renegotiations causes coarse predictions.

Many descriptors of traffic are known such as peak rate, average rate etc. However, these descriptors do not capture the traffic patterns over different time scales. To overcome this, a deterministic bounding interval dependent traffic descriptor (D-BIND) has been described by Knightly et. al. in [34]. D-BIND is a vector that includes a maximum allowed arrival rate for various time intervals. The allocation algorithm stores the currently reserved D-BIND parameters and calculates the D-BIND parameters for the last M frames. A Renegotiation takes place whenever a difference occurs between the reserved and measured D-BIND parameters. RED-VBR introduced in [32] is based on the D-BIND model. RED-VBR does neither use nor measure the QoS for allocation. It attempts to dynamically allocate bandwidth to provide zero losses. It allocates higher bandwidth than arrival rates, and renegotiates more often to meet stringent loss requirement. [28] presented another online renegotiation method called DSA+ (Dynamic Search Algorithm). The goal of DSA+ is to adjust the server rate so as to provide the desired cumulative cell/packet loss probability (CLP/PLP) as efficiently as possible with few renegotiations. At each renegotiation point DSA+ adjusts the server rate according to the following formula:

$$\mu_{n+1} \leftarrow \mu_n + \frac{K}{\alpha}\log(\frac{P_n}{Q_l}), \qquad \alpha = \begin{cases} 1 if & P_n > Q_l \\ 2 if & P_n < Q_l \end{cases} \tag{1.1}$$

where P_n is the CLP/PLP of the nth interval and Q_l is the CLP/PLP desired by the user. DSA+ requires initial renegotiation interval and constant K to be given by the application

and updates the server rate μ_n based on the observed CLP/PLP during the *n*th interval. Also, in [35] renegotiation schedule is studied. They discuss simple measurement-based admission control (MBAC) schemes suitable for RCBR (Renegotiated Constant Bit Rate) sources. They show that a memory less scheme is not robust and history about the past bandwidth allocations is needed to achieve satisfactory robustness. In their proposed method, a constant cost ϕ per renegotiation and a cost γ per allocated bandwidth are assumed. The total cost is given by

$$\phi.\sum_{i=1}^{N-1}(1-\delta(s_{i-1},s_i))+\gamma\sum_{i=0}^{N-1}s_i \tag{1.2}$$

with $\delta(x,y)=\begin{cases}1, & x=y\\ 0, & oth\end{cases}$. For a given r_i, the amount of data entering the queue during time slot *i,* and service rate s_i, the optimal allocation minimizing the total cost is found subject to the buffer constraint, $0 \le b_i \le B$ for $i \ge 0$ where b_i is the queue size at the end of time slot i with $b_i=\begin{cases}0, & if \quad i<0\\ max\{b_{i-1}+r_i-s_i,0\}, & i\ge 0\end{cases}$. They solve this optimization with a Viterbi-like algorithm [36]. This Viterbi-like algorithm cannot be used to determine optimal renegotiation points for online interactive sources. For such sources, causal heuristics have to be used to make decisions about requesting new rates. Such heuristics predict the future bandwidth requirement based on some statistics collected in the past. They introduce a heuristic based on a *AR(1)* bandwidth estimator and on buffer thresholds. Three parameters have to be tuned: a high and a low buffer thresholds B_h *and* B_l respectively and a time constant *T*. The rate predictor used in [36] is simply

$$\hat{r}_{i+1}=(1-T^{-1})\hat{r}_i+T^{-1}(r_i+\max\{b_i-B_h,0\}) \tag{1.3}$$

where r_i is the actual incoming rate during slot i, and b_i is the buffer size at the of time slot i. The bandwidth necessary to flush the current buffer content within T is compensated by the term $T^{-1}(r_i + \max\{b_i - B_h, 0\})$. Which they have not included in [36] and we would like to add to the cost function is the cost of inter-renegotiation interval. If an inter-renegotiation interval were less than the processing time of the last renegotiation, the new renegotiation request would have a high cost. Renegotiation is only feasible in time scales of several seconds [37]. In [38], it is suggested that minimum of 1 sec and an average of 5 seconds or more for renegotiation is a good compromise. It is crucial that optimal number of bandwidth renegotiations must be performed under predetermined cost constraints such as underutilization ratio and packet/cell transmission delay, inter-renegotiation times.

1.3. Our Approach and Contributions

In this dissertation, the impacts of selfsimilarity on applications delay and jitter performance are investigated, and a method consisting of a traffic smoother , a bandwidth predictor and a resource renegotiation unit optimizing the bandwidth prediction amount and time to allocate are invented. The dynamic resource allocation method develop in this dissertation is deployable in QoS routers and switch fabrics, and networks with real time response capability to the changing traffic and network conditions. A novel method that minimizes the number of bandwidth renegotiations within a network is introduced.

The block diagram of the designed " Dynamic Resource Management and Allocation" system is given in Fig. 1.4. The input to the S/P (serial-to-parallel conversion) is traffic bit arrival information within each predetermined time interval. The

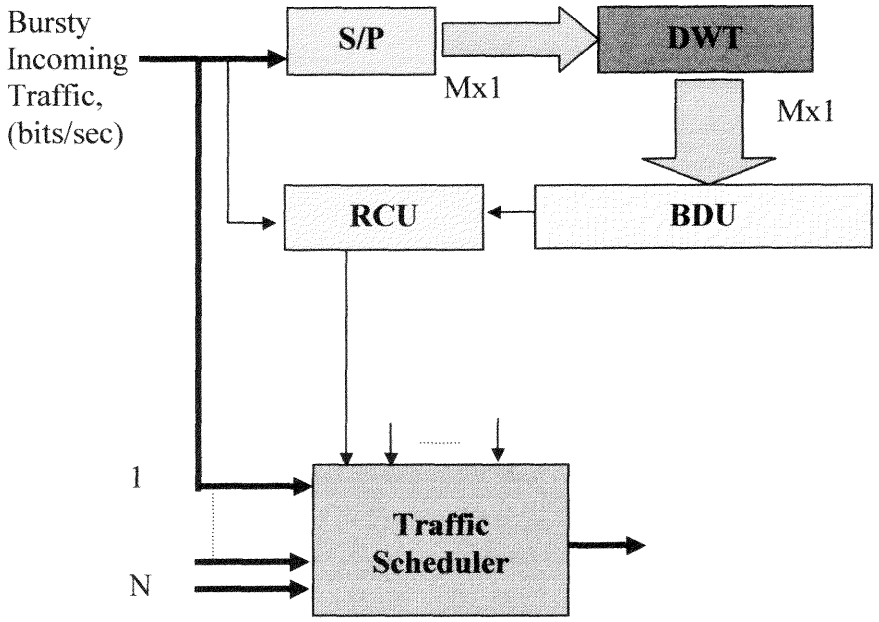


Figure 1.4 Block diagram of the developed resource management system.

size of the S/P also sets the length of the traffic data to be analyzed, that is the length of the time window that captures the data. The discrete wavelet transform (DWT) processes the input from S/P and provides BDU with parameters needed to predict the new resource demand of the VBR traffic for the next time slot. These parameters are the second order statistical parameters of energy distribution of the traffic bit arrival counting process among different frequency sub-bands. Based on the prediction result and the level of the cost terms defined, the Renegotiation Control unit (RCU) decides the time of a resource renegotiation and the amount of resources to allocate. The traffic scheduler manages the available backbone capacity among traffic traces of different priority levels. The scheduler implies a modified WRR or WFQ mechanisms in which the weight coefficients of each input queue is updated dynamically.

The new approach using a dynamic resource management complex with bandwidth predictor and renegotiator units aims at the best use of traffic descriptors and temporal network characteristics to provision required delay constraints to VBR traffic and maximum achievable utilization of a network capacity. The measures and techniques developed for renegotiation based dynamic bandwidth management should take into account;

- Traffic descriptors (e.g. bit arrival rate, self-similarity level, CBR, PLP)
- Temporal network characteristics (e.g. utilization)

As a summary of the past review and discussion, it is concluded that decreasing the impact of traffic self-similarity on queuing delays and packet/cell loss rates can only be controlled by adaptive bandwidth allocation to a self-similar traffic. Queue length distribution decays more slowly for long range dependent sources than short range dependent sources. This heavy tailedness of the queue size distribution by a dynamic bandwidth allocation can be decreased. Dynamic bandwidth allocation methods have been used in a number of previous studies. However, most methods suffer from a large number of renegotiations , and rely on a very complex measurement and allocation algorithms. Simplicity of an algorithm is very crucial to decrease delay due to processing time and to apply the changes in a timely manner. Motivated by these observations this research study aims at:

- Establishing an experimental and analytical framework for the analysis of traffic selfsimilarity and its measures by well-known methods. Evaluation of the suitability of both these methods and information they return to real-time

dynamic resource management units in terms of their complexities and robustness.

- Developing traffic descriptors such as second order temporal statistical properties of the energy distribution of a traffic bit arrival counting process and unique traffic related cost functions (e.g. under-utilization cost, cost of renegotiation and buffering).
- Applying these descriptors and functions in the context of a renegotiation-based dynamic resource allocation for VBR streams.

The contributions in this dissertation are as follows:

1. Empirical and analytical framework for linking the measure of selfsimilarity and estimators of the measure to online resource management [47, 49, 50, 51].
   - Analytical expression of the computational complexities of the variance-time method and the wavelet analysis in the estimate of the measure of traffic self-similarity.
   - Investigation of the impact of a window size used to capture data to analyze on temporal statistical traffic descriptors.
   - Determination of the frequency aliasing between neighboring frequency sub-bands in multi-resolution analysis of a traffic trace, and its impacts on the resulting energy distribution and the estimate of Hurst parameter for Haar and Daubechies wavelets.
   - Variation in the estimate of Hurst parameter with use of the Haar and Daubechies wavelets, and different number of sub-bands.
2. A new descriptor for dynamic bandwidth prediction [52]

- The use of discrete wavelet transform (DWT) to reveal the energy distribution information of traffic data.
- The use of energy distribution information and its second order statistical properties to predict future bandwidth demand of a VBR source.

3. Design of a bandwidth renegotiation unit [53, 54].

- Introduction of robust cost functions in bandwidth allocation decisions.
- Improvement of the performance of bandwidth-predictor based dynamic bandwidth allocation methods after the deployment of the renegotiation unit.
- Development of a method to minimize the number of bandwidth renegotiations.
- Achievement of higher utilization and less 0.99 queue size quantile by using less number of renegotiations than other methods in the literature (e.g. MSE, RED-VBR, PSN-TDNN).

1.4 Outline

In Chapter 2, the issue of how to estimate the measure of selfsimilarity in real time applications is investigated. This involves the comparison of the computational complexities of the available estimators. The variance-time and wavelet analysis methods are tested and the impact of different wavelet filters and number of sub-band frequencies in the estimate of Hurst parameter is studied. Frequency characteristic of a wavelet filter affects the amount of aliasing between neighboring sub-band frequencies, and accordingly amount of energies measured in each frequency sub-band. This creates offset in the estimate of Hurst parameter. Chapter 3 starts with a definition of a new traffic descriptor, energy distribution among sub-band frequencies, and explains a novel

approach in the design of a bandwidth predictor that takes the new traffic descriptor into an account. The new bandwidth predictor is compared with commonly used methods in the literature. In Chapter 4, the design of a bandwidth renegotiator unit with newly defined cost functions is presented. This renegotiator unit receives feedback from the dynamic bandwidth allocation unit, which is explained in Chapter 3, and decides when and at what amount to request a new bandwidth renegotiation by also keeping the number of renegotiations at minimum. In Chapter 5, the entire RDBA system, in its integrity, is applied to a WFQ scheduler with two priority levels to dynamically update weights assigned to each priority queue. Chapter 6 presents concluding remarks and future work.

CHAPTER 2

REAL TIME MEASUREMENT OF SELF-SIMILARITY

With will, fire becomes sweet water; and without will, even water becomes fire.

Rumi

In Chapter 1, the measure of self-similarity, Hurst parameter (H), is already defined, and the common methods that have been used in the literature [4, 5, 8, 13] to determine the self-similarity level of a given traffic bit arrival counting process are noted. It is pointed out that increasing level of self-similarity has negative impacts on the performance of networks and also degrades the QoS of individual applications. The difficulties in dynamic resource allocation to a self-similar traffic in real time are due to the practical limitations in information capturing at very small time scales (e.g. ms), and processing delay, which is an obstacle to a timely response to a need for resource adjustment. The correct determination of the self-similarity level and how much of the previous data information will be used in that computation is, therefore, crucial.

In this chapter, the three commonly used methods for estimation of the Hurst parameter are analyzed in terms of their computational complexities: *i)* Variance-time plot, *ii)* R/S pox plot and *iii)* Wavelet decomposition. As a result of the analysis performed, it is concluded that the use of wavelet decomposition is the most efficient in dynamic resource reservation and allocation where traffic self-similarity must be taken into account. A defined new traffic descriptor is an energy distribution of traffic bit arrival information process among different frequency sub-bands. With measure of this descriptor the frequency aliasing between neighboring frequency sub-bands and how the

aliasing affects the estimate of H for different wavelet types are studied. Recommendations are made for the selection of which wavelet type and length to use and the number of sub-bands in frequency domain to decompose traffic data.

Section 2.1 gives short background information on variance-time plot, R/S pox plot and wavelet methods. Section 2.2 reveals the computational complexities of the methods and drawbacks of each method for real time deployments, and compares the precisions of wavelet methods when different scaling and wavelet filters are used in the measure of Hurst parameter. Section 2.3 discusses the simulation results.

2.1. Hurst Parameter Measurement Methods

Intuitively, in traffic data consisting of data arrival information per unit time, mean is simply the average number of data units (e.g. cell, packet, bit) generated in periodic time intervals, and the peak is defined to represent the highest rate generated (shown in Fig. 2.1). For example, in Fig. 2.1 the mean and the maximum value of the synthetically generated VBR bit stream are 4.21Mbps and 5.98Mbps respectively. If the service rate is equal to or higher than the peak rate, no buffering would be needed, and the PLR would be zero. On the other hand, a service rate lower than the mean rate results in very large delays and higher PLR.

Choosing the service rate equal to the peak data rate is wasteful because very rarely does the source reach its peak. The effective bandwidth, or the optimal service rate, is the minimal service rate that can serve a stream such that the desired QoS parameters such as average packet delay and PLR are met. Optimal decisions on effective bandwidth would have to be made to operate the network economically.

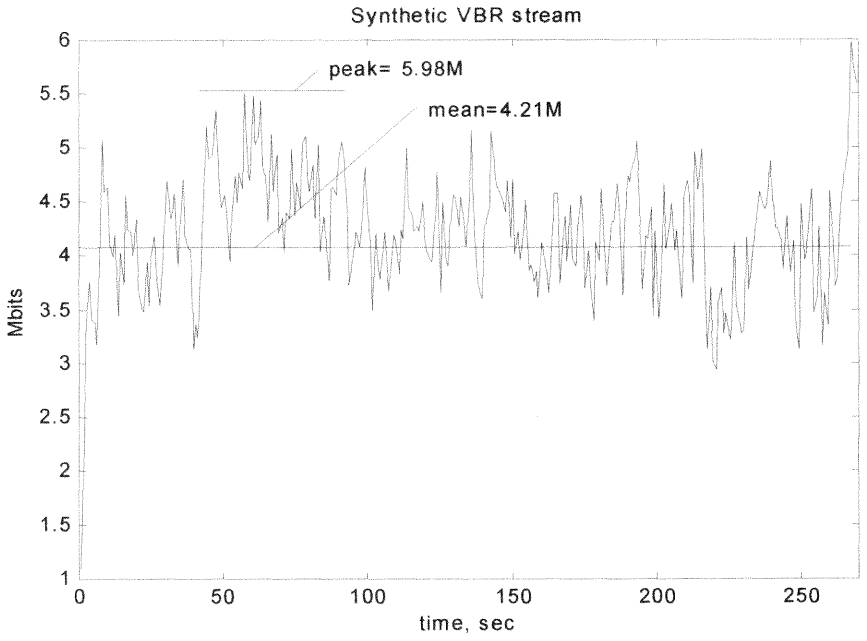


Figure 2.1 Illustration of peak and mean values on a synthetic VBR stream

The decision of how to set the effective bandwidth is a function of traffic burstiness that is delineated by the notions of variance/mean ratio of the traffic data arrival counting process, and the self-similarity level represented by Hurst parameter, H. Increase in the variance/mean ratio and in the value of H is integrated with raising burstiness, and accordingly the queuing performance is degraded. Therefore, finding and accurately measuring parameters to express the burstiness carry great importance. Especially in network systems that dynamically allocate network resources based on the measurement of these parameters, computational efficiency and accuracy of the algorithms or methods employed become significant.

This study, first of all, compares currently well-known and largely used three methods that return *H* as an estimate of the burstiness of any traffic data. These methods are: variance-time plot, rescaled-adjusted range statistics (R/S) plot and wavelet

approach. Optimum aggregation levels to be used in first two methods are determined. It is proven that there is a limit in choosing the level of maximum aggregation and this limit depends on the length of the data directly. If it is larger than a certain threshold, the resulting Hurst parameter gets affected. Also, in real time and dynamic measurements, there is a need to define a window length. Any change in the size of the window causes fluctuations in the value of H. In the wavelet approach, it is illustrated that the energy at each stage and the related stage relation is linear after a certain scale index and also that the length of the wavelet filters directly impacts the energy amount computed in each sub-band due to varying degree of aliasing that decreases with an increase in filter lengths.

2.1.1. Variance-Time Method

A stochastic time series $X = [X_1\ X_2\ X_3 \ldots X_N]$ is said to be self-similar if the process is covariance stationary. It means the process has constant mean and finite constant variance and the corresponding aggregate process has the same correlation structure as the original process or follows asymptotically the correlation structure of the original process over large intervals. Any aggregate series with block sizes of m is obtained by summing the original series in X over non-overlapping blocks of size m. This aggregate process is shown as $X^{(m)}$. The most important feature of self-similar processes is that the variance of the arithmetic mean, μ, decreases more slowly than the reciprocal of the sample size m as expressed in (2.1).

$$\mathrm{var}[X^{(m)}] \sim a.m^{(-\beta)}, m \to \infty \text{ and } a > 0 \qquad (2.1)$$

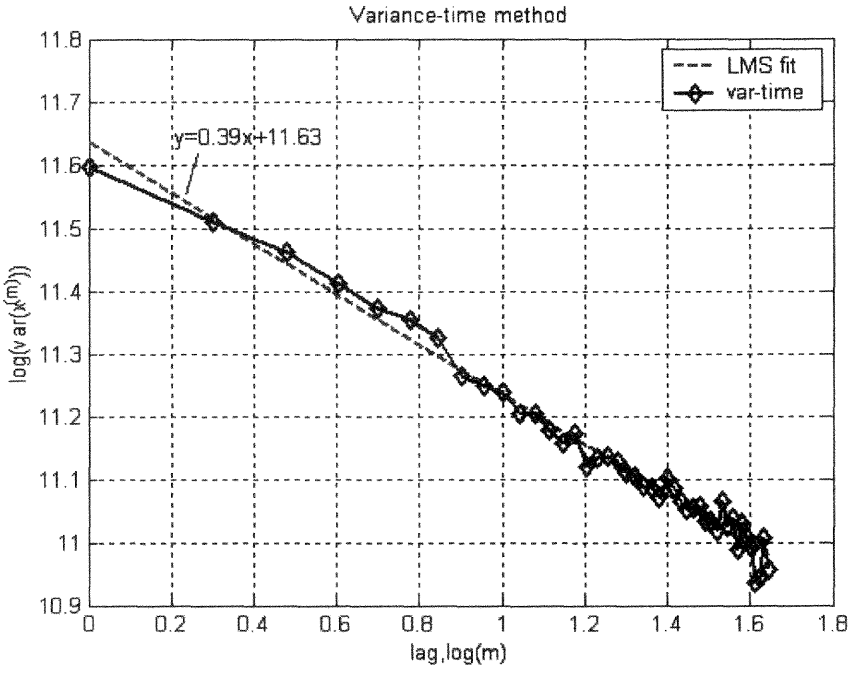


Figure 2.2 Typical output of the variance-time plot

After taking logarithms of both sides, the equation takes the form

$$\log(\mathrm{var}(X^{(m)})) \sim c - \beta \log(m) \tag{2.2}$$

The slope, β, is related to the Hurst parameter in a way that $H=1-\beta/2$. The typical output of the variance-time method after applying to a synthetic traffic trace is given in Fig. 2.2. The detailed related reading can be found in [4-8, 17].

2.1.2. R/X Pox Plot

R/S analysis provides another method of estimating the Hurst parameter. In practice, it is based on a heuristic graphical approach that tries to exploit as fully as possible the information in a given record. Given an observation vector X of size N, $X = [X_1\ X_2\ X_3 \ldots X_N]$, the entire sample is sub-divided into k non-overlapping blocks,

and the rescaled adjusted range $R(t_i,n)/S(t_i,n)$ for each of the new starting points t_i satisfying $t_i - 1 + n \le N$ is computed. The function $S^2(t_i,n)$ is the sample variance of $[X_{t_i+1}\, X_{t_i+2}\, X_{t_i+3} \ldots X_{t_i+n}]$. Hence, as many as k samples of R/S is obtained for a given value of n. $R(t_i,n)$ is computed by (2.3).

$$R(t_i,n) = \max_{0\le c\le n}[(\sum_{i=t_i}^{t_i+c} X_i) - (c+1)\mu(t_i,n)] - \min_{0\le c\le n}[(\sum_{i=t_i}^{t_i+c} X_i) - (c+1)\mu(t_i,n)] \quad (2.3)$$

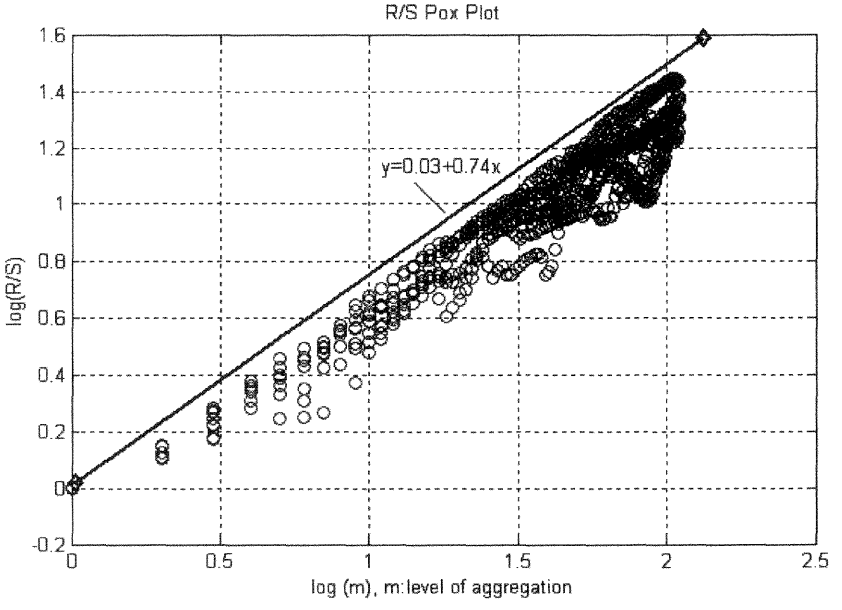


Figure 2.3 Typical output of the R/S pox plot

Hurst [55] found that many naturally occurring time series were well represented by the relation $R(t_i,n)/S(t_i,n) \sim an^H$ as $n \to \infty$ with H typically around 0.73. On the other hand, if the observation X_i comes from a short-range dependent process, H is around 0.5. It is clearly shown that the slope of the log of *R/S* is related to *H,* and this relation is used to estimate the value of *H* by R/S method for a given time series as illustrated in Fig. 2.3.

2.1.3. Wavelet Energy Method

Intuitively, the discrete wavelet transform divides a signal into different frequency components and analyzes each component with a resolution matched to its scale. By fixing a given scale j and studying time series data X at that scale across time, an information about the scaling behavior of X can be obtained as a function of j. On the other hand, fixing a point t_0 in time and investigating the wavelet coefficients across finer and finer scales results in powerful techniques to be able to investigate the nature of local irregularities or singularities in signal X, as a function of t_0.

Wavelets with their built in scale localization are ideal tools to analyze the scaling behavior of self-similar processes across wide range of time scales. An approximation of a signal X at scale j, X_j can be written as the sum of a coarser approximation X_{j+1} at scale $j+1$ and the detail D_{j+1}, the difference between these two approximations (Fig. 2.4). This procedure can be iterated for further scales. These coarse and fine approximations give the multi-resolution analysis (MRA) of a signal. MRA guarantees the existence of a scaling function ϕ (to express the approximation) and a wavelet φ (for the definition of details) such that signal X can be written as

$$X = \sum\sum < X, \phi_{j.k} > \phi_{j,k} + \sum\sum < X, \varphi_{j.k} > \varphi_{j,k} \qquad (2.4)$$

where

$$\phi_{j,k}(t) = 2^{-j/2}\phi(2^{-j}t - k) \text{ and } \varphi_{j,k}(t) = 2^{-j/2}\varphi(2^{-j}t - k)$$

The representation in (2.4) is called the *wavelet decomposition* of the signal X. The inner product of X, $d_{j,k}$, is referred to as the wavelet coefficients at scale j and time $2^j k$. The quantity $|d_{j,k}|^2$ measures the amount of energy in X at that scale [39, 40]. If X is self-

similar process with Hurst parameter H∈ (1/2, 1), then the expectation of the energy E_j that lies within a given bandwidth 2^j around frequency $2^j\lambda_0$ is given by

$$E[E_j] = E\left[\frac{1}{N_j}\sum_k |d_{j,k}|^2\right] = c\left|2^{-j}\lambda_0\right|^{1-2H} \tag{2.5}$$

N_j denotes the number of coefficients at scale j and c is a pre-factor that doesn't depend on j. Taking the logarithm of the both sides results in (2.6). The Hurst parameter can be derived from the slope of the relation between *log* (E_j) and scale index j.

$$\log(E_j) = [(2H-1)\log(2)]j + const \tag{2.6}$$

For an asymptotically self-similar signal, a linear relationship between the plot of *log* (E_j) and j will be apparent only for large times or scales (sec, min). This finding is very crucial especially if the concern is dynamic resource allocation. Because, increasing the number of scales or capturing low-resolution data introduces delay in computation of applications resource needs. Therefore, determining minimum required number of wavelet tree branches and feasible time scales to capture data directly impinge on the performance of real time dynamic resource allocation algorithms.

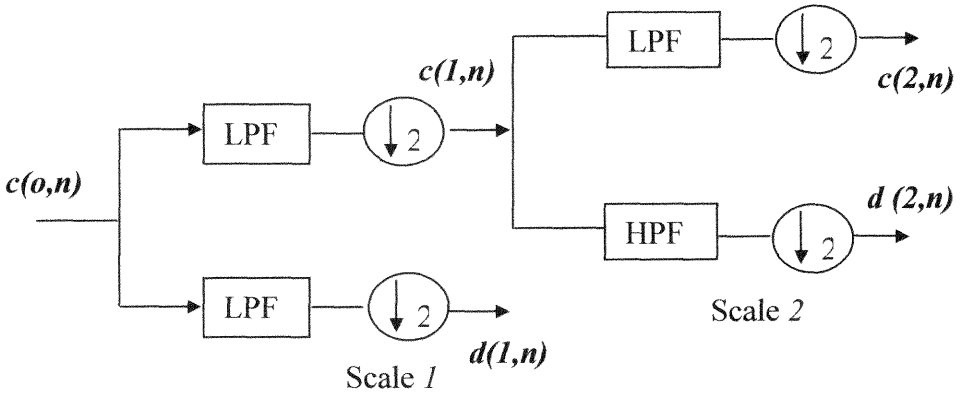


Figure 2.4 Illustration of the two level dyadic tree structure (*d(i,n)*:wavelet coefficients at scale i)

In the next chapter entitled "One-Step Ahead Bandwidth Predictor Design", the wavelet analysis is used not necessarily to compute self-similarity level and Hurst parameter of the incoming traffic, but to compute temporal energy distribution in different frequency sub-bands to predict the characteristic near future traffic behavior. The wavelet approach in bandwidth prediction is explained elaborately in Chapter 3.

2.2. Empirical Studies and Results

2.2.1. Computational Efficiency of the Algorithms

Real time dynamic resource management systems involve buffering (windowing) certain amount of traffic data to analyze temporal traffic characteristics, and certain decisions are made based on the returned results. After each measurement the window is moved either in overlapping or non-overlapping blocks of its size (WS). As illustrated in Fig. 2.5, the size of the time window is directly related to the processing time of an algorithm. The longer the window size, the larger the processing delay is. In order to capture long time scale properties of a traffic trace, the window size should be enlarged accordingly. However, in that case, the ability to track short time scale non-stationeries of the trace becomes difficult.

