Université d'Ottawa • University of Ottawa

PERMISSION DE REPRODUIRE ET DE DISTRIBUER LA THÈSE

PERMISSION TO REPRODUCE AND DISTRIBUTE THE THESIS

NOM DE L'AUTEUR / NAME OF AUTHOR:	HUANG, Aimin
ADRESSE POSTALE / MAILING ADDRESS:	405-345 CLARENCE STREET OTTAWA ON K1N5R5
GRADE / DEGREE:	**ANNÉE D'OBTENTION / YEAR GRANTED**
M.A.Sc. (Electrical Engineering)	2003
TITRE DE LA THÈSE / TITLE OF THESIS: QOS Support and Performance of the Petaweb Optical Network	

L'auteur permet, par la présente, la consultation et le prêt de cette thèse en conformité avec les règlements établis par le bibliothécaire en chef de l'Université d'Ottawa. L'auteur autorise aussi l'Université d'Ottawa, ses successeurs et cessionnaires, à reproduire cet exemplaire par photograhie ou photocopie pour fins de prêt ou de vente au prix coûtant aux bibliothèques ou aux chercheurs qui en feront la demande.

Les droits de publication par tout autre moyen et pour vente au public demeureront la propriété de l'auteur de la thèse sous réserve des règlements de l'Université d'Ottawa en matière de publication de thèses.

The author hereby permits the consultation and the lending of this thesis pursuant to the regulations established by the Chief Librarian of the University of Ottawa. The author also authorizes the University of Ottawa, its successors and assignees, to make reproductions of this copy by photographic means or by photocopying and to lend or sell such reproductions at cost to libraries and to scholars requesting them.

The right to publish the thesis by other means and to sell it to the public is reserved to the author, subject to the regulations of the University of Ottawa governing the publication of theses.

N.B. LE MASCULIN COMPREND ÉGALEMENT LE FÉMININ

04/07/2003
DATE

(AUTEUR) SIGNATURE (AUTHOR)

Université d'Ottawa · University of Ottawa

Université d'Ottawa · University of Ottawa

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

HUANG, Aimin

AUTEUR DE LA THÈSE - AUTHOR OF THESIS

M.A.Sc. (Electrical Engineering)

GRADE - DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT - FACULTY, SCHOOL, DEPARTMENT

TITRE DE LA THÈSE - TITLE OF THE THESIS

QOS Support and Performance Analysis of the Petaweb Optical Network

Dimitrios Makrakis

DIRECTEUR DE LA THÈSE - THESIS SUPERVISOR

EXAMINATEURS DE LA THÈSE - THESIS EXAMINERS

I. Lambadaris

O. Yang

J.-M. De Koninck, Ph.D.

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

SIGNATURE

DEAN OF THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

QOS SUPPORT AND PERFORMANCE ANALYSIS OF THE PETAWEB OPTICAL NETWORK

by

Aimin Huang

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa
Ottawa, Ontario, Canada

A Thesis submitted to
The Faculty of Graduate Studies
University of Ottawa
in partial fulfillment of the requirements for the degree of

Master of Applied Science
in Electrical Engineering

UNIVERSITY OF OTTAWA
FACULTY OF GRADUATE STUDIES

CERTIFICATE OF EXAMINATION

Chief Advisor

Examining Board

Advisory Committee

The thesis by

Aimin Huang

Entitled:

QoS Support and Performance Analysis of the PetaWeb Optical Network

is accepted in partial fulfillment of
the requirements for the degree of
Master of Applied Science

Date

Chair of Examining Board

ABSTRACT

With the dramatic increase of Internet users and the development of new high volume Internet applications, both the volume and dynamic behavior of the Internet traffic has changed. Services based on high-bandwidth multimedia applications will be the dominant sources of traffic in the future. In order to cope with the immense diversity of applications and traffic volume, highly dynamic optical networks will be needed. Recently, optical network technologies have dramatically advances due to several major developments such as Dense Wavelength Division Multiplexing (DWDM), innovative Optical Amplifiers (OA), and wavelength switch. These set the foundation for the Next Generation Optical Networks (NGON) and have profound impact on the design of NGON architectures and technologies.

The research work in this thesis focuses on an alternative next generation optical network architecture, to be operated at multiple petabits per second speeds, called PetaWeb network. The PetaWeb simulation model is implemented, the network functionalities are verified and the network performance metrics involving tandem traffic, reconfiguration period, the number of channel and channel granularity, and the traffic variation are quantified. Furthermore, in order to provide Quality of Service (QoS) over the PetaWeb network, the QoS-Aware edge node and the QoS-Aware channel allocation algorithm are proposed. The performance of the QoS-Aware PetaWeb network is also evaluated and analyzed through simulations.

ACKNOWLEGEMENTS

First and foremost, I would like to express my sincere gratitude and appreciation to my supervisor, Dr. Dimitrios Makrakis. Dr. Makrakis always gave me his advice, and guidance whenever I need them, as well as his trust and patience throughout my study. He also assisted me extensively in the publications generated through this research. Dr. Makrakis shared the wealth of his knowledge and experience in computer communication networks with me, and for these I am grateful.

I would also like to thank Dr. Ognian Kabranov. Dr. Kabranov spent a lot of time with me discussing my research work, giving me a lot of suggestions, helping me with my publications. I also would like to thank my colleagues for sharing their intelligent ideas and thoughts with me and giving me a lot of feedback. Special thanks go to Mr. Tingzhou Yang, Quanyou Zhou, Yixin Dong, Bing Xu, and Mrs. Hong Yu for their useful discussions and help.

My true gratitude and deep appreciation are given to my family for their love, encouragement and support, especially to my wife, Wenhong, who contributed significantly to the completion of this thesis and pushed me to get all of things done. Without them, this study would never have begun and completed.

Finally, I would like to thank all of the ONIT Project members and Nortel Networks' scientists for helping me in this project. Also thank NCIT (National Capital Institute of Telecommunications) for financially supporting this project.

TABLE OF CONTENTS

LIST OF FIGURES

TABLE OF ACRONYMS

AF	Assured Forwarding
ANSI	American National Standards Institute
ATM	Asynchronous Transfer Mode
BA	Behavior Aggregate
BE	Best Effort
DARPA	Defense Advanced Research Projects Agency
DS	Differentiated Services
DSCP	Differentiated Service CodePoint
DWDM	Dense Wavelength Division Multiplexing
EDFA	Erbium-Doped Fiber Amplifier
EF	Expedited Forwarding
FDL	Fiber Delay Line
FIFO	First In First Out
FTP	File Transport Protocol
GPS	Global Positioning System
HTTP	HyperText Transfer Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISP	Internet Service Provider
ITU	International Telecommunications Union
LAN	Local Area Network
MAN	Metropolitan Area Network
MPLS	Multi-Protocol Label Switching
NGI	Next Generation Internet
NGON	Next Generation Optical Network
OA	Optical Amplifier
OPNET	OPtimized Network Engineering Tool

OXC	Optical Cross-Connect
PHB	Per Hop Behavior
PRS	Primary Reference Sources
PSTN	Public Switched Telephone Network
PQ	Priority Queuing
QoS	Quality of Service
RSVP	Resource reSerVation Protocol
RTT	Round Trip Time
RWA	Routing and Wavelength Assignment
SDH	Synchronous Digital Hierarchy
SOA	Semiconductor optical amplifier
SONET	Synchronous Optical Network
TCP	Transmission Control Protocol
TDM	Time Division Multiplexing
TOS	Type Of Service
TSW	Time Slide Window
UDP	User Datagram Protocol
VPN	Virtual Private Networks
WADM	WDM Add/Drop Multiplexers/Demultiplexers
WAN	Wide Area Network
WDM	Wavelength Division Multiplexing
WRR	Weighted Round Robin
WWW	World Wide Web

Chapter 1

INTRODUCTION

It is unthinkable what our life would be without networks. In modern life, there are networks and network based services all around us: private networks, corporation networks, or public networks including the public switched telephone networks (PSTN) and Internet. Using them, we can contact somebody within seconds regardless of the distance, we can retrieve information from remote sites, and we can do banking and stock transactions, shopping without leaving home. More importantly, some of us can receive medical assistance by expert personnel within minutes or even seconds from the development of a life threatening situation, or even be able to receive regular medical examinations or be remotely monitored, to ensure that life threatening situations do not develop. As the time passes, networks are becoming pervasive, existing but not being noticeable, impacting our lives considerably but not been "seen". As the networks become more and more important in our every lives and our functioning, we impose higher and higher service requirements on them. This is fuelling a very rapid technological evolution. A classic example is the fast growth of Internet in terms of capabilities, size, number of users, services. Taking these issues into consideration and the importance Internet is playing today in our every day life is enough to make us understand how important the development of powerful future Internetworking technologies is to us as individuals and as members of a society. At the same time the technical challenges are serious and interesting.

1.1 Trends and Issues in the Future Network

From the time ARPANET was conceived to the current high-speed networks, Internet has seen remarkable growth, especially during the past decade. Similarly, Internet based applications have grown in numbers, users population, quality and sophistication. This explosive growth has been made possible through the technological developments used to

implement the Internet architecture. Services provided by Internet such as File Transport Protocol (FTP), World Wide Web (WWW) browsing, electronic mail, and commercial services (e.g. web banking, online shopping, and Internet telephony) are used by millions of people around the world everyday. However, there is plenty of targets to be met in the future, including better security, development of the foundation for further economic competitiveness, and meeting more challenging social goals (e.g. improved tele-health applications, advanced tele-education services to remote areas and handicapped or sick people) [1].

Applications are one of the factors driving the advancement of networking technologies. In the near future, E-mail with multiple megabytes attachments and downloading multi-megabyte files will be a routine function, as it will be the demand for high quality video content, tele-education service and multimedia conferencing. The bandwidth requirement of the Internet will skyrocket. This bandwidth demand will have dramatic impact on the network design. In the mean time, these bandwidth intensive applications (e.g. high quality audio and video, real-time broadband visualization services and tele-immersive applications) will require that the network has the capability to guarantee the desired quality-of- service (QoS) levels (such as high bandwidth, low latency, and low packet loss rates). A lot of work has been done by the Internet Engineering Task Force (IETF) Working Groups [2] towards the development of IP based QoS supporting technologies, eventually producing two architecture models: Integrated Services (InteServ) [3] and Differentiated Services (DiffServ) [4] [5]).

With the Internet's rapid expansion, Internet users increase dramatically. Currently there are nearly two hundred million hosts [6] and at least half billion of people who use the Internet [7]. These numbers have been doubling approximately every 18 months. The combined explosive growth of the Internet and increasing popularity of bandwidth-intensive applications will generate in the near future bandwidth demand considerably higher of what current high-speed networks can deliver. It is expected that the increasing transmission speeds over fiber reaching terabits per second or even petabits per second and the development of highly efficient optical networks will provide a backbone network infrastructure, capable of supporting these capacity levels.

Measurements conducted through operational networks revealed that the traffic pattern between node pairs shows considerable fluctuation over time [8] [9]. This traffic behavior owes to be taken into consideration while designing the Next Generation Internet (NGI).

1.2 Issues in WDM Optical networks

The rapid advancement of optical technologies will have a major impact on the next generation network's evolution.

Optical fiber based communications, have significant advantages, including high channel capacity, relatively low cost (approximately $0.30 per yard), ability to provide very low bits error rates (typical values of bit error rates are in the range of 10^{-12}; compared to copper cable where error rates are close to 10^{-6} it evident that fiber is considerably more reliable medium), low signal attenuation (0.2 decibels per kilometer [dB/km]), low signal distortion, low power requirement, small space requirements [10]. In addition, optical fiber offers higher security than other media, such as radio and copper cables.

Over the past few years, optical network technologies have dramatically advanced due to several major developments, first is multiple channels optical fiber, which can provide an abundance of bandwidth up to the order of Terabits, and long distance transmission at low cost. It makes cities to cities or continents to continents connectivity become true. Second is wavelength division multiplexing (WDM), which is a method of sending many light beams of different wavelengths simultaneously down the core of an optical fiber. It includes some components such as tunable lasers, tunable filters, WDM add/drop multiplexers/demultiplexers (WADMs), and Optical Cross-Connects (OXCs). And last is the Optical Amplifier (OA), which can amplify optical signals at many different wavelengths simultaneously, regardless of the modulation scheme or speed. These technologies set the foundation for the next generation optical networks and enable NGON with high flexibility in switching, bandwidth provisioning, simplify network management, survivability, and cost reduction.

Currently, commercial wavelength division multiplexing (WDM) systems have been deployed extensively. However, these are all opaque optical networks, meaning that the

data transmitted through a path is not always in the optical domain. They encounter Optical–Electronic–Optical conversion operations at some intermediate elements. In the next generation network, WDM optical networks should be all-optical networks, where data transmitted through the network elements remain in the optical domain. At present, the all-optical network architectures can be classified into two categories: the broadcast and select network architecture, which is suitable for use in high-speed local area networks (LANs) and the metropolitan area networks (MANs); another is wavelength routing network architecture, suitable for used in MAN and wide area networks (WAN) networks. However, there are two kinds of constraints existing in a wavelength routing network: wavelength continuity constraint and distinct wavelength assignment constraint.

Wavelength routing networks have been extensively studied, with a considerable amount of the research activities focusing on the Routing and Wavelength Assignment (RWA) problem. In general, the traffic patterns - either static or dynamic - have been considered. In static traffic demand scenario, the objective is to select routes and assign wavelengths for each demand so as to minimize the number of used wavelengths. Some typical literature can be found in [11] [12] [13]. In the dynamic traffic demand scenario, the objective is to minimize the call blocking probability. Some typical literature can be found in [14] [15] [16]. Also, in order to minimize the effect of wavelength continuity constraint and distinct wavelength assignment constraint, some possible methods such as wavelength conversion, multiple fibers, and wavelength rerouting are employed. Some typical literature can be found in [12] [16] [17] [18] [19].

However, in this kind of networks, there are still some serious limitations in addition to the complexity of algorithms [20]. As each optical node is only connected with several optical nodes, light paths between any edge node pair need to pass through multiple optical nodes. In large-scale networks, it becomes difficult to be able to establish end-to-end connectivity with reasonably low blocking probability and without wavelength conversion. However if wavelength conversion were used, network costs would increase. Also traffic engineering is needed in this kind of networks, but it is difficult and costly to be implemented. Furthermore, due to the absence the optical buffers, timing coordination between several optical nodes along the light path would be extremely hard, if not

impossible. Without timing coordination, the network resource utilization and network throughput would be low.

Fortunately, the optical network technologies have advanced considerably since the last decade. These developments generate new opportunities and research challenges in our journey towards the development of a structure that enables the effective integration of new technologies into the next generation networks.

1.3 Objectives

In this dissertation, we study an alternative optical network, called the PetaWeb network [21] [22]. The PetaWeb network has been introduced by Nortel Networks [23] - through a project supported by Defense Advanced Research Projects Agency (DARPA) - as a possible Next Generation Internet technology. According to the authors of [21] and [22], *"The PetaWeb architecture exploits advances in electronic and photonic switching technology to realize a network that is scalable, tractable, efficient, and almost inexhaustible"*. The PetaWeb network is an agile network comprising several electronic edge nodes and an agile core, comprising of a smaller number of bufferless core nodes that can be reconfigured rapidly in response to traffic variations, which scales to a total external capacity of several petabits per second (Pb/s)—three orders of magnitude higher than the external capacity of the current global Internet".

Introducing a new network architecture requires careful verification. Simulation is an economic and rapid way to perform such task. Building the PetaWeb network in a real environment is challenging, even impossible in the case of the PetaWeb, since some components are still unavailable.

Through simulation, we can assess the performance expectations from the PetaWeb, as well as quantify the network performance metrics (tandem traffic, reconfiguration period, the number of channel and channel granularity, the traffic variation etc.). Through the analysis, we can gain the insight on how these parameters impact the PetaWeb network behavior and network performance.

Furthermore, with the flourishing of e-business and value-added services, network should have the capability to provide specific quality of service. In order to provide QoS over the PetaWeb network, some mechanisms need to be deployed. The QoS-Aware edge node and A QoS-Aware channel allocation algorithm are proposed. The performance of the QoS-Aware PetaWeb network is evaluated and analyzed through simulations.

1.4 Thesis Organization

The remaining chapters of the thesis are organized as follows. In Chapter 2, we give a brief introduction to the state of the art in optical networks and optical networking technologies. In Chapter 3, we discuss the design principles of the next generation network architecture, and briefly overview the PetaWeb network architecture and its main components including the edge node, core node, channel allocation mechanism, and so on. In Chapter 4, we discuss the QoS-Aware PetaWeb network. The QoS-Aware edge node and QoS-Aware channel allocation algorithms are also introduced. The implementations of the PetaWeb network simulation model and QoS-Aware PetaWeb network simulation model based on OPtimized Network Engineering Tool (OPNET) are discussed in Chapter 5. Simulation results and performance analysis are provided in Chapter 6. We first analyze the performance of general PetaWeb network, and quantify the effect of the factors impacting agility of the PetaWeb network. Then the performance of the QoS-Aware PetaWeb network is evaluated and analyzed. Finally, the dissertation is concluded and some suggestions for future work are presented in Chapter 7.

Chapter 2

STATE OF THE ART IN OPTICAL NETWORKS

The dramatic increase of Internet users and the development of new high traffic volume Internet applications make optical communications and networks, which can provide high throughput (up to the order of terabits or petabits), as the preferred technology for backbone networks. Over the past few years, optical networking technologies have advanced dramatically The development of high capacity multiple channels optical fiber with low attenuation levels has produced an abundance of bandwidth - up to the order of terabits- and allows long distance transmission at low cost. It makes cities to cities or continents to continents connectivity become true. Also, the rapid advances in DWDM technologies and optical components (e.g. dense WDM add/drop multiplexers (ADMs), wideband optical amplifiers, tunable lasers, and novel optical cross-connects (OXCs)) have pushed the frontiers further. These technologies set the foundation for the next generation optical networks and enable NGON with high flexibility in switching, bandwidth provisioning, simple network management, survivability, and cost reduction. According to [24], The next generation of optical transmission will be based on N x 40 Gbit/s systems, providing capacities of several Tbit/s.

In this chapter, we will survey several promising technologies for optical data communication. In particular, focusing on the dense wavelength division multiplexing (DWDM) technologies. The organization of this chapter is as follows: Section 2.1 overviews the DWDM system and optical components. The architecture of all-optical networks is described in section 2.2.

2.1 DWDM System Overview

2.1.1 Dense Wavelength Division Multiplexing (DWDM)

Dense Wavelength Division Multiplexing (DWDM) is a fiber-optic transmission technique. It transmits data using different wavelengths of light over an optical fiber. Using DWDM, up to 128 (and theoretically more) separate wavelengths or channels of data can be multiplexed into one light stream transmitted on a single optical fiber. Each optical channel or wavelength may transport different services, such as Asynchronous Transfer Mode (ATM), Internet Protocol (IP), Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH), or voice, Video-IP. DWDM is an important innovation in optical networks. A general overview of DWDM can be found in [25], [26], and [27]. DWDM systems and networks are applicable in access networks, in local area networks, in metropolitan area networks, and in backbone networks that have large geographical coverage and can transport ultrahigh bandwidth at very high transmission rate.

Currently, commercial systems with 16, 32, 40, 80, and 128 channels per fiber have been announced [26]. The more data channels over one fiber, the closer space between the wavelengths. The International Telecommunications Union (ITU) [28] sets the industry standards. It is reasonable for us to assume that DWDM systems with hundreds of channels in one fiber will become available in the near future. Theoretically, more than 1000 channels may be multiplexed in one fiber. Figure 1 illustrates the concept of combing multiple channels over a single fiber in a DWDM system.

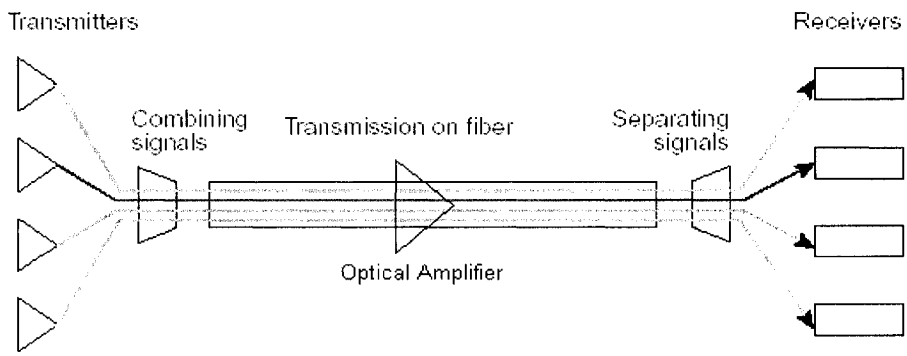


Figure 2.1 multiple channels over a single fiber in a DWDM system

The system performs the following functions:

- Transmitters generate the signal --- the light source, made from solid-state technology, provides the specific wavelength carrying the modulated signals for data transmission.

- Multiplexers combine the signals --- wavelength multiplexers are used to combine the signals to the single fibers.

- The signals transmitted in fiber --- The effect of the crosstalk and optical signal loss must be considered in fiber optic transmission. These should be minimized through adjust controllable parameters such as channel spacing, laser power levels etc.

- Optical Amplifiers amplify the optical signals --- In order to achieve long distance transport, inevitably in MAN/WAN, optical amplifiers are used to compensate for various losses (e.g. fiber attenuation, coupling and splitting loss in couplers etc.).

- Demultiplexers separate the signals --- the multiplexed signals must be isolated respectively according to the specific wavelength at the receiving end.

- Photo-detectors receive the signals—The demultiplexed signal is received by the photo-detectors and modulated into digital signal.

Most DWDM systems are designed to support several services, and have the capability to process all protocols required by each service supported (e.g. ATM, SONET/SDH, IP, telephony, video, and so on). As result, data signals (e.g. ATM, SONET, and IP) are transmitted through the same stream. The fact that during the transmission process there is no signal termination and no optic-electric-optic conversion within the optical layer, allows the independence of the bit-rate and protocols in DWDM system. This allows DWDM technology easily integrated with the existing devices in the network. This also allows provides flexibility, allowing the service providers to expand the network easily. The service providers can lease only one dedicated wavelength rather than the entire fiber to their customers.

2.1.2 Optical Components

DWDM is a core technology for future optical networks. Its development depends on several optical components such as optical fiber, multiplexer/demultiplexer, optical amplifier, optical source like laser, photodetector, optical cross-connector, and so on. Although the technologies in optical and semiconductor evolved fast over in recent years, the optical components technology is far from being mature. Production of optical components in large quantities is still at a very early stage and the price of these components is still very high (even though the cost has already dropped several orders of magnitude over the past few years). Development of lower cost, higher performance compact optical components, manufactured at higher volumes is essential. Some of optical components of the DWDM system are described as follows.

2.1.2.1 Optical Amplifier (OA)

As the optical signal travels through the optical network, it suffers various losses such as fiber attenuation, coupling and splitting at the couplers, as well as coupling loss at the wavelength routers. In order to maintain the bit error rate at low level, as well as support long distance transmission in all-optical networks such as MAN/WAN, optical amplifiers are required, to amplify the signal.

Some key features of the optical amplifier are gain, gain bandwidth, gain saturation, amplifier noise, and polarization sensitivity.

- Gain is the ratio of the output power of a signal to the input power measured in dB.
- Gain efficiency is the gain as a function of the input power in dB/mW.
- Gain bandwidth is the range of frequencies or wavelengths over which the amplifier is effective.
- Gain saturation is the maximum value of the amplifier output power. Beyond this value, the output power does not increase further by increasing the input power.

In optical amplifiers, noise arises from the spontaneous emission of photons. The amount of noise depends on several factors (such as the amplifier gain spectrum, the noise bandwidth, and so on).

Polarization sensitivity is the dependence of the optical amplifiers on the polarization of the signal.

There are two basic types of optical amplifiers: rare-earth-ion-doped fiber amplifiers [29] [30] and semiconductor laser amplifiers. A general overview of optical amplifiers can be found in [31].

Semiconductor optical amplifiers (SOAs) were developed in the 1980s; SOAs are normally manufactured with InGaAsP. A semiconductor optical amplifier consists of a modified semiconductor laser. The weak signal transmits through the active region of the amplifier, which, via stimulated emission, results in a stronger signal. Thus they are small, compact, and able to be integrated with other semiconductor and optical components. However, the semiconductor optical amplifiers have some limits such as high crosstalk and high noise, thus are not suitable for long-distance transmission (detailed please see [39]).

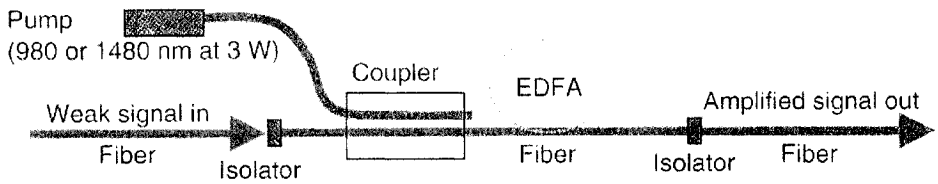


Figure 2.2 EDFA amplifier

Rare-earth-ion-doped fiber amplifiers are manufactured by doping a piece of fiber with an element (normally rare earth), which can amplify the light. Among the rare-earth-ion-doped fiber amplifiers, the most attractive and important, particularly for DWDM systems, is the erbium-doped fiber amplifier (EDFA) [30]. The typical EDFA amplifier (see figure 2.2, quoted from [26]) consists of a coupling device, an erbium-doped fiber, and two isolators (one per EDFA end). Pump light produced from an external laser in either of two

pump bands, 980 nm or 1480 nm, is coupled into the fiber, exciting the erbium atoms. Optical signals at wavelengths between 1530 and 1620 nm entering the fiber stimulate the excited erbium atoms to emit photons at the same wavelength as the incoming signal. This amplifies a weak optical signal to higher power. EDFAs can simultaneously amplify signals over a range of wavelengths, making them compatible with wavelength division multiplexing (WDM) systems. Fiber amplifiers allow overcoming attenuation losses in optical fiber, stretching transmission to hundreds or thousands of kilometers without the need for a more-complex regenerator. Simultaneous amplification of many optical channels made WDM practical over long distances. Amplification also can compensate for losses that occur when optical power is split among multiple routes, which is done to reduce costs and/or allow new network topologies.

2.1.2.2 Optical Cross-connect (OXC)

Optical cross connecting is a key function in optical communication systems. With the emergence of WDM networks that carry large number of wavelengths channels, a new level of cross-connect in the optical domain is highly desirable [32].

Optical Cross-connect is an optical reconfigurable network element, which addresses flexible optical routing of individual channels. Due to the transparency of the protocols and the ability to switch wavelengths, the Optical Cross-connect allows service providers to efficiently move wavelengths from point to point in the network, without making unnecessary and costly optical-electric conversions along the path. The network reliability is increased and operation costs are decreased as well because of the fewer network elements. Currently, the development of genuine optical cross connects is still at an early stage. Most of the existing ones aren't all optical; they are based on electric switch fabrics. All-optical cross-connect based on fully optical switch fabrics can support only few channels, up to 256 X 256, due to the technical hurdles standing in the way of all-optical network components [34]. The all-optical cross-connect with the range of 1000 X 1000 is still in the experimental phase. At present, an economically feasible and reliable 1000 X 1000 all-optical, non-blocking, and dynamically reconfigurable switch is challenge, but the technology is promising [26].

A typical Optical Cross-Connect (OXC) is shown in Fig.2 (quoted from [33]). It consists of a total of *N* demultiplexers at the input end, *N* multiplexers at the output end, and *M* optical space switches. Each of the input fibers contains *M* different wavelengths to an optical demultiplexing. The demultiplexer spatially separates the incoming optical stream into M wavelengths. Then, the wavelengths are sent through the corresponding space switches to the output ports of the optical cross-connect [33]. The multiplexer combines the output wavelengths from the M space switches into one optical stream at the output end.

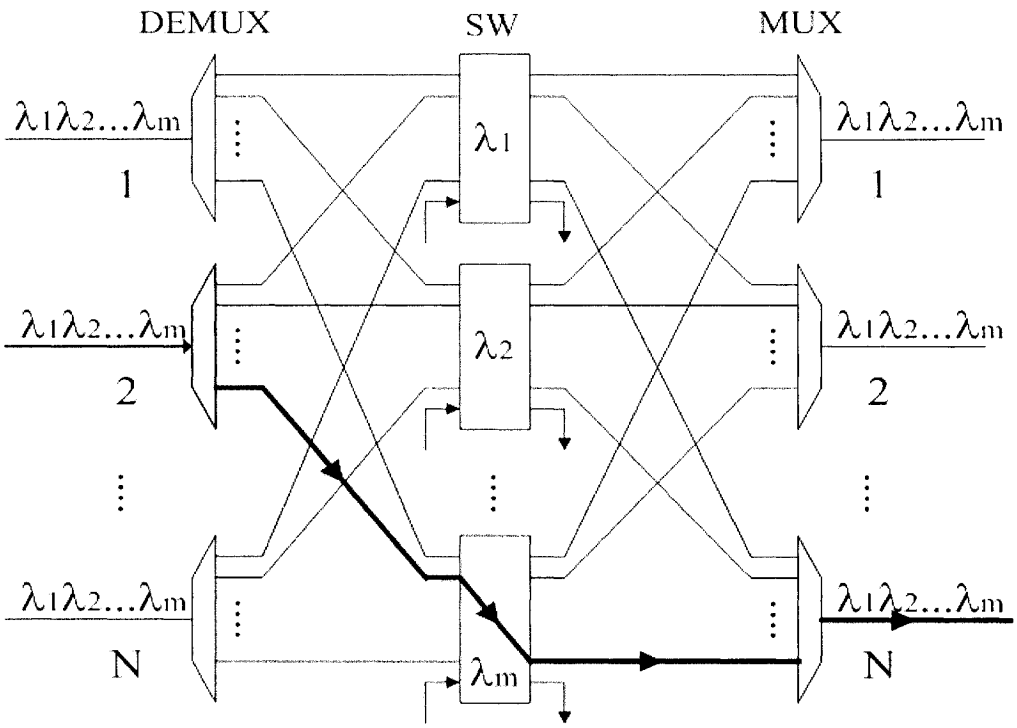


Figure 2.3 Structure of Optical Cross-Connect

Generally, all-Optical Cross-Connect fabrics are based on the following methods [26]:

- Free space optical switching
- Optical solid-state devices

- Electro-mechanical mirror-based devices
- The polarization properties of liquid crystals and other properties of materials

Currently, the reconfiguration time of the Optical Cross-Connect depends on several factors, such as the materials used to make the switch, the principle on which the switch operates, and the design technologies. The times vary from milliseconds to nanoseconds [26]. However, the reconfiguration time is only one of the important parameters in the selection of the switch type. More detailed introduction of Optical Cross-Connect can be found in [35][36].

2.2 The Architecture of an All-Optical Network

In the all-optical network, the data is transmitted on an optical signal waveform through all network elements, from the ingress node to egress node; optical-to-electric conversions are not allowed at intermediate nodes. At present, the all-optical network architectures can be classified into two categories: one is broadcast and select network architecture; the other is the wavelength routing network architecture. The architectures will be briefly described below. More detailed description can be found in [37] [38] [39].

2.2.1 Broadcast and Select Networks

A broadcast and select network consists of nodes interconnected to each other via a star coupler. The structure of broadcast and select network is shown in figure 2.4. There are two optical fiber links between the node and the star coupler. The coupler combines the incoming signals from all the nodes, and distributes the combined optical signal evenly to all its outputs. Each node selects the specific desirable signal through filtering, performed by a tunable optical filter.

However, this kind of network has some severe limitations. First, it is not scalable, since it requires as many wavelengths as nodes in the network. Also, since every node is associated with a wavelength, wavelength reuse is not possible. Furthermore, it cannot span over a wide area, since the splitting of the transmitted power among the various nodes allows only a small fraction of the transmitted power to reach each node.

This kind of network is simple and suitable for high-speed local and metropolitan area networks. Examples of such network are Lambdanet [40], Rainbow [41] [42], Lightning [43] and STARNET [44].

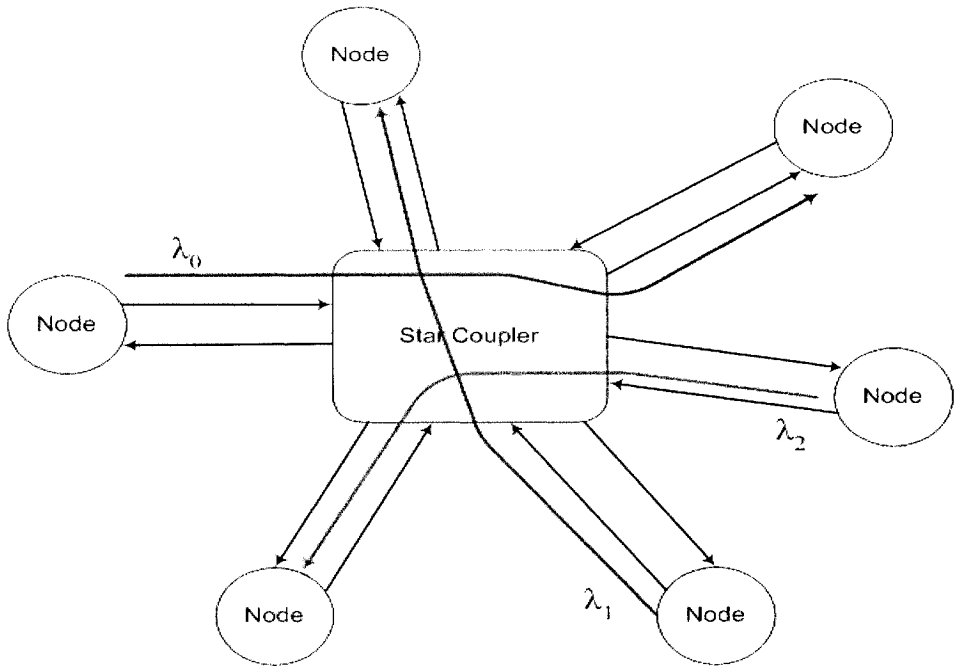


Figure 2.4 The Structure of Broadcast and Select Network

2.2.2 Wavelength Routing Networks

A wavelength routing network consists of nodes interconnected by WDM fiber links, as shown in figure 2.5. The nodes have the capability to route and switch the optical signals from an input port to different output ports. There are two kinds of nodes: one is the static wavelength router (or Optical Cross-Connect) with fixed, non-reconfigurable routing pattern; the other is the reconfigurable wavelength router, which can dynamically change its routing pattern. Based on the type of routers used, a network is either, static wavelength routing network or reconfigurable wavelength routing network. It is possible to set up several simultaneous lighpaths using the same wavelength in the network; as result the resources could be reused spatially. A lightpath is a connection consisting of a path between two nodes and a specific wavelength assigned on this path. The lightpath can pass through several nodes.

However, there are two kinds of constraints existed in this kind of network. One is the wavelength continuity constraint. It means the same wavelength is required on each of nodes and links along the lightpath. The other constraint is the distinct wavelength assignment constraint, which implies that any two of the lightpaths on any fiber cannot be assigned the same wavelength. Routing and assigning wavelengths to lightpaths is an important and complex problem. Highly intelligent and simple algorithms are needed to ensure the network's performance remains at high levels. Generally, this kind of network is suitable for use in MAN and WAN networks. Examples of such network are MONET [45], which is sponsored by Defense Advanced Research Projects Agency (DARPA), AON [46], which is deployed by AT&T, MIT, and DEC. Some tutorials and literature on this kind of networks can be found in [47] [48] [49] [50] [51].

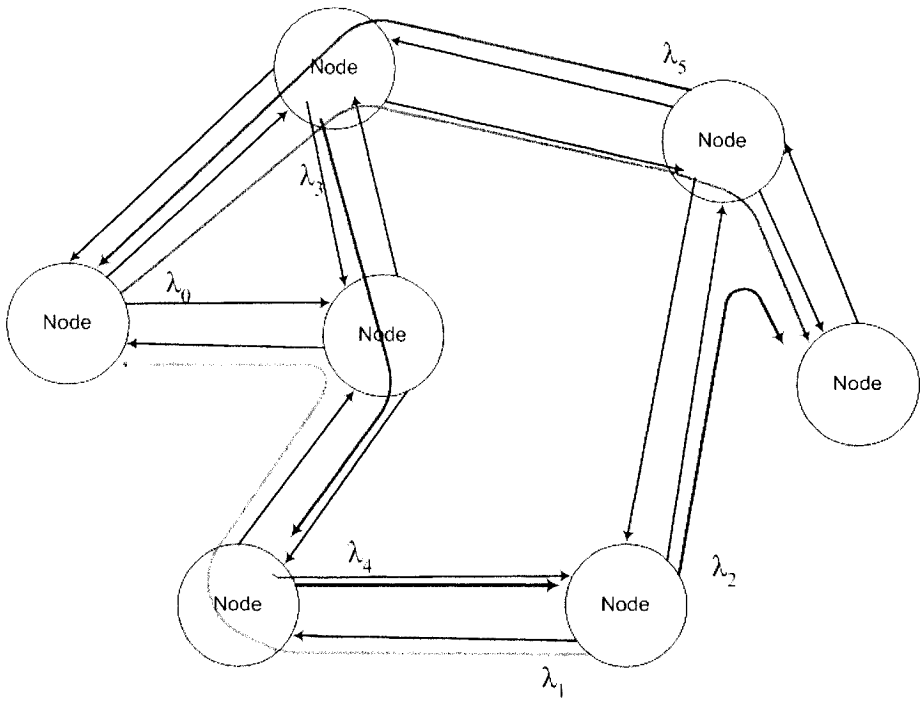


Figure 2.5 The Structure of Wavelength Routing Network

Chapter 3

NEXT GENERATION OPTICAL NETWORKS AND THE PETAWEB NETWORK ARCHITECTURE

3.1 Introduction

As we mentioned earlier, during the past decade, Internet and the Internet application have experienced remarkable growth. However, the structure of current Internet lacks extendability and prohibits its growth, thus, the current Internet architecture could not be used in next generation Internet. Fortunately, the Integrated circuit technology and optical technologies had seen rapid advancement. Some technologies (such as DWDM) have been exploited extensively in the data communication networks, shown considerable growth and are making significant impact. These developments create new opportunities and research challenges for the effective integration of new technologies into the next generation networks.

Since the end of 1990's, there have been significant research contributions working in the area of next generation networks. A topic of interest is the design of a high-capacity wide coverage controllable and scalable data network. In this dissertation, we focus on possible alternative network architecture, called the PetaWeb Network, which is proposed by the Nortel Networks [21] [22]. In the remaining part of this chapter, we first overview the network architecture principles, then briefly introduce the PetaWeb Network architecture and its key components.

3.2 Architecture Principles

3.2.1 Issues impact on Network

3.2.1.1 Network Connectivity and scalability

Generally speaking, there are two kinds of networks, those without core nodes, with the edge nodes interconnected directly; and those where edge nodes are interconnected through several geographically distributed core nodes. Here we consider one network consisting of a large number of edge nodes; For simplicity, we assume that all edge nodes are identical; each node having L fiber links, and is connected with other edge nodes, either directly or through a network core; each fiber link is a multi-channel fiber supporting W wavelengths (channels).

The network without core nodes is shown in Figure 3.1 (quoted from [22]). Each edge node can only be interconnected to at most L neighboring edge nodes; thus, there are ($N - L$ - 1) edge nodes that cannot be connected directly with this given node; this becomes possible only through tandem edge nodes. Thus, the network traffic between non-connected edge nodes must be sent through several intermediate edge nodes (through multi-hop path). The traffic between adjacent edge nodes can also use a multi-hop alternate route if the capacity of the link cannot accommodate the traffic volume. A network of this structure is not scalable to very high capacities.

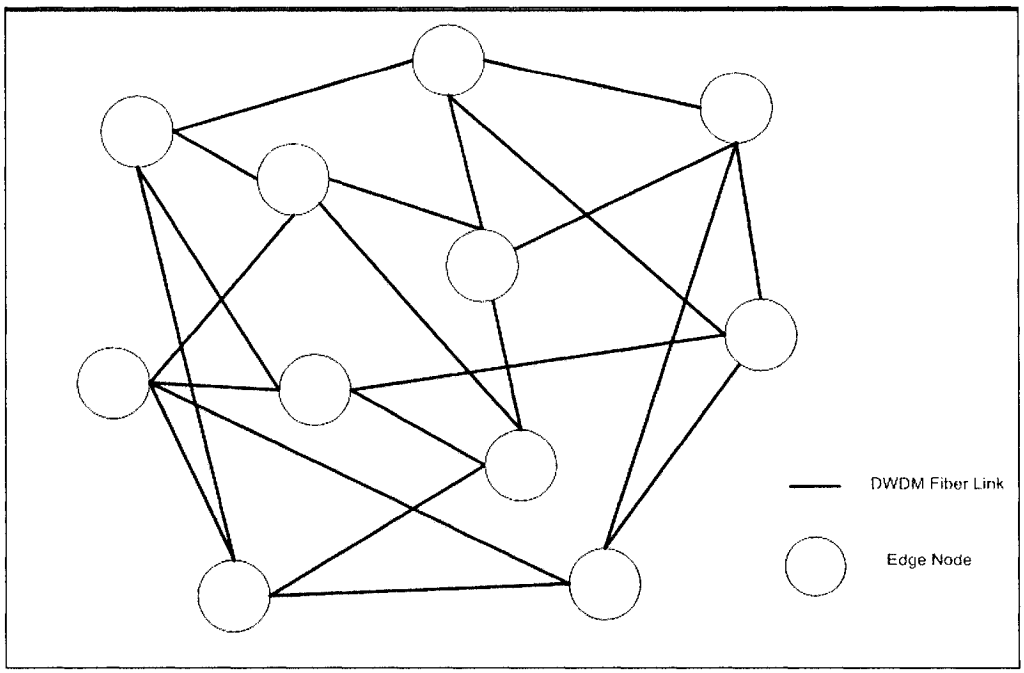


Figure 3.1 Networks without Core Nodes

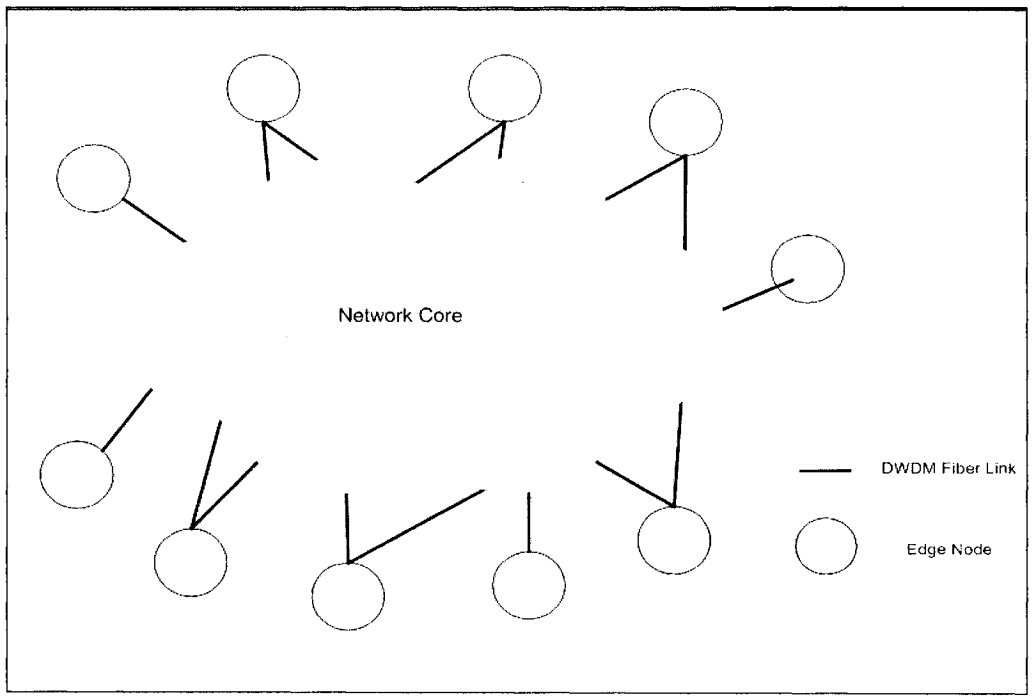


Figure 3.2 Networks with Core Nodes

The network with core nodes is shown in Figure 3.2 (quoted from [22]). There are L fiber links from each source node to the core, and L fiber links from the core to each sink node. Here we define the topological reach of an edge node, which is the number of sink nodes the source node can reach through the network core, without having to switch through one or more intermediate edge nodes [22]. The topological reach determines the network scalability. In a network with core nodes, the traffic at any two edge nodes can be transmitted through the network core; multiple hop routes can also be used if necessary. Obviously, the network scalability is better than the previous case.

3.2.1.2 Mean number of hops and outer-to-inner capacity ratio

Usually, the network performance and network efficiency are key indicators used to evaluate the network. Network performance can be improved by reducing the mean number of hops (a hop is the trip a data packet takes from one router or intermediate point to another in the network.). Network efficiency can be improved by through increasing outer-to-inner capacity [22].

Outer capacity of a network is the total capacity of the outer ports of the network, which are connected to the traffic sources and sinks. It is the capacity available to the network users. Inner capacity is the total capacity of the inner ports, which usually interconnect the network nodes. Increasing the ratio of the outer-to-inner capacity improves the network efficiency. The lower the outer-to-inner capacity ratio is, the less efficient the network design is.

In general, for a good network design, the ratio of outer-to-inner capacity should be low, and the mean number of hops should be small.

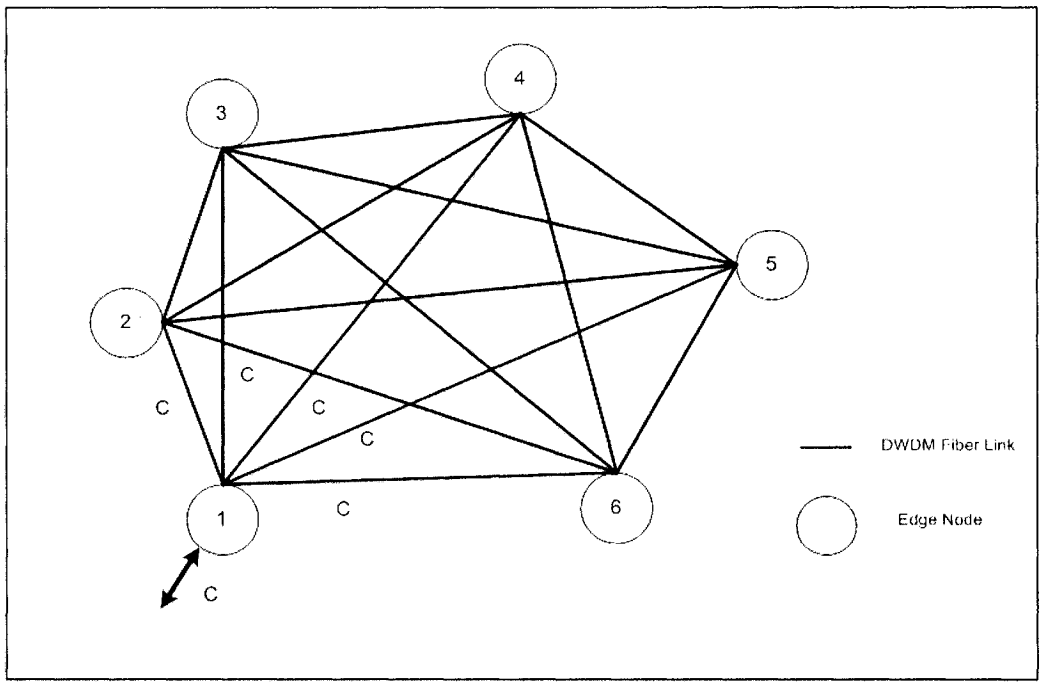


Figure 3.3 Full Mesh Networks

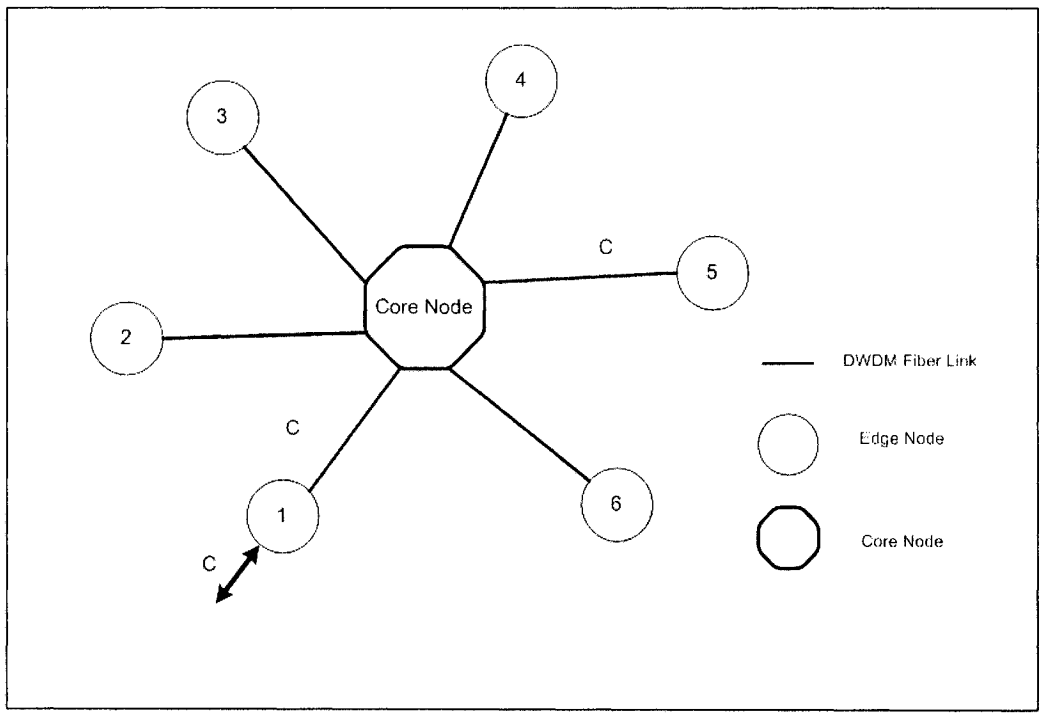


Figure 3.4 Non-blocking Star Networks

For example, here are two networks, illustrated in Figure 3.3 and Figure 3.4 (quoted from [22]); we will compare these two networks. Let us make the following assumptions: each network has N nodes; each node has an outer capacity denoted C; each link, which connects any two nodes, is a bi-directional link with capacity C in each direction. Either a full mesh structure network or a star network can realize a non-blocking network. In the full mesh network depicted in Figure 3.3, where each node is connected to each other node, regardless of the traffic distribution, each traffic stream can be transmitted through a single hop from ingress to egress. As a result, the ratio of the outer capacity to inner capacity is $1/(N - 1)$. In the star network, depicted in Figure 3.4, each edge node is connected to a single core node. Assuming that the core node is constructed with sufficient capacity, the network becomes non-blocking. In this case, each data stream from any edge node to any other edge node must traverse over two links and through the core node. As a result, the ratio of the outer capacity to inner capacity is one half. If N is a large value, the star network provides a considerably higher level of efficiency. However, on the other hand, the mean number of hops is always one in the full mesh network and two in the star network.

3.2.1.3Tandem switching

In the case where the topological reach is less than the number of sinks nodes, the sink nodes that cannot be reached directly from a source node through the core can be reached by tandem switching through intermediate edge nodes. Traffic, which is transmitted through a tandem path, is called tandem traffic. Tandem switching has some shortcomings. The outer capacity of the network trends to be lower than the inner capacity, as well as adds delay and delay jitter to the traffic streams. Minimization or elimination the tandem traffic passing through a network is a positive step towards the development of a well-designed efficient optical network.

3.2.2 Network Design Principles

When designing an architecture for the next generation Internet network, the most critical aspect we need to consider is the simplicity of the structure. Simplicity is the key attribute of any wide coverage network.

According to [22], the design of the PetaWeb network architecture is focusing on the minimization of the most important cost affecting network metric: the inner-to-outer traffic ratio (as we mentioned above). Consequentially the weighted mean number of hops for each communicated data unit is reduced. Furthermore, as we mentioned before, reducing the weighted mean number of hops enhances the network performance, reduces the cost per data unit, and simplifies network control, thereby enabling the creation of an autonomous self-governing network. Coincidentally, reducing the number of hops facilitates network expansion both in capacity and coverage.

Network structural simplicity also is linked to the simplicity of network access. The complexity of network access is somewhat influenced by the addressing scheme of the end devices. However, an addressing scheme is architecture independent, and facilitates the introduction of simple access devices to support the end devices. The PetaWeb design addresses this matter as well. However, because this is beyond the scope of the research conducted in this dissertation, we will not elaborate further. The interested reader can find detailed information in [22].

Finally, any new network cannot replace the current Internet network; it's desirable to make the next generation network be compatible with the current Internet.

3.3 PetaWeb Network Architecture Overview

3.3.1 PetaWeb Architecture Definition

Nortel Networks, proposed an alternative next generation optical network architecture, to be operated at multiple petabits per second speeds, known as the PetaWeb, which is based on recent developments in DWDM technologies. The PetaWeb architecture (shown in Fig. 3.5) is based on a star topology, with an adaptive core in the middle and edge switches at the periphery. The light paths only need to pass through a single core node located between edge nodes. The PetaWeb is a composite hub-and-spoke network, which can provide coverage at a global scale. It is designed to operate at multiple petabits per second speeds, and is based on the use of adaptive channel switching core and edge switches. The adaptive core consists of several bufferless core nodes, which can be independent and distributed over a wide geographic area; each core node comprises a

number of optical space switches, having the ability to reconfigure channels between edge nodes several times per second under the core controller, in order to accommodate traffic fluctuations. Edge nodes are electronic packet switches with high capacity, up to multi-Terabits, which can be distributed over a wide area. Each edge node can be connected to multiple core nodes in order to improve the network survivability and connectivity.

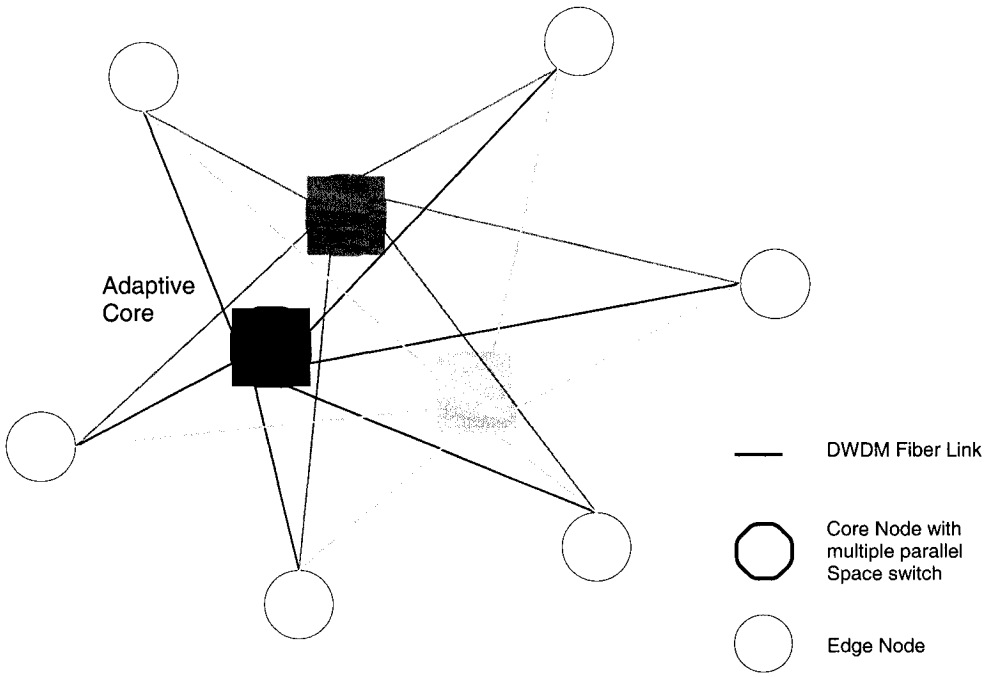


Figure 3.5 Composite Hub-and-Spoke structure of the PetaWeb

The PetaWeb is an edge-controlled or edge-driven network. The incoming traffic is aggregated and controlled at the edge nodes, where the network control is exercised. The core controllers select light paths through associated core nodes, and reconfigure the light paths between the ingress edge nodes to egress edge nodes in response to requests for traffic accommodation, generated by the edge nodes. Most of the traffic can be transported from ingress node to egress node on direct light paths, going through the optical core. However, some of traffic may be transported through intermediate edge nodes due to the fact that there is no direct lightpath between the two corresponding edge nodes available. This traffic (that is forced to go through intermediate edge nodes where lightwave conversion will occur, with the packets going – for this purpose – through optical to

electrical to optical transformations) is called tandem traffic depending on its quality of service requirements and the loading of the network. It is desirable that that the amount of tandem traffic remains low.

The reconfiguration of channels is performed periodically, based on the traffic volume, measured at each edge node. This increases the effective capacity of the network, reduces the average end-to-end packet delay and delay jitter, and improves the performance.

In order to make the PetaWeb a geographically distributed network and avoid loss of data information, synchronization between edge nodes and core nodes is required, which is supported by appropriate timing and synchronization mechanisms.

3.3.2 Edge Node — Universal Packet Switch

In PetaWeb, the network capacity is determined by the capacities of the edge nodes and core node. In order to build high-performance high-capacity networks, the high-capacity switches for variable packet size traffic are desirable. The edge node is a high capacity universal switch with capacity up to several terabits per second, preferably having a large number of ports, including a pair of input/output ports. Each edge node consists of a source node that receives traffic from input ports and a sink node that delivers traffic to the output ports. In general, the universal packet switch has the capability to switch variable-size packets under rate control, as well as has the capability to handle rate-regulated traffic streams as well as unregulated traffic streams (for example IP traffic). In order to support IP traffic, the universal switch has the capability to allocate service rates internally, based on the occupancy of buffer size. Here we briefly introduce the high-capacity switches. The detailed description of universal switch is given in [22].

In a high-capacity switch, it is necessary to use a switching technique that allows the switches to have the ability to coordinate the transfer of data between ingress and egress nodes (move traffic through the network) while at the same time are capable of creating or breaking a path from any ingress port to any egress port with required capacity (connection establishment and breakdown). Two alternative architectures may be considered. Each switch node includes an edge switch and core switch. One is using a space-switched core

(shown in figure 3.6, quoted from [22]), which uses a space-switch stage in the core; the other uses a rotator core (shown in figure 3.7, quoted from [22]). It consists of two rotators and an array of memories. A detailed description of the rotator switch architecture is given in [52]. Each of these architectures has been used for fixed-sized packet applications such as ATM. Control methods are required to make these known switch architectures capable of transmitting variable size packets at high speed switching. In [22] one method of reciprocal traffic control in a switching node for use in data network is proposed. The method includes the following steps:

- Incoming data packets are sorted into ingress buffers at the source modules according to the sink module order, such that the packets are separated into traffic streams logically;

- Variable-sized packets are segmented into predetermined length, and each segment is labeled using a unique identifier associated with a source module at which the packet entered; and

- At the sink modules, the segments of the packets are sorted into egress buffers at the sink modules according to their unique identifier. The segments of any packet are juxtaposed in consecutive order in an egress buffer.

- The packet is delivered to the network by the sink module only after its last segment is completely received.

Each packet leaves the switch in the same format in which it arrived at the switch. Traffic streams flowing through the switch are rate controlled from ingress to egress. Consequently, transmitting variable-sized data packets across a network prevents from excessive waste of network link capacity due to overhead. An important feature of the universal switch is the transfer-rate control of variable-size packets.

