ABSTRACT

NALATWAD, SRIKANT S. Self-Sizing Techniques for Locally Controlled Networks. (Under the direction of Dr. Michael Devetsikiotis).

The Internet was designed to provide best effort service for delivery of data packets and to run virtually across any network transmission and system platform. Its exponential growth has turned it into a multiservice complex network of heterogeneous elements with dynamically changing traffic conditions. To regulate such a large scale network it is necessary to place intelligence in the nodes. Network control should be decentralized to make such a system reliable and manageable. It is necessary to find simple local rules and strategies that can produce purposeful and coherent behavior. These control mechanisms must be adaptive to effectively respond to continually varying network conditions. Such adaptive, distributed, localized mechanisms would provide a scalable solution for controlling these large networks. Our comprehensive study on *QoS developments in the Internet*, reveals the necessity and requirement of a new QoS framework which provides absolute guarantees to the underlying traffic. We propose an innovative self-sizing framework for *locally controlled* networks such as the Internet, which can support the stringent requirements of interactive applications. A "self-sizing" network can allocate link/switch capacity automatically and adaptively using online traffic data. Our unified, critical and comparative analysis of online resource allocation algorithms of two different classical approaches, leads us to a novel adaptive wavelet predictor. Our results show that by performing online resource allocation at each node based on their *local knowledge*, we can achieve considerable bandwidth savings and also satisfy QoS at the packet level. We further discover that by making some of the nodes aware of their neighbors resource availability, higher self-sizing gains can be attained.

Self-Sizing Techniques for Locally Controlled Networks

by

Nalatwad, Srikant S

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial satisfaction of the
requirements for the Degree of
Doctor of Philosophy

Computer Engineering

Raleigh

2005

Approved By:

Dr. Mihail Sichitiu	Dr. Khaled Harfoush
Dr. Michael Devetsikiotis Chair of Advisory Committee	Prof. Harry Perros

To my dear parents, SrinivasRao and Ushabai.

Biography

Srikant Nalatwad was born in Gulbarga, Karnataka, India. He finished his high school studies from Nutan Vidyalaya and preuniversity course from Sharana Basaveshwar College of Science. He received his Bachelors degree in Electronics and Communication Engineering from Poojya Doddappa Appa College of Engineering, Gulbarga, in 1989, and Masters degree in Electronics Engineering from Veermata Jijabai Technological Institute (then, Victoria Jubilee Technical Institute), Mumbai, in 1992. He has worked as a lecturer in Electronics Engineering Department of Poojya Doddappa Appa College of Engineering, and also at Khaja Banda Nawaz College of Engineering, affiliated to Gulbarga University, Gulabrga. He has worked as an Assistant Professor of Electronics Engineering Department at Thadomal Shahani Engineering College, Mumbai, from June 1994 to December 2000. Additionally he was also holding the position of In-charge HOD, from July 1999 to December 2000. His special interest in computer communication and networking lead him to pursue a PhD degree in Computer Engineering. His research interests include network traffic modelling and QoS provisioning in IP Networks.

Acknowledgements

It gives me a great pleasure to acknowledge the involvement of Dr. Michael Devetsikiotis, my advisor, and committee chair in my research work. While his faith in me provided encouragement, his guidance gave invaluable support and inspiration in the long journey of writing this dissertation. I would like to thank my committee members Prof. Harry Perros, Dr. Mihail Sichitiu and Dr. Khaled Harfoush for their comments and suggestions. This work drew upon the interaction and communication with one of my colleague, Dr. Bobby M. Ninan, who became a dear, treasured friend. I am thankful for his listening, his feedback and patience to endure my propensity to follow up on parallel interests. My special thanks to Dr. Dev Niyogi for his invaluable help and guidance at a critical juncture in my academic life. I would like to thank Alcatel, USA, for financially supporting my work in early years of PhD.

I would like to thank my parents (both were teachers) and grandparents for teaching me the importance of education in our life, which guided me to NCSU for my doctoral studies. This dissertation would not have been possible without Rajasri, my wife and my friend. I would like to thank her for believing in me and being with me during my difficult times. I am thankful to my father-in-law, Shashikant Kalmankar, who has been an inspiration to me and has provided support at all times. I would like to acknowledge the help of my brother-in-law, Mukund Kalmankar, during these years, who has become one of my closest friend. I thank my daughter Akanksha, and son Shreesh, for inspiring me in their own sweet ways. I would like to thank Sriram, Shaila, Sripad, Suhas, Girija and all other relatives who have helped me in making this happen.

I would like to thank my friend Ravindra Harsoor, for his moral and financial support throughout my graduate studies. I also thank my friends Arun Sonth, Eshwar Pastapur, T.S.Haraprasad, Ratnakar Shetty, and Ravindra Hegde for their continuous help and support. It would have been impossible for me to come to USA with out their help. I would like to thank all my teachers and friends who have helped me directly or indirectly during these five years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

The Web and the Internet together constitute a critical information, entertainment, and commercial infrastructure that is rapidly evolving from a best-effort service model to one in which service differentiation can be provided for users, services, and applications. In spite of having the best technology and sufficient bandwidth in the core, the Internet Service Providers are struggling to satisfy the stringent Quality of Service requirements of real time applications, such as interactive video and Voice over IP with their present QoS architecture. Future Internet is looking for a multiservice network framework, which can support such applications with absolute QoS guarantees and also use its resources efficiently. Even though, *major* portion of the current network traffic is still best-effort, emerging real time networked applications with different QoS requirements have become popular and have started contributing an appreciable amount to the total network traffic. Most of the multimedia applications demand stringent Quality of Service requirements in terms of bandwidth, delay or loss eg. VoIP needs loss $< 1\%$, one way latency $< 150ms$ and jitter$< 30ms$(as per DSCP EF RFC3246). Meeting these service requirements has to be facilitated by sorting the traffic into number of classes and will have to be transported with respective quantitative QoS guarantees throughout the network.

Today's multi-service Internet is facing the difficult challenge of allocating resources efficiently and to provide guaranteed QoS to an ever increasing network traffic, which is unpredictable, and whose statistical characteristics are unknown. Network researchers have reaffirmed that either capacity over provisioning or connection level resource reservations (static or dynamic) cannot provide a scalable solution to this problem. Differentiated Services (DiffServ) along with Multiprotocol Label Switching (MPLS) provides a scalable and powerful QoS architecture, either by providing more bandwidth to one traffic aggregate than other, or by implementing dropping preferences among aggregates. In spite of DiffServ being the most popular QoS framework, it cannot provide quantitative packet level service guarantees to real time applications such as Voice over IP (VoIP) and videoconferencing. There is an urgent need of an end-to-end QoS framework which can utilize resources optimally and also support such applications which need very strict QoS constraints to be satisfied.

There were doubts in the minds of a section of the Internet engineering and research community regarding the need of QoS mechanisms for IP networks. As a result, a workshop was conducted in October 2003 [5], to critique the evolution and deployment of QoS mechanisms in IP networks. The outcome of the workshop was very much encouraging and it emphasizes the need for deployment of QoS frameworks in future IP networks to support the increasing real time applications over the Internet. In [41], C. Macian observes that over-provisioning is not a viable option for ISPs and J. Crowcroft [35] argues that the ratio of access network and core network capacities change over time, moving the QoS problem back and forth, and the current core/access ratio is moving towards the core being a congested resource.

The self-sizing technique is an optimal network resource allocation mechanism based on traffic conditions observed in real time, which was proposed for ATM networks (globally controlled). A self-sizing network can allocate network capacity automatically and adaptively using on-line traffic data to satisfy the quantitative QoS at packet level. We consider the problem of extending the self-sizing framework to locally controlled networks such as the Internet, in which the resource allocation decisions are made at the node level. A multilevel view of the self-sizing network for locally controlled networks is shown in Fig. 1.1, which illustrates that the network resource optimization problem can be divided into a number of node level, link level and queue level dynamic resource allocation problems. In our work we show that by performing online resource allocation at each node based on their local knowledge, we can provide guaranteed QoS at packet level and also achieve

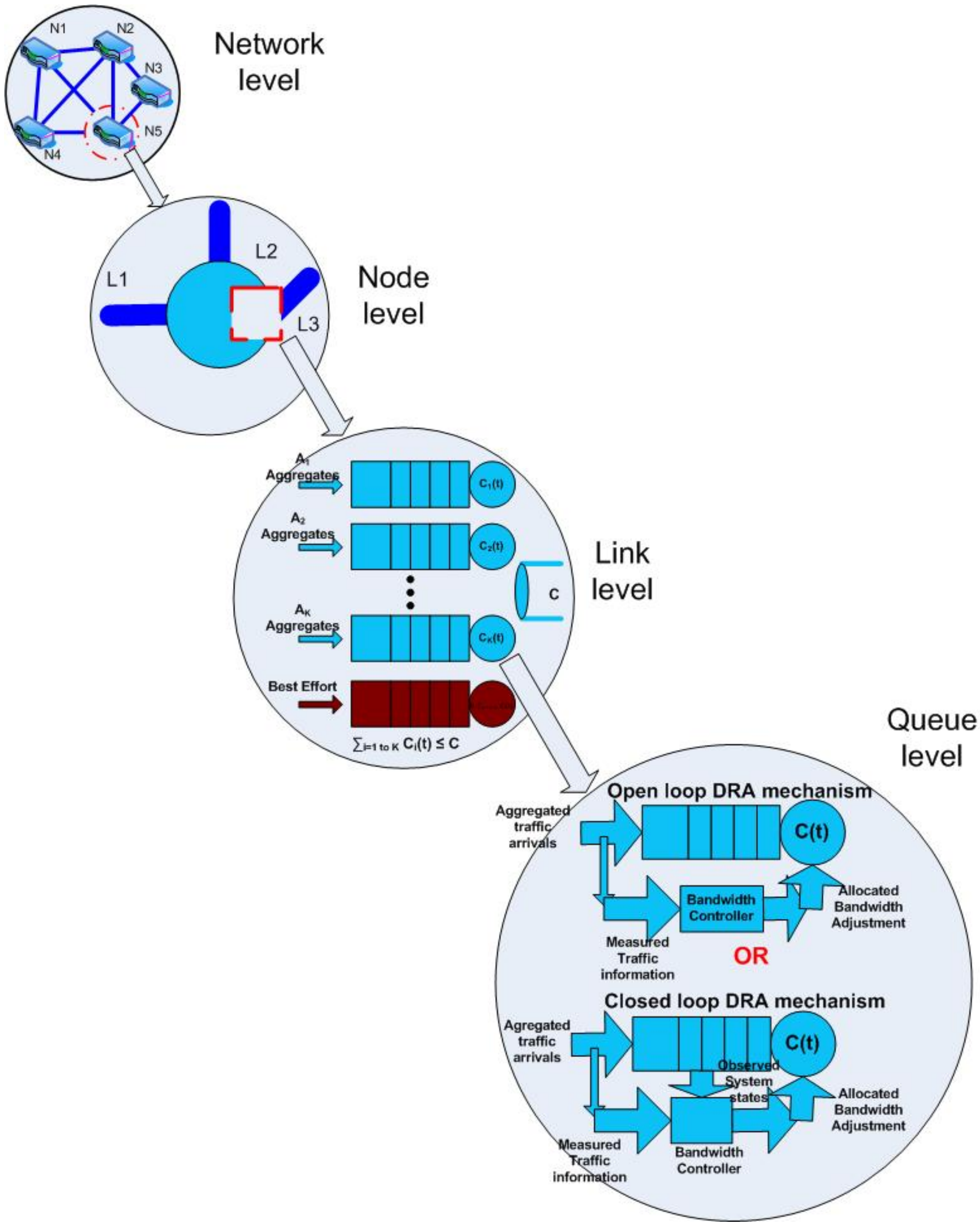

Figure 1.1: Architectural view of locally controlled Self-Sizing network

considerable bandwidth gains.

1.2 Internet: A Multiservice Network

Multi-service network design has always been a compromise between network efficiency and Quality of Service. Traditional telecom networks have provided the best quality for voice service for years at the cost of efficiency. ATM networks proposed by ITU-T were successful in providing QoS as well as statistical multiplexing to the packetized traffic, but failed to replace the Internet due to their inherent complexity. The Internet has been successful in getting most out of the network by fully sharing of network resources but is still struggling to provide quantitative QoS guarantees to its traffic.

The Internet was designed to provide best effort service for delivery of data packets and to run virtually across any network transmission and system platform. The increasing popularity of IP has shifted the paradigm from "IP over anything" to "everything over IP". After the advent of real time applications like streaming video, videoconferencing and voice over IP, demands for strict QoS constraints have rapidly increased. To facilitate true end-to-end QoS in an IP network, Internet Engineering Task force (IETF) proposed Integrated Services (IntServ) [54] in 1994 followed by DiffServ [46] in 1998.

In IntServ, the focus was on long-lived unicast and multicast flows and the Resource Reservation Protocol (RSVP) [12] was used with this approach. IntServ can provide the requisite QoS for each flow in the network, provided it is signaled using RSVP and the required resources are available. But there were many practical flaws in this approach, such as every device along the path need to be fully aware of RSVP and capable of signaling the required QoS. The soft state maintenance along with admission control increased the complexity of a node which made it difficult to support large number of reservations.

When the IETF realized that IntServ was not going to be implemented, it developed the DiffServ model, in which flows with the same service requirements are combined into a single aggregated flow. This aggregated flow receives levels of service relative to other traffic flows. This is accomplished by dividing the packets into different classes by marking the type of service (ToS) byte in the IP header. A six bit pattern called as the Differentiated Services Code Service Point (DSCP) in ToS byte of IPv4 octet or Traffic Class octet of IPv6. Along with MPLS, DiffServ presents a scalable architecture for QoS provisioning for the Internet. DiffServ can handle service differentiation and MPLS can control the data path. In spite of DiffServ being the popular QoS framework, it cannot provide quantitative packet level service guarantees to real time applications such as Voice over IP (VoIP) and videoconferencing.

1.3 Adaptive Resource Allocation Techniques for the Internet

Since static capacity allocation and over provisioning can not be a viable option, we need efficient dynamic resource allocation techniques (DRA) for multiservice IP networks to provide quantitative QoS to the aggregated network traffic. Due to unpredictable and unknown statistical characteristics of the aggregate traffic, the DRA method must be based on online traffic measurements. We conducted a detailed survey of DRA methods which have been proposed for the Internet in the past.

At flow level, quantitative QoS guarantees have been accomplished by static allocation methods, based on the knowledge of the stochastic properties of the incoming traffic [36, 27]. But these methods allocate the resources inefficiently, since the underlying traffic model may not accurately capture the statistic behavior real traffic. The heterogeneous mix of aggregates with different statistical properties can result in a traffic stream whose characteristics are not known. When this traffic flows though different nodes in the network, the traffic characterization is prone to change. Also user defined traffic descriptors may not actually represent the real traffic. When the traffic is generated from applications like videoconferencing, it is not possible to parameterize it before hand.

Measurement Based Admission Control Techniques (MBAC) used for connection admission control have overcome some of the above limitations by using measured values of traffic parameters along with the user declared parameters and analytical models. Many MBAC algorithms have been compared based on their performance by S. Jamin et.al. [34] in a simulated environment. An implementation based comparison of MBAC algorithms has been done in [44]. MBACs are designed to operate only in the ingress nodes, where admission decisions are taken and the period of execution of MBAC algorithms is at the *connection level* time scales. Also, in the present aggregate based traffic management and control scheme there is no per flow information available. So we can not use MBAC methods to provide guaranteed QoS in such scenarios.

Many researchers have instead proposed schemes to improve the quality of best effort traffic by providing a fair share of bandwidth and preventing congestion. A survey of such methods has been done by Gervos et.al. in [48]. All these techniques use scheduling mechanisms or queue management schemes to prevent misbehaved or aggressive flows from receiving more bandwidth than their fair share. Also they notify their sources of TCP flows to reduce their transmitting rates at the onset of congestion rather than waiting for packet loss to occur due to queue overflows. These methods are successful in reducing the congestion but they can not provide quantitative packet level QoS.

We need algorithms which are dynamic, simple and efficient to provide absolute QoS requirements for aggregate traffic. These algorithms take the QoS requirements as input and they provide absolute guarantees based on online traffic measurement without the knowledge of the underlying traffic model. Dynamic Resource Allocation (DRA) algorithms have been proposed by researchers in the past under different contexts. These algorithms have been studied in the literature as "Adaptive Bandwidth Control" algorithms [64] and "On-line Measurement based Capacity Allocation schemes " [28]. Even though the techniques used by both families of algorithms are different, they use the same working model and try to achieve the same requirements. In most of these techniques *bandwidth* is the resource which gets allocated dynamically. Since some of the algorithms [14] consider some other resources such as buffer in their implementation, we would like to address them under a single heading as "Dynamic Resource Allocation" algorithms.

1.4 Self-Sizing Networks

The concept of "Self-Sizing" was introduced in [71, 62] by industrial researchers from NTT and Nortel respectively. In [71], J. Yan proposes that self-sizing can be achieved by virtual band partitioning of network capacity, where each band corresponds to a traffic class. A multiservice network design process consists of two steps: determining the aggregate network topology and capacity, and the partitioning of the capacity among the elastic network bands. Since the aggregate network design is a slow process based on a long term forecast of the overall traffic, is normally done off line. But, the network bands must

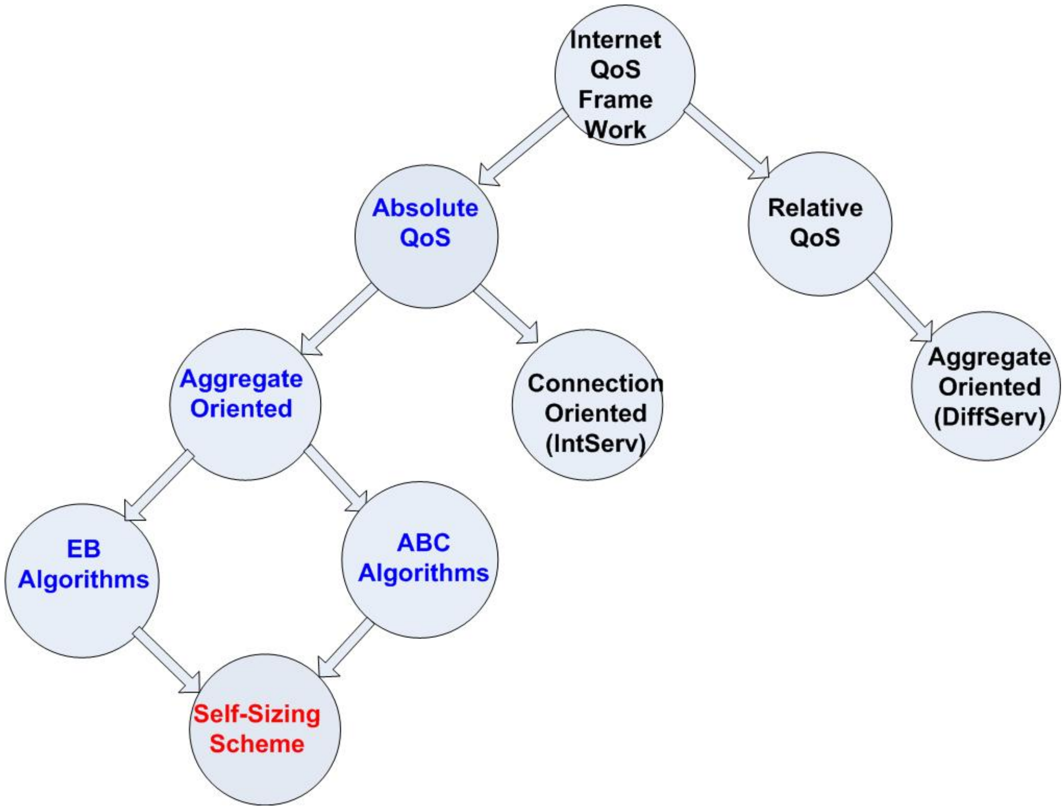

Figure 1.2: Positioning of Self-Sizing scheme in the Internet QoS

be modified frequently based on fresh traffic data and network state information. Our research on self-sizing framework for locally controlled networks can be positioned as shown in Fig. 1.2.