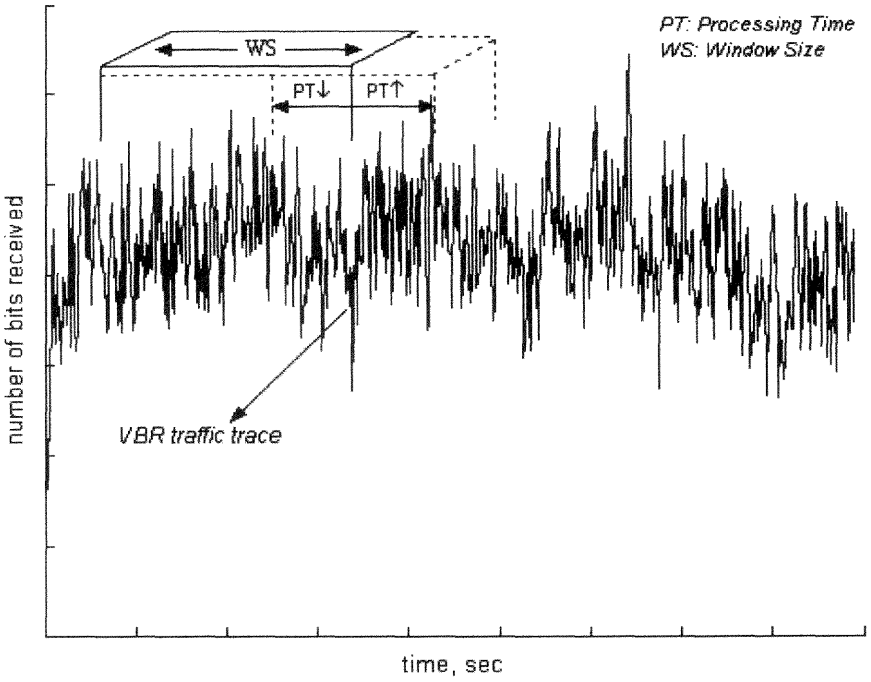


Figure 2.5 Illustration of windowing a traffic trace and "window size"-"processing time" relation.

Processing complexities of variance time plot and wavelet methods used in estimating Hurst parameter are compared at different trace lengths, aggregation levels (only for the variance-time method) and number of scales in a dyadic tree. Assuming m is the maximum aggregation level allowed in variance-time method and N the length of the trace, in estimating the Hurst parameter the following number of arithmetical operations is needed: In variance-time method at each aggregation level up to m, $\frac{(i-1)}{i}N$ addition operations are needed where $i=1,2..m$. Mean at each aggregation level is the same and therefore the total contribution is N due to the mean computation. Also, m different

variance values are needed generating $\sum_{i=1}^{m} 2N/i + 2m$ operations. The total number of arithmetical operations $f_{vt}(m,N)$ needed in the variance time method is

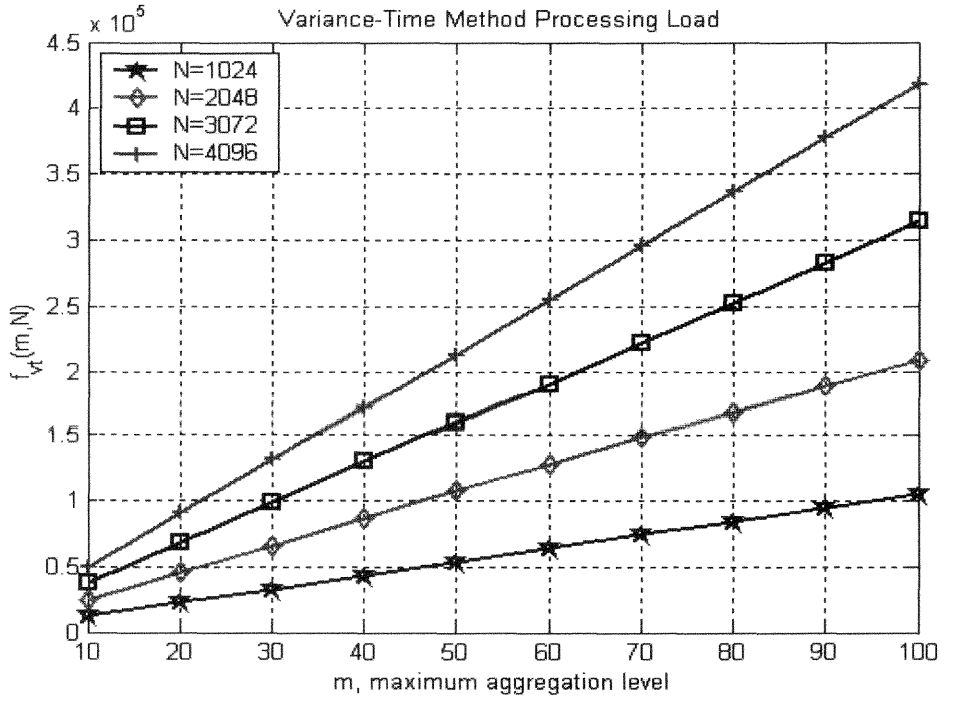
$$f_{vt}(m,N) = N(\sum_{i=1}^{m} 1/i + m + 1) + (2m-1) \tag{2.7}$$

When the wavelet method is preferred (Haar wavelet and scaling filters) with n representing the number of scales in a dyadic tree, the total number of operations $f_w(n,N)$ is

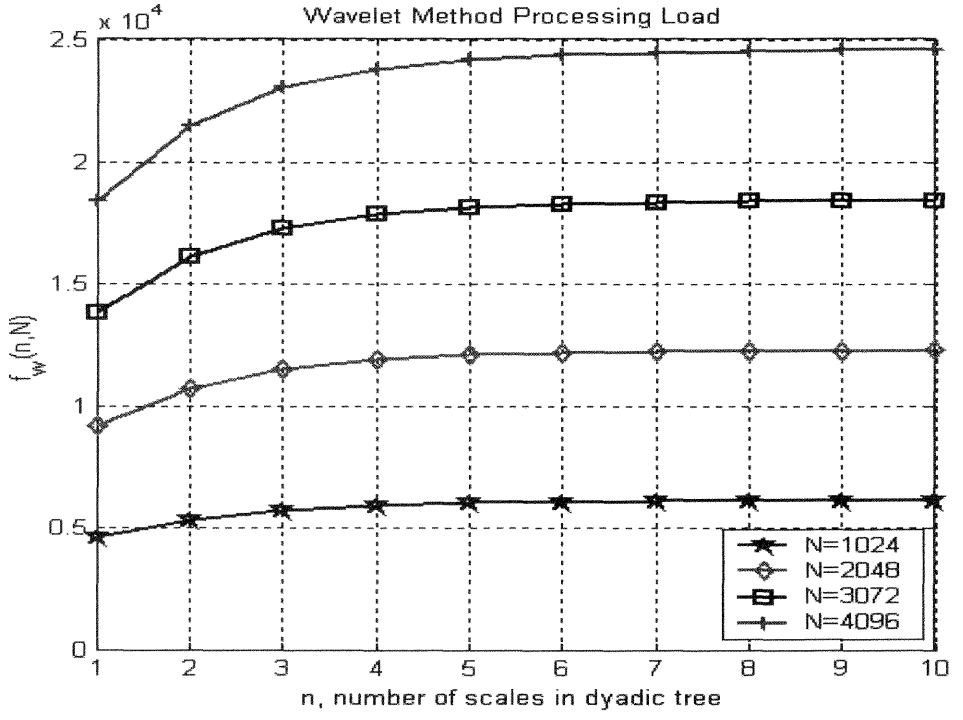
$$f_w(n,N) = (\sum_{i=1}^{n} 3N/2i) + 3N - n \tag{2.8}$$

This formula is valid only for Haar wavelet. Haar wavelet, due to its simplicity, constitutes the lower bound compared with other wavelet types in terms of the number of processing operations generated.

Figure 2.6 simulates the number of arithmetic operations needed in each method when different length windows are deployed. It is obviously seen from the figure that wavelet method needs less number of operations, and therefore is faster than variance time method. In variance time method, the algorithmic run time linearly increases as the number of window size and the determined maximum aggregation level increase. However, in wavelet method increase in the length of the window has a linear effect on the total number of operations performed, but not the number of scales has the similar type of linear effect. $f_w(n,N)$ is a function of the dyadic three level m such that $f_w(n,N) = K_1(N) - K_2(N)e^{-n}$ where functions $K_1(N)$ and $K_2(N)$ are determined from Fig. 2.6.



(a)



(b)

Figure 2.6 Variability of algorithmic complexities in (a) the variance-time method (b) the wavelet method.

Intuitively, for the same number of tree scales, $f_w(n,N)$ gets higher as N increases. Marginal contribution of each scale to the computational complexity decreases as 2^n approaches the window length. As for R/S pox plot method, we have not done any detailed computational time analysis. However, it creates, $o(N^2)$ arithmetic operations that is quite more than those of variance-time and wavelet methods. Therefore, R/S pox plot is not convenient to use for online computations.

2.2.2. Statistical Parameters and Their Dependence on Window Size

Depending on the size of analyzed data, statistical traffic descriptors may show temporal variations. Therefore, correct pre-determination of the window size is non-trivial. Hurst parameter is not a good representative of traffic characteristics such as burstiness. Two time series with the same Hurst parameter might have quite different mean and variance. Therefore, additional statistical measures are also needed and must be input to resource management algorithms. Some of the parameters used as traffic descriptors in the literature are:

Mean - Mean value of the time series generated in a given time interval.

Peak - The highest rate generated.

Variance/Mean - The ratio of variance to mean value.

Burst Length - The length of the burst. It is defined to be such that if a packet inter-arrival time is greater than a time threshold, the burst ends.

Fig.2.7 illustrates two test arrays X_1 and X_2 with size 5 such that $E[X_1]=8$ and $E[X_2]=8$, and $\max[X_1]=18$, $\max[X_2]=15$. In this example, X_1 has a higher standard deviation. However, the peakedness of X_2 given by *peak/mean* is higher. The selection of

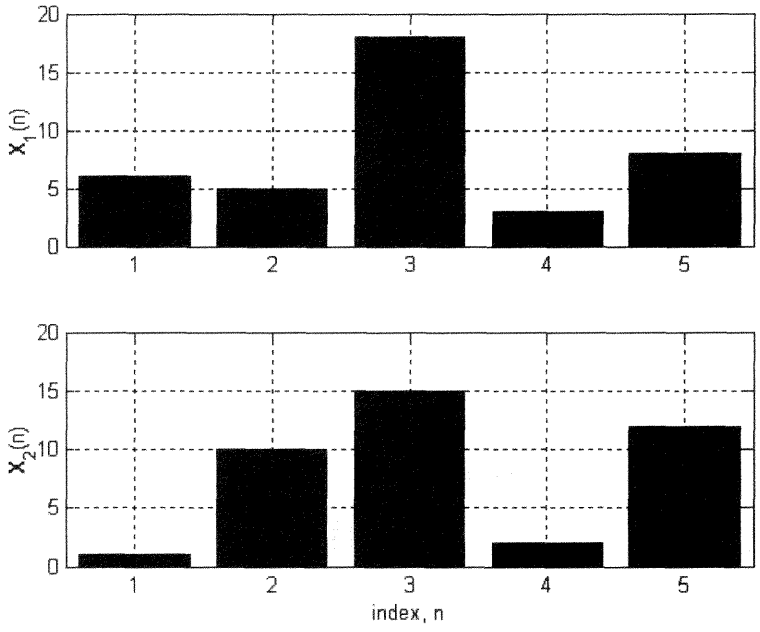


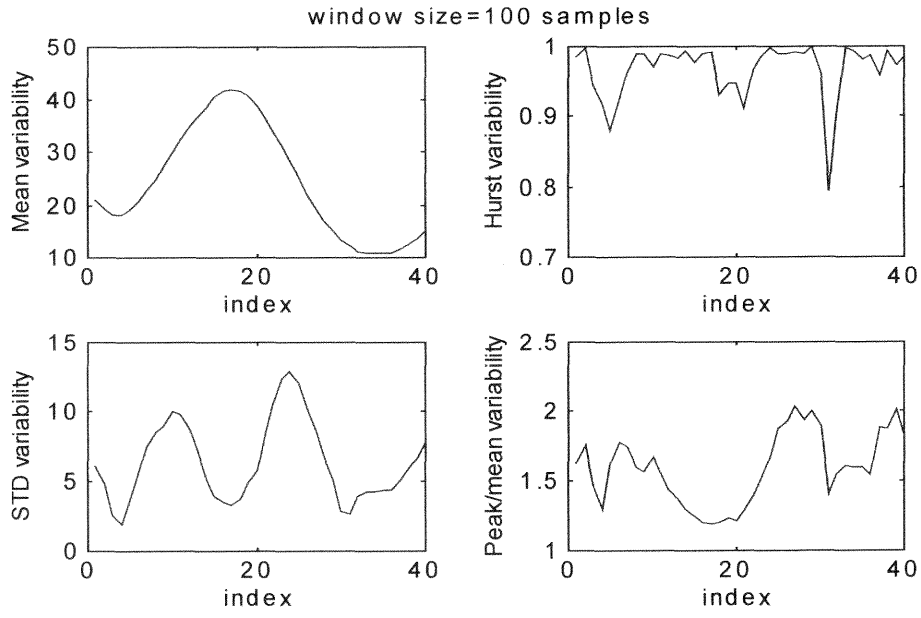
Figure 2.7 Illustration of two synthetic traces with the same mean, but different burstiness and peakedness.

the arrays proves that a trace with higher standard deviation may not necessarily have the higher peakedness. These parameters are used in the invented bandwidth prediction method given in Chapter 3 as traffic descriptors, not in the time domain but in frequency domain, to characterize energy distributions in different frequency sub-bands (e.g. mean and variance of sub-band energies).

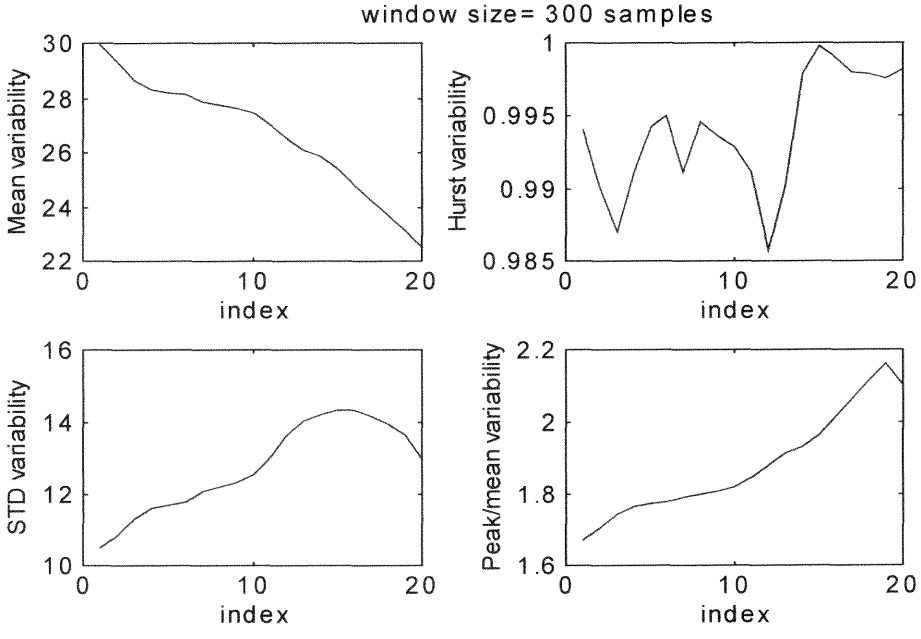
The statistical parameters such as Hurst parameter, mean and variance of bit arrivals per unit time do not show stationary characteristics at different window sizes. Selection of the window size as illustrated in Fig. 2.5 has a direct impact on the temporal values of these parameters. A window size dependent variation in the burstiness parameters is illustrated in Fig. 2.8. First, a synthetic time series of 1024 samples are generated. Each sample of the synthetic trace represents the number of bits arrived within a corresponding time slot. Second, a moving window function *g(n)* is defined, of which

the size is 100 samples. Third, the statistical parameters of the data encapsulated by *g(n)* is computed. These parameters are mean, peak/mean ratio, standard deviation and Hurst parameter in time domain. Finally, the window is shifted in overlapping samples of size 10, and the computations are repeated. The results are compared with those of the window size 300. The series of these values for each shift give the variability of the parameters as a function of the windowing index.

From Fig. 2.8, it is concluded that although the Hurst parameter floated around 0.9, large-scale fluctuations for the mean and standard deviation of the data were observed. Allocation of resources like effective bandwidth under the consideration of only Hurst parameter would be misleading in this case and poor as well. Increase in the mean value can be interpreted as the admission of new applications to the link or the increase in transmit rate of currently available applications at that specific time. Although with the window size of 100 samples, there is an increase and decrease in the mean rate, when the window size is enlarged to 300 samples, the increase in the mean rate is not observable. Similar inconsistencies are also apparent for standard deviation of the bit arrival rate and peak to mean variability in each case as well. It is also clear that depending on the size of the window the traffic statistics follow different trends. Therefore, for real time resource allocation algorithms, optimum selection of the window size is important to make decisions regarding at what amount to allocate network resources to a single source or at what ratio to share the available resources among multiple applications.



(a)



(b)

Figure 2.8 Temporal variations of traffic statistics depending on the time window size (a) window size: 100 samples (b) 300 samples.

2.2.3. Effect of Wavelet Filter Types and Filter Lengths on Energy Estimations

The effects of the number of filter taps in a dyadic tree in the estimation of Hurst parameter are studied. Different length Daubechies wavelet filters are used in the analysis. It is shown in [39] that there is a linear relation between the log of energy in each frequency resolution and the related scale. The frequency spectrums of different tap Daubechies low pass filters in Fig. 2.9 show that as the number of taps increases, the tail of the frequency spectrum decays much steeper. 10-tap filter characteristic is closer to the ideal rectangle frequency response than 4-tap filter is. This difference affects the alias between sub band frequency spectrums and expectedly the resulting energy content in each sub band. Explicitly in the high frequency sub-band, the energy content would be less for 10-tap filter case than when the filter has 6 taps. Increasing the number of taps in the deployed wavelet filter, based on the tail distribution of the frequency response as

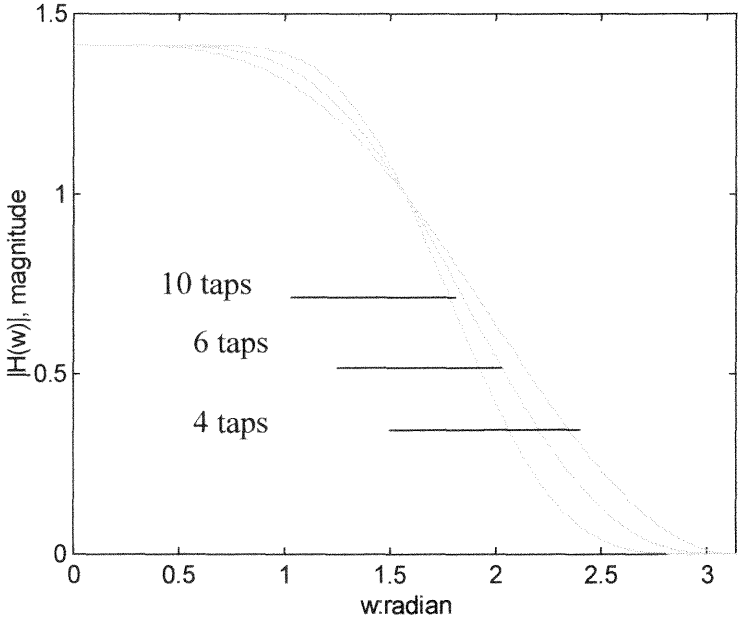


Figure 2.9 Frequency response of Daubechies filter with 4,6 10 taps

shown in Fig. 2.9, introduces offset in energy content of each frequency sub-band. This offset is less for the sub-band with highest scale index, that is the lowest frequency sub-band in which most of the energy is available. The offset value can be illustrated by applying the wavelet analysis to the first 8192 frames of MPEG-1 coded Star Wars trace (40ms inter frame interval and 25 frame/sec). Fig. 2.10 exemplifies the energy offset due to frequency aliasing. Intuitively, the energy offset is less between filter orders 2 and 5 than that between 4 and 5. The LMS fit returns Hurst parameters within the range 0.86±0.02 for wavelet orders of 2, 4 and 5. Two percent deviation in the estimate of H due to different order wavelet filters is negligible. Considering the fact that deployment of high order filters elevates the processing time and results in additional delay, low order wavelet filters can be traded off in real time applications to the higher order wavelet filters because the deviation in the estimate of H due to the number of taps is trivial.

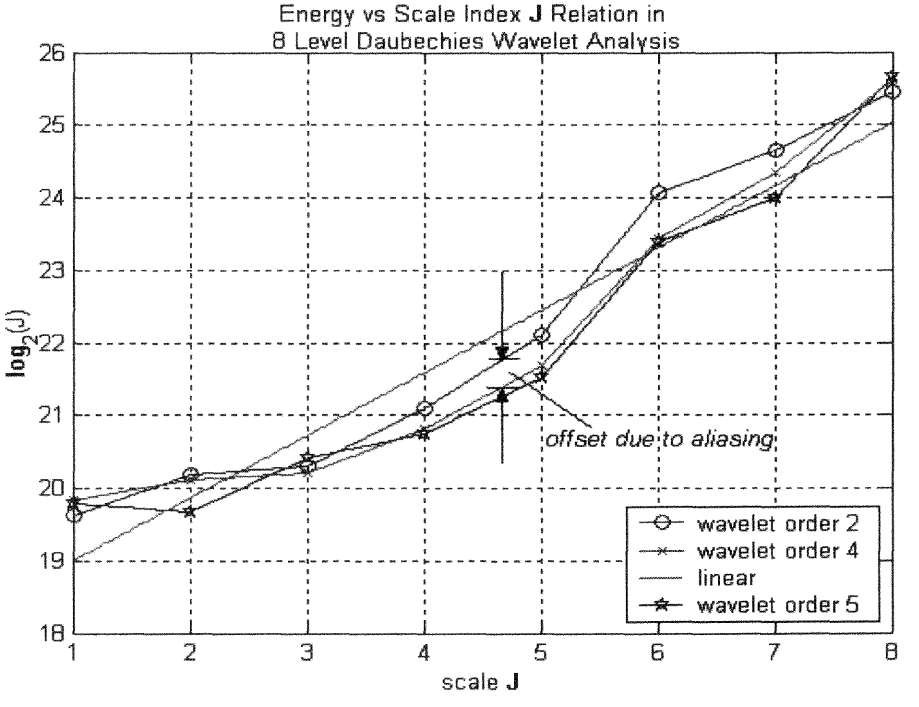


Figure 2.10 Impacts of the number of wavelet filter taps on signal energy distribution among different sub-band scales.

How does using a different type of wavelet filter affect the estimate of Hurst parameter? This question is answered by comparing the estimates returned by Haar and Daubechies wavelets.

Table 2.1 Wavelet Coefficients for Haar and Daubechies

n	0	1	2	3
Haar	*0.707*	*-0.707*	-	-
Daubechies 4-tap	*-0.1294*	*-0.2241*	*0.8365*	*-0.4830*

Table 2.1 gives the wavelet coefficients for Haar wavelet and *4*–tap Daubechies wavelet. It is analytically proven that the energy content in the highest frequency sub-band varies with the selection of the wavelet filter type, and that it is less in case of Haar wavelet than that of Daubechies. The power spectrum of the self-similar signals asymptotically follows a *1/w* behavior where $w=2\pi f$, that is $1/|w^{\gamma}|$. In a linear system, having *X(Z)*, H(Z) and Y(Z) as the *Z* transforms of random input signal *x(n)*, the transfer function *h(n)*, the output signal *y(n)* respectively, then

$$Y(Z) = X(Z)H(Z)H(Z^{-1}) \tag{2.9}$$

after substituting $Z = e^{jw}$

$$Y(e^{jw}) = X(e^{jw}) \mid H(e^{jw}) \mid^2 \tag{2.10}$$

For Haar and 4-tap Daubechies wavelet filters, the frequency responses *h(n)* from Table 2.1 are

$$h_{haar}(n) = 0.707\delta(n) - 0.707\delta(n-1) \tag{2.11}$$

$$h_{daub}(n) = -0.1294\delta(n) - 0.2241\delta(n-1) + 0.8365\delta(n-2) - 0.4830\delta(n-3) \tag{2.12}$$

where $\delta(n-r)=\begin{cases}1, n=r\\0, n\neq r\end{cases}$. In frequency domain, after substituting $1/|w^{\gamma}|$ into (2.10) for both Haar and Daubechies wavelets, we have

$$\int_0^{\pi} Y_{haar}(e^{jw}) = \int_0^{\pi} w^{-\gamma} \left| \sum_{i=0}^{1} h_{haar}(i) e^{-jwi} \right|^2 dw \tag{2.13}$$

$$\int_0^{\pi} Y_{daub}(e^{jw}) = \int_0^{\pi} w^{-\gamma} \left| \sum_{i=0}^{3} h_{daub}(i) e^{-jwi} \right|^2 dw \tag{2.14}$$

γ is related to the level of self-similarity and its relation to H is $\gamma=2H-1$. For different γ, the comparison of the sub-band energies returned by Haar and Daubechies wavelet filters are illustrated in Fig. 2.11. The offset in the high frequency sub-band energy computation between two wavelet types increases as γ goes larger. As explained before, Haar wavelet analysis returns higher high frequency energy than Daubechies because its tail in frequency domain decays slower than that of Daubechies.

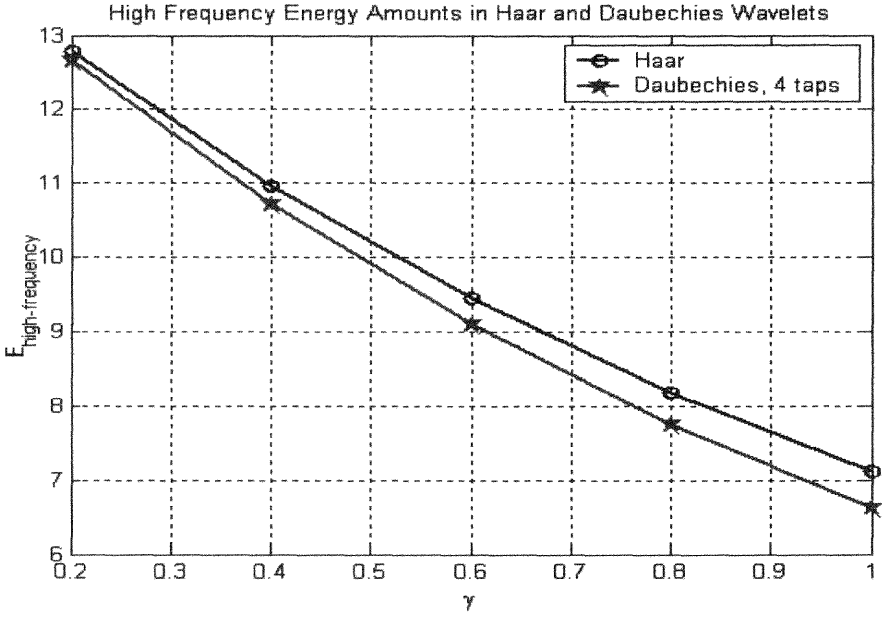


Figure 2.11 High frequency energy estimation by the Haar and 4-tap Daubechies wavelet filters.

Analysis of the Star-Wars trace also revealed the variation in energy estimation between Haar and Daubechies wavelets with eight level filter bank. It is interesting to note that, as illustrated in Fig. 2.12, throughout the first four scales the slope of the energy curve is the same for each wavelet filter type. The resolution at scale 1 is 80ms, that is twice the inter-frame intervals in Star Wars frame. The sampling frequency at the fourth scale is 16 times smaller than the original due to 4 consecutive down-sampling processes in the wavelet filter bank. The time resolution of the trace at scale four is, therefore, 640 ms. This result suggests that either Haar or Daubechies wavelet filters can be deployed in energy analysis as long as the number of scales is less than 4. Other conditions to satisfy in choosing data vector size and number of wavelet filter bank scales are discussed in the next section.