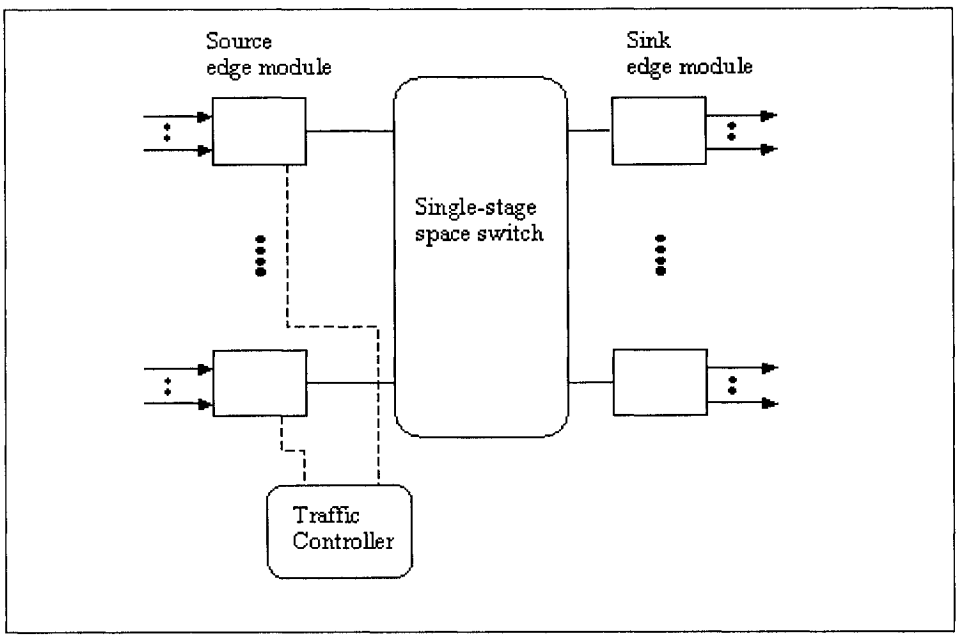


Figure 3.6 the structure of Universal Switch with a space-core

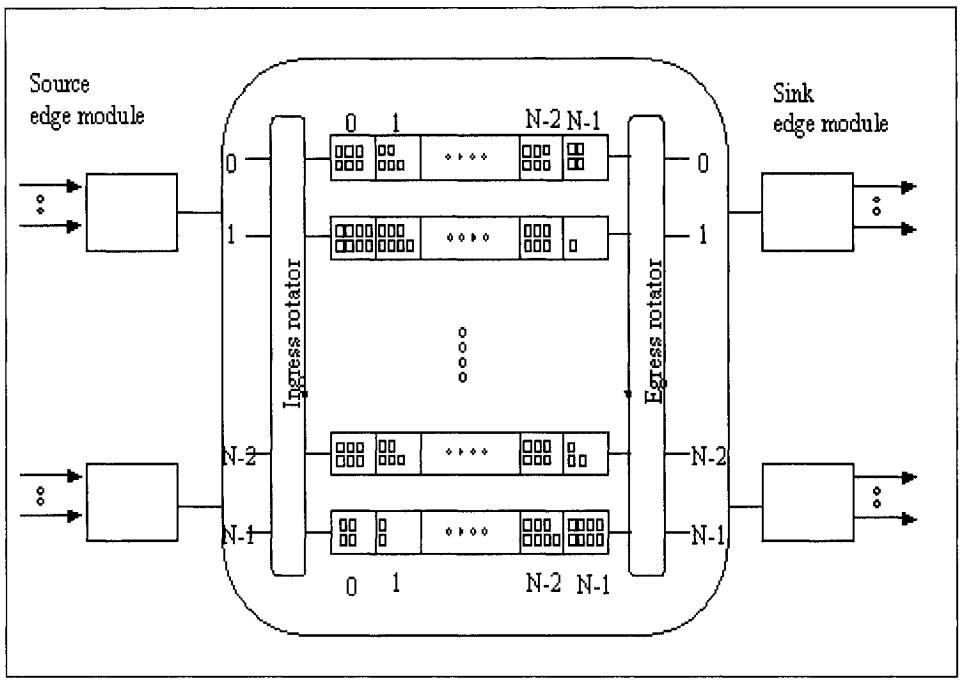


Figure 3.7 the structure of Universal Switch with a rotator core

3.3.3 Agile Core

As mentioned before, the PetaWeb network comprises a number of electronic universal switches interconnected by an optical core; the optical core comprises a number of core

nodes. The core nodes are preferably optical space switches. Clearly, the structure of the core node determines the capacity of the core nodes, thus determines the capacity of the network. The functionality and connectivity of the core nodes have a significant impact on the network function and performance.

3.3.3.1 Core Node Structure

Two alternative structures of the core node are investigated here. Assume the DWDM fiber link has W wavelengths (channels), each space switch has P ports, and the number of links, which may be connected to the core node, is denoted as L. In the first scenario, depicted in figure 3.8a (quoted from [22]), each channel of the DWDM fiber links is connected directly to one port of the optical space switch. It is possible to allow any channel in one link to be connected to any channel in another link. In order to implement arbitrary (or unrestricted) channel switching within the same space switch, wavelength conversion is inevitable. In the second scenario, depicted in figure 3.8b, there are W space switches used in the core node and the W channels of each link are connected to different space switches respectively. Wavelength conversion is not needed in order to enable unrestricted channel switching.

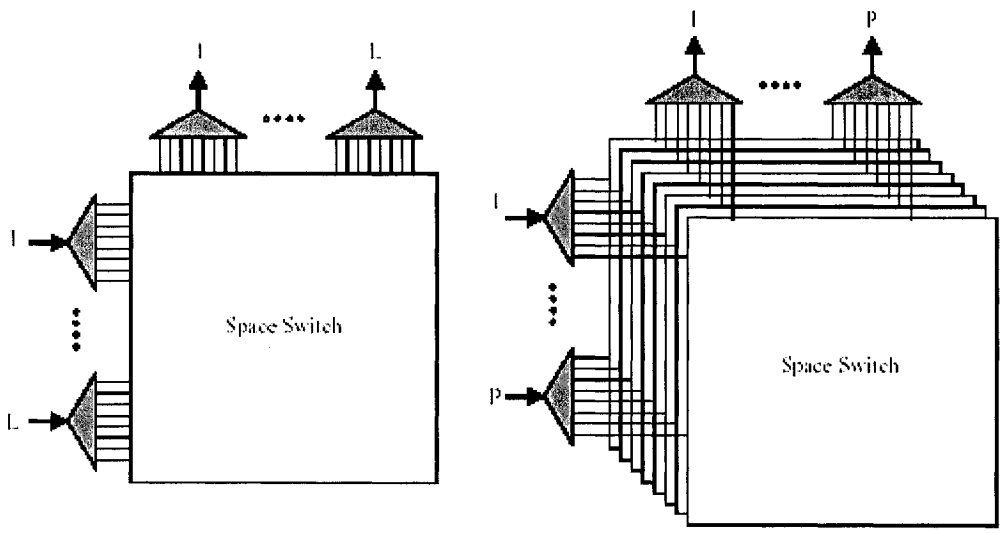


Figure 3.8 the structure of Core Node

Clearly, the second structure can support more DWDM fiber links than the first one. However, several space switches can be interconnected in a mesh or partial mesh structure, in order to support a larger number of DWDM fiber links, while permitting connection of any channel to any other. Internal links, to be connected between space switches, are inevitable. Consequently, the second structure of core node has a lot of advantages compared with the first one. For example, fewer space switches are needed in order to construct the same capacity of core node; channel switching and TDM switching modes are supported, as a result, providing higher topological reach and higher efficiency; more ease of core reconfiguration. Core nodes with parallel space switches are used in the PetaWeb core.

3.3.3.2 the Functionality and Connectivity of the Core Nodes

According to current photonic technologies, optical buffer is not as mature as the electronic counterpart. Aside from laboratory experiments, use of a fiber delay line (FDL), proposed in [53] by D. B. Sarrazin et al. is the only one practical approach to provide buffering without optical-electronic-optical conversion. Due to the ultra high transmission rates over fiber, a very long fiber is needed to buffer a data unit (packet). This leads inevitable to practical difficulties. At present, core nodes without buffers are desirable in all-optical network.

The tremendous growth of Internet traffic makes optical network with agile core a very promising solution, able to address the existing and projected high traffic volumes. In an agile core, the core capacity allocations are adapted in response to variations of the traffic distribution. The core node controller selects paths through an associated core node and reconfigures the paths in response to dynamic changes in data traffic loads. Control at the edge provides one degree of freedom. Adaptive control of the core channel connectivity adds a second degree of freedom.

The PetaWeb network, considered here, is a high capacity and wide coverage data network. The edge nodes and core nodes are distributed over a large geographic area, leading to significant propagation delays among them. Thus, time coordination between the

edge nodes and the core is needed to enable adaptive reconfiguration and to ensure no data loss during channel reconfiguration. Time coordination can be accomplished by exchanging timing packets between the edge nodes and core node. (Detailed description will be provided later). Without time coordination, a large guard time, of the order of several milliseconds, would be needed. It is certainly unacceptable that some of the channels would be out of service during reconfiguration. The reconfiguration functions of the data switch nodes and the core nodes are coordinated to keep the reconfiguration guard time at a minimum.

However, the connectivity of the core nodes also has a strong impact on the feasibility of the time coordination. Generally, there are two kinds of connectivity for a number of core nodes in the network core. One is the *"independent core nodes"*. As the name implies, in this case, the core nodes are disconnected with each other, but connected directly with a number of edge nodes, as illustrated in figure 3.9 (quoted from [22]). Another is the *"interconnected core nodes structure"*, in which case the core nodes are interconnected with each other partially, as illustrated in figure 3.10 (quoted from [22]).

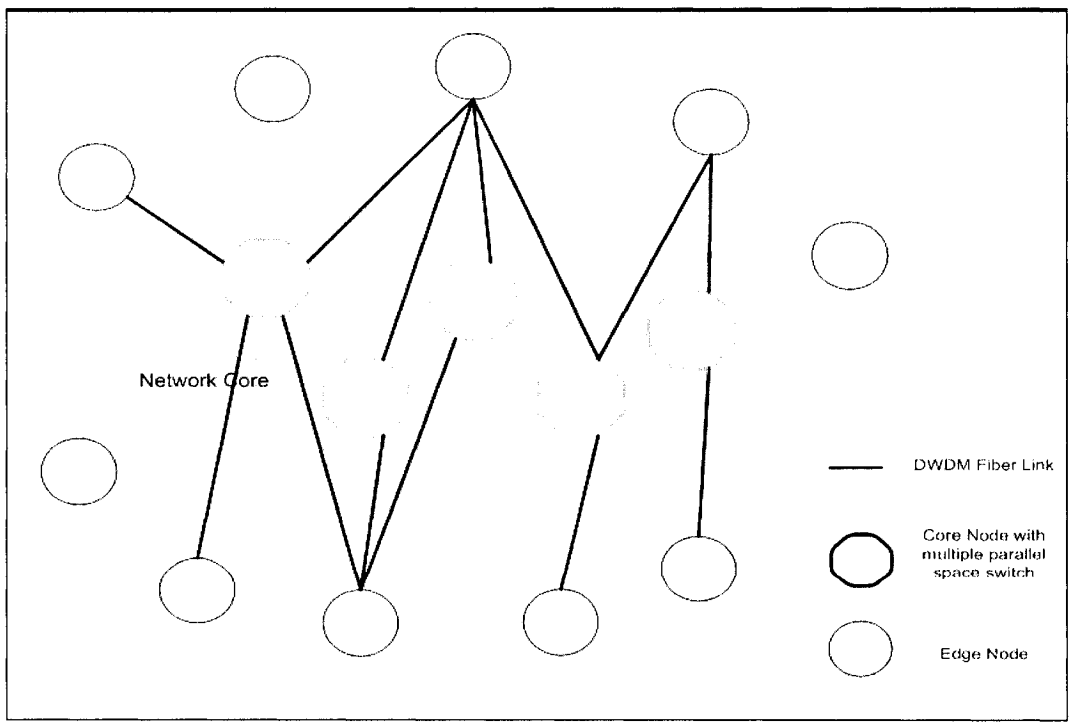


Figure 3.9 Core Structure with Independent Core Nodes

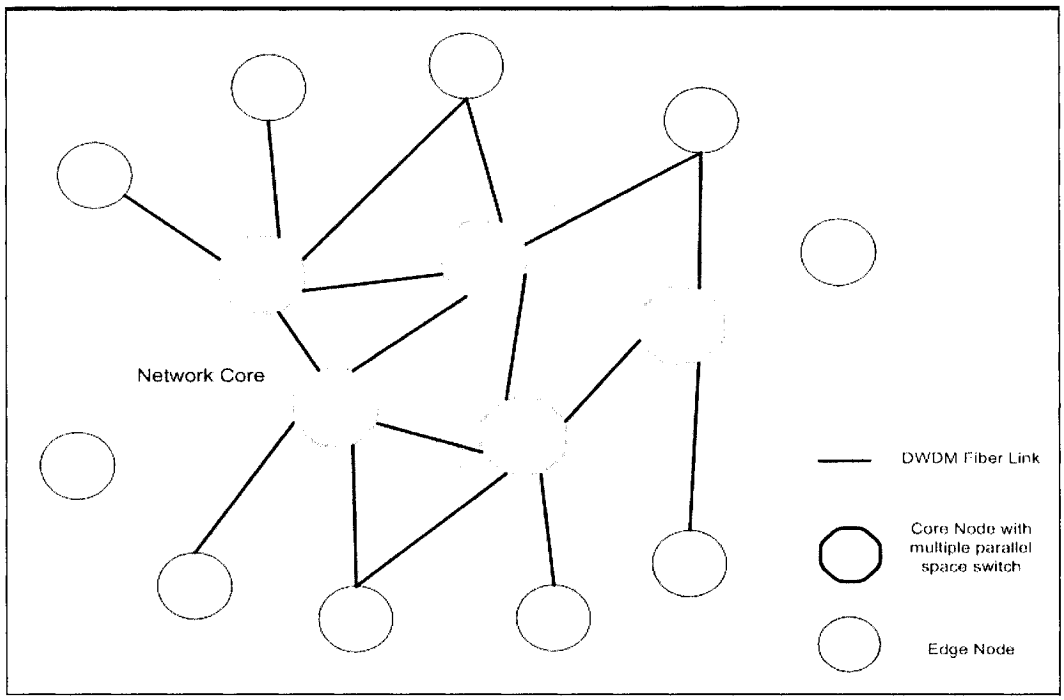


Figure 3.10 Core Structure with Interconnected Core Nodes

If the network core consists of interconnected core nodes, time coordination becomes extremely difficult; thus it is not a good solution for an agile core network (it is however suitable for static networks). Thus, in the PetaWeb network, in order to make time coordination feasible, network core with independent core nodes is required.

The capacity of the network core is limited by the number of the core nodes, thus limited by the capacity of each of the space switches and the number of channels in each DWDM fiber link.

3.3.4 Channel Reconfiguration Mechanism

Agility is a key feature of PetaWeb. The core nodes can adaptively reconfigure connectivity of the inter-module channels based on the traffic fluctuation at the source edge nodes. It is emphasized that the objective of reconfiguration is to maximize the proportion of the inter-module traffic that can be transmitted through a direct path without recourse to tandem path. Here a path consists of a number of channels; a direct channel between the

ingress edge nodes to egress edge nodes comprises a channel from the ingress node to a space switch in the core and a channel from the space switch to the egress node. A tandem channel between an ingress node and an egress node comprises two direct channels, one from the ingress node through a space switch in the core to an intermediate edge node, and one from the intermediate edge node through a space switch in the core to the egress node. The intermediate edge node is any edge node except the ingress and egress nodes. The traffic through the tandem path is called tandem traffic. The tandem traffic can be reduced significantly, or even eliminated, by channel reconfiguration in the core at specific time interval, yielding time-variant inter-module channel connectivity.

The core is reconfigured in response to reconfiguration requests made according to traffic pattern changes at the source edge nodes, which are sent from the ingress edge nodes. The steps of reconfiguration include:

- At ingress/source edge nodes, incoming traffic streams are monitored and measured constantly according to the egress/sink edge nodes; (the specific traffic estimator, which is used in our work, is described later in the simulation-modeling chapter.)

- Then the reconfiguration requests are sent to the particular core module. Requests may be sent from the ingress edge modules at any time;

- The core node reallocates the channel connectivity and sends connection-change requests to all participating nodes. However, the core nodes can only reconfigure in accordance with a predetermined schedule (according to the reconfiguration period). An ingress edge node may have two or more channel connections to an egress edge node. The connections between the ingress and the egress edge node are preferably routed through the same core node so that they have the same propagation delay from ingress to egress. (In this work, we focus on the network with single core node and evaluate the performance of the network with single core node; for the network with multiple core nodes, the network throughput and performance, and complexity for network survivability

increases considerably; however, some protocols need be considered such as routing protocol, a protocol for the maximization of channel. Also the complex mechanism for time-coordination of multiple core nodes and edge nodes need be exploited; these works can be done in the future.)

- Reconfiguration at the core nodes and edge nodes is performed according to instructions received from core node controllers.

In the PetaWeb, the edge nodes and core nodes are distributed over a wide area. The propagation delay between any two nodes could be different, thus it is important that the reconfiguration of the connectivity of space switches in the channel switching core is coordinated with corresponding switching processes at the edge nodes. Without precise time coordination between the edge nodes and the core nodes, some connections may be forced to an idle state for relatively long periods of time, of the order of milliseconds, to ensure that data is not lost during channel reconfiguration.

3.3.5 Time coordination in the PetaWeb

As mentioned above, time coordination is required in the PetaWeb. It may be realized by exchanging timing packets between the source nodes and core node. The time coordination of the single core switch uses a system of timing counters as illustrated in Figure 3.11 (quoted from [21]). The time counter circuit associated with the core controller includes a core clock and a time counter. A time counter circuit in the edge controller of each edge node includes an edge node clock and a time counter. The edge time counter and core time counter need have an identical period. The period of a time counter is at least equal to the sum of the largest propagation delay between any edge node and any core controller and the guard time, which is used to compensate for switch channel reconfiguration delay.

All clocks in the time counter circuits are synchronized using well-known techniques. American National Standards Institute (ANSI) and Telcordia standards [79], along with operational needs for quality, have dictated clear rules for optical network synchronization timing. Many providers have chosen a complete overlay network, or some combination of

overlay and use of traffic lines for inter-node synchronization distribution. These are made up of external Primary Reference Sources (PRS) and Building Integrated Timing Supplies or Synchronization Supply Units for intra-node distribution. The synchronization of long-haul networks is typically accomplished using an external synchronization source, such as global positioning system (GPS), and a hierarchical timing distribution system. Core switching sites are timed to a Stratum 1 PRS, and are synched to lower level switching sites and other nodes where there may be a Stratum 2 or 3 clocks.

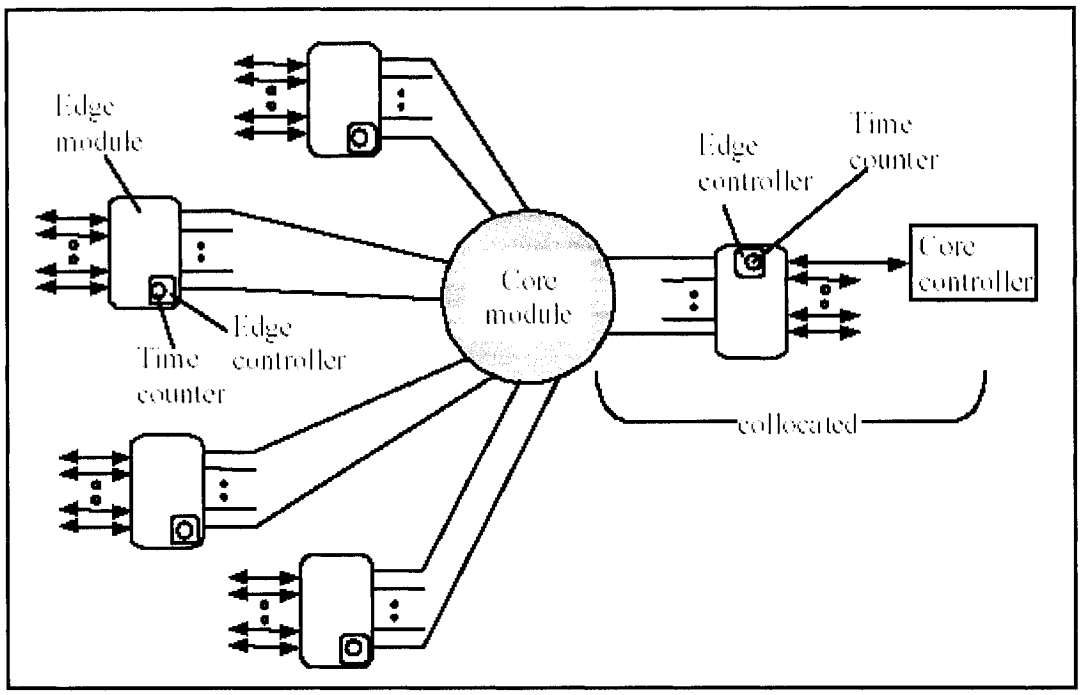


Figure 3.11 Time Coordination Mechanism in single core PetaWeb

Timing packets are exchanged between each edge time counter circuit and core time counter circuit. Each time the edge time counter reaches zero, the edge node sends a timing packet to the core controller. When receiving a timing packet, the core controller stamps the timing packet according to the reading of core time counter and sends the timing packet back to the correspond edge nodes. Upon receipt of the returned timing packets, the edge node extracts the time stamp information, and then uses it to adjust its time counter at an appropriate time.

Figure 3.12a illustrates timing packets exchanges between edge nodes and core controller. For example, one edge node sends a timing packet at time t_1, as indicated in the time axis. The core controller receives the timing at time t_2. The core controller inserts the value of time t_2 in the timing packet, and then sends the timing packet back to the edge node. The edge node controller extracts the values t_1 and t_2, and uses this information to adjust its time counter. Similarly, another edge node transmits its timing packet to the core controller at time t_1'. The timing packet is received at the core controller at time t_2'. The core controller stamps the timing packet and returns it to this edge node.

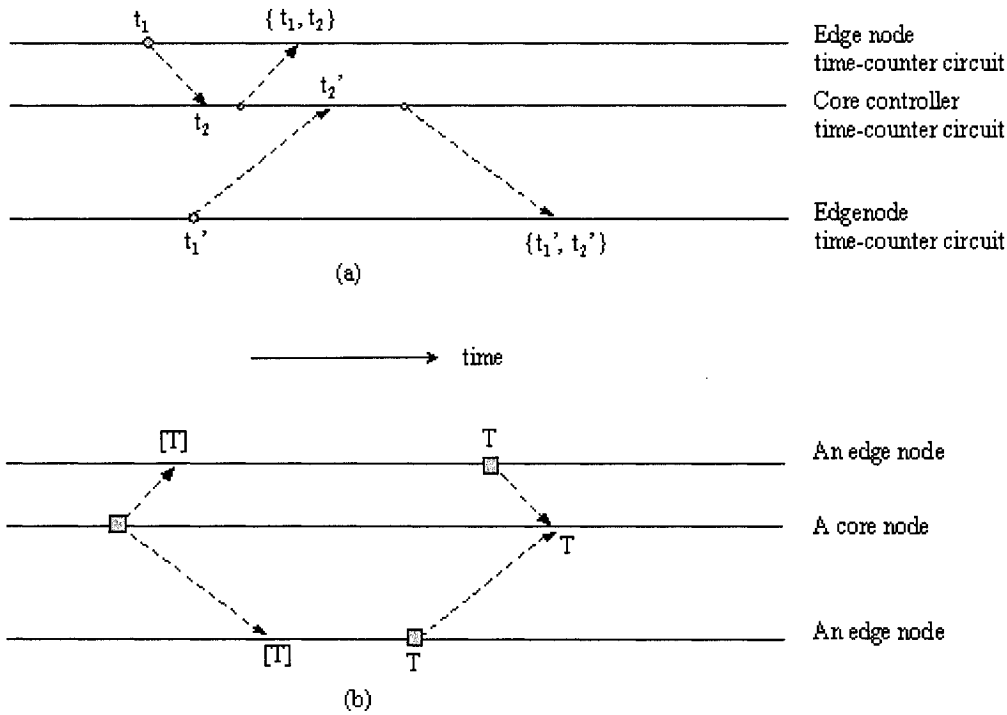


Figure 3.12 time packets exchanging and time coordination process

Figure 3.12b illustrates the time coordination process to enable paths to be reconfigured, necessitating changes in the core. The time coordination process requires that the core controller send the reconfiguration request packet simultaneously to all participating nodes. The reconfiguration request packet includes the desired reconfiguration time T, and channel allocation information. The local time in the first edge node time

counter and the local time T in the second edge node time counter are different due to the propagation delays to the core controller. At the local time T, edge nodes send data; the respective data from the edge nodes reach simultaneously the channel switch core at the core time T. By doing so, the data loss, due to channel reconfiguration, is avoided.

3.4 Summary

In this chapter, we first introduce some network architecture principles and network design principles, which are fundamentals in optical networks. Ideally, the network should have several features, such as minimum inner-to-outer capacity ratio, simplicity structure, high scalability, low traffic blocking probability, and self-governance. Secondly, we brief overviewed the PetaWeb network architecture. Its important components and functionalities are also presented. The PetaWeb network is designed as one kind of alternative network architectures having high network capacity up to several petabits per second; it is composite hub and stroke structure network having agile optical bufferless core nodes connected to large number of high-capacity electrical edge nodes. Agile core node can dynamically reconfigure edge-and-core channel connectivity in response to the traffic load fluctuation at the edge node. Channel allocation mechanism and time coordination mechanism between edge and core nodes are presented.

Chapter 4

QOS-AWARE PETAWEB NETWORK

4.1 QoS Overview

Forced by the demand to service QoS applications (e.g. mission-critical and multimedia applications), the Internet infrastructure is progressing gradually from the traditional best-effort service toward QoS enables architectures, the most commonly known as Integrated Services (IntServ) [3] or Differentiated Services (DiffServ) [4] [5]. Spearheaded by the Internet Engineering Task Force's (IETF) Working Groups [2], a considerable amount of work was performed over the past few years in this direction, producing a large number of service models and protocols.

The integrated service model proposes two types of services for the support of real time applications (e.g. voice, video, and interactive applications), namely guaranteed service and controlled load service. IntServ provides guaranteed QoS based on explicit resource reservations on a flow-by-flow basis. The Resource reSerVation Protocol (RSVP) [54] is used as the signaling protocol for applications to reserve resources. RSVP imposes per-flow state on every router in the network. The router has to maintain flow-related state information and handle the periodic refresh messages for each flow. However, this approach has scalability problems, especially in the network core, where routers need have the capability to manipulate a population flows in the order of millions.

On the other hand, DiffServ provides differentiated quality of service to aggregated flows. Packets are classified at the edges into different classes. Within the core of the network, the routers treat packets with a given class identically, irrespective of which flow

they belong to, and differentiate between different classes. This approach can scale well to a very large number of flows. DiffServ only provides a certain level of qualify of service, however, not strict guaranteed service for individual flows.

It is believed that in the next generation Internet, both architectures must be incorporated together to provide a broad spectrum of QoS supported capabilities. IntServ can be located at the edges of the network and closer to end user systems, while DiffServ can be deployed at the core [55]. Under this vision, it is advantageous to map DiffServ to the PetaWeb.

In this chapter, we address the issues of how to support the QoS in a PetaWeb network. In order to achieve end-to-end QoS over a PetaWeb network, the QoS-Aware edge node and QoS-Aware Channel Allocation Algorithm are considered. This increases the effective capacity of the network, reduces packet loss, and packet delay, satisfying the desired services. The rest of the chapter is organized as follows. In Section 2, we give a brief introduction to Differentiated Services (DS). Then, in Section 3, we discuss the QoS-Aware PetaWeb network model and QoS-Aware Channel Allocation Algorithm.

4.2 Differentiated Service

4.2.1 Differentiated Services Architecture Overview

DiffServ architecture achieves scalability by implementing complex traffic classification and traffic conditioning functions only at the edge nodes of the network, and by applying specific per-hop behaviors policies on the aggregated traffic at the core routers. The traffic handling policy applied on a specific packet is determined by the value of the DS field in the IPv4 or IPv6 headers [56]. These values are set at the edge routers. In IPv4, the type-of-service (TOS) byte is used to mark packets. DiffServ renames the TOS byte as DS field, of which, six bits make up the Differentiated Service CodePoint (DSCP) field, and the remaining two bits are not used. In DiffServ, some of per-hop behaviors are defined [57] [58]. Figure 4.1 shows the functionalities of nodes in a DiffServ network.

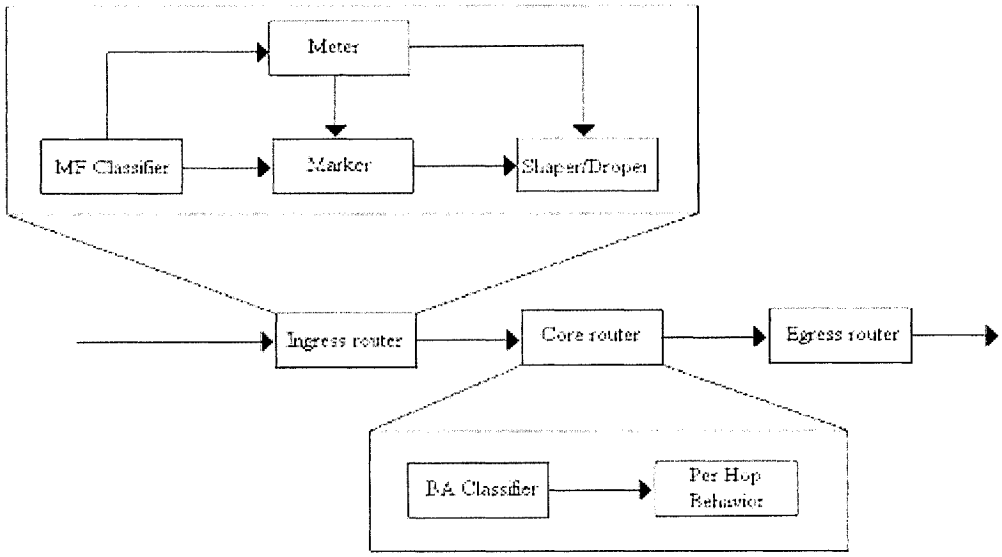


Figure 4.1 DiffServ Nodes Functionalities Diagram

As mentioned earlier, in the DiffServ architecture, packets are classified, policed, marked and possible shaped at the edge nodes (ingress router, egress router) of the DiffServ network. In contrast, simple operation such as behavior aggregate (BA) classification is implemented at the core of the DiffServ network.

The packet classification policy identifies the class of traffic, which will receive a differentiated service by being conditioned and/or mapped to one or more behavior aggregates within a DS domain. There are two types of classifiers in the DiffServ, one is Behavior Aggregate (BA) Classifier, which classifies packets based on the DS code point; another is Multi-Field (MF) classifier, which classifies packets based on the multiple fields on the packet header, such as source and destination IP addresses, source and destination port numbers, protocol ID, and the TOS octet. From the classification result, the traffic profiles and corresponding policing, marking and shaping can be performed on the traffic.

Traffic conditioning function includes meter, marker, shaper and dropper. A meter is used to measure the traffic stream against the traffic profile specified in the service level agreement, and determine whether a packet is in-profile or out-of-profile. This process is

also known as policing. Marker is to set the DSCPs of the packets. Shaper/dropper will delay/discard some of the packets in a traffic stream in order to bring the stream into compliance with a traffic profile. For example, for customers with assured service, policing normally is implemented with a token bucket so that some kind of burst is allowed. When a packet arrives, if there are tokens in the bucket, the packet is marked into in-profile packet; otherwise, the packet is marked into out-of-profile packet. Note that the in-profile packets and out-of-profile packets belong to the same traffic class, but with different drop precedence.

In addition to the traditional best effort (BE) service, DiffServ provides another two services: Premium service and Assured Forwarding service.

Premium service provides guarantees of low loss, low delay, low delay jitter, and desired peak rate service (as long as the user's traffic remains levels defined by the service agreement). Deviations from the agreement would result in dropping of the excess traffic. One example of this service is to create "virtual leased lines", saving the cost of building and maintaining a separate network [59]. This service also can be used for videoconference, voice over IP, Virtual Private Networks (VPN), and so on.

Assured service is based on statistical provisioning. It provides more reliable service than best effort. This service provides a relative QoS assurance, allows the Internet Service Provider (ISP) to offer different levels of forwarding assurances to certain customers. It is also called Olympic Service, which consists of gold, silver and bronze service classes. If desired, packets within each class may be further separated by giving them either low, medium, or high drop precedence.

4.2.2 Per Hop Behaviors

Differentiated Services introduces the notion of Per Hop Behaviors (PHBs). As described in [4], a per-hop behavior (PHB) is a description of the externally observable forwarding behavior of a DS node applied to a particular DS behavior aggregate. PHBs may be specified in terms of their resource (e.g., buffer, bandwidth) priority relative to other PHBs, or in terms of their relative observable traffic characteristics (e.g., delay, loss).

PHBs are implemented in nodes by means of some buffer management and packet-scheduling mechanisms. In DiffServ network nodes, a codepoint in the IP header is used to select a per-hop behavior (PHB) as the specific forwarding treatment for that packet [4] [5].

The IETF Working Group has standardized a small number of specific per-hop behaviors (PHBs), and recommended a particular bit pattern or "code-point" of the DS field for each one, including Expedited Forwarding (EF) PHB [57], Assured Forwarding (AF) PHB [58].

The EF PHB is used to provide low loss, low delay, and low delay jitter, assured bandwidth and end-to-end service through the DiffServ domain. This service has also been described as Premium service in [58]. Code point 101110 is recommended for the EF PHB.

The AF PHB is used to provide different levels of forwarding assurances for IP packets with low loss rate, but without specified delay and delay jitter. The AF PHB group provides delivery of IP packets in four independently forwarded AF classes. Within each AF class, an IP packet can be assigned one of three different levels of drop precedence. Recommended codepoints for the four general use AF classes are given below.

Drop Precedence	*Class 1*	*Class 2*	*Class 3*	*Class 4*
Low drop precedence	001010	010010	011010	100010
Medium drop precedence	001100	010100	011100	100100
High drop precedence	001110	0101110	011110	100110

4.3 QoS-Aware PetaWeb Network

As the PetaWeb will have to service IP traffic, it is important that the PetaWeb is integrated into Internet, and becomes capable of supporting QoS as well becomes compatible with the remaining Internet architecture. As we know, Differentiated Service has good scalability, and can be deployed at the core. However, a problem caused by the uneven distribution of premium or assured traffic exists in the DiffServ network. As no policing occurs in the core routers, traffic fluctuations may occur. This may cause performance degradation to the streams services through the premium or assured class. DiffServ cannot avoid this problem by itself. Fortunately, the PetaWeb has the capability to accommodate traffic fluctuation, which could very well be treated as a mechanism to diffuse the effect of these unwanted significant traffic fluctuations. This, making the PetaWeb DiffServ-capable could be used as a measure to countermeasure this problem. We are proposing below an architecture making the PetaWeb have the quality of service capability. We call it QoS-Aware PetaWeb network.

In the QoS-Aware PetaWeb network, we assume that each edge node of the PetaWeb is connected with a number of DiffServ core routers or DS edge routers, and the PetaWeb is a backbone network interconnecting several DiffServ domains or a core network within one DiffServ Domain. Therefore, the incoming traffic streams at the PetaWeb edge node are highly aggregated traffic with different PHBs (EF, AF, BE). As mentioned earlier, the traffic volume may fluctuate with time even though there are conditioned at the DiffServ edge nodes. QoS-Aware PetaWeb Network is illustrated in Figure 4.2. In order to make the PetaWeb support quality of service, some parts of the PetaWeb need be enhanced. The QoS-Aware PetaWeb edge node and QoS-Aware Channel Allocation Algorithm are proposed; they are described in the following section.

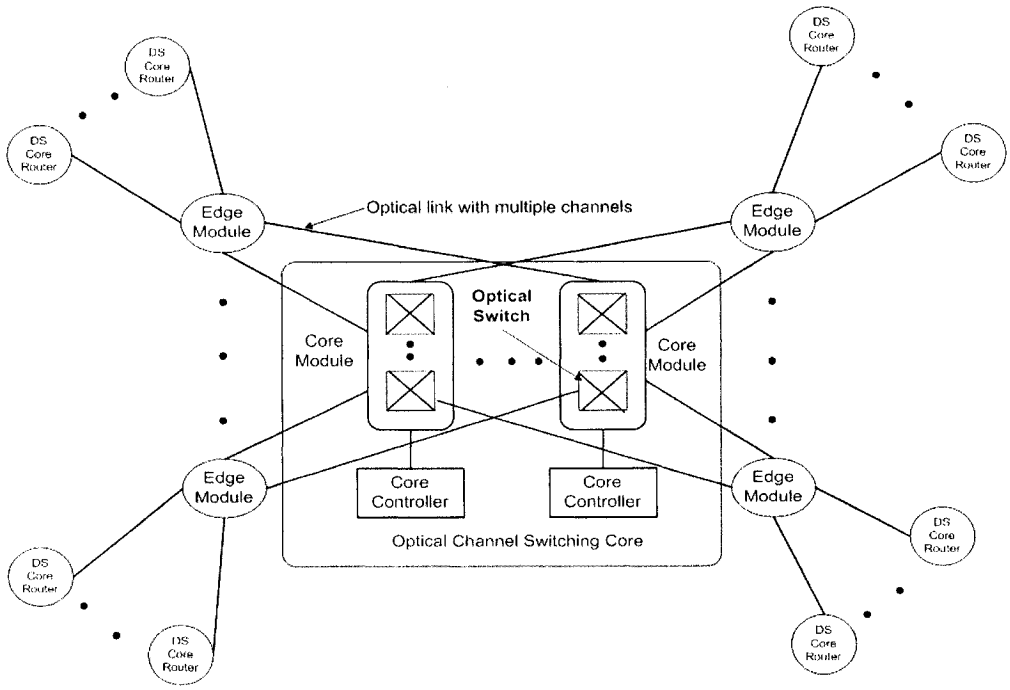


Figure 4.2 QoS-Aware PetaWeb Network Architecture

4.3.1 QoS-Aware PetaWeb edge node

In the QoS-Aware PetaWeb network, at the PetaWeb edge nodes the incoming packets are aggregated according to the corresponding PHBs (such as EF, AF, and BE), and are placed into different queues. Then these aggregated traffic rates are estimated individually. Specified queuing scheduling disciplines such as Priority Queuing (PQ) and Weighted Round Robin (WRR) is used to manage the access of the allocated channels, and as a result, to support the differentiated service classes. The relative computing simplicity of both queue-scheduling algorithms allows them to be implemented in hardware, which is necessary in order to meet the processing speeds in the core of the network.

4.3.2 QoS-Aware channel allocation algorithm

In the PetaWeb, there exists relatively small amount of tandem traffic. Packets transmitted through tandem paths [22], are encountering larger end-to-end delay and delay jitter, therefore the transmission of real-time application and delay sensitive packets through the tandem paths should be avoided as much as possible. In order to solve this problem, we propose a QoS-Aware Channel Allocation Algorithm (QACAA), which is

located at the PetaWeb core node controller. The QoS-Aware channel reallocation includes the following steps:

- The incoming traffic streams are aggregated at the PetaWeb edge nodes according to different PHBs; arrival rates are estimated, and the channel capacity demand is calculated respectively;
- The different PHBs' channel capacity demands are sent to the core node controller;
- During the reconfiguration, the channel allocation algorithm first satisfies the EF traffic request and allocates the direct channels to them;
- After that is performed, the algorithm allocates the rest of free channels to AF traffic;
- Finally allocates free channels to BE traffic.

In order to satisfy the desired services, the PQ, WRR, or even Adaptive WRR (detailed see [60] [61]) scheduling disciplines could be used at the PetaWeb edge node. Through these schemes, the PetaWeb network can support differentiated services, and provide desired end-to-end quality of services.

4.4 Summary

In this chapter, we first overview and compare IntServ and DiffServ, two kinds of Internet service models having capability to provide quality of service. Differentiated Service architecture, which is based on a simple, scalable model, is suitable to be deployed at the network core. We then introduce an extension of the DiffServ architecture to PetaWeb. DiffServ employs traffic classification and conditioning to shape and police the packet traffic. Packets are forwarded according to the per-hop behavior. Assured forwarding and expedite forwarding are provided in DiffServ. Improving the PetaWeb edge node to incorporate QoS-Aware capabilities and using a QoS-Aware channel allocation algorithm at the core controller, the PetaWeb network develops the capability to support differentiated services.

Chapter 5

PETAWEB MODELING

5.1 Introduction

"Simulation is the imitation of the operation of the real-world process or system over time." [62]. Simulation leads to improvements in cost reduction and shortens the research & development cycle and the ability to incorporate arbitrary levels of details. Simulation modeling is desirable, when introducing new network architecture such as the PetaWeb or the network communication protocols. The simulation model is used to quantify the network performance and verify the functionality of the network.

In the present work, we implemented the simulation model of the PetaWeb architecture, using the OPtimized Network Engineering Tool (OPNET) [63]. Our work is funded by the National Capital Institute for Telecommunications (NCIT) [64] and is conducted in co-operation with Nortel Networks [65]. The simulator allows us to test the functionality of the PetaWeb network under a variety of traffic loading conditions. Our simulator implements the main components of the network: the core module, the edge module, the edge to core communication protocol and the channel allocation algorithm. The first objective of modeling is to demonstrate that the PetaWeb network and its components can work correctly.

In PetaWeb, most of the incoming traffic can be transmitted through a direct path to the egress node without the wavelength conversion, due to the PetaWeb's ability to dynamically reconfigure the light paths based on the ingress edge node traffic demand. As result, it reduces the packet loss, packet delay and delay jitter. These features are important to several types of applications. Furthermore, the PetaWeb can accommodate unexpected traffic; it makes the PetaWeb be resilient to the spatial and temporal traffic variations. It simplifies the traffic engineering, and reduces the end-to-end network provisioning. In

order to quantify the benefits of the agility, the simulation model is needed. However, the benefits vary with the nature and values of various specific factors such as input traffic model and certain engineerable parameters of the PetaWeb. Through the analysis, we can gain the insight on how to set these parameters in order to obtain a desired level of performance.

The structure of this section is as follows: Section 5.2 discuses the factors impacting the performance of the PetaWeb. In Section 5.3, several traffic source models such as sinusoidal, alpha-stable used in the study are introduced. The simulation models of generic PetaWeb Network and QoS-Aware PetaWeb Network developed for this work are described respectively in Section 5.4 and Section 5.5.

5.2 The factors impacting the performance of the PetaWeb

As mentioned before, agility is the key feature of the PetaWeb Network. Ideally, the PetaWeb has the capability to distribute the network resources according to the edge node demands. There are some physical constrains and practical constrains regarding reconfiguration times. Optical switch switching time and propagation time of control messages between edge and core controllers are elements in the list of constraint parameters. The PetaWeb performance gains as compared to the static core network, rely on several factors (see figure 5.1, quoted from [22]), including the following: the reconfiguration period, the number of channels per optical link and the granularity of the optical channel, the edge node buffer size, the propagation time between ingress edge-to-core and core-to-egress edge, the traffic pattern and traffic variation, etc. The effect of the traffic characteristics will be discussed in later, the other factors are described below.

The agility of the PetaWeb network is mainly determined by the reconfiguration period. The shorter the reconfiguration period, the more accurately the PetaWeb network can accommodate the traffic fluctuation; as result, the higher the network resource utilization becomes. However, the time reconfiguration period is limited by the longest round-trip propagation time between the edge nodes to the core node. How to select and set the reconfiguration period in order to achieve satisfactory performance needs to be

investigated.

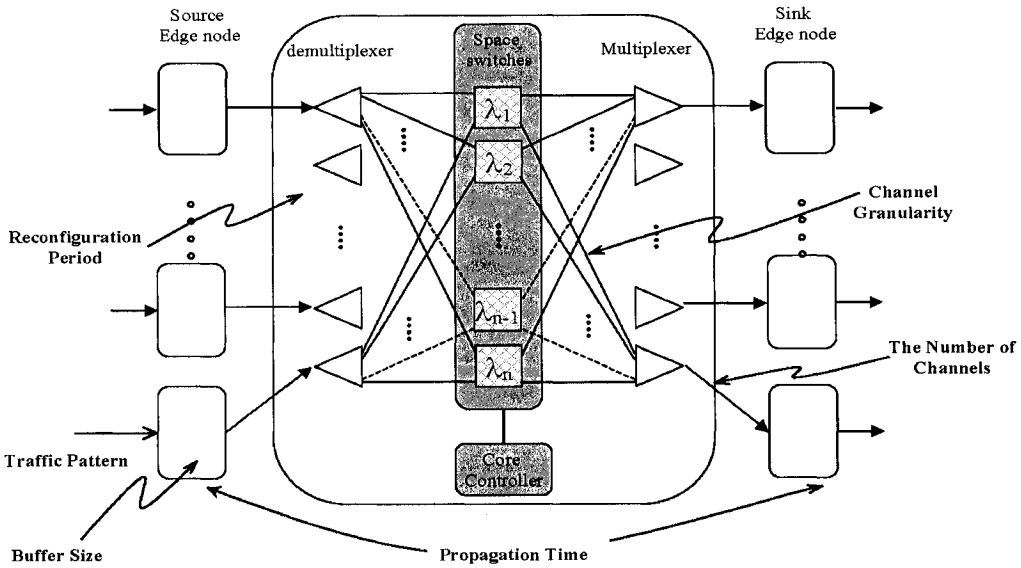


Figure 5.1 Factors impacting PetaWeb Performance

In the PetaWeb network, the number of channels per fiber link and the granularity of the optical channel between edge nodes and core nodes are one of the factors, which can affect the degree of PetaWeb agility. They determine the ability of the PetaWeb reaction to the traffic fluctuations. The finer the granularity, the more agile the PetaWeb becomes, because it can allocate more precisely the required bandwidth on the optical channels without having to go through tandem paths. This decreases the tandem traffic. At the time of the writing of this dissertation, systems carrying 160 channels of 10Gb/s or 40Gb/s each on a single fiber have been announced.

In the PetaWeb network, the optical core is bufferless, however the edge nodes have some buffer in order to avoid data loses at the edge. However, too much buffering can cause high delay and delay jitter; keeping the buffer size reasonable is important. We are interested in this work to find what the buffer size should be in order to obtain satisfactory performance. In other words, the network should not only provide higher throughput, lower tandem traffic, but also maintain the queuing delay under certain bound.

The propagation time is the time needed for the traffic transmitted from ingress edge node to reach the egress edge node. It impacts the agility and the performance of the PetaWeb in two ways. First, the reconfiguration period cannot be shorter than the longest round-trip propagation time between the edge nodes and the core node, as mentioned above. Second, the core controller decides the new channel allocation based on slightly outdated traffic demand information. The longer the transmission delay, the more outdated the information becomes, and the less accurate the channel reconfiguration. The effects of propagation time on the PetaWeb should remain small overall.

5.3 Traffic Source Models

Due to the Internet's rapid expansion and the appearance of new applications, it is quite hard to foresee the Internet traffic patterns to appear in the future; those patterns that will be passing through the next generation Internet network. In the present work, we will attempt an evaluation based on existing models, in several cases extrapolated to the future through logic and use of common sense. However, by no means we claim that our analysis covers with accuracy the traffic patterns of the feature Internet.

The Poisson model has been used extensively in the past in the performance evaluation of computer networks, primarily because it allows reaching analytical solutions of reasonable complexity. However, studies carried out during the last decade (continuing today) demonstrated that the traffic passing through modern networks behaves quite different from Poisson, to the point that results based on Poisson analysis were quite inaccurate. Specifically, the traffic traversing through modern computer networks demonstrates high degree of self-similar behavior, detailed see [66].

Traffic traces measured on backbone links reveal that over longer periods, there is considerable variability of traffic volumes, depending on the time of the day (morning versus night, weekdays versus weekend). Traffic volumes are significantly higher during the daytime than it is during the night. Self-similarity and daily patterns are two important characteristics of today's Internet traffic. However, we stress again that today's traffic patterns might not be as relevant tomorrow. Our changing habits, higher dependence to automation and the increasing importance of the individual private user in

the world of telecommunications might change the traffic patterns drastically. As an example the reader should consider the development and deployment of high quality video-on-demand services to residential users. The high traffic volume of such applications will increase drastically the levels of traffic during night hours. This unpredictability of future traffic is an additional motivation for deploying agile networks such as PetaWeb. Our ability to adapt to the unknown will be higher.

In order to investigate the effect of traffic on the behavior of the protocols and the performance of the network, our approach was to use a range of traffic models to gain more insight, rather than to rely on a single one. In this work, we apply several traffic models such as Poisson, sinusoidal, alpha-stable and combinations of them to evaluate the network performance. The sinusoidal traffic model is used to model the macroscopic traffic variations. The Poisson and alpha-stable traffic models are used to represent the microscopic behavior.

5.3.1 Sinusoidal Traffic Model

In this model, the input traffic from an ingress edge node to an egress edge node follows the sinusoidal pattern with period *T*. However, at each ingress edge node, the input traffic pattern is shifted on the time axis. Figure 5.2 shows the sinusoidal traffic patterns, which are used in our simulation work, with 120 seconds periodicity. There are 5 edge nodes totally, 4 acting as ingress (source) and / or egress (sink) nodes (depending on the evaluated scenario), one acting as sink node only. Because of limitations in our simulation, we are using a 2 minutes period for the sinusoidal traffic. The traffic pattern at a node has a shifted 30 seconds from the previous and next node.

Although simple and deterministic, this model was useful in the early stages of the study to test the functionalities and algorithms of the PetaWeb, as it forces the re-allocation of optical channels from link to link.

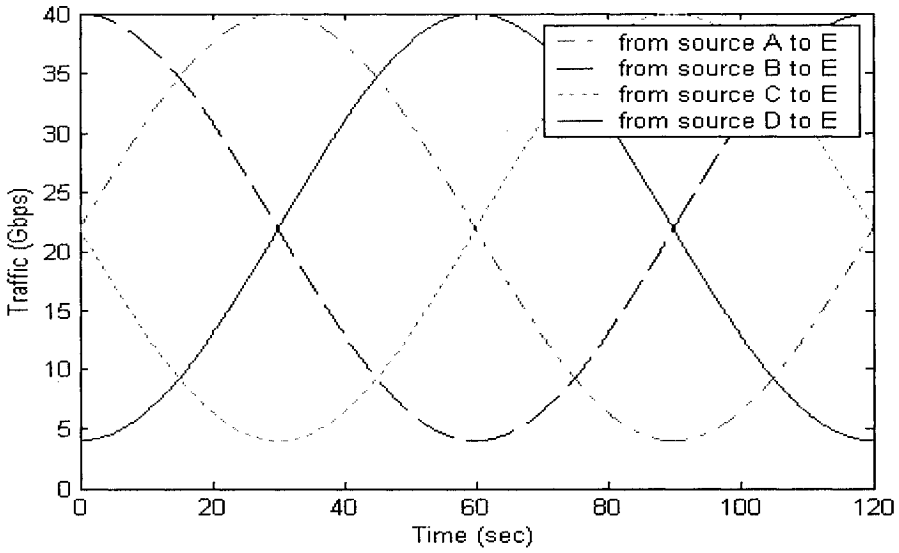


Figure 5.2 Sinusoidal Traffic Pattern

5.3.2 Alpha-stable traffic model

It is well known that highly aggregated Ethernet, variable-bit-rate video, FTP, and WWW traffics demonstrate self-similar behaviour. Self-similarity manifests itself in several aspects. One is the implication of long-range dependence. This means that traffic intensity remains correlated with values in a distant past. Another feature of a self-similar process is the presence of burstiness at all time scales. A lot of traffic models have been proposed, which can reflect the real network traffic's statistical behavior. Some of the models described in the papers by Tsybakov and Georganas [67], by Willinger [68], by Norros [69], and by José R. Gallardo [70].

In our simulation model, we use the alpha-stable traffic model, which was proposed and implemented by José R. Gallardo, Dimitrios Makrakis and Luis Orozco-Barbosa (detailed see [70]). This traffic model not only reflects the self-similarity and long-range dependence characteristics of aggregated traffic, but also enables to capture the appropriate level of burstiness of different types of traffic. The author has proved that aggregate traffic traces that present very high burstiness can be modeled accurately with self-similar processes by using alpha-stable distributions. According to [70], each traffic process has to

be modeled as a function of a fractional stable noise process Y_j. The traffic model was defined as follows:

$$W_j = m + a \cdot Y_j \qquad (5.1)$$

where W_j represents the actual number of bytes arriving during the j-th time interval of unit length. The parameter m is the mean value of the number of arrivals per unit time, a is a scale factor, Y_j is a fractional stable noise (FSN) process with index of stability α and Hurst parameter H.

A slightly modified version of the previous model was proposed for some traffic exhibit noticeable positive spikes, whereas the negative spikes are less pronounced (such as Ethernet and WWW traffic):

$$W_j = m + a \cdot \left(|Y_j| - \mu_Y \right) \qquad (5.2)$$

where μ_Y is the mean value of $|Y_j|$, m, a and Y_j have the same meaning as in Equation (5.1).

The traffic self-similarity coefficient H, as well as the parameters of the stable distribution, is estimated through analyzing different types of self-similar traffic traces. Figure 5.3 (quoted from [70]) shows an example of a traffic profile generated by the alpha-stable traffic model with the Hurst parameter $H = 0.903$, where α is equal to 1.94.

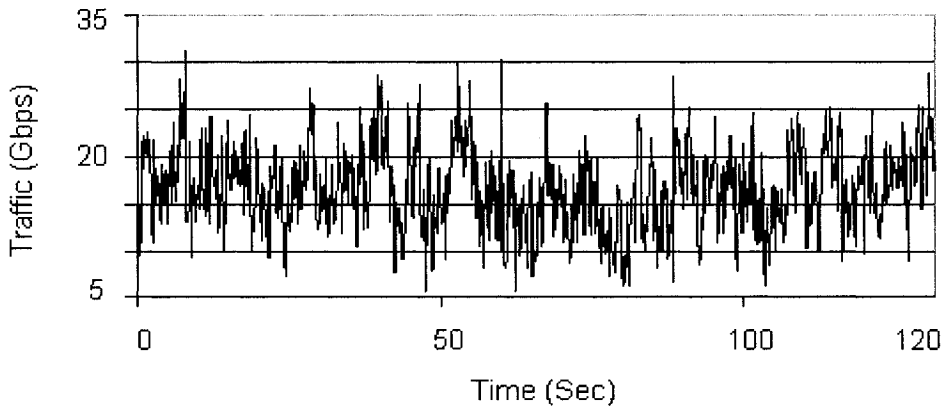


Figure 5.3 Self-similar traffic generated by α-stable traffic model

In this work, the traffic models, used in the experiments, are the self-similar traffic generated by the alpha-stable traffic model combined with the sinusoidal traffic described above. We use these two traffic models as the input source model at each of the source edge nodes. The traffic pattern of the input source at the edge nodes is depicted in figure 5.4 as follows.

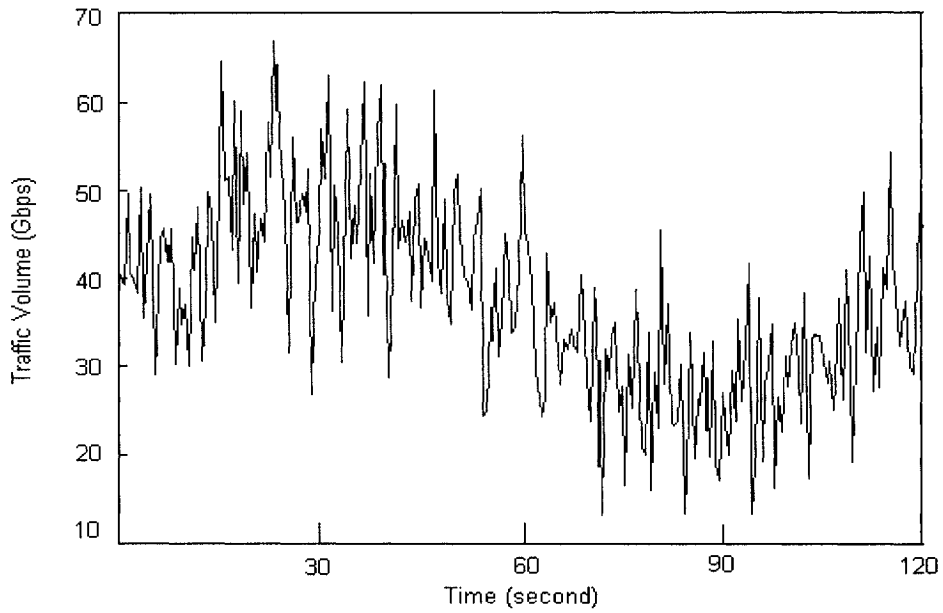


Figure 5.4 Traffic Pattern at the Source Node

5.4 The Generic PetaWeb Network Model

5.4.1 Introduction

In the section, we present the developed model of the generic PetaWeb network used for our simulations to verify the functionality of the PetaWeb and analyze the performance of the PetaWeb.

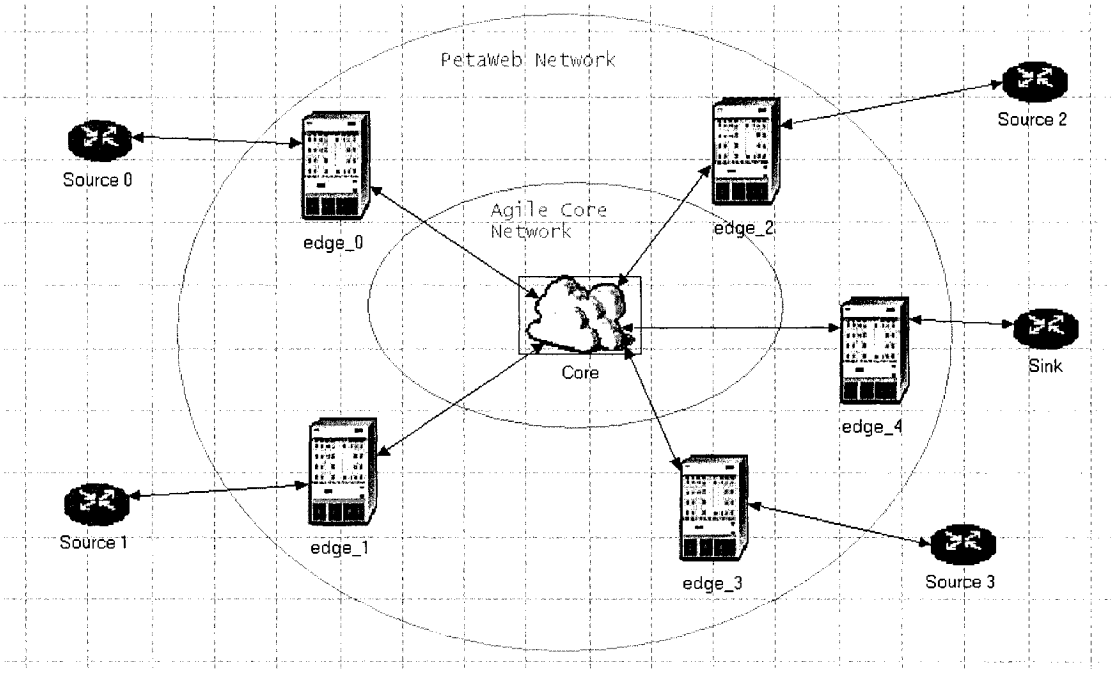


Figure 5.5 PetaWeb Simulation Model

The PetaWeb model is implemented using OPNET 8.0 depicted in Figure 5.5. It includes the main logic components of the network: the core module, the edge module, the edge to core communication protocol and the channel allocation algorithm. It consists of five edge nodes and one optical core network, which may have several independent core nodes. The optical core node could be static or agile. Also these nodes are interconnected by multi-channel optical fibers. For simplicity in our simulation and analysis, we use a single optical core node. We also assume that these edge nodes are distributed evenly around the core node, over a wide geographical area, forming a global transport network. The propagation delay is 50ms from any edge to the core, and 100ms from an edge to any other edge. The optical link, which connects each edge node to core node, has total 16 optical channels for data transmission and one control channel. Each optical channel has a capacity λ = 10 Gbps.