In [62], S. Shioda et.al. defines *self-sizing* as an autonomous adjustment of Virtual Path (VP) bandwidths based on traffic conditions observed in real time. By using the bandwidth demand concept they propose a VP-bandwidth sizing procedure in which real estimates of VP bandwidth demand and successive VP bandwidth allocation are jointly utilized. They develop an operating system to provide self-sizing functions, the experimental use of first prototype on an ATM testbed was carried out in 1996. They state that *self-sizing* in real ATM networks could not be implemented since the data in terms of number of cells was not collected at lower time scales.

Authors in [29], propose a an optimization model to partition bandwidth among bands to minimize total system cost under capacity constraints while the QoS at call level

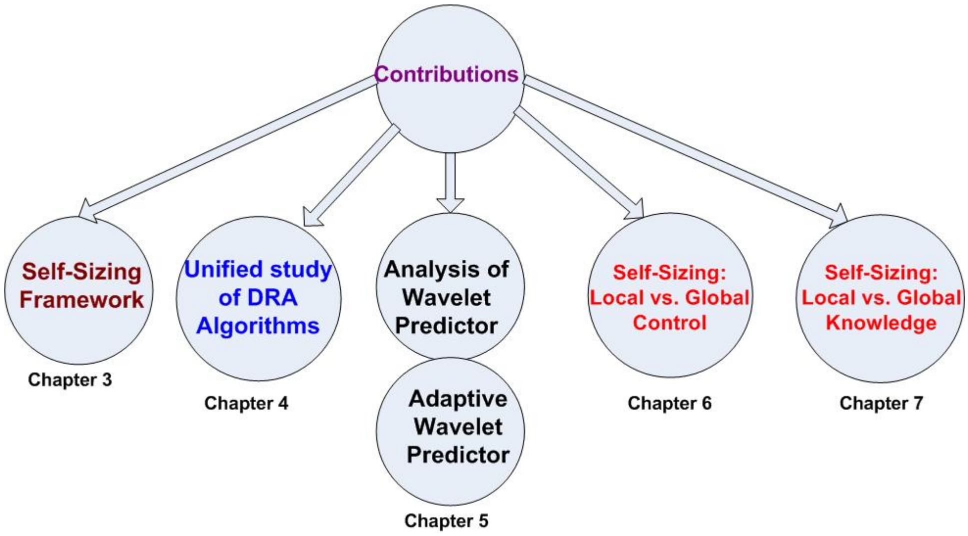

Figure 1.3: Contributions of this dissertation

and cell level are guaranteed. They also develop a fast algorithm based on simulated annealing for frequent band partitioning. In [60], authors introduce a general measurement-based self-sizing framework for resource allocation in label-switched networks. They also propose a hierarchical approach to improve adaptation performance for large networks and show that performance improves as the size of the network increases.

1.5 Contributions of this Dissertation

The significance of this dissertation is threefold. First, we identify the necessity and requirement of a new QoS framework for the Internet to support the stringent requirements of interactive traffic. Second, we introduce a novel wavelet-based adaptive resource allocation method for such a distributed network scenario. Third, we propose a self-sizing architecture to locally controlled networks and provide a scalable neighborhood control solution. The contributions have been shown in Fig. 1.3 with respect to the chapters in which they have been presented.

- We conduct a systematic and detailed study of the evolution of the Internet from providing a single best-effort service to providing multiple types of services with Quality-of-Service (QoS) guarantees in terms of packet loss or delay. We justify the need for such an adaptive, intelligent, and scalable QoS framework for the Internet.

- We propose a self-sizing framework for the *locally controlled* networks. Our results from a Local versus Global control experiment show that local self-sizing network can provide absolute QoS to the network traffic and also achieve considerable bandwidth gain.

- Adaptive resource allocation methods for the Internet have been proposed by two different research communities: One group of researchers belong to traditional *Effective Bandwidth* (EB) algorithms and the other group belongs to *Adaptive Bandwidth Control* (ABC) methods. We provide a unified, critical and comparative analysis of online resource allocation algorithms of these two different classical approaches.

- We investigate the role of *wavelet* as a traffic predictor and perform extensive experimental analysis in this regard. We show that wavelet can provide fast, accurate and robust online traffic predictor. We then propose an adaptive wavelet predictor technique based on the number of decomposition levels which increases the robustness of the predictor.

- We further define a novel *neighborhood control* self-sizing model in which each node has the knowledge of their neighbors resource availability. The results suggest that this framework is more efficient and stable than the locally controlled self-sizing architecture.

1.6 Outline

The following portions of the report are organized as follows: In the next Chapter we illustrate the evolution of the Internet as a multiservice network. We also justify the requirement and necessity of an absolute QoS framework for the Internet. In the third Chapter, we define the self-sizing framework for locally controlled networks. We present the comparative study of different DRA methods used in the Internet and highlighted their advantages and disadvantages in the fourth chapter. In Chapter 5, we introduce wavelets and their applications to network traffic modelling. In this chapter we discuss our experimentation with wavelet predictor and the comparison of its performance with the Gaussian predictor. We also propose the idea of an adaptive wavelet predictor. The details of the local versus global control experiments and their results are presented in the sixth chapter. Simulation based experiments demonstrate the advantage of neighborhood control over other local control techniques in chapter 7. In the last chapter we summarize our results and discuss the open issues.

Chapter 2

QoS Developments in the Internet

2.1 Introduction

In this chapter we discuss the different schemes developed by the Internet research community to accommodate the service differentiation. We also provide the justification for the requirement and necessity of a new framework which supports the real time applications over the Internet. IP was designed to provide best effort service for delivery of data packets and to run virtually across any network transmission and system platform. The increasing popularity has shifted the paradigm from "IP over anything" to "everything over IP". After the advent of real time applications like videoconferencing and Voice over IP, demands for service quality have rapidly increased. To facilitate true end-to-end QoS in an IP network, IETF (Internet engineering Task force) proposed IntServ [54] in 1994 followed by DiffServ [10] in 1998. The time line for the QoS mechanisms in IP networks is as shown in Table. 2.1.

2.2 IntServ

The focus was on long-lived unicast and multicast flows and the RSVP (Resource Reservation Protocol) [12] was used with this approach. IntServ relies on RSVP to signal and reserve the desired QoS for each flow in the network. A flow is defined as an individual, unidirectional data stream between two applications, and can be uniquely identified by the

Table 2.1: Time line for QoS developments in IP networks

Description	RFC Number	Year
IP Protocol	791	1981
Type of Service	1349	1989
IntServ	1633	1994
RSVP	2205	1997
Controlled Load Network Element Service	2211	1997
Guaranteed Service	2212	1997
DiffServ(DS Field)	2474	1998
DiffServ Architecture	2475	1998
Assured Forwarding PHB	2597	1999
Expedited Forwarding PHB (Revised)	3246	2002
2 bit DiffServ Architecture	2638	1999
IntServ over DiffServ	2998	2000
Aggregation of RSVP for IPv4 and IPv6	3175	2001
Multiprotocol Label Switching (MPLS)	3031	2001
Traffic Engineering	3272	2002

5-touple, i.e., source IP address and port number, destination IP address and port number and the transport protocol. IntServ assumes that all the nodes along the path from the source to destination support RSVP and can provide two different types of services. A very strict [61] guaranteed service that provides for firm bounds on end to end delay and assured bandwidth for the traffic that conforms to the reserved applications. The other service is a controlled load service [68] that provides for a better than best effort and low delay service under light to moderate network loads.

IntServ can provide the requisite QoS for each flow in the network, provided it is signaled using RSVP and the required resources are available. But there were many practical flaws in this approach, such as every device along the path need to be fully aware of RSVP and capable of signaling the required QoS. It was based on soft reservations at each node which are to be refreshed periodically, there by adding to the traffic on the network and increasing the chance that the reservation may time out if refresh packets are lost. The soft state maintenance along with admission control increased the complexity of a node which made it difficult to support large number of reservations. IntServ is not scalable since RSVP does not have the ability to aggregate individually reserved sessions into a single class.

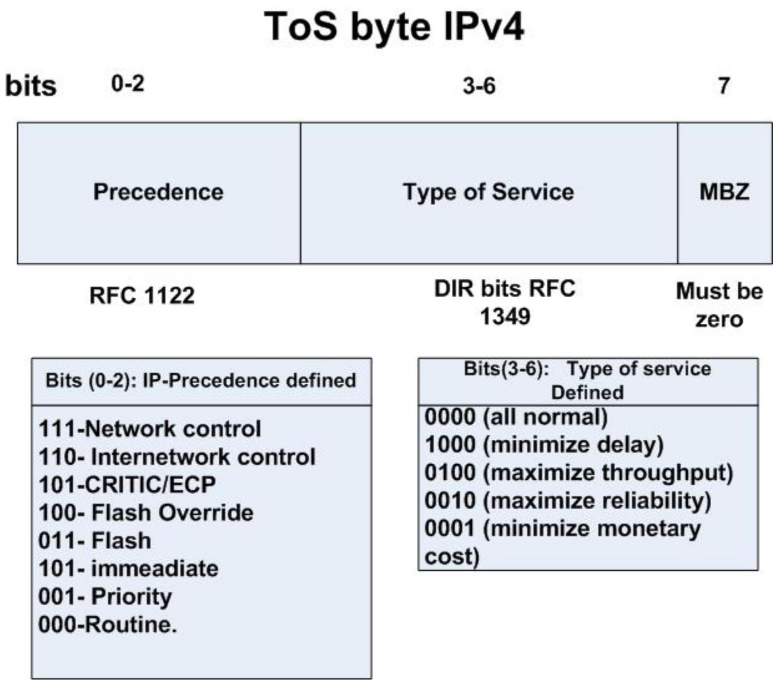

Figure 2.1: ToS byte as defined in original IPv4.

2.3 DiffServ

When IETF realized that IntServ is not going to be implemented, it developed a differentiated service (DiffServ) model, in which flows with the same service requirements are combined into a single aggregated flow. This aggregated flow receives levels of service relative to other traffic flows.

The ToS/IP precedence (Type of service) concept has been incorporated into the IPv4 since its definition, which can be seen in [32, 11]. IP protocol definition states that lower three bits (precedence bits) of the ToS byte can be used to classify packets at the edge of the network into one of the eight possible categories as shown in Fig. 2.1. Packets of lower precedence can be dropped in favor of higher precedence when there is congestion on a network. Also each packet can be marked to receive one of the two levels of reliability, throughput and delay. Later in [2] these bits were redefined to state the type of service in addition to priority.

Since the earlier definitions of the ToS/IP precedence were unable to provide the true “differentiated classes”, IETF came up with the architecture for Differentiated Services

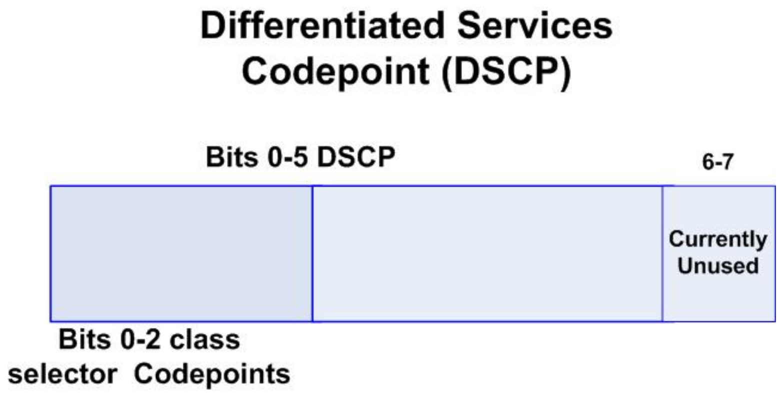

Figure 2.2: DiffServ codepoint field.

[46, 10]. The ToS byte was redefined as DS field [10] and six bits are now used to classify the packets as shown in Fig. 2.2. The six bits replace the three IP-precedence bits, and called the DSCP (Differentiated Services Code Point). In a given node with DSCP we can support up to 64 different classes or aggregates. DSCP was basically used to mark the packets and then they are forwarded by the nodes based on their behavior aggregate (BA).

RFC 3246 [17] defines Expedited Forwarding (EF) PHB which provides a low-loss, low-latency, low-jitter, and assured bandwidth service. Real time applications like voice over IP, video conferencing and online trading programs which need such robust network-treatment are supported with this PHB. EF can be implemented using priority queuing, along with rate limiting on the class. The recommended value for EF is 101110 and suitable for applications (like VoIP) which require very low packet loss, guaranteed bandwidth, low delay, and low jitter. Another important PHB defined in RFC-2597 [30] is Assured Forwarding which is rough equivalent of the IntServ Controlled Load Service. This defines a method by which different forwarding assurances can be given to different Behavioral Aggregates. Here traffic can be divided into gold, silver and bronze classes with gold being offered highest quality and bronze being lowest quality.

In order for a customer to receive differentiated services from an Internet Service Provider (ISP), the customer must have a Service Level Agreement (SLA) with that service provider [69]. The Service Level Agreement basically specifies the number of service classes supported and the total amount of traffic allowed in respective classes. The static SLAs are negotiated on a regular basis (monthly or yearly) and Assured Services can be supported

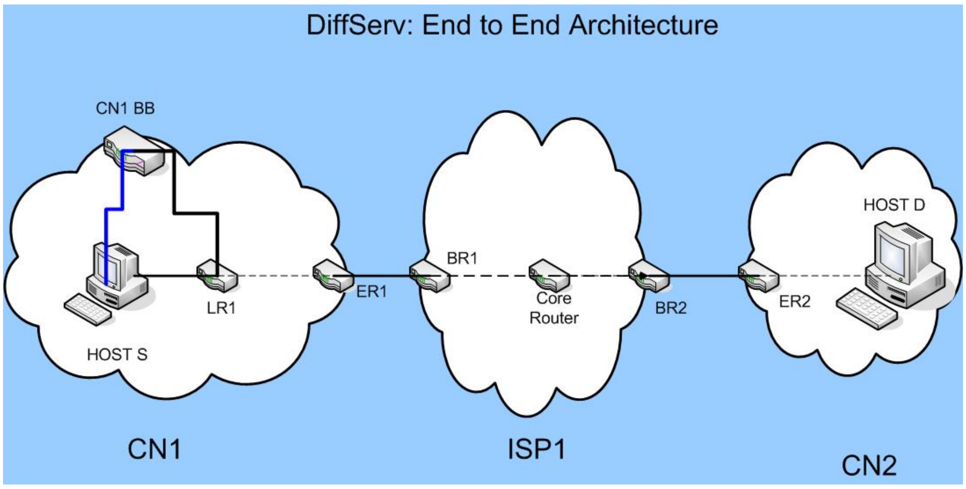

Figure 2.3: Assured Service delivery process with a static SLA.

using this type of agreement. When the customers need premium service (EF) they use Dynamic SLAs, which must use a signaling protocol like RSVP to request services on demand.

2.4 End to End DiffServ Architecture

A typical end to end DiffServ architecture may consist of one or more ISP DiffServ domains in between two corporate networks or customer domains as shown in Fig. 2.3. This architecture is based on RFC 2638 [47] and can provide both assured service and premium service in addition to best effort service. Service allocation in both customer domains and ISP domains are done either by the hosts/boundary routers or by resource controllers called Bandwidth Brokers (BB).

In Assured Service implementation, classification and policing are done at the ingress routers of the ISP networks. The traffic is considered *in* profile if it does not exceed the bit rate specified by the SLA else the excess packets are considered *out* profile. To avoid out of order delivery, all the packets, in and out, are placed into an assured queue (AQ). And the queue is managed by RIO, random early detection (RED) with in and out queue management scheme.

Premium service can be associated with either static or dynamic SLAs. At the customer site some entity will decide which application flow can use the premium service. The leaf routers connected to the hosts will do MF (multifield) classifications and shape the traffic. Conceptually there is a P bit in the DS field, if set the packet belongs to premium service else packet belongs to assured service or best effort class. The egress routers of the corporate network might need to reshape the traffic to make sure that the traffic does not exceed the peak rate specified by the SLA. Again at the ISP domain ingress routers police the traffic and excess packets are dropped. The packets with "P-bit" set enter into a premium queue and potentially they can use hundred percent of the bandwidth of output links. However, premium service does not provide any guarantee on the delay or jitter.

2.5 IntServ over DiffServ

Scalable admission control methods have been proposed for IP networks based on the IntServ over DiffServ architecture in RFC 2998 [9] and integrated IntServ/RSVP framework. In the former method IntServ is supported in the access networks and DiffServ in the backbone network as shown in Fig. 2.4. IntServ access networks are connected through a virtual link provided by DiffServ cloud [42]. DiffServ works to allocate backbone network resources to connect the access networks. IntServ reallocates the allocated resources to each call to satisfy the resource request. Data packets carry the signaling messages like RSVP PATH and RESV in the DiffServ backbone network.

Admission control is carried out by the access routers based on the instructions given by policy server. The backbone might consist of one or many DiffServ domains and the resources are allocated to the aggregated flows passing between the domains based on SLAs. Dynamic SLAs and resource allocation with support of a bandwidth broker are desirable when there is a change in the traffic characteristics. But, if resource allocation between domains does not correctly reflect the characteristics of the aggregated traffic, admission control for each call in the access network may not be consistent with the virtual link congestion state, and individual QoS requirements may not be satisfied.

RFC 3175 [7] proposes an aggregated RSVP framework where in scalability has been provided by aggregating the states in the router or employing resource reservations between subnets. But each flow is not completely isolated in the resource allocation since multiple flows share the same service class.

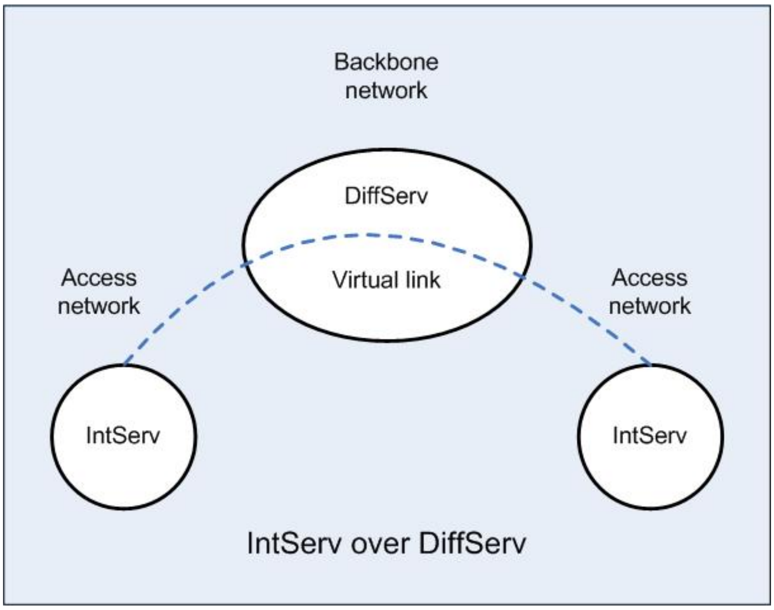

Figure 2.4: Framework for IntServ over DiffServ.