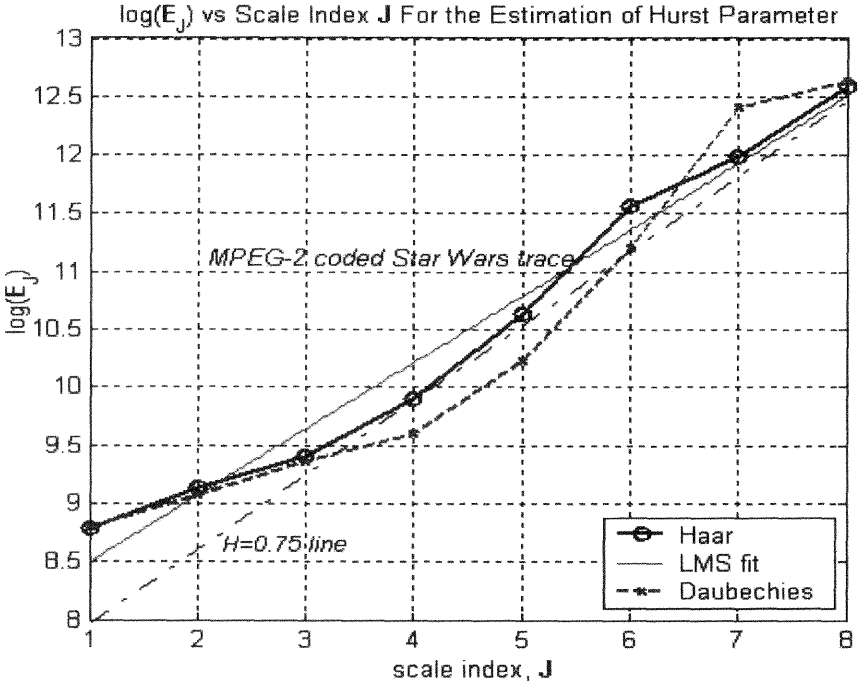


Figure 2.12 Comparison of the Hurst parameter estimate of Haar and Daubechies wavelet analysis.

2.3. Summary and Discussion of the Results

Taxonomy of techniques for real time detection and measurement of self-similarity in network traffic is provided in this chapter. The pros and cons of the methods cited in the literature are discussed in terms of the processing power they require to compute Hurst parameter, and their applicability for online algorithms by using them on VBR traces are demonstrated. The resolution at which traffic data is collected is seen to have a profound impact on the estimation of the Hurst parameter, and on the performance of a renegotiation based dynamic bandwidth management algorithms.

Assume *s(t)* is a bit counting process within the time interval (0, t]. Then, we can express the increment process *x(i)* of *s(t)* in time slot i with length ϕ such that $x_i = s(t + i\phi) - s(t + (i-1)\phi)$. Data vector X consisting of the latest M values of the increments is $X = [x_1 x_2 ..... x_M]$. X contains the data within time interval of length ϕM. In dynamic resource allocation, we have to update vector X such that it might include both new increments and some of the previous increments. If X does not include any increment from its previous instant, it is called a non-overlapping vector, otherwise overlapping data vector. Assuming a renegotiation is feasible in every a second, the requirements in the following scenarios must be satisfied for effective online renegotiation capability.

Scenario A: If $\phi M > a$ and we create vector X in non-overlapping fashion, we lose the advantage of renegotiation capability. To generalize, lets have m present the vector interval. If $m>0$, non-overlapping occurs as in Fig. 2.13. It takes $(m+M)\phi$ time unit to update vector X and fill it with all new samples. In such a case, it would be difficult to follow high-frequency fluctuations with a period smaller than $m\phi$ in a traffic pattern and update allocated bandwidth to an application in a timely manner before queue-size constraint is exceeded. On the other hand, increase in $M\phi$ introduces additional processing delay to the completion of a renegotiation request.

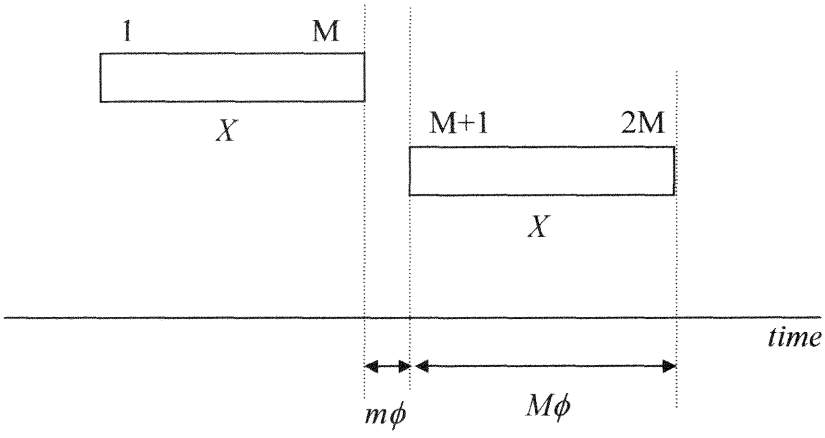


Figure 2.13 Illustration of the non-overlapping vector X of traffic bit arrival counting incremental process with sample size M

Scenario B: Assume k is the overlapping sample size in vector X (Fig. 2.14). Inefficient use of renegotiation capability occurs when $(M-k)\phi > a$. Therefore one of the conditions to satisfy is $(M-k)\phi < a$. Another condition is derived from DWT processing delay. The vector X with sample size M can be fed into a wavelet filter bank having $\log_2 M$ scales and $(1+\log_2 M)$ frequency sub-bands. M must be a positive integer

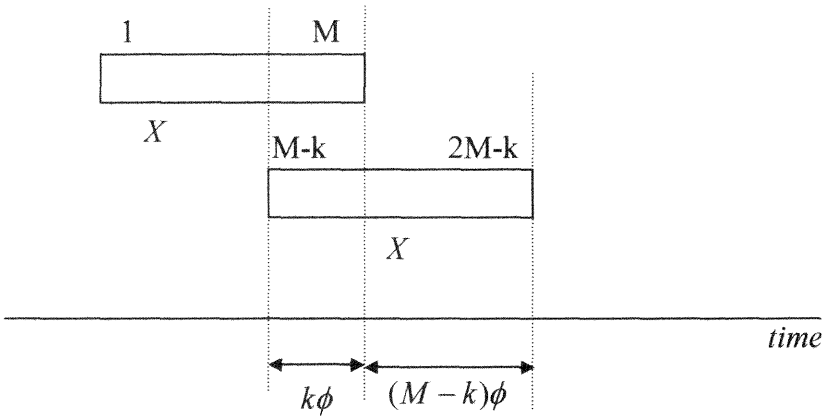


Figure 2.14 Illustration of the overlapping vector X of traffic bit arrival counting incremental process with sample size M

and a power of 2. A long vector size is a limitation due to increasing computation time in DWT. Remembering the assumption that acceptable renegotiation interval is a seconds, the sum of the processing time of DWT η_p and fill time η_f of vector X with new samples must be less than a seconds.

$$\eta_p + \eta_f < a \tag{2.15}$$

It is known that $\eta_f = (M-k)\phi$. Therefore, η_p is derived from Fig. 2.6b such that

$$\eta_p = c[K_1(M) - K_2(M)e^{-n}] \tag{2.16}$$

where n stands for the number of scales in wavelet filter bank, and c is a constant dependent on a CPU speed. After substituting n in (2.16) with $\log_2 M$, the processing time is

$$\eta_p = c[K_1(M) - K_2(M)e^{-\log_2 M}] \tag{2.17}$$

The condition to satisfy in selection of k and M is then,

$$c[K_1(M) - K_2(M)e^{-\log_2 M}] + (M-k)\phi < a \tag{2.18}$$

In its entirety, the comparative study reveals that the wavelet decomposition method is a faster and more reliable method for real time analysis. The contributions in this chapter are as follows:

- The number of arithmetic operations required by the variance-time analysis is a positive linear function of the maximum aggregation level determined to be one eight of the trace size to have the enough number of samples in the variance computation at the highest aggregation level.
- The number of arithmetic operations generated in wavelet analysis asymptotically follows $K_1 - K_2 e^{-x}$ curve where x is the number of levels in wavelet filter tree, and K_1 and K_2 are functions of window length N determined from Fig. 2.6b.
- In wavelet analysis, the residual of the number of computations performed at two consecutive scales is smaller as the scale index gets higher.
- The size of the data windowed to analyze directly affects the values of statistical parameters such as H, mean and variance of the traffic. Larger window size brings about heavy computational load, and it also makes the tracking of the changes in traffic characteristics difficult in real time resource management.
- Deploying wavelet filters with large number of taps reduces the aliasing between adjacent frequency sub-bands. On the other hand, small number of taps generates positive offset value in energy amount in each sub-band due to increased aliasing.
- Due to the *1/w* behavior of selfsimilar signals and tail behaviors of the frequency responses of filters deployed, the energy amount estimated by the Haar wavelet is slightly higher within the first three sub-bands. However, this difference is insignificant, and the Haar wavelet is preferred for its simplicity in computations.

CHAPTER III

A NOVEL ADAPTIVE BANDWIDTH ALLOCATOR

Sell your cleverness and buy bewilderment; Cleverness is mere opinion, bewilderment is intuition.

Rumi

Traffic measurements are keys in the development of robust resource management and scheduling algorithms. Bandwidth allocation methods are either static, or they adaptively change the bandwidth assigned to applications. Adaptive algorithms predict the near future behavior of a traffic based on its statistical properties such as average arrival rate and peak to mean ratio and update bandwidth allocations.

The developed novel feedback mechanism uses wavelets, and measures the energy of a bit counting process at the output of each dyadic sub band filter. Energy information in each frequency band from fine to coarse scales in forecasting future traffic arrivals is the key component. Specifically in VBR traffic, bandwidth requirements usually change due to coding structures [8, 22, 23, 25, 29, 33]. For example, MPEG introduces B frames and employs a group of pictures (GOP) structure that do not exist in H.261. Each GOP starts with I frame followed by a P frame. B frames are smaller in size (bits) than both P and I frames [22, 25, 45]. Traffic rate changes within a GOP can be considered as short-term behavior. On the other hand, rate changes due to scene changes can be considered as a long term traffic behavior. It is shown that short-term and long-term fluctuations in the arrival pattern can be separated by wavelets. One can determine dominance of either of these by looking into the energy distribution in respective frequency bands. Finally, this information is used to update bandwidth allocation.

3.1 An Optimum Bandwidth Allocation for Bit Arrivals During Two Consecutive Time Slots

3.1.1 Aggregate Bandwidth Allocation Error

Assume x_i is the number of bits arrived within time interval i, and $\hat{x}_i$ the allocated service rate equal to the prediction of x_i. The difference $|x_i - \hat{x}_i|$ results in either buffering or capacity under-utilization. In traffic management systems where buffering and under-utilization constraints are given, an adjustment in resource allocation may be needed depending on the level of this difference. Assume that this threshold level is denoted by r_{neg}, and that arrival rates within n consecutive time slots are greater than their corresponding predictions, and an aggregate error up to interval n is e_n such that

$$
\begin{aligned}
&|\hat{x}_1 - x_1| = e_1 \\
&|\hat{x}_2 - x_2| + e_1 = e_2 \\
&|\hat{x}_3 - x_3| + e_2 = e_3 \\
&\cdots\cdots\cdots\cdots \\
&|\hat{x}_n - x_n| + e_{n-1} = e_n
\end{aligned}
$$

The generalized form for the aggregate error in terms of each prediction error is

$$\sum_{k=1}^{n} |(\hat{x}_k - x_k)| = e_n \tag{3.1}$$

The probability that this aggregate error e_n does not exceed a threshold level r_{neg} is given in (3.2).

$$Pr\{e_n < r_{neg}\} = Pr\{\sum_{k=1}^{n} |(\hat{x}_k - x_k)| < r_{neg}\} \tag{3.2}$$

The perfect solution to keep this probability at maximum is that $\forall k, (\hat{x}_k - x_k) = 0$. This requirement actually involves a zero delay perfect rate predictor and a zero delay

bandwidth allocation during each time slot. This solution is not practical. In bandwidth allocations, if the aggregate error threshold is distributed among n allocations, and the constraint that $x_k - \frac{r_{neg}}{n} < \hat{x}_k < x_k + \frac{r_{neg}}{n}$ is satisfied for $\forall k$, the probability in (3.2) can be also maximized.

3.1.2 Optimum Allocation

The main purpose in dynamic bandwidth allocation is to keep the queue size and under-utilization levels below their constraints at minimal number of reallocations. To have a gain in a number of bandwidth renegotiations, the starting point is at least to allocate a fixed bandwidth r_a during two consecutive time intervals *(t, t+Δ)* and *(t+ Δ, t+2Δ)* as illustrated in Fig. 3.1. If this is accomplished for each two consecutive time intervals in non-overlapping fashion, the resulting number of bandwidth reallocations for a stream of

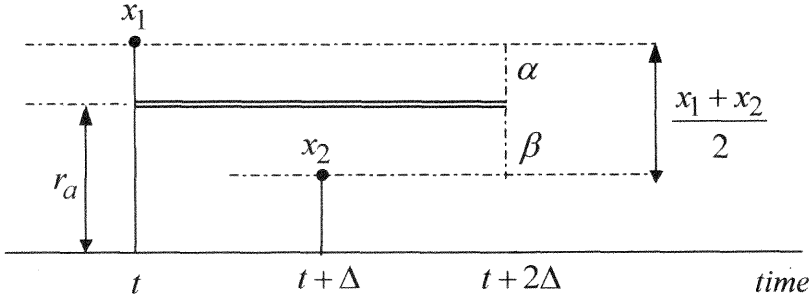


Figure 3.1 An illustration of an optimum bandwidth allocation for bit arrivals within two consecutive time slots

N samples would be $N/2$. To further decrease the number of renegotiations the same approach is iterated over coarse time scales. Assume that x_1 is the number of bits arrived within interval *(t, t+Δ)*, and x_2 within *(t+ Δ, t+2Δ)*, and that r_a is the service rate during

each time interval such that $x_2 < r_a < x_1$. Considering equal buffering and under-utilization costs, the goal is to find the best possible r_a minimizing both costs simultaneously. Under-utilization cost C_u can be expressed as in (3.3).

$$C_u = | r_a - x_2 | - | r_a - x_1 | \tag{3.3}$$

where the term $| r_a - x_1 |$ gives the buffering cost C_b within *(t, t+Δ.)*. The best allocation would be the one which keeps both costs at the same level, that is $C_u = C_b$. After solving this equality for r_a,

$$r_a = \frac{x_1 + x_2}{2} \tag{3.4}$$

The difference operation $\frac{x_1 - x_2}{2}$ gives the deviation from the optimum allocation towards both x_1 and x_2, that is $\alpha = \beta$.

3.2 Multi-resolution Analysis and Energy Distribution

The new approach decomposes the time series traffic data, each element of which consists of arrival rate information, into different frequency bands. This method separates low and high frequency components in the arrival process. The energy distribution in each sub-band frequency informs us of the contribution of these components to the main traffic pattern. This information is used as a feedback parameter in prediction of the new arrival rate. Assume a vector $\underline{X}_k = [X(n-M+1)\ X(n-M+2) \ldots X(n)]$ at any time instant n where k is the time scale and M an integer. Each element $X(i)$ stands for the number of bits received in time slot i. Any two consecutive bit arrival rate information can be identified by their sum and difference. The difference operator reveals sharp changes in

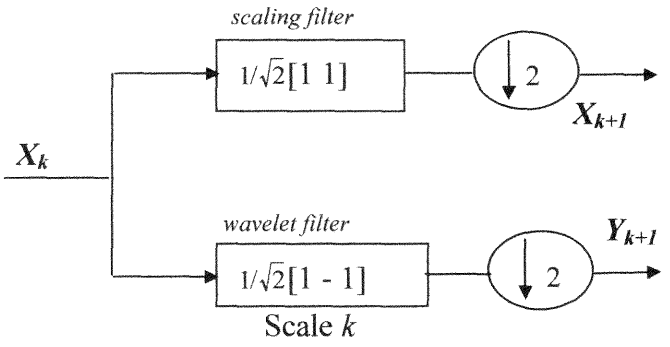


Figure 3.2 An illustration of the first scale filtering in a dyadic tree structure with Haar wavelet coefficients.

the arrival rate. Arrival rate vector $\underline{X}_{k+1}$ consisting of M consecutive time slots when represented at time scale $k+1$ *is*

$$\underline{X}_{k+1} = 1/2[X(n-M+1)+X(n-M+2)\ X(n-M+3)+X(n-M+4)....\ X(n-1)+X(n)] \tag{3.5}$$

Difference of the arrivals between two consecutive time slots is denoted by vector $\underline{Y}_{k+1}$ such that

$$\underline{Y}_{k+1} = 1/2[X(n-M+1)-X(n-M+2)\ X(n-M+3)-X(n-M+4)....\ X(n-1)-X(n)] \tag{3.6}$$

It is easily seen that

$$\underline{X}_{k+1}(i) = 0.5(\underline{X}_k(2i-1)+\underline{X}_k(2i)) \tag{3.7}$$

$$\underline{Y}_{k+1}(i) = 0.5(\underline{X}_k(2i-1)-\underline{X}_k(2i)) \tag{3.8}$$

The subject of interest is the dynamic behavior of the traffic that manifests itself through the differences among neighboring samples. (3.7) and (3.8) imply the scaling and wavelet transform coefficients of Haar wavelet at scale k with only difference being the value of the constant multiplier. The scaling and wavelet coefficient vectors of the Haar wavelet are $\phi = [1/\sqrt{2}\ \ 1/\sqrt{2}]$ and $\varphi = [1/\sqrt{2}\ \ -1/\sqrt{2}]$ respectively (Fig. 3.2). It is true that for $\forall i,j$,

traffic bit arrival rate process is positive, that is all $\underline{X}_i(j) \geq 0$. Wavelet domain modeling of positive processes requires the constraint that a positive output is ensured. To guarantee the constraint that the process is positive, the sufficient and necessary condition is $|\underline{Y}_i(j)| \leq \underline{X}_i(j)$. The Haar wavelet provides this constraint, but not always Daubechies does. The proof is easily seen by conforming (3.7) and (3.8) to the Haar wavelet transform such that

$$\underline{X}_k(2i-1) = 2^{-1/2}(\underline{X}_{k+1}(i) + \underline{Y}_{k+1}(i)) \tag{3.9}$$

$$\underline{X}_k(2i) = 2^{-1/2}(\underline{X}_{k+1}(i) - \underline{Y}_{k+1}(i)) \tag{3.10}$$

The Haar wavelet coefficients of a stationary signal will be identically distributed with $E[\underline{Y}_i(j)] = 0$ for $\forall i$. Passing the original traffic data $\underline{X}_k$ through scaling and wavelet filters in the dyadic tree returns the coarse component, $\underline{X}_{k+1}$, and, the details, $\underline{Y}_{k+1}$, of the original trace. Therefore, using a discrete Haar wavelet transform on the measured arrival rate data is a strong tool to reveal irregularities and sharp changes in traffic behavior. Having $\boldsymbol{R}$ as the MxM wavelet transform matrix composed of parameters of vectors ϕ and φ, and $\boldsymbol{X}$ as the vector data with length M, the wavelet transform operation can be expressed as $W=X.\boldsymbol{R}$ *[22, 29]* where W is the wavelet transform vector with size M. Energy of a stochastic process $\boldsymbol{X}$ at scale k can be found in (3.11).

$$E_k = \sum_{n=2^{k'-1}+1}^{2^{k'}} |\underline{W}(n)|^2 \tag{3.11}$$

where k: scale index and $k' = \log_2 size(X) - k + 1$

3.3 Self-similarity and Sub-band Energies

Let X denote a finite wide-sense stationary long-range dependent process with Hurst parameter H where $1/2<H<1$. The time series $X=[X_1, X_2, ...X_n]$ is said to be self-similar if the process is covariance stationary. It means the process has constant mean and finite constant variance and the corresponding aggregate process has the same correlation structure as the original process or follows asymptotically the correlation structure of the original process over large intervals. The spectral density $S_X(w)$ of X follows a power law near the origin [4, 5].

$$S_X(w) \propto c|w|^{1-2H} \text{ as } w \to \infty \tag{3.12}$$

where c is a finite positive constant. For selfsimilar signals, with $1/w$ behavior, the expectation of the energy within a given bandwidth 2^{-j} around frequency $2^{-j}w$ is given by

$$E[E_j] = \int S_X(w) 2^j \left|\hat{\varphi}(2^j w)\right|^2 dw \tag{3.13}$$

where $\hat{\varphi}(w)$ is the Fourier transform of $\varphi(n)$ [39]. It is also shown in [39] that

$$log_2 E[E_j] = log_2\left(\frac{1}{N_j}\sum_n \left|Y_j(n)\right|^2\right) = (1-2H)j + C \tag{3.14}$$

N_k stands for the number of elements in $\underline{Y}_j$ at filter-bank scale j. It is clear from (3.14) that energy distribution at each scale is related to the level of selfsimilarity of the traffic data. The higher the degree of selfsimilarity, from the perspective of queuing performance, the longer buffer is needed [47 and references therein]. Energy distribution in multi-resolution analysis provides information about the level of H. Therefore consideration of energy contents of a signal at different time scales in dynamic bandwidth allocation algorithms brings out a strong advantage over other methods to prevent the

impacts of traffic selfsimilarity. Because of the drawbacks mentioned in Chapter 1 of the wavelet transform when looked at the energy per scale and Hurst parameter relation, usage of (3.14) is not feasible for real time resource prediction and management. It requires a long traffic trace with enough sample size to be buffered for analysis within more than 8 dyadic tree branches. Our approach therefore stays away from this drawback by not directly looking at the energy distribution vs. Hurst parameter relation, but second order statistical properties of the energy distribution of the analyzed trace.

3.4 Multi-Resolution Algorithm Design

3.4.1 General Theory and Basic Properties of the Energy Vector

In order for the multi-resolution algorithm to respond to sudden changes in traffic pattern, window size with 8 samples is deployed. The area covered by each window is called X and the vector output of the wavelet transform of X is a "wavelet unit", W such that $W = RX = [w_1 w_2 w_3 w_4 w_5 w_6 w_7 w_8]$. The transform coefficients and their relation to the multi-resolution scaling is presented in Fig. 3.3. Energy in each scale can be found by applying (3.11) to the data vector W. scale 1 reveals the highest frequency detail within the original traffic data and the detail is expressed by four coefficients. The detail in scale index 2 is assigned to only two coefficients. The first element in vector W stands for the data content in coarsest scale. The sum of the energies in each frequency band is bounded above by the total energy of original data X. Strong empirical evidence contends that the wavelet coefficients from 1/f processes are weakly correlated both along and across the scales. In [39] it is shown for the Haar wavelet, the variance progression of the wavelet transform of Fractional Gaussian noise satisfies $var(W_k(n)) \propto 2^{-k(2H-1)}$. The ratio of the

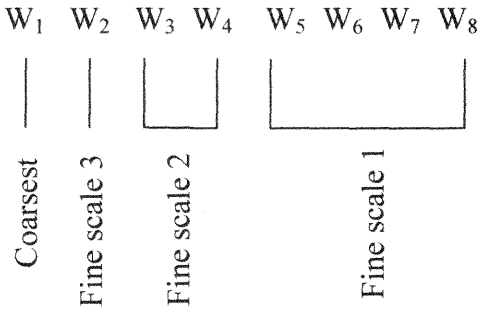


Figure 3.3 Wavelet transformed data vector and transform coefficients at each scale

energies in two subsequent scales can then be related to the Hurst parameter as $\gamma = \frac{var(W_k(n))}{var(W_{k+1}(n))} = 2^{2H-1}$, [48]. The stability of the arrival rate is proportionally related to the increase in this ratio.

Assume that $\boldsymbol{X}$ is a vector of size N of which each element represents bit arrival rate in time slots (Fig. 3.4). We introduce the following properties for the Haar wavelet transformed data vector which are used in dynamic bandwidth allocation.

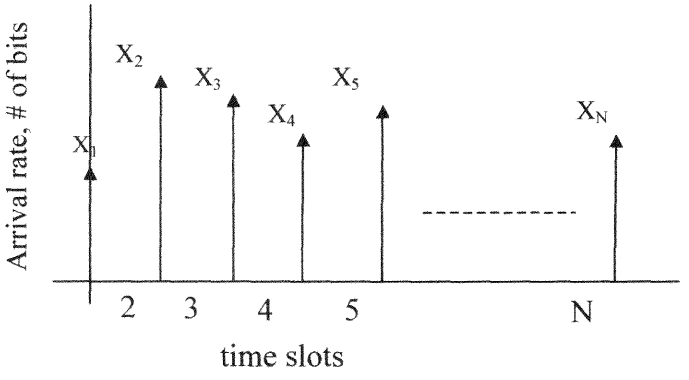


Figure 3.4 Illustration of the increments of packet counting process.

Property-I

If for $\forall i, j, X(i)=X(j)$ where $0<i, j<N, i \neq j$ and $i, j \in Z^{+}$, energy content in each high frequency band becomes zero and traffic is considered to be constant bit rate (CBR) throughout the measurement interval. In this case the entire energy of the input data is present within the lowest frequency band. The minimum bandwidth to allocate to this traffic must be equal to $X(i)$. We call this rate a DC source rate.

Property-II

If X is heavy tailed according to a Pareto distribution with shape parameter α, and location parameter s, a new process X_{new} generated by $(X - X_{min})$ also has the same shape parameter where X_{min} is the minimum of X.

Proof: Assume $P\{X<x\}=1-(\frac{k}{x})^{\alpha}$. Let Y be a new random process such that $Y=X-a$ where a is a constant. $P\{Y \leq y\}=P\{X<y+a\}$. The cumulative distribution function of Y is $F_y(y)=F_x(y+a)=1-(\frac{k}{y+a})^{\alpha}$. As a result, both X and Y has the same level of self-similarity, however different bandwidth demands. This property is given to point out that in wavelet transformation we use X-X_{DC} as an input, and this constant subtracting from the original data does not impact the self-similar characteristics of the original signal.

Property-III

If the elements of X show an increasing characteristic in time such that $X(n) > X(n-1)$ where $1<n< (N+1)$, the energy distribution in each frequency band increases from finest to a coarser scale, being the lowest in the highest frequency region, $E[E_{j+1}] > E[E_j]$.

Property-IV

A decrease in the coarse scale energy while the fine scale energies are stable is an indicator of a linear decrease in traffic volume.

Property-V

Assume that V_i stands for the variance of the energies among k scales in time slot i, M_i for the mean value of energy contents, and U_i for the utilization. The following statements hold:

- $V_{i+1} > V_i$ and $M_{i+1} < M_i \Rightarrow U_{i+1} < U_i$

  (Representative of possibly approaching OFF interval or departing flows (in case of an aggregate traffic))

- $V_{i+1} > V_i$ and $M_{i+1} > M_i \Rightarrow U_{i+1} > U_i$

  (Representative of added flows to an aggregate traffic or an approaching ON interval)

- $V_{i+1} < V_i$ and $M_{i+1} < M_i \Rightarrow U_{i+1} < U_i$

  (Representative of traffic getting smoother than its current pattern)

- $V_{i+1} < V_i$ and $M_{i+1} > M_i \Rightarrow U_{i+1} >> U_i$

  (Representative of traffic getting smoother with increasing utilization)

3.4.2 Bandwidth Decision Unit (BDU) & Four Different Decision Types

BDU is used to analyze the wavelet transformed traffic data. It takes the energy vector as an input, and also some external feedback parameters such as momentary queue size (Fig. 3.5). BDU keeps track of the second order statistical properties of the energy vector and

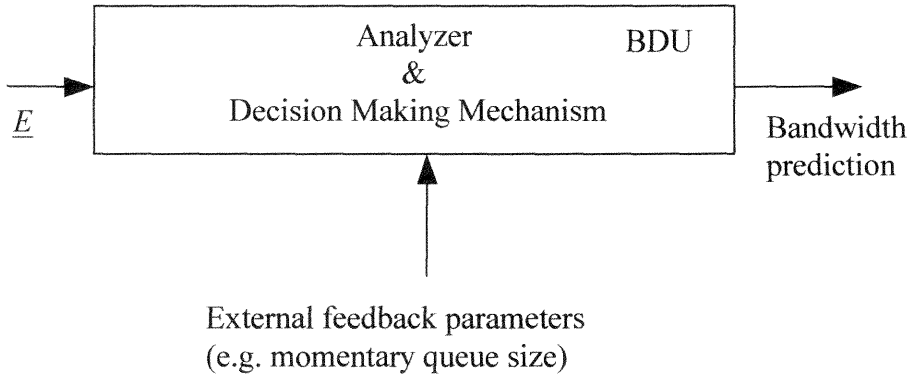


Figure 3.5 Analyzer & Decision Making Mechanism (DMM) having the energy in each scale as a feedback parameter and returning the new bandwidth to allocate.

computes the new variance and mean value of energy distribution among all the frequency sub-bands. The new values are compared with the previous values maintained in a state table. Under the highlight of the properties of energy vector explained in the previous section, one of the four modes are activated within the unit and a new bandwidth decision is released to the renegotiation control unit (RCU) of which the functionality is explained in the next chapter.