5.4.2 Edge Node

The edge node model (see Figure 5.5) we implemented consists of several components, which includes incoming traffic process model, traffic estimator, and the edge controller. Figure 5.5 illustrates the logic diagram of the edge node.

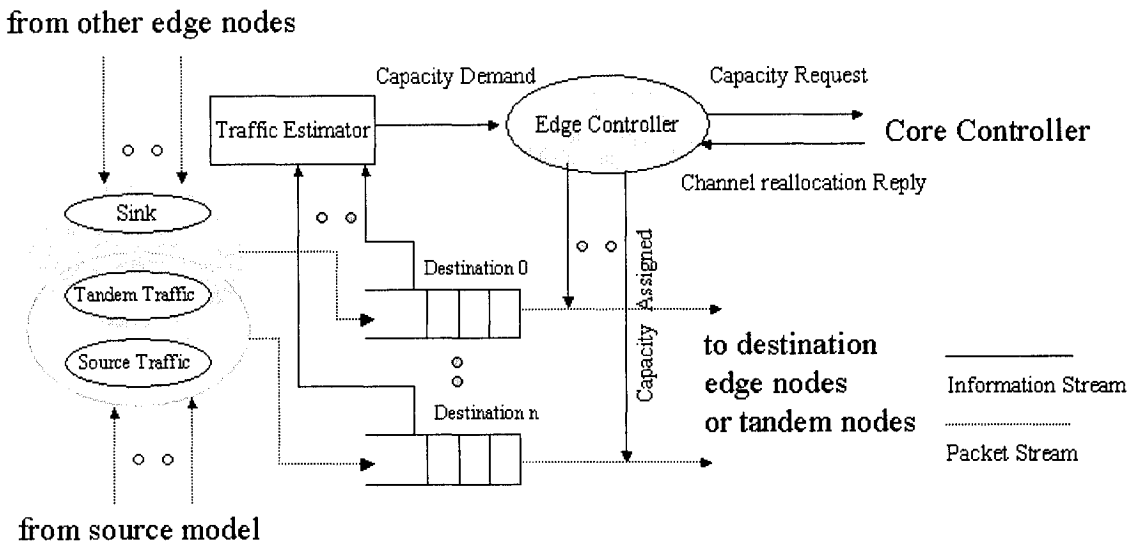


Figure 5.6 Edge Node Components Diagram

At the edge nodes, there are two kinds of input traffic. One of them is input traffic coming into the optical domain from the electrical peripheral electrical networks, generated by the input source in our simulator, the other arriving from other edge nodes (tandem traffic). The packets are sorted and inserted into individual buffers according to their destination. In the case of traffic arriving from other edge nodes, if the incoming packet destination is the local node, the packet is absorbed and relative parameters are recorded; otherwise, the packet is inserted into a buffer according to its destination. In this dissertation, we call the traffic generated by the source traffic and tandem traffic as incoming source traffic. The packets are removed from the destination buffers according to the assigned transfer rate.

5.4.2.1 Traffic estimator

At the edge node, the incoming source traffic streams are aggregated according to their destination address. Traffic estimator is used to estimate the aggregated traffic rate at the edge node according to the corresponding destination; the estimated value determines the capacity demand from a certain edge node to a certain destination node.

We use Time Slide Window (TSW) algorithm [71][72] to meter the traffic in our simulation model. The TSW algorithm is a rate estimator algorithm. TSW provides an estimate of the running average rate over a specific period of time. It takes into account burstiness and smoothes out the estimate to approximate the traffic stream. The principle of TSW is extremely simple. TSW only maintains three state variables:

- Time_Win: is measured in specific units of time;
- Avg_rate: the rate estimate upon each packet arrival;
- T_front: is the time of last packet arrival.

We use TSW to estimate the average rate upon each packet arrival, state variables Avg_rate and T_front are updated when a packet arrives; however, Time_Win is predefined but changeable. The TSW algorithm of the traffic rate estimator from [71] is described below.

Initialization:

Time_Win = a constant;

Avg_rate = initial traffic rate;

T_front = 0;

Upon each packet arrival, TSW updates its state variables as follows:

Bits_in_TSW = Avg_rate * Time_Win;

New_bits = Bits_in_TSW + pk_size;

Avg_rate = New_bits / (arrival_time – T_front + Time_Win);

T_front = arrival_time;

Where, arrival_time is the time of current packet arrival and pk_size is the packet size of the arriving packet.

The estimation value is changed into the capacity demand, which is preferably represented as a fraction of a capacity of a channel.

5.4.2.2 edge controller

The edge controller communicates the capacity requests of its edge node to the core controller. It is also responsible for path management operations, such as path setup and path capacity modification, and then acts according to available capacity assigned by the channel allocation algorithm at the core controller.

The capacity demands are received and placed into a capacity demand array according to the destination. The edge controller updates the capacity demand array when a new capacity demand is received. At the predetermined time interval, such as 5 milliseconds, the edge controller transmits the capacity request packet to the core controller through the control channel.

The channel allocation reply packet is sent back to the edge controller from the core controller after the channel allocation. It includes the capacity assigned to each destination with specific channel and switching reconfiguration time.

At the reconfiguration time, the edge controller controls the transmission rate for each destination according to the newly assigned capacity.

5.4.3 Core Node

In the simulation model, in order to evaluate the benefits of the PetaWeb, we set two scenarios for the core node. One is the static core node scenario; another is agile core node scenario.

In the static core node scenario, the inter-nodal channel connectivity is fixed; core node does not reconfigure the channels in order to respond to the traffic fluctuations at the edge nodes. The edge nodes transmit traffic according to predetermined capacity with each other.

In the agile core node scenario, the inter-nodal channel connectivity is changeable, responding to the traffic changes at the edge nodes. According to the functionality of the core node we mentioned before, the core node model (see Figure 5.5) we implemented here consists of several components, which includes, the core controller, the space switches. Figure 5.7 illustrates the block diagram of core node.

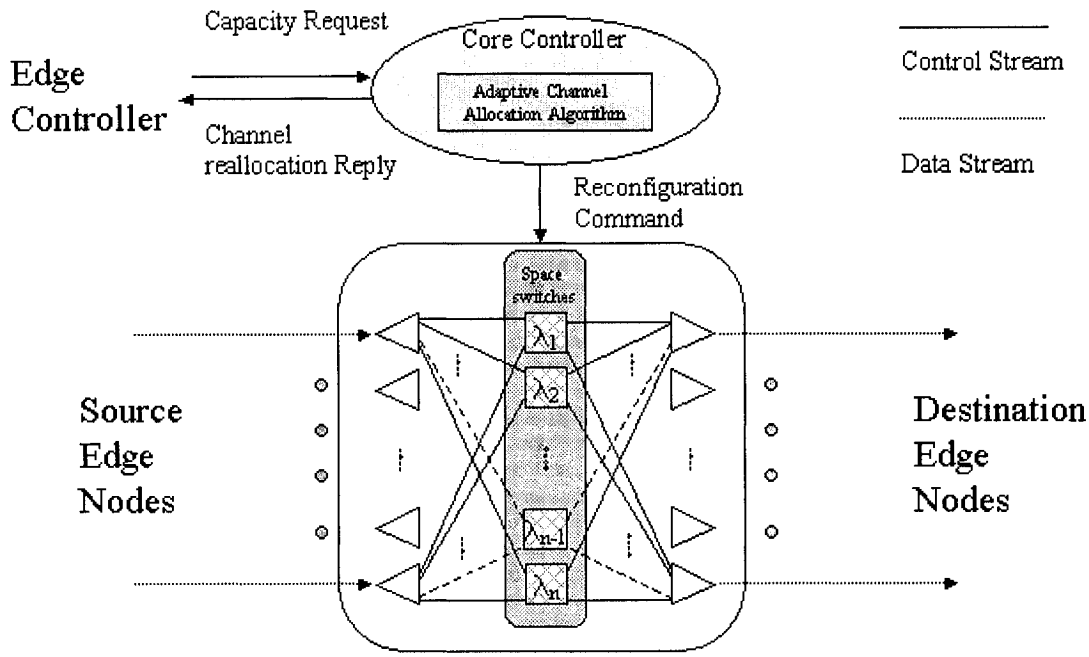


Figure 5.7 Core Node Components Diagram

5.4.3.1 Core Controller

Core controller determines network configurations according to the capacity requests from the edge controllers, communicates channel reallocation replies to edge controllers, and controls the space switches according to the reconfiguration commands.

Aggregated traffic streams are estimated at the source nodes according to the destination address, and the capacity requests are sent to the core controller (Figure 5.7) from the edge nodes. The capacity requests are placed in a reconfiguration request table shown in Figure 5.8. Each entry in the reconfiguration request table includes three fields: a source node identifier, a destination node identifier, and a capacity demand. The capacity demand is preferably represented as a fraction of a capacity of a channel. The

reconfiguration request table is updated when the core controller receives new capacity requests. The request table is sorted in descending order according to the capacity demand.

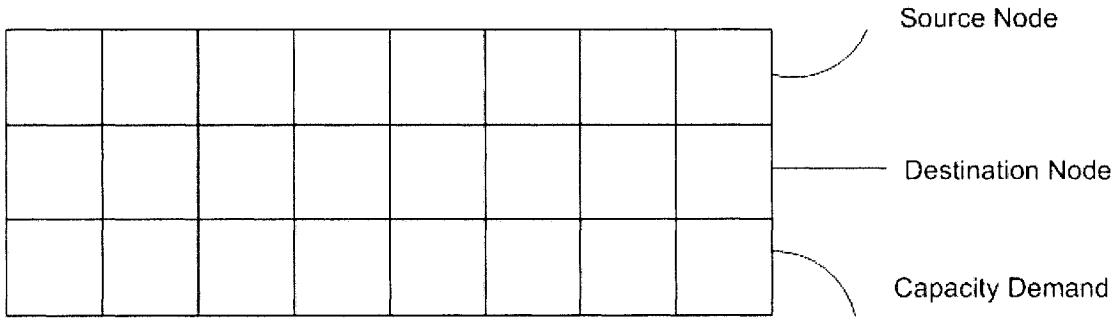


Figure 5.8 Reconfiguration Request table

After channel reconfiguration, the results of the channel allocation are placed into several tables (such as assigned integer channel table, assigned decimal channel table, and assigned tandem channel table, illustrated in figures 5.9, 5.10, and 5.11 respectively), and then are sent to the corresponding nodes for switch reconfiguration.

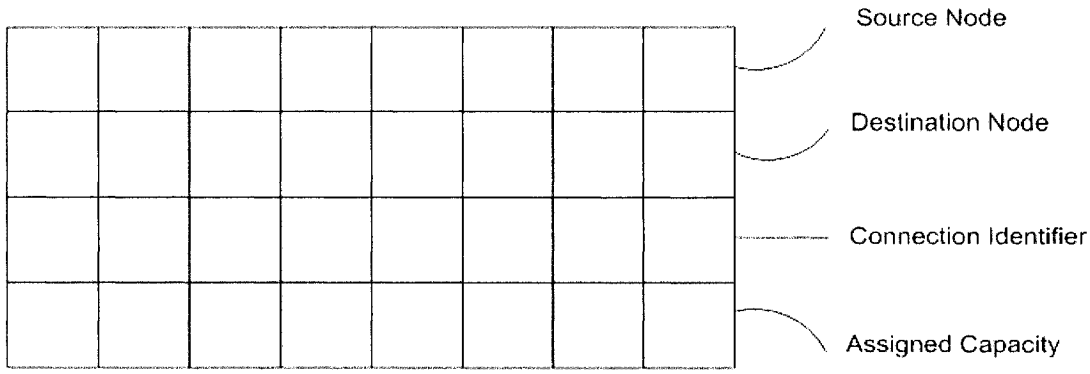


Figure 5.9 Assigned integer channel table

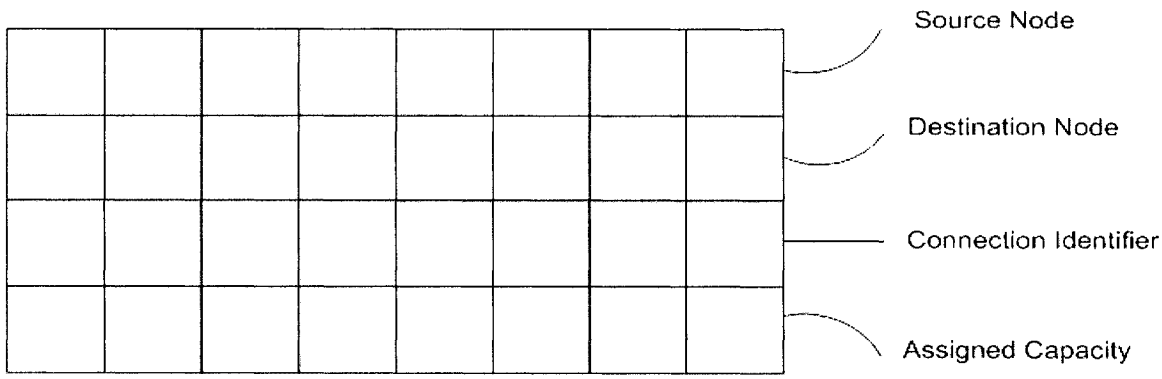


Figure 5.10 Assigned decimal channel table

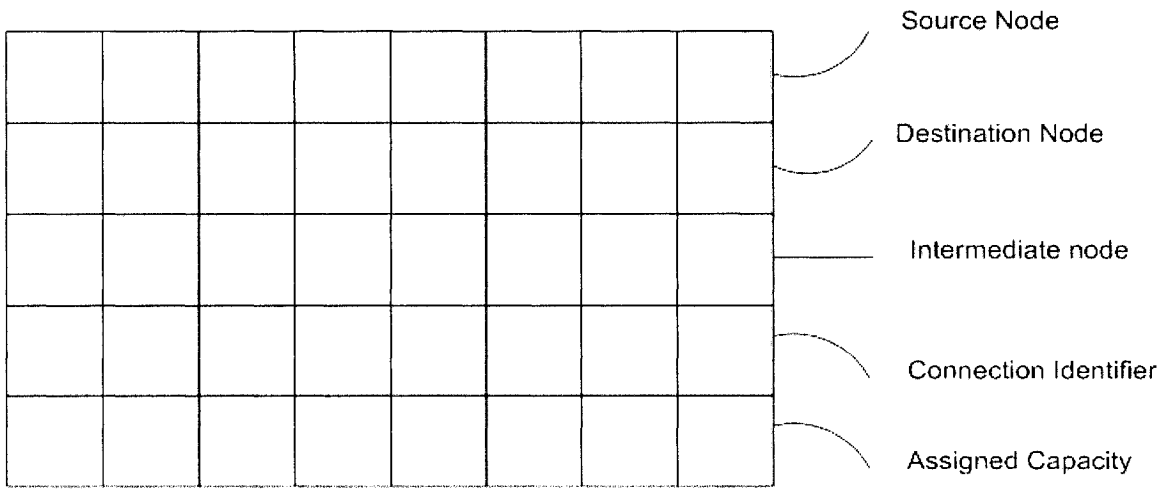


Figure 5.11 Assigned tandem channel table

In the agile core case, optical channels are re-allocated at predetermined time intervals, so that the number of channels from the source node to each destination edge node varies. It is even possible to have all channels allocated to a single source node, if the traffic warrants such an allocation.

In the static core case, the inter-nodal channel connectivity is fixed (time-invariant). Each edge module has its 16 channels connected permanently in a predefined configuration. However, in order to compare the performance of static core networks with the agile core networks, we implement two static cores, one is the static core without tandem path, another is the static core with tandem path. In the later case, the tandem paths are allocated for each edge nodes in responding to the traffic demands the same time intervals as the agile core case.

5.4.3.2 Adaptive Channel Allocation Algorithm

The channel allocation algorithm is the important part in the core controller. In order to accommodate the fluctuating traffic, an adaptive channel allocation algorithm is implemented inside the core controller with reference to Figure 5.7.

In the channel allocation process, in order to reduce the optical switch reconfiguration time, we need to minimize the number of channel connectivity changes between any edge

node and the core node; furthermore, we need to maximize the utilization of the channel capacities and minimize the number of packet loss, so that tandem paths are built for desired nodes. The detailed channel allocation algorithm is depicted as figure 5.12.

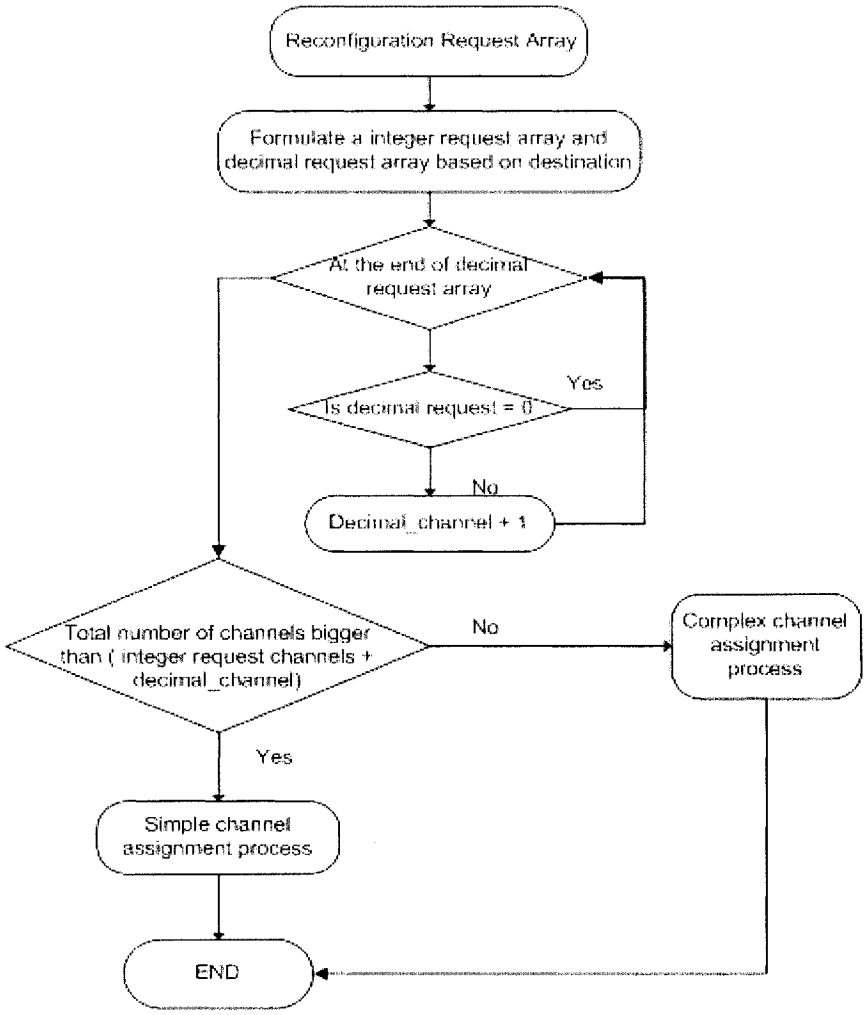


Figure 5.12 Channel Allocation Algorithm Flow-Chart Diagram

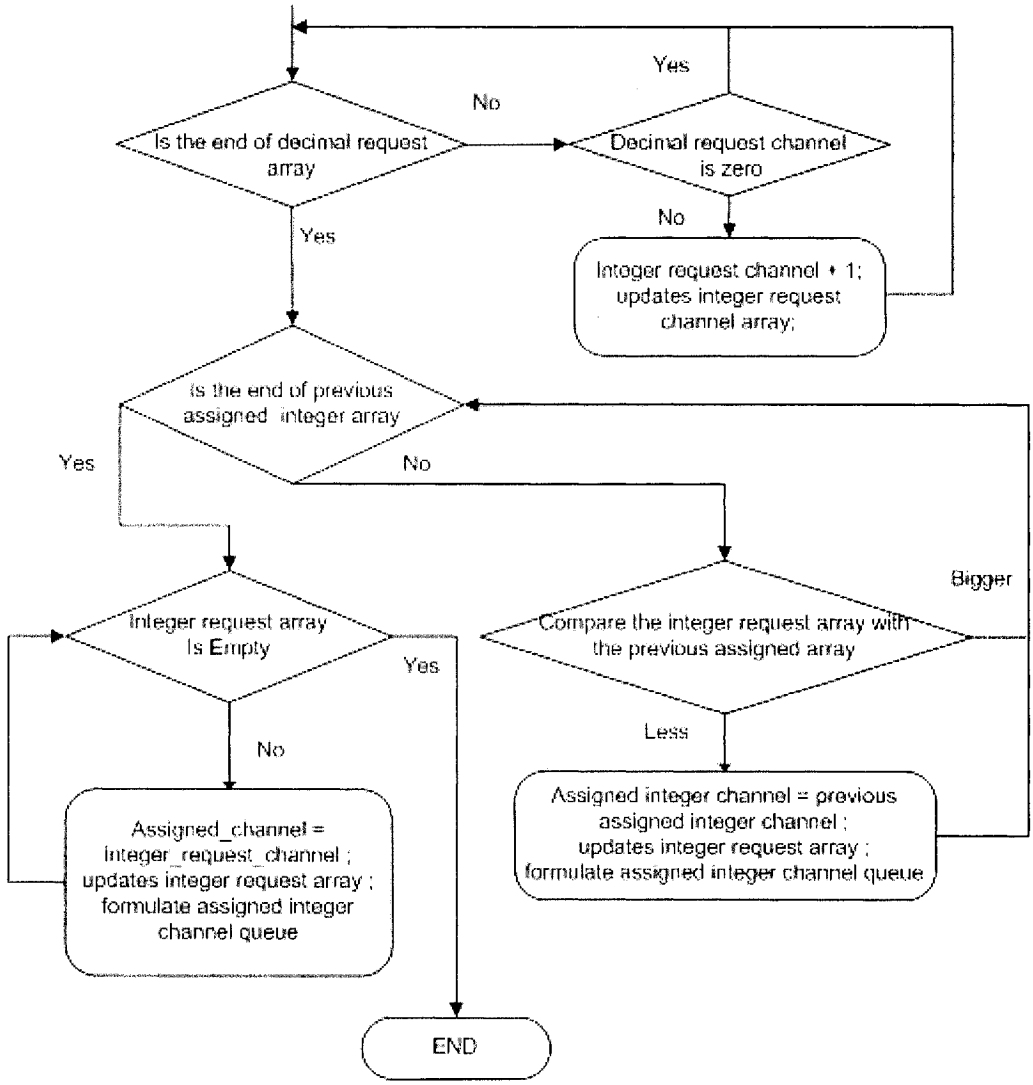


Figure 5.13 Simple channel assignment process

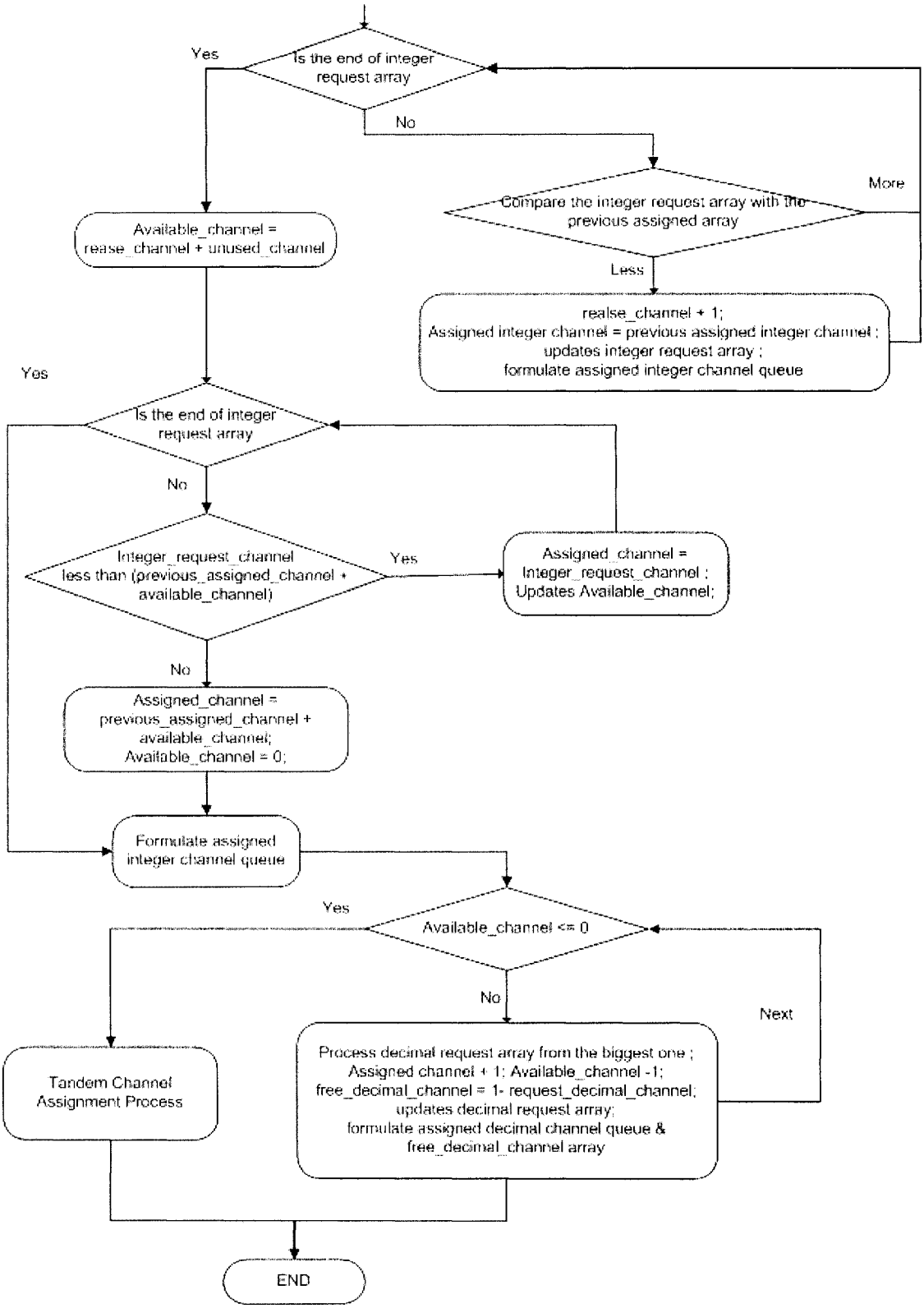


Figure 5.14 Complex channel assignment process

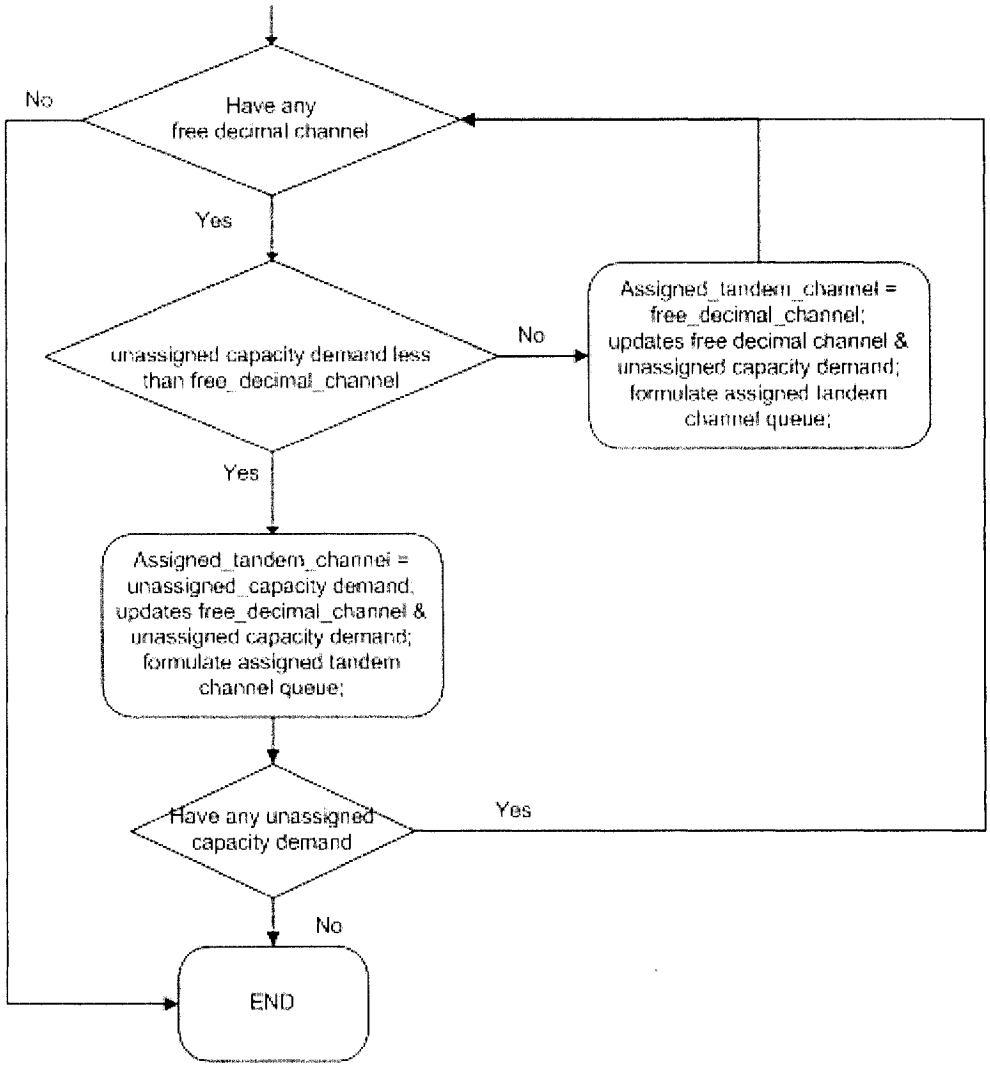


Figure 5.15 Tandem channel assignment process

Using reconfiguration request table (Figure 5.8) sent from the ingress edge modules, an integer channel request array and a decimal channel request array are formulated first, then the decimal channel request array is examined to determine if it is empty. If it is not, each request is assumed as one channel request. Compare the total number of the channels with the total number of requests. If the previous bigger than the latter, the simple channel assignment process is used, otherwise the complex channel assignment process is used. In the simple channel assignment process depicted in figure 5.13, first the decimal request

channel array is examined. If it is not empty, one more channel request will be added to the integer channel request array. Then, in order to reduce the channel connectivity changes in the core node, we release the unneeded channels. According to the integer channel request array, the number of integer channel request is compared with the previous assigned number of integer channels. If it is less, then the extra channel connections are released and the assigned integer channel array is updated. Finally, the integer channel requests, which are bigger than the pre assigned, are processed, new channels are added and connection identifiers are assigned.

In the complex channel assignment process depicted in figure 5.14, for the same reason, the release channel process is executed first, and then distributes the available channels to the integer channel requests and decimal channel requests. During the channel allocation, the biggest channel request (integer or decimal) is executed first, followed by the second and so on. This mechanism maximizes the channel utilization and the amount of traffic going through direct (non-tandem) paths. If the decimal channel request array is not empty while no more available channels can be assigned, then the tandem channel assignment process is used, illustrated in figure 5.15. Through this method, the traffic loss is reduced to the least, however this has side effects to the network performance such as larger delay and delay jitter.

In the tandem channel assignment process, first the free decimal channel array is examined. If no more decimal channel is available, the channel allocation will end. If there are some decimal channels available, then the biggest decimal channel request in decimal channel request array is compared with the biggest available decimal channel. If the request is bigger than the available decimal channel, the available decimal channel will be assigned to this request; otherwise, the request is assigned as the assigned tandem channel. After this, the request array, free channel array, and assigned tandem channel array are updated. Iterations are executed until all of the free decimal channels run out or all of the decimal channel requests are satisfied.

5.5 The QoS-Aware PetaWeb Network Model

As mentioned in chapter 4, the QoS-Aware PetaWeb network is designed to provide improved quality of service. Here we implement the simulation model, evaluate the performance of the QoS-Aware PetaWeb network, and verify the functionality of the QoS-Aware PetaWeb network. The simulation model of the QoS-Aware PetaWeb architecture using OPNET is developed, depicted in figure 5.16. In this scenario, five edge nodes are interconnected by one optical agile core. Each edge node is connected with two traffic sources/sinks. The traffic source models, which we use here in the experiments, are same with those used in the PetaWeb network model, (which is the combination of sinusoidal traffic, Poisson traffic, and the self-similar traffic described previously). However, traffic sources with different PHBs such as EF, AF, and BE are used in the experiments. The simulation model deploys the main components of the proposed network (detailed in chapter 4), the QoS-Aware edge node, QoS-Aware Channel Allocation Algorithm at core module, and others components in simulation model of the PetaWeb architecture, described in section 5.4. Also these nodes are interconnected by the multi-channel optical fibers. For simplicity, in our simulation and results analysis, we just use a single optical core node. We also assume that these edge nodes are distributed evenly around the core node over a wide geographical area, making up a global transport network. The propagation delay is 50ms from any edge to the core, and 100ms from an edge to any other edge. Optical link, which connects each edge node to core node, has a total 16 optical channels for data transmission and one control channel. Each optical channel has a capacity λ = 10 Gbps.

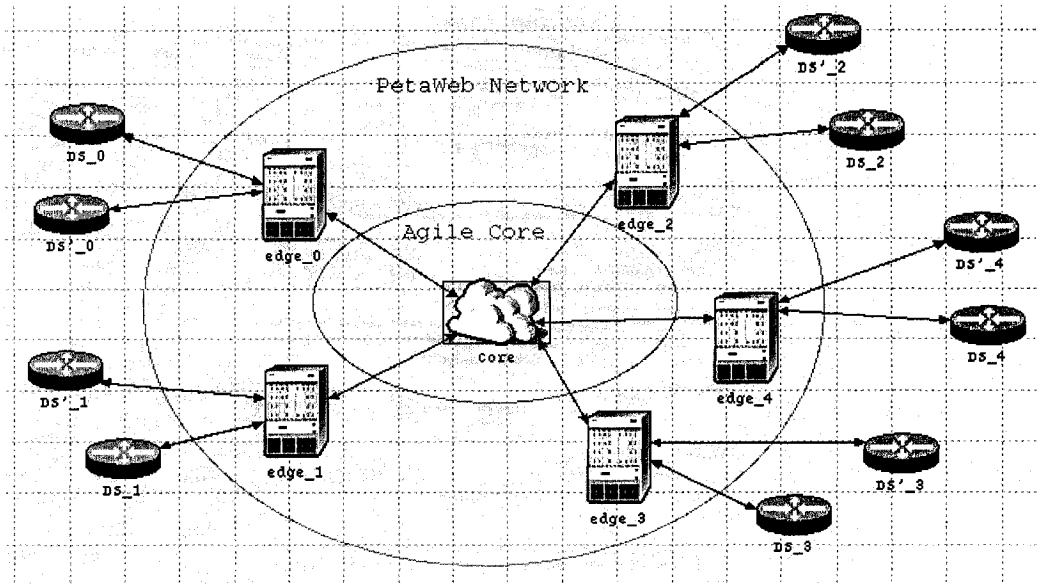


Figure 5.16 QoS-Aware PetaWeb Network Model

5.5.1 QoS-Aware Edge Node

At the edge node of PetaWeb, in order to enable edge node to support differentiated services, we add two kinds of queuing scheduling disciplines to manage the access of the allocated channels; one is Priority Queuing (PQ), the other is Weighted Round Robin (WRR). These queuing disciplines are simple and can be implemented in hardware, thus they are suitable for use in extremely high-speed router interfaces [73]. The incoming traffic streams at each edge node are sorted according to the class of service and destination address. Aggregated traffic streams are estimated at the edge nodes, and the capacity requests are sent from the edge nodes to the core controller. The highest-class packets are inserted into the highest priority queue such as EF traffic, and the lowest-class packets are inserted into the lowest priority queue such as BE traffic. In PQ, the scheduler will serve the highest priority queue if it has packets, the second highest, and so on. In WRR, the scheduler will serve different queues according to predetermined weighs assigned to the corresponding queue.

5.5.2 QoS-Aware Channel Allocation Algorithm

As mentioned in chapter 4, in order to support the differentiated service classes under any circumstance, we proposed QoS-Aware channel allocation algorithm. This algorithm solves the problem resulted by network congestion.

As we know, the channel allocation algorithm is implemented inside the core controller (see figure 5.7). At the core controller, the capacity requests, sent by the edge nodes, are placed in a QoS-Aware reconfiguration request table shown in Figure 5.17, based on the class of service and sorted according to the class of service with descending capacity order. These reconfiguration request tables are updated when the core controller receives new capacity requests.

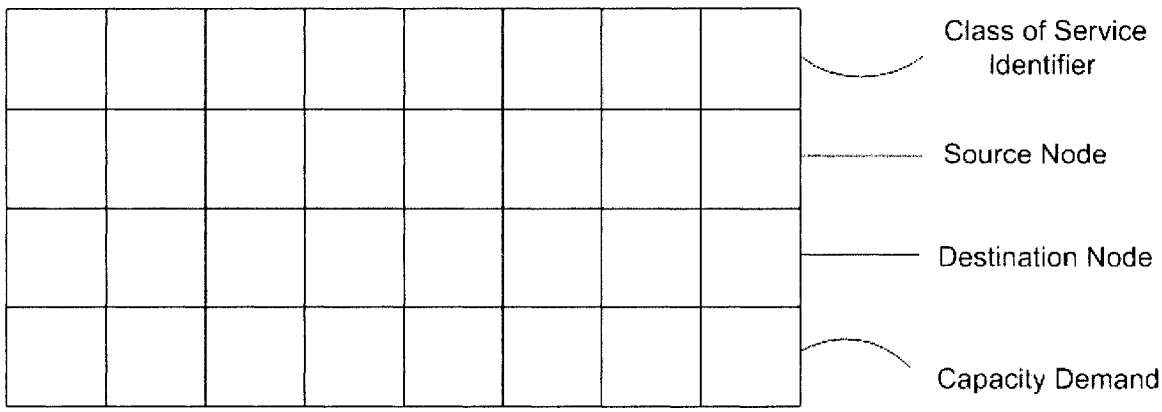


Figure 5.17 QoS-Aware Reconfiguration Request table

After channel reconfiguration, the results of the channel allocation are placed into several tables. In order to identify them from the previous one, we call them QoS-Aware assigned integer channel table, QoS-Aware assigned decimal channel table, and QoS-Aware assigned tandem channel table, illustrated in figures 5.18, 5.19, and 5.20 respectively, and then sent to the correspond nodes for switch reconfiguration.

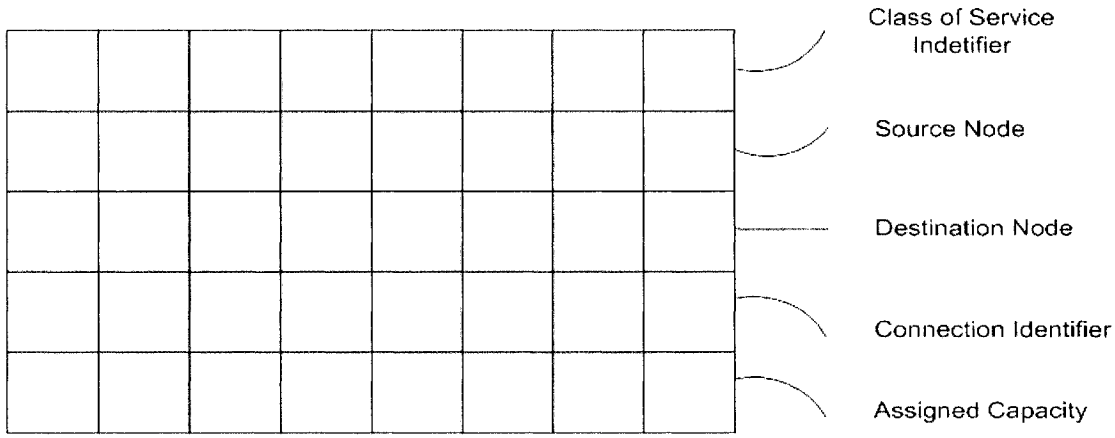


Figure 5.18 QoS-Aware Assigned integer channel table

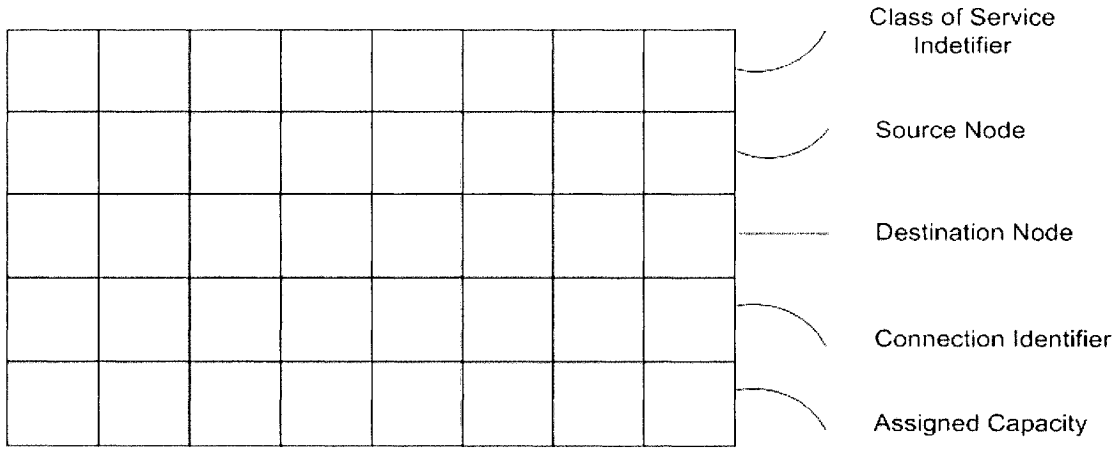


Figure 5.19 QoS-Aware Assigned decimal channel table

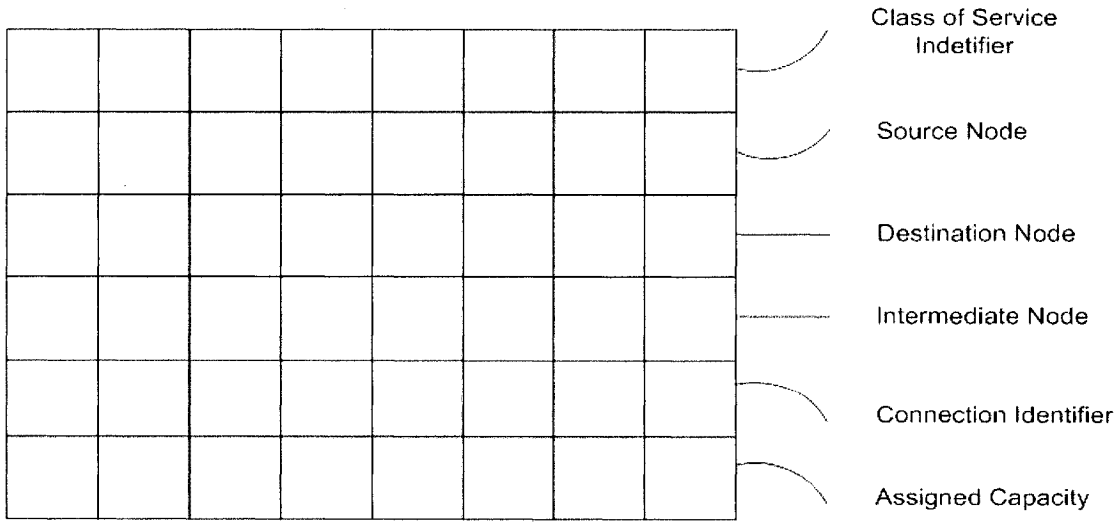


Figure 5.20 QoS-Aware Assigned tandem channel table

We modify the existing channel allocation algorithm to support different class of service. The detailed channel allocation algorithm is depicted as figure 5.21.

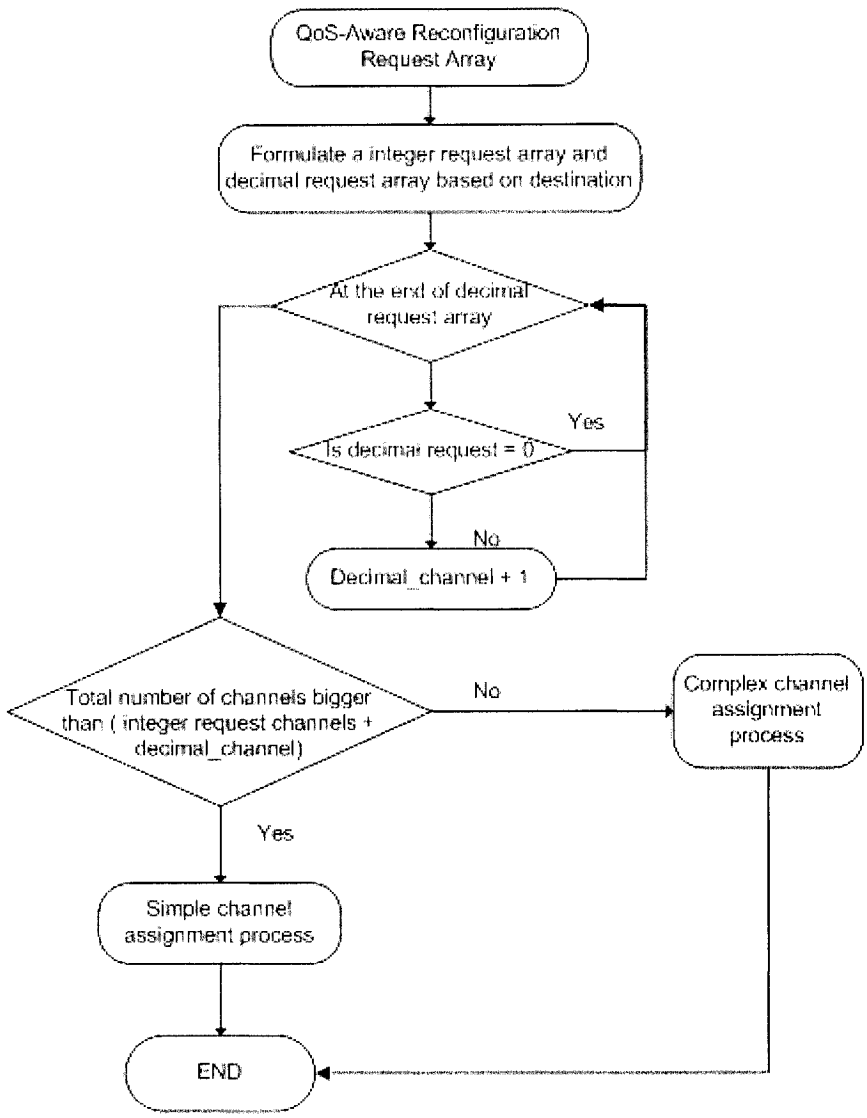


Figure 5.21 QoS-Aware Channel Allocation Algorithm Flow-Chart Diagram

Similar to the traditional algorithm, using a QoS-Aware reconfiguration request table (Figure 5.17), an integer channel request array and a decimal channel request array only based on the destination address are formulated first, then next several procedures are same as the previous, finally either simple channel assignment process or complex channel assignment process will be selected and used. In QoS-Aware channel allocation algorithm,

the simple channel assignment process and tandem channel assignment process are unchanged, and only the complex channel assignment process is modified. The detailed complex channel assignment process for QoS-Aware channel allocation algorithm is depicted in figure 5.22.

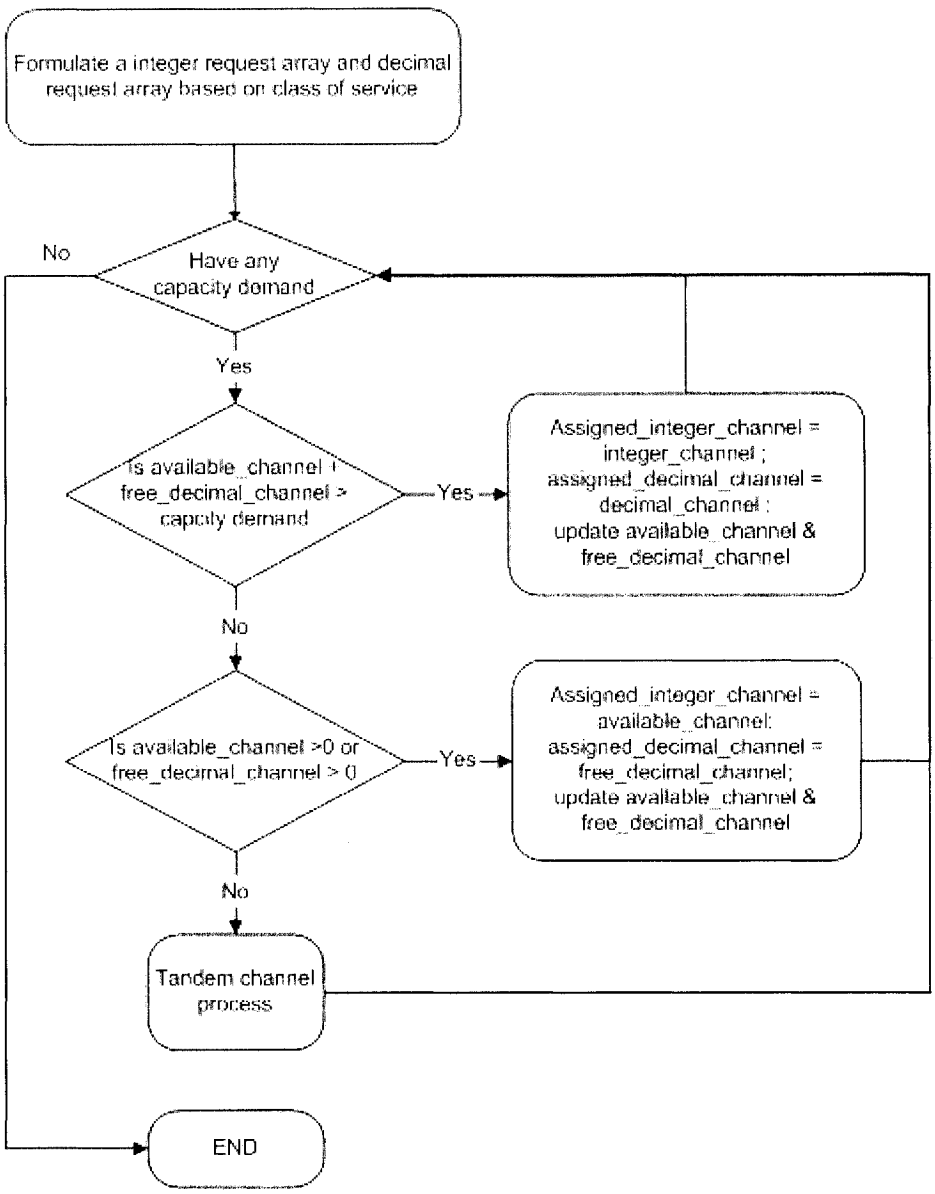


Figure 5.22 QoS-Aware Complex Channel assignment process

In the QoS-Aware complex channel assignment process, the capacity demand is sorted according to the class of service, destination and capacity. The algorithm first satisfies the

first class of service traffic, then the second class of service, and so on. If no direct path is available, then tandem paths will be assigned to the requests. Through this mechanism, the incoming traffic will be served according to their different service requirements.

5.6 Summary

In this chapter, we describe the simulation models of the general PetaWeb network and QoS-Aware PetaWeb network implemented in OPNET. Traffic models such as sinusoidal traffic model and self-similar traffic model are developed and used in the network simulation. The main logical components of the PetaWeb network are implemented, including the agile core node, the edge node, the channel allocation mechanism, the time coordination mechanism, etc. In order to evaluate the QoS over PetaWeb performance, the QoS-Aware edge node and the QoS-Aware channel allocation algorithm are also implemented.

Chapter 6

PERFORMANCE ANALYSIS

6.1 Introduction

In general, when evaluating a network and its components, we normally focus on several performance aspects, such as packet loss ratio, bit error rate, traffic delay, connection setup time, blocking probability, time between failures, reliability, scalability, throughput, and so on. However, most of these can be tested only when the components are developed and put together. Obviously, this is not possible when addressing the testing of a next generation optical network such as PetaWeb. Some components do not exist yet, so implementing an actual prototype of the PetaWeb network is not possible at this stage. Thus, we have to resort to simulation or emulation approaches.

Two main measurement parameters could be used to quantify the effect of the factors impacting on the agility of the PetaWeb. One is the amount of tandem traffic; another is the packet loss ratio. The higher the throughput of a network, the less the number of links, switches and routers for serving certain traffic volumes become, thus the lower the network cost becomes. An important traffic measure in networks is the amount of traffic passing through the network. Packets transmitted over tandem paths encounter additional transmission delay, optical to electronic and electronic to optical conversion latency, and also waste resources. Some parameters normally described the requirements of service level, such as packet loss ratio, delay and delay jitter are frequently used to describe the service level requirements. They are used in our work to quantify the effect of the factors impacting the performance of a QoS-Aware PetaWeb network.

In this chapter, simulations are carried out to quantify the effect of the factors impacting on the agility of the PetaWeb network, and to evaluate the performance of the QoS-Aware PetaWeb network. (Its detailed description can be found in Chapter 3, 4 and 5). The present

chapter is organized as follows. First, an introduction contrasts the behavior of the PetaWeb with that of a static core network and gives an illustration of the metrics used to quantify this behavior. Then, simulation results presented, are analyzing the effect of each of the following factors:

- Traffic type and traffic characteristics
- Reconfiguration period
- Buffer size at the edge node
- The number of channels and the granularity of optical channels

In section 6.8, the simulation results showing the performance of the QoS-Aware PetaWeb Network are presented. In section 6.9, we investigate Transmission Control Protocol (TCP) performance under the PetaWeb network, when compared to the static core network, under different reconfiguration periods.

6.2 The Benefits of PetaWeb

In our work, we first investigate the agility of the network, which is the most novel and important performance aspect for the PetaWeb, and the performance gains that agility brings. We compare the PetaWeb network with a static core network, which is a network with a similar architecture (i.e. star-like) as the PetaWeb network; the core of the network is operation with optical channel switching mode without wavelength conversion. In the static core network, the connectivity of channels or light paths between the edge nodes and core node is fixed. It does not change with the traffic pattern. However, in the PetaWeb, the connectivity of channels or light paths between end-to-end edge nodes can be reconfigured dynamically, depending on the traffic demand, in order to maximize the usage of network resources and minimize the sum of the tandem traffic and blocked traffic. Tandem traffic is transmitted over multiple hops (a hop is the trip that a data is transmitted from one edge node or intermediate node to another passing through optical core node). Here in this work, we restrict the tandem traffic only transmitted through one intermediate node; in other word, the tandem traffic is only transmitted through 2 hops in the network; as a result, they consume more network switching and transport resources than direct traffic (they reserve the wavelength over multiple links, use the resources of edge routers for optical to

electronic and electronic to optical conversions etc.). In addition, packets transferred through tandem paths end up experiencing considerably longer end-to-end delays than those of direct traffic.

6.3 Comparison of Agile and Static core configurations

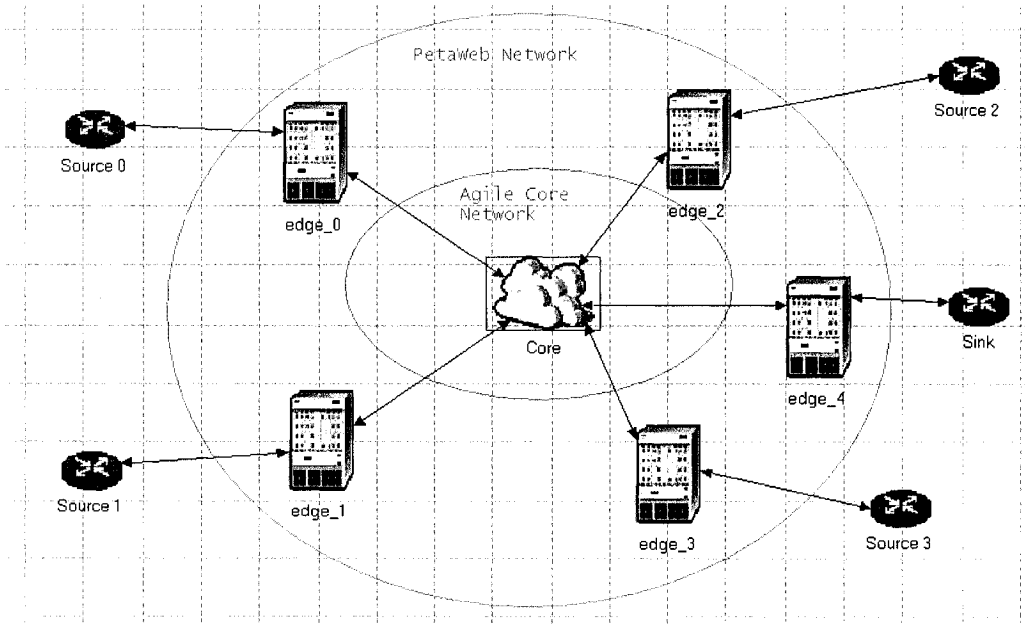


Figure 6.1: Simulated Network Architecture

Figure 6.1 (also presented in Figure 5.5) illustrates the architecture of the network we are using for evaluation. It consists of one core router and 5 routers. For the case of the static core network, a specific lightpath is fixedly connected to one of the other edge nodes; each edge node has 4 light paths available. For the agile core (the PetaWeb) network, light paths between each edge node pair are re-allocated at specific reconfiguration periods (e.g. 250 milliseconds, 500 milliseconds, 1 second, 2 seconds, and so on); thus, the number of light paths from the ingress edge nodes to each egress edge node may change with time, according to traffic demand. In our experiment, edge_4, which is set as the egress node, is connected with the "sink", whereas edge_0 to edge_3, set as source nodes, are connected with the traffic sources. The traffic sources, used in the experiments, are the self-similar traffic generated by the alpha-stable traffic model combined with the sinusoidal traffic described in Chapter 5.3. The traffic load is varied from 80 Gbps to 160 Gbps. In this work,

tandem traffic is defined as the proportion of the offered load that is transmitted through tandem paths; packet loss ratio is defined as the proportion of offered load that is dropped in the network; traffic load is defined as the total traffic transmitted through the link connected core node with the sink node divided by the link capacity (*Σ Traffic from source i to sink / Link capacity*).