2.6 Multiprotocol Label Switching (MPLS)

Multiprotocol Label Switching is a packet forwarding scheme evolved from Cisco's Tag switching. Motivation for using MPLS is to use a fixed length label to decide packet handling and it is also a useful tool in Traffic Engineering. In the OSI seven layer model it can be placed between layer2 (link) and layer3 (network). Therefore, the MPLS header is encapsulated between link layer header and the network layer header.

The MPLS packet header consists of a 20-bit label, a 3 bit Class of Service (COS) field, a 1 bit label stack indicator and an 8 bit Time to Live (TTL) field [21]. Label switched router (LSR) is a MPLS capable router, checks only the label in forwarding the packet. For MPLS the network protocol can be either IP or any other network protocol, hence the name *multiprotocol* label switching. MPLS uses either Label Distribution Protocol (LDP) or extended RSVP to distribute labels to set up label-switched paths (LSPs). An LSP is similar to a virtual circuit as incase of ATM networks and is unidirectional from sender to receiver. LSP set up can be triggered by routing updates (control driven) or triggered by the request of a flow or traffic trunk (data driven). Here the traffic trunk is an aggregation of flows with the same class of service. An LSP can be either specified by the sender explicitly

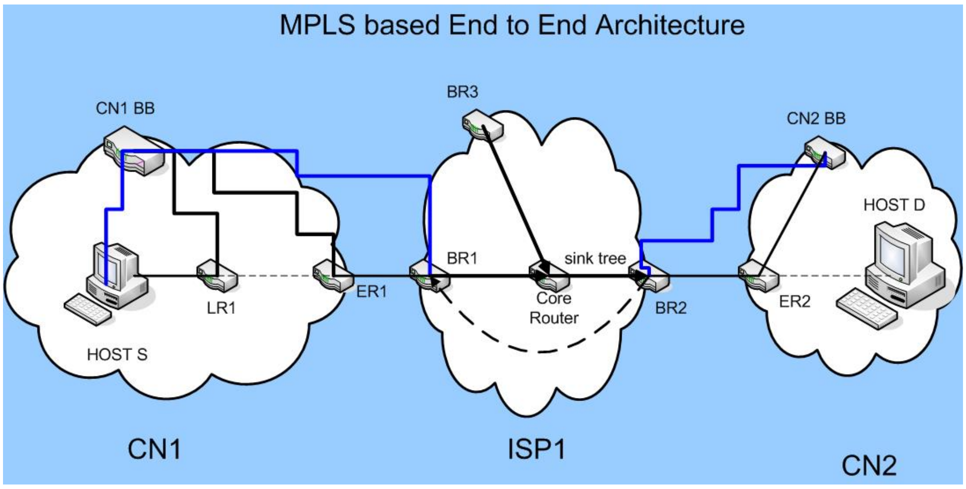

Figure 2.5: Dynamic Premium Service in a MPLS-based Architecture.

(ER) or it can be same as the one inlayer 3 (hop-by- hop route).

Packets are classified and routed at the ingress LSRs of an MPLS capable domain and the headers are then inserted. A LSR uses this label as an index to look up the forwarding table which is faster than normal longest match IP routing. The outgoing label replaces the incoming label and then switches the packet to next LSR. Packet forwarding, classification and QoS service are determined in MPLS by labels and COS fields which makes core router functioning simpler. The MPLS label is removed when the packet leaves the MPLS domain. MPLS LSPs can be configured as tunnels where in a packets path can be completely determined by the label assigned by the ingress LSR. MPLS tunnelling mechanism is unique in the sense that it can control the complete path of a packet with out explicitly specifying the intermediate routers. An end-to-end MPLS based service architecture is shown in Fig. 2.5, which can also be used along with differentiated services to provide QoS.

In the above architecture, LSPs are first configured between ingress and egress routers. To reduce the number of LSPs we can merge all the LSPs which run between same ingress and egress routers in to a sink tree. We can also use the same sink tree to transport packets belonging to different traffic classes and use the different COS bit combination to differentiate between them. As the number of flows increase in this architecture, number of flows in each LSP or sink tree increases. This feature makes MPLS a scalable architecture.

The functioning of the above architecture is similar to Fig. 2.3. which is explained in case of DiffServ, except following differences:

At the ingress of the ISP network a MPLS header is inserted into the packet. Core routers process the packet based on MPLS label rather than DS field. At the egress inter-domain LSPs are configured and the MPLS header is removed.

The MPLS effect is confined to ISP domains that use MPLS. Therefore we can inter-operate between MPLS and DS field domains with out extra effort.

2.7 Traffic Engineering and Constraint Based Routing

All the QoS techniques are unnecessary if the network is not under congestion or operating at light load. So if we find mechanisms to reduce the congestion in the network we may not need to device QoS mechanisms. The motivation for Traffic Engineering is avoiding congestion. Congestion in the network may occur due to scarcity of resources or uneven distribution of network traffic. In the former case when all the links and queues are overloaded and the only solution is to add more resources to the network by upgrading the infrastructure. But in the other case, some of the resources are heavily loaded and some of them are under loaded. This uneven allocation of resources may be due to the dynamic shortest path routing techniques such as OSPF, RIP and IS-IS. These protocols tend to route the traffic through the shortest path and thereby causing congestion on that route. But recent versions of OSPF and IS-IS with options like equal cost multi-path try to distribute load to several shortest paths. Traffic Engineering (TE) is the process of distributing the traffic in network such that congestion caused by uneven network utilization can be avoided [6].

Constraint based routing (CBR) is nothing but computing routes that are subject to multiple constraints. It has evolved from QoS routing, which finds the best route that most likely to be able to satisfy the QoS requirements for the given flow or aggregation of flows. In constraint based routing even other constraint such as policy is also considered. Basic aim of the constraint based routing is to satisfy QoS constraints as well as to increase the network utilization. Some times CBR ends up finding a longer but lightly loaded path over the heavily loaded shortest path. To compute the QoS route, the router needs knowledge of topology as well as resource availability. This has been incorporated into the extended shortest path routing protocols OSPF and IS-IS which include the bandwidth availability information in their link state advertisements [3]. The complexity of routing table computation algorithms in CBR depends on the number of constraints (metrics) chosen for the route selection. Common metrics used are bandwidth, delay, jitter, hop count, reliability and monetary cost.

As CBR algorithms re-compute routing table more often than traditional dynamic routing algorithms, they can introduce instability in the network. High computation overhead due to CBR implementation may also cause instability. This can be overcome to some extent by carefully choosing the timer value for re-computation and using efficient algorithms. Even though CBR chooses a route which satisfies QoS constraints it can not replace differentiated services, but it helps in achieving them. CBR and RSVP are basically independent but they are complementary to each other. When both are used CBR chooses the path and then RSVP reserves resources on it. Also MPLS and CBR are mutually independent. MPLS uses its label distribution protocol to set up the LSPs and does not care about the route selection. But MPLS per LSP statistics provide CBR with precise information about the amount of traffic between every ingress and egress pair. With such information CBR can better compute the routes for setting up LSPs.

Table. 2.2 shows the summary of the above discussion which depicts the relative positions of the different QoS framework components:

Table 2.2: Relative positions of the different QoS components.

Application Layer	
Transport Layer	Integrated Service/ RSVP Differentiated Services
Network Layer Link layer	Constraint based routing MPLS

2.8 Why do we still care for QoS in IP Networks ?

A workshop was conducted in October 2003 [5] to critique the evolution and deployment of QoS mechanisms in IP networks. The workshop tried to discuss/answer some of the following questions which were there in the minds of QoS research community: Is "IP QoS" research a game for academic intelligentsia? Do we have enough money to implement such mechanisms and these mechanisms have enough ability to return the investments? Is providing more bandwidth not a solution? It is important to discuss outcome of the workshop as they lay foundation for future multi-service IP networks.

In [41] Macian observes that over-provisioning is not a viable option for ISPs, ultimately boiling down to realities. Links of suitable capacities may be available and yet priced too high to justify, or there may simply be no links available between different sites on a provider's IP network. So we need to deploy alternative mechanisms to share the existing links in a controlled and differentiated manner. Even though technological advances have taken place, the business environment still has a long way to go. The solution needs to be completely end to end or not at all, which implies inter-domain relationships to support QoS across provider boundaries. Also QoS area lacks a billing and accounting model that can work inter-provider and inter-domain.

Despite the doom and gloom from many quarters of research community, there is considerable deployment of basic Differentiated Services (DiffServ) in real world networks solving real world problems [16]. B. Davie's paper has a presumption that over-provisioning was not the solution and Traffic Engineering was not defined as the QoS mechanism. Most DiffServ deployment today was in private networks or managed VPN contexts. It is commonly deployed by ISPs supporting customers who have high expectations of mixing Voice over IP and regular IP traffic over shared links running at very high utilizations. The VPN scenarios are tractable since they avoid the issues of maintaining QoS across inter-domain boundaries or having to mix IP traffic from uncontrolled and uncontrollable sources. This paper also observes that future QoS research needs to look at deployment issues rather than develop new mechanisms.

J. Crowcroft [35] argues that the ratio of access network and core network capacities change over time, moving the congestion (hence forth QoS) problem back and forth. As QoS research tracks resource constraints in networks, hence a researcher's work tends to be a trailing indicator of where the access/core ratio stood at the time they embarked on their particular QoS scheme. A quarter of research community failed to realize that we are victims of the *wheel of time* which ensures that we do not break out of cycle. He concludes that we need QoS at lowest layer, below IP and it only needs to be very simple- a two level (one bit) will suffice. We need to map QoS to marketable services, close the gap between mechanisms and revenue. Also we need QoS mechanisms in the core. The current core/access ratio is trending towards the core being a congested resource, despite all current evidence to the contrary. Aim for the future, don't aim for the near term issue.

2.9 Summary

In this chapter we have seen the development of QoS schemes starting from flow based IntServ to an aggregate based DiffServ framework. Differentiated Services (DiffServ) along with Multiprotocol Label Switching (MPLS) provides a scalable and powerful QoS architecture, either by providing more bandwidth to one traffic aggregate than other, or by implementing dropping preferences among aggregates. We understand that in spite of DiffServ being the most popular QoS framework, it cannot provide quantitative packet level service guarantees to real time applications such as Voice over IP (VoIP) and videoconferencing.

The discussion in section 2.8 justifies the need of an end to end QoS framework which can utilize resources optimally and also support such applications which need very strict QoS constraints to be satisfied. We define the Self-Sizing frame work for locally controlled networks in the next chapter.

Chapter 3

A Framework for Local Self-Sizing

3.1 Overview

In this this chapter we define the self-sizing framework for a localized network such as the Internet. Multilevel view of the self-sizing network is shown in Fig. 1.1, which illustrates that the network resource optimization problem can be divided into number of node level, link level and queue level dynamic resource allocation problems. Our main focus in this dissertation is the network level and queue level strategies. The network level problem is studied in chapters 6 and 7 by comparing the performances of local versus global schemes. At the node level admission control becomes a problem as we don't know when the new aggregates are going to join the queue. This problem can be dealt using the measurement based admission control with aggregate traffic envelopes as given in [52].

The link level problem arises when more than one class share the resources, the single queue control must be extended to deal with fair access as well as priority access. In [13] the authors consider the case of multiple ATM virtual circuits by allocating the link capacity at time t proportional to its required bandwidth derived from the control algorithm. Since the link capacity is finite, the bandwidth request due to the control is not always satisfied. so we need to devise a control which enables each queue to equally access the resource. We can formulate some mechanisms to prevent bandwidth hogging of one class or aggregate over the other. Or we can consider a priority access where in one

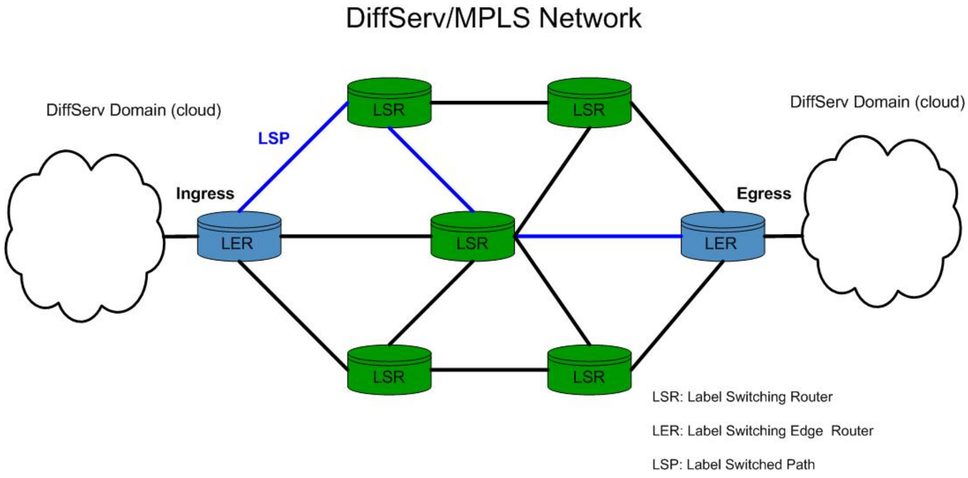

Figure 3.1: Basic block diagram of a DiffServ/MPLS network

class may be preferred over the other. The queue level problem is basically to predict the incoming traffic and allocate bandwidth based on this prediction.

The model used for locally controlled network can be used to describe the DiffServ/MPLS network framework, in which QoS is differentiated on a per node, per class basis. A typical block diagram of a DiffServ/MPLS network is as shown in Fig. 3.1. It consists of two Diffserv domains connected to ingress and egress LERs (Label Switched Edge Router) of a MPLS enabled network. MPLS domain consists of LERs, LSRs (Label switched routers) and LSPs (Label Switched Paths).

The model can also be used to describe a Virtual Private Network (VPN) environment, in which each queue is dedicated to aggregate traffic belonging to a particular VPN user. A more complex scenario might consist of each VPN user being assigned a fixed bandwidth pool which supports multiple traffic classes to provide service differentiation to traffic within the same VPN user. Fig. 3.2 shows a typical virtual private network scenario, which consists of three customer routers and six service provider routers.

3.2 Modelling Self-Sizing Networks

Consider the output port of an output queued node, where the port has an outgoing link capacity of **C** bits/sec. It is assumed that the port supports **K** traffic classes (QoS)

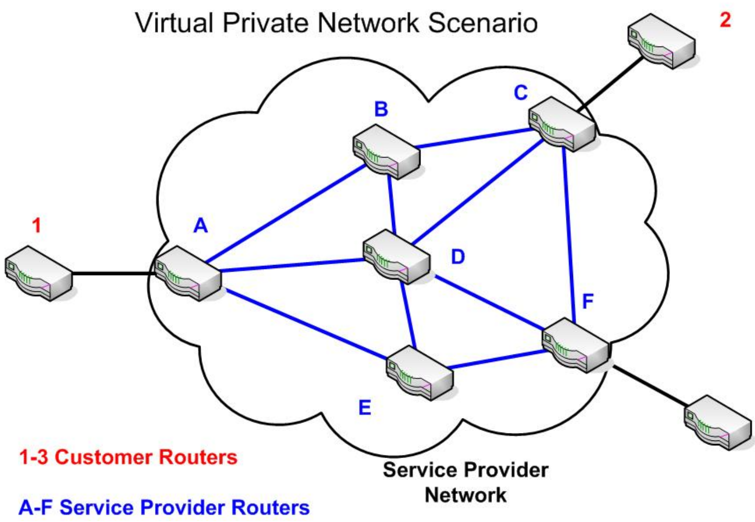

Figure 3.2: Basic block diagram of a Virtual Private Network scenario (VPN)

and one best effort traffic class, each of which has its own logical queue as shown in Fig. 3.3. We consider the problem of QoS guarantee for traffic in each QoS queue by allocating appropriate amount of bandwidth $C_i(t)$, $i = 1, 2, ...K$ with the constraint that

$$\sum_{i=1}^{K} C_i(t) \leq C \tag{3.1}$$

The remaining unallocated bandwidth is consumed by best effort traffic whose value is equal to $C - \sum_{i=1}^{K} C_i(t)$. According to many of the DRA methods discussed in this proposal, QoS guarantee can be achieved independently in each queue, with the constraint on the output link capacity.

The single queue model for the local self-sizing network can be either closed loop or open loop as shown in Fig. 3.4. In open loop control method the input traffic is predicted based on the past history. The service rate is then adjusted to match the predicted rate to attain zero packet loss or low queueing delay. In the closed loop control/feedback control method, any QoS metric such as packet loss or average queue length are observed to provide the feedback to adjust the allocated bandwidth. These issues are discussed in detail in Chapter 4.

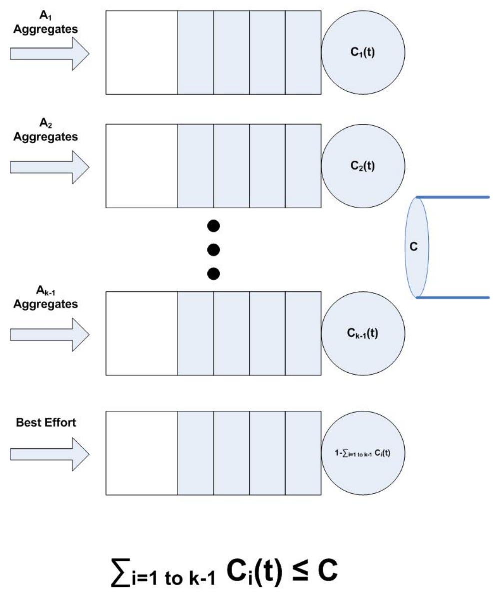

Figure 3.3: Link level model for Locally controlled Self-Sizing network

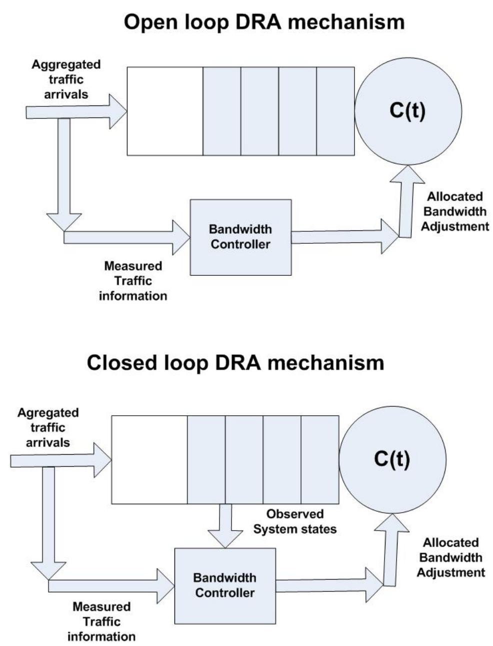

Figure 3.4: Single queue level models for Locally controlled Self-Sizing network

3.3 Summary

We have defined the self-sizing framework for a locally controlled network in this chapter. From Fig. 1.1 it follows that the main constituent of the local self-sizing framework is the DRA algorithm used to compute the allocated bandwidth. So, in the next chapter we present a unified study of the different DRA algorithms proposed for multiservice IP networks.