Pseudecode representation of the decision mechanism after performing the wavelet transformation is given below:

```
if      V_new ≥ V_pre
        if M_new ≥ M_pre
                {Perform method-IV}
        else
                {Perform method-II}
        end
else if M_new < M_pre
                {Perform method-I}
        else
                {Perform method-III}
end
```

In Figure 3.6, the architectural diagram of these four methods using the signal energies in each sub-band and momentary queue size as illustrated. The algorithm first filters out the DC component in traffic measurements. This DC value is taken as the lower bound for the bandwidth allocation in the next time slot to prevent the application from bandwidth starvation. The signal at the output of the DC filter, $\tilde{X}$, consists of low and high frequency components. The signal is fed into a filter bank in which high pass filter is composed of Haar wavelet coefficients and low pass filter is of Haar wavelet scaling coefficients. The signal $\tilde{X}$ is analyzed by decomposing it into three high frequency sub-bands based on the recommendation in Chapter 2.

Energies in each sub band are used as representatives of the traffic volume within the related frequency band. The traffic volume that generates the maximum energy is added as the second component to the bandwidth allocation in the next time slot. Because, it is the underestimation that may cause the buffer congestion while the overestimation can only result in the underutilization of the available bandwidth, we take

the contribution of maximum energy content in each adaptation interval for the bandwidth allocation. Four different methods of bandwidth prediction decisions within the DMM are as follows:

X is the data vector before low pass filtering, R the wavelet transformation matrix, and $W=X.R$ is the wavelet transform coefficients vector. If the wavelet unit consists of M samples where $M=2^k$, k>0 and $\in Z^+$, the matrix R has the size MxM. The wavelet transformation is performed over data vector X in moving blocks of size M with $(M-1)$ overlapping samples. In Fig. 3.6 the new bandwidth allocation is based on

$$BW(n+1) = X_{DC}(n) + \sqrt{E_{MAX}(n)} + queue\_size(n) \tag{3.15}$$

where n is the index of time slot. We modify the decision block in Fig. 3.6 such that the queue size information is not needed for bandwidth renegotiation without degrading the queuing performance and without increasing the capacity underutilization. Here we introduce four additional wavelet-energy approaches and compare their performances.

Method-I

This method dismisses the queue size information from (3.15) and compensates the new bandwidth allocation by including a component that provides bandwidth contribution with energy equivalent to the superposition of all the energies.

$$BW(n+1) = X_{DC}(n) + \sqrt{\sum_{i=1}^{K} \underline{E}_n(i)} \tag{3.16}$$

where K is the number of scales the data is decomposed into.

Note: $\underline{E}_n(i)$: Energy in sub-band i at discrete time instant n.

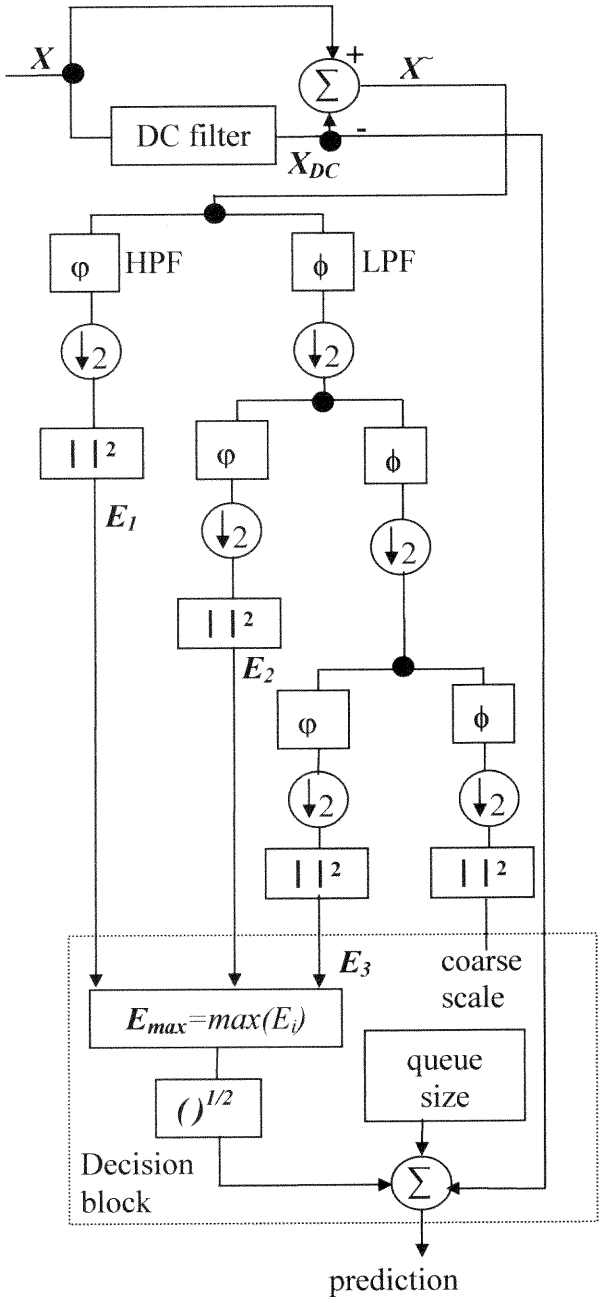


Figure 3.6 Block diagram of the wavelet decomposition technique in dynamic bandwidth allocation

Method-II

Being different from *method-I*, the bandwidth compensation is provided as the sum of K components each of which is representing a bandwidth contribution such that the energy of each component is the same as the energy in one of the sub-bands.

$$BW(n+1) = X_{DC}(n) + \sum_{i-1}^{K} \sqrt{\underline{E}_n(i)} \tag{3.17}$$

Method-III

It is illustrated before in Fig.3.6 that $\bar{X}=X-X_{DC}$. Let us denote $\underline{\bar{W}}_k$ as the wavelet transform of k^{th} block of $\bar{X}$, and $\underline{W}_k$ of the k^{th} block of X, the data unit before DC filtering. The first element of vector W_k^- is $\underline{\bar{W}}_k(1) = \frac{1}{M}\sum_{i=1}^{M}\underline{\bar{X}}_k(i)$, and similarly $\underline{W}_k(1) = \frac{1}{M}\sum_{i=1}^{M}\underline{X}_k(i)$. It is easily shown that $\underline{\bar{W}}_k(1)=\underline{W}_k(1)-X_{DC}$.

This method replaces X_{DC} in (3.16) with $\underline{W}_k(1)$ which is the average of X in k^{th} block. The condition that $\underline{W}_k(1)>X_{DC}$ is always true. Therefore, it is clear that this approach allocates more bandwidth in each renegotiation than *method-I* does. We compare the percentage improvement in queue size performance by the marginal increase in bandwidth reservation with *method-I.*

Method-IV

This method replaces X_{DC} in (3.17) with $\underline{W}_k(1)$. It intuitively causes lower utilization than *method-II* does.

3.5 Results and Discussion

3.5.1 Comparison of Wavelet Analysis with RLS and LMS

In this section, the drawbacks of LMS and RLS based predictors are presented as compared to the wavelet analysis. RLS algorithm's main disadvantage is its computational complexity. For real time applications, RLS is not proper to use. On the other hand, LMS is an adaptive approach and it does not require prior knowledge of the autocorrelation structure of predicted data. Therefore, it can be used as an online algorithm. The algorithm starts with an initial estimation of the filter coefficients w*(0)*. However, when the LMS algorithm is applied on a test sequence shown in Fig. 3.7, the finding has revealed that LMS fails to follow the level changes, and creates large spikes in both falling and rising edge of the test pattern meaning that these large spikes maps onto the over or underestimation of the traffic level in the next interval. Even though it recovers the utilization level after a spike, the high value of the spike may force the renegotiator unit to start a new renegotiation request and cycle, though it is not needed. Because of this impact on the renegotiation requests, we hesitate to deploy LMS method in real time bandwidth allocation system with bandwidth renegotiation capability. The wavelet method does not create spikes, and further it follows the traffic level with a smooth trend. Wavelet based algorithm has a similar transition performance as RLS. However, when the computational complexities are subject to discuss, RLS is not preferred.

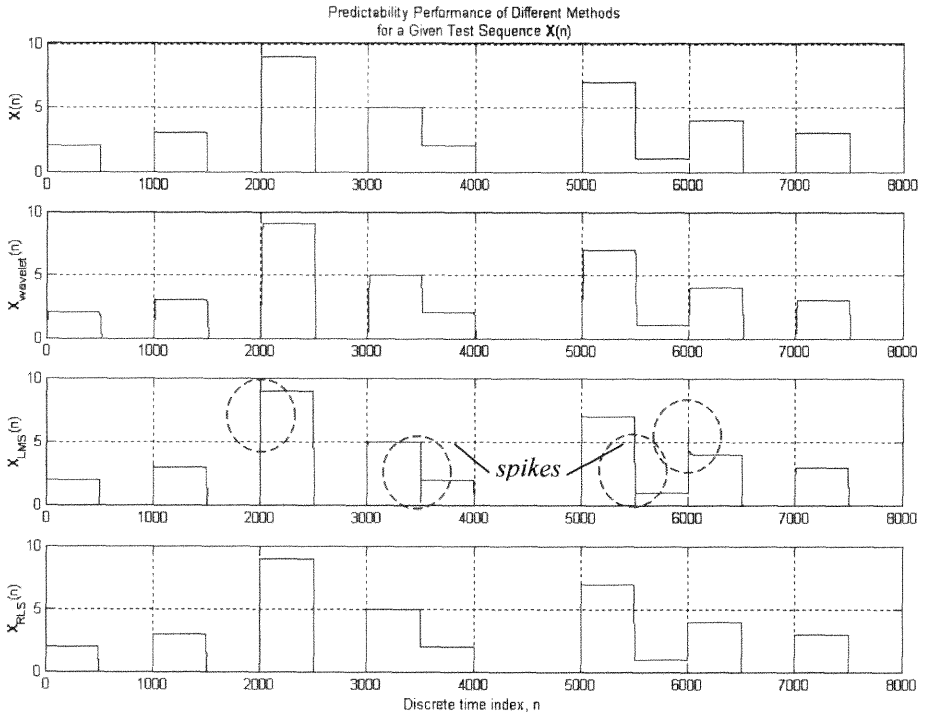


Figure 3.7 Predictability performance of wavelet, LMS and RLS based algorithms.

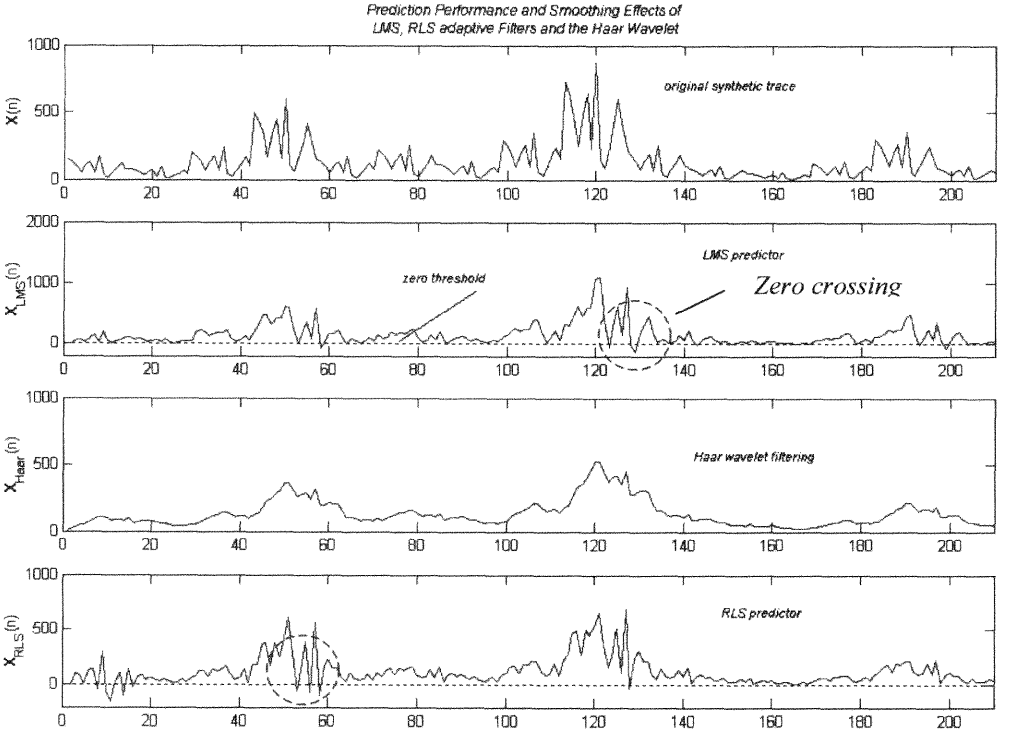


Figure 3.8 Zero crossing effect in LMS and RLS based algorithms

Traffic bit arrival information process at time interval n, $X(n)$ is a positive process and therefore, the prediction of $X(n+1)$, $\hat{X}(n+1)$ must also be positive. As illustrated in Fig. 3.8, RLS and LMS based predictors violate this property and returns a negative prediction result. Negative predictions also cause RCU to malfunction and to start a renegotiation cycle that may not be needed. These extra renegotiation requests increase the number of granted renegotiations. To summarize, wavelet based method is superior to RLS and LMS based predictors because of its simplicity and suitability for positive signal constraint.

3.5.2 Queue Size Performance Results

First, the *wavelet-energy* method is compared to the other classic approaches under the constraint that the number of renegotiations are kept the same, that is one renegotiation every sec. The comparison is made in terms of average queue size and mean square bandwidth allocation error. An infinite buffer is assumed, and therefore, no packet loss due to a buffer overflow. In every time slot that maps onto a 1s interval, the algorithm returns a bandwidth prediction.

The subject input traffic is of heavy WEB browsing activities generated in the OPNET environment with peak/mean 17.03. In Fig.3.9a the traffic is shown with vertical axis representing the number of bits received and horizontal axis the time index with an interval of 1s. Fig. 3.9b illustrates the simulation of the queuing performances of classic *previous*, *average*, *previos+average*, and *previous+queue size* and wavelet-energy bandwidth allocation methods.

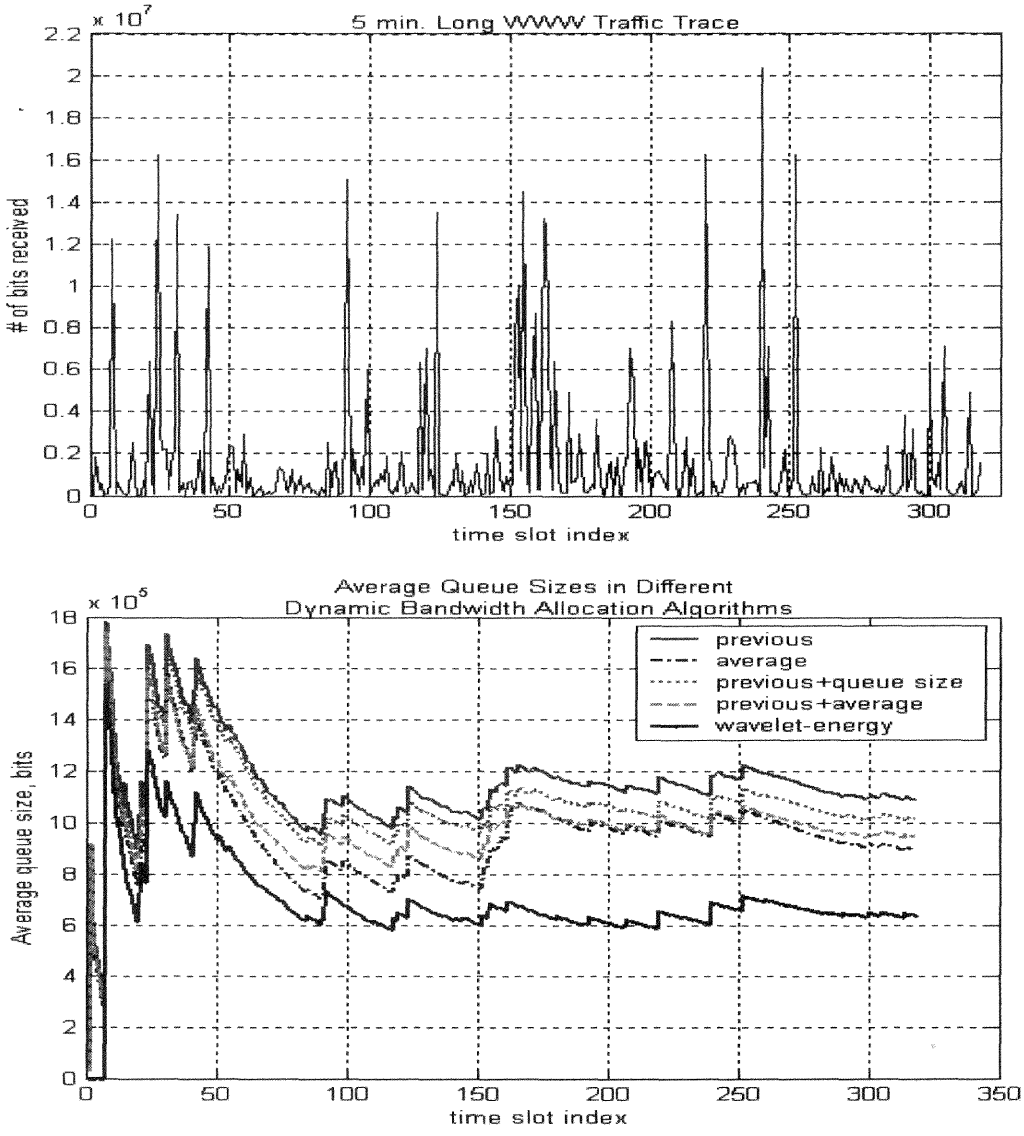


Figure 3.9 *(a)* A 5 min long synthetic WWW traffic trace with *peak/mean=17.03*. *(b)* Queuing performance of different bandwidth allocation algorithms.

The most conservative allocation method is to reserve bandwidth by the traffic's peak rate, which would lead to the excessive use of transmission capacity with zero buffer size. A real time peak rate algorithm would require an overestimation of the traffic by a significant percentage. It also results in zero loss rates. Small but nonzero losses and delays are acceptable for typical multimedia applications. The ability to manage delay

requirements of applications together with effective use of network resources is important. Therefore, the peak rate algorithm is not comparable to the other methods mentioned in this paper.

Performance of *average* strictly depends on the number of past samples, which are averaged to predict the bandwidth demand in the next time slot. Therefore, correct determination of this parameter is a must. Recalculation of this parameter is needed for different traffic traces with different characteristics, which makes this method inept. It is quite clear in Fig. 3.9b that *Previous+average* outperforms *previous*. However, this method involves the correct determination of two parameters: the number of samples to average and the weight ratios between previous allocation and the result of averaging. A large improvement in queuing performance by the deployment of *wavelet-energy* information against other classic algorithms is observed. The *wavelet-energy* algorithm converges faster than all others, results in less queue size, and accordingly smaller queuing delays.

Average inefficiency, $\phi(a,b)$, in resource allocation within time interval $[a,b]$ can be analytically given by the ratio of the area between the curves of $x(t)$, the original traffic data and $y(t)$, the allocated bandwidth function, to the length of the time interval, $(b-a)$, (3.18).

$$\int_0^{\infty} \phi_i(a,b) = \frac{1}{N} \frac{\sum_{i=1}^{N} (y(i)-x(i))^2}{b-a} \tag{3.18}$$

where i is the index of each time slot and N the total number of time slots. This performance metric has acceptance in the literature [35]. However, it is not a good indicator of queue size performance of an algorithm. A method with higher $\phi(a,b)$, may

result in smaller queue sizes. Assuming the updating period, which is *(b-a)*, is 1s, (3.18) becomes the MSE of bandwidth allocation process. Table 3.1 denotes the mean square bandwidth allocation error of each method.

Table 3.1 Mean Square Bandwidth Allocation Error (MSBAE)

Method	MSBAE (kbits)	Avg. Queue Size
Previous+queue size	362	1142Kb
Wavelet-energy	302	703Kb
Previous	239	1198Kb
Previous+average	203	1083Kb
Average	174	939Kb

The values in Table 3.1 are obtained by applying the methods to WWW traffic with self-similarity parameter H=0.52. According to Table 3.1 wavelet-energy method is not the best to have a minimum MSBAE. However, it results in the smallest queue size among the others. The difference in MSBAEs of *average* and *wavelet-energy* is about 134kb, that is twice less than the improvement in queue lengths gained by deployment of the wavelet-energy method. Four different wavelet-energy decision types are also compared in terms of their utilization and queue size performances. Table.3.2 below contains queue size and utilization performances of these four wavelet methods for the same synthetic trace.

Table 3.2 Statistical Properties of Bandwidth Utilization of Four Different Wavelet Energy Methods.

	Mean utilization	Avg. Queue Size (Nrm)	PMR
I	0.98	0.121	1.58
II	0.84	0.051	1.51
III	0.75	0.022	1.41
IV	0.65	0.008	1.44

Note: *PMR-Peak to Mean Ratio*

Assume BW_i is a bandwidth amount allocated by wavelet method i where $i=1,2,3.4$. Considering the fact that, from (16) and (17), $BW_4 > BW_3 > BW_2 > BW_1$, the highest utilization and the highest queue size are expected from *method-I*, and the lowest utilization and the lowest queue size from *method-IV*. As shown in Fig. 3.10 *method-I* has the worst queuing performance and *method-IV* the best.

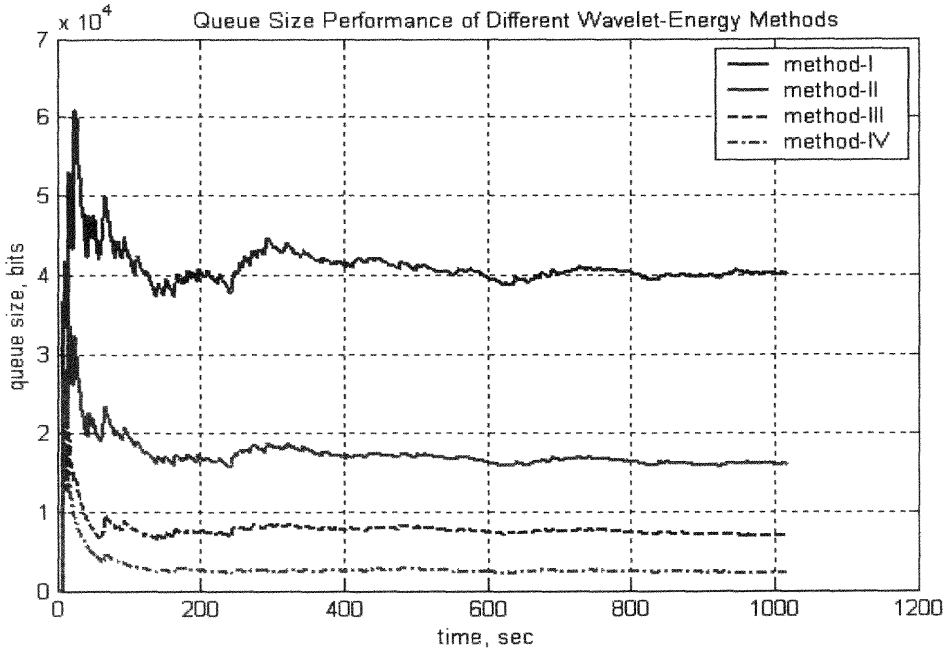


Figure 3.10 Queuing performances of four different wavelet-energy approaches in dynamic bandwidth allocation to applications traffic.

The wavelet methods are applied to VBR Star-Wars trace. Fig. 3.11 denotes the utilizations and average queue sizes achieved by each method. Average utilization from has a negative slope, being minimum for *method-IV* and maximum for *method-1* as expected. However, as going from *method-I* through *method-IV* the marginal gain in average queue size is not linear. In a scenario such that the buffering cost is higher than the under-utilization cost, and a low queue size is required, deployment of *method-IV*

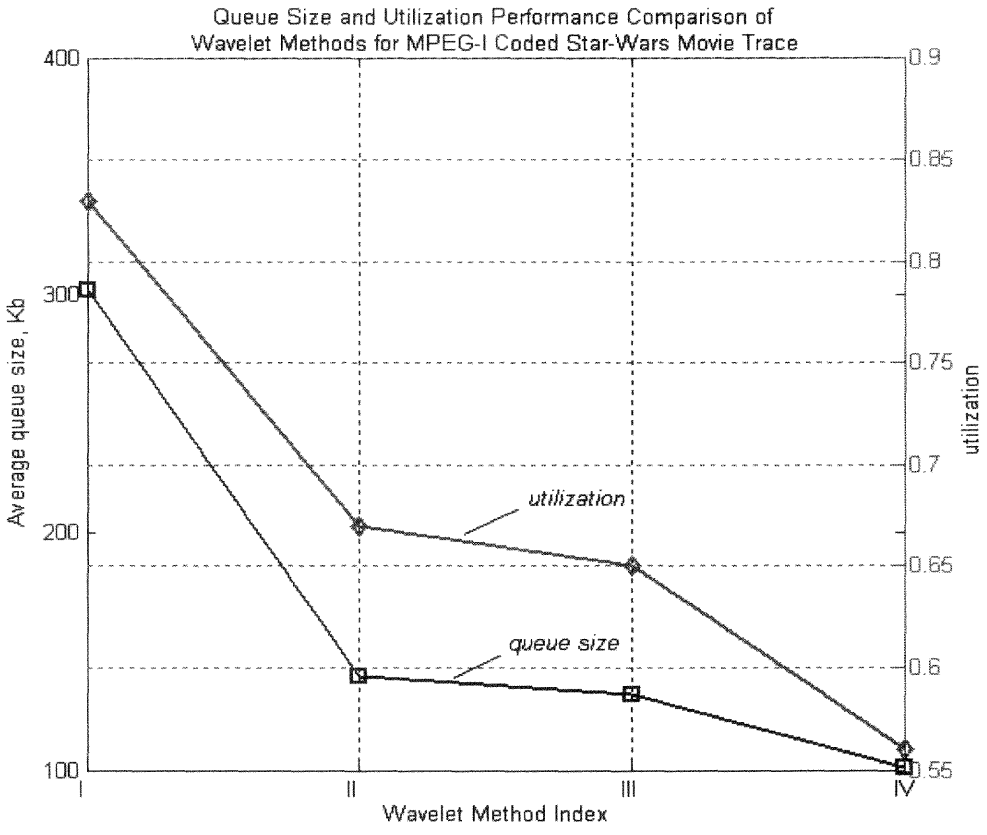


Figure 3.11 Utilization and queue size trade-off in the introduced wavelet-energy methods when applied to Star-Wars trace.

would be preferred to the other methods. Similarly, if the capacity is required to be highly utilized, *method-I* would perform better. If the costs are time variant, switching from one method to another would increase the achievable utilization level and decrease the queue size.

3.6 Summary

The wavelet-energy based bandwidth predictor is an easy to implement method, and therefore suitable for online prediction based bandwidth allocations. Its basic purpose is to split high frequency and low frequency fluctuations of input traffic, and compare the

signal energy volumes in different frequency sub-bands by looking into the traffic energy distribution. It does have neither the "negative prediction" drawbacks of LMS and RLS methods, nor their computational complexities. On the other hand, it achieves low queue sizes at the expense of relatively low utilization level when compared with traditional bandwidth predictors such as *average*, *previous*, *average+previous, RLS*, *LMS*. Even though this resulting low utilization seems to be a disadvantage of the wavelet analysis, the main purpose is to support bandwidth renegotiation unit with proper feedback information. The determination of an actual bandwidth amount to allocate for VBR traffic is performed by a Renegotiation Control Unit (RCU). The new RCU design explained in Chapter 4 increases the utilization level returned by BDU, and also maintains given queue size constraints.