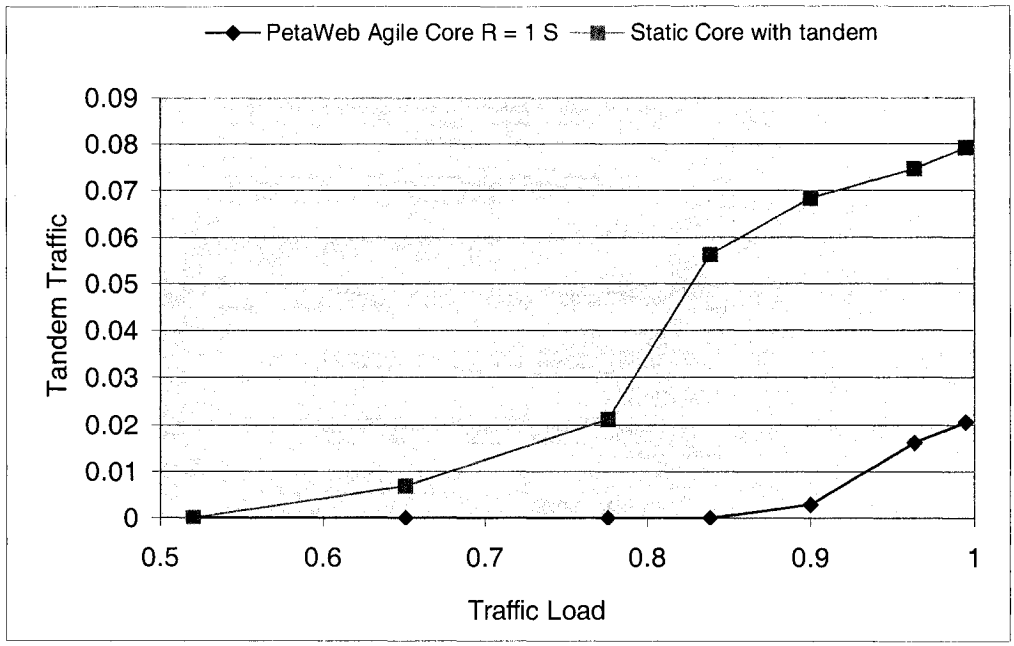


Figure 6.2: Fraction of Tandem Traffic vs. Traffic Load

Figure 6.2 and figure 6.3 displays the fraction of Tandem traffic and the packet loss ratio for the two cases of network configurations, as a function of the offered traffic load. The reconfiguration time for PetaWeb in this case is R = 1 second. The traffic source, we used in this experiments, is the sinusoidal traffic combined with α-stable traffic (for simplicity, later we call it *combined α-stable traffic*), and *the traffic volatility* (in other words, the ratio of the half amplitude of sinusoidal traffic to the average sinusoidal traffic) is set as 0.6. The ratio of average busty α-stable traffic to average sinusoidal traffic (for simplicity, later it is represented by symbol $Ratio_{\alpha\_stable}$) is set as 0.2. Note that, in reality, the traffic variation of such magnitude is very pessimistic. From figure 6.2, we see that as expected the tandem traffic increases with the traffic load. In the static core scenario, the tandem traffic is considerably higher than in the agile core (the PetaWeb network) scenario,

and the tandem traffic increase very fast when the traffic load becomes larger than 0.65. However, in the PetaWeb agile core scenario, the tandem traffic does not increase too much when the traffic load is less than 0.9. Under full loading, it reaches 0.08 for the static core, while it doesn't exceed 0.02 for the PetaWeb. Just as we expected, the tandem traffic can be eliminated in the PetaWeb network for all practical purposes. When the proportion of tandem traffic increases in the network, the average packet end-to-end delay and delay jitter increase and the network efficiency is lessened. This is demonstrated in Figures 6.4 and 6.5, where the average packet end-to-end delay and delay jitter are shown.

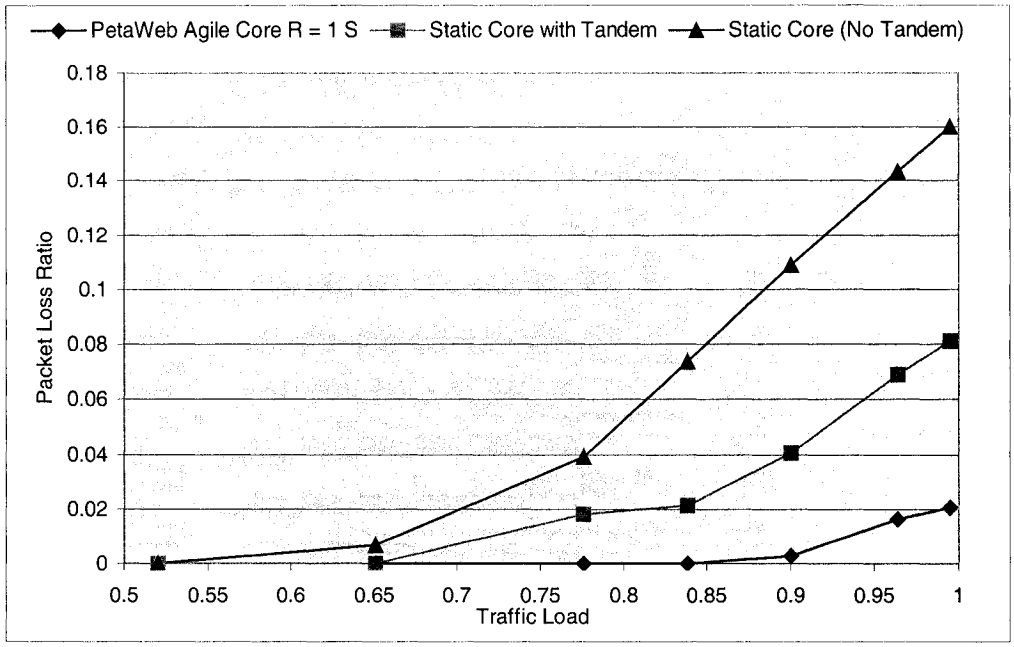


Figure 6.3 Packet Loss Ratio vs. Traffic Load

A key parameter in evaluating optical networks is the fraction of blocked traffic or the packet loss ratio. This occurs when the input traffic volume is higher than the capacity of the bandwidth available at the possible links between source and destination or buffer overflow. The packet loss ratio is calculated as the proportion of the offered load that is dropped. Figure 6.3 shows the packet loss ratio for the static core (No tandem), static core with tandem, and PetaWeb agile core cases. From the figure 6.3, we can find that packet loss ratio (or the blocked traffic volume) in static core scenario is considerably higher than in agile core scenario (the PetaWeb network). When the blocked traffic increases, the

network grade of service and quality of service decreases. So it is important to maximize the network throughput before the network quality of service degrades beyond a specified threshold, especially for TCP application and real-time multimedia application such as voice over IP, video on demand. Here we define the effective network throughput as the highest traffic load that can be offered to the network with the packet loss ratio under a certain level. If the packet loss ratio threshold is set to 0.01, the effective network throughput is 0.66 for the static core (no tandem) network, 0.72 for the static core (with tandem) network, 0.95 for the PetaWeb network. These results represent that, for the PetaWeb, the network capacity gain is 44% over the static core (no tandem) network, 32% over the static core (with tandem) network. Furthermore, the PetaWeb network's packet loss ratio increases very slowly, the reason is the PetaWeb can accommodate the traffic fluctuation through dynamic channel allocation; the packets are dropped due to the bursty characteristic of the traffic.

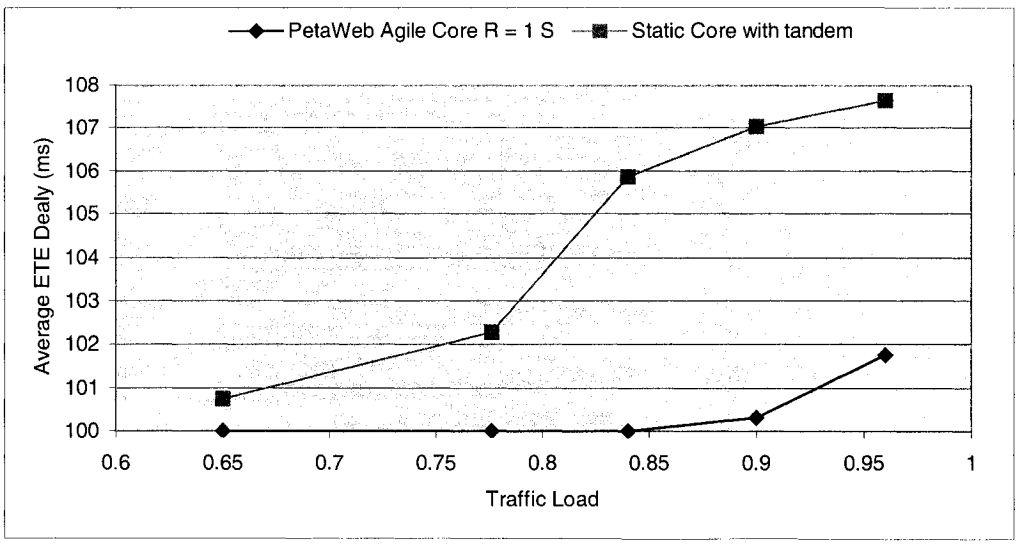


Figure 6.4 Average Packet end-to-end Delay vs. Traffic Load

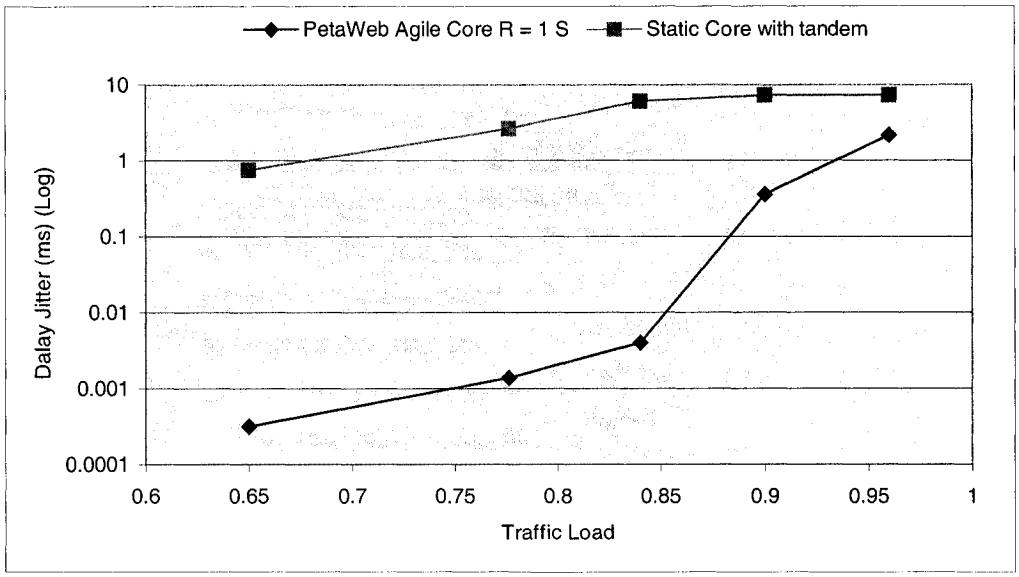


Figure 6.5 Packet Delay Jitter vs. Traffic Load

As we mentioned before, the advantages of the PetaWeb include increasing the effective network throughput and providing tandem path for excess traffic through reconfiguring the channel connectivity based on the edge nodes' traffic demands. However, packets switching through tandem path encounter higher propagation delay (in our experiments, the propagation delay from edge to edge passing through the core node is 100 milliseconds, the distance of the edge to edge is the half circumference of the earth), and processing latency (time for optical-to-electrical and electrical-to-optical conversion, in the experiment was set to 10ns, this common value for O/E/O conversion). As a result, tandem traffic increases average packet end-to-end delay and delay jitter extensively. From figures 6.4 and 6.5, we can see that in the PetaWeb network, when the traffic load excesses 0.84, there are some of packets transmitted through tandem path; as a result, the average end-to-end delay and delay jitter increase almost linearly in the PetaWeb network. It means the tandem traffic has a majority impact on the packet delay and delay jitter; due to the fact, traffic propagation delay is very high in long distance optical network. However, compared with the static core with tandem scenario, the traffic average end-to-end delay and delay jitter are considerably lower, due to lower tandem traffic in the PetaWeb network; as a result, the performance of the PetaWeb network increases significantly. However, as we

know, the high delay and delay jitter will definitely impact multimedia application performance, especially for video conferencing. So it is important to keep tandem traffic under low level in the network. There are several methods to eliminate the effect of the tandem traffic, for example, let the non-delay/jitter sensitive application traffic (Web text/image, e-mail, FTP, and Telnet, etc.) be switched through tandem paths, whereas delay/jitter sensitive application traffic can be transmitted through direct paths. This is one of the motivations we are proposing QoS-Aware PetaWeb architecture to support differentiated services (detailed description see Chapter 4, and 5), especially for real-time multimedia applications such as video on demand, video conferencing, voice over IP, etc.

In order to investigate the performance of the PetaWeb network more detail, we run simulations under different parameters (such as reconfiguration period, 250 ms, 1 s, 10 s) and the simulation results are presented to demonstrate that the performance of the PetaWeb satisfy the expectation or not.

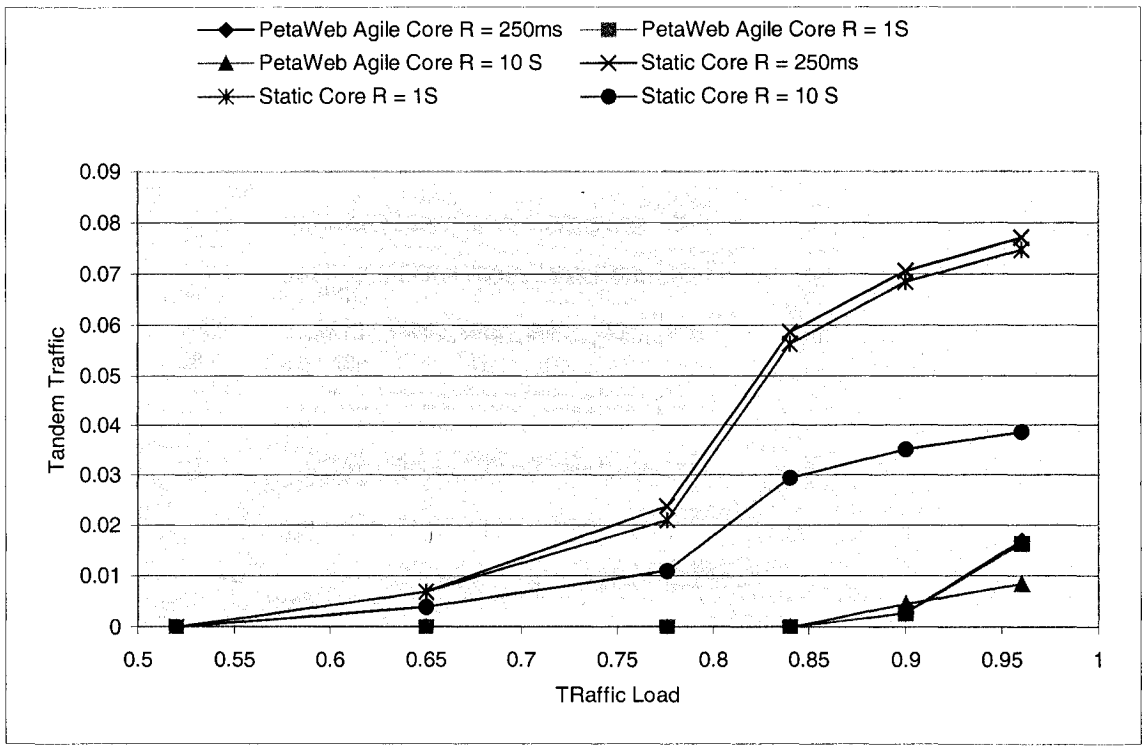


Figure 6.6 Tandem Traffic vs. Traffic Load with different reconfiguration period (R = 250 ms, 1 S, 10 S)

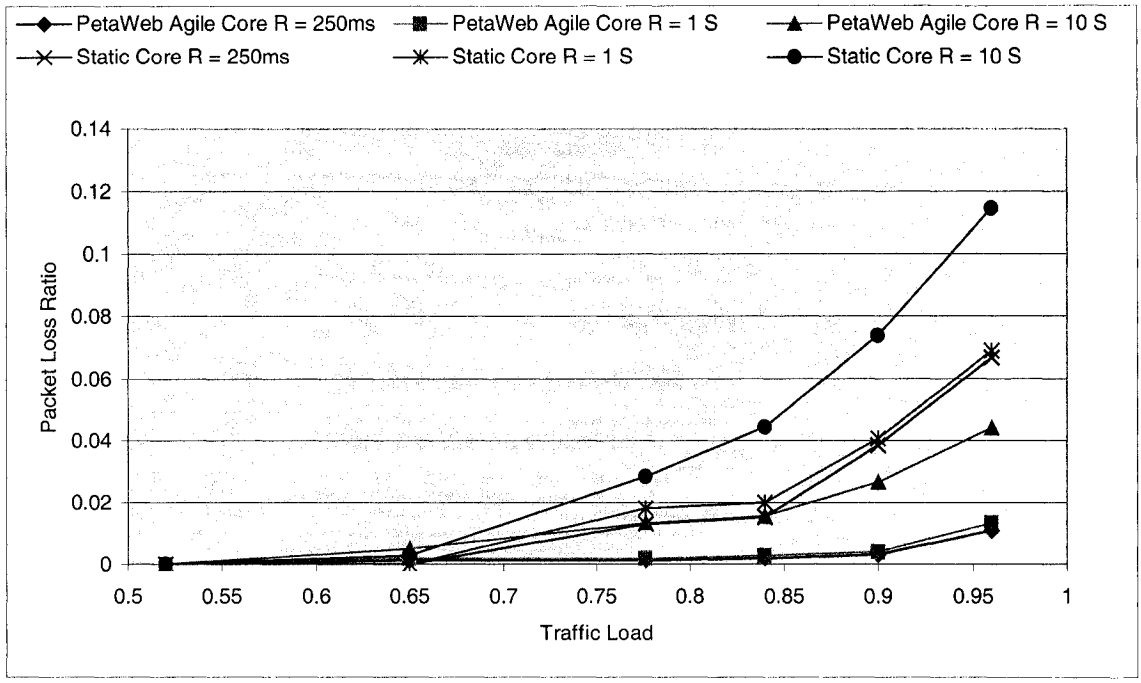


Figure 6.7 Packet Loss Ratio vs. Traffic Load with different reconfiguration period (R = 250 ms, 1 S, 10 S)

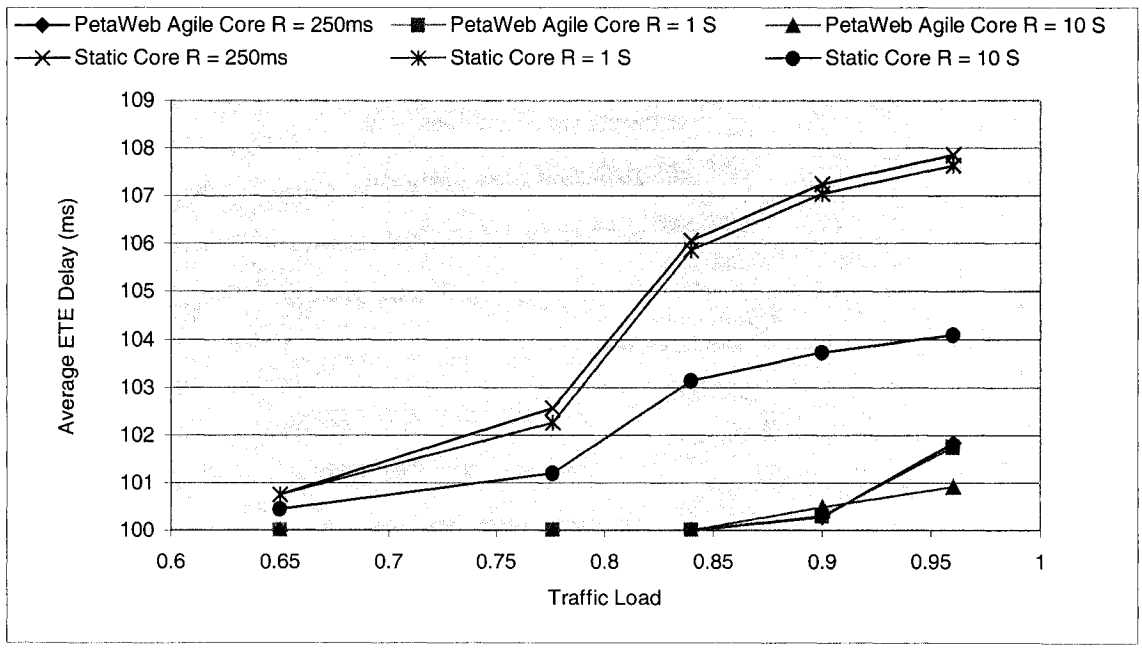


Figure 6.8 Average ETE Delay vs. Traffic Load with different reconfiguration period (R = 250 ms, 1 S, 10 S)

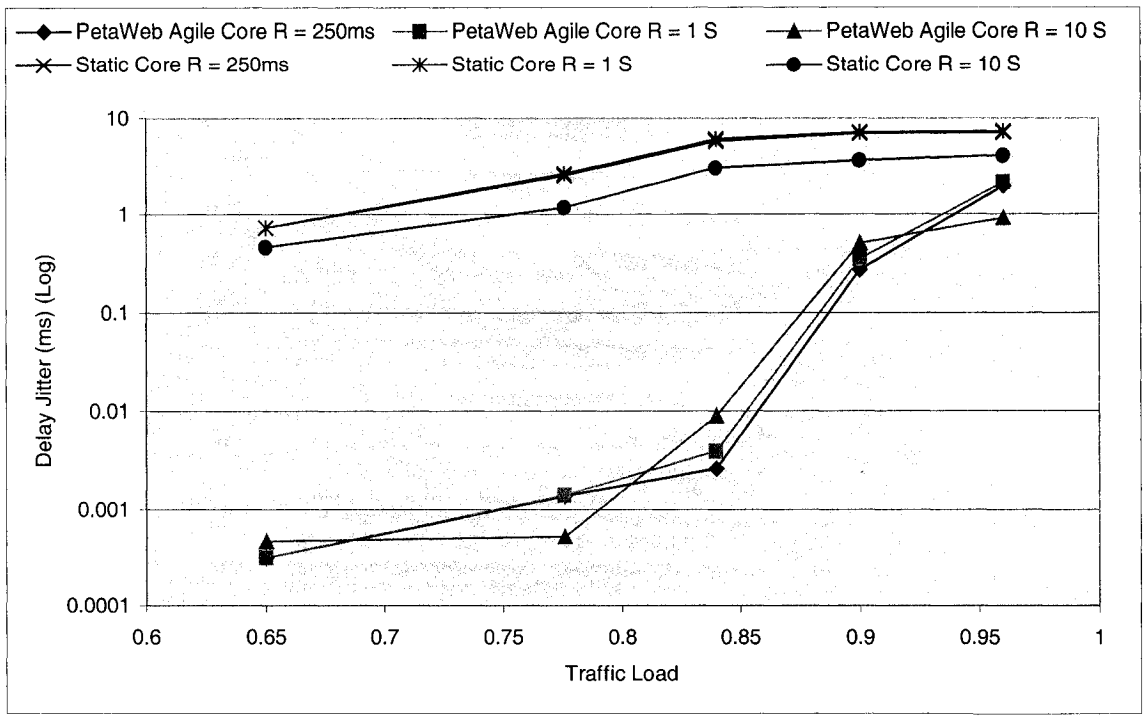


Figure 6.9 Delay Jitter vs. Traffic Load with different reconfiguration period (R = 250 ms, 1 S, 10 S)

The conclusions we draw from figure 6.6 to figure 6.9 are no matter what kind of reconfiguration time would be, the tandem traffic and packet loss ratio in the PetaWeb network decrease sharply compared with these in the static core with tandem scenario. In the PetaWeb network scenario, the tandem traffic and packet loss ratio increase quickly when the traffic load exceeds 0.84; the tandem traffic and packet loss ratio decreases when the reconfiguration period becomes shorter. However, in the static core (with tandem) network scenario, the tandem traffic increases with the shorter reconfiguration period, whereas the packet loss ratio decreases. Also from figure 6.8 and figure 6.9, we see that, in any cases, the average end-to-end delay and delay jitter increase with the traffic load no matter what the reconfiguration period is. The average end-to-end delay and delay jitter in static core (with tandem) scenario are considerably higher than in the PetaWeb network scenario. Later we will study the effect of reconfiguration period on the PetaWeb performance more detail in section 6.5. The results shown above demonstrate that the PetaWeb network has lower packet loss ratio and higher network utilization.

6.4 The Effect of the Traffic Type and Traffic Characteristics

Traffic injected to the network is one key factor in the network performance evaluation. It is not easy to predict the characteristics of the traffic in the future network due to Internet's rapid growth and new emerge applications. In our work, different traffic models (such as sinusoidal, Poisson, and alpha-stable self-similar) and their combinations are taken into account, and different traffic parameters, which may impact network performance, are considered. The important parameters include the traffic volatility (in other words, the ratio of the half amplitude of sinusoidal traffic to the average sinusoidal traffic), the ratio of average of bursty α-stable traffic (or Poisson) to average sinusoidal traffic, and the bursty level of the self-similar traffic. In order to acquire a clearer idea of how different traffic characteristics and parameters affect the PetaWeb performance, we are presenting in figures 6.10 to 6.19 the tandem traffic, traffic blocking probability, and average end-to-end delay and delay jitter under different traffic load. In the experiments, we use three different types of traffic combinations, which are alpha-stable self-similar traffic combined with sinusoidal traffic (for simplicity, later we call it *combined α-stable traffic*), Poisson traffic combined with sinusoidal traffic (for simplicity, later we call it *combined Poisson traffic*), and *single sinusoidal traffic* as the network input traffic sources. Two network configuration cases (agile core and static core) are examined. The reconfiguration periods for PetaWeb are set as R = 250 milliseconds, 1 second, and 10 seconds respectively. In the first set of experiments, *combined α-stable traffic* source is used; *the traffic volatility* is set as 0.6; $Ratio_{\alpha\_stable}$ is set as 0.2. In the second set of experiments, Poisson traffic is replacing alpha-stable traffic; the same traffic compositions are used. In the third set of experiments, *single sinusoidal traffic* source is used. The simulation results are presented as follows.

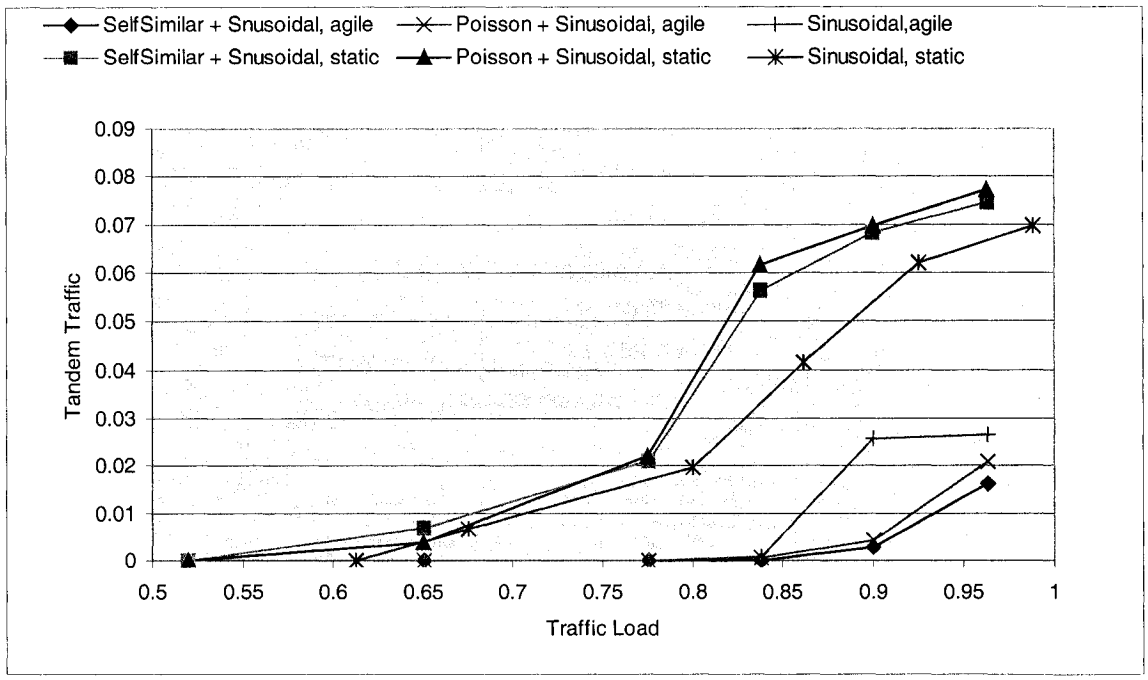


Figure 6.10 Tandem Traffic vs. Traffic Load for Different Traffic Compositions (Reconfiguration Period R = 1 S)

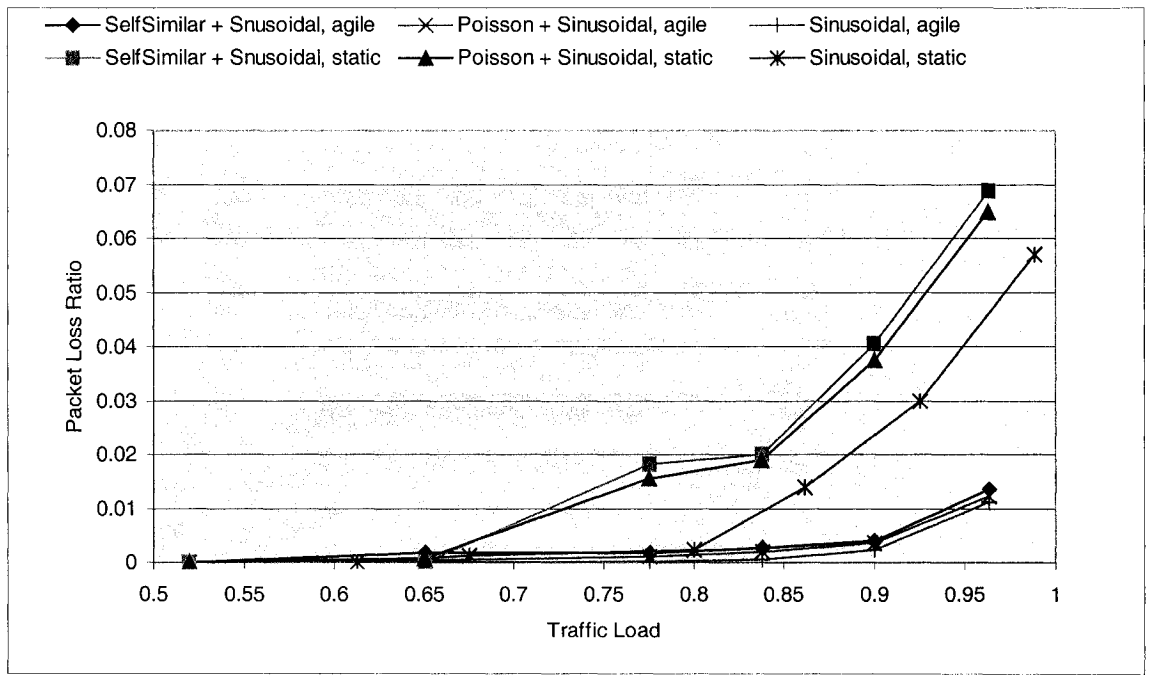


Figure 6.11 Packet Loss Ratio vs. Traffic Load for Different Traffic Compositions (Reconfiguration Period R = 1 S)

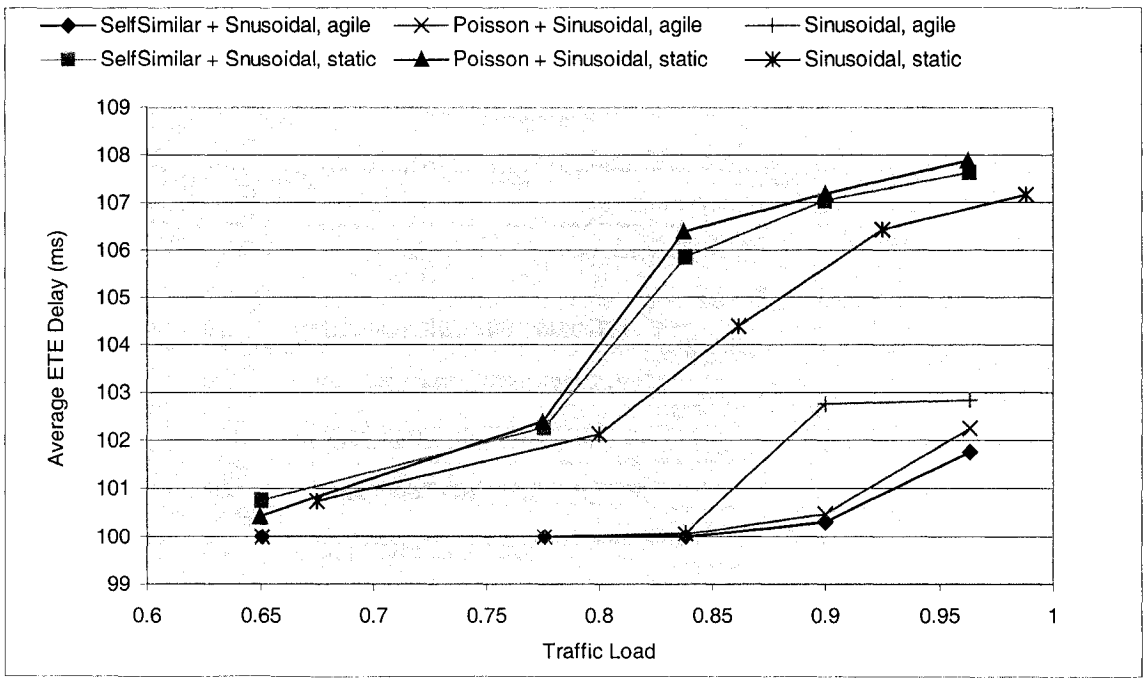


Figure 6.12 Average End-To-End Delay vs. Traffic Load for Different Traffic Compositions (reconfiguration period R= 1 S)

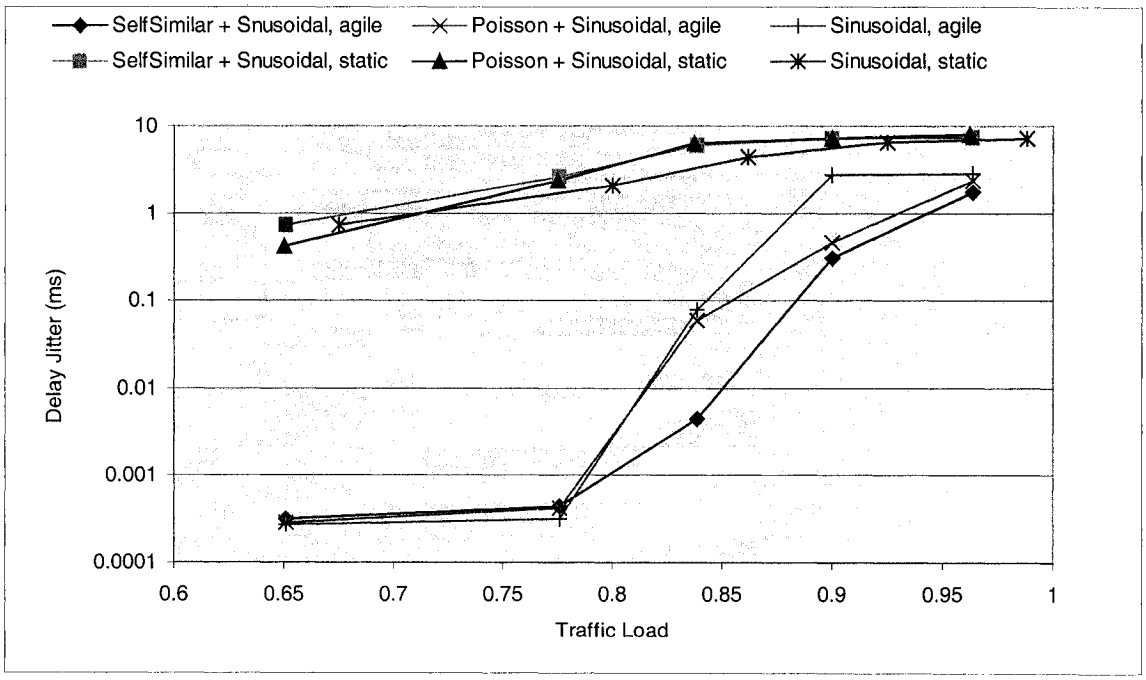


Figure 6.13 Delay Jitter vs. Traffic Load for Different Traffic Compositions (reconfiguration period R= 1 S)

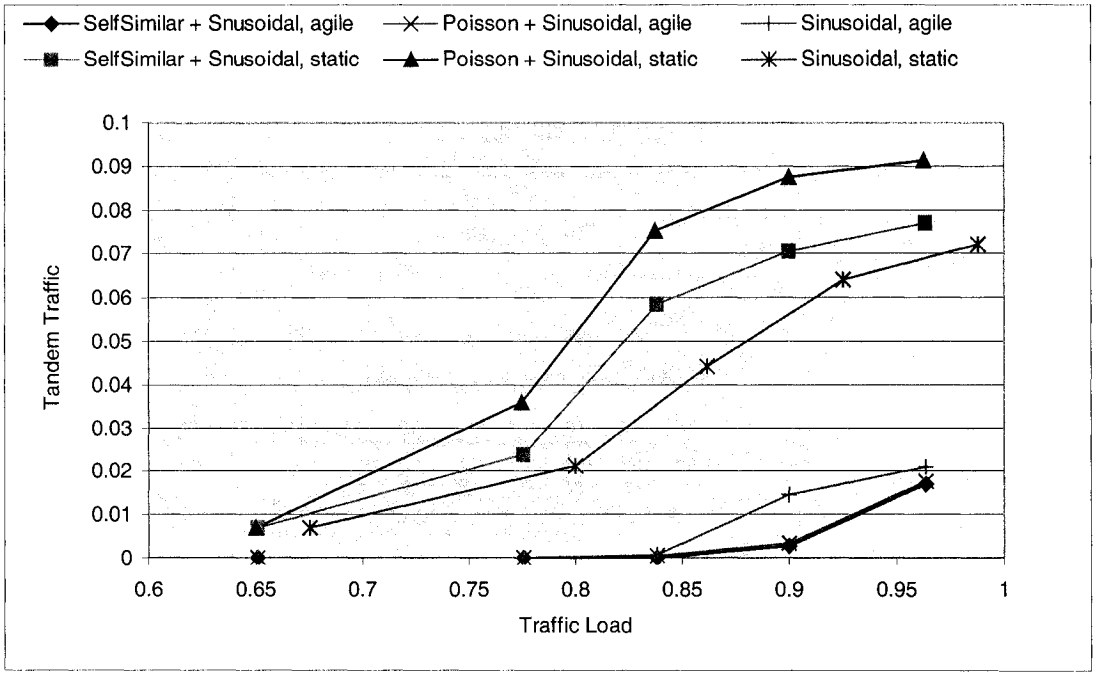


Figure 6.14 Tandem Traffic vs. Traffic Load for Different Traffic Compositions (reconfiguration period R= 250 ms)

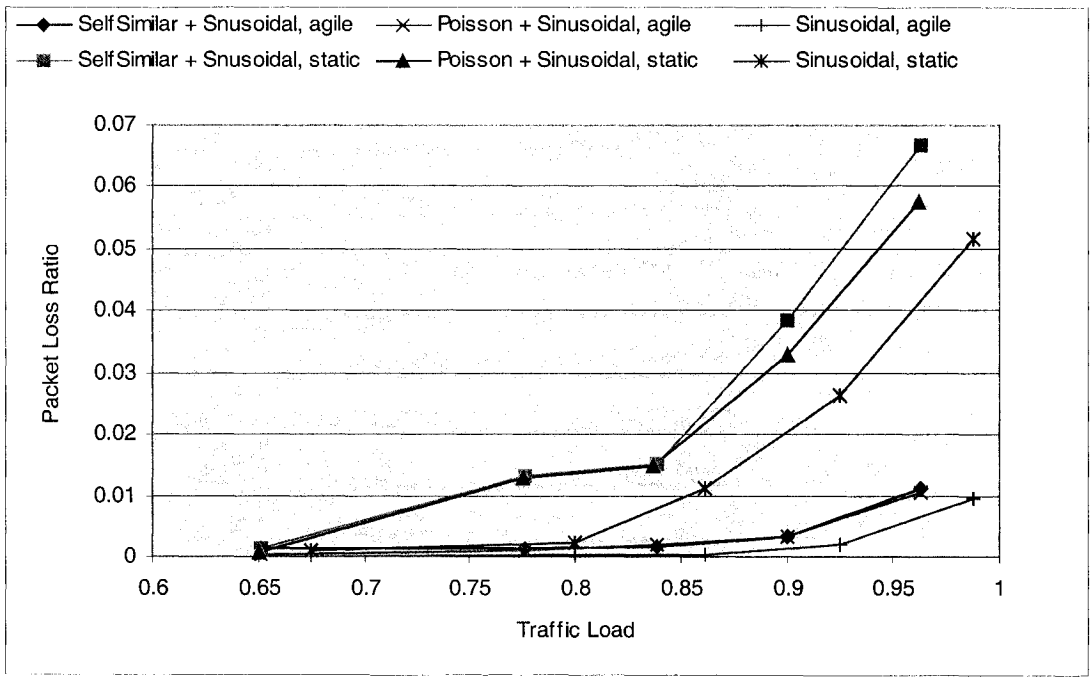


Figure 6.15 Packet Loss Ratio vs. Traffic Load for Different Traffic Compositions (reconfiguration period R = 250 ms)

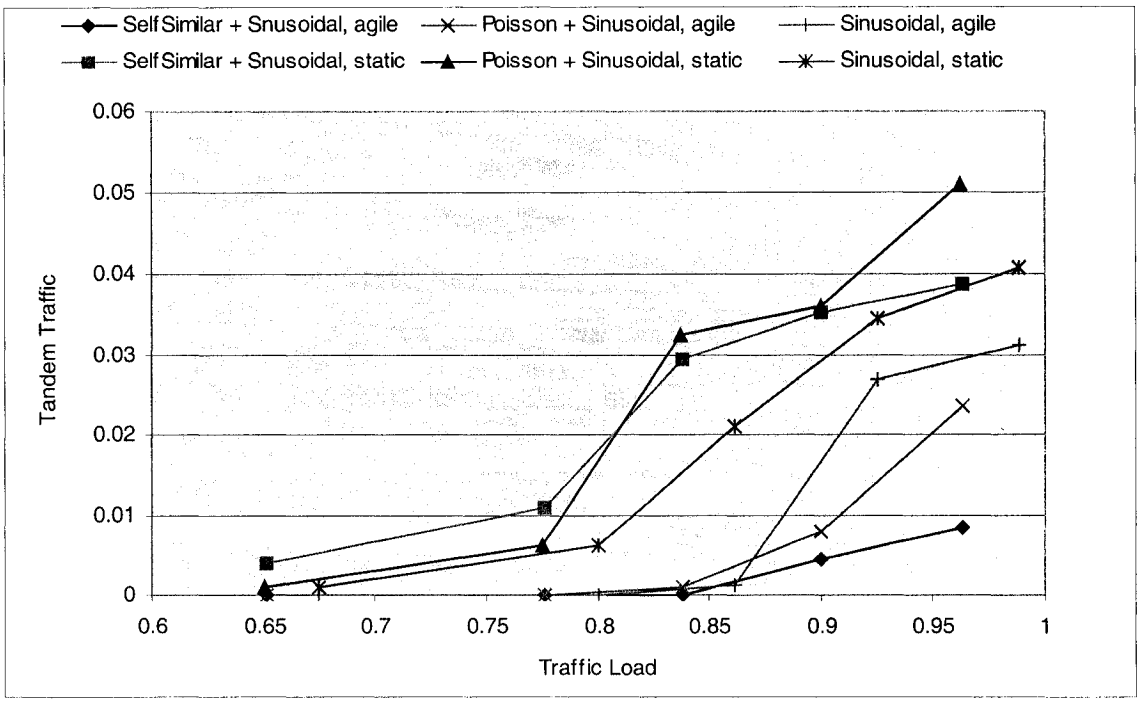


Figure 6.16 Tandem Traffic vs. Traffic Load for Different Traffic Compositions (reconfiguration period R= 10 S)

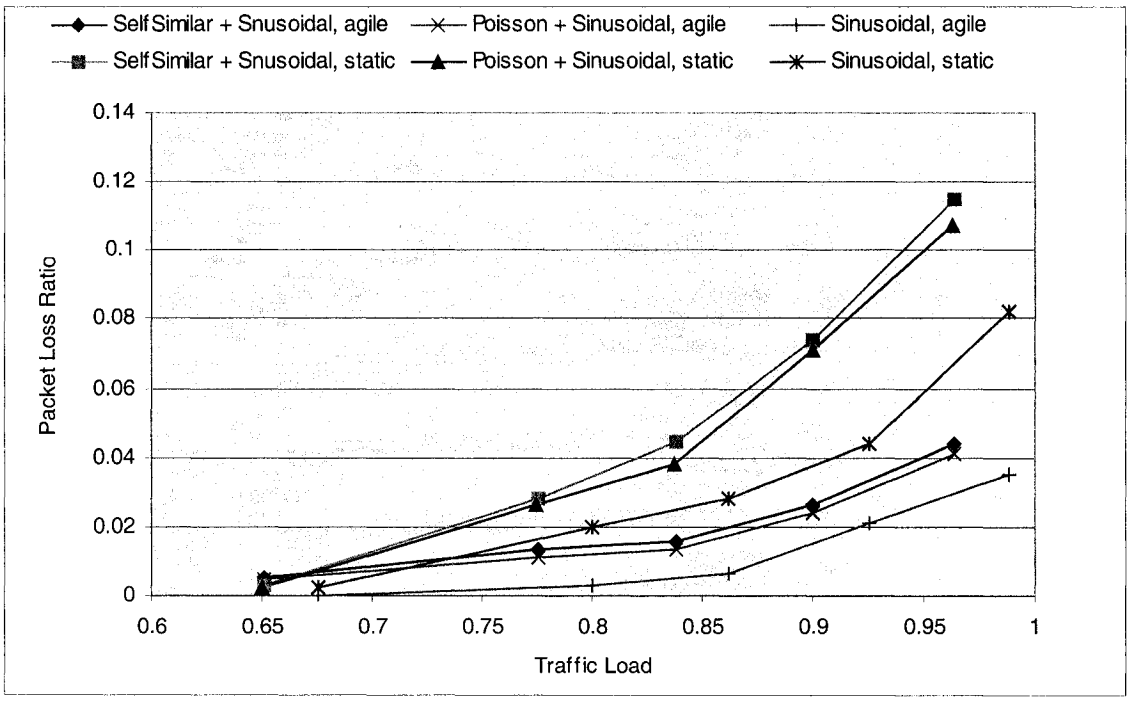


Figure 6.17 Packet Loss Ratio vs. Traffic Load for Different Traffic Compositions (reconfiguration period R = 10 S)

In figure 6.10 and 6.11, the tandem traffic and packet loss ratio for the static core (with tandem) and agile core with reconfiguration period R = 1S are presented. In the static core (with tandem) scenario, the results of tandem traffic under combined alpha-stable self-similar and sinusoidal traffic source pattern are very close to those under combined Poisson and sinusoidal traffic. As expected, the packet loss ratio of single sinusoidal traffic source is lower than both of the combined traffic sources. The reason is that the single sinusoidal traffic source is not bursty. In the PetaWeb scenario, the tandem traffic and packet loss ratio is very small no matter what type of traffic sources we use, it demonstrates the advantage of the PetaWeb and its robustness to traffic burstiness. From the figures 6.10 to 6.13, we can find that the packet loss ratio, tandem traffic, delay, and delay jitter in the PetaWeb are quite low regardless of the traffic composition as compared to the static core network. Also, the tandem traffic for any traffic source is almost 0 when the traffic load does not exceed 0.85. Furthermore, we see that the tandem traffic has a dominant impact on the packet delay and delay jitter; the packet average end-to-end delay and delay jitter increase with the tandem traffic. From figures 6.11, 6.15, and 6.17, we can see that for all traffic compositions, the packet loss ratio decreases with the reconfiguration period. The packet loss ratio for single sinusoidal traffic source scenario is always lower than alpha-stable combined with sinusoidal traffic and Poisson combined with sinusoidal traffic. The packet loss ratio of alpha-stable and sinusoidal traffic is a little larger than Poisson and sinusoidal traffic due to the higher burstiness of the alpha-stable traffic.

To further investigate the effect of traffic behavior on the PetaWeb and static networks, we implemented three more sets of simulations. The alpha-stable traffic combined with sinusoidal traffic source is used and the traffic load is set to 0.65, 0.84, and 0.96 respectively in all cases. In the first set of experiments, we change *the traffic volatility*, setting to 0.4, 0.6, and 0.8 respectively. In the second set of experiments, the ratio of average of burst α-stable traffic to average sinusoidal traffic ($Ratio_{\alpha\_stable}$) is changed to 0.2, 0.4, and 0.6 respectively. The traffic volatility is set as 0.6. In the third set of experiments, different bursty levels of the alpha-stable self-similar traffic are used, the traffic volatility is set as 0.6, the $Ratio_{\alpha\_stable}$ is set as 0.2. In the simulation, by setting

alpha-stable parameters α = 1.05 and H = 0.997, the heavy bursty traffic is obtained; the light bursty traffic is obtained by setting α = 1.95 and H = 0.834.

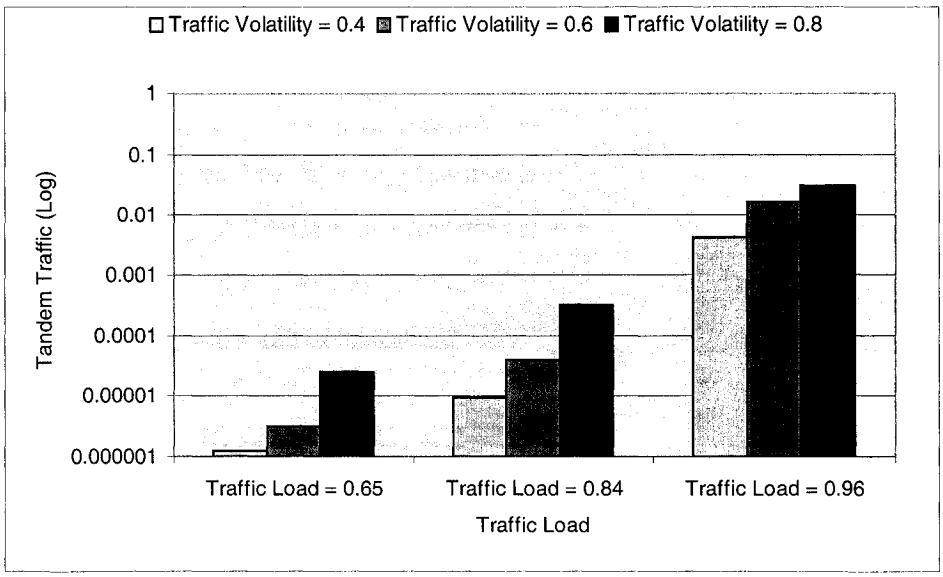


Figure 6.18 Tandem Traffic vs. Traffic Load for different traffic volatility (Reconfiguration period R = 1 S)

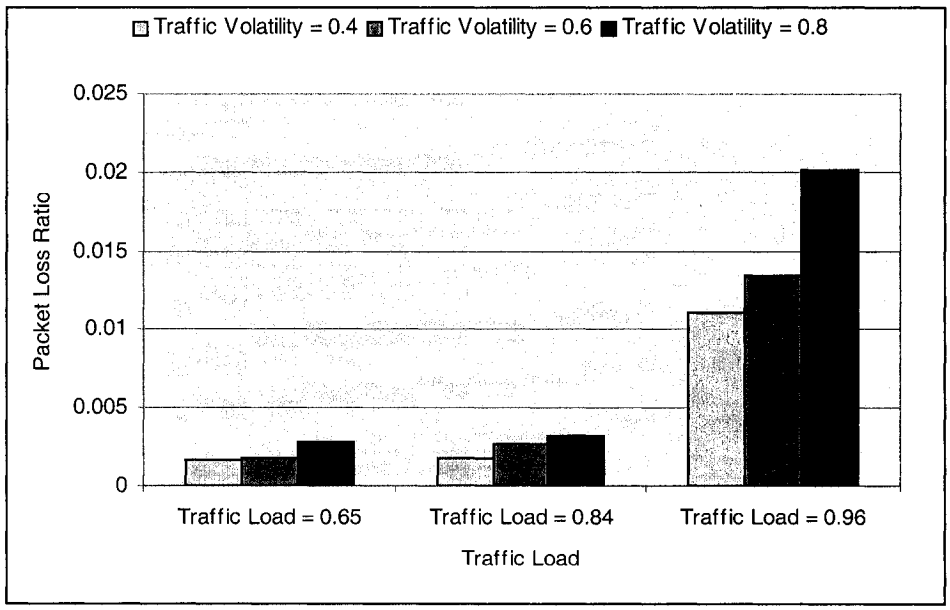


Figure 6.19 Packet Loss Ratio vs. Traffic Load for different traffic volatility (Reconfiguration period R = 1 S)

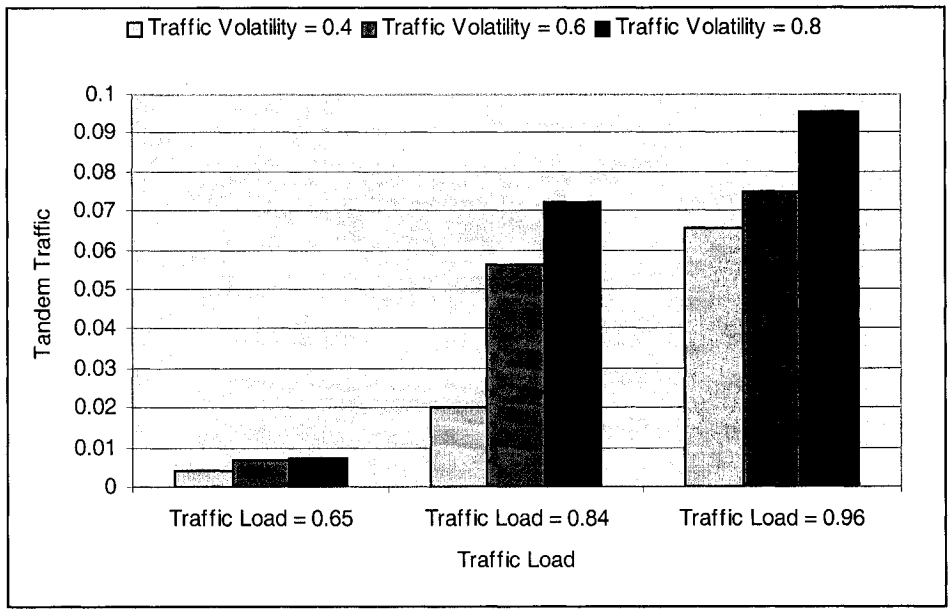


Figure 6.20 Tandem Traffic vs. Traffic Load for different traffic volatility (Static Core Scenario)

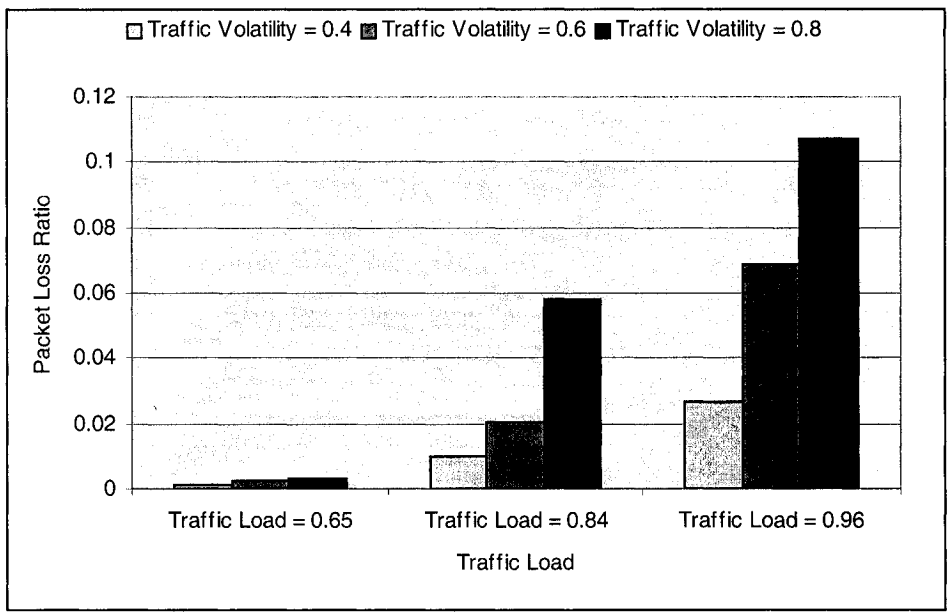


Figure 6.21 Packet Loss Ratio vs. Traffic Load for different traffic volatility (Static Core Scenario)

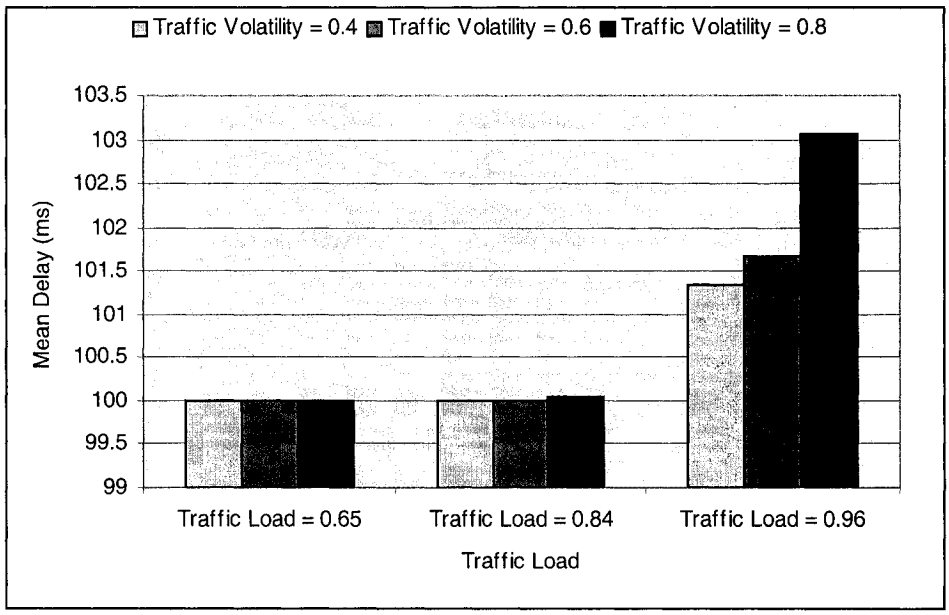


Figure 6.22 Average end-to-end Delay vs. Traffic Load for different traffic volatility (Reconfiguration period R = 1 S)