Chapter 4

Unified study of Dynamic Resource Allocation Methods in Multiservice IP Networks

4.1 Overview

Dynamic Resource Allocation algorithms (DRA) have been proposed by different researchers in different contexts. They can be classified by the control technique used or by the QoS metric which they can guarantee or the domain in which they are performing their prediction. Fig. 4.1 gives the classification of the algorithms studied in this document and also shows typical characteristics. Some of these algorithms have been studied in the literature as Adaptive Bandwidth Control algorithms [64] or On-line Measurement based Capacity Allocation schemes [28]. Even though they have been studied and used in different contexts, all these algorithms use the similar working model. In our unique study we have combined theses algorithms and studied them based on their control scheme and the QoS metric.

Table 4.1: DRA algorithms for IP Networks and their typical characteristics

Control Technique	Algorithm	QoS Metric	Domain	Special Features
Open loop	Guassian (1991)	Packet loss	Time	Simple, inefficient, can be used as upper bound
	Kesidis (1993)	Delay	Time	Fits measured queue statistics to specified queue length distribution, expensive
	Courcoubetis (1994)	Packet loss	Time	Uses Index of Dispersion, good for large buffer
	Chong(1995)	Packet loss	Frequency	Sets BW as peak rate of low-pass filtered traffic,need to find cut off frequency offline
	Norros(1995)	Packet loss	Time	Good for both SRD and LRD traffic, needs Hurst value
	Adas (1998)	Packet loss	Time	Uses Least Mean Square error predictor, large error
	Duffield(1999)	Packet loss	Time	Local maximum and Guassian predictors, accurate and robust
	Wentao(2002)	Packet loss	Time	Perfect online algorithm, less complex,poor with LRD
Closed loop	Pitsillidis (1995)	Queue length	Time	Control based on M/M/1 queue, unrealistic model
	Hsu(1996)	Packet loss	Time	Simple, proportional loss controller, poor performance
	Liao(2002)	Packet loss	Time	Maps target to loss to target, poor performance
Hybrid	Sahinoglu(2001)	Queue length	Time and Frequency	Uses wavelets, good performance

4.2 Open Loop Control

The different DRA algorithms we reviewed under this category can be classified based on the domain in which they treat the traffic measurement history, to compute the future capacity allocations to satisfy the QoS requirements, e.g., time domain,frequency domain. These techniques are basically used to provide very low or zero packet loss guarantee to the arriving traffic. All the EB algorithms, Adas and Duffield work in the time domain and Chong's algorithm is an example for the frequency domain technique.

4.2.1 Time Domain Techniques

LMS Algorithm

In [72], the D-BIND deterministic traffic model proposed in [38] is fitted to M successive frames of the video sequence in real-time, and the bandwidth is adjusted accordingly to match the worst case traffic arrivals to guarantee zero loss. But here the deterministic traffic can result in underutilization and also the fitting process is computationally intensive. Adas uses the Least Mean Square (LMS) error linear predictor to predict the bandwidth requirement of future frames in MPEG video sequences [1]. The Pth order predictor has the form

$$x^*_{k+n} = \sum_{l=0}^{p-1} w_l x_{k-l} \tag{4.1}$$

Where x_k represents the frame rate at time step k, and w_l are the predictor filter coefficients (previous P values are used). In LMS method, w_l are time varying and determined from the prediction error $x_{k+n} - x^*_{k+n}$ and previous values of x_k. For n=1 and p=12, it has been shown that the bandwidth allocated according to predicted values results in less delay and fewer buffers needed. The LMS technique is adopted in [24] for the dynamic bandwidth allocation of self-similar video conferencing traffic. Even with a predictor of the order of p=20, we find that the error is as high as 10%. In [23], Gallardo *et al* propose stable self similar process to model aggregate traffic. The predicted traffic information is used to control the token rate of a leaky bucket regulator at the edge node to reduce packet loss.

Duffield Algorithm

Duffield *et al* in [19], consider adaptive bandwidth allocation in the context of VPNs where they suggest dynamically resizing a VPN link capacity in their VPN hose model to attain better bandwidth sharing among traffic aggregates in a sink tree VPN. In the hose model, a VPN customer specifies a set of end points to be connected to, as well as a rough estimate of incoming and outgoing aggregate traffic from and to those end points, comprising the required bandwidth on the access link to the network. From initial bandwidth assignments bandwidth on both access link and internal links are dynamically resized. The bandwidth resizing is done from the ingress node, which in turn signals the reallocation along the path to the egress node.

Even though the objective of this work is not to provide zero loss, the two methods are discussed here because the required bandwidth, which is calculated from time domain prediction from past traffic arrivals, is intended to match the input traffic. In the local maximum predictor, where by the maximum of the sampled rates over the measurement window is used. In the local Gaussian predictor method, it sets the allocated bandwidth to $m+\alpha\sqrt{v}$, where m and v are, respectively, the mean and variance of the sampled rates over the measurement window. The value of α is selected such that the arrival rate will exceed allocated bandwidth with probability $\zeta = 1 - \phi(\alpha)$, where $\phi(\alpha)$ is a standard normal CDF. The paper also discusses the robustness of the predictors with respect to non-stationarity provided that measurement time scale is less than the non-stationary time scale. The measurement error can be avoided for n samples by modifying α to

$$\sqrt{(n+1)(exp((\phi^{-1}(1-\zeta))^2/n-1)} \tag{4.2}$$

To deal with burstiness in multiple scales, which results in underestimating the required bandwidth, it is suggested that a priori knowledge of the rate variance at different time scales be used to correct α. It has been shown that zero loss can be achieved with Gaussian traffic rate as input however as the input traffic is changed to an aggregate of Pareto on/off sources, the performance of local Gaussian predictor degrades, resulting in some packet loss. Therefore obtaining the right α can be a difficult task.

Kesidis algorithm

G. Kesidis in [37] considers bandwidth adjustment to guarantee a probabilistic delay bound. For a single server with service rate C, the probabilistic delay bound QoS requirement is that the probability that a packet queuing delay d will be greater than D is less than ε, i.e., $P(d > D) <\varepsilon$. The delay bound is converted into the bound on the queue tail distribution, $P(Q \geq \theta)$ where $\theta = D * C$. The problem is to estimate $P(Q \geq \theta)$ through online measurement and adjust C from its initial allocation so that QoS requirements are met.

The problem can be solved in two steps. First the tail distribution of the queue occupancy Q must be assumed to follow some specific distribution. Then change in the queue threshold violation probability $\Delta\varepsilon$ as the bandwidth changes from C to $C + \Delta C$ must be determined . The primary drawback of this method is that the assumption of a specific queue length distribution may not hold and the relationship between $\Delta\varepsilon$ and ΔC may not be easily determined for other probability distributions.

4.2.2 Effective Bandwidth Algorithms

The *effective bandwidth* technique is a simple way of relating the queuing performance through large deviation theory concepts [36, 25, 18, 20]. Although originally conceived as a large deviations/analytical approach, it can also be used as traffic measurement technique used to satisfy the QoS requirements implicitly. There are many EB estimation algorithms available in the literature [28].

The algorithms in this section take a window of traffic measurements as input, estimate the parameters they need in a bandwidth calculation formula and output the required amount of capacity to be reallocated. The measurements correspond to the amount of the incoming traffic during a *slot duration*, t_{slot} (Every measurement is a sample of the arrival process $X(0, t_{slot})$, where $X(0, t)$ denotes the arrival process in the interval $[0, t]$). Consequently, the algorithm is called periodically every $N * t_{slot}$ seconds (the reallocation period), where N is the window size in terms of slots.

Gaussian Approximation Algorithm(GA)

Even though this algorithm allocates the bandwidth in a very *loose* way compared to other techniques (since it ignores the buffer), it is computationally very simple [28]. According to this technique, the distribution of the arriving traffic can be approximated to be Gaussian if there is a considerable aggregation of traffic sources [58]. The following formula gives the effective capacity according to Gaussian distribution such that QoS requirement (probability of loss) is satisfied by the tail probability:

$$C = m + \sigma * \sqrt{-2 * \ln{(P_{loss})} - \ln{(2 * \pi)}} \tag{4.3}$$

where,

m —> mean of the traffic

σ—> variance of the traffic

P_{loss} —> probability of packet loss

Direct Effective Bandwidth Allocation (DEB)

The DEB algorithm relies on a direct analytical evaluation of (4.4), using the definition in [36]:

$$eb(s,t) = \frac{\ln(E(e^{sX(0,t)}))}{st} \tag{4.4}$$

$X(0,t)$ is the amount of incoming work during a duration of t. The (s,t) parameters are the so-called space and time parameters. Parameter s is calculated from an asymptotical exponential decrease assumption of the overflow probability (4.5). This is based on Large Deviations Theory (LDT) and large buffer assumption.

$$P(B < Q) = e^{-sB} \tag{4.5}$$

In (4.5), B is the buffer size and Q is the queue occupation random variable. The time parameter t is related to the time scales which are responsible for buffer overflow. It should be chosen small enough so that traffic is observed for buffer overflow analysis. Finally, the expectation in (4.4) is approximated by a time average, as suggested in [37].

The empirical evaluation of (4.4) is simulated and compared with analytical effective bandwidth of known Poisson and ON-OFF source types in [65].

Courcoubetis EB Allocation (CEB)

This method [14] is based on LDT and a large buffer assumption, similar to the DEB algorithm:

$$eb = m + \frac{ID\, s}{2B} \tag{4.6}$$

The parameters m, B, s and ID are the mean rate, buffer size, space parameter and index of dispersion of arrival process. The space parameter s is calculated using (4.5).

The ID estimation process has a computational complexity proportional to N^2.

$$ID = lim.n \rightarrow \infty * 1/n * E[(\Sigma_{i=1}^{n} X_i)^2] \tag{4.7}$$

where X_i is the arrival process.

Norros EB Allocation (NEB)

In [33], besides introducing modeling of real traffic by fractional Brownian motion (FBM), an effective bandwidth formula (4.8) for FBM is also given.

$$eb = m + (B^{H-1} * K(H) * \sqrt{-2 * a * m * P_{Loss}})^{1/H} \tag{4.8}$$

where $K(H) = H^H (1 - H)^{1-H}$ and m, H, $Ploss$ and a are the mean, Hurst parameter, buffer overflow probability, buffer size and coefficient of variation respectively. Parameter a, is approximated by the index of dispersion. In fact, this is a valid assumption only when the traffic is short range dependent.

The Hurst parameter can be set from a priori measurements. However, to react to unexpected traffic changes, a measurement-based on-line algorithm is favored. Difficulties of H estimation methods are analyzed in [43], where the comparison of several H estimation algorithms revealed that Abry-Veitch estimator (AV estimator) based on Wavelet theory is the best approach [66].

DRDMW (Improved Empirical EB Allocation)

This method in [59] is an improved version of empirical effective bandwidth methods. A unified phenomenological framework to estimate overflow probability of both long range dependence (LRD) and short-range dependence (SRD) is put forward by including the Hurst parameter in traditional analytical effective bandwidth methods.

$$P(B < Q) = e^{-s(C)B^{2-2H}} \tag{4.9}$$

Second, the difficulty of measuring the effective bandwidth of real-time traffic online by using direct estimator is alleviated by using an approach based on dual recursive algorithm with double moving windows (DRDMW), which is introduced in an empirical calculation of analytical effective bandwidth formula instead of using the direct estimator.

4.2.3 Frequency Domain Techniques

In this approach a cutoff frequency for the input traffic is obtained for traffic filtering. Then this filtered traffic is applied with a time domain prediction method for adaptive bandwidth allocations. In [13, 39], the authors have shown that, by considering the rate of an MPEG video trace as a signal, frequency domain analysis shows that the bandwidth requirement to guarantee no packet loss is essentially captured by the low frequency traffic filtered at a properly selected cutoff frequency. Basically if $\hat{x}(t)$ is the filtered traffic of the input $x(t)$ at ω_L, the minimum static bandwidth requirement is $max_t\hat{x}(t)$, which must be obtained only through an off line process. The cutoff frequency ω_L depended on the buffer size. At lower buffer size, ω_L will be high since the link capacity has to accommodate high frequency components, i.e., burstiness of the traffic. As the buffer size increases, the high frequency components can be absorbed by the buffer such that ω_L can be reduced and higher utilization can be obtained.

Since $\hat{x}(t)$ has low time variation and hence is predictable, it is suggested that [39] dynamic bandwidth allocation based on the online observation, and prediction of $\hat{x}(t)$ can be implemented. To implement the above approach in real time, RLS and Time Delay Neutral Network (TDNN) predictors are used to predict the low frequency variation, given that the traffic is passed through a low pass filter with an appropriate cutoff frequency [13]. The filtered input rate is predicted M steps in advance and the bandwidth is periodically adapted every M-step window. The allocated bandwidth is the maximum of these M predicted values multiplied by some factor. Although ω_L can be determined analytically for some special input processes such as MMPP or Gaussian , this is not the case in general and cut off frequency on an off-line trial and error basis.

4.3 Closed Loop Control

4.3.1 Queue Length Control

DRA algorithms which use the control based on fluid flow model of a queue are discussed in [49, 50, 51]. According to fluid flow model the ensemble average queue length in a single server queue can be represented by,

$$dQ(t)/dt = -C(t)G(Q(t)) + l(t) \tag{4.10}$$

where $C(t)$ is the service rate, $G(.)$ is a function obtained from matching the average utilization as a function of the queue length, and $l(t)$ is the ensemble average of packet arrival rate. This method has many disadvantages. The control is performed in continuous time and input arrival rate may not be measurable. Also, $G(.)$ is dependent on the queuing system and configuration.

The desired trajectory of $Q(t)$ is first specified in terms of $\lambda(t)$ and the aim is to compute $C(t)$ which minimizes the tracking error. Pistillides *et al* have worked on $M/M/1$ queue model under different contexts. In [49], a multilevel optimal control approach is used to improve global network performance and robustness. A proportional control based solution is derived in case of ATM ABR service to handle a finite buffer and capacity case assuming the bound on input arrival rate is known [50]. A feedback linearization and robust adaptive control is applied to derive the bandwidth control law to maintain a low average queue length in the context of a DiffServ Premium service class [51]. The main disadvantages of these methods is the control is performed in continuous time and $\lambda(t)$may not be measurable. Also the form of $G(.)$ is different for different queue configurations.

4.3.2 Packet Loss Control

These DRA algorithms which target packet loss rates of the order of 10^{-3}. They use either integrated control method (direct feedback) or the control method which attempts to achieve target packet loss rate indirectly by maintaining some other related performance measure.

Direct Feedback

In [53], the discrete time bandwidth adjustment at time instant t_k is given by,

$$C_{k+1} = C_k + G_k ln(P_k/\varepsilon) \tag{4.11}$$

where C_k is the service rate, $G_k$is the feedback gain, P_k is the measured packet loss rate over $[t_{k-1}, t_k]$, and ε is the target packet loss rate. The error feedback term ln (P_k/ε) governs how much C_k is to be adjusted: the amount of adjustment becomes smaller as P_k goes closer to ε. G_k basically amplifies the feedback and controls how quickly the algorithm converges. Rampal *et al* also propose a heuristic procedure to dynamically adjust G_k which makes it smaller as P_k goes closer to ε. In [22], the performance of the modified version of this algorithm for MPEG video traces and multi-hop scenario is discussed.

A similar DRA algorithm has been proposed by Hsu and Walrand [31]:

$$C_{k+1} = C_k + (G/k)(L_k - \varepsilon A_k) \tag{4.12}$$

where G is a positive constant, and L_k and εA_k are respectively, the number of lost packets and the expected number of losses during the time interval $[t_{k-1}, t_k]$. Even though no performance analysis if presented, the algorithm is proved to converge to the minimum bandwidth for Markov modulated fluid source.

Indirect Feedback

In [40], a target loss rate is translated into the average queue length as well as target utilization through a M/M/1/K queue assumption, and the control attempts to maintain the utilization by observing changes in the average queue length. The control action, invoked when measured average queue length deviates from its target beyond some thresholds, is used to adjust the service rate in such a way that the difference between the measured and target utilizations is minimized by multiplying current service rate by the ratio the utilizations. Since in the real networks the traffic is complex and does not follow M/M/1/K queue, the resulting loss is far for the desired target with fGn traffic input.

In [63], a fuzzy control is applied to maintain the queue length between a queue threshold pair in which depends on the target loss rate. Fuzzy control is a convenient approach to synthesize a non-linear controller whose control laws are heuristically derived through some key insights of the process under control. The results of this algorithm indicate that with the right target queue length, the target loss rate and the target queue length, the target loss rate can be achieved with high utilization and performance appears to be robust against the control time scale, although utilization decreases as the control time scale increases. Therefore the issues to be resolved for the fuzzy controller are the choice of control time scale and the mapping between target queue length and the target loss rate.

4.4 Hybrid Control

Z. Sahinoglu et al. have developed a hybrid model which combines the input information and the measured queue size [57]. The short term and long term fluctuations of the arrival rate have been separated into different frequency bands using wavelets and this information is used with the measured average queue length to compute the allocated bandwidth. The results show that the queue length can be maintained at some constant level with the wavelet-energy approach, but one has no control over the value at which the queue length is to be maintained.

4.5 Summary

In this chapter we have studied the adaptive resource allocation methods proposed by two different research communities: One group of researchers belong to traditional *Effective Bandwidth* (EB) algorithms and the other group belongs to *Adaptive Bandwidth Control* (ABC)methods . We provide a unified, critical and comparative analysis of on-line resource allocation algorithms of these two different classical approaches. We find that wavelet based algorithm proposed by [57] is not studied in detail and has enough potential to explore. So, we have carried out an experimental study on this algorithm in the next chapter.

Chapter 5

Dynamic Resource Allocation using Wavelets

5.1 Overview

Wavelet based traffic models for networks have been proposed by number of authors for different applications. In [66], Abry and Veitch have used wavelets to analyze the long-range dependent traffic and have proposed a semi-parametric estimator for the *Hurst* parameter. In [55], a multi-fractal wavelet model has been proposed for positive valued data with long range dependent correlations, using Haar wavelet transform. Wang *et al* have proposed an adaptive wavelet predictor for modelling VBR video traffic in [67]. They show that in comparison with the LMS predictor, the wavelet predictor reduces the prediction error by an average of 11 percent over the six half-an-hour-long empirical MPEG-1 traces. Xusheng *et al* use wavelet based models to provide a unified view to include most important understanding of the network traffic in [70]. In [45], Muralikrishna *et al* present the results on the modelling and synthesis of broadband traffic processes namely ethernet inter-arrival times using the VVGM (variable variance gaussian multiplier) multiplicative multifractal model. Authors have modelled variable bit rate (VBR) MPEG-4 traffic using wavelet based models in [4].

Z. Sahinoglu *et al* have developed a hybrid model which combines the input information and the measured queue size [57]. The short term and long term fluctuations of the arrival rate have been separated into different frequency bands using wavelets and this information is used with the measured average queue length to compute the allocated bandwidth. The technique used here can be applied to online traffic and works at very low time scales (packet level, one tenth of a second to seconds). We have improved the work presented in [57], by analyzing the performance of different ortho-normal wavelets. We have studied the effect of other wavelet parameters and have proposed an adaptive technique which exploits one of these parameters. We also develop an adaptive wavelet predictor based on this algorithm.