CHAPTER IV

MINIMIZATION OF BANDWIDTH RENEGOTIATIONS

In this chapter, a new scheme is introduced for real time asynchronous bandwidth renegotiation for variable bit rate (VBR) traffic. This new method presents a fine-tuning functionality for the dynamic bandwidth allocation scheme designed in Chapter 3, and determines the optimum bandwidth renegotiation time and bandwidth amount to allocate to a VBR traffic source as illustrated in Figure 4.1 by minimizing predefined cost functions. Unlike the bandwidth allocation with maximum number of renegotiations, renegotiation-cost effective solution produces a step pattern. The Renegotiation Step Size (RSS) and the Inter Renegotiation Interval (IRI) are traffic dependent time varying parameters. The traffic rate predictor designed using wavelets in Chapter 3 provides a feedback to the system. The results show that this new scheme minimizes both under-

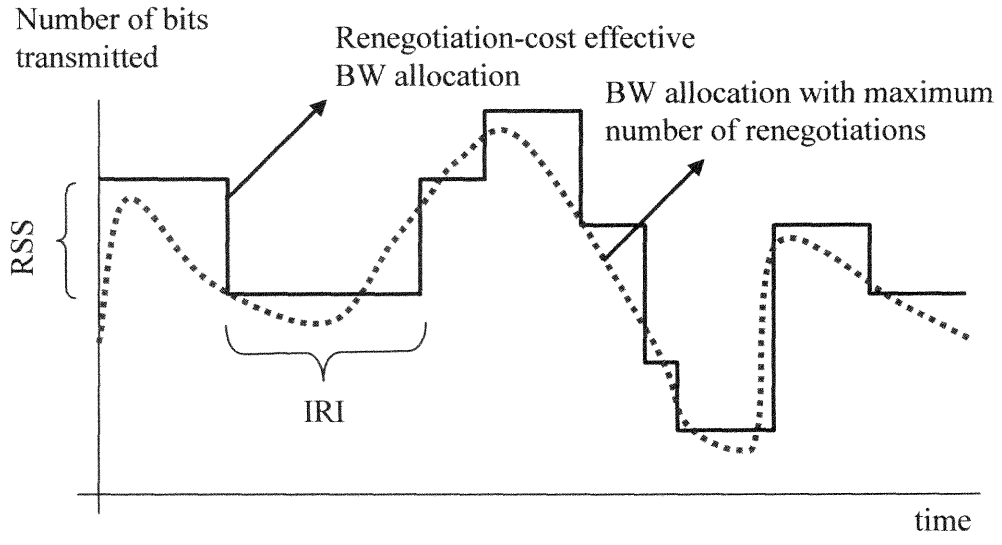


Figure 4.1 An illustration of renegotiated bandwidth allocation.

utilization of the available capacity and queuing delays. The new method can be deployed in Priority Queuing disciplines to dynamically manage the resources allotted to each priority levels.

ATM networks provide connection-oriented services with guaranteed bandwidth. in order to carry an IP datagram in such networks, a virtual circuit (VC) has to be setup with an associated bandwidth requirement. Once a VC is setup, the adaptation layer has to decide how long to keep the VC active with the initial bandwidth assignment. If the rate of the incoming packets matches the specified bandwidth allocation, the VC is kept active [26, 27, 41]. However, if packets arrive at a higher or lower rate, there is a need to readjust the allocated resource or even to terminate the VC. Periodic algorithms adjust the bandwidth allocation in fixed time intervals. On the other hand, adaptive algorithms respond whenever a change is necessary as long as the updating process is not frequent. Readjusting can be done in two different ways namely by either closing the existing VC and setting-up a new one with new allocation and by changing the allocation of the current VC in lieu of terminating it [26]. The latter option, if preferred, must be supportable by the network. Indeed, the Q.2963 series of recommendations belongs to the DSS 2 family of ITU-T Recommendations, and specifies the procedure of the modification of traffic parameters of a call/connection in the active state. Recommendation Q.2963.3 defines the procedure of the ATM Traffic Descriptor modification with renegotiation that is equivalent to that specified in Recommendation Q.2962. Therefore, it is assumed that network provides such a support that bandwidth allocation can be updated without terminating a VC.

Each renegotiation process involves a signaling between the network and the source. The rate of renegotiation is a trade-off between signaling overhead and bandwidth utilization. High renegotiation frequency loads the network with heavy overhead. On the other hand, long inter-renegotiation intervals make the follow-up of the traffic bit rate pattern difficult. Renegotiation is only feasible in time scales of several seconds [37]. In [38], it is suggested that minimum of 1 sec and an average of 5 seconds or more for renegotiation is a good compromise. It is crucial that optimal number of bandwidth renegotiations must be performed under predetermined cost constraints such as underutilization ratio and packet/cell transmission delay.

The new end-to-end design brings a solution to the real time dynamic resource management problem. This system consists of two main parts: the Bandwidth Decision Unit (BDU) and the Renegotiation Control Unit, RCU. The BDU, as explained in Chapter 3, uses wavelets and signal energy distribution in frequency domain to compute the bandwidth demand of a source for the next discrete time slot. This chapter elaborates on the RCU.

The cost functions and strategies in determination of IRI and RSS are explained in Section 4.1. Performance results of the new optimized renegotiation method when applied to synthetic VBR bit stream are given in Section 4.2, together with the discussions. In Section 4.3, the RCU design is implemented to provision QoS to MPEG-1 Star-Wars traffic trace. The 0.99 queue size quantile and average utilization are compared with static bandwidth allocation and other methods in the literature.

4.1. Definitions of Cost Functions

In order to avoid either over-utilization (increasing queue size) or under-utilization, the bandwidth allocation should be dynamically updated to follow-up the temporal variations in the traffic pattern. However, each update (renegotiation) process involves a signaling between a network and a traffic source. High renegotiation frequency loads the network with heavy overhead. On the other hand, a long IRI decelerates the follow-up of the traffic bit rate pattern. The new method takes three cost functions into consideration together with the output of bandwidth prediction and allocation scheme given in Chapter 3. These three functions are *(i)* under utilization cost *(ii)* renegotiation cost and *(iii)* buffer size (delay) cost.

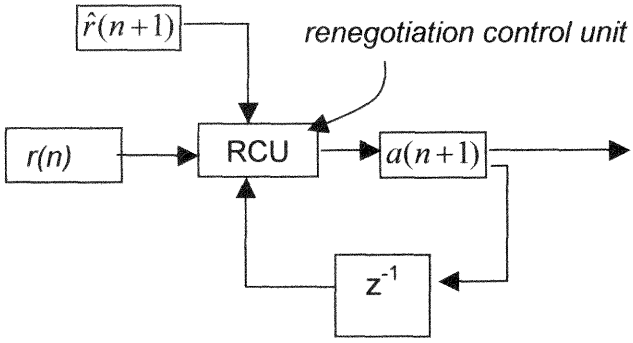


Figure 4.2 Renegotiotion control unit (RCU) with input and output parameters.

Let *a(n)* be the optimum renegotiated bandwidth, and *r(n)* be the traffic bit arrival information where *n* represents a time index. An optimum bandwidth allocation *a(n+1)* for real-time traffic for a future time slot *n+1* is determined, given a traffic bit arrival information *r(n)*, allocated bandwidth *a(n)*, and the prediction for the next time slot $\hat{r}(n+1)$ which is provided by the BDU as in Fig. 4.2.

In order to reach the optimum solution, a total cost function J is designed which includes costs of under utilization $u(n)$, under allocation (buffering) $b(n)$, and renegotiation $T(n)$. Buffering and under-utilization events are illustrated in Fig.4.3. Total cost function J is defined as

$$J = w_b b(n) + w_u u(n) + T(n) \tag{4.1}$$

The parameters w_b and w_u represents the unit cost (cost weight) shaping functions for under and over-utilization. If equal cost weight, w, is considered for both $b(n)$ and $u(n)$

$$J = w(e(n)) + T(n) \tag{4.2}$$

where $e(n)$ is the bandwidth error such that

$$e(n) = \hat{r}(n+1) - a(n) + \sum_{i=0}^{n} r(i) - a(i) \tag{4.3}$$

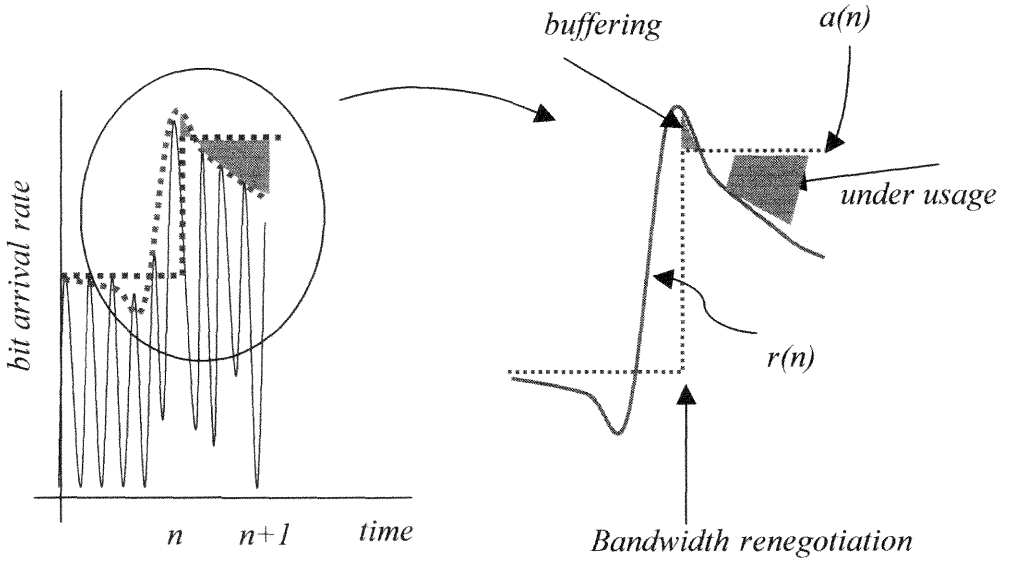


Figure 4.3 Areas of under-utilization and buffering with relation to $a(n)$ and $r(n)$.

The bandwidth error function *e(n)* is equal to *u(n)* in case of under-utilization, and equal to *b(n)* in case of under-allocation. It also includes the predicted bandwidth error for time *n+1* to include the effect of keeping the same bandwidth allocation level. The total cost *J* and its relation to the defined cost parameters are analytically represented in Fig. 4.4. It is clearly seen that the new method satisfies hard buffer constraint and relevantly delay requirement for an application via proper characterization of *w(e(n))*. The crossing point of *w(e(n))* with *T(n)* line maps onto the time instant to resume renegotiation process, and therefore possible queue size exceeding the given buffer size constraint is avoided. Also, the intersection point on the left side of the *J* axis limits the under-utilization level, and the system starts a new renegotiation cycle.

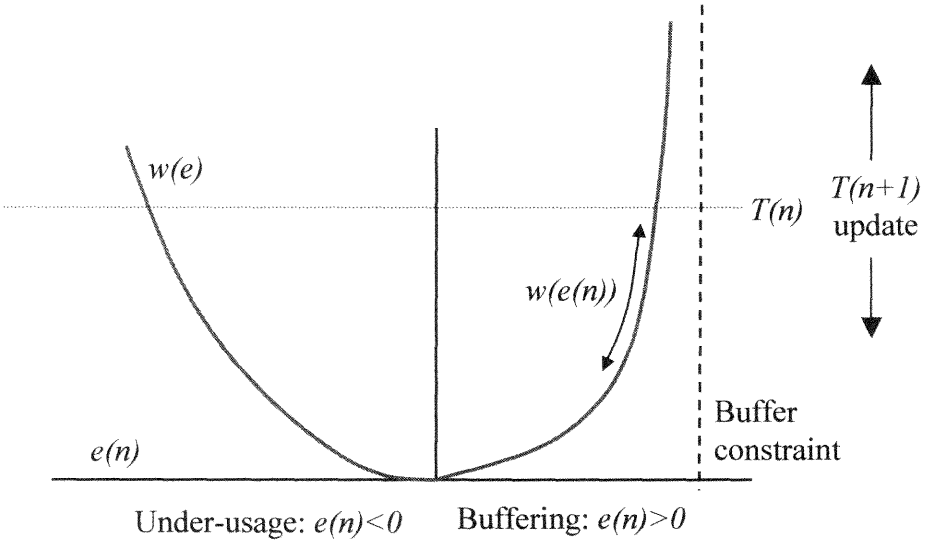


Figure 4.4 Analytic representation of the predefined cost functions.

Using separate cost terms for under-utilization and under-allocation enables us to adapt the optimization method for various types of platforms: Weighted Fair Queuing (WFQ) algorithms, ATM switches, etc. In a WFQ scheduler, weights assigned to delay

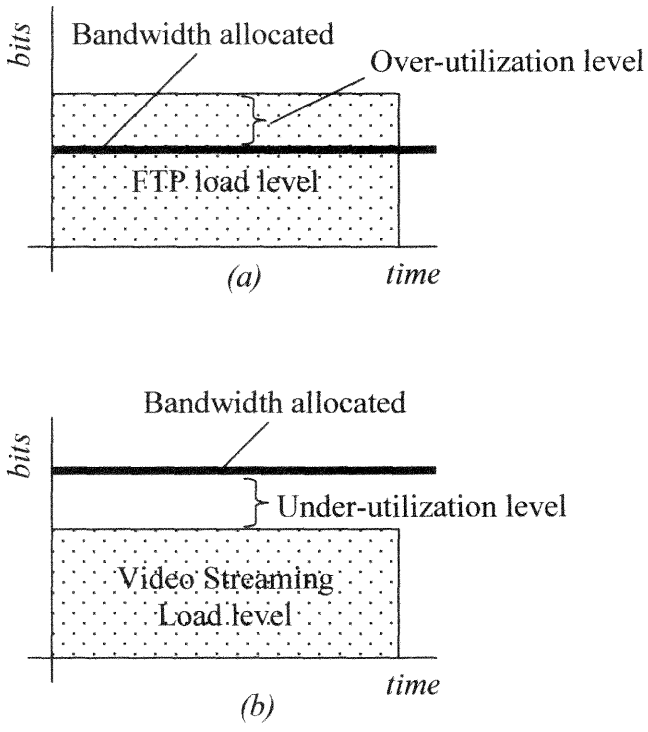


Figure 4.5 Illustration of the application dependent error controlling term suitable for WFQ (a) over-utilization level for FTP (b) under-utilization for video streaming.

sensitive applications (i.e. video streaming) might be controlled by under-utilization error term, while service weights for low priority applications such as FTP might be controlled by over-utilization error term as illustrated in Fig. 4.5. An under-allocation happens if the allocated bandwidth is not enough to handle the bit arrivals. In case of an under-allocation, the excess bits are buffered. The buffered bits are sent when the arrival bit rate is less than the allocated bandwidth, thus, there is available bandwidth to forward bits from buffer. An under-utilization $u(n)$ occurs, when the allocated bandwidth is greater than the bit arrival rate, and the buffer is empty. Therefore, the allocated bandwidth is not fully utilized.

One cost term can be given a higher weight than the others with respect to the changing network conditions, i.e. buffer cost can have nonlinear dependence on the

current queue size, cost of bandwidth can be alternating at the certain times of the day. The vectors $\underline{u}$ and $\underline{b}$ of size M consisting of the last M values of under-utilization and over-utilization levels are orthogonal to each other, that is $\underline{u} \perp \underline{b}$, because either of them occurs at a given time slot. Therefore, each moment of the total cost function J indeed contains only one of them in addition to the renegotiation cost term. Obviously, the bandwidth cost $w(e(n))$ corresponds to the under allocation cost if $e(n)$ is greater than zero, and under utilization cost otherwise.

Optimization of J outputs information about the next RSS and the time of activation. Assume that a renegotiation is needed for time slot $n+1$. RSS would be equal to $[a(n+1)-a(n+1-k)]$ where k is the number of time slots since the last renegotiation. Afterwards, $a(n+1)$ which minimizes the cost function is computed.

$$a(n+1) = \arg\min(J) = \arg\min[w(e(n)) + T(n)] \tag{4.4}$$

To understand the properties of minimization, let us investigate the impact of each cost term on J. It is worthwhile to realize that the renegotiation cost $T(n)$ should be high if there was another bandwidth renegotiation which occurred prior to time instant n, that is at time instant equal to $(n-\delta)$ where δ is a small interval. An increase in the value of δ means a long IRI since the last renegotiation time. Increasingδ lowers the cost of renegotiation, because renegotiation becomes more affordable. Therefore, the time difference δ between the current time and the last renegotiation time determines the magnitude of the cost function $T(n)$. Let ϕ be a fixed renegotiation cost per unit time. After a renegotiation process is complete, the cost to resume a new renegotiation in the next time slot would be maximum, $T_{\max}$. However, the cost would be decremented byϕ per time slot until a breaking point is reached. The "breaking or the crossing point" is

defined as the time at which renegotiation cost function and *w(e(n))* intersects. Since there would be *k* time slots passed after the last granted renegotiation request, the lessening in the renegotiation cost from time *(n+1-k)* to *n+1* is expressed as

$$T(n) = T_{\max} - \phi[\sum_{i=n-k+1}^{n+1} \delta(a(i) - a(i-1))] \tag{4.5}$$

where $\delta(m) = \begin{cases} 1, & m = 0 \\ 0, & m \neq 0 \end{cases}$. Changes in queue size and under-utilizations costs after each renegotiation process are illustrated in Fig. 4.6. In this example, a renegotiation is requested because of increasing queue-size. After the renegotiation at time instant t_2 If the bandwidth cost function *w(e(n))* becomes larger than renegotiation cost *T(n)* for the predicted traffic, it becomes advantageous to renegotiate to prevent *w(e(n)* from getting larger. If a time varying dynamic renegotiation cost function is not used, but a constant

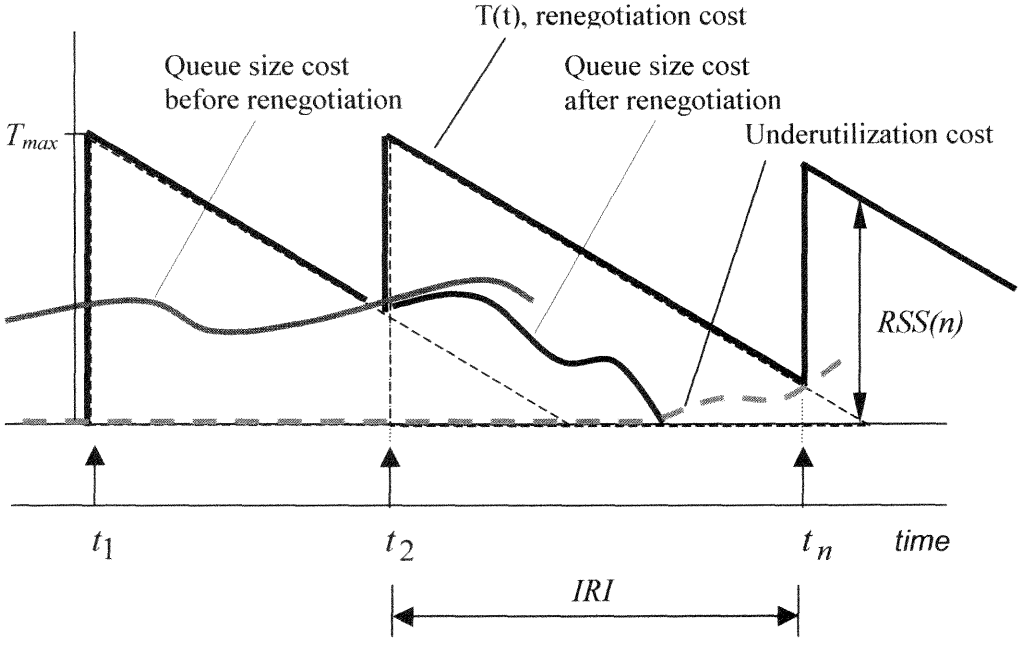


Figure 4.6 Queue size and underutilization costs with relation to renegotiation cost function and renegotiation times.

one such that *T(n)=const.* regardless of *n*, the number of renegotiations may multiply when the newly allocated bandwidth is incapable of reducing the under-allocation or under-utilization costs quickly, because *J* starts to fluctuate close to the decision boundary. Each time bandwidth cost function becomes higher than the renegotiation cost, a new renegotiation has to be resumed. This drives the system into instability. *w(e(n))* is formulated as a combination of exponential components with different exponents *K* and *L*.

$$w(e(n)) = \begin{cases} e^{e(n)K} & e(n) > 0 \\ e^{|e(n)L|} & e(n) \le 0 \end{cases} \tag{4.6}$$

K and *L* take their values depending on whether a strict control would be used for queue size or for under-utilization. The QoS parameter with higher priority would have a higher

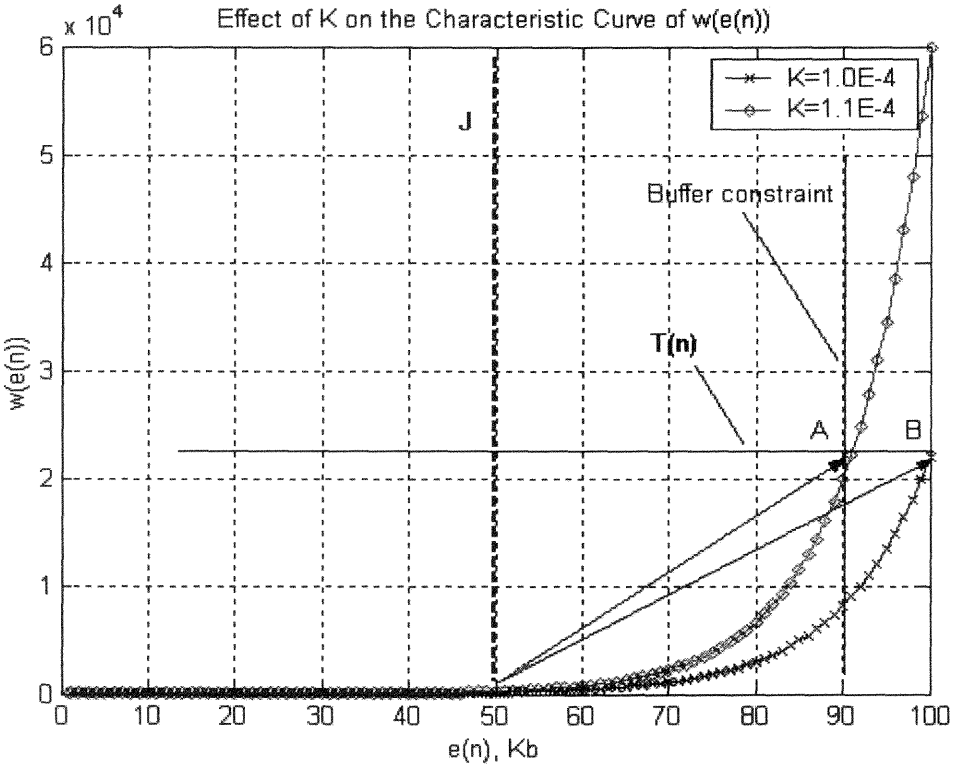


Figure 4.7 Illustration of the effect of exponent *K* on the characteristic of an under-allocation curve.

exponent (i.e. if queuing cost is higher than the cost of underutilizing a link, then $K>L$). The exponent K controls the asymptotical behavior of $w(e(n))$ to the right of J axis, while L to the left. Fig. 4.7 shows the effect of the selection of K on the characteristic of $w(e(n))$ for two different values of K, 0.0011 and 0.0010. As can be seen in the figure, there are two crossing points with $T(n)$ line: A and B. For the same buffer constraint ($e(n)$=90), $w(e(n))$ with smaller K initiates a renegotiation at point A. On the other hand, as K gets smaller $w(e(n))$ reaches the crossing point with $T(n))$ axis after exceeding the given buffer constraint. Therefore, it is crucial to select the exponents of $w(e(n))$ such that $w(e(n))$ does not exceed the buffer constraint before it crosses the renegotiation cost line in order to provide an applications traffic with strict buffer size and delay requirements. The same analogy applies to the selection of exponent L in order to limit under-utilization level.

After proper selection of K and L, renegotiation decision points are generalized as illustrated in Fig. 4.8 when $w(e(n))>T(n)$. At times $e(n)>0$, incoming bits are started to be buffered. When the cost due to buffering exceeds a current renegotiation cost, the RCU starts a new renegotiation, and increases the allocated bandwidth, that is $a(n+1)>a(n)$. As soon as the renegotiation is approved by the network (i.e. ATM networks) or WFQ scheduler (i.e. embedded in DSL modems), renegotiation cost is raised to its highest level again. If $e(n)$ gets less than 0, it is the sign of resource under-utilization. In this case, exceeding of the total allowable under-utilization cost higher than the current renegotiation cost is the crossing point to resume a new renegotiation cycle. The allocated bandwidth is decreased upon approval of the renegotiation request and $T(n)$ is set to its maximum again. Fig. 4.9 illustrates the behavior of the RCU when $w(e(n))<T(n)$.

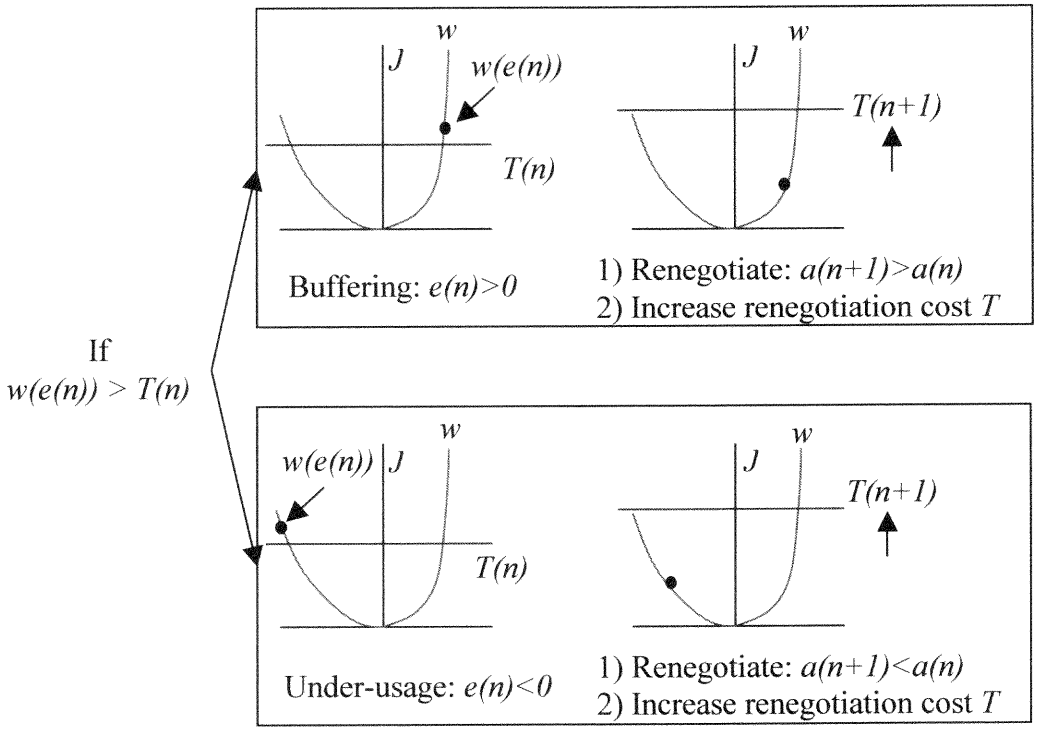


Figure 4.8 Bandwidth allocation decision regions when $w(e(n)) > T(n)$.

Basically, as long as the costs due to under-utilization and buffering do not exceed the current level of $T(n)$, there would be no need to resume a renegotiation request. However, the previous allocated bandwidth amount is maintained, and the renegotiation cost is

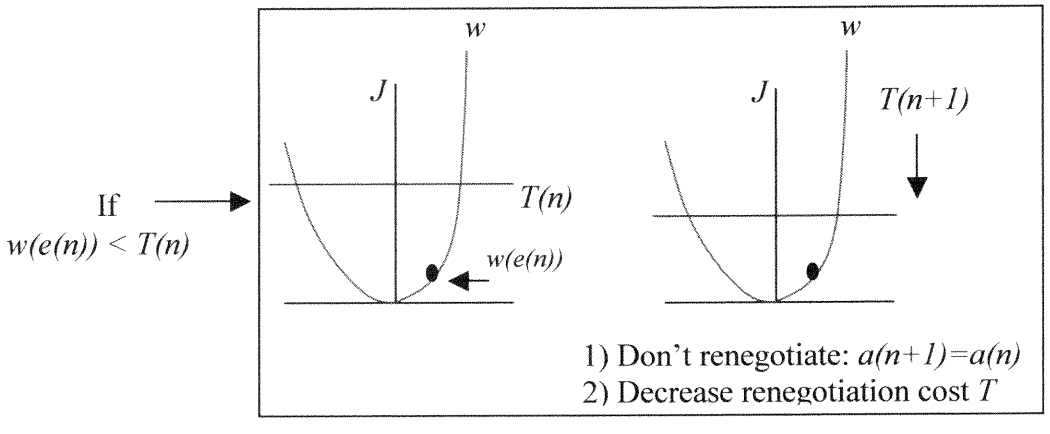


Figure 4.9 Bandwidth allocation decision regions when $w(e(n)) < T(n)$.

decreased by lowering the $T(n)$ level. In a case with hard buffer size constraint (i.e. no buffer overloading permitted), $w(e(n))$ asymptotically diverges to infinity at $e(n)=\beta$ where β is the buffer size constraint. The architecture of the solver of the cost minimization problem with inputs $a(n)$, $r(n)$ and $\hat{r}(n)$ where $\hat{r}(n)$ is the predicted value returned by the introduced bandwidth predictor is given in Appendix B.