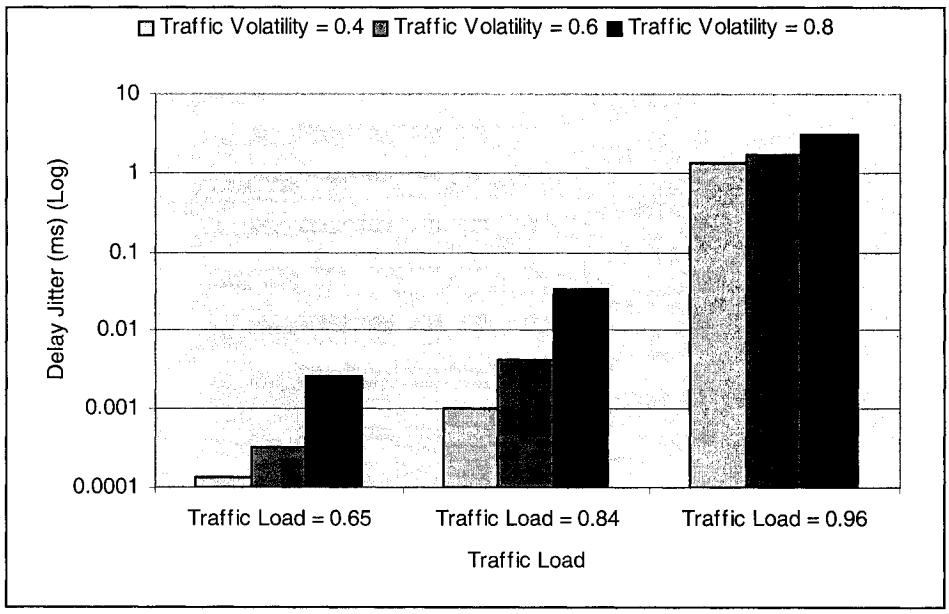


Figure 6.23 Delay Jitter vs. Traffic Load for different traffic volatility (Reconfiguration period R = 1 S)

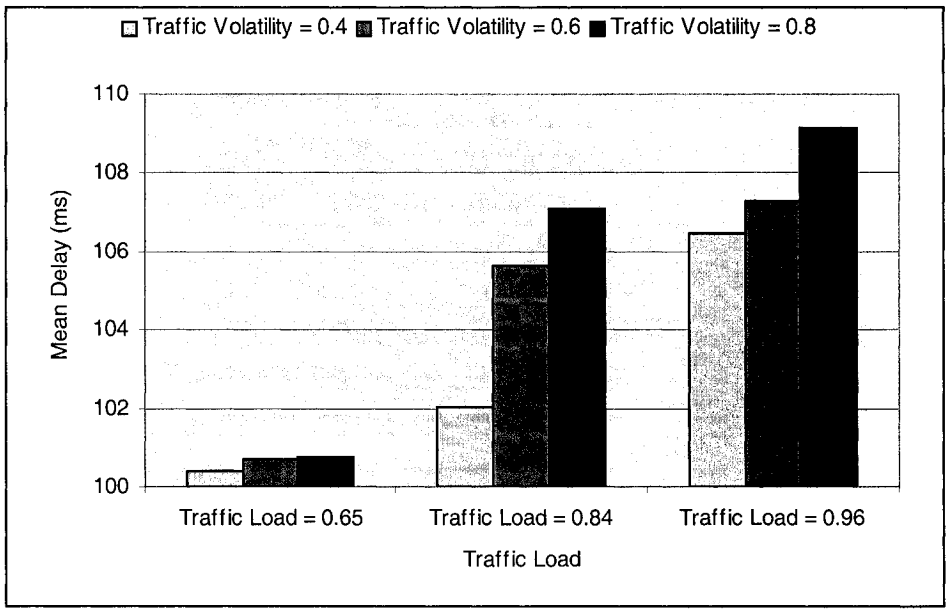


Figure 6.24 Average end-to-end Delay vs. Traffic Load for different traffic volatility (Static Core Scenario)

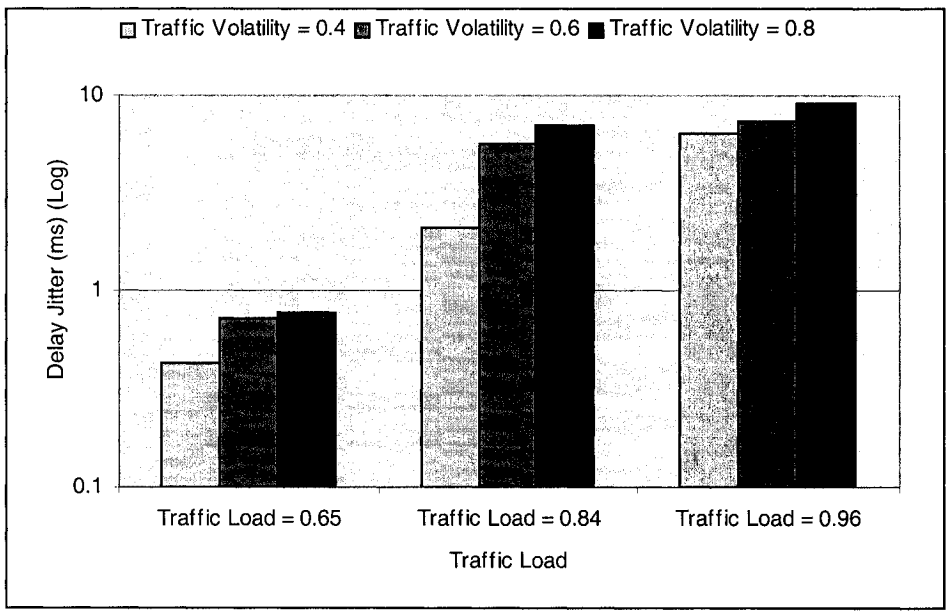


Figure 6.25 Delay Jitter vs. Traffic Load for different traffic volatility (Static Core Scenario)

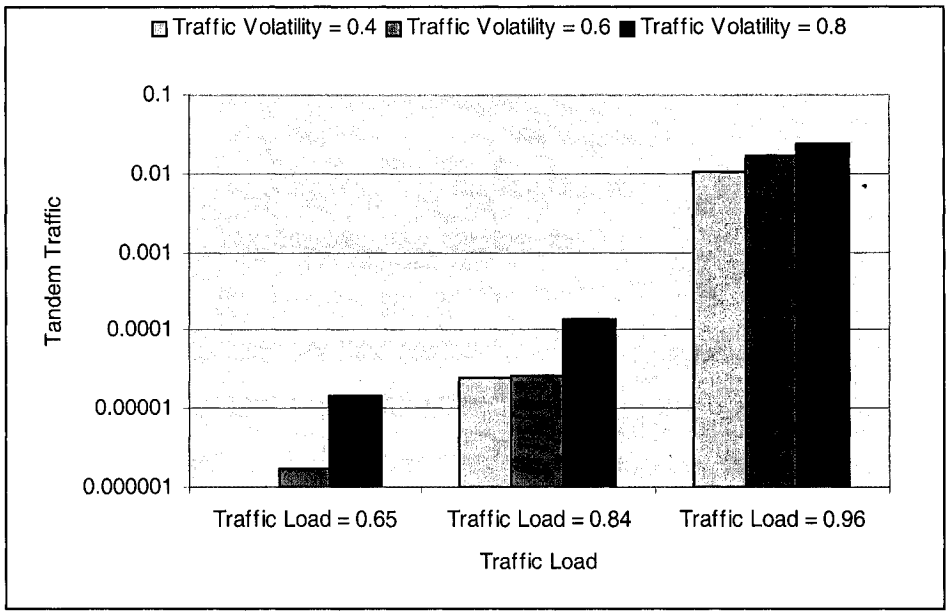


Figure 6.26 Tandem Traffic vs. Traffic Load with different traffic volatility (Reconfiguration period R = 250 ms)

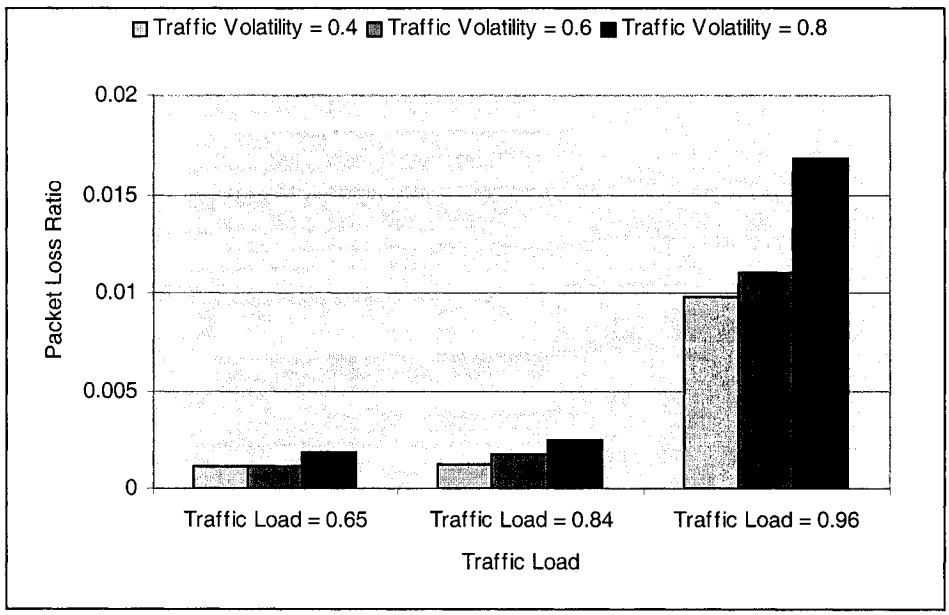


Figure 6.27 Packet Loss Ratio vs. Traffic Load with different traffic volatility (Reconfiguration period R = 250 ms)

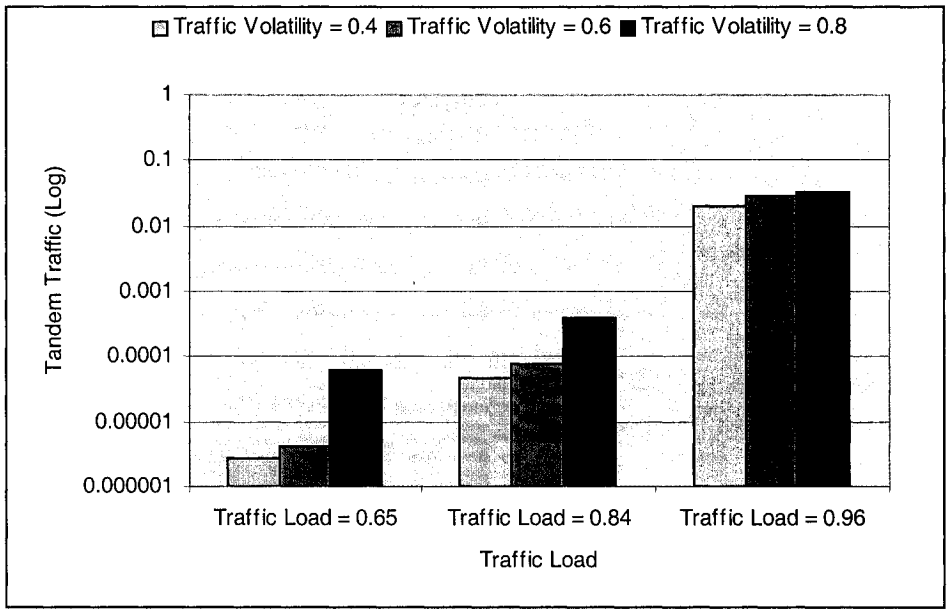


Figure 6.28 Tandem Traffic vs. Traffic Load with different traffic volatility (Reconfiguration period R = 10 S)

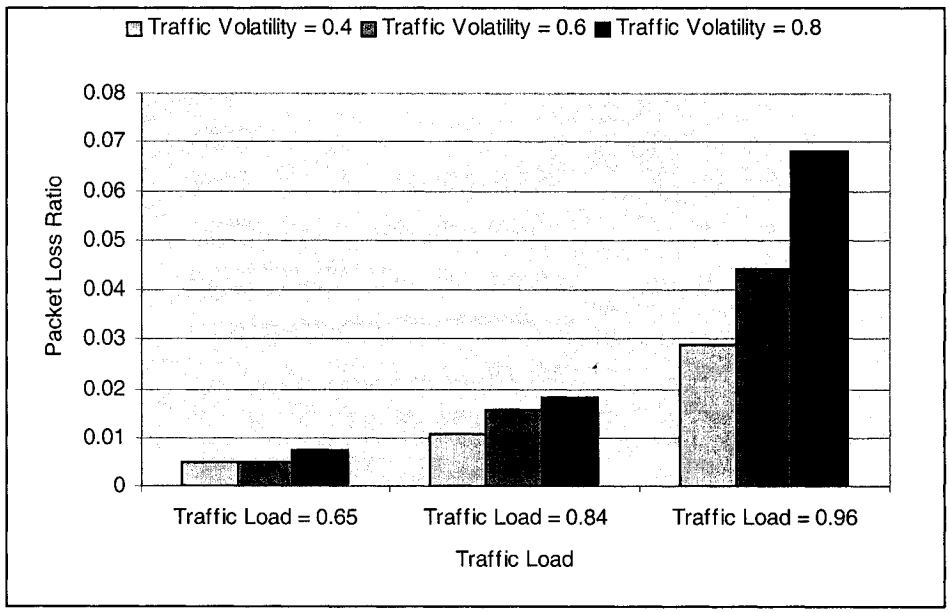


Figure 6.29 Packet Loss Ratio vs. Traffic Load with different traffic volatility (Reconfiguration period R = 10 S)

From figure 6.18 to figure 6.29, we can conclude that by increasing the traffic volatility, the tandem traffic and packet loss ratio will increase under any traffic load and any reconfiguration period. As a result, the network performance degrades, when the traffic

volatility increases. However, by comparing the PetaWeb with the static core network (see figure 6.20 and figure 6.21), it is evident that the effect of traffic volatility on the PetaWeb is far less than its effect on static core network, meaning that the PetaWeb has better capability to adapt traffic fluctuations than the static core network.

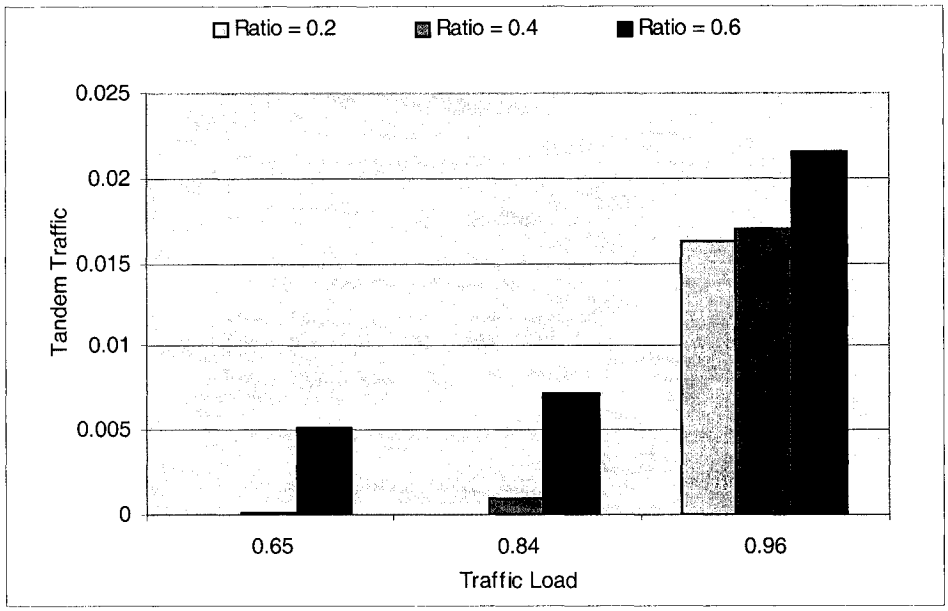


Figure 6.30 Tandem Traffic vs. Traffic Load for different $Ratio_{\alpha\_stable}$ values (reconfiguration period R= 1 S)

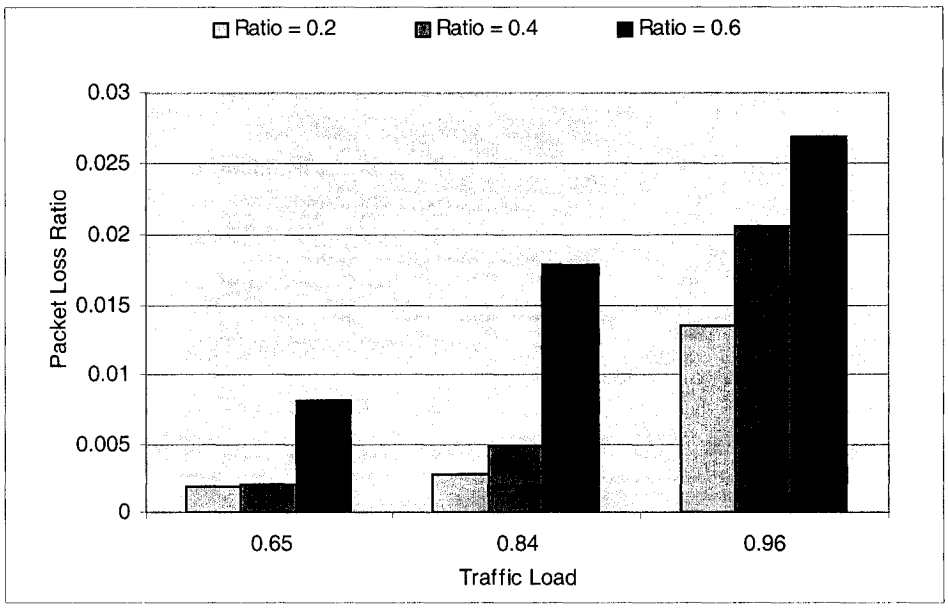


Figure 6.31 Packet Loss Ratio vs. Traffic Load for different $Ratio_{\alpha\_stable}$ values (reconfiguration period R= 1 S)

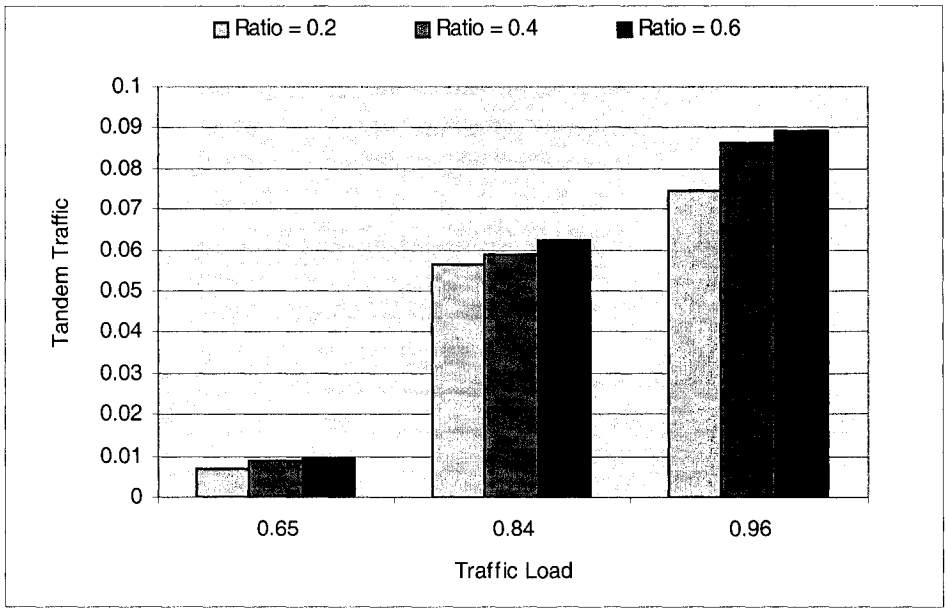


Figure 6.32 Tandem Traffic vs. Traffic Load for different $Ratio_{\alpha\_stable}$ values (Static Core Scenario)

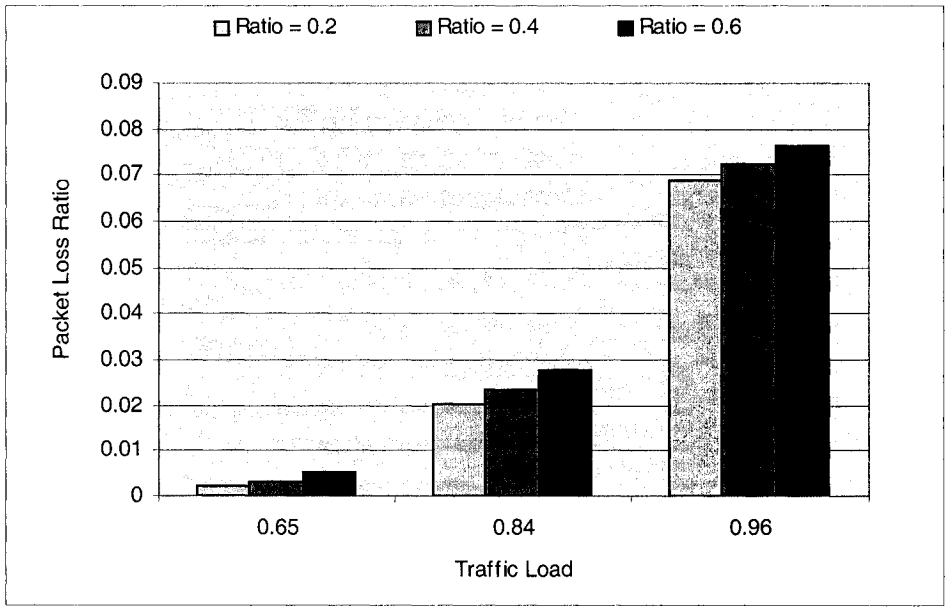


Figure 6.33 Packet Loss Ratio vs. Traffic Load for different $Ratio_{\alpha\_stable}$ values (Static Core Scenario)

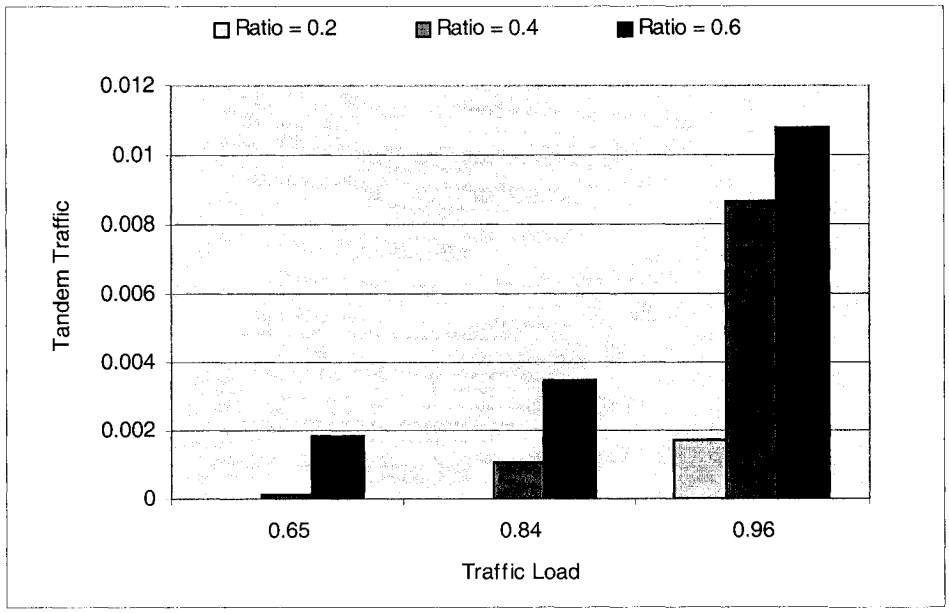


Figure 6.34 Tandem Traffic vs. Traffic Load for different $Ratio_{\alpha\_stable}$ values (reconfiguration period R= 250 ms)

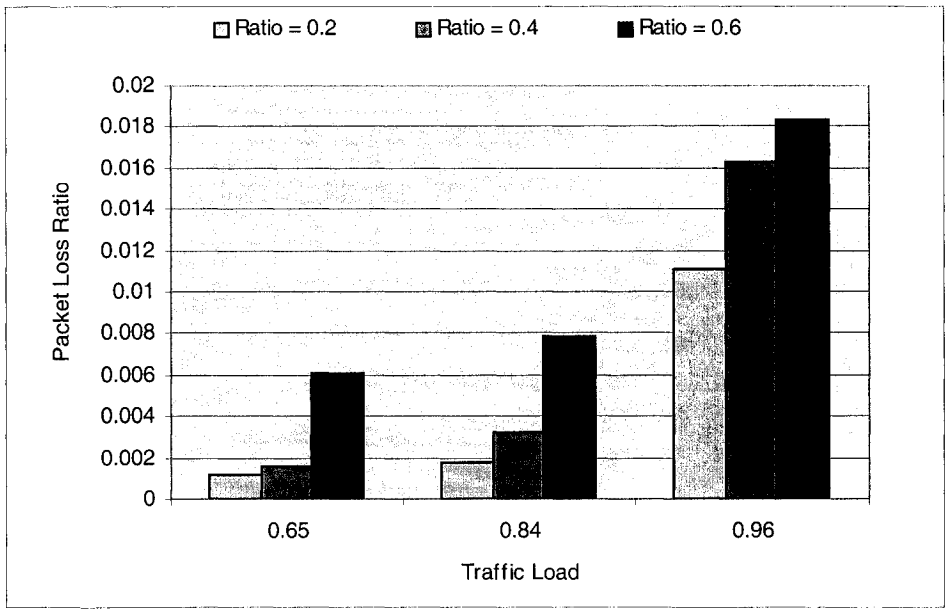


Figure 6.35 Packet Loss Ratio vs. Traffic Load for different $Ratio_{\alpha\_stable}$ values (reconfiguration period R= 250 ms)

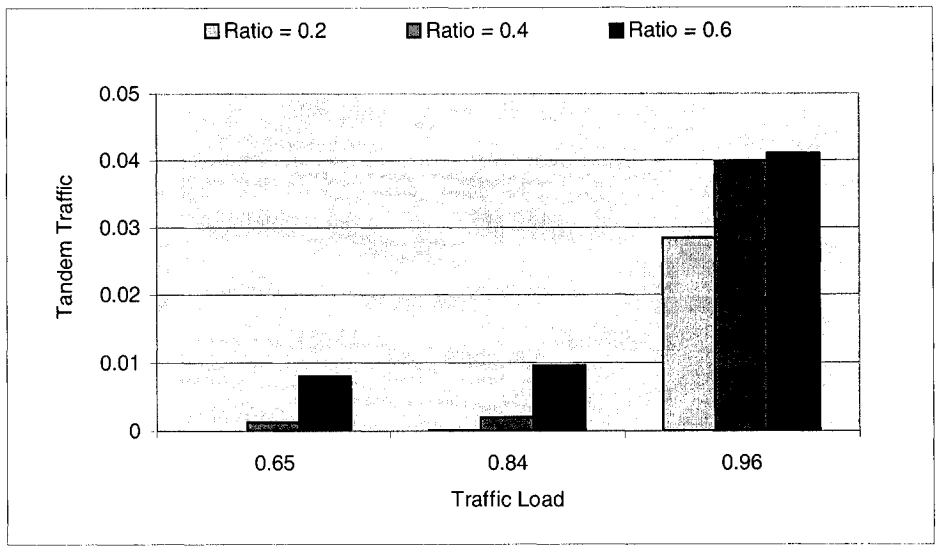


Figure 6.36 Tandem Traffic vs. Traffic Load for different $Ratio_{\alpha\_stable}$ values (reconfiguration period R= 10 S)

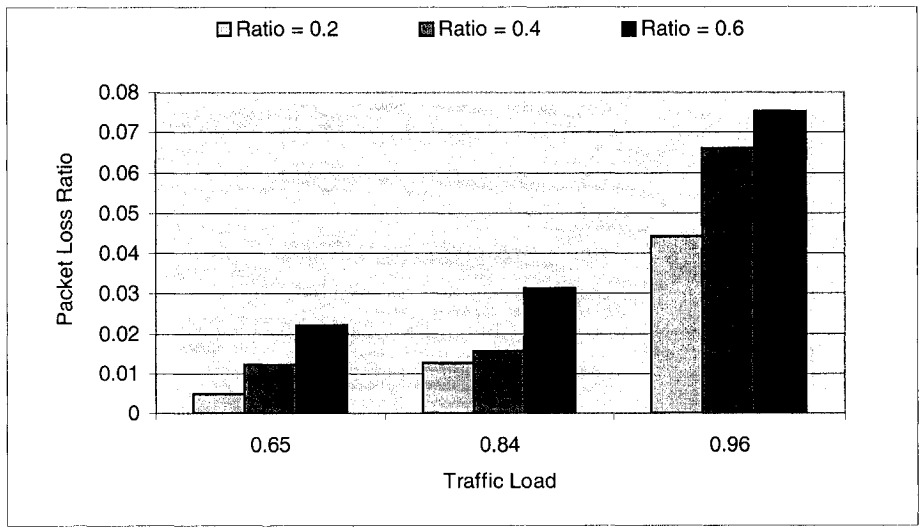


Figure 6.37 Packet Loss Ratio vs. Traffic Load for different $Ratio_{\alpha\_stable}$ values (reconfiguration period R= 10 S)

Figures 6.30 to 6.37 display the effect of different $Ratio_{\alpha\_stable}$ on the PetaWeb and static core network are shown. The results presented in the above figures show that the higher the ratio of average alpha-stable traffic to the average sinusoidal traffic is, the higher tandem traffic becomes; as a result, the network performance deteriorates. This is because the higher ratio of average alpha-stable traffic to the average sinusoidal traffic increases the

burstiness of traffic, forcing more traffic goes through tandem paths and a larger value of packets dropped. Also from figure 6.32 and figure 6.33, we can conclude that, in the static core scenario, the tandem traffic and packet loss ratio does not change too much with the ratio of average alpha-stable traffic to average sinusoidal traffic under any traffic load. On the contrary, in the PetaWeb, the tandem traffic and packet loss ratio reduces extensively with the ratio value decreasing. This demonstrates that the PetaWeb has more capability to handle the different traffic.

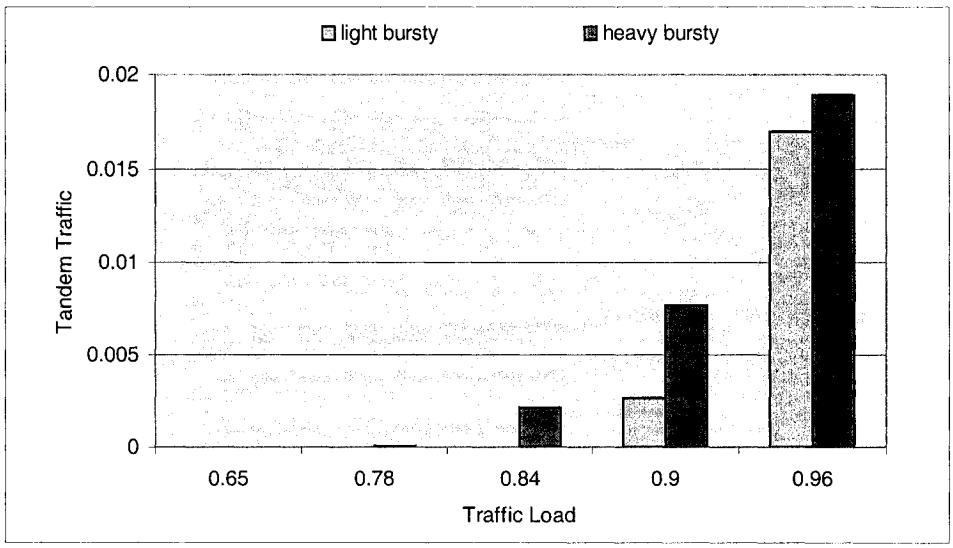


Figure 6.38 Tandem Traffic vs. Traffic Load for different burstiness level of alpha-stable traffic (reconfiguration period R = 250 ms)

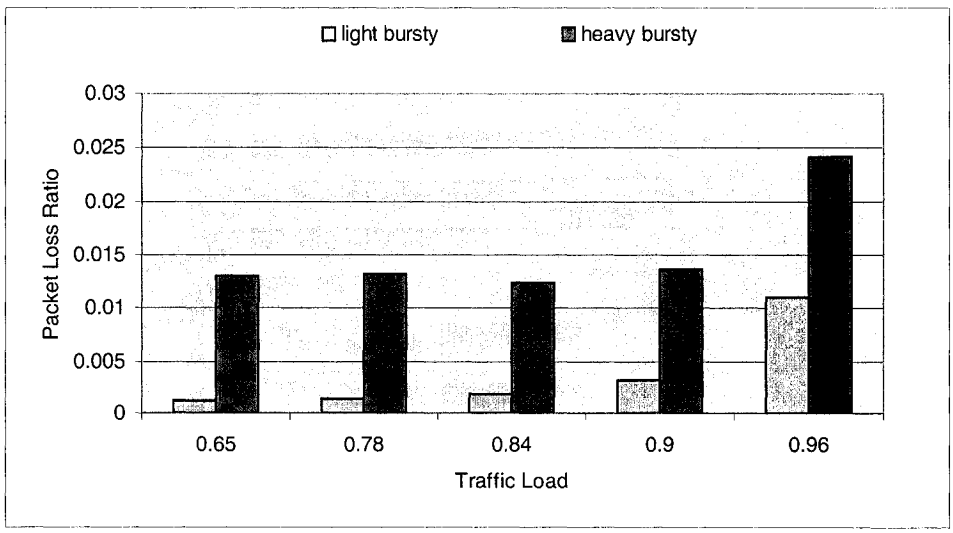


Figure 6.39 Packet Loss Ratio vs. Traffic Load for different burstiness level of alpha-stable traffic (reconfiguration period R = 250 ms)

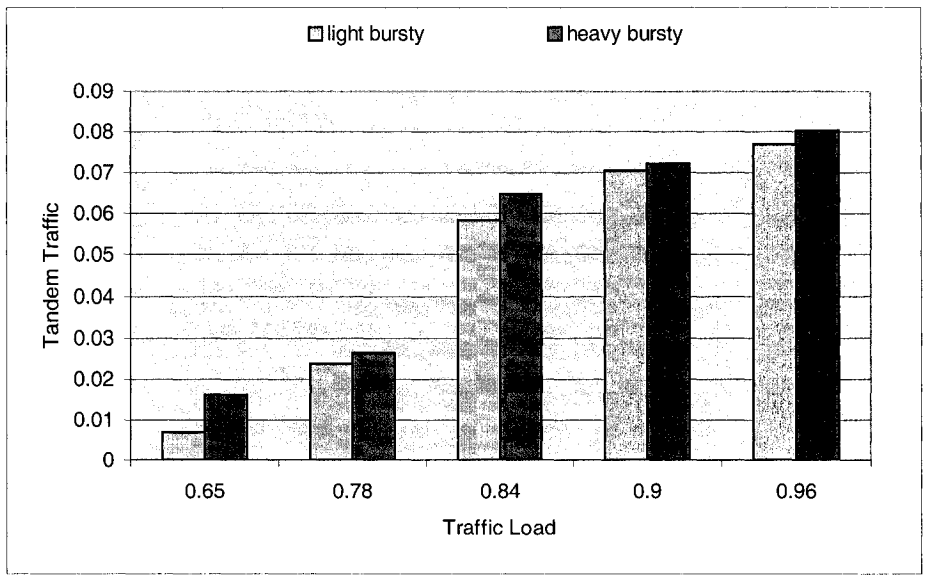


Figure 6.40 Tandem Traffic vs. Traffic Load with different bursty level of alpha-stable traffic (Static Core Scenario)

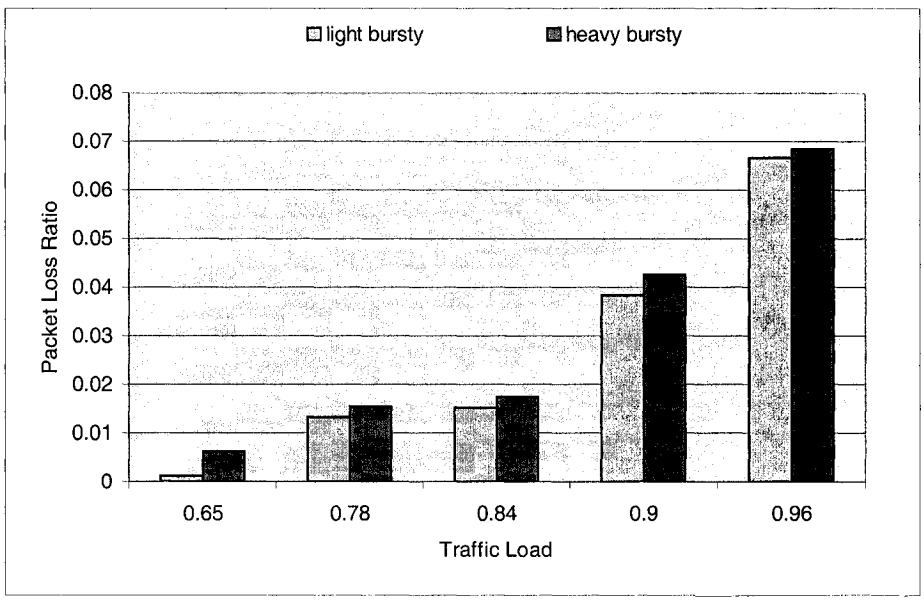


Figure 6.41 Packet Loss Ratio vs. Traffic Load for different burstiness level of alpha-stable traffic (Static Core Scenario)

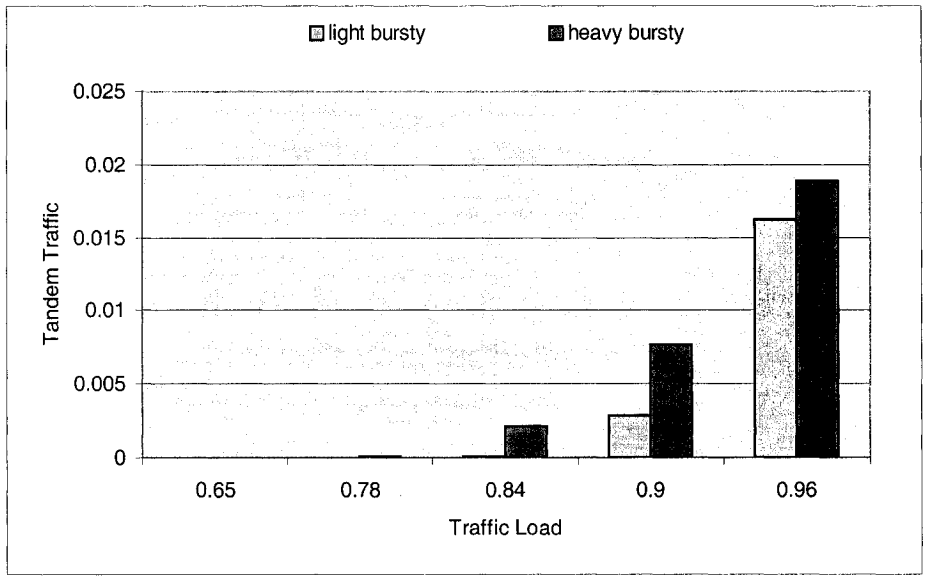


Figure 6.42 Tandem Traffic vs. Traffic Load for different burstiness level of alpha-stable traffic (reconfiguration period R = 1 S)

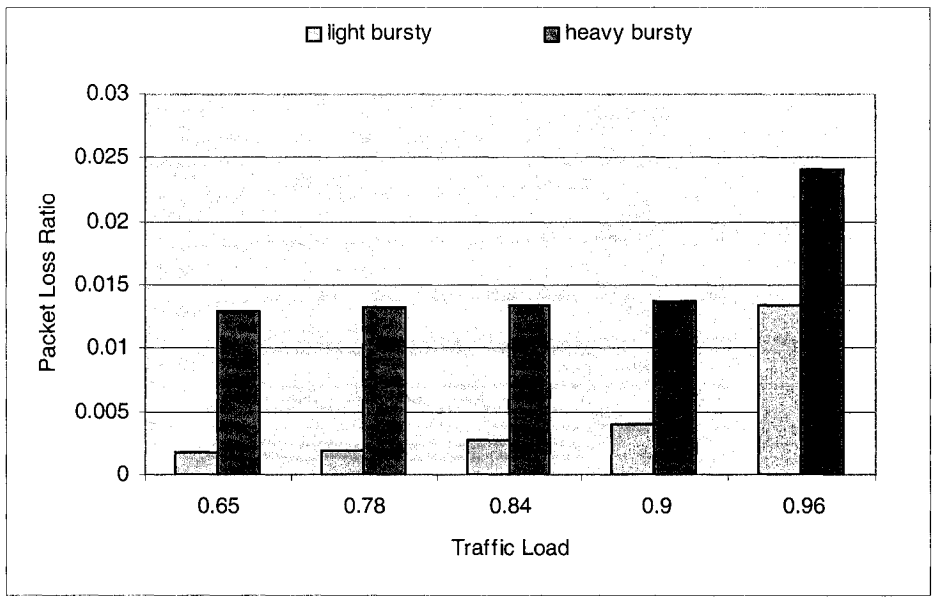


Figure 6.43 Packet Loss Ratio vs. Traffic Load for different burstiness level of alpha-stable traffic (reconfiguration period R = 1 S)

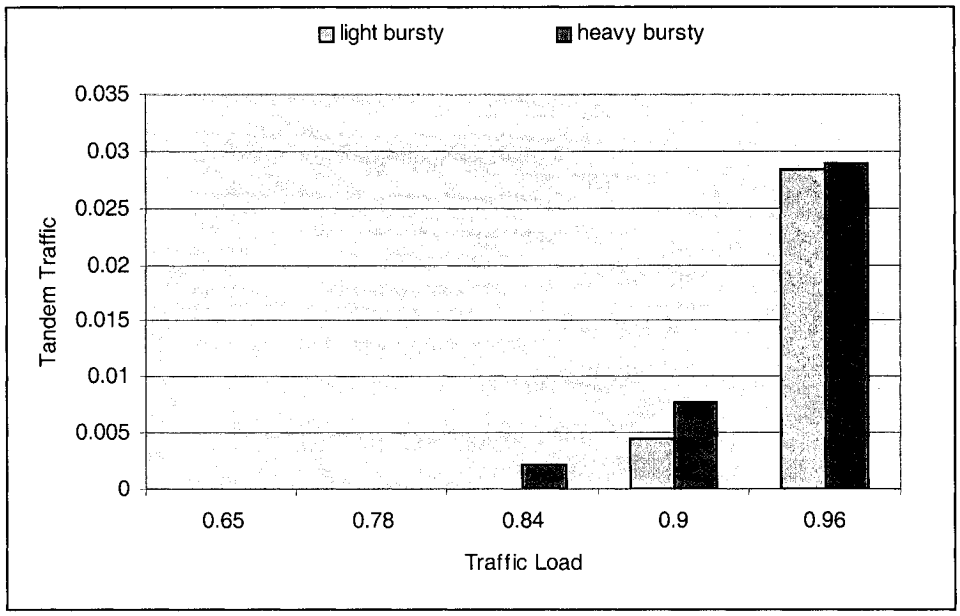


Figure 6.44 Tandem Traffic Demand vs. Traffic Load with different burstiness level of alpha-stable traffic (reconfiguration period R = 10 S)

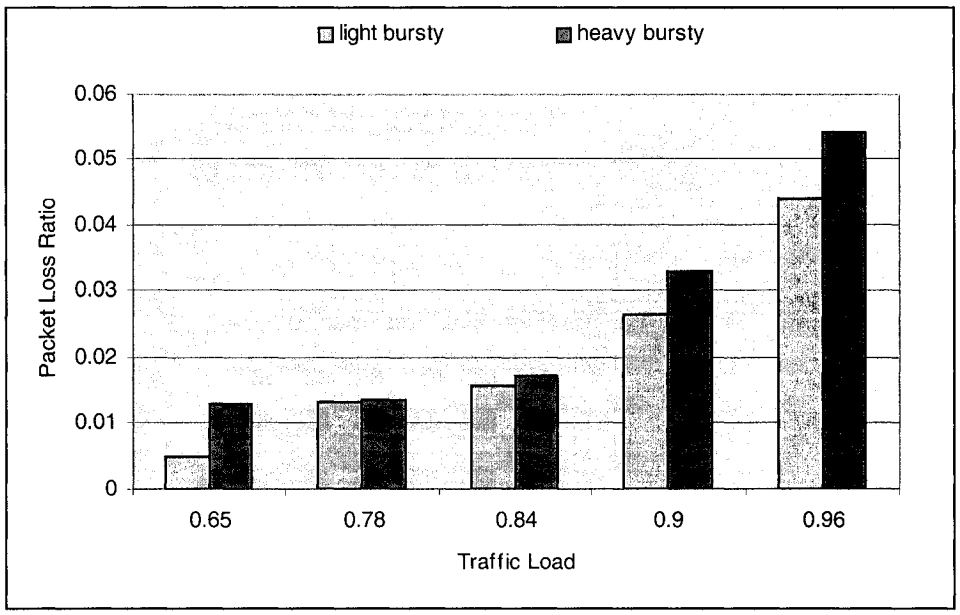


Figure 6.45 Packet Loss Ratio vs. Traffic Load for different burstiness level of alpha-stable traffic (reconfiguration period R = 10 S)

In figure 6.38 to 6.45, the effect of different burstiness level of the alpha-stable traffic on the performance of the PetaWeb and static core network is presented. From these figures, we can find the static core network and PetaWeb are all sensitive to the different bursty level of traffic. The burstier the traffic is, the higher the tandem traffic and packet

loss ratio become; thus, the worse the network performance becomes as well. The traffic bursty level reduces the agility of the PetaWeb. However, compared with the static core network, the PetaWeb still shows high performance and network utilization even with high bursty traffic.

6.5 The Effect of Reconfiguration Period

As we mentioned before, the reconfiguration period is one of the important factors affecting the performance of the PetaWeb network. The reconfiguration period, *R*, is the interval of time between the successive network channel reallocation processes according to the traffic demand at the source edge nodes. The reconfiguration period is limited by the propagation delay of the longest core-to-edge roundtrip time, due to control message transmitted to and from the core controller. It should be larger than the longest RTT between edge nodes. The shorter reconfiguration period, the more responsive the PetaWeb network becomes to traffic changes, reducing the tandem traffic and packet loss ratio. In this section, we examine in more detail the effect of the PetaWeb reconfiguration period on the network performance. Three traffic sources (*alpha-stable traffic, Poisson traffic, and Sinusoidal traffic*) are used to examine the sensitivity of the effective capacity of the PetaWeb to the reconfiguration period.

In the experiments, the reconfiguration periods are set as 250ms, 500ms, 1s, 2s, 5s, 10s respectively for the scenarios. In the first set of experiments, we use alpha-stable traffic combined with sinusoidal traffic as the network input source. The traffic volatility is set as 0.6; $Ratio_{\alpha\_stable}$ is set to 0.2. In the experiment, we set traffic load as 0.65, 0.84 and 0.96 respectively. In the second set of experiments, we replace the alpha-stable traffic with Poisson traffic and keep the parameters the same. In the third set of experiments, we use alpha-stable traffic combined with sinusoidal traffic as the network input source. The traffic volatility is set as 0.4, 0.6, and 0.8 respectively; the traffic load is fixed to 0.65, 0.84, and 0.96, $Ratio_{\alpha\_stable}$ is set to 0.2. In the fourth set of experiments, we use alpha-stable traffic combined with sinusoidal traffic as the network input source. The traffic volatility is set as 0.6; the traffic load is fixed to 0.65, 0.84, and 0.96, but $Ratio_{\alpha\_stable}$ is changed to 0.2, 0.4, and 0.6 respectively. Figure 6.45 to figure 6.56 illustrate the tandem traffic demand for

different reconfiguration periods under different traffic loads and different traffic characteristics.

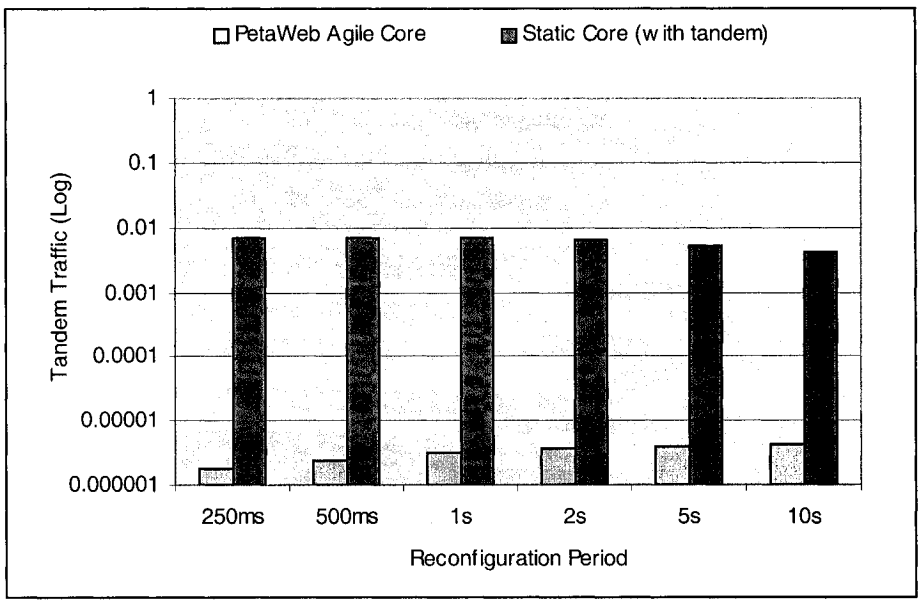


Figure 6.46 Tandem Traffic vs. Reconfiguration Period (with *combined α-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.65)

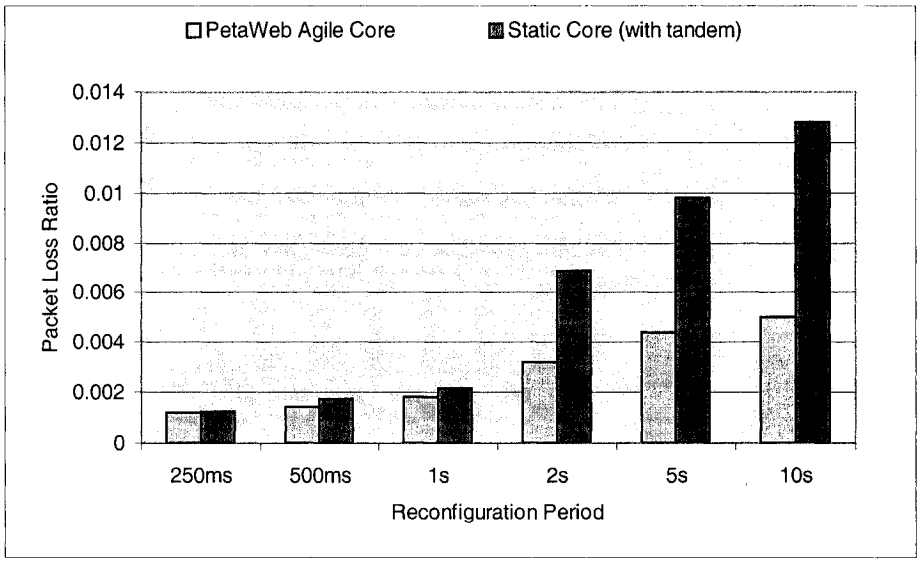


Figure 6.47 Packet Loss Ratio vs. Reconfiguration Period (with *combined α-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.65)

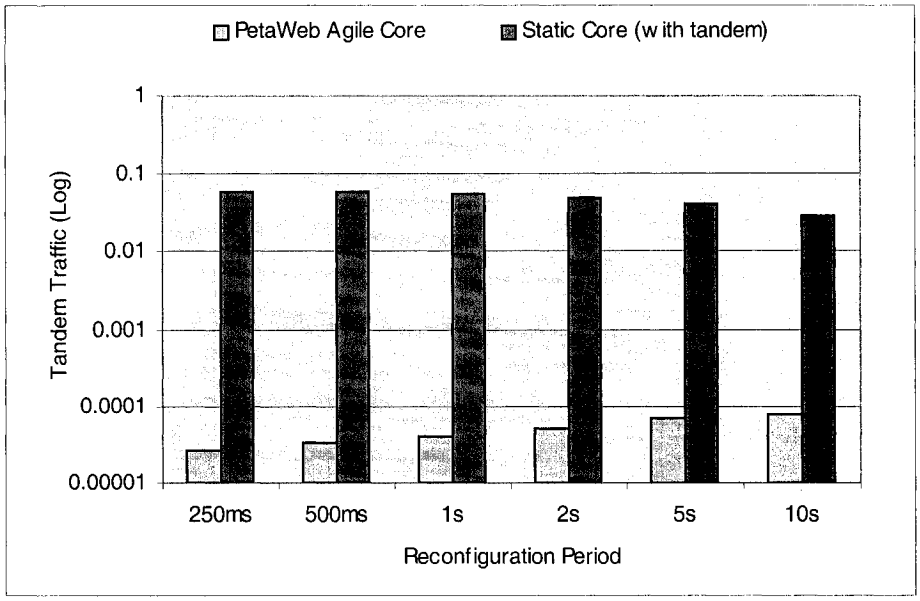


Figure 6.48 Tandem Traffic vs. Reconfiguration Period (with *combined α-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.84)

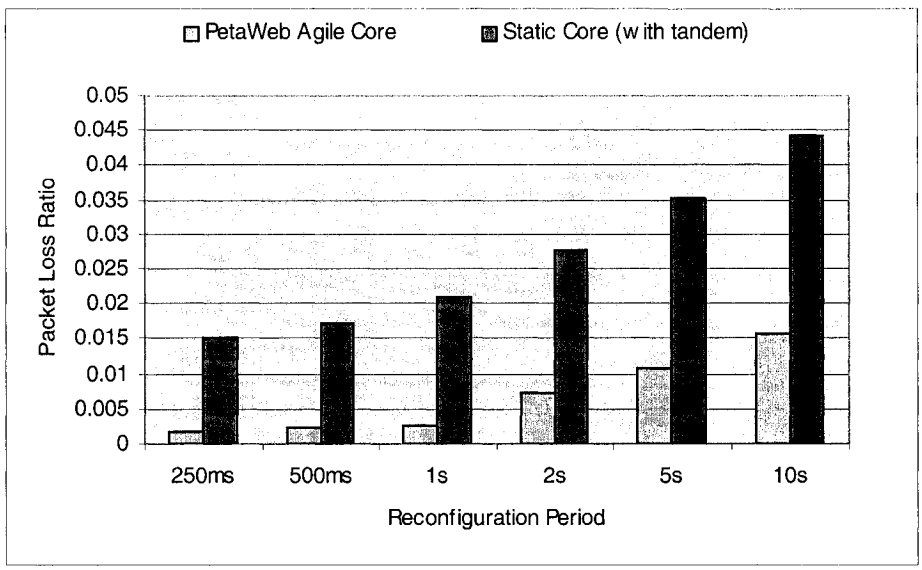


Figure 6.49 Packet Loss Ratio vs. Reconfiguration Period (with *combined α-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.84)

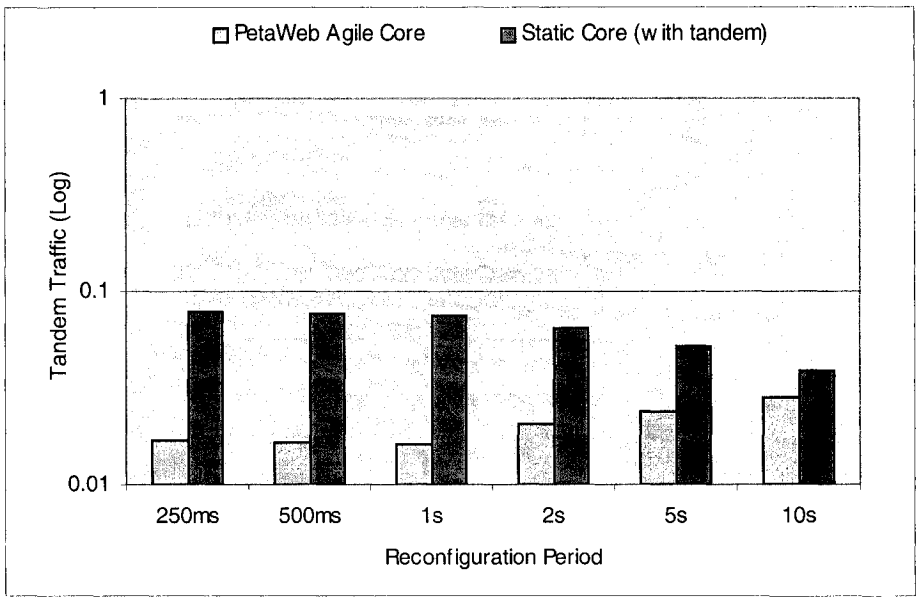


Figure 6.50 Tandem Traffic vs. Reconfiguration Period (with *combined α-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.96)

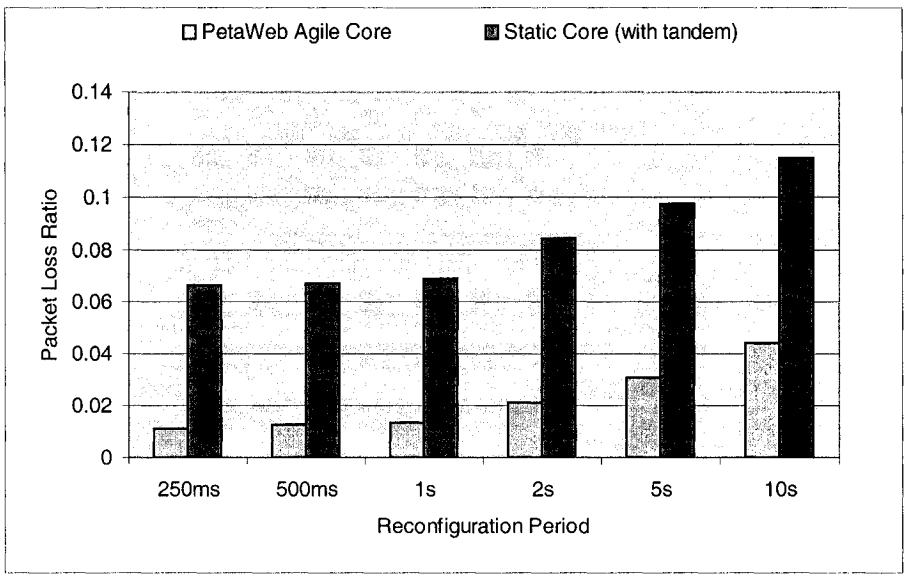


Figure 6.51 Packet Loss Ratio vs. Reconfiguration Period (with *combined α-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.96)

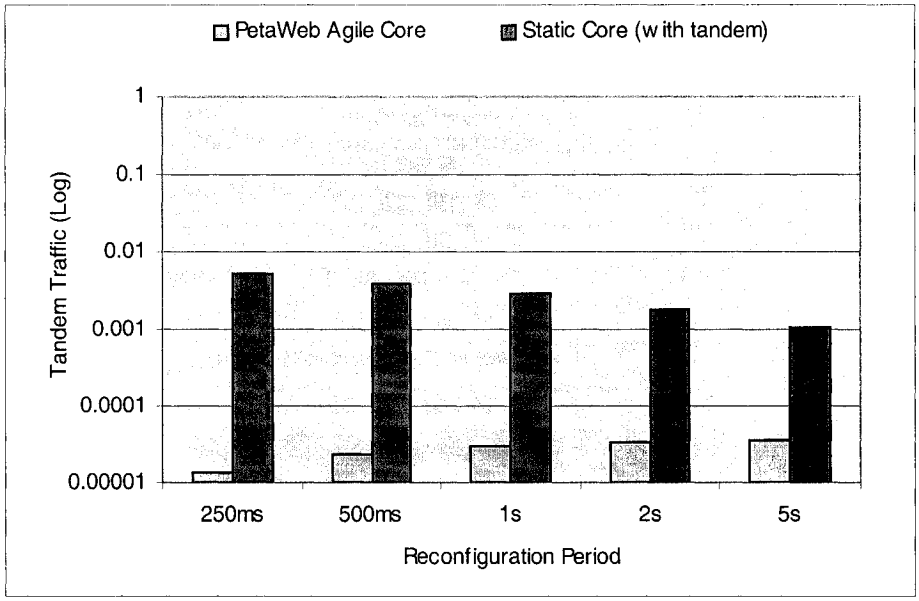


Figure 6.52 Tandem Traffic vs. Reconfiguration Period (with *combined Poisson traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.65)