5.2 Introduction to Wavelets

The wavelet transform is a tool that divides the functions or data into different frequency components, and then studies each component with a resolution matched to its scale. The wavelet transform of a signal evolving in time depends on two variables: scale (or frequency) and time; wavelets provide a tool for time frequency localization. The wavelet transform in continuous time domain is defined as [15],

$$(T^{wav}f)(a,b) = |a|^{-1/2} \int f(t)\psi(t - b/a)dt \tag{5.1}$$

where, a is scaling variable and b is time variable. Discrete time version of the wavelet transform can be found by restricting a, b to only discrete values with m, n ranging all over Z, and fixing $a_0 > 1, b_0 > 0$ in following equation,

$$(T^{wav}f) = |a|^{-m/2} \int f(t)\psi(a_0^{-m}t - nb_0)dt \tag{5.2}$$

It is assumed that ψ satisfies $\int dt\psi(t) = 0$. Eq. 5.1 takes the inner products of f with a family of functions given by

$$\psi^{a,b}(s) = |a|^{-1/2}\psi(s - b/a) \tag{5.3}$$

The functions $\psi^{a,b}$ are called 'wavelets' and ψ is called as the 'mother wavelet'. Since $\psi^{a,b}$ have time-widths adapted to their frequency, the wavelet transform is better able to 'zoom in' on very short lived high frequency phenomena, such as transient signals.

5.2.1 Orthonormal Discrete Wavelet Transform

If we choose $a_0 = 2, b_0 = 1$,then there exist ψ,

$$\psi_{a,b}(x) = |2|^{-m/2}\psi(2^{-m}x - n) \tag{5.4}$$

constitute an orthonormal basis for $L^2(R)$, which can be used for multi resolution analysis (MRA). Eq. 5.4 consists of quadrature mirror filter (QMF) pair, ϕ and ψ. A quadrature mirror filter is a filter bank which splits an input signal into two bands which are sub-sampled by a factor of 2. The upper band has inverted frequencies, i.e., low frequencies are encoded as high frequencies and vice versa. All available wavelets can not be used for MRA. Some of the wavelets which form orthonormal basis which are used for analaysis are shown in Fig. 5.1. One of the simplest and oldest examples of a function ψ, which constitute an orthonormal basis for $L^2(R)$, is the Haar function,

$$\psi(x) = \begin{cases} 1 & \text{if } 0 \leq x < 1/2 \\ -1 & \text{if } 1/2 \leq x < 1 \\ 0 & \text{otherwise.} \end{cases}$$

$$\phi(x) = \begin{cases} 1 & \text{if } 0 \leq x < 1 \\ 0 & \text{otherwise..} \end{cases}$$

5.3 Multiresolution Decomposition

In this approach we decompose the time series traffic data into a number of frequency bands, every element of which has the traffic arrival rate information [57]. Here, we separate the low and high frequency components of the traffic arrival process. This gives us the contribution of each frequency band on the main traffic pattern. We use this information to predict the new traffic arrival rate.

Consider a vector,

$$X_k = [X(n - M + 1)X(n - M + 2)....X(n)] \tag{5.5}$$

at time n, where k is the time scale and M is an integer. Each element of vector $X(i)$ represents amount of traffic received in time slot i. Arrival rate information between any

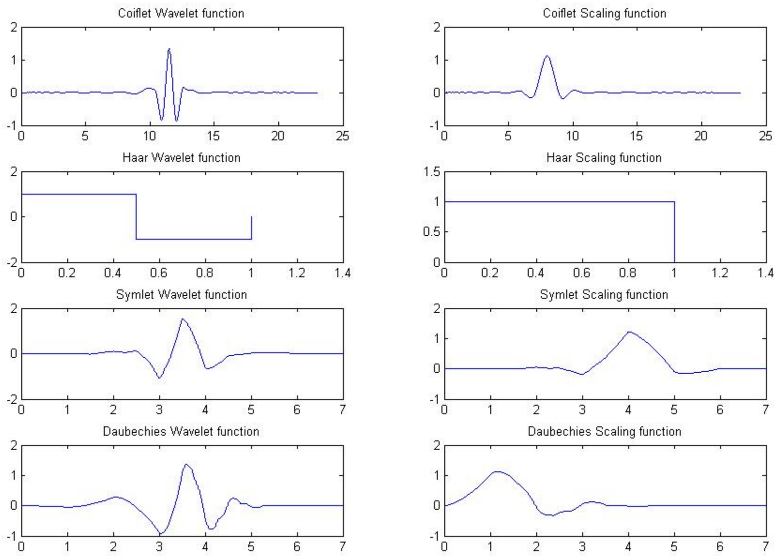

Figure 5.1: Wavelet and scaling functions of different wavelets

two consecutive states can be computed by their sum and difference. The difference would reveal the sharp changes in the arrival rate. We can represent the arrival rate vector consisting of M consecutive time slots when represented at time scale $k+1$,

$$X_{k+1} = 1/2*[X(n-M+1)+X(n-M+2)X(n-M+1)+X(n-M+2)....X(n-1)+X(n))] \tag{5.6}$$

$$Y_{k+1} = 1/2*[X(n-M+1)-X(n-M+2)X(n-M+1)-X(n-M+2)....X(n-1)-X(n))] \tag{5.7}$$

where Y_{k+1} is the difference vector. Equations 5.6 and 5.7 can be generalized for any time scale i as,

$$X_{i+1}(j) = 1/2 * [X_i(2j-1) + X_i(2j)] \tag{5.8}$$

$$Y_{i+1}(j) = 1/2 * [Y_i(2j-1) - Y_i(2j)] \tag{5.9}$$

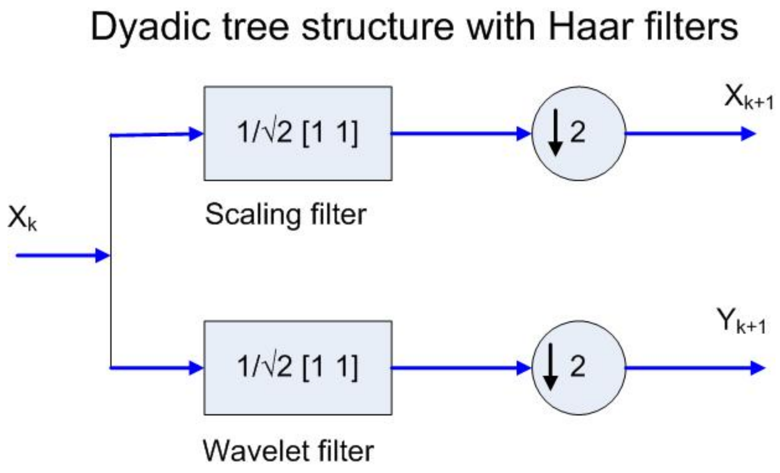

Figure 5.2: Haar wavelet decomposition by decimated filter banks

Since we are interested in the dynamic behavior of the traffic, we would like to look into the differences among neighboring vectors. Equations 5.8 and 5.9 can be implemented by the scaling and wavelet transform coefficients of Haar wavelet at a scale k. The difference would be just the value of the constant multiplier. The scaling and wavelet coefficients for Haar wavelet are as shown in Fig. 5.2.

We have for all values of i and j, $X_i(j) \geq 0$. Wavelet domain modelling of positive processes requires the constraint that a positive output is ensured [55]. This is true if $Y_i(j) \leq X_i(j)$ and the Haar wavelet satisfies this constraint. The proof can be very easily shown by modifying the above equations,

$$X_i(2j-1) = 1/\sqrt{2} * [X_{i+1}(j) + Y_{i+1}(2j)] \tag{5.10}$$

$$X_i(2j) = 1/\sqrt{2} * [X_{i+1}(j) - Y_{i+1}(j)] \tag{5.11}$$

The output of the scaling filter in the dyadic tree represents the coarse component X_{i+1} of the original traffic data X_i, and the output of the wavelet filter gives the details Y_{i+1}.

5.4 Multiresolution DRA Algorithm

The wavelet transform operation can be expressed as,

$$\hat{W} = R\hat{X} \tag{5.12}$$

where $\hat{W}$ is the wavelet transform vector, R is the $M * M$ wavelet transform matrix, and $\hat{X}$ is the input data vector of length M. The energy content in each sub-band frequency or time scale k can be computed as,

$$E_k = \sum_{n=2^{k-1}+1}^{2^k} |\hat{W}(n)|^2 \tag{5.13}$$

where k is the scale index and $\hat{W} = [w1w2w3....w8]$for M=8. The energy at different scales can be found by applying above equation to $\hat{W}$. For this case eight sample moving data is created by using eight unit delays. This data is passed through three stages of dyadic filters since $M = 2^3$ and we get four detail coefficients. A three level wavelet decomposition is shown in Fig. 5.3. The wavelet filter is often referred as high pass filter (HPF) and scaling filter as low pass filter (LPF). The window size M decides the number of subband filters used in decomposition. Energy in each band represents the traffic volume within that frequency band. The bandwidth allocated in the next time slot is computed as shown below. Here $\hat{W}_k(1)$ is used as the average allocation and the energy contributed from different frequency bands are added to this average.

$$C_{i+1} = \hat{W}_k(1) + \surd\sum_{i=1}^{K} |\hat{E}_n(i)| \tag{5.14}$$

5.5 Simulation Experiments and Results

We have used the bellcore trace for our experimentation and five set of experiments have been conducted to highlight the effect of different parameters on the predictor output. For all the experiments we have used the measurement time scale as 0.1 sec and adaptation time scale is based on window size. We have used a window size of 8 as well as 16 for the different simulation experiments.

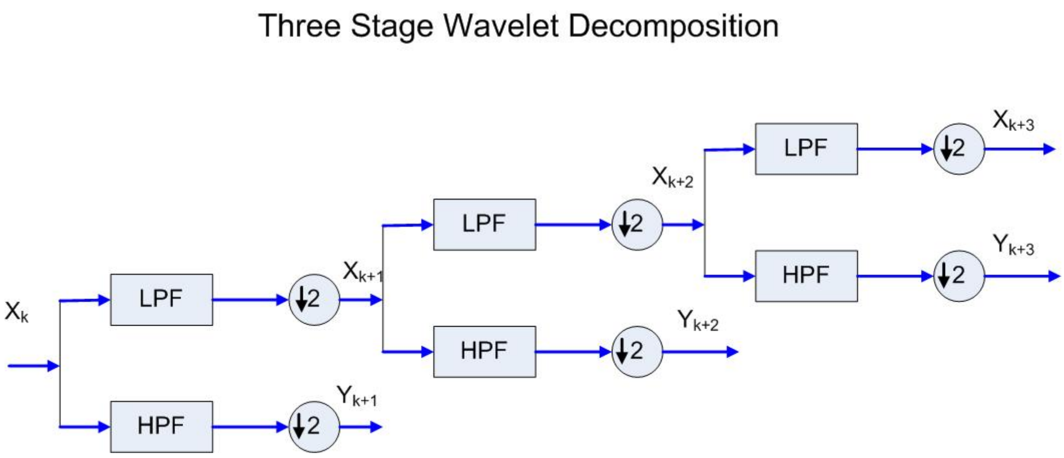

Figure 5.3: Multilevel wavelet decomposition by decimated filter banks

- We compare the performance of different predictors with respect to Mean Square Bandwidth Allocation Error (MSBAE in bytes). It is the sum of the squares of the difference between each bandwidth demand and the allocation.
- We compute the improvement due to the wavelet predictor as,

  Improvement in allocation = 1- (MSBAE with Wavelet predictor / MSBAE with peak allocation)
- We also compare the queueing performance in terms of mean and maximum queue sizes (in bytes).

To generalize the performance of the predictor we have tested it against following traces:

- Bellcore trace
- UNC trace (Chapel Hill, NC)
- Pareto on/off traffic
- Exponential on/off traffic

Comparison of the effect of wavelet predictor on these different traces is shown in Table. 5.1.

Table 5.1: Effect of predictor on different traces

Type of trace	Percentage Improvement (MSBAE)
Exponential on/off traffic	60.88
Pareto on/off traffic	67.31
UNC Trace	69.11
Bellcore Trace	73.95

5.5.1 Comparison of Wavelet Predictor with other Predictors

In this experiment we have compared the performance of Average predictor, Peak allocator and Gaussian predictor with that of the Haar predictor. Also we have used 2nd level decomposition results for this experiment. Results are shown in Table. 5.2 and Table. 5.3. Fig. 5.4 shows the predictor performance and Fig. 5.5 compares the queue performance. From the results it follows that the MSBAE due to Haar predictor is almost half of that of the Gaussian predictor.

Table 5.2: Comparison of wavelet predictor with other predictors for window size 8

Type of Predictor	MSBAE	Improvement	Average Queue	Maximum Queue
Average Allocation	2273	79.67	42421	116790
Peak Allocation	11185	00.00	0	0
Gaussian	6865	38.62	8	3240
Haar	3436	69.27	125	7806

Table 5.3: Comparison of wavelet predictor with other predictors for window size 16

Type of Predictor	MSBAE	Improvement	Average Queue	Maximum Queue
Average Allocation	2297	79.46	32825	115180
Peak Allocation	11185	00.00	0	0
Gaussian	7373	34.07	3	2052
Haar	3614	67.69	315	9462

5.5.2 Comparison of the Performances of different Wavelets

In this section We have compared the performances of Haar, Daubechies (db4), Symlet (sym4) and Coiflet (coif4) wavelets. From Table. 5.4 it follows that all the other wavelets do far better than Haar with respect to MSBAE but Haar shows a very good queue

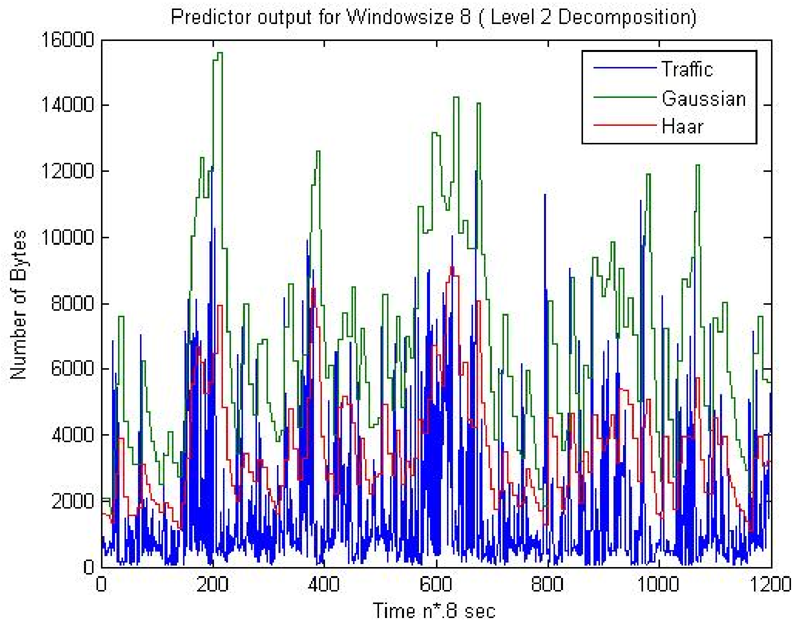

Figure 5.4: Comparison of MSBAE for Haar wavelet predictor with Guassian predictor for window size 8

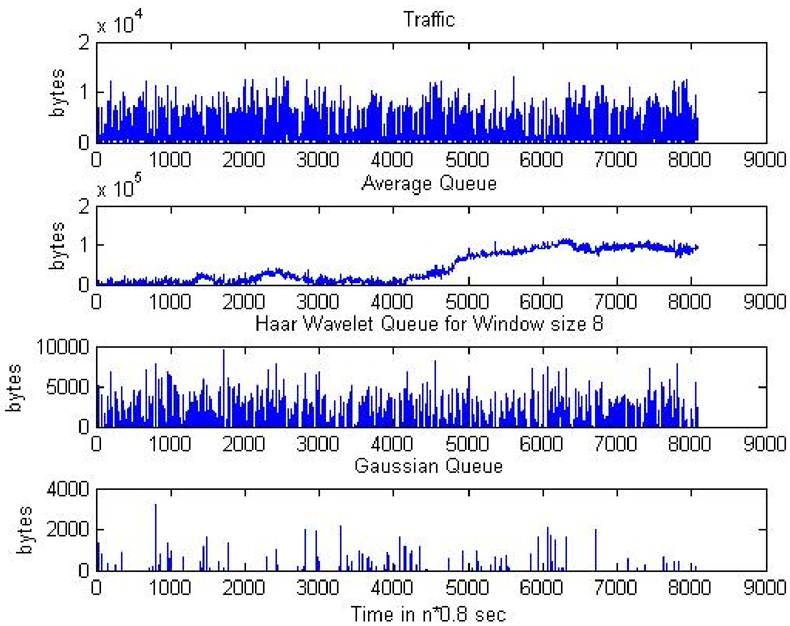

Figure 5.5: Queue status for different predictors for window size 8

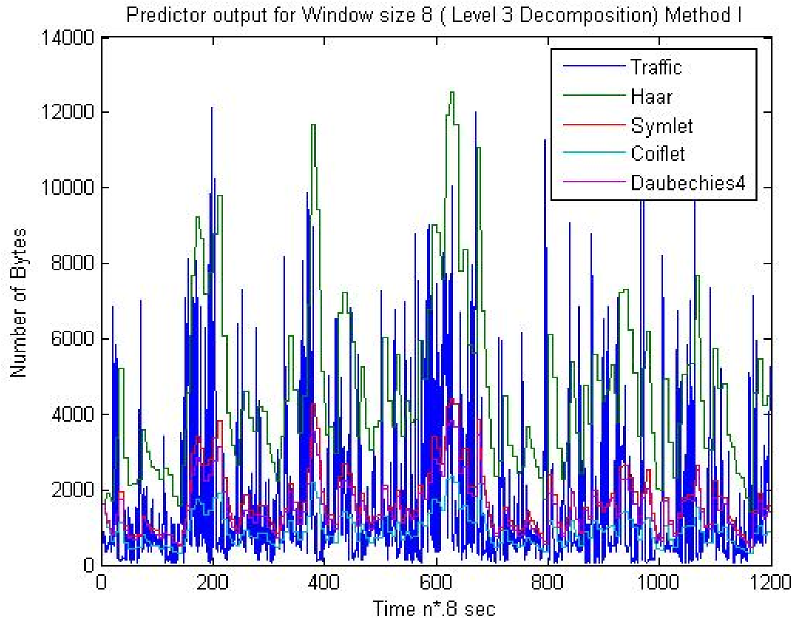

Figure 5.6: Comparison of all the wavelet predictors

performance. Fig. 5.6 shows the performance of the different wavelets.

Table 5.4: Comparison of all the wavelet predictors

Type of Predictor	MSBAE	Improvement	Average Queue	Maximum Queue
Haar	3436	69.27	125	7806
Daubechies	2263	79.77	92065	225160
Symlet	2284	79.57	36528	105180
Coiflet	2370	78.81	386750	838460

5.5.3 Effect of the Decomposition Level on Predictor Performance

In this experiment we have compared the performance of Haar with respect to number of decomposition levels. Since we know that number of decomposition levels is dependent on the window size, we have used both window sizes of 8 and 16. This allows us to study upto 4 decomposition levels in case of window size of 16. From Table. 6.1 and Table. 5.6 it follows that as we go for higher decomposition levels queue size improves but the error increases. Fig. 5.7 shows the performance of the different decomposition levels in case of Haar wavelet.