4.2 Simulation Results

Sample simulation results for a synthetic VBR trace are presented in Fig. 4.10 in which *(a)* shows the original bit arrivals and the renegotiated bandwidth allocation. The algorithm is run in real-time. The computation time of the bandwidth renegotiation method is negligible, and it is assumed that renegotiation takes place after the request with no delay. The scenario with hard buffer constraint that does not permit overshoot of

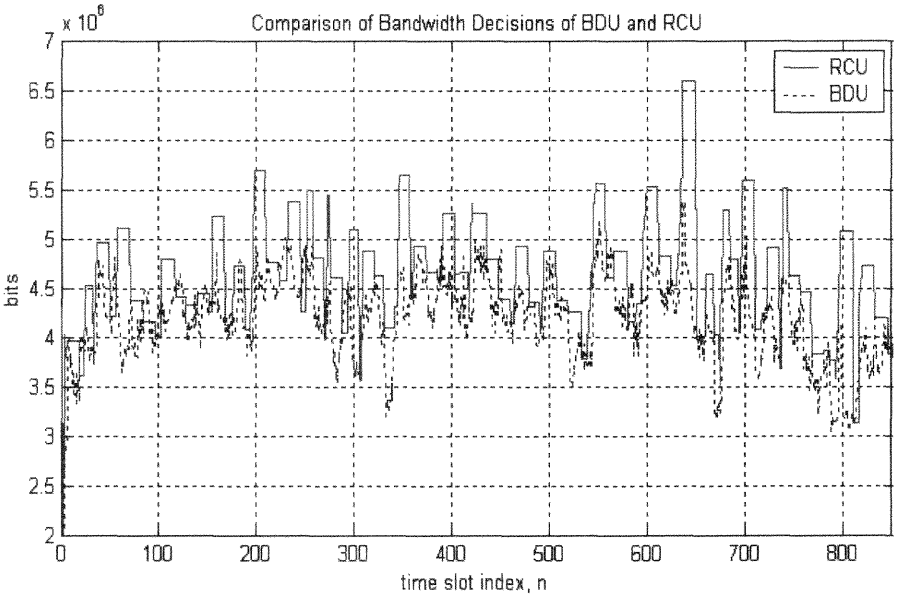
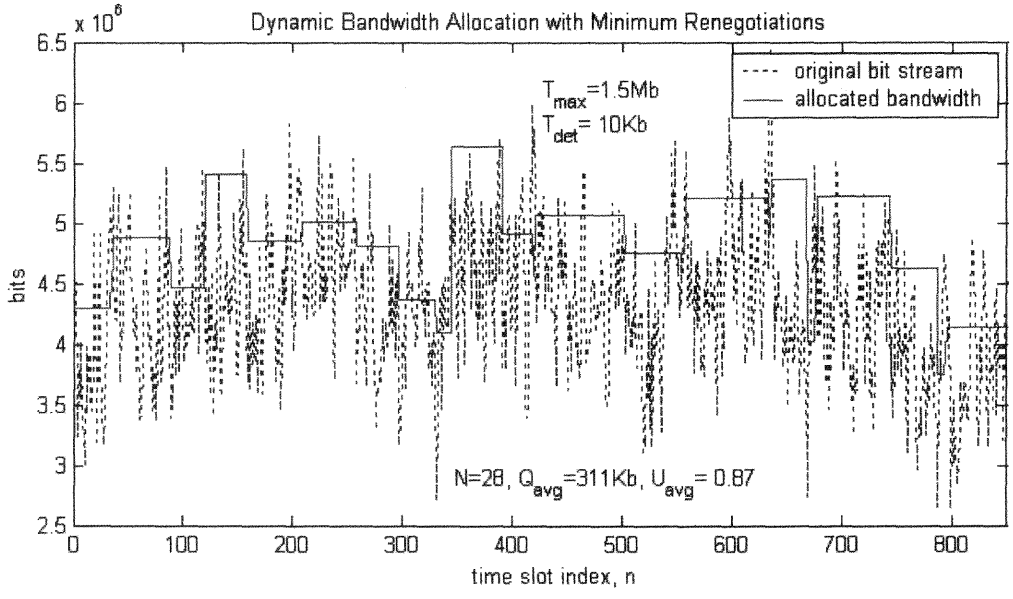
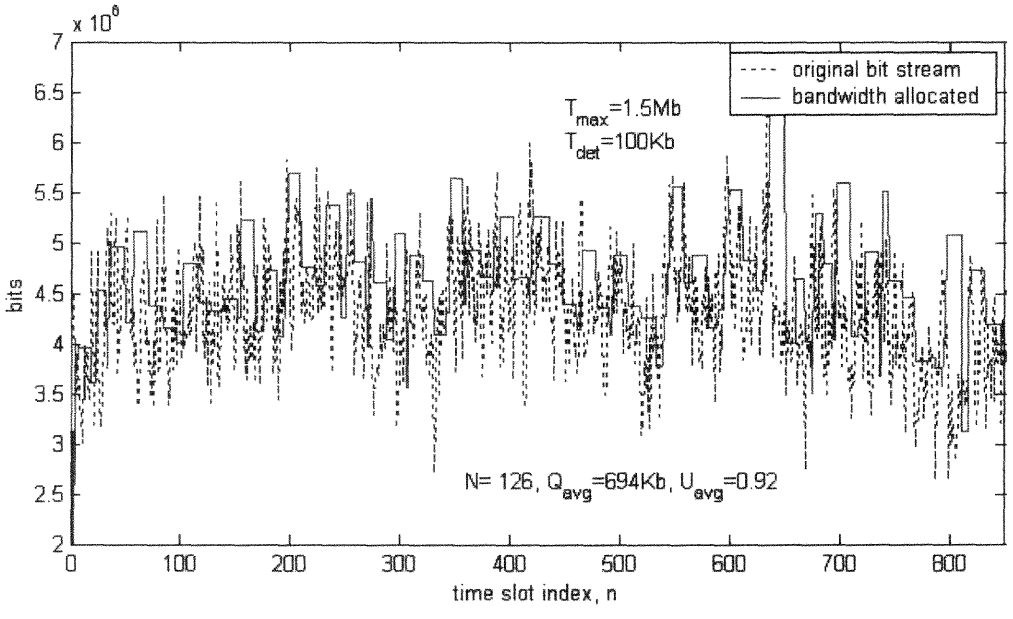


Figure 4.10 Comparison of bandwidth decisions of BDU and RCU. Note: BDU provides feedback to the RCU.

the queue size is simulated. For different severity degrees of the renegotiation cost ($T_{\max}$ and its decrement), and under-utilization and buffering costs, achieved minimum queue size and maximum utilization levels are recorded. It is evident from Fig. 4.11 that



(a)



(b)

Figure 4.11 The effect of T_{max} and T_{det} on renegotiated bandwidth allocation and QoS parameters.

the RCU is more generous in allocating bandwidth than the BDU alone. Intuitively, this diminishes average queue size. On the other hand, the number of renegotiations falls down, being controlled by parameters T_{max} and T_{det}. Fig. 4.11 presents an example output when T_{max} is kept fixed at 1.5Mb and set T_{det} to first 10Kb and then 100Kb. T_{det}=100Kb means that the renegotiation cost decreases 10 times faster than when T_{det}=10Kb ,and therefore more number of renegotiations occurs. The rate of increase in the number of renegotiations for this specific example is 4.5. The change in the number of renegotiations is not a linear function of the ratio T_{max}/T_{det}. It is clear in Fig. 4.11 that increasing T_{det} 10 times improves the average utilization from 0.87 to 0.92 at the expense of increase in the average queue size from 311Kb to 694Kb. This trade off can be fully controlled in favor of queue size performance rather than utilization level by changing the set values for K, L in (4.6), and T_{max}, T_{det}.

Table 4.1 presents the simulations results to reveal the impact of renegotiation cost terms T_{max} and T_{det} on queue size, average utilization and the number of renegotiations. The synthetic trace has 6.4Mbps peak value. It is a common observation from Table 4.1 that increase in T_{det} extends the IRI and lowers the number of renegotiations causing both 0.99 queue size quantile and average utilization to decrease. Lowering T_{max} also increases the probability of having a renegotiation. This high probability reflects on the utilization level and 0.99 queue size quantile as an increase.

K/L is 'queue size to under-utilization cost" weight ratio. If $K<L$, $K/L<1$ meaning that underutilization cost is higher than buffering cost. In this case, the RCU becomes more strict to keep under-utilization level as minimum as possible than when K/L=1. A

comparison of the first four and the last four rows in Table 4.1 gives the variations in 0.99 queue size quantile and average utilization levels for K/L=0.5 and 1, when T_{max}=1.5Mb. It is clear that increase in L escalates the utilization level by allowing 0.99 queue size quantile to raise from 433Kb to 712Kb.

Table 4.1 Impact of Renegotiation Cost on QoS Parameters

T_{max}, T_{det}	Average Utilization	0.99 Queue Size Quantile	Number of Renegotiations	K/L
1.5Mb,100Kb	0.91	433Kb	248	1
1.5Mb,10Kb	0.82	189Kb	94	1
1.5Mb,1Kb	0.79	114Kb	47	1
1.5Mb,0.1Kb	0.77	90Kb	36	1
1 Mb,100Kb	0.93	486Kb	391	1
1Mb,10Kb	0.89	369Kb	244	1
1Mb,1Kb	0.87	314Kb	200	1
1Mb,0.1Kb	0.86	304Kb	196	1
0.5Mb,100Kb	0.96	519Kb	615	1
0.5Mb,10Kb	0.94	492Kb	502	1
0.5Mb,1Kb	0.93	489Kb	484	1
0.5Mb,0.1Kb	0.92	483Kb	483	1
1.5Mb,100Kb	0.92	712Kb	126	0.5
1.5Mb,10Kb	0.87	391Kb	28	0.5
1.5Mb,1Kb	0.81	146Kb	9	0.5
1.5Mb,0.1Kb	0.78	140Kb	5	0.5

4.3. Performance of the RDBA for MPEG-I Star-Wars Movie Trace

To be able to compare the performance of the new RDBA design with other approaches in the literature under realistic scenearios, MPEG-1 coded Star-Wars movie trace is used. MPEG-1 has three frame types: I, P, and B. I frames use intra-frame coding, P frames intra-frame coding and motion compensation based on previous frames, and B frames motion compensation based on previous and future frames. A periodic frame pattern used by the encoder is called a Group of Pictures(GOP).

4.3.1 Comparison of the RDBA with Static Bandwidth Allocations (SBA)

First, the performance of the RDBA design is compared with both static bandwidth allocations (SBA), and dynamic bandwidth allocations (DBA) returned by the BDU. The Star-Wars trace is sent through a single buffer served by a single server at a fixed bit rate. The time resolution for the computations is 40msec which is an inter-frame interval. At different service rates, average queue size, average bandwidth allocation and number of bandwidth renegotiations are computed. The results are illustrated in Fig. 4.12.

It is clearly seen in Fig. 4.12 that both the DBA and the RDBA perform better than any static bandwidth allocation. Average bandwidth that the DBA allocates to the Star-Wars trace is 494Kbps. On the other hand, the RDBA allocates 525Kbps. As far as

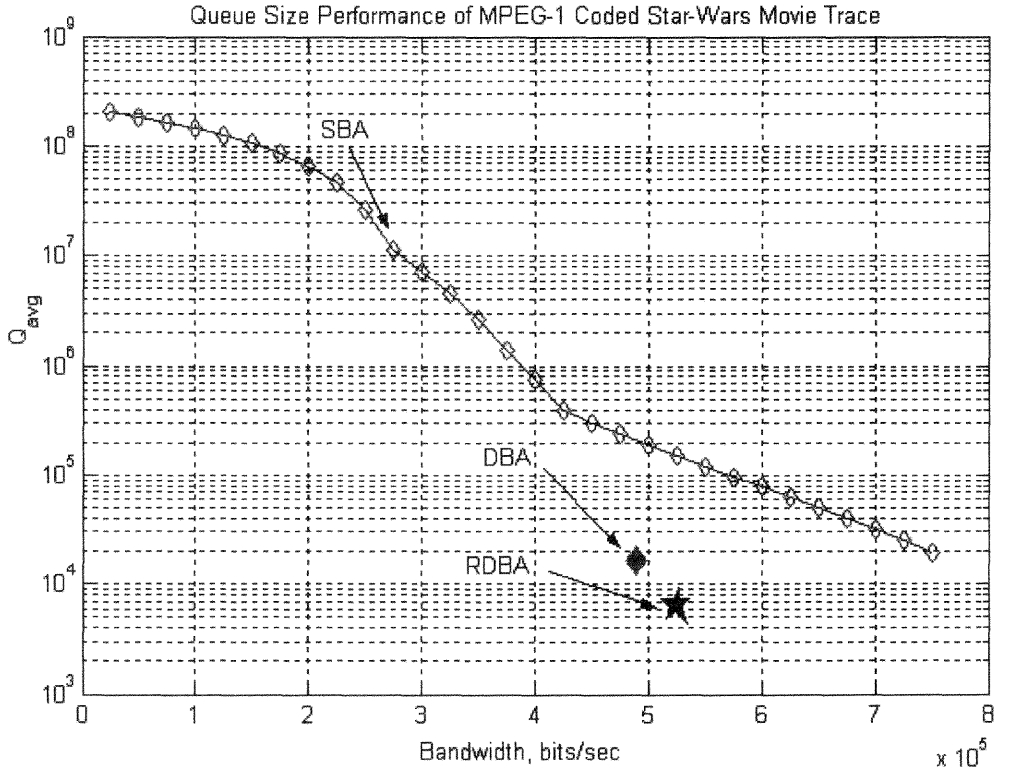


Figure 4.12 Queue size performance of MPEG-1 coded Star-Wars movie trace when serviced by static bandwidth allocation (SBA), only DBA and RDBA(BDU+RCU)

the queue size is concerned, the RDBA outperforms both the SBA and the DBA, and results in 85Kb 0.99 queue size quantile. It is important to note that the RDBA uses less number of renegotiations than the DBA, since the DBA creates as many number of renegotiations as the data samples. It is worth noting that the BDU is designed only to support the RDBA with feedback information for renegotiation decisions.

4.3.2 Comparison of the RDBA with MSE , RED-VBR and PSN-TDNN

A. M. Adas [29] developed a dynamic bandwidth allocation strategy to support VBR video traffic. This strategy predicts the bandwidth requirements for future frames using adaptive linear prediction that minimizes the mean square error. It is designed for VBR traffic and based on the predicted rate of the next GOP. According to the MSE algorithm, to reduce the number of renegotiations it divides the predicted rate by α where $0< \alpha <1$. Hence, more bandwidth is requested than the predicted. This approach is poor because of the possibility that the predicted rate may already be higher than the true arrival rate. Further increasing the requested bandwidth drastically decreases momentary utilization. They also use the approach that the difference between the predicted rate and reserved rate is passed to the averaging filter, and the output of the averaging filter is then compared to the threshold values. This helps to ensure that renegotiation is done only when there is an actual change of rate of the VBR video and not just a spike. Table 4.2 gives the comparison of our results with those in [29].

Table 4.2 Performance Comparison of MSE in [29] with RDBA

Parameter	MSE			T_{max}, T_{det} (230Kb,10Kb)	T_{max}, T_{det} (230Kb,50Kb)	T_{max}, T_{det} (230Kb,70Kb)
N	152	435	762	156	472	593
0.99 Quantile	269Kb	304Kb	320Kb	68Kb	84Kb	86Kb
Avg. Utilization	0.64	0.81	0.88	0.67	0.83	0.85

The value pairs of T_{max} and T_{det} are set such that the same number of renegotiations are obtained with the MSE. Therefore, a common denominator is acquired to compare the resulting queue length quantile and average utilization in each method. Table 4.2 clearly shows that the RDBA returns smaller queue sizes (approximately $1/4^{th}$) at the same number of renegotiations than the MSE-based does. Average utilization levels in the MSE and the RDBA vary within ±3% range. RDBA achieves 3% higher utilization than the MSE (0.64, 0.67) after renegotiating 2% more than the MSE (152, 156). On the other hand, the MSE returns 88% utilization, while the RDBA achieves 85% utilization with 38% less renegotiations than the MSE, (762, 472).

RED-VBR, a renegotiation based approach to support delay sensitive VBR traffic, is developed in [32]. It is. based on deterministic guarantees with client controlled renegotiation of traffic and QoS parameters. RCBR [35] published in parallel with the RED-VBR is closely related to the RED-VBR service. The difference is that the RED-VBR builds the renegotiation service on top of a deterministic variable bit rate (D-VBR) service with the D-BIND traffic model, while the RCBR builds a renegotiation service on top of a constant bit rate service. D-VBR is a more efficient service in that it can achieve a higher network utilization than a CBR service for the same level of QoS, because it models traffic burstiness. Consequently, the RED-VBR requires less number of renegotiations than the RCBR for the same level of utilization. The heuristic online RED-

VBR algorithm maintains the reserved D-BIND parameters and dynamically computes the D-BIND parameters of the previous P frames. If any rate in the measured D-BIND curve exceeds the corresponding rate in the reserved D-BIND curve, a renegotiation immediately takes place. The algorithmic structures of RED-VBR and D-BIND are also shown in Appendix B.

A Pi-Sigma Network (PSN) time delay neural network (TDNN) approach is introduced in [23]. It is a high degree polynomial prediction method. The PSN architecture consists of a single hidden layer of L linear summing units per output, and $M+1$ output product units. The purpose of the approach is to find an approximate L^{th} degree relationship between input data vector and the desired output vector. The architectural diagram of the PSN-TDNN method is given in Appendix B. Comparison of the RBDA with the RED-VBR and the PSN-TDNN when applied to Star-Wars trace has revealed the results in Table 4.3.

Table 4.3 Performance Comparison of the RDBA with RED-VBR and PSN-TDNN for Star-Wars Trace

	RDBA	RED-VBR *(H. Zhang et. al., 1995)*	PSN-TDNN *(S. Chong et. al.,1995)*
0.99 Queue Size Quantile	84Kb	126kb	79Kb
Number of Renegotiations	472	457	494
Average utilization	0.83	0.69	0.64

It is shown in Table 4.3 that the transmission efficiency of the RDBA is better than both the RED-VBR and the PSN-TDNN. The utilization achieved by the PSN-TDNN is 64% after 494 renegotiations. On the other hand, RDBA reaches 83% utilization with 472 renegotiations. The queuing performance of the RDBA scheme is also better than those

of the RED-VBR and the PSN-TDNN. By renegotiating 3.2% less than the RED-VBR, the RDBA results in 33% less queue size than the RED-VBR does.

4.3.3 Improvement in Heavy Tailedness of Queue Sizes with the RDBA

One of the impacts of self-similarity on queuing performance is that heavy-tailedness of queue size distributions increase together with traffic selfsimilarity level. Unlike static or

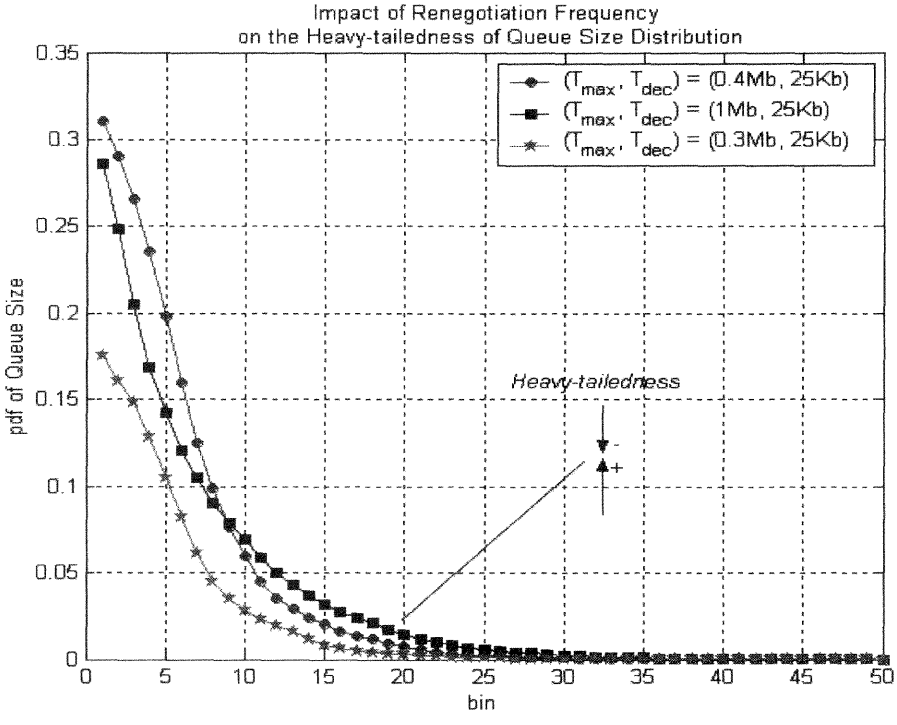


Figure 4.13 Impact of renegotiation frequency on the heavy-tailedness of queue size distribution

synchronous dynamic bandwidth allocations, asynchronous dynamic bandwidth allocation schemes can deal with this impact. At different renegotiation frequencies which are controlled by varying renegotiation cost parameters in the RDBA scheme, queue size distribution of the Star-Wars trace is computed. The results are presented in Fig. 4.13. It is clearly shown in the figure that an increase in the ratio T_{max} / T_{det} reduces

the number of renegotiations. As the number of renegotiations increases, the heavy-tailednes of the resulting queue size decreases. In other words, the queue size performance can be improved by decreasing renegotiation costs.

4.4. Discussion

In this chapter, a method is developed for dynamic renegotiated bandwidth allocation for VBR traffic bit streams. The renegotiation parameters include buffering cost, under-utilization cost, and renegotiation cost. The aim of the design is to provide the required QoS to an applications traffic (i.e. VBR source) under given cost constraints with minimal number of bandwidth renegotiations. The new RDBA unit gets a feedback from the BDU that uses wavelet analysis to predict the next bandwidth demand of a subject traffic stream. An error term is defined as a function of three cost parameters: buffering cost, under-utilization cost, and renegotiation cost. Whenever the error exceeds the cost threshold either due to underutilization or excessive buffering, a renegotiation request is resumed, and next the time and amount of bandwidth to allocate is computed. The renegotiation cost is defined as a joint function of two parameters: T_{max} and T_{det}. T_{max} presents the maximum renegotiation cost which takes its peak at a time slot which is after each completed renegotiation. As time passes, the cost of renegotiation decreases in discrete steps controlled by T_{det}. By proper selection of T_{max} and T_{det}, flexibility is achieved in controlling the frequency of renegotiations, queue size quantile and average utilization. As shown in Table 4.1, an increase in T_{max}/T_{det} decreases the frequency of renegotiations. As a result, both 99 percentile queue size and average utilization decrease. The RDBA approach outperforms SBA in all QoS provisions regardless of the bandwidth

amount used in SBA. After testing the performance of the RDBA against the MSE which is introduced in [29], it is proven that the RDBA outperforms the MSE method. It achieves better 0.99 queue size quantile and average utilization by requesting quite less number of renegotiations than the MSE (Table 4.2). On the other hand, the comparison of the RDBA with the RED-VBR shows that with 3% more number of renegotiations than the RED-VBR, the RDBA can achieve 20% higher utilization and 33% smaller queue size.

To summarize, the new method RDBA outperforms the well-known MSE and the RED-VBR methods. It also has a better queue size and transmission performances than the PSN-TDNN. The only difficulty in the deployment of the RDBA comes from a-priori determination of T_{max} and T_{det}, because selection of these two parameters are crucial to achieve required QoS constraints. The RDBA, due to its suitability for real-time resource management, is deployable in QoS routers, rate adaptive DSL modems, and in ATM switch complexes. The delay in processing renegotiation requests, and its impact on the performance of the RDBA is not studied in this dissertation. This would be one of the topics of further studies in the area.

CHAPTER V

PRACTICAL IMPLEMENTATIONS OF THE RDBA

The provision of QoS guarantees such as bandwidth, delay, jitter, and loss ratio to applications of widely different characteristics is a primary objective in current high-speed networks. In order to properly multiplex traffic in such networks, packet scheduling disciplines are deployed to satisfy the QoS requirements of delay-sensitive applications, and ensure that real-time traffic (e.g. VBR video) and best effort traffic can coexist on the same network infrastructure.

Existing priority based scheduling disciplines basically fall into one of three main categories: delay-based, loss-based, or delay-and-loss-based [56]. In those methods, the performance trade off can be managed from the *space* or *time* perspective. For instance, in Earliest Due Date discipline (EDD), [57], a real time packet is allowed to precede non-real time packets arriving not prior to a duration of certain number of time slots. On the other hand, as an illustration of the space based approach, the precedence can be given to real time packets only when the number of non-real time packets exceeds a preset threshold. The success of such disciplines centers on the effective determination of the thresholds.

Packet Fair Queuing (PFQ) algorithms have been extensively studied for provision of Quality of Service guarantees in Integrated Services Networks. Because of a fixed weight assignment, the inherent PFQ delay-bandwidth coupling imposes limitations on the range of QoS that can be supported. A generalized processor sharing (GPS) server assumes a fluid flow model, and is approximated by a PFQ algorithm which has the notion of "*virtual time*" or "*system potential*". GPS operates at a fixed rate *r* and is work-

conserving. In the worst case, the minimum guaranteed rate g_i given to session i is $g_i = r\phi_i / \sum_{j=1}^{N} \phi_j$ where N is the number of backlogged queues. The excess bandwidth available from sessions not using their reserved service rate is distributed at each instant among all backlogged sessions in proportion to their individual service rates. To overcome the performance limitations caused by the delay-bandwidth coupling of GPS, a time-varying assignment of weights is suggested.

In this chapter, the RDBA system is embedded into WFQ (packetized GPS) traffic scheduler with two priority levels, and the improvement in delay performance of the applications entering each priority level is quantified. Static weight assignment might drive low priority traffic into starvation of bandwidth and cause queuing delays to diverge infinity. It is proven that the RDBA increases the statistical multiplexing gain provided by any static bandwidth allocation, or static weight assignment. The delay in the network derives from several causes. The largest delay component is due to the propagation of the packet. The second one is the delay in transmission at the switch or router nodes. Each packet spends some time in switch or routers' service queues and the amount of this type of delay is variable from packet to packet. This variation (delay jitter) must be minimized to achieve adequate real time service. It is shown that the delay variation in each priority sub-queue is decreased by deployment of the RDBA.

5.1 Static Weighted Fair Queuing Scheduler

WFQ is a generic service discipline, and it is widely applied to QoS routers. In WFQ, the scheduler serves flows or classes at the ratio of their assigned weights. The weights of a class can be determined by its QoS parameters such as delay or service rate. The WFQ scheduler isolates each traffic source from one another, by providing a flow with a specified share of the total available bandwidth determined by their weight ratios under load.

Suppose that a network provides N service classes, and they share a link of capacity C. The share of class i is represented by parameter ϕ_i. The relation between the reserved bandwidth C_i for class i and ϕ_i is such that

$$\frac{\phi_i}{\sum_{n=1}^{N}\phi_i} C \geq C_i \tag{5.1}$$

Let $W_i(t, t+\tau)$ the amount of session i traffic served within the interval $(t, t+\tau)$. For any two queues i, j that are backlogged during $(t, t+\tau)$ (5.2) holds.

$$\frac{W_i(t,t+\tau)}{W_j(t,t+\tau)} = \frac{\phi_i}{\phi_j} \tag{5.2}$$

Each fair queuing (FQ) algorithm maintains a global variable called system virtual time $V(.)$. In addition, a virtual start time $VS(.)$ and virtual finish time $VF(.)$ are associated with each session. $VS(.)$ and $VF(.)$ of k^{th} packet of session i are

$$VS_i^k = max(VF_i^{k-1}, V(t_i^k)) \tag{5.3}$$

$$VF_i^k = VS_i^k + \frac{L_i^k}{C\phi_i} \tag{5.4}$$

where t_i^k is the arrival time and L_i^k the length of the k^{th} packet. The packets/cells are further scheduled based on their timestamps. The delay and the fairness properties of a Fair Queuing policy depend on the virtual time function. In FQ algorithms, complexities come in both sorting the time stamps to select the packets to schedule and the implementation of the selected virtual time functions. Some of the well-known virtual time functions are

- *SFTF (smallest finishing time first):* packets are scheduled in the increasing order of virtual finishing times.
- *SSTF (smallest starting time first):* packets are scheduled in the increasing order of virtual starting times.
- *SEFF (smallest eligible finishing time first):* Eligible packets are with virtual start times less than or equal to the system virtual time.