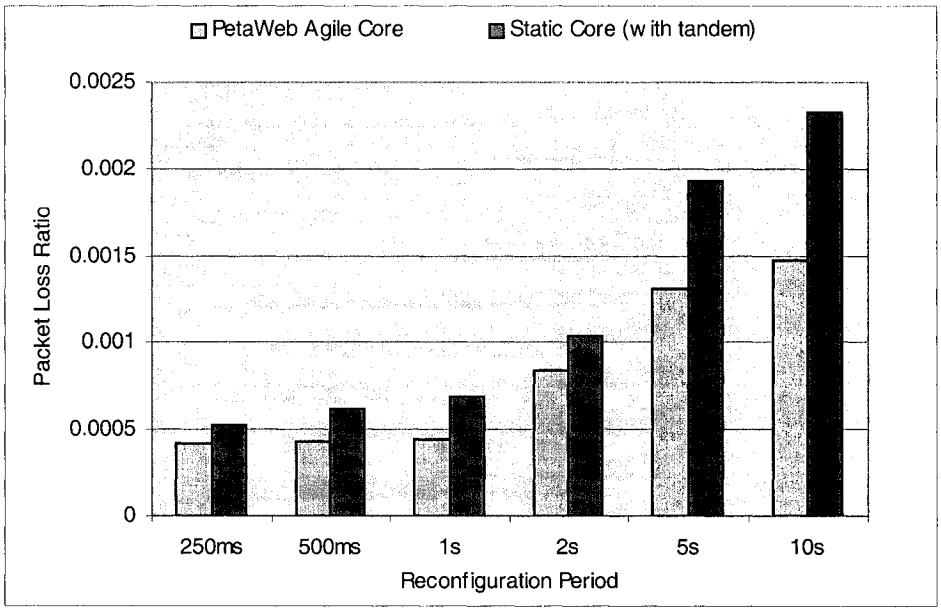


Figure 6.53 Packet Loss Ratio vs. Reconfiguration Period (with *combined Poisson traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.65)

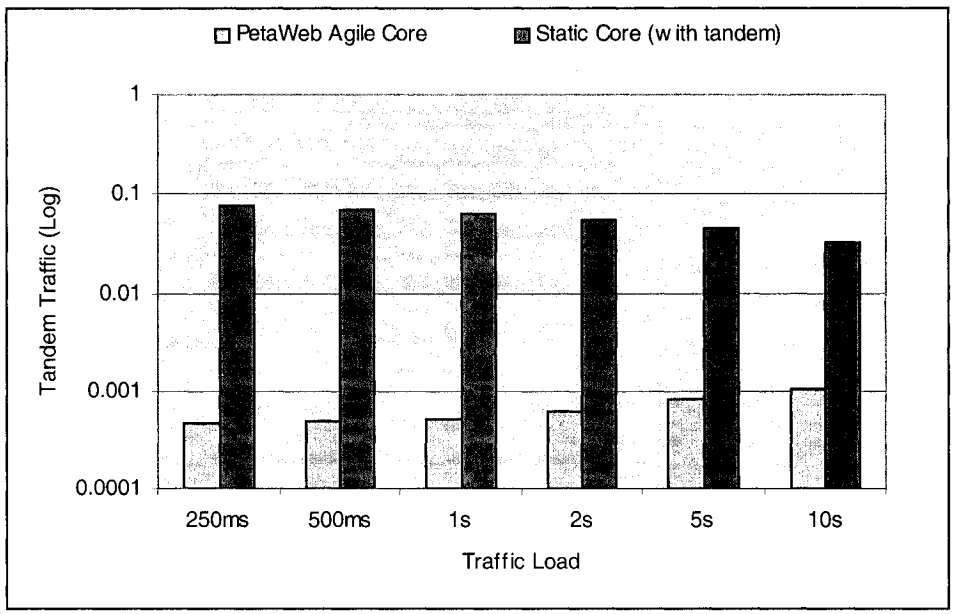


Figure 6.54 Tandem Traffic vs. Reconfiguration Period (with *combined Poisson traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.84)

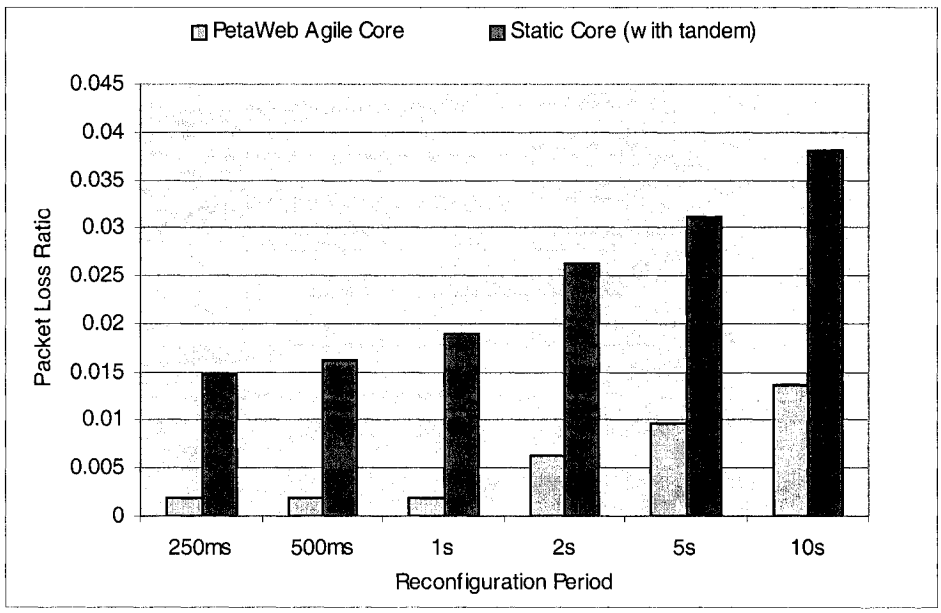


Figure 6.55 Packet Loss Ratio vs. Reconfiguration Period (with *combined Poisson traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.84)

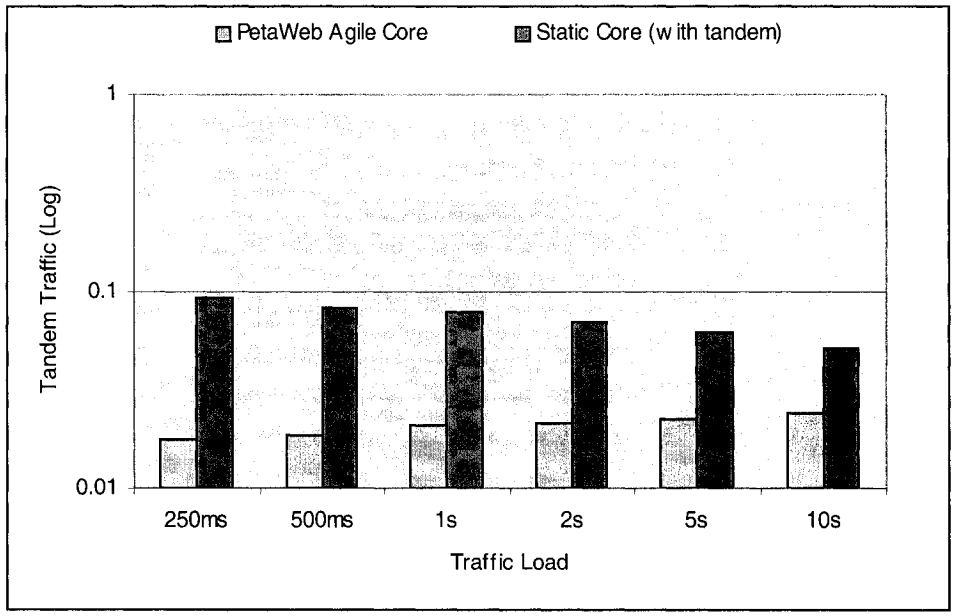


Figure 6.56 Tandem Traffic vs. Reconfiguration Period (with *combined Poisson traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.96)

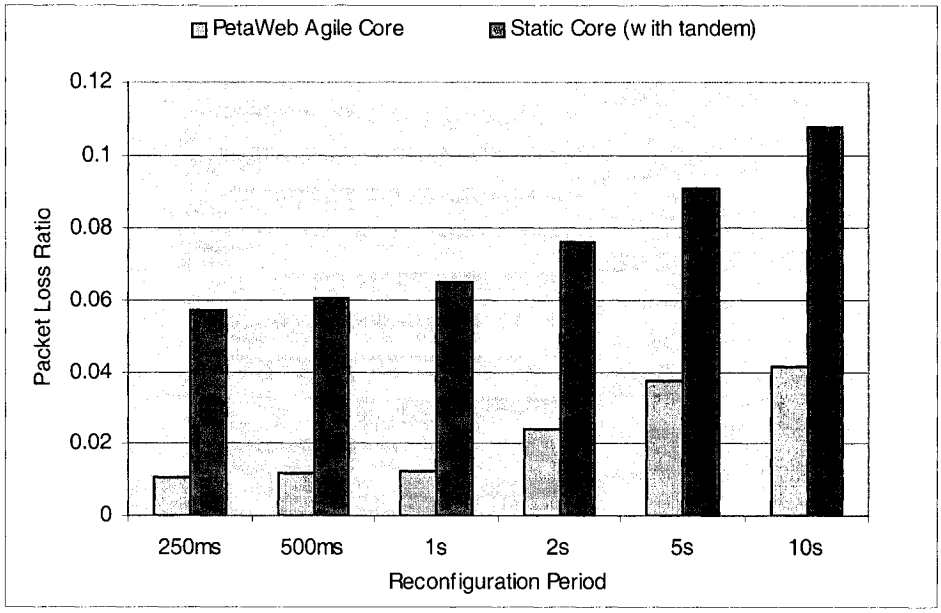


Figure 6.57 Packet Loss Ratio vs. Reconfiguration Period (with *combined Poisson traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.96)

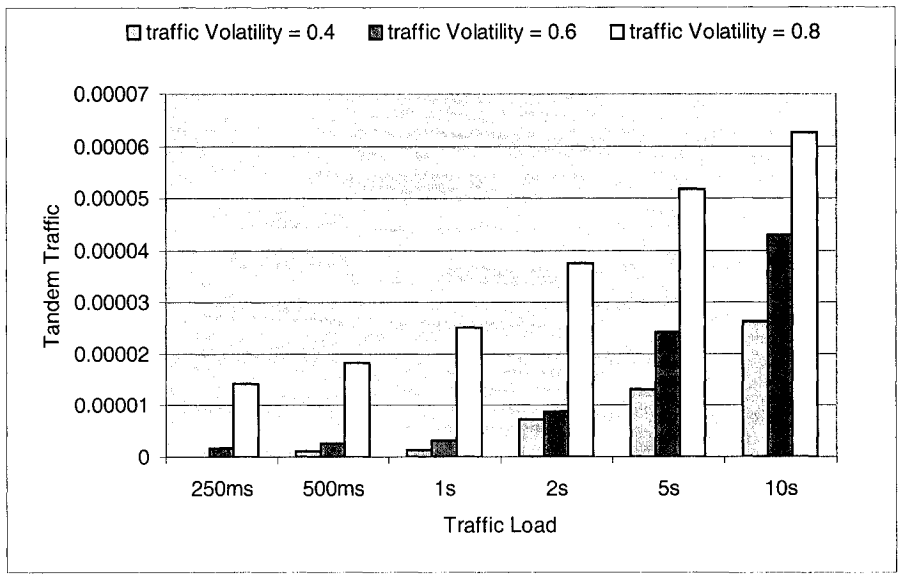


Figure 6.58 Tandem Traffic vs. Reconfiguration Period (for *combined α-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, *traffic volatility* is 0.4, 0.6, 0.8 respectively, traffic load is 0.65)

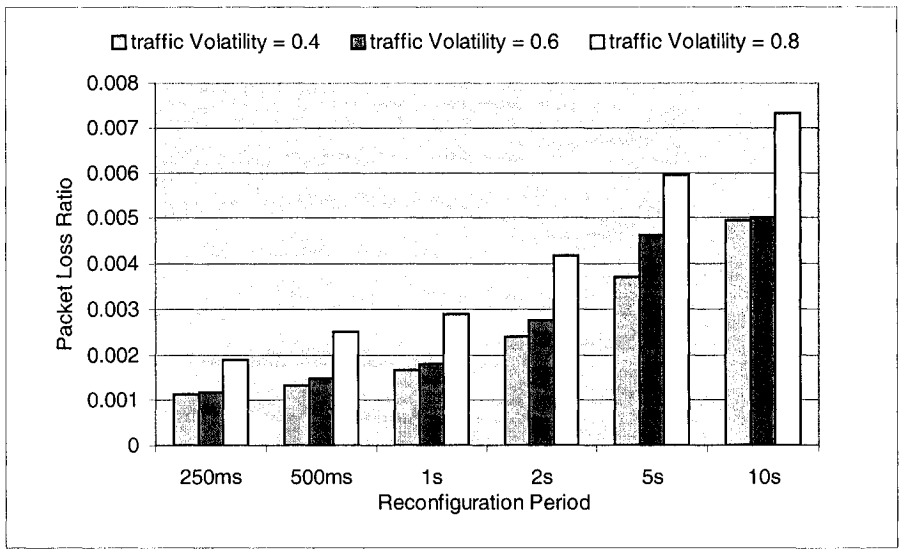


Figure 6.59 Packet Loss Ratio vs. Reconfiguration Period (for *combined α-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, *traffic volatility* is 0.4, 0.6, 0.8 respectively, traffic load is 0.65)

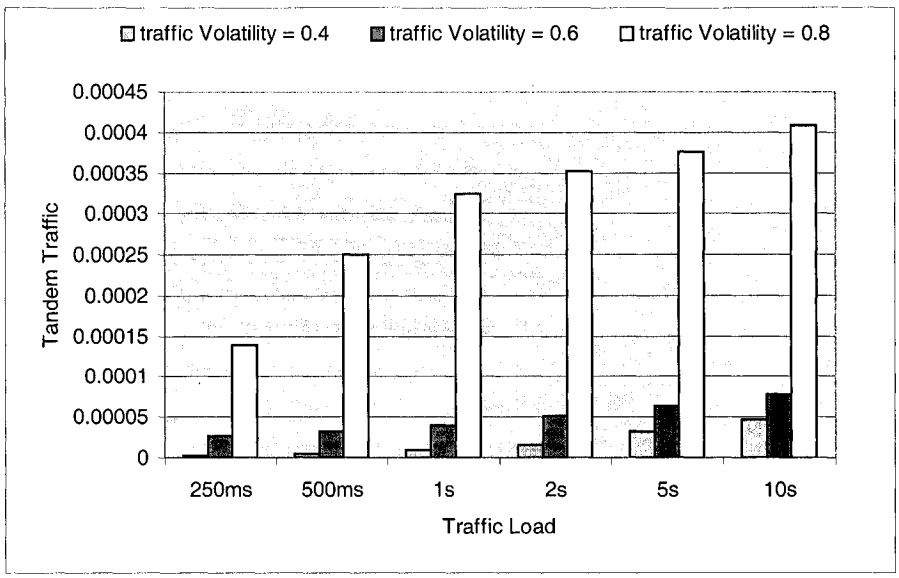


Figure 6.60 Tandem Traffic vs. Reconfiguration Period (for *combined α-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, *the traffic volatility* is 0.4, 0.6, 0.8 respectively, traffic load is 0.84)

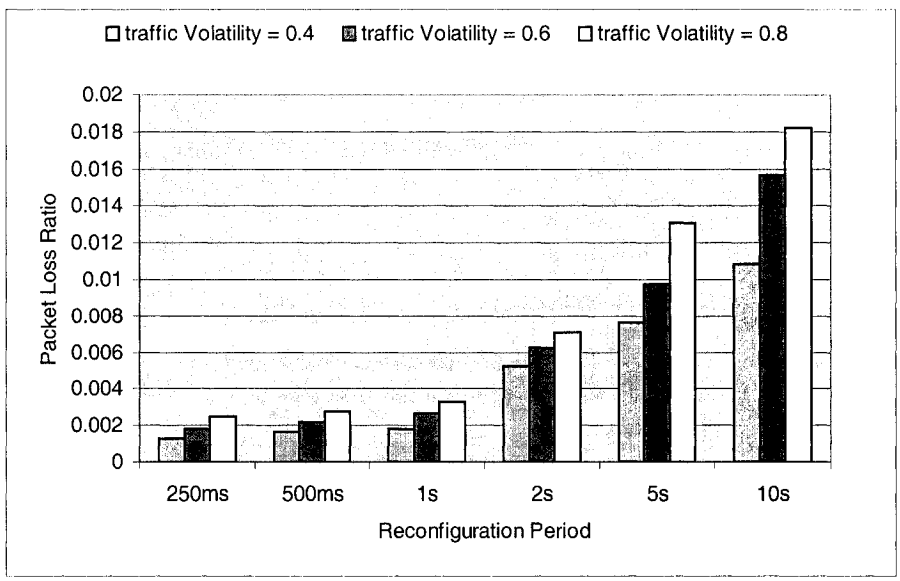


Figure 6.61 Packet Loss Ratio vs. Reconfiguration Period (for *combined α-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, *the traffic volatility* is 0.4, 0.6, 0.8 respectively, traffic load is 0.84)

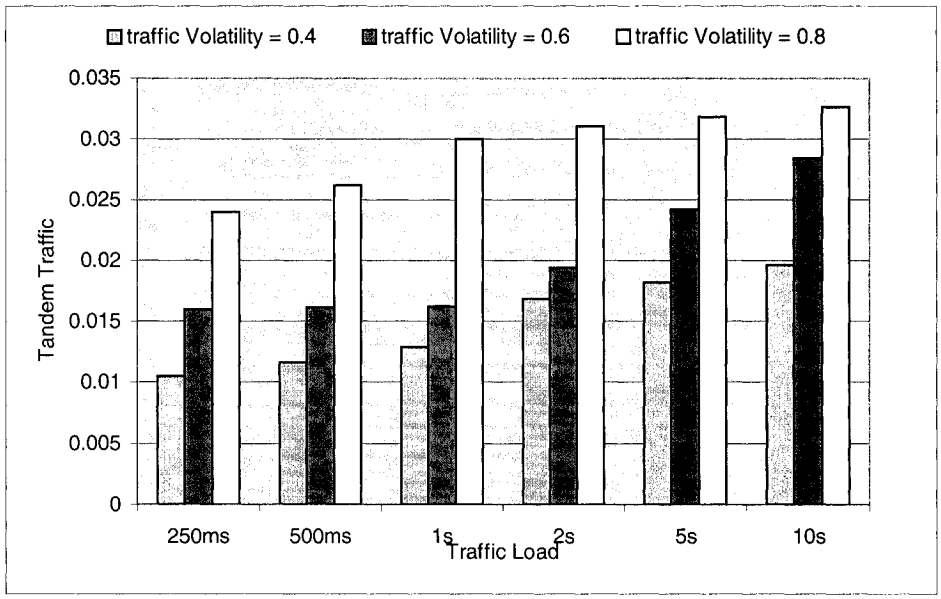


Figure 6.62 Tandem Traffic vs. Reconfiguration Period (for *combined α-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, *the traffic volatility* is 0.4, 0.6, 0.8 respectively, traffic load is 0.96)

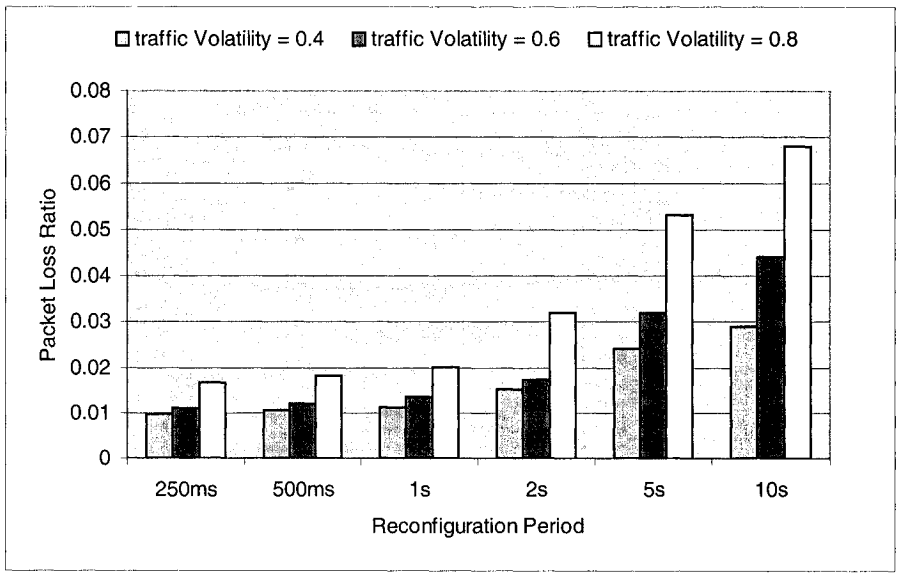


Figure 6.63 Packet Loss Ratio vs. Reconfiguration Period (for *combined α-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, *the traffic volatility* is 0.4, 0.6, 0.8 respectively, traffic load is 0.96)

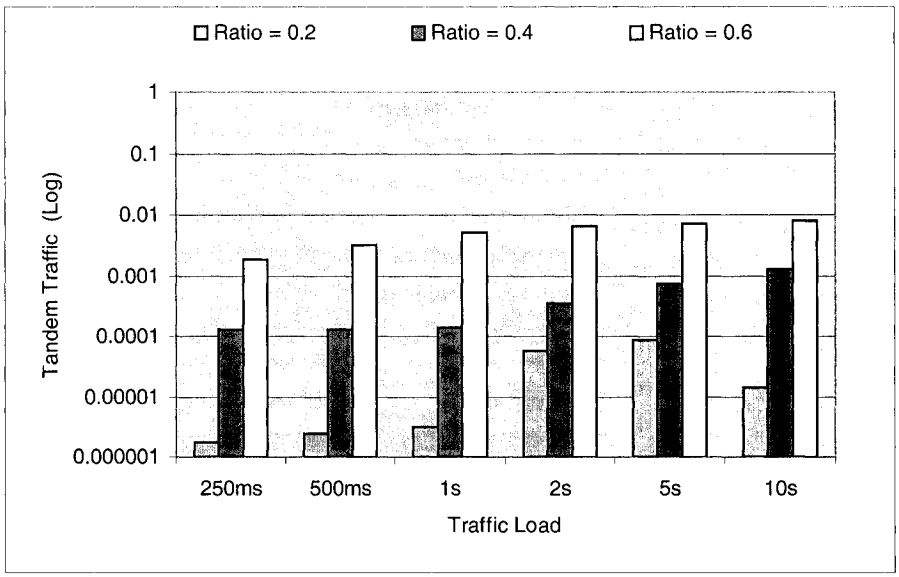


Figure 6.64 Tandem Traffic vs. Reconfiguration Period (with *combined α-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, 0.4 and 0.6, *the traffic volatility* is 0.6, traffic load is 0.65)

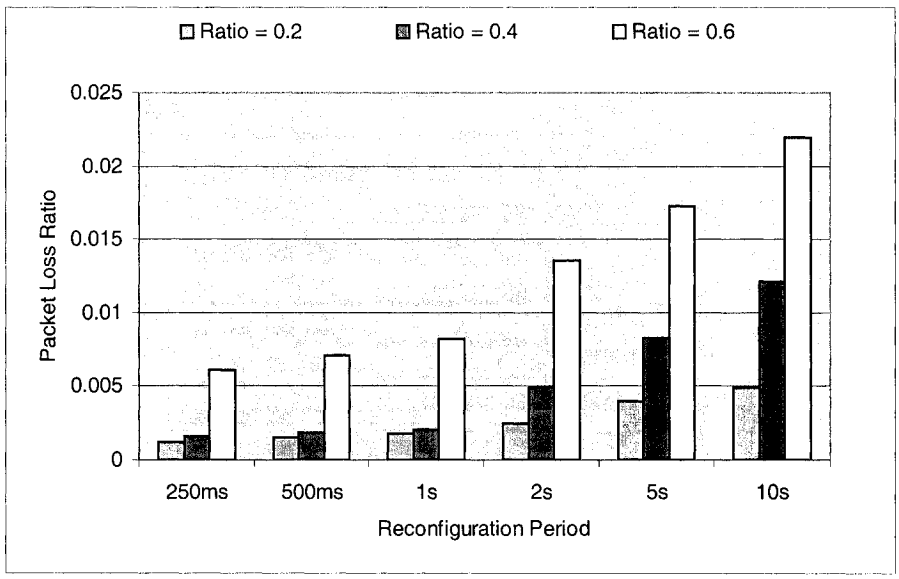


Figure 6.65 Packet Loss Ratio vs. Reconfiguration Period (with *combined α-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, 0.4 and 0.6, *the traffic volatility* is 0.6, traffic load is 0.65)

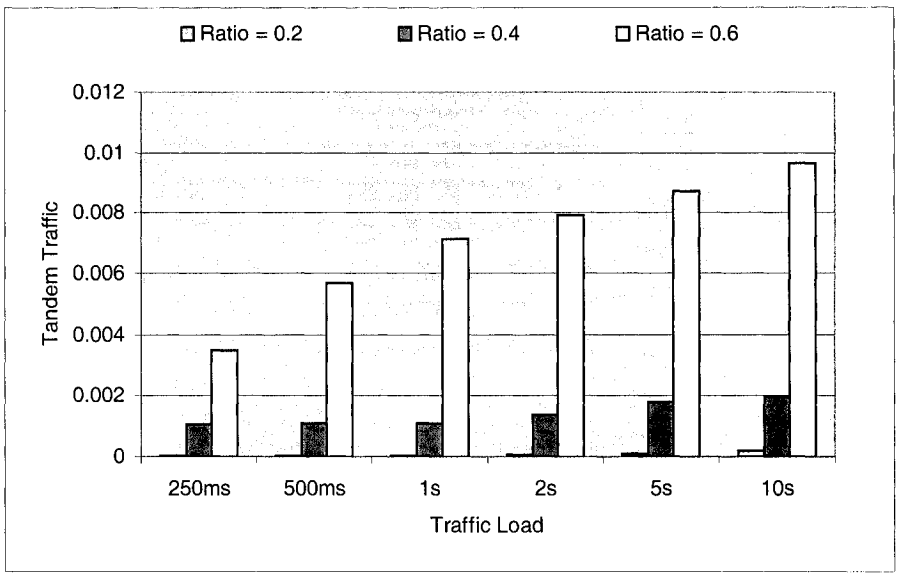


Figure 6.66 Tandem Traffic vs. Reconfiguration Period (with *combined α-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, 0.4 and 0.6, *the traffic volatility* is 0.6, traffic load is 0.84)

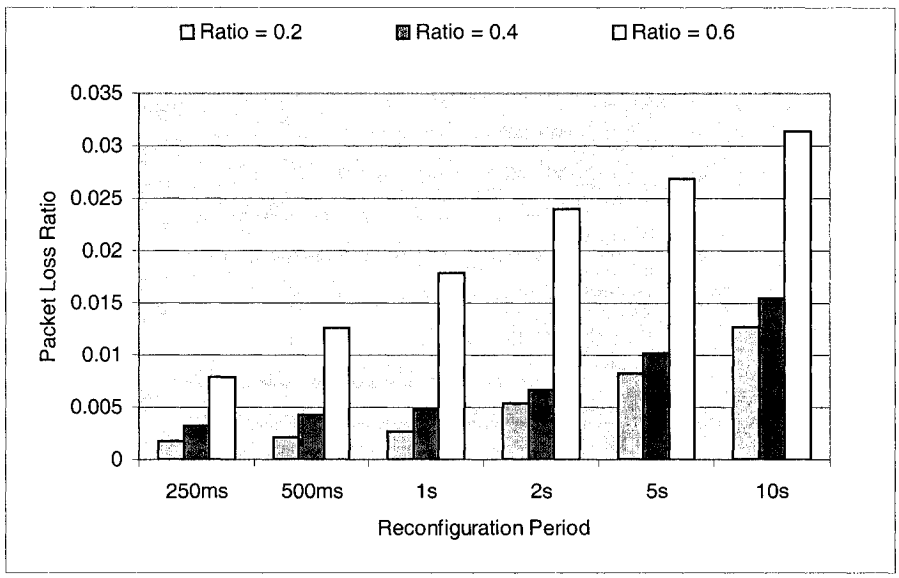


Figure 6.67 Packet Loss Ratio vs. Reconfiguration Period (with *combined α-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, 0.4 and 0.6, *the traffic volatility* is 0.6, traffic load is 0.84)

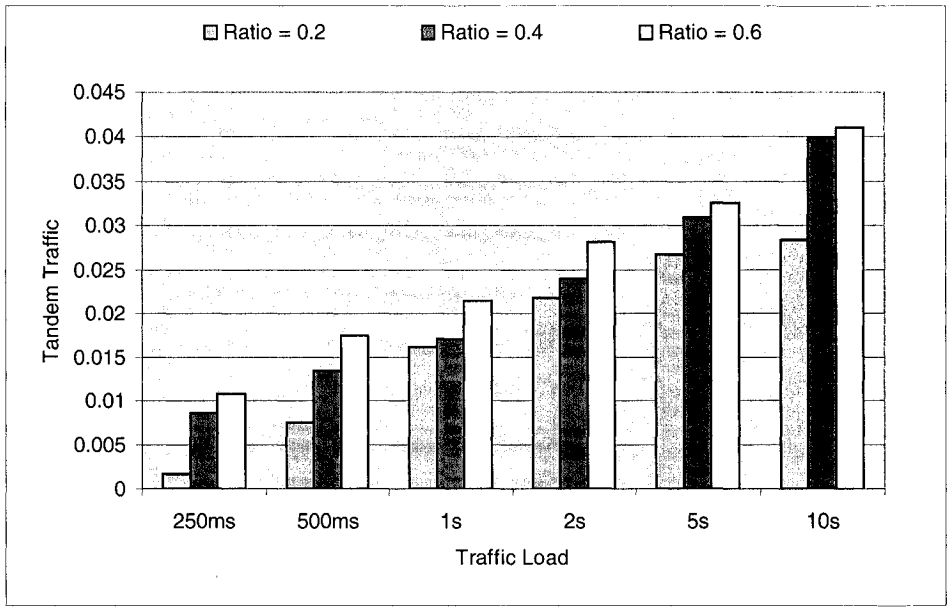


Figure 6.68 Tandem Traffic vs. Reconfiguration Period (with *combined α-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, 0.4 and 0.6, *the traffic volatility* is 0.6, traffic load is 0.96)

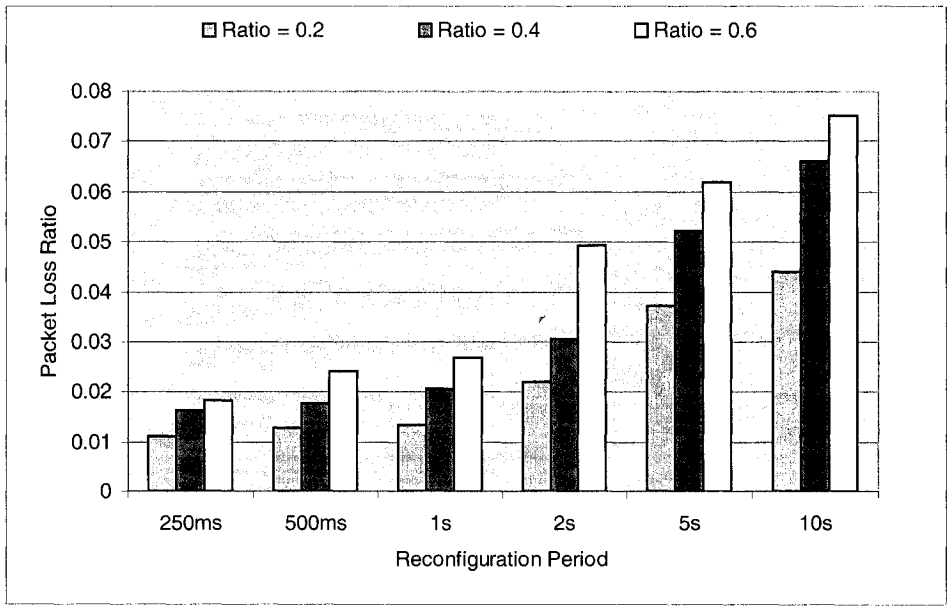


Figure 6.69 Packet Loss Ratio vs. Reconfiguration Period (with *combined α-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, 0.4 and 0.6, *the traffic volatility* is 0.6, traffic load is 0.96)

It is obvious from the figure 6.46 to figure 6.57 that for the PetaWeb agile core network, the shorter reconfiguration period, the less the tandem traffic and the lower the packet loss ratio, regardless of the traffic type. Conversely, for the static core (with tandem)

network, the shorter reconfiguration period, the higher the tandem traffic. However, the packet loss ratio decreases with the shorter reconfiguration period. This is because the shorter the reconfiguration period, the better the PetaWeb can response to the traffic fluctuation. By comparing the results under different traffic load, figure 6.46 to figure 6.51 for alpha-stable combined with sinusoidal traffic source scenario and figure 6.52 to figure 6.57 for Poisson combined with sinusoidal traffic source scenario, we can find that in any input traffic sources, the higher traffic load, the more significant effect of the reconfiguration period for the PetaWeb, compared with the static core network. On the other hand, reconfigurations consume computational resources and increases system cost; thus, the shorter reconfiguration period, the more network cost. Another key factor impacting the reconfiguration period is the network size. Because the reconfiguration period should be larger than the longest RTT between the edge nodes, so the shorter reconfiguration period reduces the size of network coverage area. As a result, the reconfiguration should be performed only when needed. We should choose reconfiguration period based on the network history data. In figure 6.58 to figure 6.63, the effect of different traffic volatility on the network performance is presented. The higher the traffic volatility is, the higher tandem traffic becomes, and thus the network performance deteriorates; the lower traffic load, the more significant the effect of the traffic volatility on the tandem traffic. In figure 6.64 to figure 6.69, the effects of reconfiguration period for different ratios of α-stable traffic to sinusoidal traffic are compared. The larger ratio value, the more proportion of alpha-stable traffic to the sinusoidal traffic. When the ratio increases, the tandem traffic rises. However, under any traffic volatility and any $Ratio_{\alpha\_stable}$, the shorter reconfiguration period, the less the tandem traffic and packet loss ratio become, thus the better network performance becomes as well.

6.6 The Effect of Buffer Size at the Edge Node

As mentioned earlier, in the PetaWeb network, the optical core is bufferless. However, the edge nodes should have some buffer at the electrical side in order to avoid data loses resulting from the burstiness of the traffic. At the same time, large buffers can cause large delay and delay jitter; balancing loses and delay, keeping the buffer size at a reasonable level is important. We investigate the effect of the buffer size on the performance of the static and agile core scenario. Our analysis addresses first the effect on the tandem traffic; then we address the case of packet losses, finally the effect on the average end-to-end delay and delay jitter are addressed.

The traffic used in this set of experiments is the alpha-stable self-similar traffic combined with the sinusoidal traffic. In order to get a better idea how the buffer size at edge core node affects the PetaWeb performance, different traffic characteristics and parameters, different values of traffic load, and different reconfiguration periods are considered and used in the experiments. In all scenarios, two network configurations the agile core and static core are examined; two different reconfiguration periods (250ms, 1 s) are evaluated. In the first set of experiments, the effect of different buffer size on the network performance under different values of traffic load (0.65, 0.77, 0.84, 0.9,and 0.96) is examined. The buffer size is set (in all edge nodes) as 2, 3, 10, 50, 100, 200, and 1000 respectively; the traffic volatility is set as 0.6; $Ratio_{\alpha\_stable}$ is set at 0.2. In the second set of experiments, the effect of different buffer sizes on the network performance under different traffic characteristics and parameters are examined. We change $Ratio_{\alpha\_stable}$ at 0.2, 0.4, and 0.6 respectively, and set the traffic load at 0.84. The traffic volatility is set at 0.6. The buffer sizes are: 2, 3, 10, 50, 100, 200, and 1000 respectively. The figures generated through the simulation are presented as below.

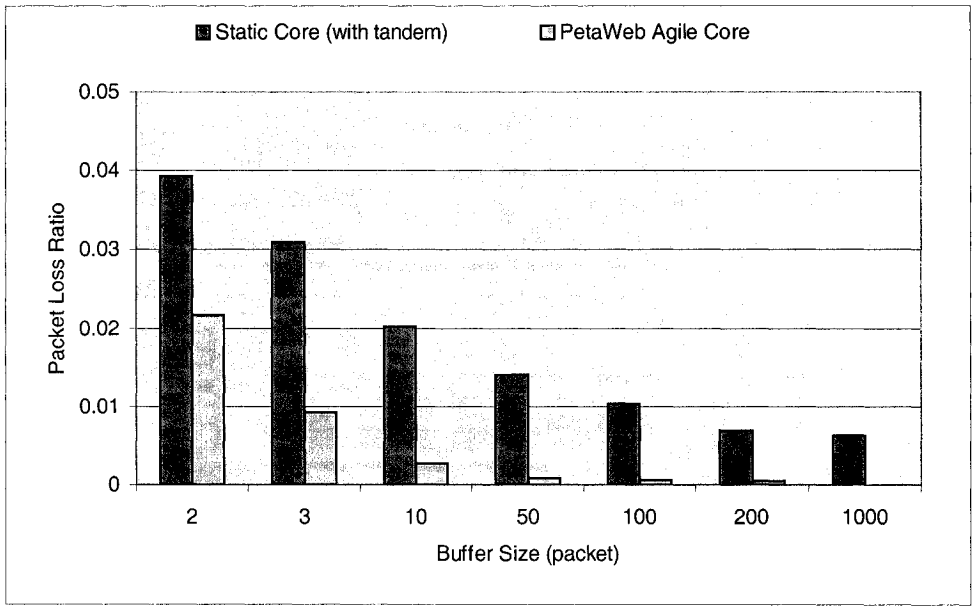


Figure 6.70 Packet Loss Ratio vs. Buffer Size (for *combined alpha-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.84, reconfiguration period is 1 S)

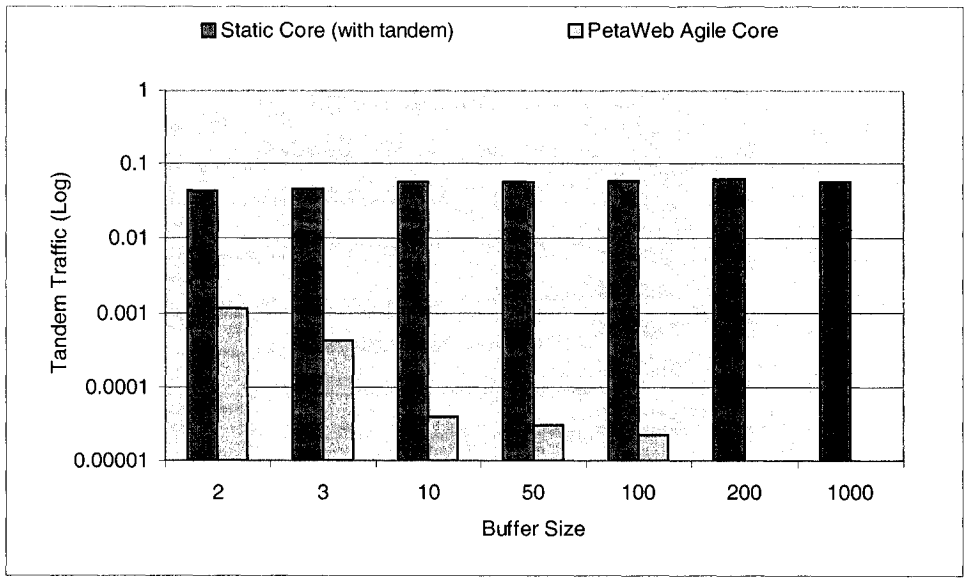


Figure 6.71 Tandem Traffic vs. Buffer Size (for *combined alpha-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.84, reconfiguration period is 1 S)

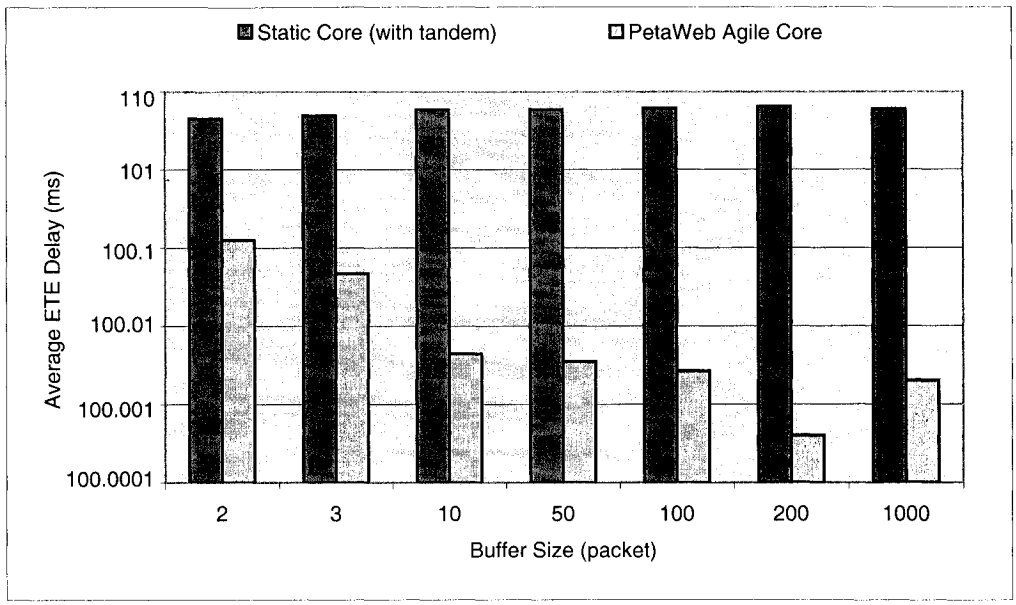


Figure 6.72 Average end-to-end Delay vs. Buffer Size (for *combined alpha-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.84, reconfiguration period is 1 S)

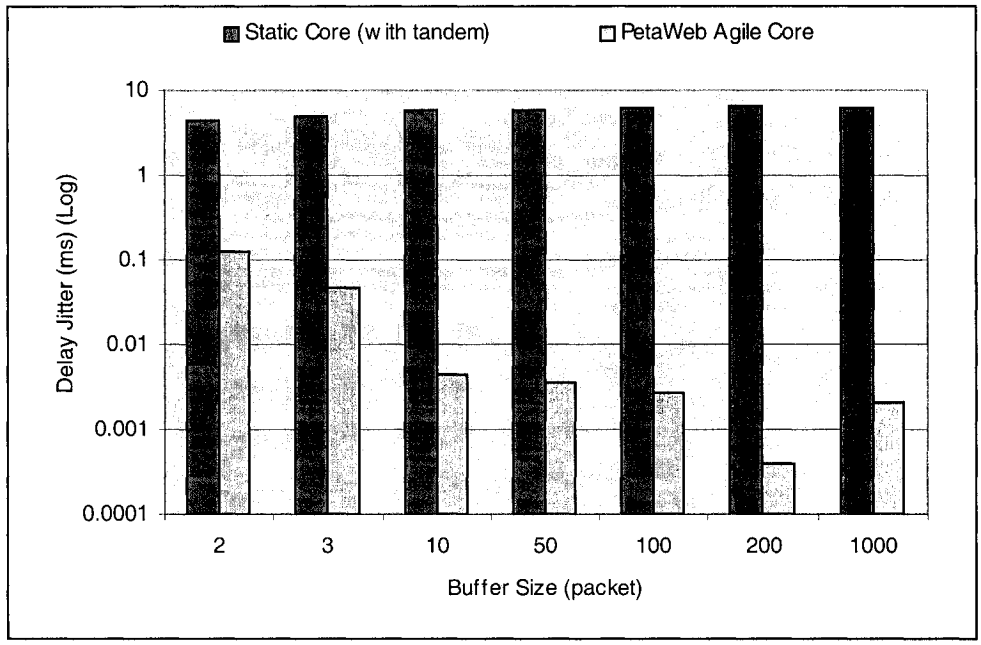


Figure 6.73 Delay Jitter vs. Buffer Size (for *combined alpha-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.84, reconfiguration period is 1 S)

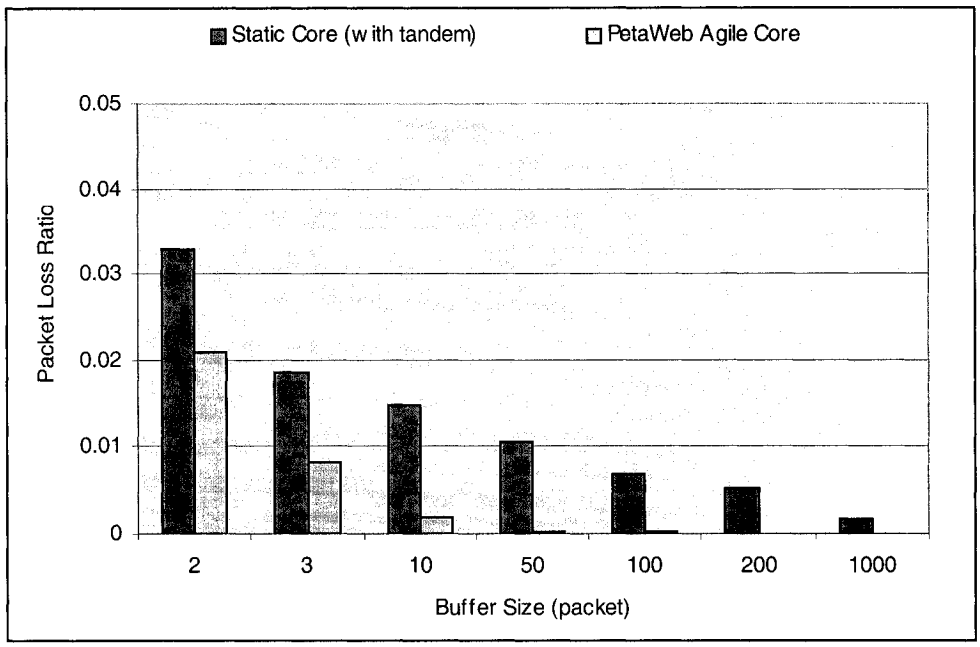


Figure 6.74 Packet Loss Ratio vs. Buffer Size (for *combined alpha-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.84, reconfiguration period is 250 ms)

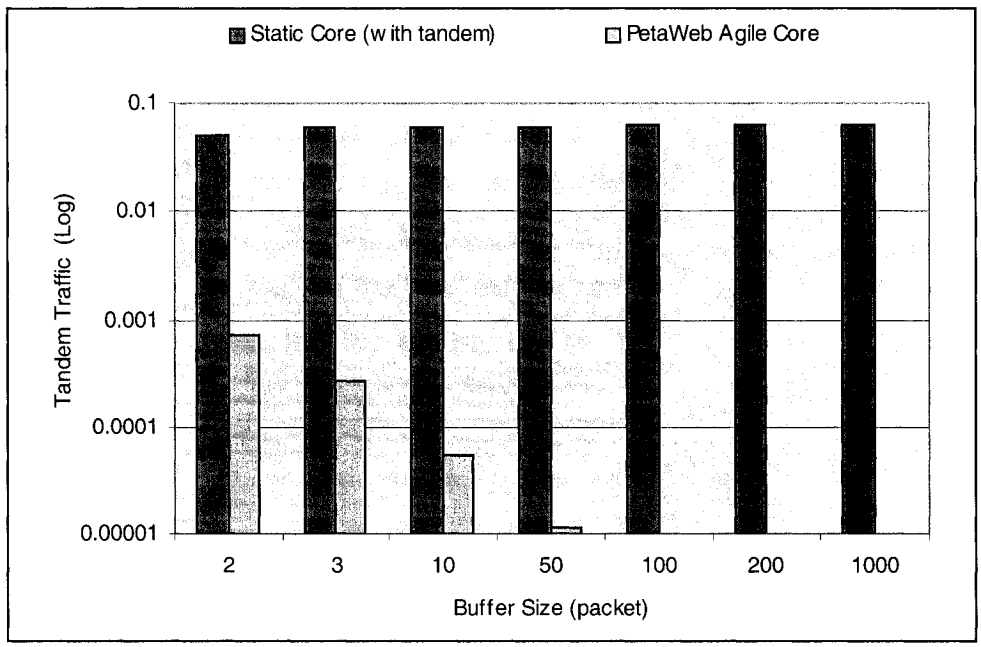


Figure 6.75 Tandem Traffic vs. Buffer Size (for *combined alpha-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.84, reconfiguration period is 250 ms)

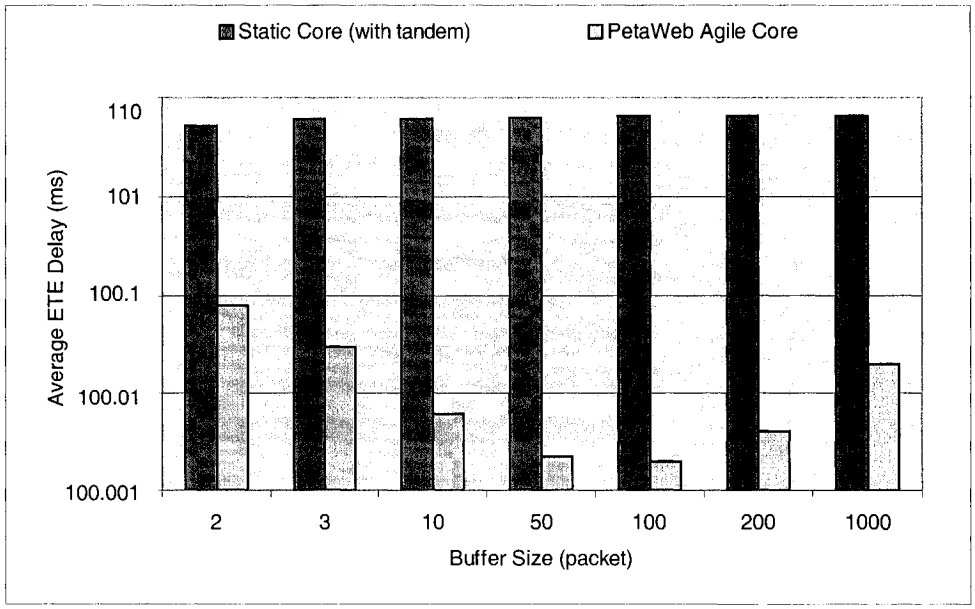


Figure 6.76 Average end-to-end Delay vs. Buffer Size (for *combined alpha-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.84, reconfiguration period is 250 ms)

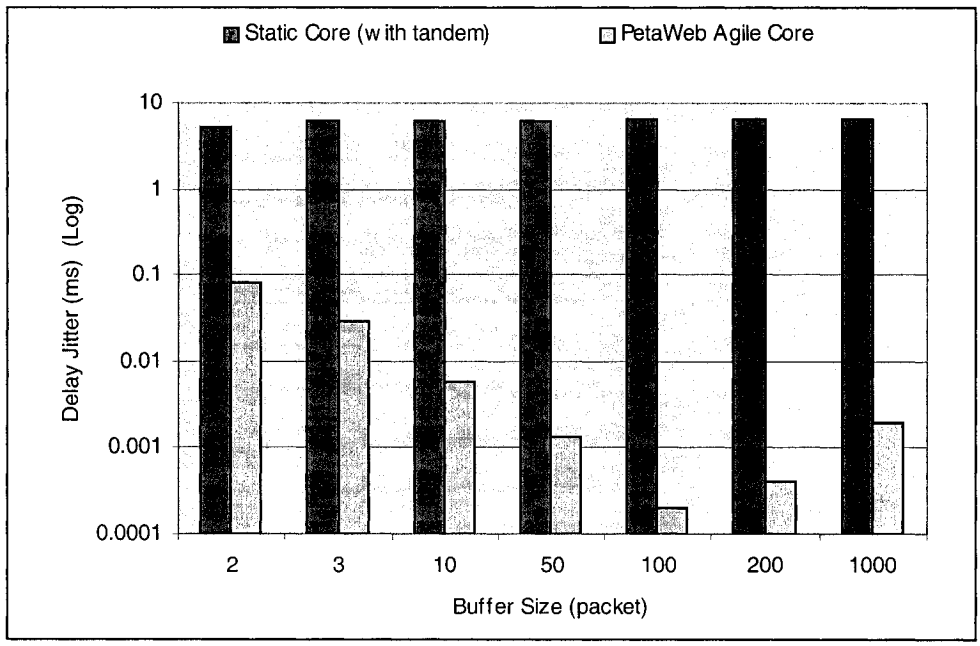


Figure 6.77 Delay Jitter vs. Buffer Size (for *combined alpha-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.84, reconfiguration period is 250 ms)

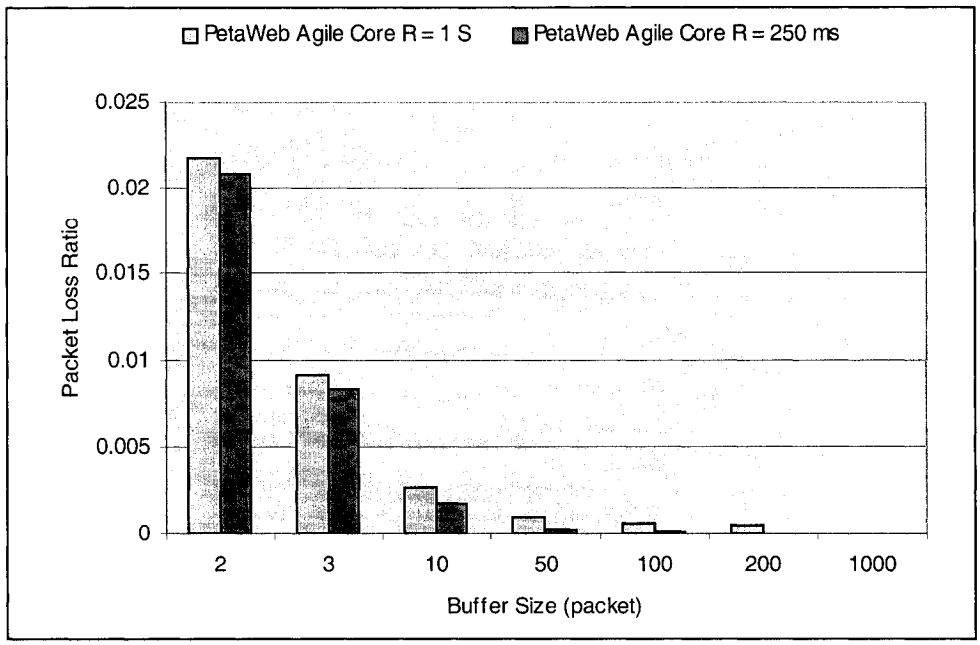


Figure 6.78 Packet Loss Ratio vs. Buffer Size (for *combined alpha-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.84, reconfiguration period is 250 ms, 1 S)

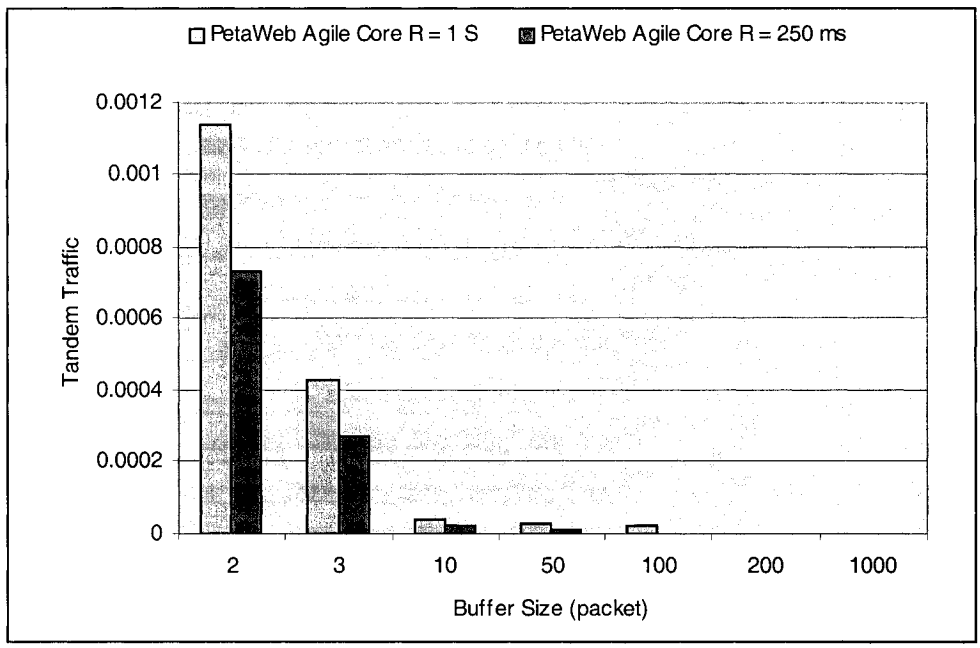


Figure 6.79 Tandem Traffic vs. Buffer Size (for *combined alpha-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.84, reconfiguration period is 250 ms, 1 S)

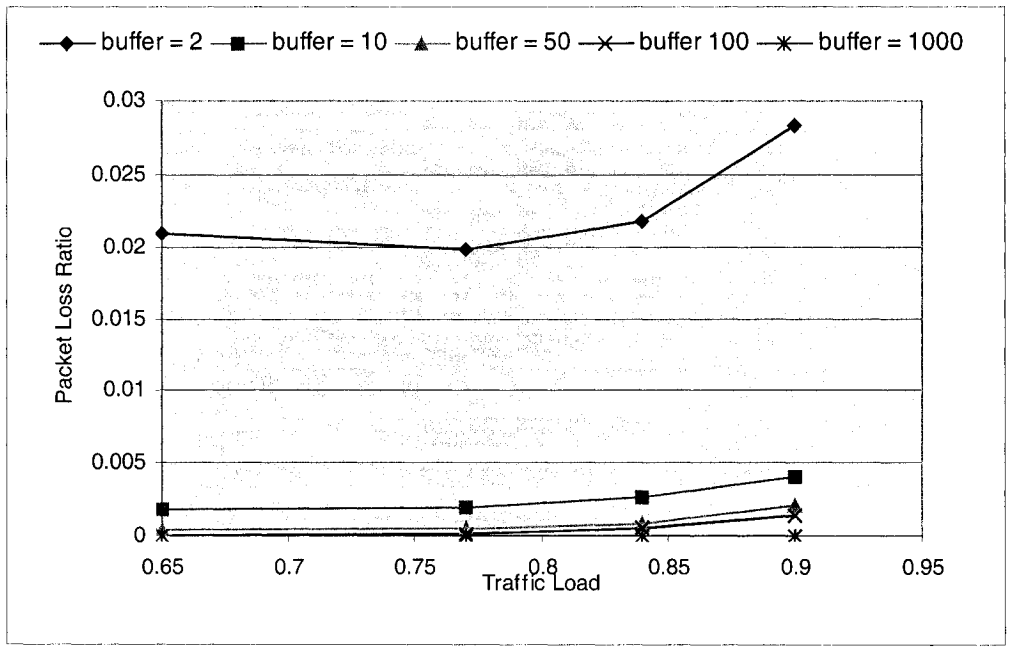


Figure 6.80 Packet Loss Ratio vs. Traffic Load (for *combined alpha-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, reconfiguration period is 1 S, buffer size is 2, 10, 50, 100, and 1000)