Table 5.5: Effect of decomposition levels for window size 8

Decomposition level	MSBAE	Improvement	Average Queue	Maximum Queue
I	2791	75.04	775	13755
II	3436	69.27	327	9437
III	4568	59.15	125	7806

Table 5.6: Effect of of decomposition levels for window size 16

Decomposition level	MSBAE	Improvement	Average Queue	Maximum Queue
I	2912	73.95	770	17355
II	3614	67.69	315	9462
III	4817	56.93	115	7970
IV	6693	40.15	31	5343

5.5.4 Effect of Adaptation Window on Predictor Performance

Here we show the effect of adaptation time scale on the predictor performance. From Table. 5.7 it follows that there is no effect of increase in the window size on the predictor performance. It may be because we the change in the window size may not be significant to show considerable difference. We may have to use higher time scales and then find the effect of window size. Fig. 5.8 and Fig. 5.9 shows comparison of Haar wavelet performance under different window sizes.

Table 5.7: Effect of window size on Haar predictor performance

Window	Level	MSBAE	Improvement	Average Queue	Maximum Queue
8	III	4568	59.15	125	7806
16	III	4817	56.93	115	7970
16	IV	6693	40.15	34	5343

5.5.5 Effect of Number of Filter Coefficients

In this experiment we increase the filter coefficients of Daubechies filter from 2 to 8. From Table. 5.8 it follows that as we increase the number of coefficients there is an

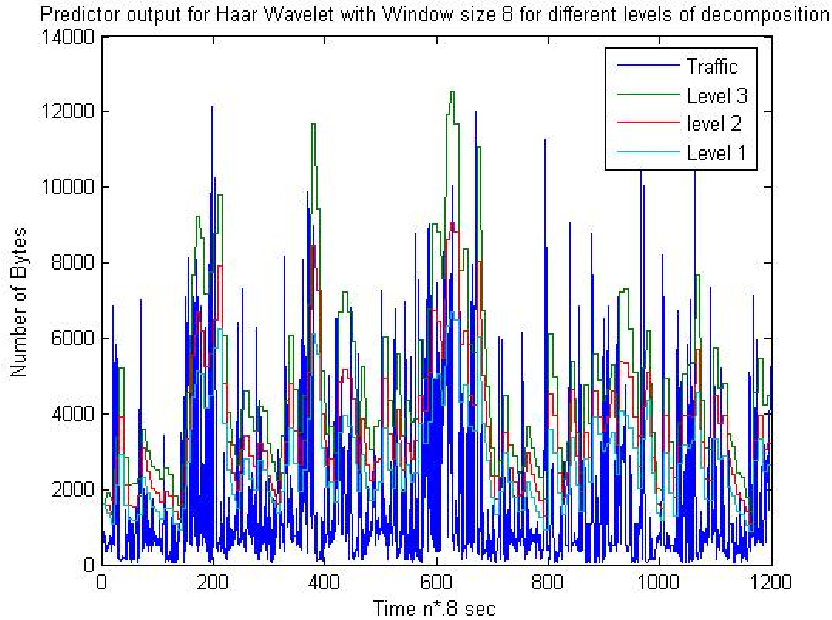

Figure 5.7: Effect of decomposition levels on Haar predictor performance

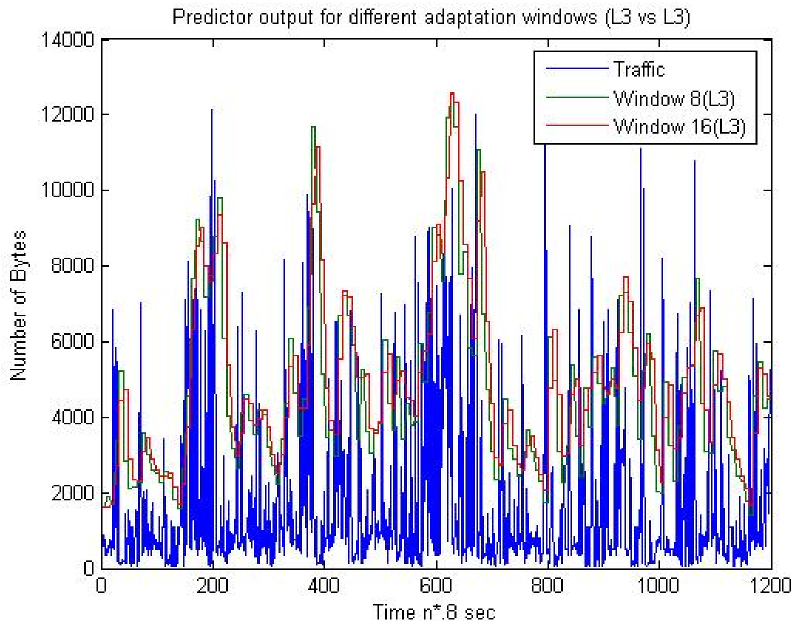

Figure 5.8: Effect of window size on Haar predictor performance

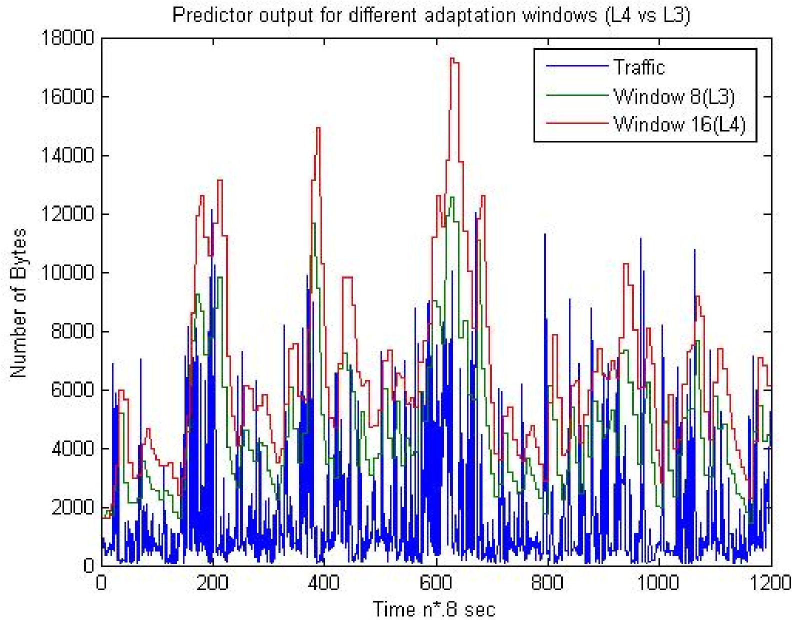

Figure 5.9: Effect of window size on Haar predictor performance

improvement in MSBAE but there is a degradation in queue performance. Fig. 5.10 shows the effect of filter coefficient on the predictor performance.

Table 5.8: Effect of number of filter coefficients

Filter co-eff.	MSBAE	Improvement	Average Queue	Maximum Queue
2 (db2)	3436	69.27	125	7806
4 (db4)	2263	79.77	92065	225160
8 (db8)	2306	79.37	278360	618160

5.6 Adaptive Wavelet Predictor(AWP)

From the above experiments it can be observed that, the performance of the wavelet predictor can be tuned by the number of decomposition levels used. Higher the number of levels used, faster the algorithm can catch up a burst and remove the excess packets received in the previous time slot. There fore we can use number of levels as a control parameter to improve the robustness of the algorithm. The basic block diagram for

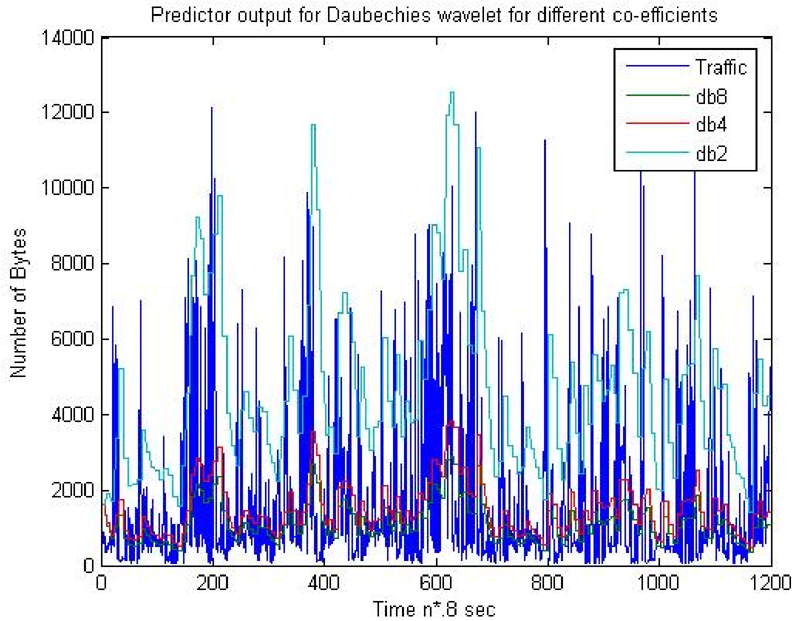

Figure 5.10: Effect of number of filter coefficients on predictor performance

the Adaptive Wavelet Predictor is shown in Fig. 5.11.

5.7 Summary

In this chapter we have investigated the role of *wavelet* as a traffic predictor and perform extensive experimental analysis in this regard. We show that wavelets can provide a fast, accurate and robust online traffic predictor. We also propose an adaptive wavelet predictor technique based on the number of decomposition levels which increases the robustness of the predictor. In the next chapter we start discussing the network level problems of the self-sizing framework.

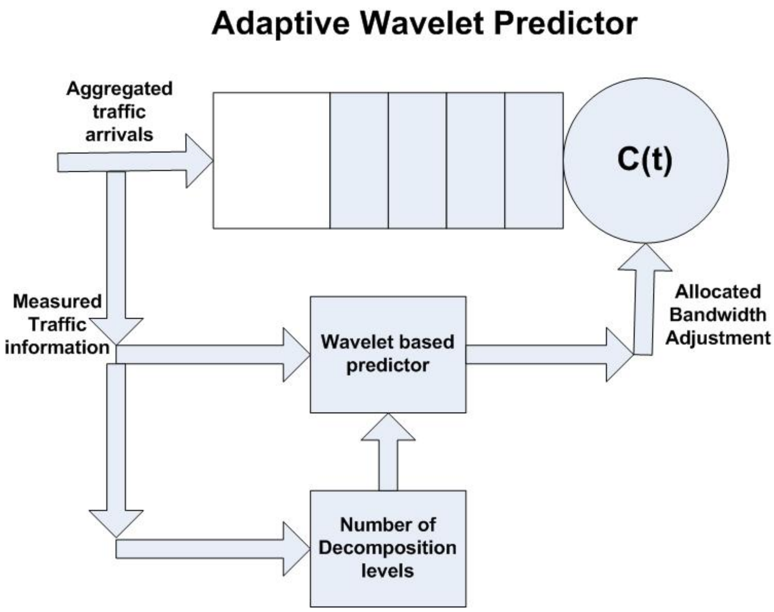

Figure 5.11: Basic block diagram of Adaptive Wavelet Predictor

Chapter 6

Self-Sizing Networks: Local vs. Global Control

6.1 Overview

The concept of "self-sizing" was introduced in [71, 62] by industrial researchers from NTT and Nortel respectively. In [71] J. Yan proposes that self-sizing can be achieved by virtual band partitioning of network capacity, where each band corresponds to a traffic class. A multiservice network design process consists of two steps: determining the aggregate network topology and capacity, and the partitioning of the capacity among the elastic network bands. Since the aggregate network design is a slow process based on a long term forecast of the overall traffic, is normally done off line. However the network bands must be modified frequently based on fresh traffic data and network state information.

In [62] S. Shioda *et al* defines *self-sizing* as an autonomous adjustment of Virtual Path (VP) bandwidths based on traffic conditions observed in real time. By using the bandwidth demand concept they propose a VP-bandwidth sizing procedure in which real estimates of VP bandwidth demand and successive VP bandwidth allocation are jointly utilized. They develop an operating system to provide self-sizing functions, the experimental use of first prototype on an ATM testbed was carried out in 1996. They state that *self-sizing*

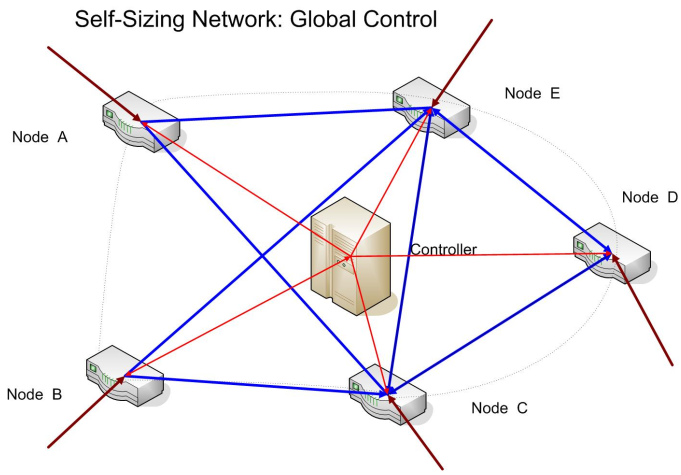

Figure 6.1: Globally controlled Self-Sizing network

in real ATM networks could not be implemented since the data in terms of number of cells was not collected at lower time scales.

The authors in [29] propose an optimization model to partition bandwidth among bands to minimize total system cost under capacity constraints while the QoS at call level and cell level are guaranteed. They also develop a fast algorithm based on simulated annealing for frequent band partitioning. In [60] the authors introduce a general measurement-based self-sizing framework for resource allocation in label-switched networks. They also propose a hierarchical approach to improve adaptation performance for large networks and they show that performance improves as the size of the network increases.

6.2 Local vs. Global Control

Functional block diagrams for local and global control techniques are shown in Fig. 6.1 and Fig. 6.2. A typical globally controlled network consists of a central controller which gets network state information from each participating node of the network [29]. Another form of globally controlled network also exists where resources are allocated on end to end basis. On the other hand, in a locally controlled network, each node is independent and resource management done by itself.

Merits and demerits of local and global control methods are given in the Table 6.1.

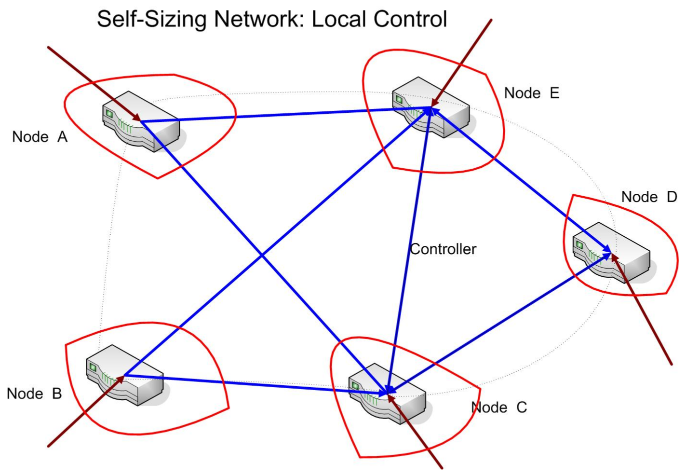

Figure 6.2: Locally controlled Self-Sizing network

Table 6.1: Local vs. Global

Feature	Local	Global
scalability	scalable	not scalable
time scales	packet level	connection level
complexity	less complex	more complex
signaling	not required	required
optimality	near optimal	optimal

We follow that local control technique is less complex and scalable. The self-sizing framework in globally controlled networks needs a signaling architecture to transmit the network state information to the controller. As the number of nodes/hops increase, the time required to communicate the network state information to the controller and the allocation information to all the nodes increases. But we do not need such a mechanism for the locally controlled network as each individual node acts as a controller, and relies on its own status information (and traffic observations) alone.

Another important aspect of the self-sizing architecture is the computation time required to find the optimal allocation. This computation time depends on the efficiency of the algorithm, and also on the complexity of the resource allocation problem. The complexity of the problem in case of global control is of the order of $o(n^2)$, as compared to $o(n)$, in case of local control.

6.3 Simulation Methodology

The topology of the network used for simulation consists of five nodes and seven links supporting two types of services (like video and voice, which are also referred to as virtual bands) and 20 SD pairs (traffic from each node to four other nodes) as shown in Fig. 6.3. We assume that links are duplex and of equal capacity. For the sake of convenience, we consider a single duplex link as two simplex links carrying traffic in the opposite direction.

We assume that the sum of EBs for all the traffic is always less than the sum of the capacities of all the links connected to that node. It is quite realistic to make this assumption for the following reason. This assumption is well supported by the fact that most of the backbones are over-provisioned [8]. As we said earlier, the self-sizing framework classifies the traffic broadly into three classes viz. video, voice and data. All the delay sensitive applications which need a QoS guarantee belong to either voice or video class. The remaining traffic which belong to data class does not need any QoS guarantee and we give lowest priority to this traffic. So, we measure only voice and video traffic and allocate bandwidth according to EB which ensures quality of service.

To establish the effectiveness of the local optimization technique, we consider this network in three different scenarios: In the "Global Control" scenario, the network has a central controller running an optimization algorithm on the network state information. In the second scenario, called "Local Control", the optimization algorithm is run on each node with only the local information. In the "Limited Control" (neighborhood control) scenario some of the nodes have knowledge of resource availability at their neighboring nodes. In this scenario, nodes A, C and D have the information of nodes B and E.

Performance analysis is done first by comparing the bandwidth requirement for instantaneous traffic with and without traffic measurement. This gives the measure of percentage bandwidth gain over the whole network by allocating resources according to their EB. The number of sources is set to 8.

Then, we compare the three different scenarios on the basis of congestion reduction. The optimization algorithm tries to balance the load over all the links and achieve uniform link utilization. Surely, "Global Control" is expected to work best, but we would like to show how effective the other two scenarios are compared to the global technique. To demonstrate the effectiveness of time scales on performance, we ran the simulations for *different adaptation times*. There is also a significant effect of *measurement times* on the

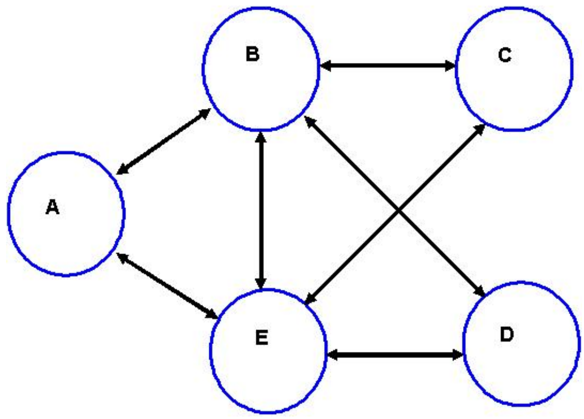

Figure 6.3: Topology of the network used for simulation.

performance of the network. To generalize the results we apply our technique using both dynamic On/Off traffic as well as self-similar traffic. Also, the effectiveness of both EB algorithms is evident from the comparison of percentage bandwidth savings.