5.2 Drawbacks Due to Dynamic Weight Adjustment in WFQ

In [58], a proportional delay differentiation service based on dynamic WFQ is presented, assuming infinite buffer. Their algorithm adjusts the weight of each class dynamically in a synchronous manner so that the delay differences between different priority traffic can be controlled. The weight adjustment is done periodically according to the packet arrival rate and buffer occupancy. The measurement period of packet arrival, the measurement period of backlog in priority queues, and the adjustment period of the weights in WFQ scheduler are configuration parameters in [58]. In this dissertation, for the design of a dynamic WFQ, an asynchronous method is used as explained in Chapter 4. This new asynchronous method takes buffer occupancy, bit arrival information, utilization and a

renegotiation cost as input parameters. A synchronous dynamic weight assignment has a poor performance when bursty traffic arrives between two consecutive weight updates.

It is clear from (5.3) and (5.4) that both *VS(.)* and *VF(.)* in a class are inversely proportional to the bandwidth share of that class. Assume for class *i* that at the arrival times of *kth* and *(k+1)st* packets the bandwidth share has values $\phi_{i,k}$ and $\phi_{i,k+1}$ respectively such that $\phi_{i,k+1} = a\phi_{i,k}$ where *a* is a constant. If *a>1, (k+1)st* packet would have smaller *VS(.)* and *VF(.)* than the *kth* packet. In such a case, even though the arrival time of *(k+1)st* packet is after that of *kth* packet, it would be served before the *kth* packet. Therefore, the packet scheduling table in WFQ scheduler would need to be updated whenever a bandwidth share for any class is changed. This introduces a signaling load proportional to the frequency of a bandwidth renegotiation, and a need for packet reordering at a network destination. In the simulations, a WFQ scheduler with *SFTF* is used after modifying the scheduler algorithm to support dynamic asynchronous weight allocation such that upon changing a bandwidth share for any class, *VF(.)s* of all the packets currently waiting to be served in all the queues are recomputed.

5.3 Traffic Generation

The implemented WFQ has two priority levels. The high priority queue is fed with MPEG-1 coded Star-Wars movie trace and the low priority queue with data traffic of heavy web browsing and ftp generated by OPNET™ simulation tools. The traffic types used in the real time simulation is presented in Fig. 5.2 and their statistical properties of are given in Table 5.1.

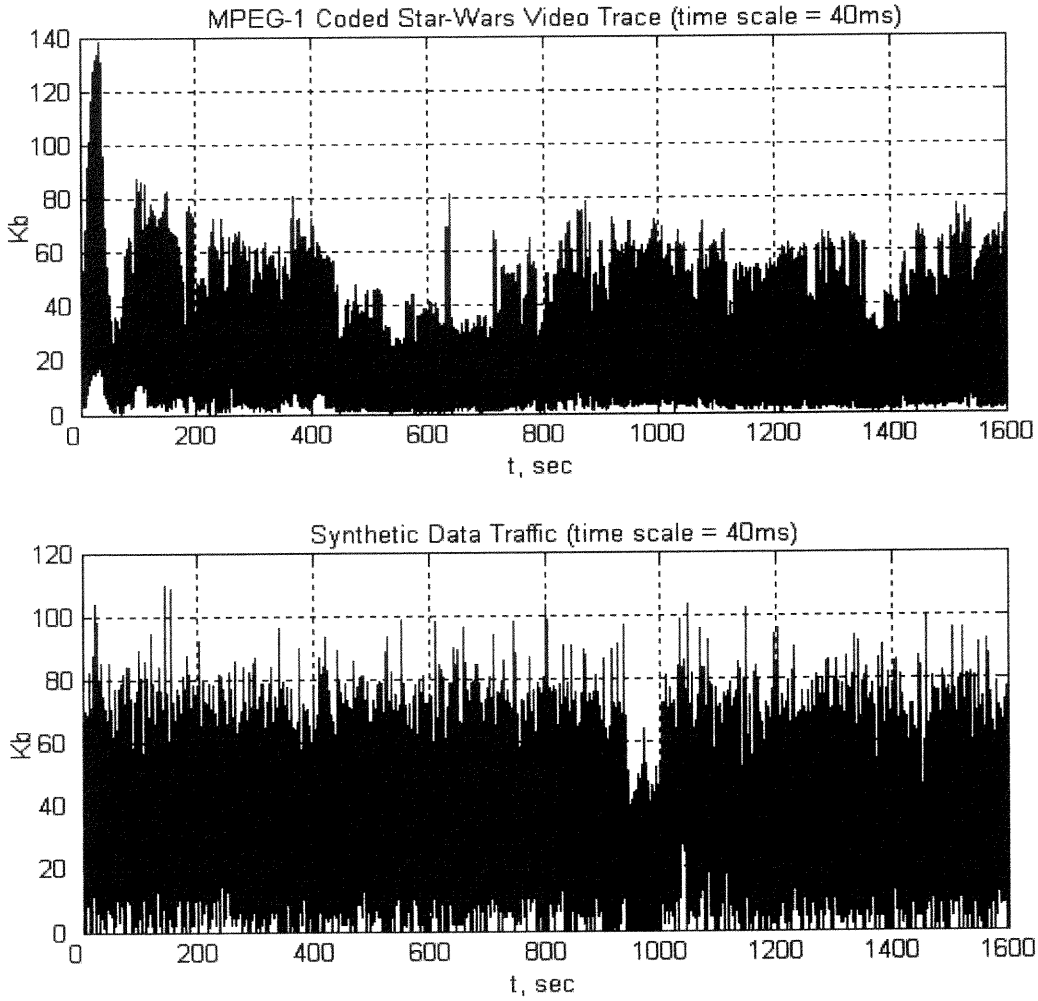


Figure 5.1 MPEG-1 coded Star-wars VBR trace and synthetic data traffic entered into WFQ scheduler at 40ms time resolution.

Table 5.1 Statistical Properties of High and Low Priority Traffic Data Used in the Simulations of WFQ with the RDBA

	Star-wars trace (Kbps)	Data traffic (Kbps)
Standard deviation	355	410
Mean	262	855
Min	10.6	0
Max	3450	2742

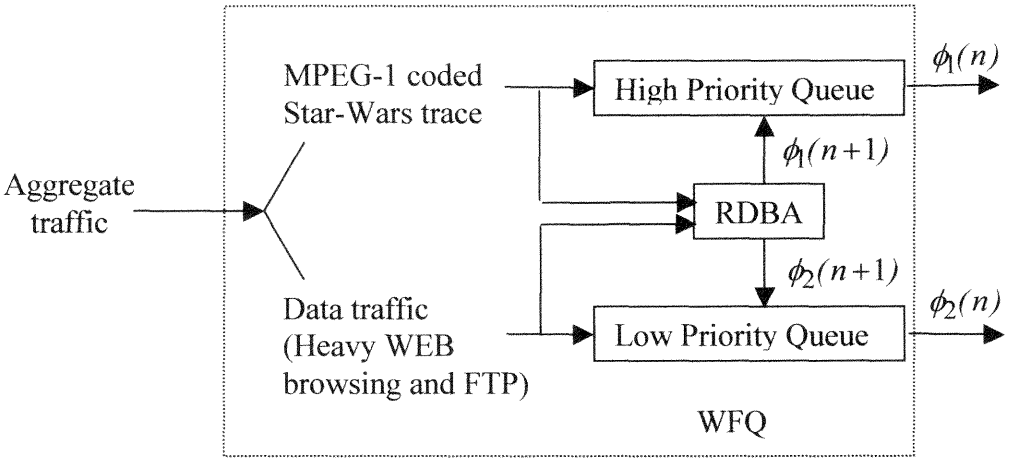


Figure 5.2 Architecture of the implemented WFQ scheduler with RDBA.

In emulating dynamic asynchronous WFQ, the incoming aggregate traffic consisting of both data traffic and Star-wars trace is split into two separate queues as shown in Fig. 5.2 The RDBA unit takes instantaneous measures of each traffic type, and updates the bandwidth share of each queue (ϕ_1, ϕ_2) based on the cost constraints defined in the previous chapter. Available server bandwidth at the output of the WFQ scheduler is selected to be 1.5Mbps which is equal to the capacity of a T1 link.

5.4 Results

5.4.1 Static Weight Assignment

Static bandwidth allocation, if it is close to the peak traffic rate, produces small queue sizes at the expense of low utilization. On the other hand, insufficient bandwidth share given to an applications traffic generates queue sizes diverging to infinity. This can be called as bandwidth starvation. In static WFQ, improper selection of queue weights might cause especially low priority queues to starve from bandwidth.

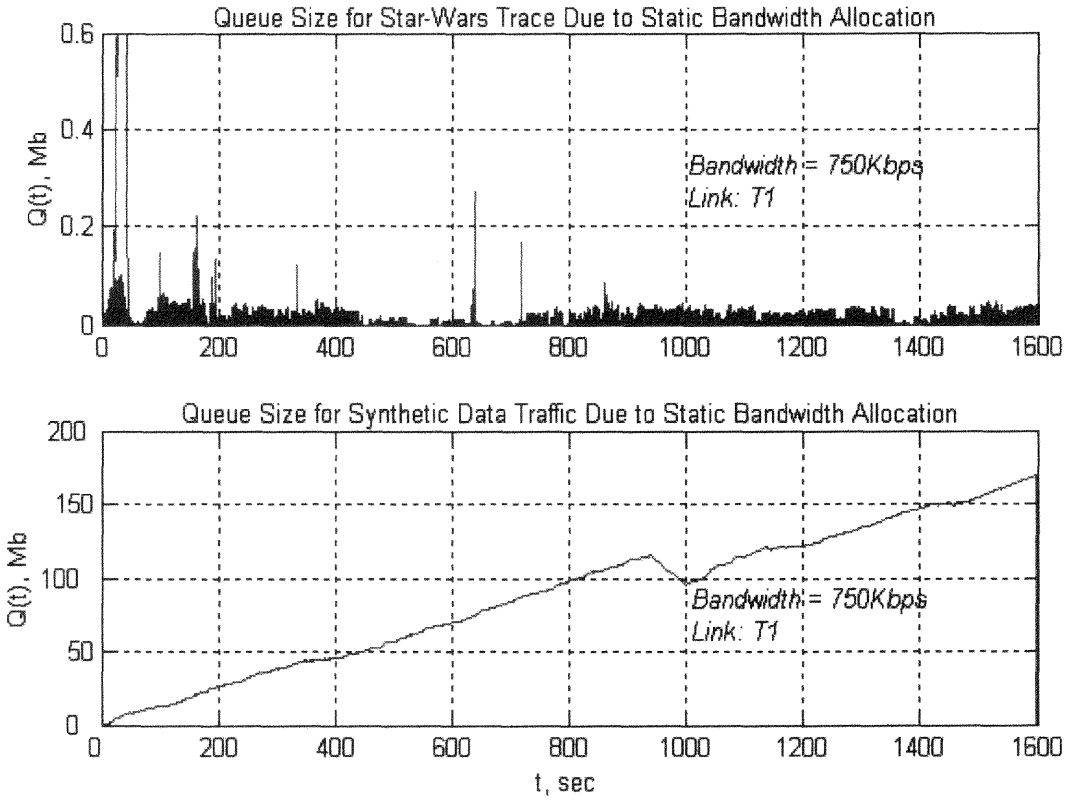


Figure 5.3 An illustration of bandwidth starvation of a low priority queue in case of static weight assignment in WFQ scheduler linked with T1 capacity.

In Fig. 5.3 an illustration of a bandwidth starvation of a low priority queue in the case of static weight assignment in a WFQ scheduler with T1 link capacity is given. The available 1.5Mbps bandwidth is equally shared between high priority Star-wars trace and low priority data traffic. It is shown in Fig. 5.3 that infinite queue size builds in the low priority queue. Static WFQ has the disadvantage that depending on the characteristics of a traffic entering each queue, correct weight assignment has to be made to prevent streams from bandwidth starvation. This is practically difficult. Because, VBR traffic shows time varying fluctuations and any static WFQ scheduler is away from providing the required QoS constraints. RDBA approach without any user control updates the bandwidth share coefficients for each stream.

5.4.2 Weight Control by RDBA in WFQ

Traffic schedulers are classified into two groups: work conserving and non work-conserving. Once bandwidth shares for each priority queue is determined, in work conserving type idle bandwidth of a queue is used by other backlogged queues. On the other hand, in non work-conserving mode, even if one of the queues are empty and it is not using its allocated bandwidth, other queues are still served at their initial bandwidth

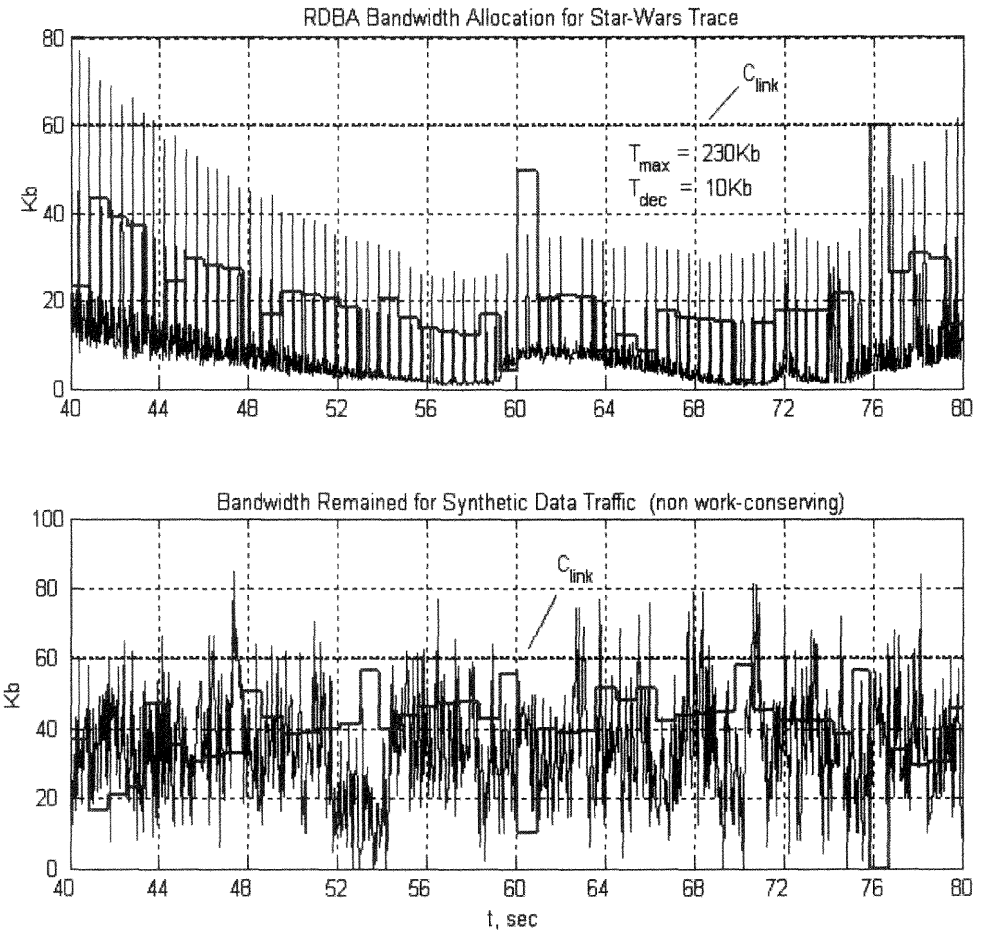


Figure 5.4 Illustration of bandwidth allocation for high priority Star-Wars trace and low priority data traffic in a non-work conserving scheduler.

shares and do not make use of the idle bandwidth assigned to the empty queue. Dynamic weight control by RDBA produces superior queue performance compared with static WFQ schedulers in both non work-conserving and work conserving mode.

In Fig. 5.4, how a non-work conserving traffic scheduler with RDBA distributes available bandwidth between high priority and low priority traffic is illustrated with time resolution of 40ms. In a non-work conserving scheduler, it is clear that once a certain

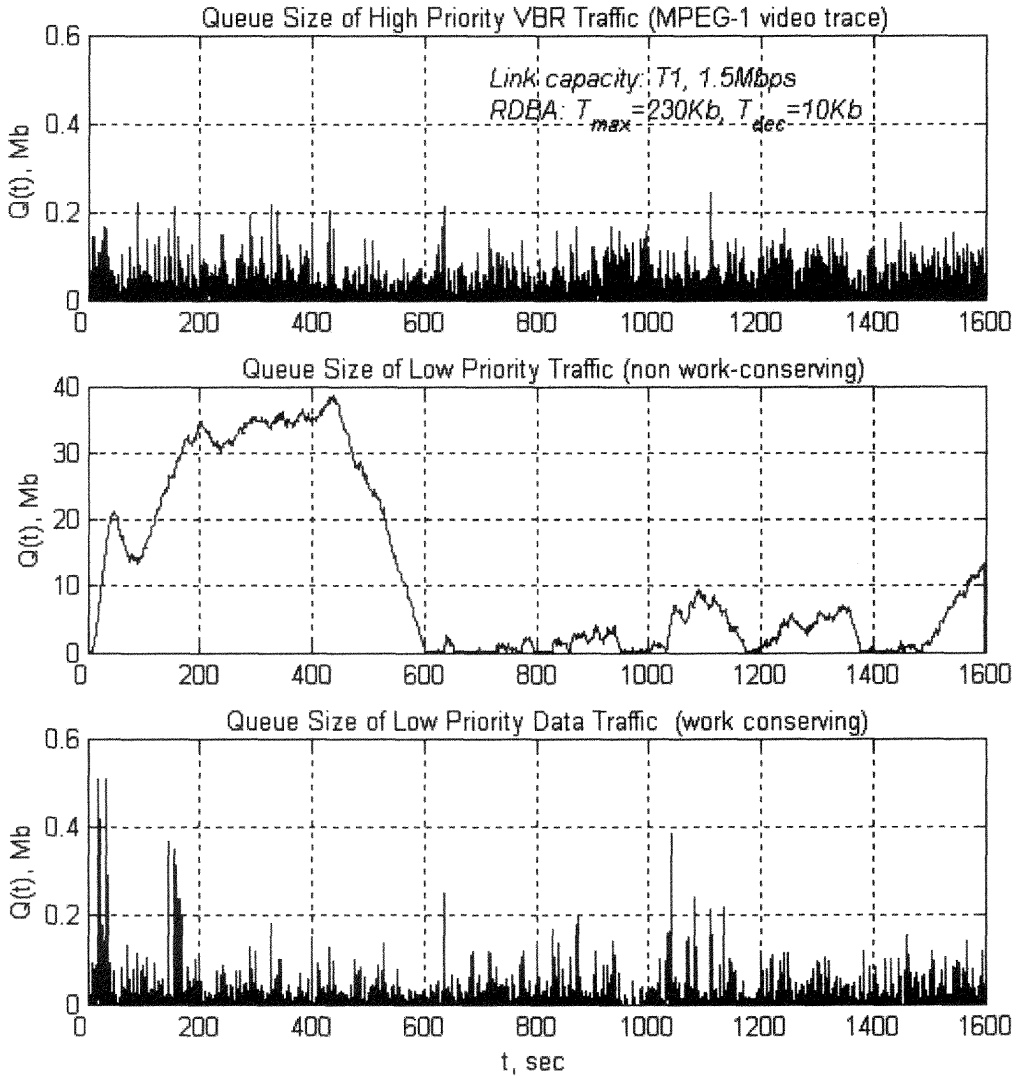


Figure 5.5 Queuing performance of a dynamic WFQ scheduler with embedded RDBA scheme.

share of bandwidth is assigned to the high priority traffic, the low priority traffic is only allowed to use the remaining bandwidth of the link capacity, even though at times the high priority queue does not use its entire bandwidth share allocated. In Fig 5.5, the queuing performance of a dynamic WFQ with embedded RDBA mechanism is given. The queue size of low priority traffic which diverges to infinity after static weight assignment in non work-conserving mode is controlled with the use of RDBA. The queue size does not exceed 37Mb , and within the time interval (570s, 1600s) is upper bounded by 10Mb. In work conserving mode, maximum low priority queue size can be constraint down to 0.5Mb. This proves that the use of RDBA maintains the delay QoS required by VBR traffic served in a high priority queue and also increases the statistical multiplexing gain for data traffic entering the low priority queue, that is larger number of data streams can be supported in dynamic WFQ than that in static WFQ.

5.5 Summary

The practical applications of the new renegotiation based dynamic bandwidth allocation scheme is simulated. The simulations consist of two prioritized traffic streams. Star-Wars movie trace is entered into a traffic scheduler as a high priority traffic trace, and data traffic synthetically generated in OPNET environment as a low priority traffic. Bandwidth share of the high priority traffic is asynchronously computed within the RCU, and the remaining bandwidth of the link capacity is assigned to the low priority traffic. Work conserving usage of the shared resources is an inherent feature of WFQ schedulers. On the other hand, non-work conserving usage is seen in ATM networks. ATM networks provide connection-oriented services with guaranteed bandwidth. In order to carry an IP

datagram in such networks, a virtual circuit (VC) has to be setup with an associated bandwidth requirement. Once a VC is setup, the adaptation layer has to decide how long to keep the VC active with the initial bandwidth assignment. Any idle portion of the bandwidth allocated for a VC is not shared among other VCs, unless it is distributed as a result of a renegotiation request. Asynchronous bandwidth renegotiation in ATM networks, and also asynchronous weight update in WFQ based traffic schedulers are not studied well in the literature. The RDBA offers a solution to these open issues. The simulations of the RDBA in both work-conserving and non-work conserving modes in two priority level traffic schedulers prove that the queue-size performance of a low priority traffic can be improved without degrading the QoS constraints of a high priority traffic. Static bandwidth share among high and low priority traffic streams may drive low priority traffic into bandwidth starvation, resulting in an infinite queue-size. A dynamic allocation with RDBA, on the other hand, prevents low priority queue from bandwidth starvation. Intuitively, in a work-conserving mode, the queue-size performance improvement is higher than that in a non-work conserving mode.

CHAPTER 6

CONCLUSION

This dissertation defends the thesis that cost effective renegotiation based dynamic bandwidth allocations for VBR traffic satisfies the constraints for link utilization and packet delay, while eliminating the heavy-tailedness of the queue size distributions due to self-similarity. It is proven that energy distribution of a traffic segment among multiple frequency sub-bands is a readily available means to asymptotically predict traffic patterns especially for real time resource management.

First, an analytical framework for measurement of traffic self-similarity under the constraints of processing time and computational load for real time QoS provisioning is presented. Comparison of the wavelet and variance-time plots that are used to measure Hurst parameter of a traffic trace reveals that the wavelet method is more suitable for online QoS provisioning because of its simplicity and 16±0.67 times less processing load than the variance-time method, especially if more than three scales in the wavelet filter bank is used, and the data vector size is greater than 1024 samples. The ratio of the increase in the size of a data vector proportionally affects processing load at a given number of scales. In the wavelet method, Hurst parameter is determined by the relation between signal energies within sub-bands of a wavelet filter bank and the index of each scale. Signal energy distribution among different frequency sub-bands presents low and high frequency components of applications traffic and their volume ratio in each time frame of a measure. This energy information can be used to predict future traffic behavior and to minimize the number of proper bandwidth adjustments.

It is proven that deployment of wavelet filters with large number of taps reduces the aliasing between adjacent frequency sub-bands. On the other hand, small number of taps generates positive offset value in energy amount in each sub-band. LMS fit applied to the energy vs. scale index curves returns Hurst parameters within the range of 0.86±0.02 for wavelet orders of 2, 4, 5 when tested on Star-Wars video trace. Two percent deviation in the estimate of H due to different filter lengths is negligible. Considering the fact that high order filters elevate the processing time and result in additional delays, low order filters are preferred in real time applications to the higher orders. Also, it is illustrated that the Haar and Daubechies wavelets return slightly different estimate of the sub-band energies. However, this difference is insignificant for online resource allocation decisions. The comparison of the Haar wavelet with 4 tap Daubechies results in that the offset in the high frequency sub-band energy between the two wavelet types increases as the source self-similarity level related to γ, the power component of the heavy-tailed frequency spectrum of the source, decreases. The offset interval is (0.012, 0.056) for values of $0.2<\gamma<1$. Therefore, the Haar wavelet is preferred to the Daubechies from simplicity point of view; in addition to that it provides the positive signal constraint in wavelet analysis.

Based on the analytical derivations, suggestions are made regarding the selection of the number of scales in a wavelet filter bank, the window size to capture data for each analysis, and overlapping size of consecutive windows to provide cost-effective bandwidth renegotiation. Finally, it is shown that Hurst parameter itself is not a good representative of traffic burstiness, and must be accompanied by other measures such as mean and peak traffic bit rate, and standard deviation of the bit arrivals per unit time.

A novel bandwidth predictor that uses wavelet analysis in computation of sub-band signal energies as traffic descriptors is presented. Input traffic information is a periodic bit counting process and its increments. Specifically in VBR traffic, bandwidth requirements vary in time due to scene changes and coding structures. Therefore, dominance of either low or high frequency fluctuations in a traffic pattern can be inferred by looking at the sub-band energies. Based on the predefined properties and second order statistical parameters of a temporal energy vector, bandwidth prediction is returned. The comparison of wavelet based bandwidth predictor with Linear Mean Square (LMS) and Recursive Least Square (RLS) predictors proves that the wavelet approach outperforms the latter two. It is possible for LMS and RSL methods to return negative predictions (zero crossing effect), and to mislead a renegotiation unit. Because of its positive signal characteristics, the Haar wavelet analysis always returns a positive prediction. It is also shown that the wavelet based approach returns smaller queue sizes than *AR(1)* and *AR(N)* based prediction models at the expense of lower utilization. Smaller queue sizes are achieved by over-predicting bandwidth demand. However, this over-prediction increases the performance measure Mean Square Bandwidth Allocation Error (MSBAE) for the wavelet method about 25-45%. This low utilization cost is compensated successfully by design of bandwidth renegotiation unit presented in Chapter 4.

A new bandwidth renegotiation method is presented. Renegotiation capability, when deployed at ATM switches, QoS routers or traffic schedulers such as Weighted Round Robin (WRR) and Weighted Fair Queuing (WFQ), increases the statistical multiplexing gain by making it possible to dynamically distribute available resources among different applications. The new method assumes different priority applications

sharing a fixed bandwidth capacity. The method is designed to be online and applicable to DSL networks in which the rate of Rate Adaptive Digital Subscriber Line Modems (RDSLs) is controlled by the central office, to WFQ to dynamically update bandwidth share of each priority traffic, to ATM switches in which bandwidth allocation to each active VC connection is updated without terminating the VCs and setting-up new ones. The method considers three cost functions: *i)* renegotiation cost *ii)* buffering (under-allocation) cost *iii)* bandwidth (over-utilization) cost. An error function is defined which combines the three defined cost functions and decision boundaries for renegotiation requests are determined. The method takes as inputs the predicted bandwidth from the BDU, momentary queue size and actual traffic bit arrival information, and determines the time instants of renegotiation requests and bandwidth amount to allocate providing minimum number of renegotiations for given QoS constraints. Performance comparison of Renegotiated Dynamic Bandwidth Allocation (RDBA) with Static Bandwidth Allocation (SBA), Mean Square Error (MSE) based methods and RED-VBR and PSN-TDNN reveals that the new method accomplishes higher utilization and less queue-sizes than these four. Also, it requests less number of renegotiations than MSE and RED-VBR and PSN-TDNN. However, there is an open issue that is not looked into and may be a subject of further studies: impact of delays (between the start time of a renegotiation request and the time the request is granted) on the performance of the RDBA. This delay is assumed to be zero in this dissertation. Also, in the literature no satisfactory work is done to address this open issue.

Finally, the RDBA is applied to a WFQ scheduler with two priority levels to dynamically change bandwidth shares of each level. High priority queue is provided with

the Star-Wars trace and low priority with synthetic data traffic. It is observed that static weight assignment may result in bandwidth starvation of low priority traffic. Dynamic weight assignment, on the other hand, eliminates this problem without degrading the QoS requirements of high priority traffic. One drawback in dynamic WFQ is that a packet entering a queue upon readjusting the priority weights might be given an earlier virtual finish time than a packet arrived before, because virtual finish and end times are functions of priority weights. In such a case, virtual finish times of all the packets in queues have to be updated according to new changes. This creates a heavy signaling load on traffic schedulers. Alternative solutions to this problem may be also investigated in future studies.