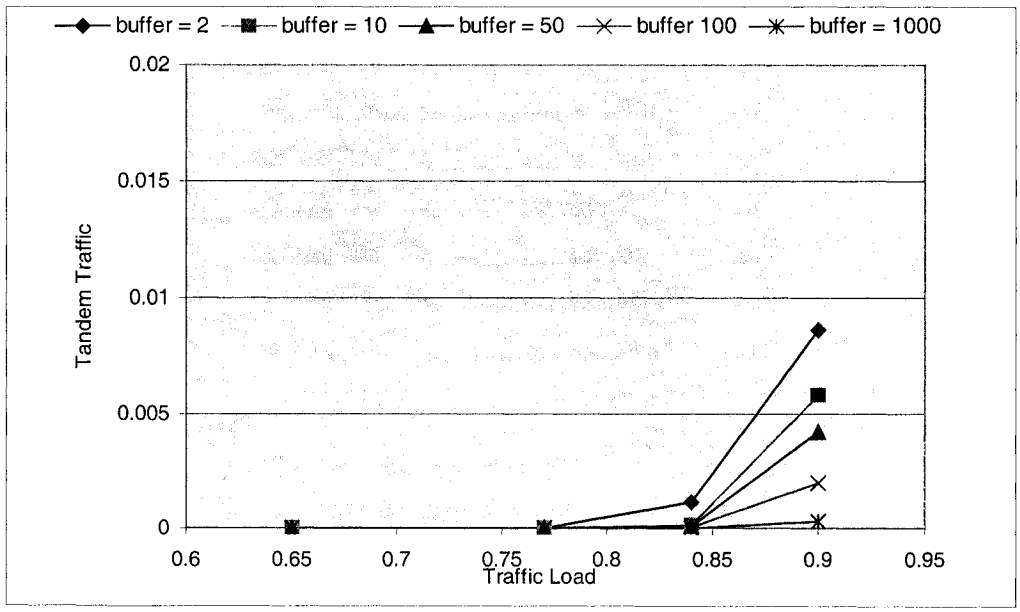


Figure 6.81 Tandem Traffic vs. Traffic Load (for *combined alpha-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, reconfiguration period is 1 S, buffer size is 2, 10, 50, 100, and 1000)

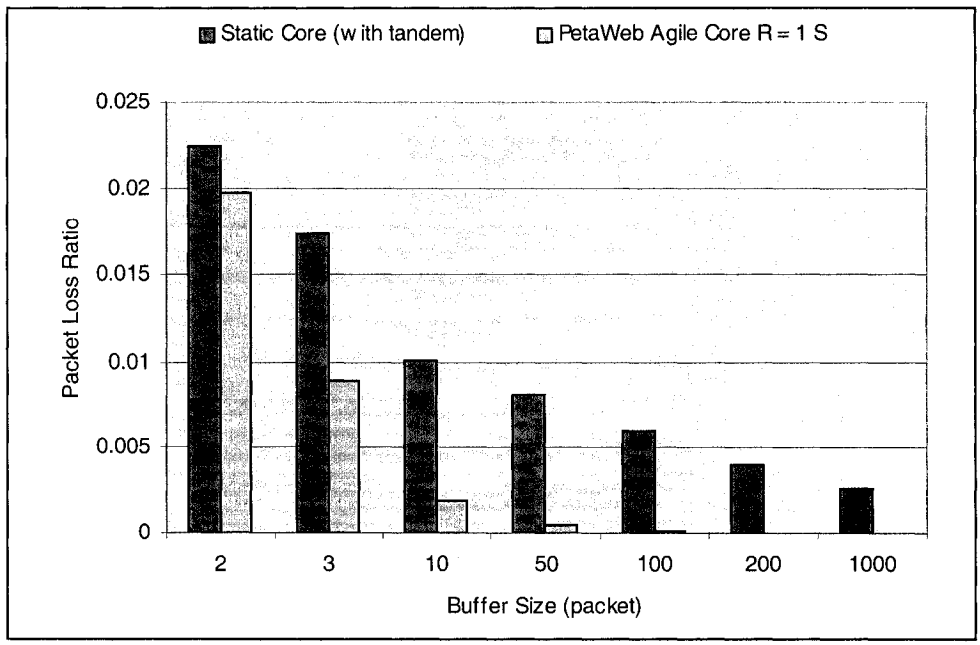


Figure 6.82 Packet Loss Ratio vs. Buffer Size (for *combined alpha-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.77, reconfiguration period is 1 S)

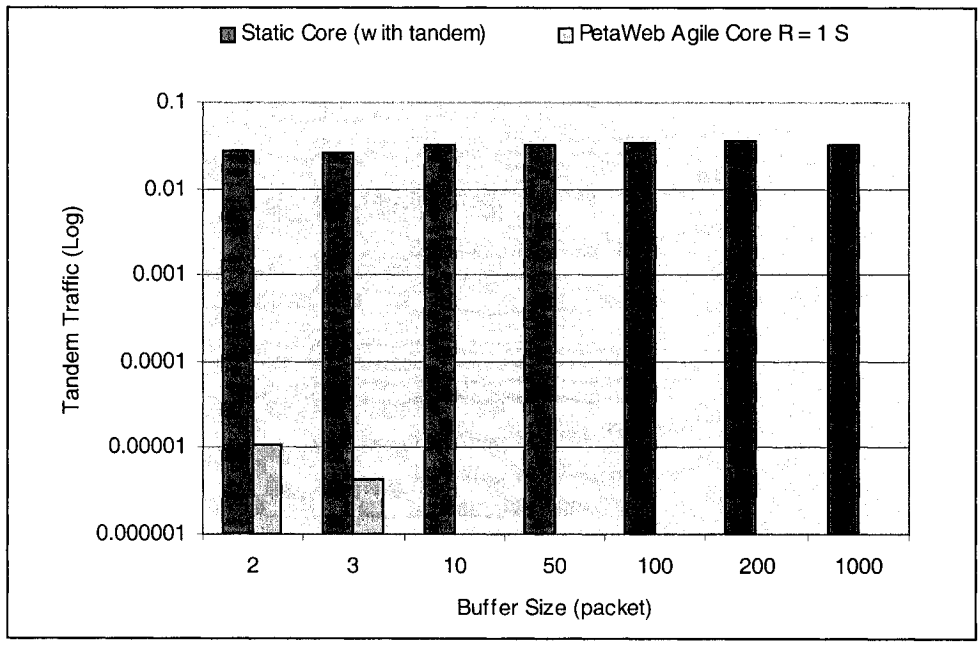


Figure 6.83 Tandem Traffic vs. Buffer Size (for *combined alpha-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.77, reconfiguration period is 1 S)

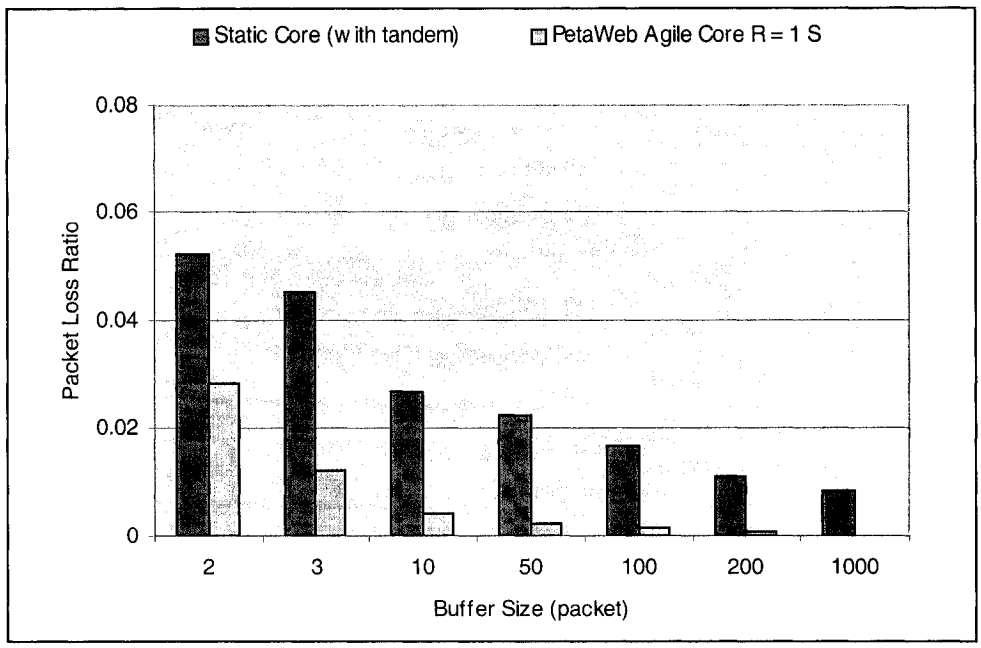


Figure 6.84 Packet Loss Ratio vs. Buffer Size (for *combined alpha-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.9, reconfiguration period is 1 S)

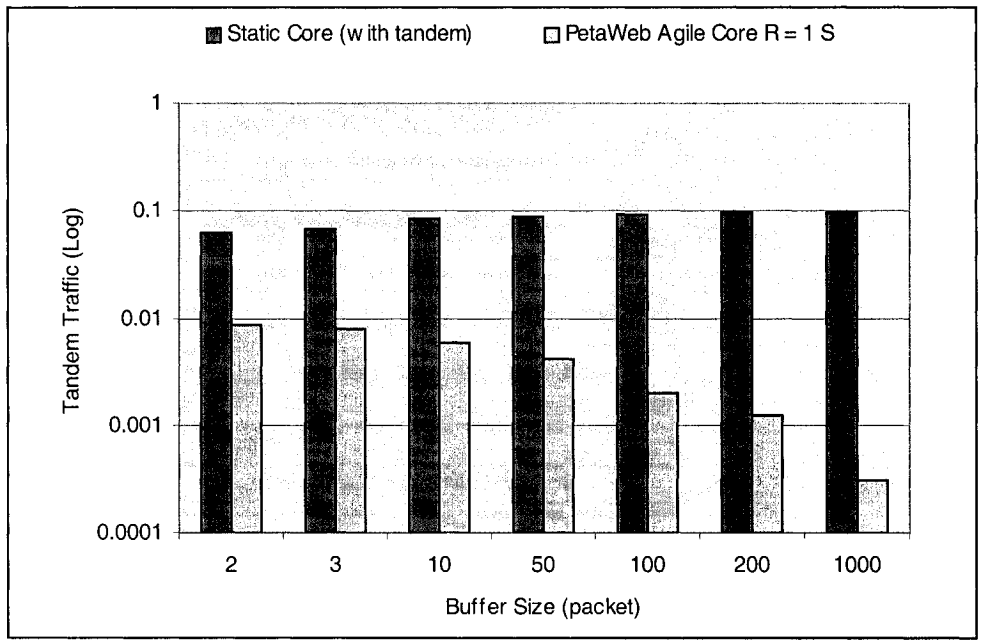


Figure 6.85 Tandem Traffic vs. Buffer Size (for *combined alpha-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, traffic load is 0.9, reconfiguration period is 1 S)

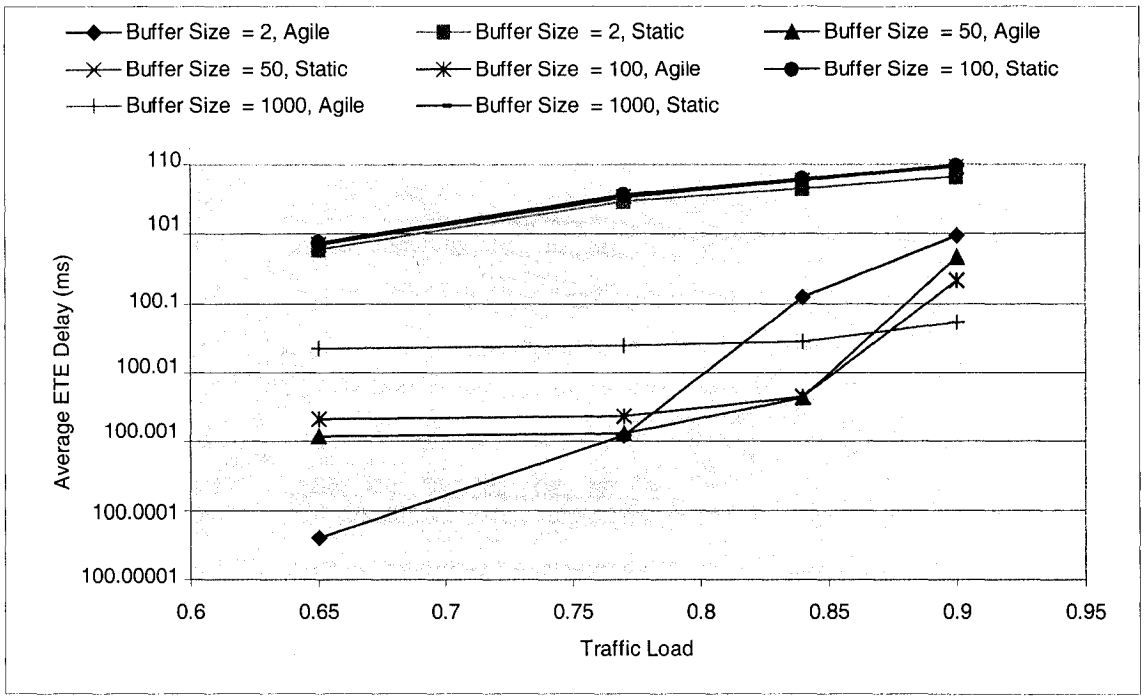


Figure 6.86 Average end-to-end Delay vs. Traffic Load (for *combined alpha-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, reconfiguration period is 1 S, buffer size is 2, 50, 100, 1000)

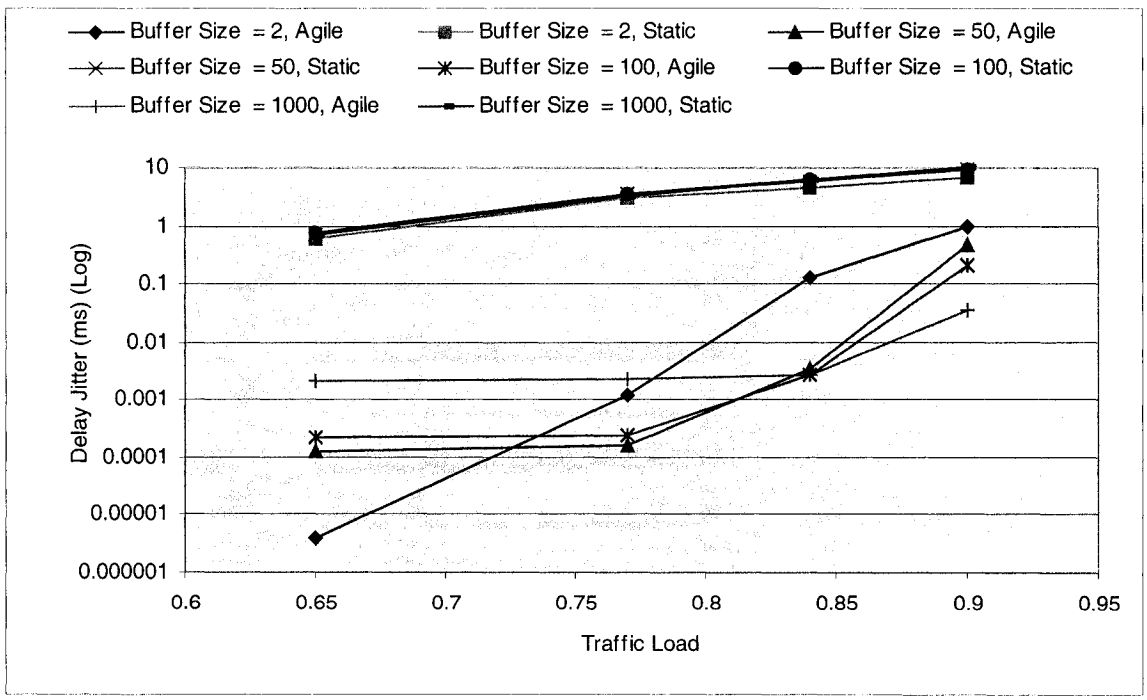


Figure 6.87 Delay Jitter vs. Traffic Load (for *combined alpha-stable traffic*, $Ratio_{\alpha\_stable}$ is 0.2, reconfiguration period is 1 S, buffer size is 2, 50, 100, 1000)

In figure 6.70 to figure 6.79, the effect of the different buffer size on the performance of the PetaWeb network and static core network (the packet loss ratio, tandem traffic, average end-to-end delay, and delay jitter) are shown. The traffic load is 0.84, and the reconfiguration period is 1 second. From the figures, we can see that for the PetaWeb network, when buffer size is equal to 2; the tandem traffic and packet loss ratio is reach to 0.0011, and 0.0217. As we know the tandem traffic will increase traffic average end-to-end delay and delay jitter, and high traffic loss ratio affects TCP application considerable badly, and as far as desktop video conferencing. So we need to achieve lower traffic loss ratio such as 0.01%, and eliminate the tandem traffic. We find that by increasing buffer size the packet loss ratio and tandem traffic are decrease significantly, the average end-to-end delay and delay jitter reduce considerably as well. When the buffer size reaches 200 packets, no tandem traffic exists. When buffer size reaches 1000, no packet is dropped. However, in the static core network, by increasing the buffer size, the packet loss ratio does not decrease too much; on the other hand, the tandem traffic increases a little. Increased buffer size only absorbs the bursts to some extend, but the bursts are large enough to force the static network to its limits. On the contrary, the combination of buffering and resource reallocation allow the agile core network successfully absorb the traffic bursts. This is another advantage that the agility of the PetaWeb brings. From figures 6.72, 6.73, 6.76, 6.77, we can find that compared with the static core network, the average delay and delay jitter of the PetaWeb network are considerably lower, due to the less the tandem traffic in the PetaWeb network. Figure 6.78 and 6.79 shows the effect of different reconfiguration period on the tandem traffic and packet loss ratio for different buffer sizes. The simulation results lead to similar conclusions. The shorter the reconfiguration period is, the lower the packet loss ratio and tandem traffic becomes. We also investigate the behavior of the network and the effect of different buffer size on the network performance under different traffic loads. The simulation results are shown in figure 6.80 to figure 6.87. From these, we find that under any traffic load, the effect of the buffer size to the network performance is similar. The higher traffic load, the more tandem traffic and higher packet loss ratio is; as a result, the higher delay and delay jitter.

The end-to-end packet delay in the PetaWeb is dominated by the propagation delay experienced by packets as they travel through the optical links. The buffer at the edge nodes reduces the tandem traffic and increases the effective network capacity. However, at the same time, it adds queuing delays and delay jitters. This is demonstrated by the results shown in figures 6.73, 6.76, and 6.77. This becomes a problem to real time services, and could also generate degradations and reduced throughput to TCP based applications. Thus, it is important that the effect of buffering on the queuing delay and delay jitter is fully understood. Figures 6.72, 6.73, 6.76, and 6.77 show how the increase of buffer size affects the average packet delay and the delay jitter. As expected, for the static core network, the average end-to-end delay and delay jitter will increase with the buffer size, due to the tandem traffic increases. On the contrary, for the PetaWeb (agile core) network, increasing buffer size, the average end-to-end delay and delay jitter will decrease until the buffer size reaches 200 packets (see figures 6.73, 6.76, and 6.77, traffic load is 0.84 and reconfiguration period is 1 second); after that, the average end-to-end delay and delay jitter will increase with the buffer size. This is because with the buffer size increasing, the tandem traffic and blocked traffic decrease at the beginning, reducing tandem traffic can make up the queuing delay resulted by larger buffer size. As we know, the tandem traffic has dominant impact on the traffic delay jitter, because the packets need to be transmitted through intermediate node, thus encountering extremely long propagation delay (in the experiments, the propagation delay from edge node to edge node is set to 100 milliseconds). If there is no tandem traffic in the network, increasing buffer size will result in larger average end-to-end delay and delay jitter due to larger queuing delay. In figure 6.86, and 6.87, the effect of different buffer size on the average end-to-end delay and delay jitter under different traffic load is presented. As expected, the packet delay and delay jitter in the static core network are extremely larger than those in the PetaWeb network. Also according to the simulation results, for the traffic load less than 0.84, the suitable buffer size for the network is 100 packets. For the traffic load beyond 0.84, larger buffer size (such as 1000 packets) is desirable. We also investigate the role of the buffer size to the different traffic characteristics. These results are presented in figures 6.88 to 6.91. From the figures we can see that the larger buffer size is, the lower the packet loss ratio and tandem traffic becomes regardless of $Ratio_{\alpha\_stable}$ (0.2, 0.4, and 0.6) and the reconfiguration period

(250 ms and 1 second) is. By increasing buffer size (thus increasing the ability to absorb bursts), the effect of the traffic ratio on the network performance (packet loss ratio and tandem traffic) is reduced considerably. In the experiments, when buffer size reaches 1000 packets, the tandem traffic and packet loss ratio are eliminated completely for any ratio.

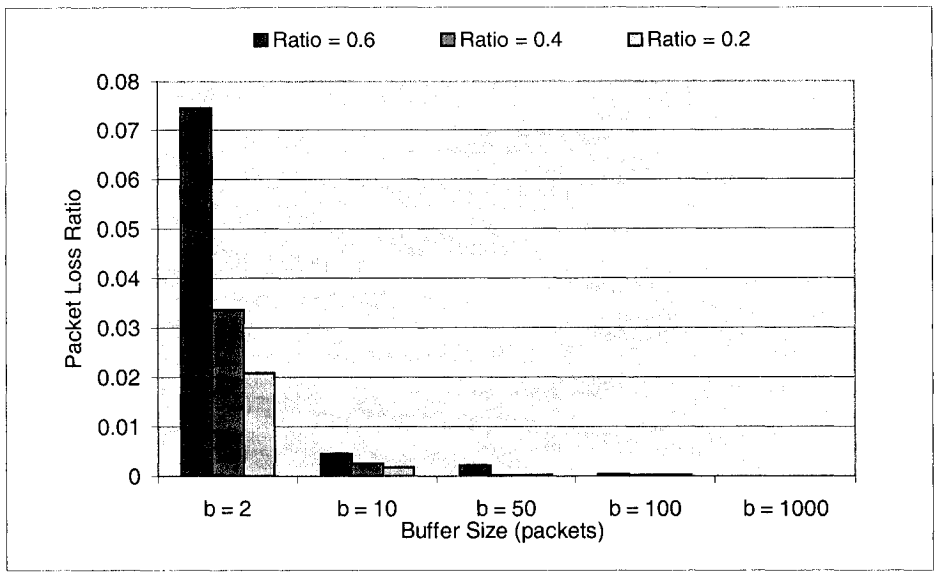


Figure 6.88 Packet Loss Ratio vs. Buffer Size (for *combined alpha-stable traffic*, traffic load is 0.84, reconfiguration period is 250 ms, and $Ratio_{\alpha\_stable}$ is 0.2, 0.4, 0.6)

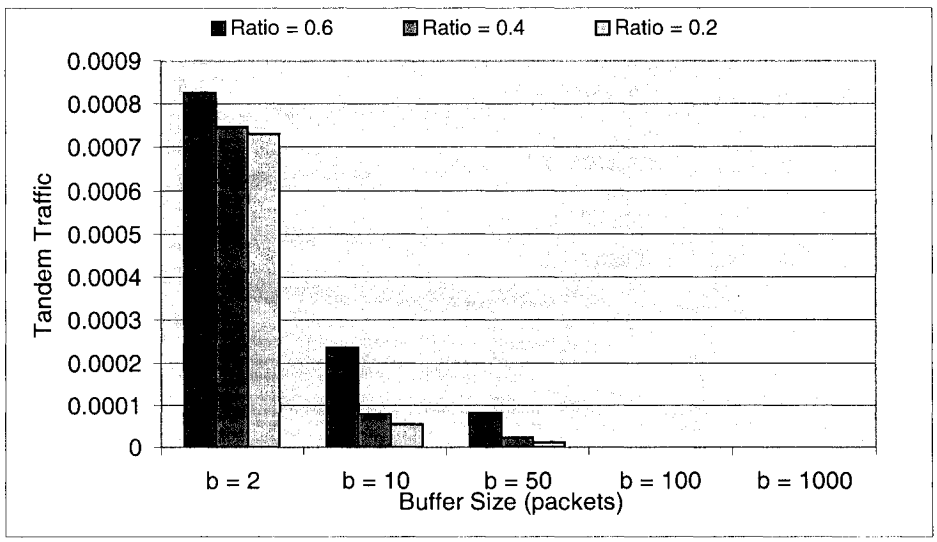


Figure 6.89 Tandem Traffic vs. Buffer Size (for *combined alpha-stable traffic*, traffic load is 0.84, reconfiguration period is 250 ms, and $Ratio_{\alpha\_stable}$ is 0.2, 0.4, 0.6)

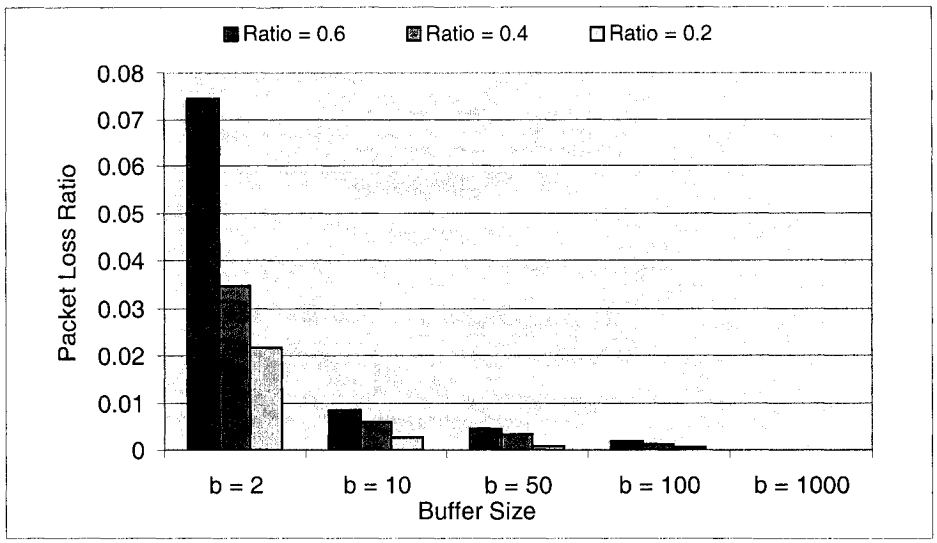


Figure 6.90 Packet Loss Ratio vs. Buffer Size (for *combined alpha-stable traffic*, traffic load is 0.84, reconfiguration period is 1 S, and $Ratio_{\alpha\_stable}$ is 0.2, 0.4, 0.6)

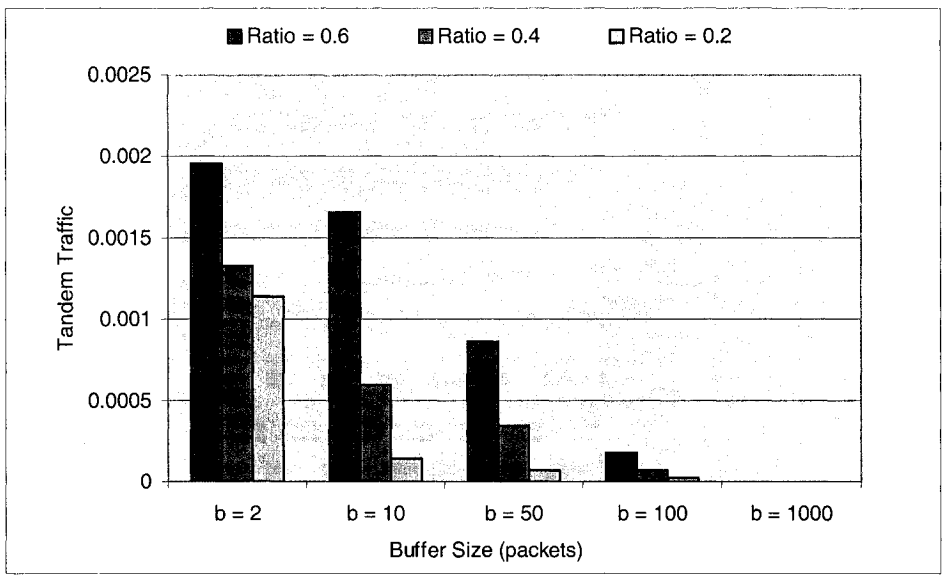


Figure 6.91 Tandem Traffic vs. Buffer Size (for *combined alpha-stable traffic*, traffic load is 0.84, reconfiguration period is 250 ms, and $Ratio_{\alpha\_stable}$ is 0.2, 0.4, 0.6)

6.7 Effect of the Number and the Granularity of Optical Channels

The number and the granularity of optical channels between edge nodes and core nodes are one of the factors, which can affect the degree of PetaWeb's agility and network performance. It determines the granularity and the effectiveness of PetaWeb reaction to the

traffic fluctuation. In the experiments, the self-similar traffic generated by the alpha-stable traffic model combined with the sinusoidal traffic source is used. The traffic load is 0.65, and 0.84, and the reconfiguration period is set as 1 s, and 10 s. The PetaWeb and static core network (with tandem) cases are examined. The performance of the PetaWeb network with 16 optical channels (10Gbps transmission speed for each channel) per link, 4 optical channels (40Gbps transmission speed for each channel) per link, and 64 optical channels (2.5Gbps transmission speed for each channel) are compared. Figures 6.92 and 6.99 show the tandem traffic and packet loss ratio achieved by the PetaWeb and the static core network under different reconfiguration periods and traffic loads. In all cases, the total link capacity remains the same. According to the results and expectations, under any traffic load and reconfiguration period, the finer the granularity is, the lower the tandem traffic and packet loss ratio becomes; also compared with static core network, the tandem traffic and packet loss ratio in the PetaWeb network under any channel granularities are considerable lower. For the PetaWeb, the capacity of direct connectivity paths can change dynamically according to the traffic demand at each edge node, the lower channel capacity, and the more accuracy for the link capacity in reaction to the traffic fluctuation. As a result, the finer the granularity, the more agile the PetaWeb becomes.

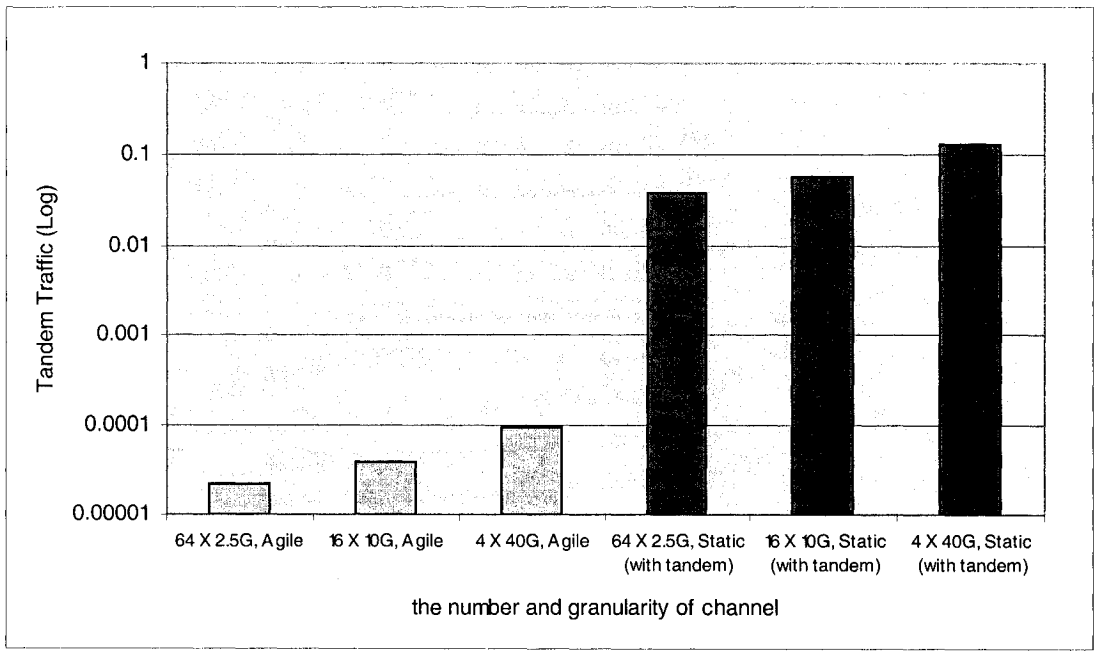


Figure 6.92 Tandem Traffic vs. the number and granularity of optical channel (traffic load is 0.84, reconfiguration period is 1 S)

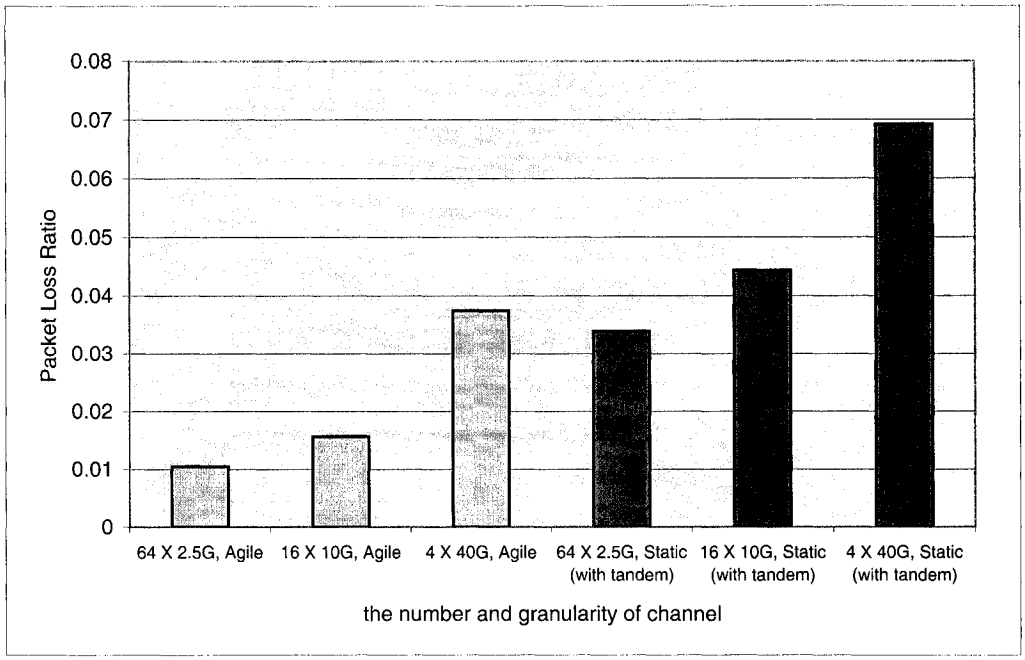


Figure 6.93 Packet Loss Ratio vs. the number and granularity of optical channel (traffic load is 0.84, reconfiguration period is 1 S)

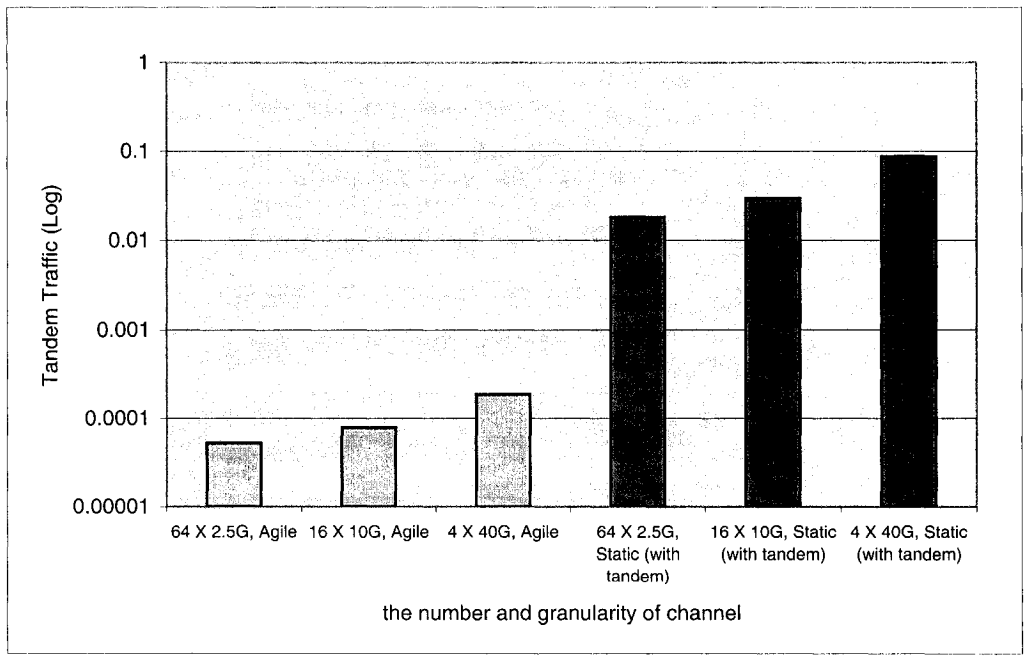


Figure 6.94 Tandem Traffic vs. the number and granularity of optical channel (traffic load is 0.84, reconfiguration period is 10 S)

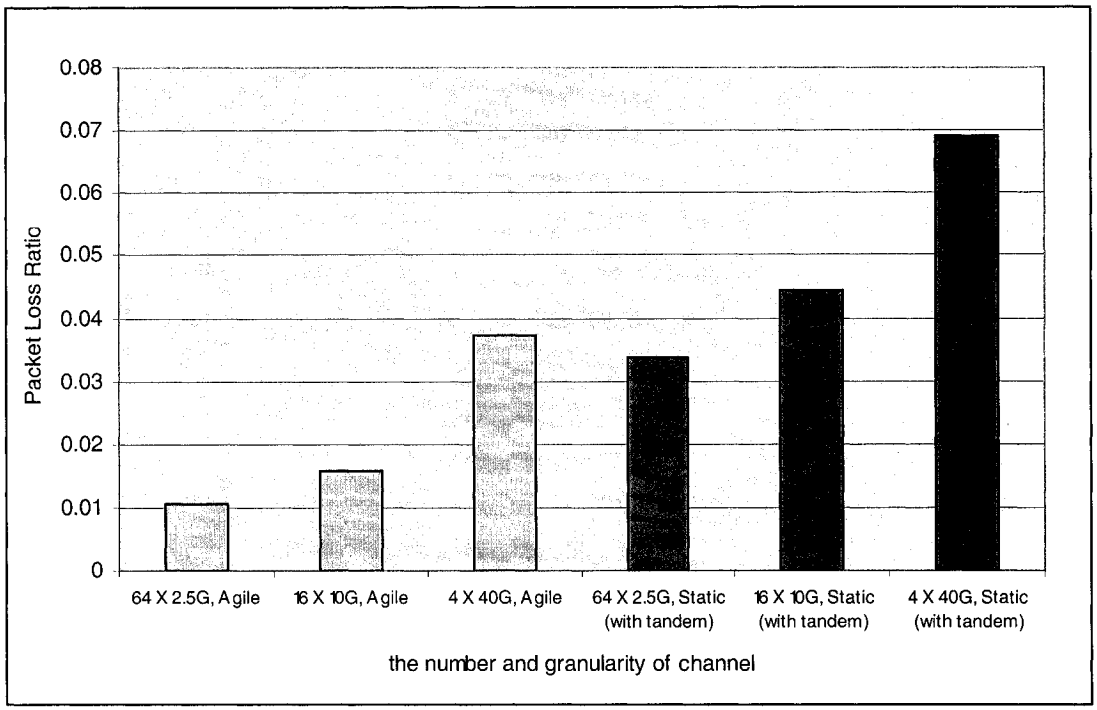


Figure 6.95 Packet Loss Ratio vs. vs. the number and granularity of optical channel (traffic load is 0.84, reconfiguration period is 10 S)

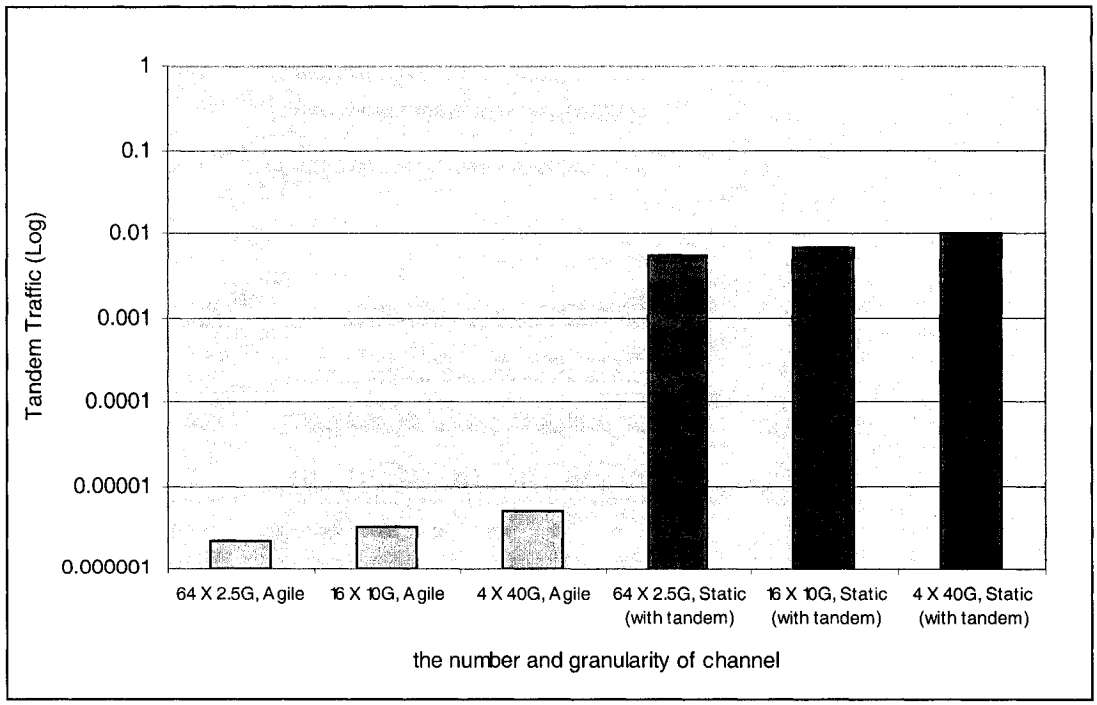


Figure 6.96 Tandem Traffic vs. vs. the number and granularity of optical channel (traffic load is 0.65, reconfiguration period is 1 S)

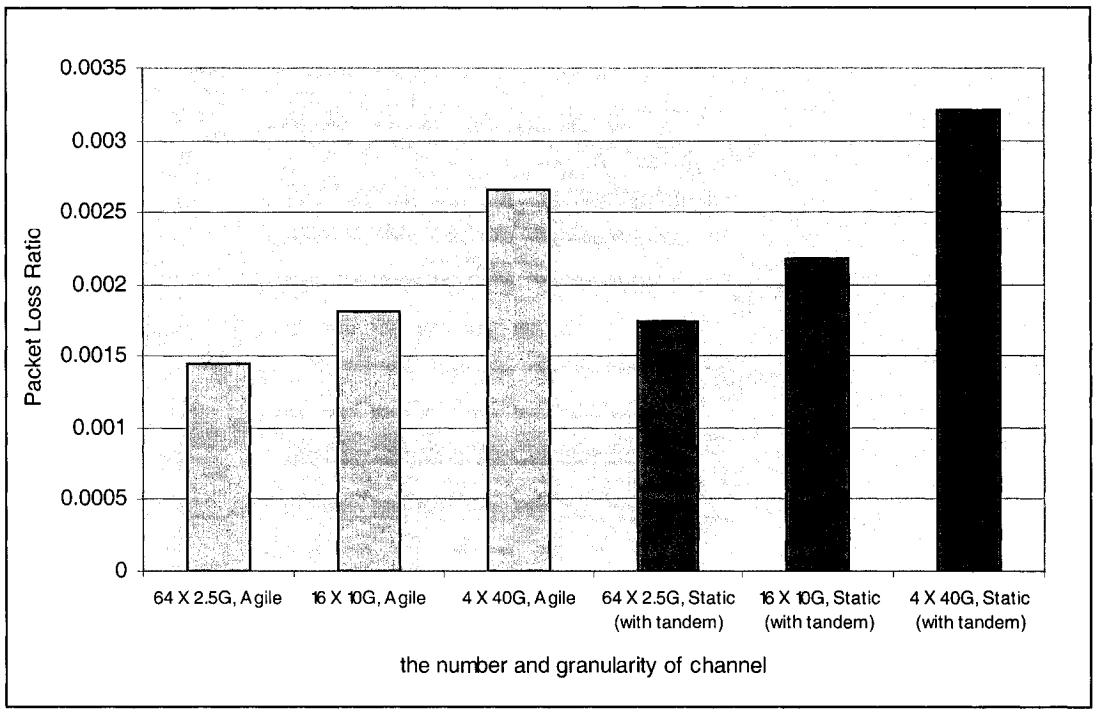


Figure 6.97 Packet Loss Ratio vs. vs. the number and granularity of optical channel (traffic load is 0.65, reconfiguration period is 1 S)

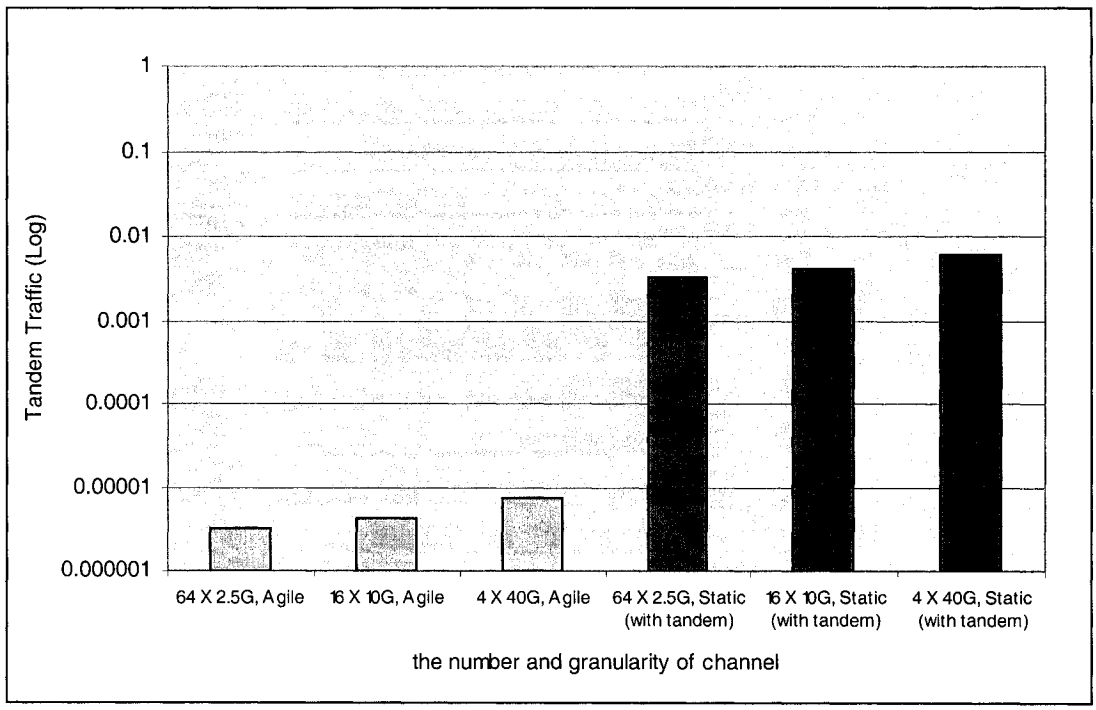


Figure 6.98 Tandem Traffic vs. vs. the number and granularity of optical channel (traffic load is 0.65, reconfiguration period is 10 S)

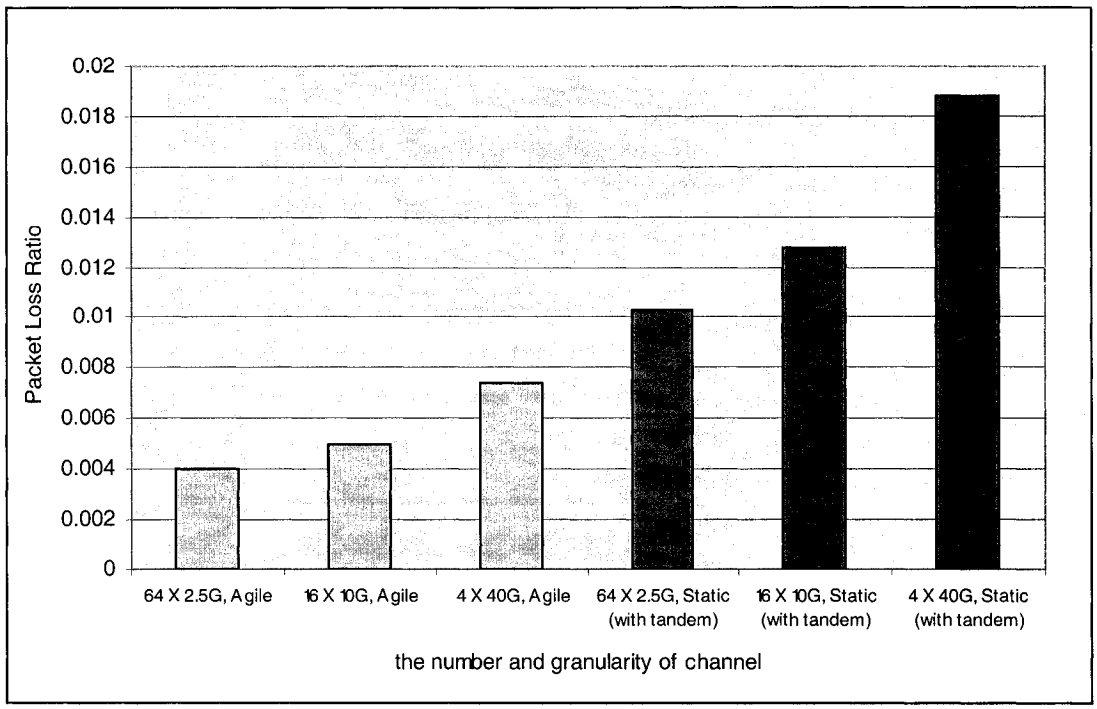


Figure 6.99 Packet Loss Ratio vs. vs. the number and granularity of optical channel (traffic load is 0.65, reconfiguration period is 10 S)

6.8 The Performance of QoS-Aware PetaWeb Network

As the complexity, sophistication and QoS requirements of Internet-serviced applications become more and more diverse, there is a need to deploy and support QoS over Internet. While optical IP networks becoming the backbone of the future Internet, there is need for QoS support. The Internet Engineering Task Force (IETF) has proposed two QoS enabled architectures for Internet, Integrated Services (IntServ) [3] and Differentiated Services (DiffServ) [4] [5]. The IntServ architecture does not scale well to large numbers of individual users. In our work, we decided to follow the DiffServ approach, which is highly scalable. The architecture is modeled in OPNET and is analyzed through simulations. As we mentioned earlier, the tandem traffic and packet loss ratio affect the network performance, especially for real time multimedia applications and TCP applications. In order to let the PetaWeb network support the quality of service, the QoS-Aware PetaWeb network is proposed. The simulation model is implemented. In the experiments, the traffic sources used are the self-similar traffic generated by the alpha-stable traffic model combined with the sinusoidal traffic; the traffic volatility is set as 0.6; $Ratio_{\alpha\_stable}$ is set as 0.2. We compare the network performance of a generic PetaWeb network with the QoS-Aware PetaWeb network and investigate the impact of different scheduling disciplines on various traffic metrics, such as packet loss, end-to-end delay, delay jitter etc. We deploy Priority Queuing (PQ) and Weighted Round Robin (WRR) as scheduling disciplines for the DiffServ capable PetaWeb network and FIFO queuing discipline for the Non-DiffServ capable PetaWeb network. In all cases (PQ, WRR), the EF traffic is buffered in the highest priority queue, AF is buffered in the middle priority queue, and the BE traffic is buffered in the lowest priority queue; each queue size is set at 10 packets. When implementing PQ, the packets in lower priority queue can be transmitted after the higher priority queue becomes empty, while when deploying WRR, the packets in different priority queues are serviced according to their specific weights respectively. However, for FIFO queue, the buffer size is set as 30 packets, the same size as the total of buffer sizes for PQ and WRR. In the WRR scenario, the EF traffic weight is set as 5/9, the AF traffic weight is set as 3/9, and BE traffic weight is set as 1/9. The relative computing simplicity of both queue-scheduling algorithms allows them to be easily implemented in

hardware, an important feature considering the difficulty of developing electronics to run complex algorithms at the speeds, PetaWeb networks will be operating.

In the experiment, we also investigate the effect of the different proportions of EF, AF, and BE traffic volumes in the offered traffic load on the traffic metrics. A set of scenarios are studied, the proportions of traffics are as EF: AF: BE = 1/3: 1/3: 1/3; EF: AF: BE = 1/2: 1/4: 1/4; EF: AF: BE = 1/4: 1/2: 1/4; EF: AF: BE = 1/4: 1/4: 1/2; and EF: AF: BE = 5/8: 2/8: 1/8 respectively. The total traffic load is fixed for each scenario.