6.3.1 Traffic Characteristics

We have evaluated the performance of the network for the following types of traffic sources:

1. On/Off Sources (Dynamic): The source activity ratio R of these sources varies randomly over time [58] as

$$R = \frac{OffTime}{OffTime + OnTime} \quad (6.1)$$

2. Self Similar Sources: We have used the Sup-FRP traffic model [56] with

Hurst Parameter = 0.75

Number of Sources = 8

We have simulated the self-sizing network scenarios using the OPNET network simulation platform. OPNET software embeds expert operational knowledge of network devices, network protocols, applications, and servers. This intelligence enables users in network operations, engineering, planning, and application development to be far more effective in optimizing performance and availability of their networks and applications. OPNET modeler allows us to modify its source code and develop our own process modules. The OPNET

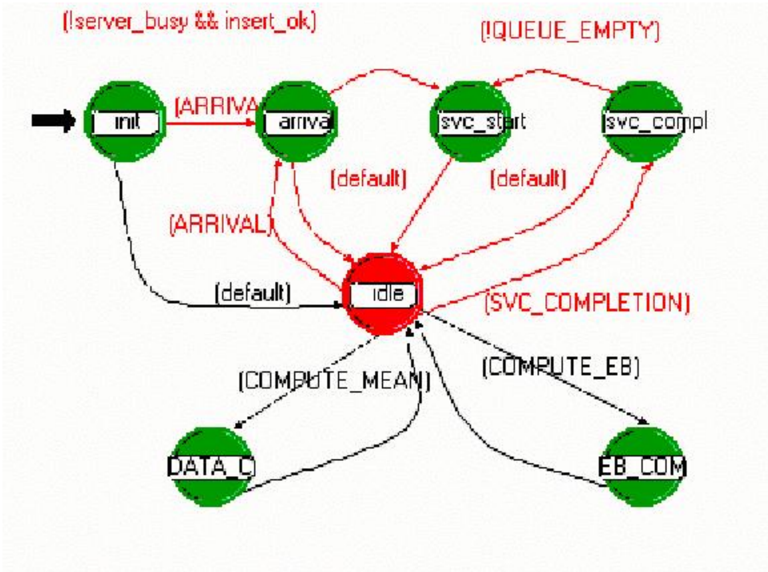

Figure 6.4: OPNET process model for self-sizing queue

process model for the self-sizing queue is given in Fig. 6.4. It measures the amount of traffic arrived in the measurement time slot and computes the effective bandwidth for the respective adaptation time window.

6.3.2 Definitions

- Measurement Time: Time slot during which traffic is measured.
- Adaptation Time: Time slot after which resources are reallocated.
- One (1) BWUnit = 100kbits.
- B: Types of services $(video, voice)$.
- P: Number of SD pairs (20).
- J: Number of links (14).
- CAP_j: Total available capacity for link j in BWUnits.
- X_{pbr}: Effective bandwidth assigned for SD pair p on link j.

- R_{pb}: Set of two alternative routes for service type b for the SD pair p.
- X_{pbr}=1 if the service b at for SD pair p goes through r from R_{pb}
- BW_j: Total allocated capacity for link j in BWUnits
- T_{av}: Average link allocation

The objective function of the optimization problem is to minimize the sum of the differences between the average link assignment and the individual link assignments. The optimization problem can be defined as,

$$min \sum_{j \in J} |T_{av} - BW_j| \tag{6.2}$$

s.t.

$$\sum_{b \in B} C_{bj} < CAP_j \qquad j \in J \tag{6.3}$$

$$\sum_{r \in R_{pb}} X_{pbr} = 1 \qquad p \in P, b \in B \tag{6.4}$$

$$X_{pbr} \in \{0, 1\} \tag{6.5}$$

where

$$T_{av} = (\sum_{j \in J} BW_j)/J \tag{6.6}$$

Table 6.2: Different cases for which traffic was measured

Traffic Type	Self-similar and On/Off
EB Algorithm	Courcoubetis and Gaussian
Measurement Times	1ms and 10ms
Adaptation Times	1ms, 10ms, 100ms, 1s and 10s

We assume that sum of all effective bandwidths allocated is always less than the sum of link capacities. Simulations were run for 300 seconds and the results have been plotted to show the bandwidth savings and percentage improvement in link utilization. Bandwidth savings is the amount of bandwidth we save when we compare the EB allocation with that of the peak allocation. Improvement in link utilization is reduction in the over all difference in the link utilizations due to optimized path assignment as compared with the shortest path assignment.

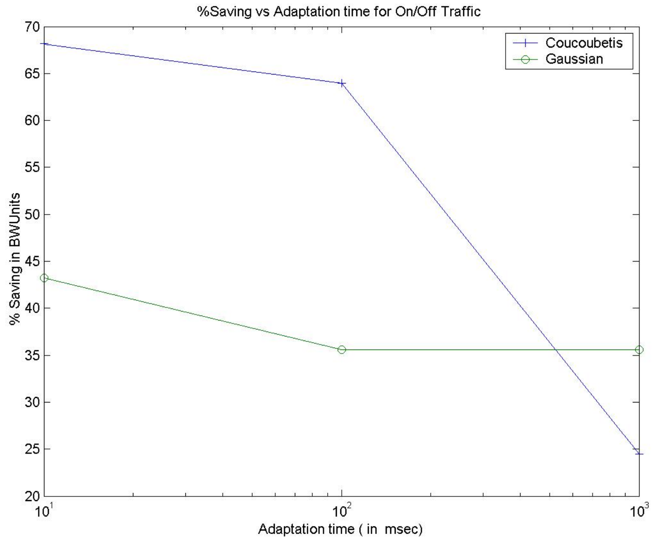

Figure 6.5: Percentage savings in bandwidth vs. adaptation time for On/Off traffic and measurement time equal to 10ms.

Fig 6.5 and 6.6 show the plot of percentage bandwidth savings against adaptation times for On/Off traffic and self-similar traffic for both the Gaussian and Courcoubetis methods.

Fig 6.7 and 6.8 compare percentage improvement in link utilization for the 'Global Scenario', the 'Limited Scenario', and the 'Local Scenario', for both On/Off and self-similar traffic, and for both of the EB estimation algorithms.

6.4 Results and Discussion

One of the objectives of the self-sizing network is to allocate the bandwidth as efficiently as possible. It follows from Fig 6.6 that with self-sizing we can save up to 32% of bandwidth with self-similar traffic. It goes up to 68% in case of exponential on/off traffic. It also follows from the above plots Fig. 6.5 and 6.6, that the Courcoubetis algorithm is more efficient than Gaussian method.

The second and important factor behind this study was to compare the performance of *local control* with that of *global control*. From Fig 6.7 it follows that with *local control* we can achieve considerable improvement (64%) as compared to that of *global control*

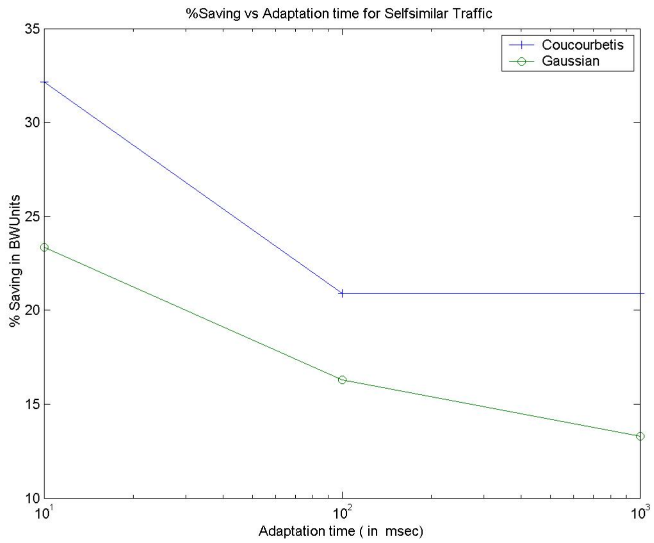

Figure 6.6: Percentage savings in bandwidth vs. adaptation time for seif-similar traffic and measurement time equal to 10ms.

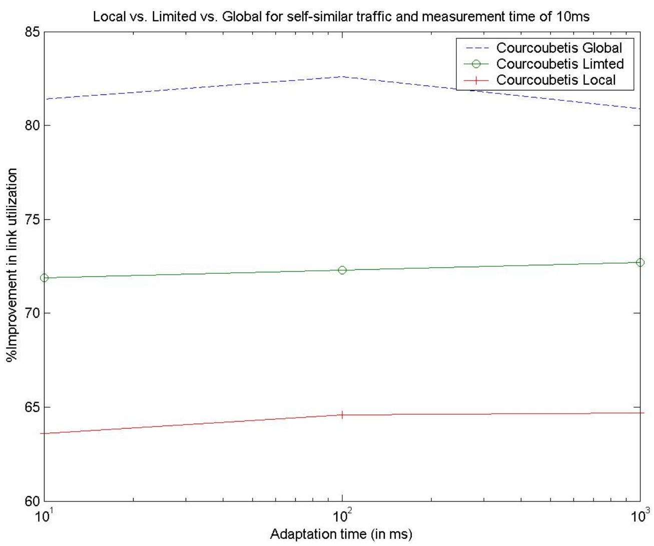

Figure 6.7: Percentage improvement in link utilization with local, global and limited optimization vs. adaptation time for self-similar traffic and measurement time equal to 10ms.

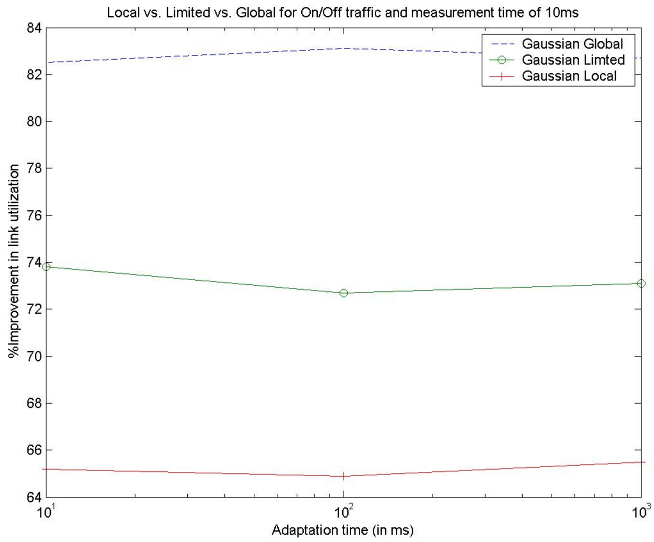

Figure 6.8: Percentage improvement in link utilization with local, global and limited optimization vs. adaptation time for On/Off traffic and measurement time equal to 10ms.

(82%).

We also found that, by providing some of the nodes with their neighbors resource availability, it is possible to improve the results by 73%. This gain comes at the expense of increasing complexity, but it is interesting to compare the overhead costs with the profit. If we devise an economical mechanism to transmit the node status to neighbors, we can achieve relatively larger improvement.

6.5 Summary

In this chapter we have compared the local and global control self-sizing models. Even though the global control provides better gains, it is not a scalable model. On the other hand local control provides lesser gains but it is a scalable solution. Near optimal solutions can be found if the local control is aided with *knowledge* of the network state. In the next chapter we study the local control mechanisms in detail by varying the knowledge of the nodes.

Chapter 7

Self-Sizing: Local vs. Global Knowledge

7.1 Overview

In the last section we have seen that a local control technique can be advantageous in complex networks which need a scalable solution. The biggest limitation of distributed control is that we can not achieve the optimality. We define optimality to be a function of throughput and QoS parameters such as packet loss, delay and jitter. But near optimal solutions can be found if the local control is aided with *knowledge* of the network state. One of the key constituents of self-sizing mechanism is prediction of the incoming traffic. If a node has the knowledge of the traffic/allocation status at the neighboring nodes then it can make a better prediction. This would help in allocating the resources judicially and increasing the optimality. But we argue that having more information may introduce the timescale problems faced in the global control technique and also may end up in making inaccurate predictions.

Fig. 7.1 shows the projected results for a locally controlled network as we increase the knowledge from 0 hops (local) to 5 hops. In this graph we consider the performance of the global control case to be 100 percent optimal. We predict that one hop knowledge case

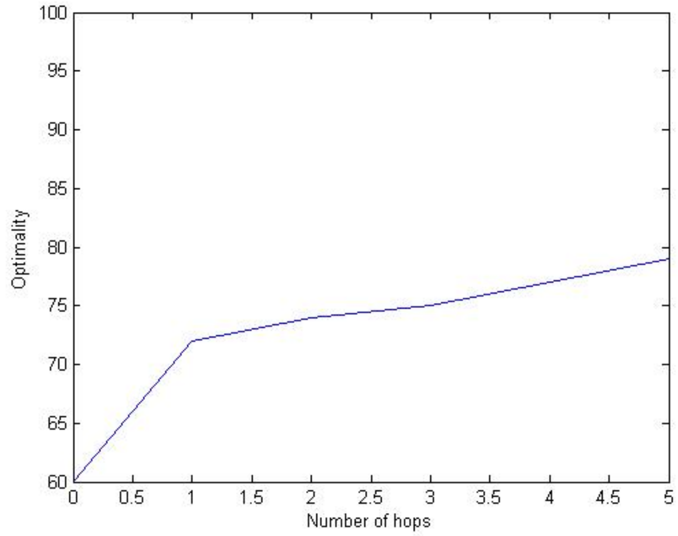

Figure 7.1: Effect of knowledge on optimality in locally controlled self-sizing network

is more practical considering the time constraints and would provide the best results. The local and neighborhood control (one hop) results are taken from our earlier experiments presented in the previous section.

7.2 A Model for Neighborhood Control

To demonstrate the importance of the neighborhood control we conduct number of experiments using the two node model shown in Fig. 7.2. In this model, a non self-sizing node feeds a queue of a self-sizing node. In the neighborhood control case the non self-sizing node provides its resource allocation information to the self-sizing node. The results of the first set of experiments are presented in Tables. 7.1 through 7.4. The queue status and the capacity allocations are shown in Fig. 7.3 and Fig. 7.4. Also the performance of neighborhood control can be visualized from Fig. 7.5 and Fig. 7.6. We find that neighborhood control method has lesser gain but better queue status as compared to local control method. This in turn suggests that neighborhood control provides better QoS performance than that of local control method. We have used the same performance measures which have been defined in Chapter 5 and for the sake of convenience the definitions are repeated below:

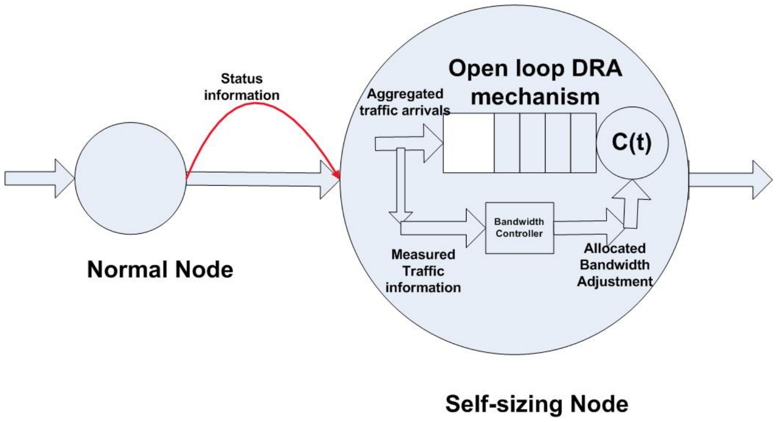

Figure 7.2: Neighborhood Control model

- We compare the performance of different predictors with respect to Mean Square Bandwidth Allocation Error (MSBAE in bytes). It is the sum of the squares of the difference between each bandwidth demand and the allocation.
- We compute the improvement due to the wavelet predictor as,

  Improvement in allocation = 1- (MSBAE with Wavelet predictor / MSBAE with peak allocation)
- We also compare the queueing performance in terms of mean and maximum queue sizes (in bytes).

Table 7.1: Local vs. Neighborhood self-sizing for Bellcore traffic

Type of control	MSBAE	Improvement	Maximum Queue
Local	2912.80	73.95	17355
Neighborhood	6613.80	40.86	5286.80

Table 7.2: Local vs. Neighborhood self-sizing for UNC trace

Type of control	MSBAE	Improvement	Maximum Queue
Local	40565	60.41	51200
Neighborhood	70629	31.42	25600

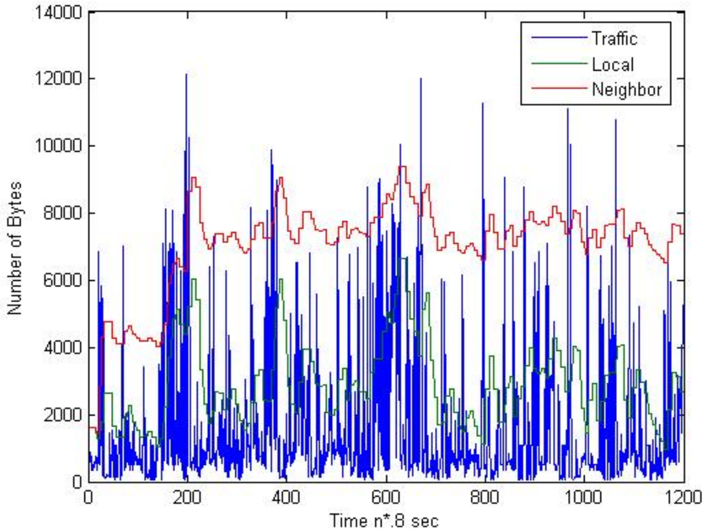

Figure 7.3: Neighborhood Control model: Capacity Allocation

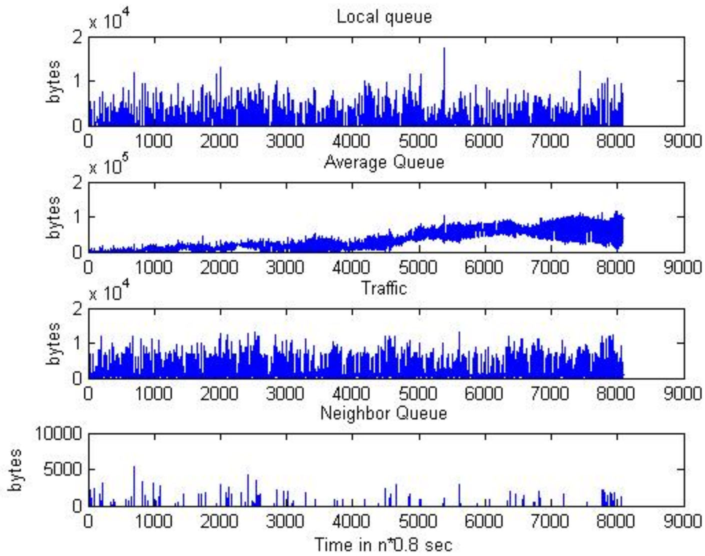

Figure 7.4: Neighborhood Control model: Queue Status

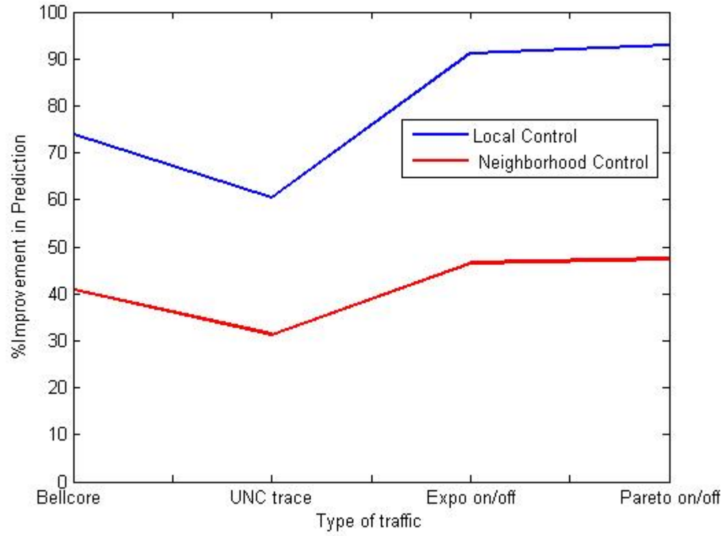

Figure 7.5: Local vs. Neighborhood improvement for different types of traffic

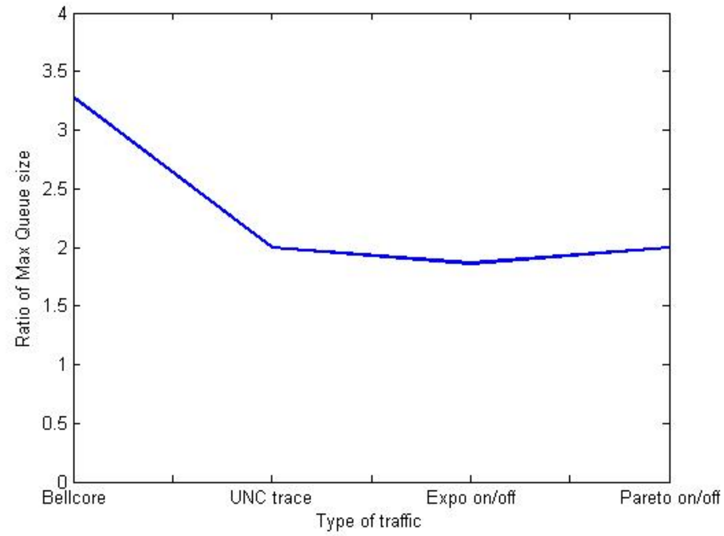

Figure 7.6: Ratio of Local to Neighborhood maximum queue size for different types of traffic

Table 7.3: Local vs. Neighborhood self-sizing for Exponential on/off traffic

Type of control	MSBAE	Improvement	Maximum Queue
Local	3155	91.21	17355
Neighborhood	19202	46.57	13303

Table 7.4: Local vs. Neighborhood self-sizing for Pareto on/off traffic

Type of control	MSBAE	Improvement	Maximum Queue
Local	3454	92.89	18621
Neighborhood	25498	47.51	9310

7.3 Applications of Local Self-Sizing

Local self-sizing mechanism provides a node the ability to predict the incoming traffic, allocate the resources efficiently and also provides the QoS at packet level. But it is a costly affair as we need to make the node intelligent. This needs lot of computing and storage resources, so it is not beneficial to implement self-sizing at all the nodes in the network. We have seen many discussions over the bottleneck or hotpot formation in the network. On that basis we advise that it would be wise to make these bottleneck nodes to run the self-sizing algorithm. We propose two possible implementations based on specific applications:

Static self-sizing

In this framework once initialized, a node always works as a self-sizing node. This can be implemented only if we can locate the bottleneck or hotspot node before hand and the traffic conditions are going to be same for quite long time.