APPENDIX A

BANDWIDTH PREDICTION METHODS

There are several different strategies in predicting the future bandwidth demand of traffic for dynamic bandwidth allocation. Each update (new allocation) consists of a prediction and a correction term based on previous updates. General structure of the dynamic bandwidth allocation algorithms can be written as

$$new\_estimate = old\_estimate + correction$$

The weighting of the correction and the old estimate, and how they are updated depend on the algorithms used. The following is a list of approaches in dynamic bandwidth allocation:

a. *Peak-Rate*

Overall peak rate is determined and assigned as the new bandwidth to traffic each time the updating occurs. Peak-rate based bandwidth reallocation results in minimum number of updates compared to other available methods. However, it suffers from low utilization and inefficient usage of available capacity [23].

b. Previous

Allocated service rate at time slot n+1 depends on the arrival rate at the same time instant [26]. However, the arrival rate is not known a priori. In this method, $\hat{X}(n+1)$, the prediction of the traffic arrival rate in time slot $n+1$, is set to the arrival rate in the previous interval, $X(n)$.

c. Average

The new bandwidth allocation to the traffic is determined depending on the average arrival rate in the past. This is a special case of AR-M model where M is the number of samples averaged [26]. This feedback algorithm is unable to follow rapid changes in the traffic condition and might cause building queuing delays.

d. Combination of Previous and Average

This method combines the previous two approaches. Each approach is assigned a weight and the superposition is taken. For instance, having $X(n)$ as the arrival rate at discrete time instant n and assuming that $\overline{X}$ is the average of last m values of X such that $\overline{X}(n)=\frac{X(n)+X(n-1)+....+X(n-m)}{m}$ where m is a positive integer, the predicted bandwidth in time slot $(n+1)$ is given as

$$\hat{X}(n+1)=(1-\rho)\overline{X}+\rho X(n) \qquad \text{(A.1)}$$

where $0<\rho<1$. The performance of this method is highly dependent on the selection of parameters m and ρ. This method has a poor queue size and utilization performance in case of VBR, bursty traffic.

e. Combination of Previous and Queue Size

New allocation in time slot $(n+1)$ is determined such that

$$\hat{X}(n+1)=max(\hat{X}(n)-X(n),0)+X(n) \qquad \text{(A.2)}$$

The term $(\hat{X}(n) - X(n))$ is the actual queue size at the end of interval n. Assuming the same arrival rate in the new interval, $(n+1)$, as in the interval n, the prediction also contains $X(n)$ and flushes the queue contents left from interval n.

f. Minimum Mean Square Error Linear and LMS Predictors

A pth-order k-step linear predictor has the form

$$\hat{x}(n+k) = \sum_{j=o}^{p-1} w(j)x(n-j) \tag{A.3}$$

where $w(j)$ are the prediction filter coefficients.

The error term is

$$e(n) = x(n+k) - \hat{x}(n+k) \tag{A.4}$$

The optimality of the predictor in the mean square sense requires minimization of the mean square error ξ, where

$$\xi = E\{e^2(n)\} = E\{(x(n+k) - \underline{w}^T \underline{x})^2\} \tag{A.5}$$

The vector $\underline{w}$ that minimizes ξ can be found by taking the gradient. The result is set to zero, and is solved for $\underline{w}$.

$$\nabla\xi = -2E\{e(n)x(n)\} = 0 = -2E\{(x(n+k) - \underline{w}^T \underline{x})\underline{x}\} \tag{A.6}$$

Taking expectation and writing it in matrix form results in

$$R_x \underline{w} = r(k) \tag{A.7}$$

where $R_x = E\{\underline{x}^T \underline{x}\}$ and $r(k) = E\{\underline{x}x(n+k)\}$. The solutions of the linear equations in (A.7) require the knowledge of the autocorrelation of $\underline{x}$, and it also assumes wide sense stationarity, i.e. the mean, variance, and auto-covariance of $x(n)$ do not change with time.

Therefore, MMSE linear predictor is not a good choice if real time resource management, bandwidth allocation, is subject to provision. It cannot be used as an online algorithm for forecasting bandwidth. In [25], they studied adaptive least mean square error linear predictor. The LMS is an adaptive approach and it does not require prior knowledge of the autocorrelation structure of $\underline{x}$. Therefore, it can be used as an online algorithm. The algorithm starts with an initial estimation of the filter coefficients w*(0)*. For each data point, the coefficients are updated according to the following recursive equation:

$$w(n+1) = w(n) + \mu.e(n)x(n) \qquad (A.8)$$

where μ is called a step size and a constant. In [23] they analyze low and high frequency characteristics of traffic trace and allocate available resources accordingly. Their work is the closest in its nature to our approach. They describe VBR traffic in frequency domain being inspired from the fact, especially for VBR video traffic, that the low frequency signal captures the slow-time variation of consecutive scene changes and that the high frequency signal exhibits the feature of strong frame autocorrelations.

g. Adaptive Wavelet Prediction

In [22] the concept of using wavelets in dynamic bandwidth allocation for video traffic is brought up. In Fig.A.1, the block diagram of their approach is given. The use of wavelet transform in adaptive filtering has also been proposed in [45, 46]. It is very analogous to the DFT based adaptive filtering. On the other hand, it has the advantages of wavelet analysis over Fourier analysis when a weighted sum of sinusoids does not adequately represent the time-varying modes of the signals involved. Even though the reduction in eigen-value spread by use of wavelet transform Their method also involves the

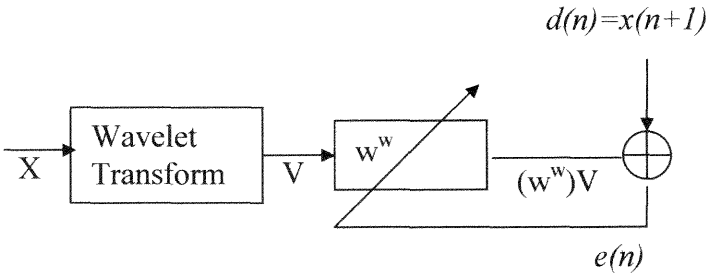


Figure A.1 The block diagram of the wavelet predictor proposed in [22]

computation of the autocorrelation function of the input trace as in MMSE. The reduction is due to that the wavelet transforms and a class of discrete orthogonal transforms create approximately diagonal autocorrelation matrix of the input data.

In the literature one of the most common metric of the performance measure of the dynamic bandwidth predictors is Q.

$$Q = \sum e^2(n) / \sum x^2(n) \tag{A.9}$$

Another metric must be the number of renegotiations (RN). The requirement in the dynamic bandwidth allocation is to get minimum Q at minimum RN.

APPENDIX B

BANDWIDTH RENEGOTIATION METHODS

The RCU is developed as an alternative method to the RED-VBR and PSN-TDNN schemes. Algorithmic structures of these approaches are given in the following sections.

a) D-BIND

The key components of the D-BIND model are that it is bounding and interval dependent [34]. Bounding is required to provide deterministic QoS guarantees. On the other hand, interval-dependency allows for a higher utilization by capturing important burstiness properties of traffic. D-BIND model is defines via rate-interval pairs $(R_k, I_k) \mid k = 1,2,..K$. Each rate R_k is an upper bound on the rate every interval of length I_k. Assume a traffic constraint function *b(t)* which bounds the traffic rate for an interval of length I_k. The constraint function *b(t)* is piece-wise linear within $(R_k, I_k) \mid k = 1,2,..K$ as in (B.1) where K is the number of intervals.

$$b(t) = \frac{R_k I_k - R_{k-1} I_{k-1}}{I_k - I_{k-1}}(t - I_k) + R_k I_k, \; I_{k-1} \le t \le I_k \qquad \text{(B.1)}$$

b) RED-VBR

RED-VBR is based on deterministic guarantees with client controlled renegotiation of traffic, QoS parameters and graceful adaptation during overload periods. RED-VBR algorithm decides when to allocate more/less resources for VBR traffic and at what amount. The policy decisions are based on the two sets of D-BIND parameters. To control the policies, two control parameters α and β are defined such that $\beta > \alpha$

and $\alpha, \beta \geq 1$. If any rate in the measured D-BIND curve exceeds the corresponding rate in the reserved D-BIND curve, a renegotiation takes place. The new bandwidth amount is chosen such that each R_k is α times its current value. An online RED-VBR based renegotiation algorithm for a VBR video sequence given below is quoted from [32].

```
**************************
initialize current_allocation_R[];
LastRenegIndex=0;
for (i=0;i<MAX_FRAMES;i++){
compute R[1:P] based on previous M frames;
if ( R_k >current_allocation_R[k] for any 0≤k≤P{
for (k=1;k≤P;k++)
        current_allocation_R[k]=α *max(current_allocation_R[k], R_k );
LastRenegIndex=i;
Renegotiate();
}elseif ( R_P < β * current _allocation _R[ P ] AND
        (i-LastRenegIndex≥MIN_RENEG_INTERVAL)){
for (k=1;k≤P;k++)
        current_allocation_R[k]=current_allocation_R[k]+ R_k /2;
        LastRenegIndex=i;
        Renegotiate();
}
}
**************************
```

c) PSN-TDNN

Fig. B.1 shows a TDNN based on L^{th} degree PSN with (N+1) inputs and (M+1) outputs. The PSN architecture consists of a single hidden layer of L(M+1) linear summing units and output layer of (M+1) product units [23]. The output from each unit passes through a sigmoid activation function defined by $\sigma(x) = \frac{1}{1+e^{-x}}$. The weights from the hidden layer to the output layers are fixed at 1.

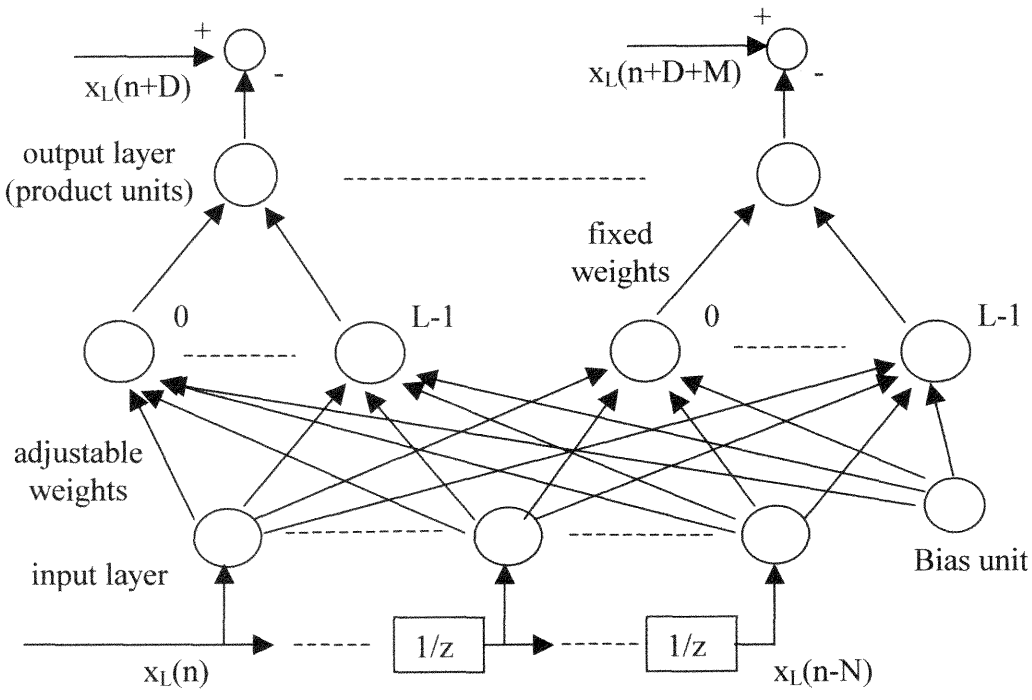


Figure B.1 An L^{th} degree PSN-TDNN architecture.

The purpose of this network is to find an approximate L^{th} degree relationship between the inputs $x_L(n), x_L(n-1), ..., x_L(n-N)$ and the desired outputs $x_L(n+D), x_L(n+D+1), ..., x_L(n+D+M)$. Input vector $\vec{e}(n)$, desired output vector and estimated output vector $\vec{y}(n)$ are defined as follows:

$$\vec{e}(n) = [1, x_L(n), x_L(n-1), ..., x_L(n-N)] \quad \text{(B.2)}$$

$$\vec{d}(n) = [x_L(n+D), x_L(n+D+1), ..., x_L(n+D+M)] \quad \text{(B.3)}$$

$$\vec{y}(n) = [\hat{x}_L(n+D), \hat{x}_L(n+D+1), ..., \hat{x}_L(n+D+M)] \quad \text{(B.4)}$$

Assume $\vec{w}_j^l$ be the weight vector for the j^{th} hidden unit of the l^{th} output where $l=0,1,...,M$ and $j=0,1,...,L-1$. The l^{th} output $y_l(n)$ at time n is then expressed as

$$y_l(n) = \sigma[\prod_{j=0}^{L-1} \vec{w}_j^l \vec{e}^T(n)] = \sigma[\prod_{j=0}^{L-1} (\sum_{k=1}^{N+1} w_{k_j}^l e_k(n) + w_{0_j}^l)] \quad \text{(B.5)}$$

The MSE objective is given by

$$J_l = \frac{1}{p} \sum_{n=0}^{p-1} [d_l(n) - y_l(n)]^2, \; l=0,1,...,M \quad \text{(B.6)}$$

where p denotes the number of training patterns.

d) RCU

The flow architecture of the decision mechanism developed for the renegotiation control unit given in Chapter 4 is shown in Fig. B.2. It takes three input parameters from other units: current allocated bandwidth a(n), last arrival rate r(n) and the prediction of the bandwidth requirement for the next time slot. Depending on instantaneous queue size and utilization values, the algorithm first computes the buffering and under-utilization costs with the assumption that the actual arrival rate within the next time slot is equal to the predicted value, and the bandwidth is kept the same in time slot $(n+1)$ as in n, that is $a(n)=a(n+1)$. New decision is made after comparison of the renegotiation cost $T(n)$ with buffering and under-utilization cost e(n).

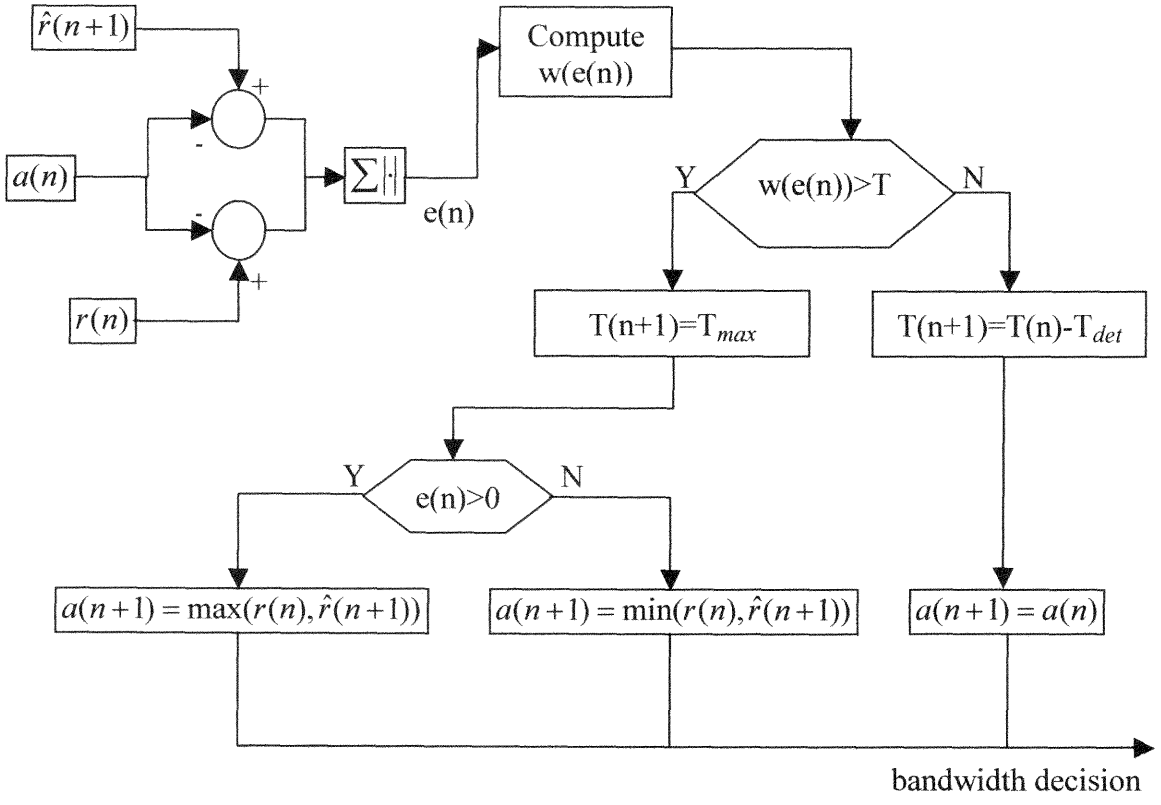


Figure B.2 Algorithmic structure of the Renegotiation Control Unit (RCU).

REFERENCES

1. R. Jain, S. Routhier, "*Packet Train Measurements and New Model for Computer Network Traffic,*" IEEE JSAC, v:4, n:6, pp. 986-995, September 1986.

2. H. Fowler, W. Leland, *"Local Area Network Traffic Characteristics with Implications for Broadband Network Congestion Management,"* IEEE JSAC, v:9, n:7, pp. 1139-1149, September, 1991.

3. R. Gusella, *"A Measurement Study of Diskless Workstation traffic on an Ethernet,"* IEEE Trans. on Comm., v:38, n:9, pp. 1557-1568, September, 1990.

4. W. E. Leland, M. S. Taqqu, W. Willinger, D. Wilson,*"On the Self-similar Nature of Ethernet Traffic,"* Proc. of the ACM SIGCOMM'93, San Francisco, CA, pp. 203-213, 1993.

5. W. Willinger, M. S. Taqqu, D. Wilson, *"Self-similarity in High Speed Packet Traffic: Analysis and Modeling of Ethernet Traffic Measurements,"* Statistical Science, v.10,pp. 67-85, 1995.

6. W. Paxon, S. Floyd, *"Wide Area Traffic: The Failure of Poisson Modeling,"* In the proc. of the ACM SIGCOMM'94, London, UK, pp. 257-268, 1994.

7. J. Beran, R. Sherman, M.S. Taqqu, W. Willinger, *"Long Range Dependence in VBR Video Traffic,"* IEEE Trans. on Comm., v.43, pp.1566-1579, 1995.

8. M.W. Garrett, W. Willinger, *"Analysis, Modeling and Generation of Self-similar VBR Video Traffic,"* Proc. of the ACM SIGCOMM'94, London, UK, pp. 269-280, 1994.

9. C. Huang, M. Devetsikiotis, I. Lambadaris, A.R. Kaye, *"Modeling and Simulation of Self-similar VBR Compressed Video: A Unified Approach,"* ACM SIGCOMM'95, Computer Communications Review, v.25, pp.114-125, 1995.

10. K. Park, G. Kim, M. Crovella, *"On the Relation Between File Sizes, Transport Protocols, and Self-Similar Network Traffic,"* In Proc. IEEE International Conference on Network Protocols, pp. 171-180, October 1996.

11. K. Park, G. Kim, M. Crovella, *"On the Effect of Traffic Self-Similarity on Network Performance,"* In Proc. SPIE International Conference on Performance and Control of Network Systems, pp. 296-310, 1997.

12. T. Tuan, K. Park, *"Congestion Control for Self-Similar Network Traffic,"* Dept. of Comp. Sci., Purdue Uni., *CSD-TR 98-014*, May 1998.

13. P. R. Morin, *"The Impact of Self-Similarity on Network Performance Analysis,"* Carleton University, Thesis, December 1995.

14. D. P. Heyman, T. V. Lakshman, *"What are the Implications of Long-Range Dependence for VBR-Video Traffic Engineering?"* IEEE Transactions on Networking, v:4, n:3, pp:301-317, June 1996.

15. F. Bonomi, K. W. Fendick, *"The Rate-Based Flow Control Framework for ABR ATM Service,"* IEEE Network, pp.25-39, March/April, 1995.

16. L. Benmohamed, S. M. Meerkov, *"Feedback Control of Congestion in Packet Switching Networks: The Case of a Single Congested Node,"* IEEE/ACM Trans. on Networking, v:1, n:6, pp. 693-708, December, 1993.

17. M.E. Crovella, *" Self-Similarity in WWW traffic: Evidence and Possible Causes,"* IEEE Trans. On Networking, v:5, n:6, pp. 835-845, December 1997.

18. W. Stallings, *" High Speed Networks; TCP/IP ATM Design Principles,"* Prentice Hall Inc, pp. 181-207, 1998.

19. R. Roberts, *"Variable-Bit-Rate Traffic Control in B-ISDN,"* IEEE Comm. Magazine, pp. 50-56, September 1991.

20. S. Guerin et.al., *"Equivalent Capacity and Its application to Bandwidth Allocation in High-Speed Networks,"* IEEE J. Selected Areas in Comm., v:9, n:7, pp.968-981, September 1991.

21. A. I. Elwalid, D. Mitra, *"Effective Bandwidth of General Markovian Traffic Sources and Admission Control of High Speed Networks,"* IEEE/ACM Trans. on Networking, v:1, n:3, pp. 329-343, 1993.

22. X. Wang, S. Jung, J. Meditch, *"Dynamic Bandwidth Allocation for VBR Video Traffic Using Adaptive Wavelet Prediction,"* 0-7803-4788-9/98, 1998 IEEE.

23. S. Chong, S. Li, J. Ghosh, *"Dynamic Bandwidth Allocation for Efficient Transport of Real-Time VBR Video over ATM,"* IEEE J. on Selected Areas in Comm., v:13, n:1, pp. 12-23, January 1995.

24. M. C. Yuang, P. L. Tien, *"Multiple Access Control with Intelligent Bandwidth Allocation for Wireless ATM Networks,"* IEEE J. on Selected Areas in Comm, v:18, n:9, pp.1658-1668, September 2000.

25. A. M. Adas, *"Using Adaptive Linear Prediction to Support Real-Time VBR Video Under RCBR Network Service Model,"* IEEE/ACM Trans. on Networking, v:6, n:5, pp. 635-644, October 1998.

26. Y. Afek, M. Cohen, E. Haalman, Y. Mansour, *"Dynamic Bandwidth Allocation Policies,"* in Proc. IEEE INFOCOM'97, v:3, pp. 1096-1104, 1997.

27. S. K. Biswas, R. Izmailov, *"Design of a Fair Bandwidth Allocation Policy for VBR Traffic in ATM Networks,"* IEEE/ACM Trans. On Networking, " v:8, n:2, pp. 212-223, April 2000.

28. E. W. Fulp, D. S. Reeves, *"Dynamic Bandwidth Allocation Techniques,"* Tech. Report TR-97/08, Center for Advanced Computing and Comm., North Carolina State Uni., August 1997

29. A. Adas, *"Supporting Real Time VBR Video Using Dynamic Reservation Based on Linear Prediction,"* In Pro. IEEE INFOCOM'96, pp. 1467-1483, 1996.

30. I. Hsu, J. Walrand, *"Dynamic Bandwidth Allocation for ATM Switches,"* Journal of Applied Probability, v:33, n:3, pp. 758-771, 1996.

31. S. Rampal, D. Reeves, Y. Viniotis, D. Argrawal, *"Dynamic Resource allocation Based on Measured QoS,"* In the Pro. Of the Fifth ICCCN-International Conf. on Computer Comm. Networks, pp. 24-27, 1996.

32. H. Zhang, E. W. Knightly, *"RED-VBR: A Renegotiation Based Approach to Support Delay Sensitive VBR Video,"* In Proc of NOSSDAV, pp. 258-272, 1995.

33. D. Reininger, G. Ramamurthy, D. Raychaudhuri, *"VBR MPEG Video Coding with Dynamic Bandwidth Renegotiation,"* In Proc. IEEE ICC'95, pp. 1773-1777, 1995.

34. E. W. Knightly, H. Zhang, *"D-BIND: An Accurate Traffic Model for Providing QoS Guarantees to VBR Traffic,"* IEEE Trans. On Networking, v:5, n:2,pp. 219-231, 1997.

35. M. Grossglauser, S. Keshav, D. N. C. Tse, *"RCBR: A Simple and Efficient Service for Multiple Time Scale Trafic,"* IEEE/ACM Trans. on Networking, v:5,n:6, December 1997.

36. J. Viterbi, J. K.Omura, *"Principles of Digital Communication and Coding,"* New York, McGraw Hill, 1979.

37. K. Lakshman, R. Yavatkar, *"An Empirical Evaluation of Adaptive QoS Renegotiation in an ATM Network,"* NOSSDAV 1996, Zuhst, Japan.

38. S. Gumbrich, H. Emgrunt, and T. Brown, *"Dynamic Bandwidth Allocation for Stored VBR Video in ATM End Systems,"* Proceedings of IFIP'97, April 28 May 2, 1997, pp. 297-317.

39. A. Feldman, A.C. Gilbert, W. Willinger, *"Data networks as cascades: Investigating the Multifractal Nature of Internet WAN Traffic,"* Proceedings of the ACM/SIGCOMM'98, September 1998, Canada.

40. P. Abry, D. Weitch, *"Wavelet Analysis of Long Range Dependent Traffic,"* IEEE Trans. Inform. Theory, v:44, pp. 2-15, Jan. 1998.

41. H. Saran, S. Keshav, *"An Empirical Evaluation of Virtual Circuit Holding Times in IP over ATM Networks,"* Proc. of INFOCOM 1994.

42. Y. Afek, M. Cohen, E. Haalman, Y. Mansour, *"Dynamic Bandwidth Allocation Policies,"* in Proc. IEEE INFOCOM'97, v:3, pp. 1096-1104, 1997.

43. S. K. Biswas, R. Izmailov, *"Design of a Fair Bandwidth Allocation Policy for VBR Traffic in ATM Networks,"* IEEE/ACM Trans. On Networking, " v:8, n:2, pp. 212-223, April 2000.

44. M. Gerla, W. Weng, R. L. Cigno, *"Bandwidth Feedback Control of TCP and Real Time Sources in the Internet,"* In the Proc. of GLOBECOM 2000, November 2000.

45. X. Wang, S. Jung, J. Meditch, *"VBR Broadcast Video Traffic Modeling- A Wavelet Decomposition Approach,"* 0-7803-4198-8/97 IEEE.

46. M. Doroslovacki, H. Fan, *"Wavelet-based Linear System Modeling and Adaptive Filtering,"* IEEE Trans. on Signal Processing, v:44, n:5, pp. 1156-1167, May 1996.

47. Z. Sahinoglu, S. Tekinay, *"Selfsimilarity and Its Effects on Network Performance,"* IEEE Comm. Magazine, January 1999.

48. R. Riedi, M. S. Crouse, V. J. Ribeiro, R. G. Baraniuk, *"A Multifractal Wavelet Model with application to Network Traffic,"* IEEE Trans. on Information Theory, v:45, n:3, April 1999.

49. Z. Sahinoglu, S. Tekinay, *"On the Optimal Detection and Measurement of Self-similarity in Network Traffic: A Comparative Study of Proposed Techniques,"* in the Proc of VTC'99, Vancouver, June 1999.

50. Z. Sahinoglu, S. Tekinay, *"Multi-resolution and Burstiness Analysis of Traffic Traces,"* In the Proc. of WCNC'99, New Orleans, 1999 September.

51. Z. Sahinoglu, S. Tekinay *"Efficient Parameter Selection for Use of Self-similarity in Real Time Resource Management,"* submitted to ICC'02, New York City, April 2002.

52. Z. Sahinoglu, S. Tekinay, *"A Novel Adaptive Bandwidth Allocation: Wavelet Decomposed Signal Energy Approach,"* GLOBECOM'01, San Antonio, October 2001.

53. Z. Sahinoglu, F. Porikli, *"Minimization of Bandwidth Renegotiations in ATM Networks,"* ICCCN'01, November 2001.

54. Z. Sahinoglu, F. Porikli, S. Tekinay, *"An Online Renegotiation Based Asynchronous Bandwidth Management to Provision QoS Constraints for VBR Traffic,"* submitted to IEEE Communications Letters.

55. Z. Pan, P. Mars," *Self-Similar Traffic Generation and Parameter Estimation Using Wavelet Transform,"* IEEE GLOBECOM 97, V:3, pp 149-23, November 1997.

56. J. Mah, M. Yuang, *"A Delay and Loss Versatile Scheduling Discipline in ATM Switches,"* IEEE 0-7803-4383, February 98.

57. S. Liang, M. Yuang," *Performance Analysis of Earliest Due Date Scheduling Discipline for ATM Switches,"* International Journal of Modeling and Simulation, V:17, N:4, 1997.

58. C. C. Li, S. L. Tsao, M. C. Chen, Y. Sun, Y. M. Huang, *"Proportional Delay Differentiation Service Based on Weighted Fair Queuing,"* In the proc. of 9th International Conf. on Computer Comm. Networks, pp.418-423, 2000.

59. Z. Sahinoglu, S. Tekinay, *"Real Time Measurement of Self-similarity in Resource Renegotiable Networks,"* submitted to IEEE Communications Letters, October 2001.