Dynamic self-sizing

Here the self-sizing mechanism can be made available in all the nodes of the network but it should be enabled only when it is needed. Such a implementation is advantageous when we expect lot of traffic variations at all nodes and all nodes have the probability of becoming bottleneck node.

7.4 Summary

In this chapter we have proposed the *neighborhood control* model in which a self-sizing node has the network state information of its neighbors. We show that neighborhood control provides better QoS performance than the local control. We have also discussed the possible implementation schemes of local self-sizing. In the next chapter we provide summary of our contributions and discuss some of the open issues.

Chapter 8

Conclusions

8.1 Summary of Dissertation

The contributions of this dissertation are threefold. First, we identify the necessity and requirement of a new QoS framework for the Internet to support the stringent requirements of interactive traffic. Second, we introduce a novel wavelet based adaptive resource allocation method for such a distributed network scenario. Third, we propose a self-sizing architecture to locally controlled networks and provide a scalable neighborhood control solution.

- We conduct a systematic and detailed study of the evolution of the Internet from providing a single best-effort service to providing multiple types of services with Quality-of-Service (QoS) guarantee in terms of packet loss or delay. We justify the need for such an adaptive, intelligent, and scalable QoS framework for the Internet.

- We provide a unified, critical and comparative analysis of online resource allocation algorithms of two different classical approaches: *Effective Bandwidth* and *Adaptive Bandwidth Control*

- We investigate the role of *wavelet* as a traffic predictor and perform extensive experimental analysis in this regard. We show that wavelet can provide fast, accurate and robust online traffic predictor. We then propose an adaptive wavelet predictor technique based on the number of decomposition levels which increases the robustness of the predictor.

- We propose a self-sizing framework for the locally controlled networks. Our results from the Local vs. Global control experiment show that local self-sizing network can provide absolute QoS to the network traffic and also achieve considerable bandwidth gain.

- We further define a novel neighborhood control self-sizing model in which each node has the knowledge of their neighbors resource availability. The results suggest that this framework is more efficient and stable than the locally controlled self-sizing architecture.

8.2 Open Issues

8.2.1 Admission Control

Since the allocated bandwidth dynamically changes over time, admission control becomes a problem as we do not know when the new aggregates are going to join the queue. A situation may arise when required bandwidth computed from the DRA algorithm can not be allocated due to the admission of too much traffic. we can deal with this problem in two possible ways.

- We need to discover the impact of allowing new aggregates with respect to bandwidth violation.

- We need to investigate the relationship between the degree of QoS degradation and the bandwidth violation. Then we can define the tolerance limit for which bandwidth violation can be allowed.

We can also combine the above methods to find a solution for the admission control problem. This problem can be tackled using the measurement based admission control with aggregate traffic envelopes as given in [52].

8.2.2 Adaptation and Measurement Timescales

We consider the time scale used to adjust the resources as “adaptation time scale” and the time window used for traffic measurement as “measurement time scale”. Present DRA methods select a time scale on a trial and error basis and there has been little work in the literature that provides insights into how to properly select it. Adjusting the bandwidth too frequently can lead to high overheads, highly fluctuated bandwidth and basically poor performance due to inaccurate prediction due to feedback obtained from too short measurement period. If the bandwidth is not adjusted frequently enough, poor control performance may result due to burstiness of the traffic. A larger measurement time scale lowers the load averages and the smaller measurement times means a higher sampling frequency which leads to higher averages.

Based on our experimental study we advise that measurement time scales can be from 10ms to 100ms and adaptation time scales can be between 100ms and couple of seconds. Since the wavelets have the ability to predict the traffic more accurately and adapt to the changes in traffic, a better time scale range can be covered.

Bibliography

[1] A. Adas. Using Adaptive Linear Prediction to Support Real-Time VBR Video Under RCBR Network Service Model. *IEEE/ACM Transactions on Networking*, 6(5):635–645, October 1998.

[2] P. Almquist. Type of Service in the Internet Protocol Suite. Internet RFC 1349, July 1992.

[3] G. Apostolopoulos, D. Williams, S. Kamat, R. Gurien, A. Orda, and T. Przygienda. QoS Routing Mechanisms and OSPF Extensions. Internet RFC 2676, August 1999.

[4] D. Arifler and B. Evans. Modeling the self-similar behavior of packetized MPEG-4 video using wavelet-based methods. In *Proceedings of IEEE International Conference on Image Processing*, pages 848–851, Sept 2002.

[5] G. Armitage. Revisiting IP QoS: why do we care, what have we learned? RIPQOS workshop report . In *Proceedings of the ACM SIGCOMM Workshops*, pages 81–88, October 2003.

[6] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao. Overview and Principles of Internet Traffic Engineering. Internet RFC 3272, May 2002.

[7] F. Baker, C. Iturralde, F. Faucheur, and B. Davie. Aggregation of RSVP for IPv4 and IPv6 Reservations. Internet RFC 3175, September 2001.

[8] C. Barkat, P. Thiran, G. Innaccone, C. Diot, and P. Owezarski. A flow-based model for internet backbone traffic. In *Second Internet Measurement Workshop (IMW'02)*, November 2002.

[9] Y. Bernet, P. Ford, R. Yavatkar, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski, and E. Felstaine. A Framework for Integrated Services Operation over DiffServ Networks. Internet RFC 2998, November 2000.

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

[11] R. Braden. Requirements for Internet Hosts – Communication Layers. Internet RFC 1122, October 1989.

[12] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource Reservation Protocol (RSVP) –Version 1 Functional Specification. Internet RFC 2205, September 1997.

[13] S. Chong, S. Li, and J. Ghosh. Predictive Dynamic Bandwidth Allocation for Efficient Transport of Real-Time VBR Video over ATM. *IEEE Journal of Selected Areas in Communications*, 13(1):12–23, 1995.

[14] C. Courcoubetis, G. Fouskas, and R. Weber. On the Performance of an Effective Bandwidth Formula. In *Proceedings of the International Teletraffic Congress, ITC-14,Antibes*, June 1994.

[15] Ingrid Daubechies. *Ten Lectures on Wavelets*. ISBN: 0898712742. Society for Industrial and Applied Math, 1992.

[16] B. Davie. Deployment of Differentiated Services. In *Proceedings of the ACM SIGCOMM Workshops*, pages 131–136, August 2003.

[17] B. Davie, A. Charney, J. Bennett, K. Benson, J. Boudec, W. Courtney, S. Davari, V. Firoiu, and D. Stiliadis. An Expedited Forwarding PHB. Internet RFC 3246, March 2002.

[18] G. deVeciana, G. Kesidis, and J. Walrand. Resource Management in Wide-Area ATM Networks Using Effective Bandwidths. *IEEE Journal on Selected Areas in Communications*, 13(6):1081–1090, August 1995.

[19] N. Duffield, P. Goyal, A.Mishra, K. Ramakrishnan, and J. Merwe. A Flexible Model for Resource Management in Virtual Private Networks. In *Proceedings of ACM SIGCOMM'99*, volume 29, pages 95–108, October 1999.

[20] N. G. Duffield, J. T. Lewis, and N. O'Connell. Predicting Quality of Service for Traffic with Long-Range Fluctuation. In *IEEE ICC'95*, pages 473–477, 1995.

[21] E.Rosen, R. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. Internet RFC 3031, January 2001.

[22] E. Fulp and D. Reeves. Online Dynamic Bandwidth Allocation. In *Proceedings of IEEE International conference on Network Protocols*, pages 134–141, October 1997.

[23] J. Gallardo, D. Makrakis, and M. Angulo. Dynamic Resource Management Considering the Real Behavior of Aggregate Traffic. *IEEE/ACM Transactions on Multimedia*, 3(2):177–185, June 2001.

[24] R. Garroppo, S. Giordano, S. Miduri, M. Pagano, and F. Russo. Statistical Multiplexing of Self-Similar VBR Videoconferencing Traffic. In *Proceedings of IEEE GLOBECOM*, volume 3, pages 1756 – 1760, November 1997.

[25] R. J. Gibbens. Traffic Characterization and Effective Bandwidths for Broadband Networks Traces. In *Stochastic Networks: Theory and Application*, volume 4 of *Royal Statistical Society Lecture Notes*, pages 169–179. Oxford University Press, 1996.

[26] M. Grossglauser and D. Tse. A framework for robust measurement-based admission control., volume = 7, year = 1999, month = October. *IEEE/ACM Trans. on Networking*, (9):252–262.

[27] R. Gurin, H. Ahmadi, and M. Naghshineh. Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks. *IEEE JSAC*, 9(7):968–981, September 1991.

[28] F. Haciomeroglu and M. Devetsikiotis. A Comparative Analysis of Online Measurement-based Capacity allocation Schemes. In *Proceedings of the 7th WSEAS International Conference on Communications*, 2002.

[29] Q. Hao, S. Tartarelli, and M. Devetsikiotis. Self-Sizing and Optimization of High Speed Multiservice Networks. In *Proceedings of IEEE GLOBECOM*, volume 3, pages 1818–1823, 2000.

[30] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured Forwarding PHB Group. Internet RFC 2597, June 1999.

[31] I. Hsu and J. Walrand. Dynamic Bandwidth Allocation for ATM Switches. *Journal of Applied Probability*, 33(3):758–771, 1996.

[32] University of Southern California Information Sciences Institute. Internet Protocol. Internet RFC 791, September 1981.

[33] I.Norros. On the use of fractional brownian motion in the theory of connection less networks. *IEEE Journal of Selected Areas in Communications*, 13(6):953–962, 1995.

[34] S. Jamin, S. J. Shenker, and P. B. Danzig. Comparison of Measurement-based Admission Control Algorithms for Controlled Load Service. In *Proceedings of IEEE INFOCOM '97*, pages 973–980, April 1997.

[35] J.Crowcroft, S. Hand, R. Mortier, T. Roscoe, and Andrew Warfield. QoS's Downfall: At the bottom, or not at all! In *Proceedings of the ACM SIGCOMM Workshops*, pages 109–136, August 2003.

[36] F. P. Kelly. Notes on Effective Bandwidths. In *Stochastic Networks: Theory and Applications*, volume 4 of *Royal Statistical Society Lecture Notes*, pages 141–168. Oxford University Press, 1996.

[37] J. Kesidis, J. Walrand, and C. Chang. Effective bandwidths for multiclass markov fluids and other atm sources. *IEEE/ACM Transactions on Networking*, 1(4):424–428, August 1993.

[38] E. Knightly and H. Zhang. D-BIND: An Accurate Traffic Model for Providing QoS Guarantees to VBR Traffic. *IEEE/ACM Transactions on Networking*, 5(2):219–231, April 1997.

[39] S. Li, S. Chong, and C. Hwang. Link Capacity Allocation and Network Control by Filtered Input Rate in High-Speed Networks. *IEEE/ACM Transactions on Networking*, 3(1):10–25, February 1995.

[40] R. Liao and A. Campbell. Dynamic Core Provisioning for Quantitative Differentiated Service. In *Proceedings of IEEE/IFIP IWQoS'01*, November 2002.

[41] C. Macian, L. Burgstahler, W. Payer, S. Junghans, C. Hauser, and J. Jaehnert. Beyond Technology: The Missing Pieces for QoS Success. In *Proceedings of the ACM SIGCOMM Workshops*, pages 121–130, August 2003.

[42] K. Mase. Internet QoS: A Big Picture. *IEEE Network Magazine*, 42(7):42–47, July 2004.

[43] S. Molnar and T. Dang. Pitfalls in long range dependence testing and estimation. In *Proceedings of IEEE GLOBECOM*, volume 1, pages 662–666, November 2000.

[44] A. Moore. Measurement-based management of network resources. Ph.D. Dissertation 528, University of Cambridge, Computer Labarotary Technical Report, June 2001.

[45] P. Muralikrishna, V. Gadre, and U. Desai. Modelling and control of broadband traffic using multiplicative cascades. *Sadhana*, 27(6):699–723, December 2002.

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

[47] K. Nichols, V. Jacobson, and L. Zhang. A Two-bit Differentiated services Architecture for the Internet. Internet RFC 2638, July 1999.

[48] P.Gevros, J. Crowcroft, P. Kirstein, and S. Bhatti. Congestion Control Mechanisms and the Best Effort Service Model. *IEEE Network*, 15(3):16–26, May 2001.

[49] A. Pitsillides, P. Ioannou, and L. Rossides. Congestion Control for Differentiated Services using Non-linear Control Theory. In *Proceedings of 6th IEEE Symp. Computer and Communications*, 2001.

[50] A. Pitsillides, J. Lambert, and D. Tipper. Dynamic Bandwidth Allocation in Broadband ISDN using a Multilevel optimal Control Approach. In *Proceedings of IEEE INFOCOM' 95*, April 1995.

[51] A. Pitsillides, J. Lambert, and D. Tipper. Integrated Control of Connection Admission, Flow Rate, and Bandwidth for ATM based Networks. In *Proceedings of IEEE INFOCOM' 96*, March 1996.

[52] J. Qui and E. W. Knightly. Measurement-based Admission Control with Aggregate Traffic Envelopes. *IEEE/ACM Transactions on Networking*, 9(2), April 2001.

[53] S. Rampal, D. Reeves, and I. Viniotis. Dynamic resource allocation based on measured qos. TR 96-2, Center for Advanced Computing and Communications, North Carolina State University, January 1996.

[54] R.Braden, D.Clark, and S.Shenker. Integrated Services in the Internet Architecture: an Overview. Internet RFC 1633, June 1994.

[55] R. Reidi, M. Crouse, V. Riberio, and R.Baraniuk. A multifractal Wavelet model with application to Network Traffic. *IEEE Transactions on Information Theory*, 45(93), 1999.

[56] B. Ryu and S. Lowen. Fractal Traffic Models for Internet Simulation. In *Fifth IEEE Symposium on Computers and Communications (ISCC 2000)*, pages 200–206, July 2000.

[57] Z. Sahinoglu and S. Tekinay. A Novel Approach Bandwidth Allocation: Wavelet-Decomposed Signal Energy approach. In *Proceedings of IEEE GLOBECOM*, volume 4, pages 2253–2257, December 2001.

[58] M. Schwartz. *Broadband Integrated Networks*. ISBN:0-13-519240-4. Prentice Hall, New Jersey, 1996.

[59] W. Shen. On-line measurement of effective bandwidth and a hierarchical self-sizing framework. Master's thesis, Institute for Electrical and Computer Engineering, Carleton University, 2002.

[60] W. Shen and M. Devetsikiotis. A Self-Sizing Framework for Adaptive Resource Allocation in Label Switched Networks. In *Proceedings of the 10th IEEE International Symposium On Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, 2002.

[61] S. Shenker, C. Partridge, and R. Guerin. Specification of Guaranteed Quality of Service. Internet RFC 2212, September 1997.

[62] S. Shioda, H. Saito, and H. Yokoi. Sizing and Provisioning for Physical and Virtual Path Networks Using Self-sizing Capability. *IEICE Transactions on Communications*, E80-B(2):252–262, February 1997.

[63] P. Siripongwutikorn, S. Banerjee, and D. Tipper. Adaptive Bandwidth Control for Efficient Aggregate QoS Provisioning. In *Proceedings of IEEE GLOBECOM*, November 2002.

[64] P. Siripongwutikorn, S. Banerjee, and D. Tipper. A Survey of Adaptive Bandwidth Control Algorithms. *IEEE Communications Surveys and Tutorials*, 5(1):14–26, July 2003.

[65] S. Tartarelli, M. Falkner, M. Devetsikiotis, I. Lambadaris, and S. Giordano. Empirical Effective Bandwidths. In *Proceedings of IEEE GLOBECOM*, volume 1, pages 672–678, November 2000.

[66] D. Veitch and P. Abry. A wavelet-based joint estimator of the parameters of long-range dependence. *IEEE Transactions on Information Theory*, 45(3):878–897, 1999.

[67] X. Wang, S. Jung, and J. Meditch. Dynamic bandwidth allocation for VBR video traffic using adaptive wavelet prediction. In *IEEE ICC'98*, volume 1, pages 549–553, 1998.

[68] J. Wroclawski. Specification of Guaranteed Quality of Service. Internet RFC 2211, September 1997.

[69] X. Xiao and L. Ni. Internet QoS: A Big Picture. *IEEE Network Magazine*, 13(2):8–18, March 1999.

[70] T. Xusheng, W. He, and J. Chuanyi. A unified framework for understanding network traffic using independent wavelet models. In *Proceedings of IEEE INFOCOM' 2002*, pages 446–454, June 2002.

[71] J. Yan. Adaptive configuration of elastic high speed multiclass networks. *IEEE Communication Magazine*, pages 116–120, May 1998.

[72] H. Zhang and E. Knightly. RED-VBR: A Renegotiation-Based Approach to Support Delay-Sensitive VBR Video. *ACM Multimedia Systems Journal*, 5(3):164–176, May 1997.