6.8.1 the effect on the packet lose ratio

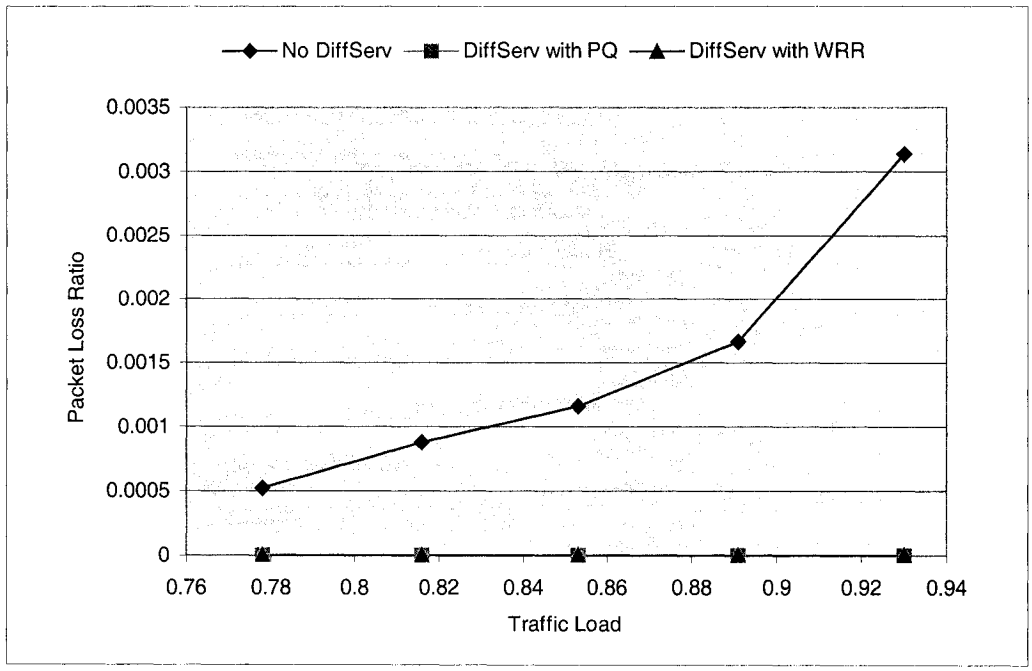
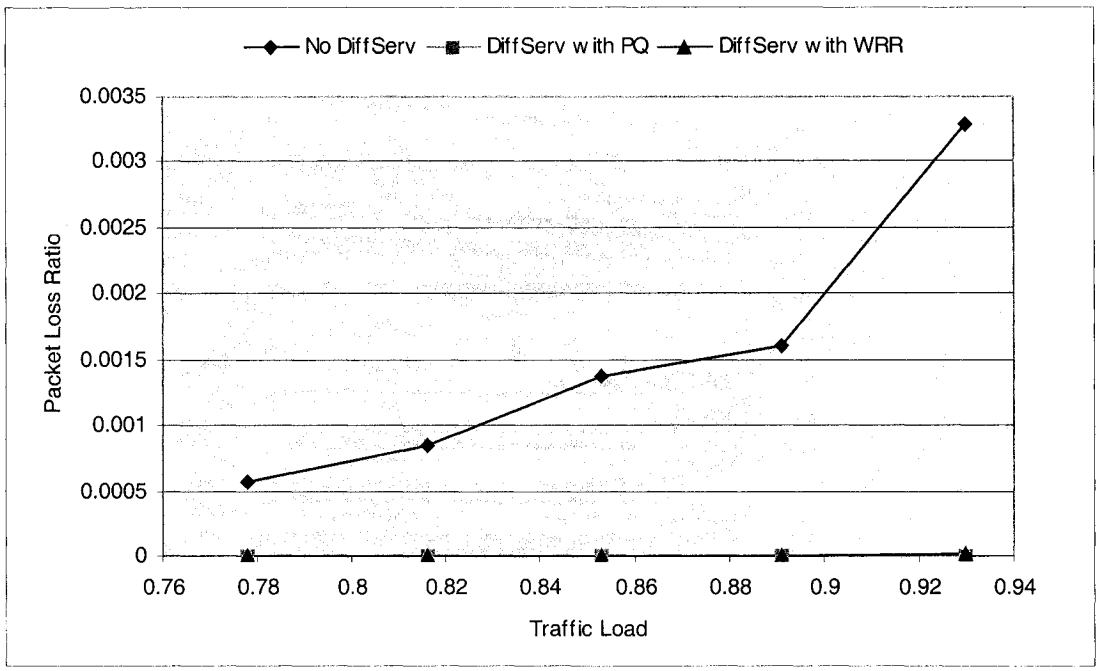
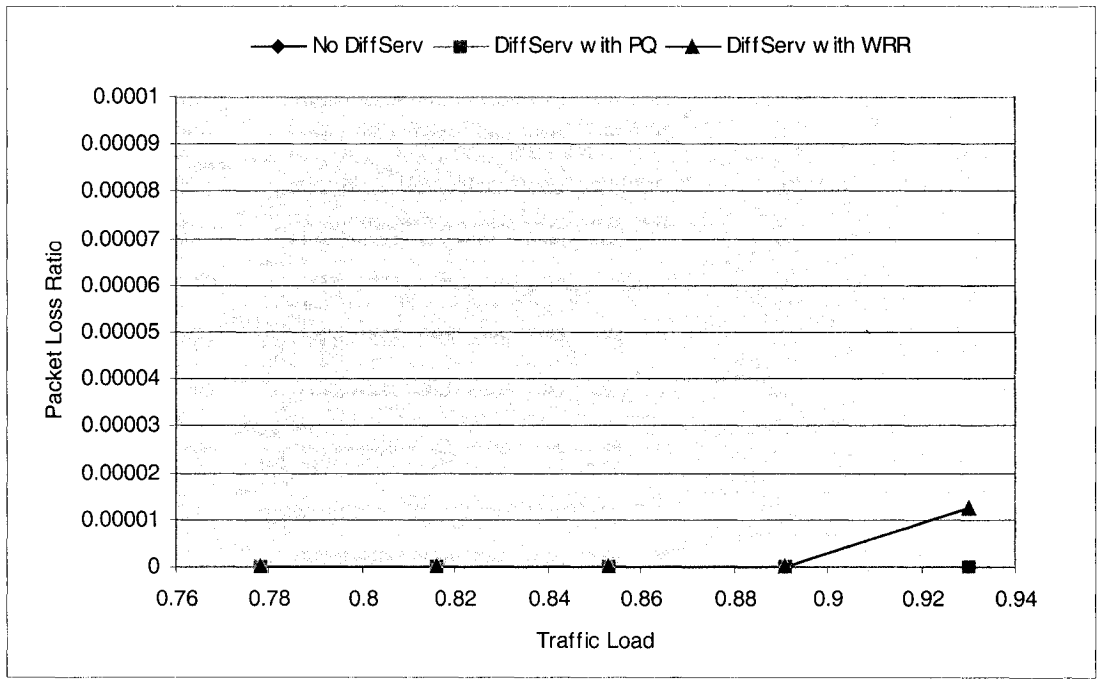


Figure 6.100 EF Packet Loss Ratio vs. Traffic Load (the proportion of traffic volume is EF: AF: BE = 1/3 : 1/3 : 1/3)



(a)



(b)

Figure 6.101 AF Packet Loss Ratio vs. Traffic Load (the proportion of traffic volume is EF: AF: BE = 1/3 : 1/3 : 1/3) (Note: figure (b) is the high resolution of figure (a))

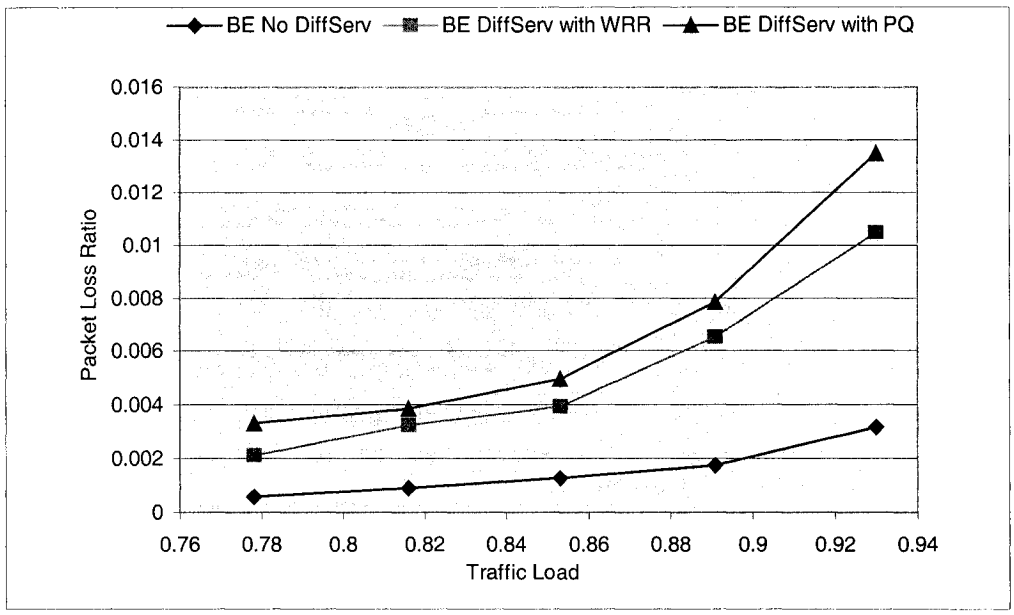


Figure 6.102 BE Packet Loss Ratio vs. Traffic Load (the proportion of traffic volume is EF: AF: BE = 1/3 : 1/3 : 1/3)

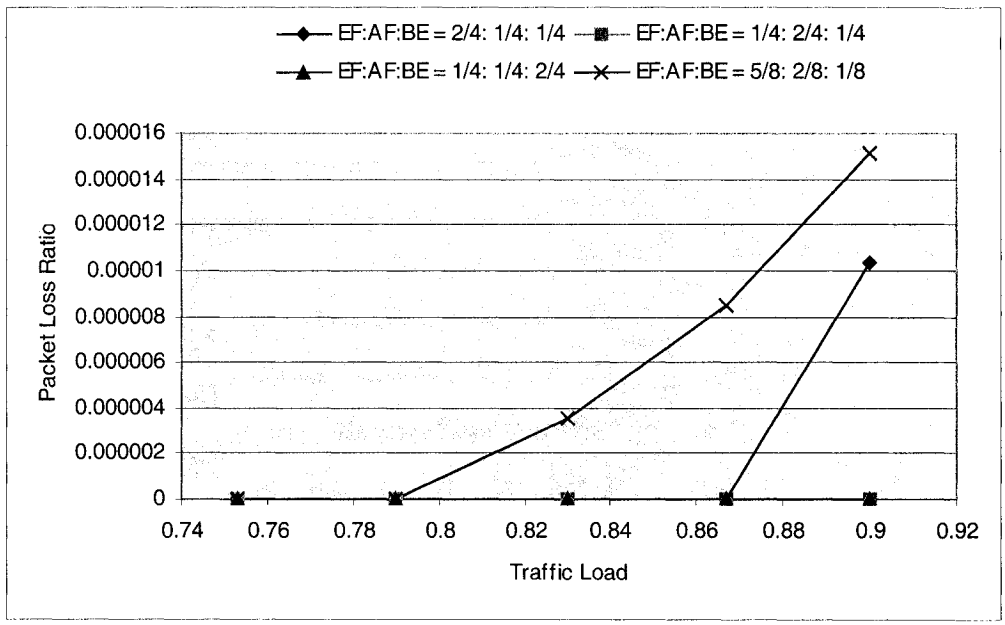


Figure 6.103 AF Packet Loss Ratio vs. Traffic Load (with PQ under the different proportions of traffic volume, EF: AF: BE = 2/4 : 1/4 : 1/4, EF: AF: BE = 1/4 : 2/4 : 1/4, EF: AF: BE = 1/4 : 1/4 : 2/4, and EF: AF: BE = 5/8 : 2/8 : 1/8)

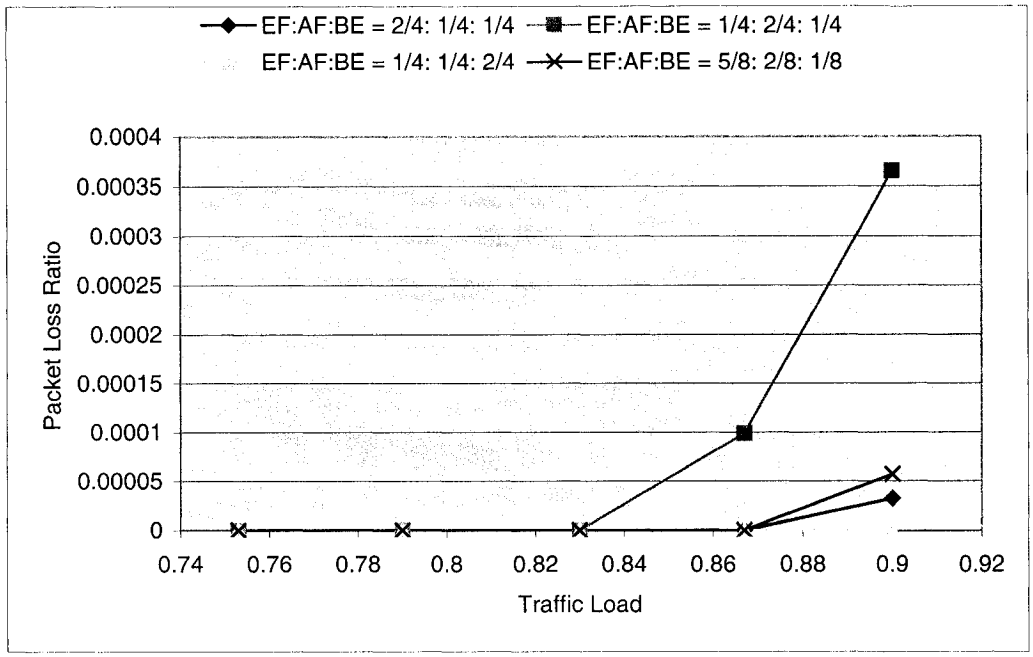


Figure 6.104 AF Packet Loss Ratio vs. Traffic Load (with WRR under the different proportions of traffic volume, EF: AF: BE = 2/4 : 1/4 : 1/4, EF: AF: BE = 1/4 : 2/4 : 1/4, EF: AF: BE = 1/4 : 1/4 : 2/4, and EF: AF: BE = 5/8 : 2/8 : 1/8)

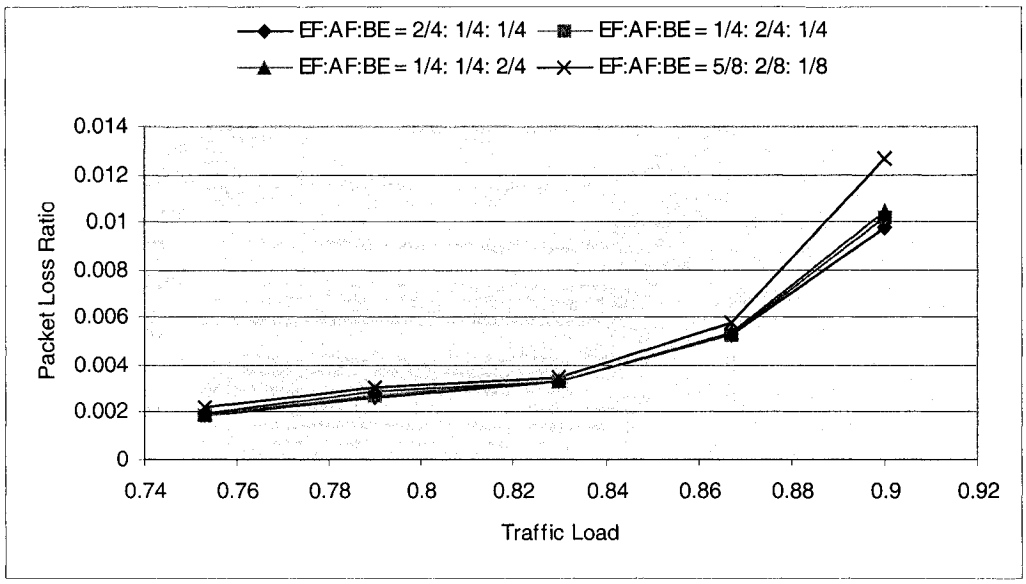


Figure 6.105 BE Packet Loss Ratio vs. Traffic Load (with PQ under the different proportions of traffic volume, EF: AF: BE = 2/4 : 1/4 : 1/4, EF: AF: BE = 1/4 : 2/4 : 1/4, EF: AF: BE = 1/4 : 1/4 : 2/4, and EF: AF: BE = 5/8 : 2/8 : 1/8)

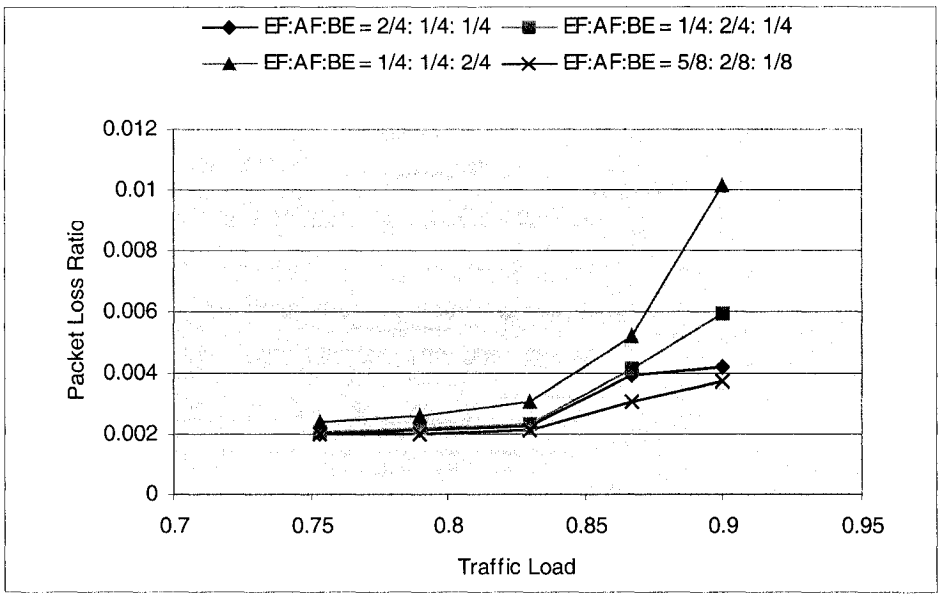


Figure 6.106 BE Packet Loss Ratio vs. Traffic Load (with WRR under the different proportions of traffic volume, EF: AF: BE = 2/4 : 1/4 : 1/4, EF: AF: BE = 1/4 : 2/4 : 1/4, EF: AF: BE = 1/4 : 1/4 : 2/4, and EF: AF: BE = 5/8 : 2/8 : 1/8)

Figure 6.100 and figure 6.101 illustrate high packet loss ratio for EF and AF traffic in non QoS-Aware network, while the EF packet loss ratio in the QoS-Aware PetaWeb network can be kept very low under both queuing schemes (WRR and PQ). The simulation did not count any loss under both schemes for premium service in QoS-Aware networks. In figure 6.101, AF traffic encounters some packet loss under WRR queuing scheme when the traffic load is higher than 0.93; however, under PQ queuing, there is no packet loss. When we increase the buffer size from 10 packets to 20 packets, the AF packet loss is eliminated. As expected from figure 6.102, we see that the best effort packet loss ratio in the PQ and WRR scenarios are higher than in Non-DiffServ PetaWeb network. The best effort packet loss with Weighted Round Robin (WRR) is less than that with Priority Queuing (PQ). This is because in PQ, the packets in low priority queue are transmitted only after the high priority queue becomes empty, while with WRR, the packets in both high and low queues are serviced according to their specific weights. As a result, under PQ scheduling, buffer overflow at lower priority queues could occur more often as compared to WRR, therefore the packet loss is higher. Figures 6.103 to figure 6.106 show the AF and Best Effort traffic packet loss ratio with different proportions of EF, AF and BE traffic volumes. From these

figures we can get that the network with WRR scheduling encountered significant impact on the BE traffic when changing the EF, AF, and BE traffic proportion. The reason is in WRR, the EF, AF, and BE traffic is serviced according to their weights, on the contrary, in PQ, the BE traffic is serviced only when the EF queue and AF queue is empty, so changing traffic proportion in WRR, the effect on the BE traffic is higher than in PQ; In WRR, increasing the BE traffic, but the weight is fixed, so more BE packets will be dropped. When increasing AF traffic at the heavy load, the AF traffic loss occurs, through experiment, when increasing buffer size from 10 packets to 20 packets, the AF packet loss can be eliminated completely. This is because in WRR, the AF packets are inserted into AF queue, when the EF and BE packets are being serviced, and each round the traffic in different queue will be serviced according to their predetermined weights, so the more AF traffic, the more probability the AF queue becoming full, the more AF packet will be dropped. If increasing the EF traffic load over reserved bandwidth, as expected, the AF and BE packet loss ratio will increase in PQ; however, in WRR, the effect on the AF and BE traffic is relatively small, due to the restriction of weights.

All of these results demonstrate that in the QoS-Aware PetaWeb network, the differentiated service can be provided, the Premium service and Assured service has better performance than best effort service. It also demonstrates the functionalities of our QoS-Aware PetaWeb network is correct, the QoS-Aware channel allocation algorithm works properly as our expectation.

6.8.2 the effect on the end-to-end delay and delay jitter

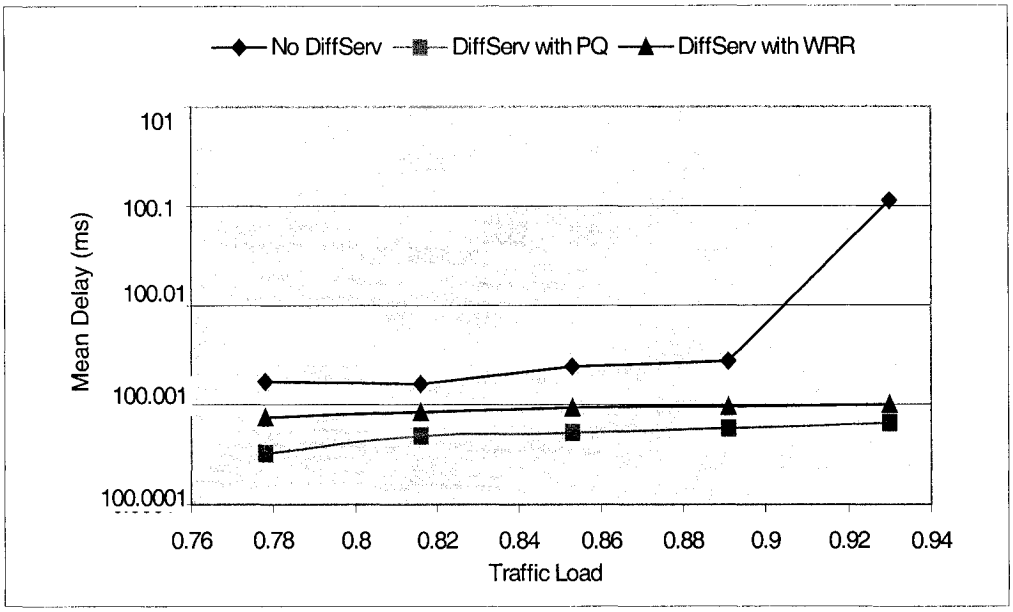


Figure 6.107 EF Traffic end-to-end Mean Delay vs. Traffic Load (the proportion of traffic volume is EF: AF: BE = 1/3 : 1/3 : 1/3)

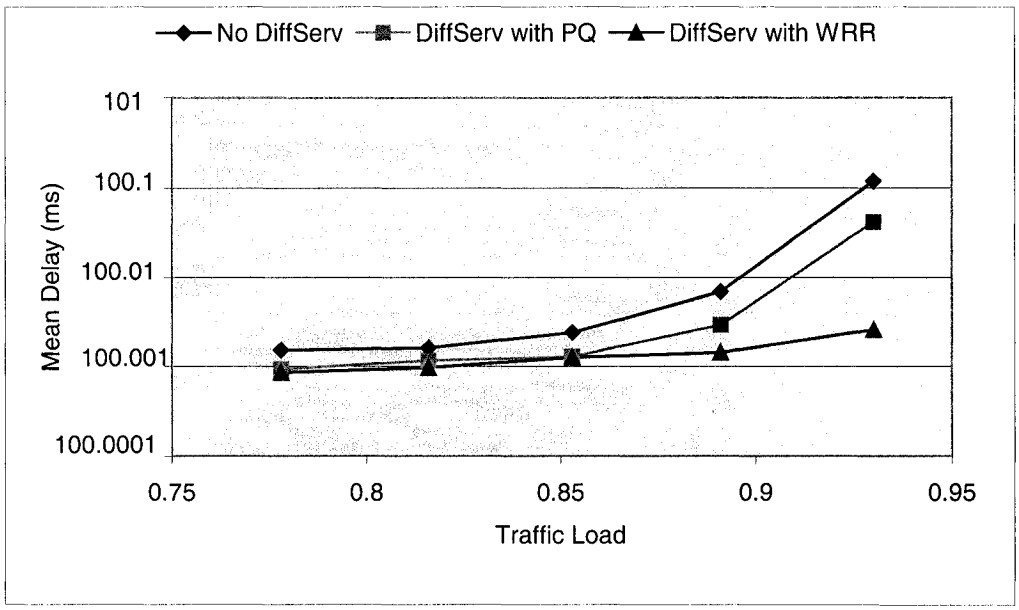


Figure 6.108 AF Traffic end-to-end Mean Delay vs. Traffic Load (the proportion of traffic volume is EF: AF: BE = 1/3 : 1/3 : 1/3)

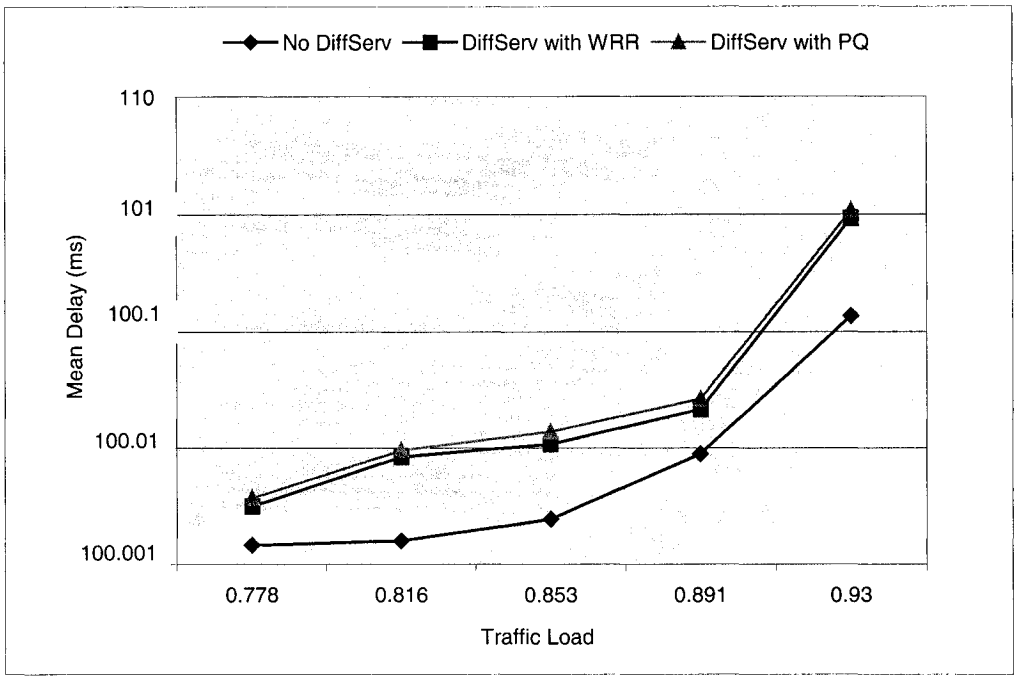


Figure 6.109 BE Traffic end-to-end Mean Delay vs. Traffic Load (the proportion of traffic volume is EF: AF: BE = 1/3 : 1/3 : 1/3)

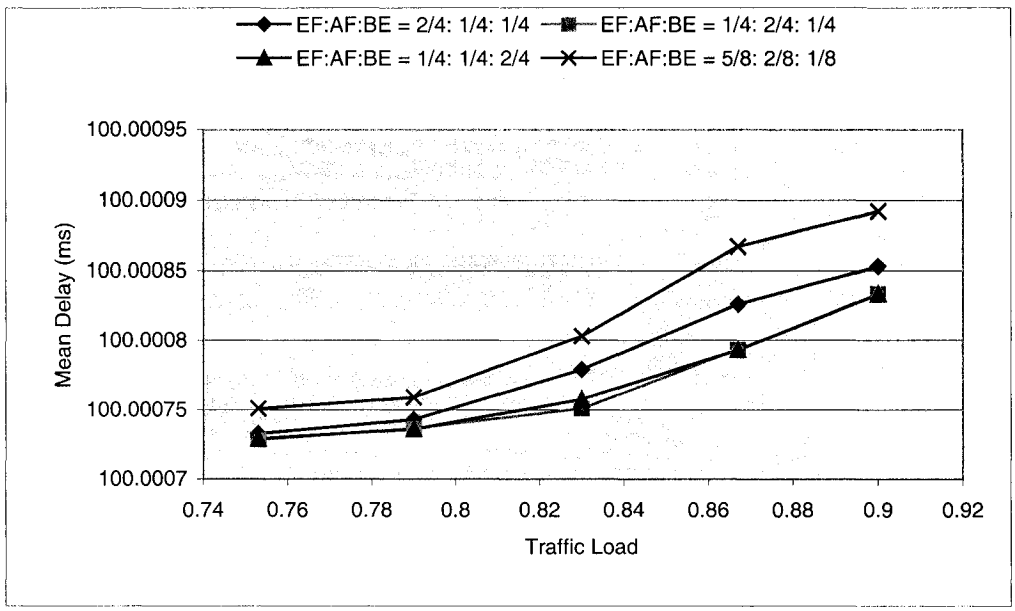
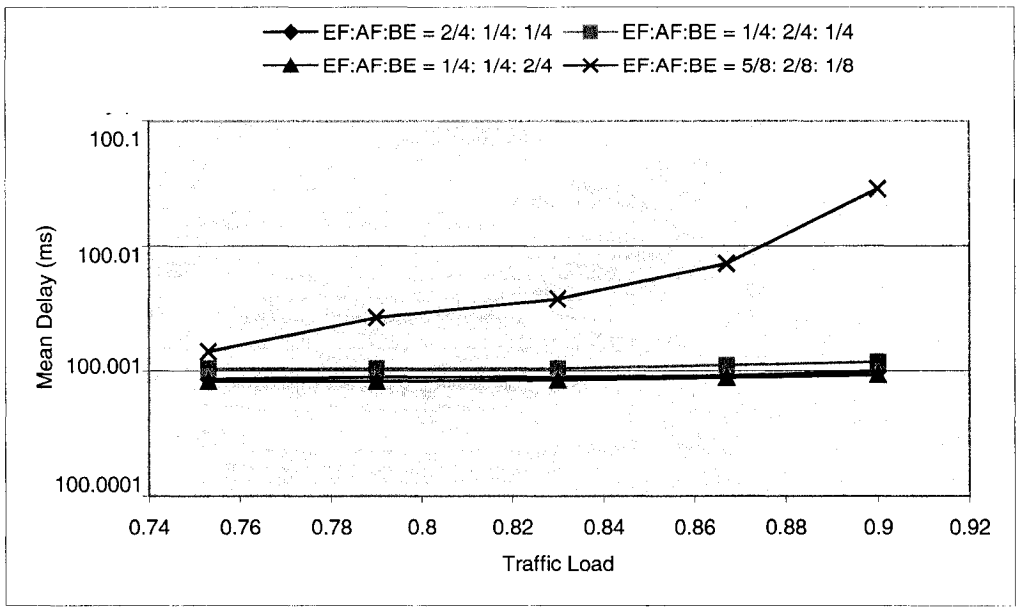
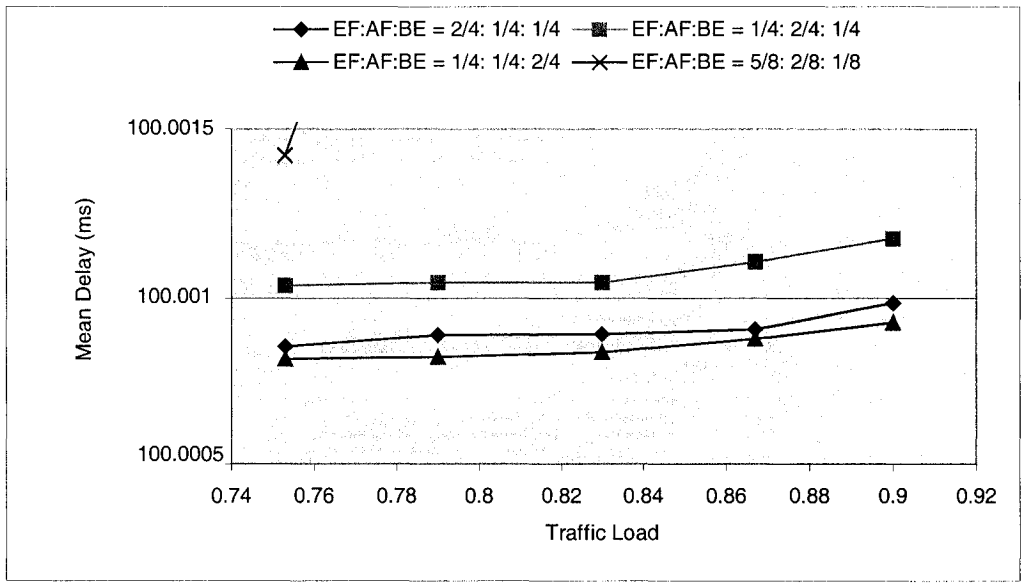


Figure 6.110 EF Traffic end-to-end Mean Delay vs. Traffic Load (with PQ under the different proportions of traffic volume, EF: AF: BE = 2/4 : 1/4 : 1/4, EF: AF: BE = 1/4 : 2/4 : 1/4, EF: AF: BE = 1/4 : 1/4 : 2/4, and EF: AF: BE = 5/8 : 2/8 : 1/8)



(a)



(b)

Figure 6.111 EF Traffic end-to-end Mean Delay vs. Traffic Load (with WRR under the different proportions of traffic volume, EF: AF: BE = 2/4 : 1/4 : 1/4, EF: AF: BE = 1/4 : 2/4 : 1/4, EF: AF: BE = 1/4 : 1/4 : 2/4, and EF: AF: BE = 5/8 : 2/8 : 1/8)
(Note: figure (b) is the higher resolution of figure (a))

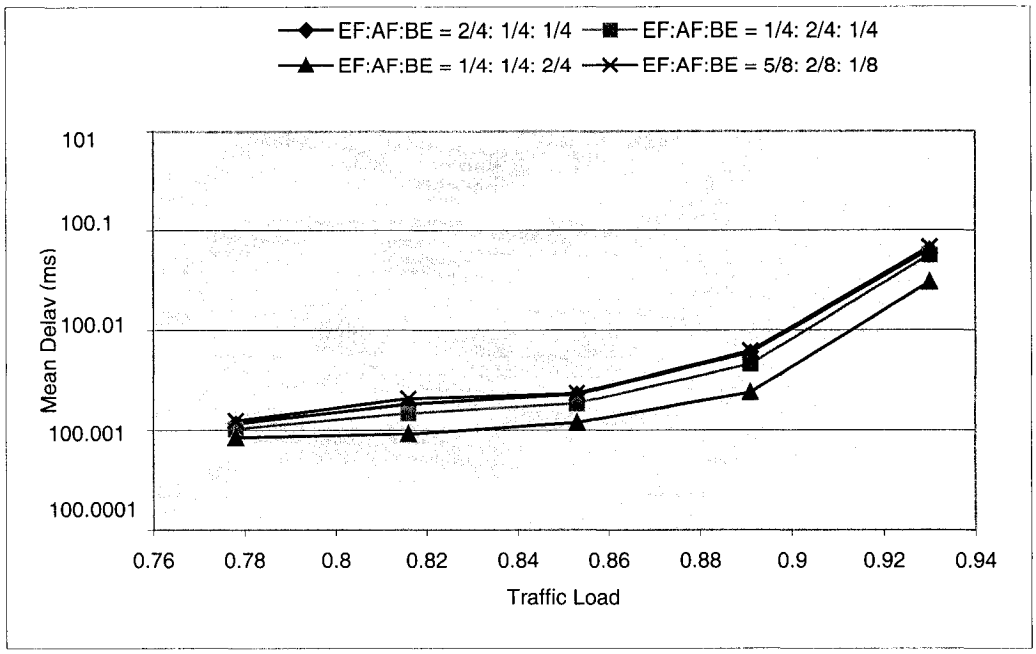


Figure 6.112 AF Traffic end-to-end Mean Delay vs. Traffic Load (with PQ under the different proportions of traffic volume, EF: AF: BE = 2/4 : 1/4 : 1/4, EF: AF: BE = 1/4 : 2/4 : 1/4, EF: AF: BE = 1/4 : 1/4 : 2/4, and EF: AF: BE = 5/8 : 2/8 : 1/8)

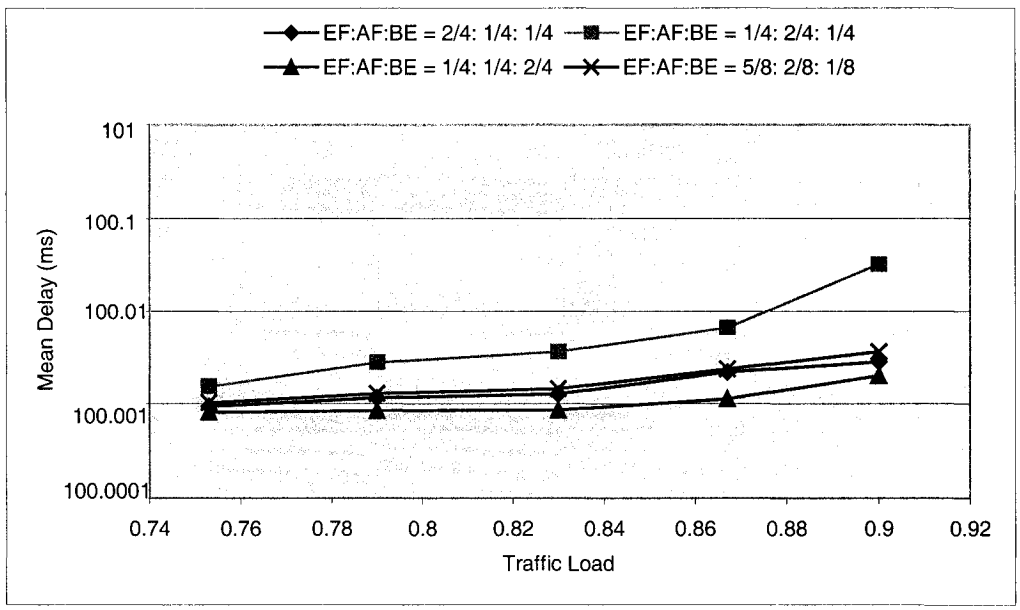


Figure 6.113 AF Traffic end-to-end Mean Delay vs. Traffic Load (with WRR under the different proportions of traffic volume, EF: AF: BE = 2/4 : 1/4 : 1/4, EF: AF: BE = 1/4 : 2/4 : 1/4, EF: AF: BE = 1/4 : 1/4 : 2/4, and EF: AF: BE = 5/8 : 2/8 : 1/8)

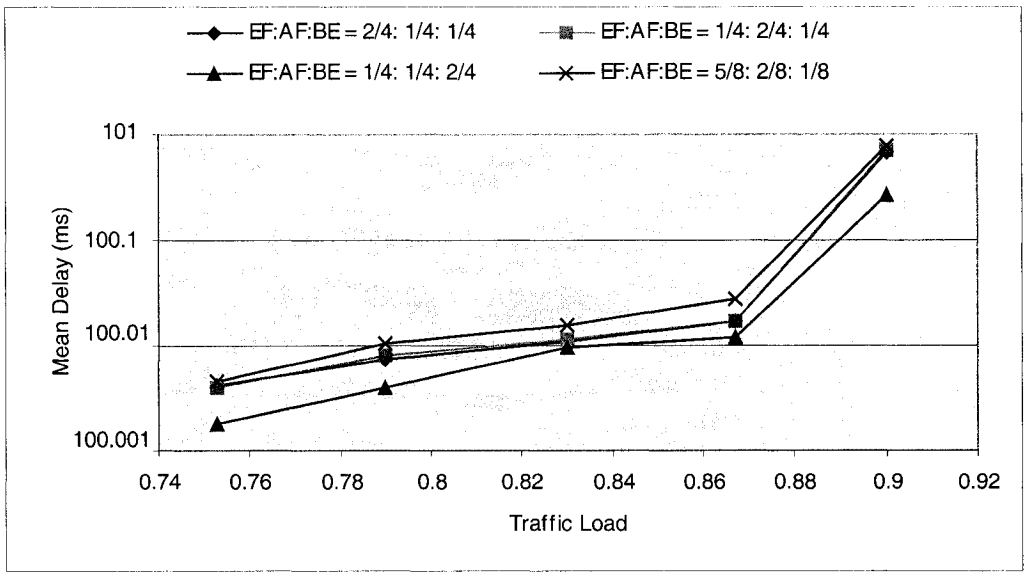


Figure 6.114 BE Traffic end-to-end Mean Delay vs. Traffic Load (with PQ under the different proportions of traffic volume, EF: AF: BE = 2/4 : 1/4 : 1/4, EF: AF: BE = 1/4 : 2/4 : 1/4, EF: AF: BE = 1/4 : 1/4 : 2/4, and EF: AF: BE = 5/8 : 2/8 : 1/8)

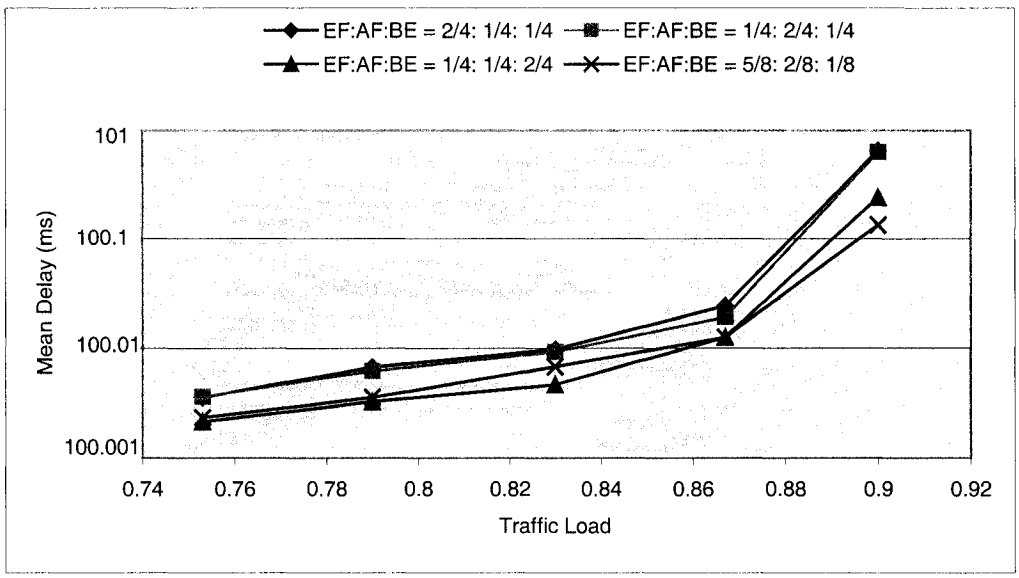


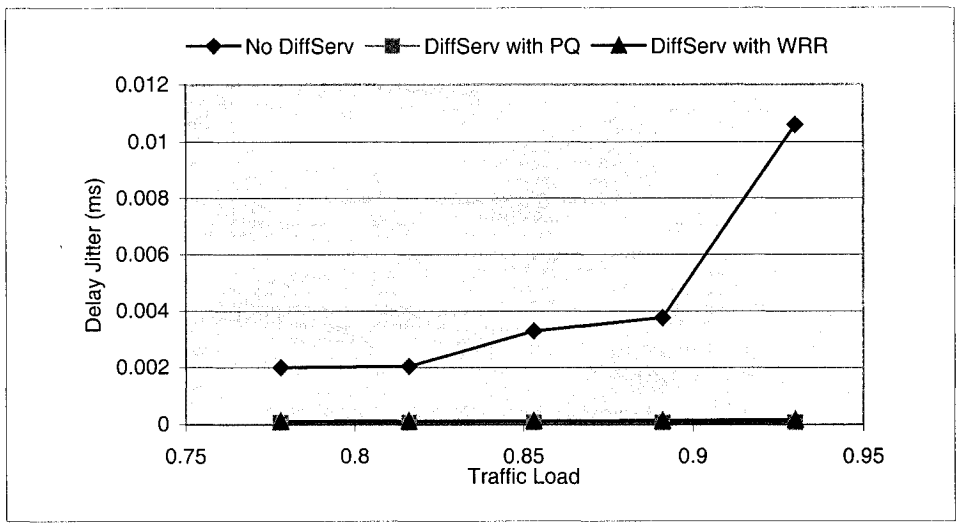
Figure 6.115 BE Traffic end-to-end Mean Delay vs. Traffic Load (with WRR under the different proportions of traffic volume, EF: AF: BE = 2/4 : 1/4 : 1/4, EF: AF: BE = 1/4 : 2/4 : 1/4, EF: AF: BE = 1/4 : 1/4 : 2/4, and EF: AF: BE = 5/8 : 2/8 : 1/8)

The average end-to-end delay experienced by EF packets is plotted in Figure 6.107. As expected, the Non DiffServ capable end-to-end delay is much larger than DiffServ capable end-to-end delay, because a considerable larger amount of EF packets are transmitted

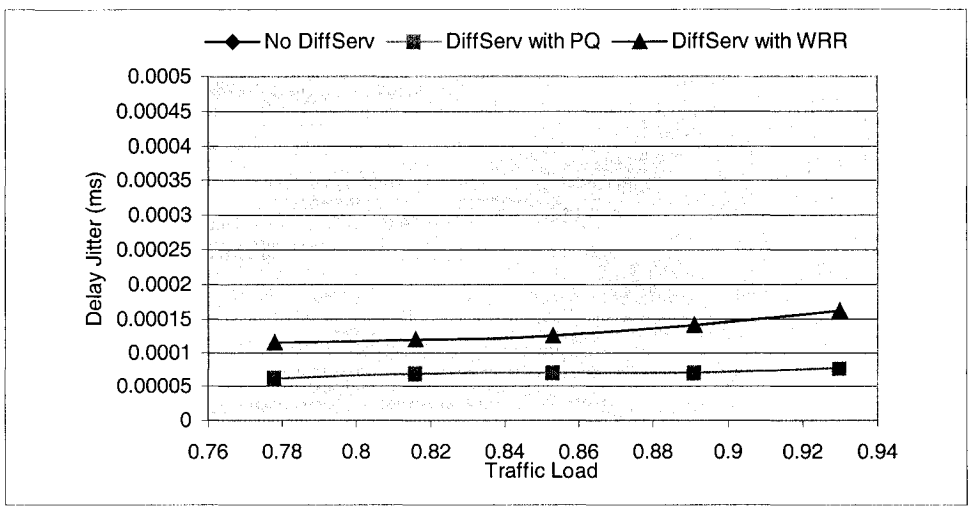
through tandem path. Also Weighted Round Robin has slightly larger average end-to-end delay than Priority Queuing as expected, because in WRR all queues are serviced, but in PQ the EF packets will be serviced firstly. On the other hand, figure 6.108 shows that the average end-to-end delay of AF traffic in QoS-Aware PetaWeb network is less than in the non-DiffServ capable PetaWeb network, due to the less amount of AF packets are transmitted through tandem path; figure 6.109 shows that the average end-to-end delay of best effort traffic in QoS-Aware PetaWeb network is larger than in the non-DiffServ capable PetaWeb network; also the figure illustrates that average end-to-end delay in the network with PQ is higher than that with WRR (which was as expected as well). For the BE queue is serviced each round in WRR; but in PQ, the BE queue is serviced only when the EF and AF queues are empty. So the BE traffic delay in PQ is higher.

Furthermore, figure 6.110 to figure 6.115 show the effect of the different proportions of EF, AF, and BE traffic on the EF, AF, and Best Effort traffic mean delay. In figure 6.110 and figure 6.111, we can see that in PQ scheduling scenario, increasing AF traffic load and BE traffic load, there is no much different influence on the EF traffic mean delay; however compared with increasing the EF traffic load, the EF traffic encounters less mean delay. The reason is the EF traffic is always serviced first in the PQ, so the effect on the EF traffic is considerably small while increasing AF and BE traffic load; On the other hand, increasing the proportion of EF traffic, the probability of EF traffic waiting for service in queue is higher, so the mean delay of EF traffic is higher. In WRR scheduling scenario, increasing the AF traffic load, the EF traffic encounters larger mean delay when compared with increasing the BE traffic load; This is because in WRR, the different queue is serviced according to their weights, and the AF queue weight is larger than the BE queue weight, so increasing AF traffic, the more AF traffic will be service in each round, as a result, the EF traffic queuing delay is larger. Figures 6.112 to 6.115 display that when increasing BE traffic load, the AF traffic and BE traffic encounters less mean delay than increasing EF and AF traffic load, no matter which kind of scheduling schemes are used; in WRR, increasing AF traffic load, the AF traffic encounters much larger mean delay than increasing EF and BE traffic load, due to the limitation of AF queue weight; but while increasing EF and AF traffic load, the BE traffic encounters nearly same influence on the

mean delay. The reason is the larger proportion of EF or AF traffic, the lower probability the BE queue is served in shorter time, and the more BE traffic is transmitted through tandem paths, so the delay is larger while increasing the proportion of EF or AF traffic. From figure 6.111, we see that, as expected, the EF traffic mean delay increases considerably when the EF traffic load over reserved bandwidth in WRR, due to the amount of EF traffic transmitted through tandem paths increases a lot; however, in PQ, the EF traffic mean delay only increases a little. From these results, we can conclude that using QoS-Aware PetaWeb network, the EF traffic end-to-end delay requirements can be satisfied through properly design.



(a)



(b)

Figure 6.116 EF Traffic Delay Jitter vs. Traffic Load (the proportion of traffic volume is EF: AF: BE = 1/3 : 1/3 : 1/3) (Note: figure (b) is higher resolution of figure (a))

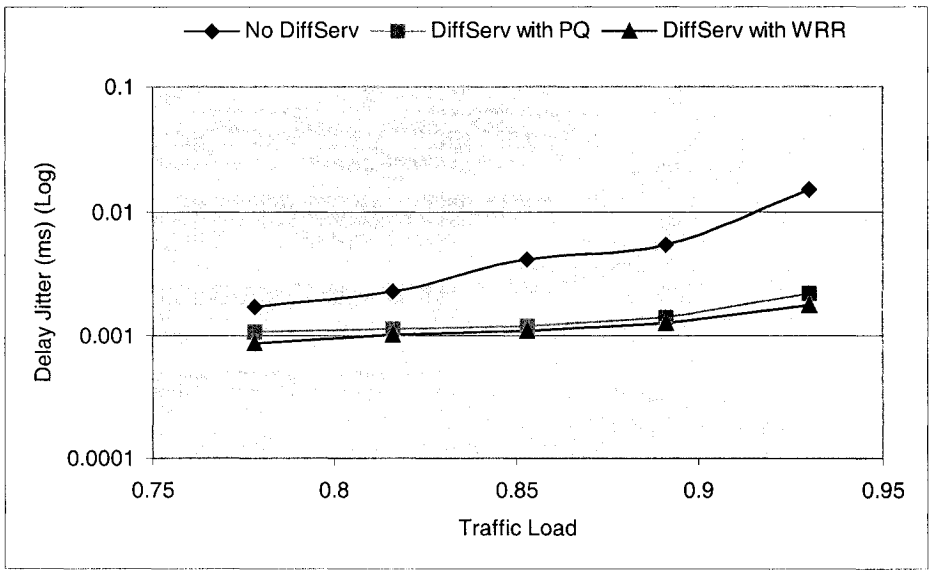


Figure 6.117 AF Traffic Delay Jitter vs. Traffic Load (the proportion of traffic volume is EF: AF: BE = 1/3 : 1/3 : 1/3)

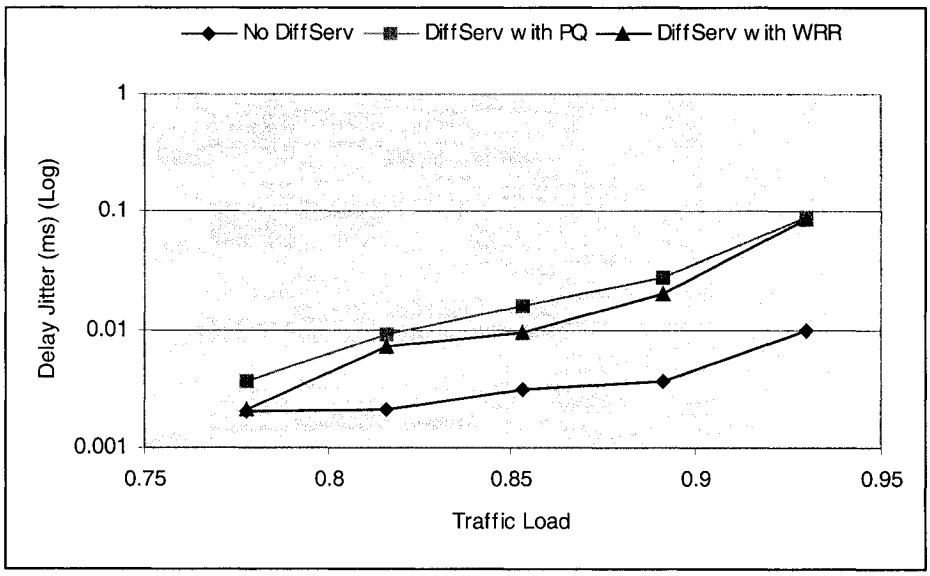


Figure 6.118 BE Traffic Delay Jitter vs. Traffic Load (the proportion of traffic volume is EF: AF: BE = 1/3 : 1/3 : 1/3)

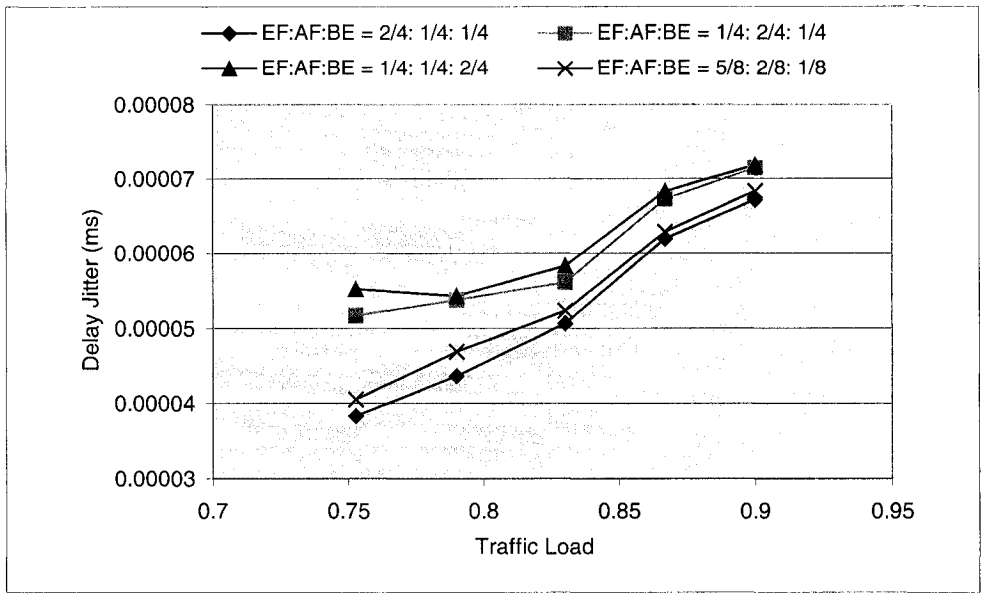


Figure 6.119 EF Traffic Delay Jitter vs. Traffic Load (with PQ scheduling under the different proportions of traffic volume, EF: AF: BE = 2/4 : 1/4 : 1/4, EF: AF: BE = 1/4 : 2/4 : 1/4, EF: AF: BE = 1/4 : 1/4 : 2/4, and EF: AF: BE = 5/8 : 2/8 : 1/8)

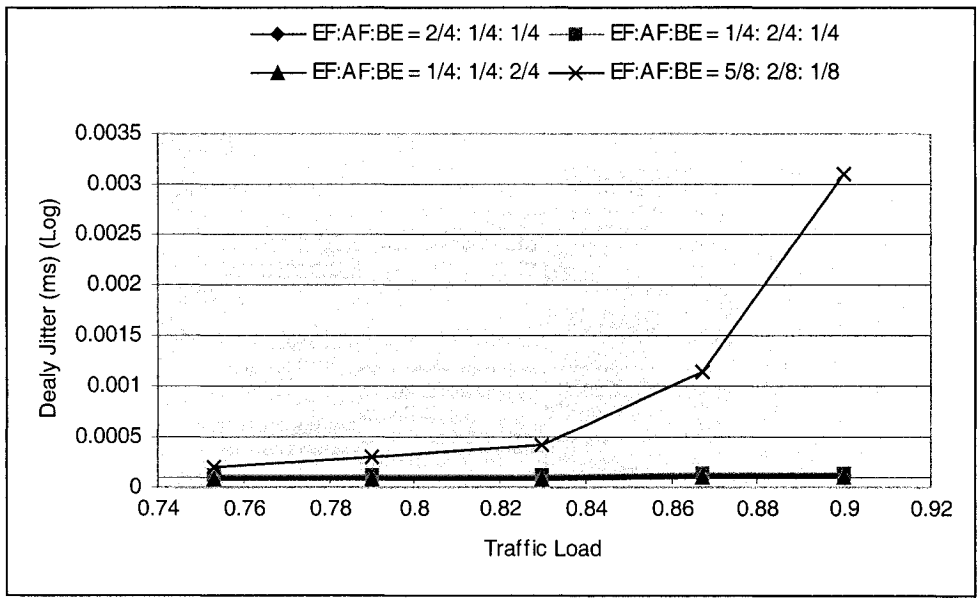


Figure 6.120 EF Traffic Delay Jitter vs. Traffic Load (with WRR scheduling under the different proportions of traffic volume, EF: AF: BE = 2/4 : 1/4 : 1/4, EF: AF: BE = 1/4 : 2/4 : 1/4, EF: AF: BE = 1/4 : 1/4 : 2/4, and EF: AF: BE = 5/8 : 2/8 : 1/8)

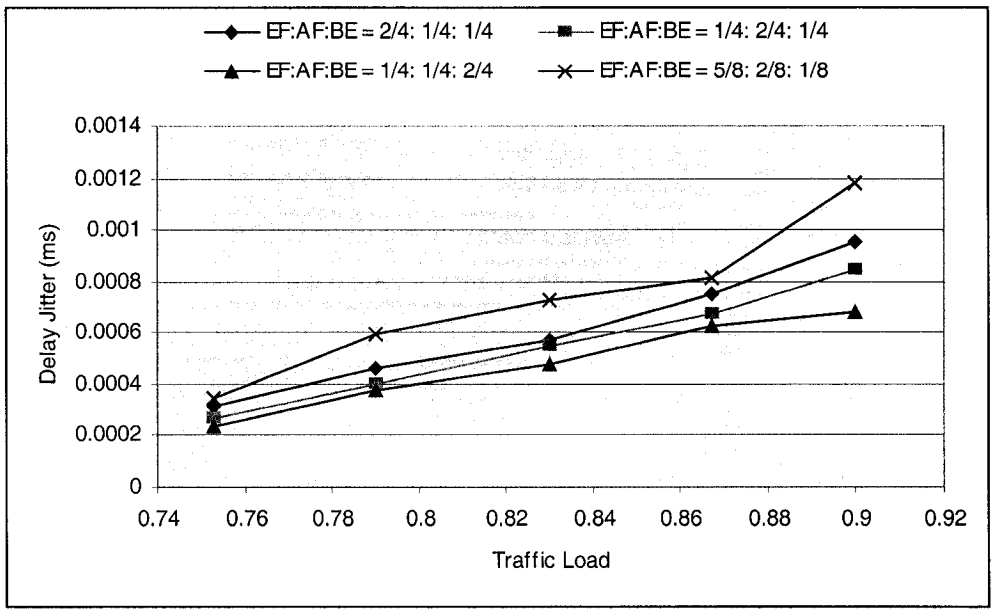


Figure 6.121 AF Traffic Delay Jitter vs. Traffic Load (with PQ scheduling under the different proportions of traffic volume, EF: AF: BE = 2/4 : 1/4 : 1/4, EF: AF: BE = 1/4 : 2/4 : 1/4, EF: AF: BE = 1/4 : 1/4 : 2/4, and EF: AF: BE = 5/8 : 2/8 : 1/8)

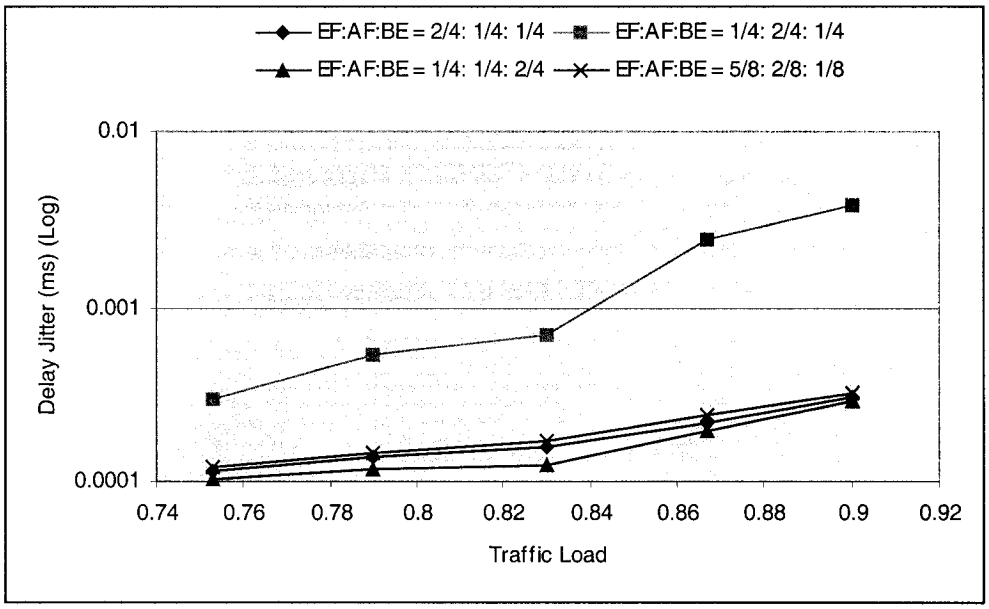


Figure 6.122 AF Traffic Delay Jitter vs. Traffic Load (with WRR scheduling under the different proportions of traffic volume, EF: AF: BE = 2/4 : 1/4 : 1/4, EF: AF: BE = 1/4 : 2/4 : 1/4, EF: AF: BE = 1/4 : 1/4 : 2/4, and EF: AF: BE = 5/8 : 2/8 : 1/8)

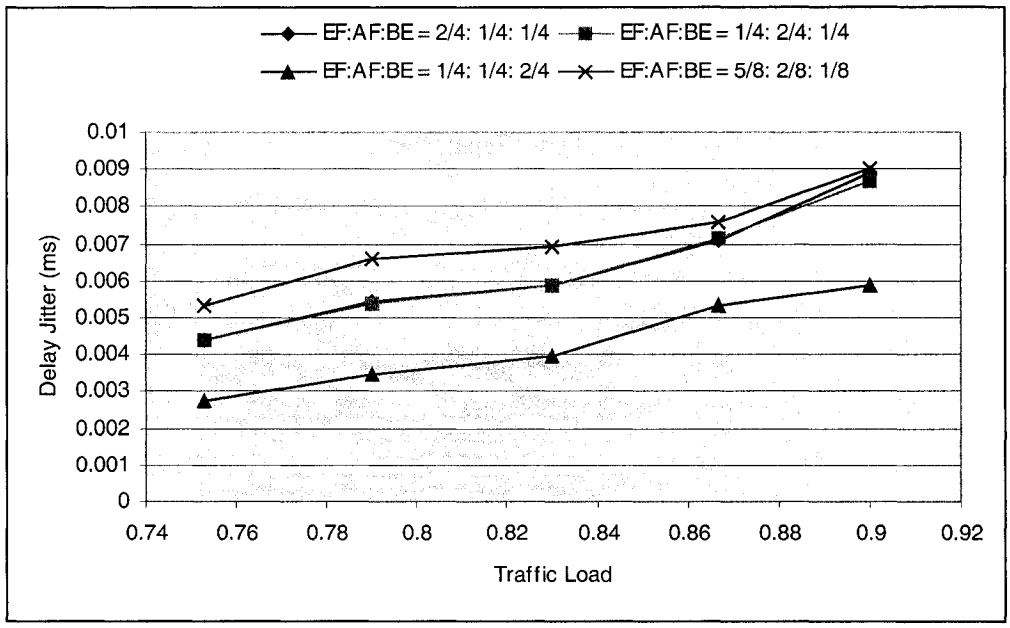


Figure 6.123 BE Traffic Delay Jitter vs. Traffic Load (with PQ scheduling under the different proportions of traffic volume, EF: AF: BE = 2/4 : 1/4 : 1/4, EF: AF: BE = 1/4 : 2/4 : 1/4, EF: AF: BE = 1/4 : 1/4 : 2/4, and EF: AF: BE = 5/8 : 2/8 : 1/8)

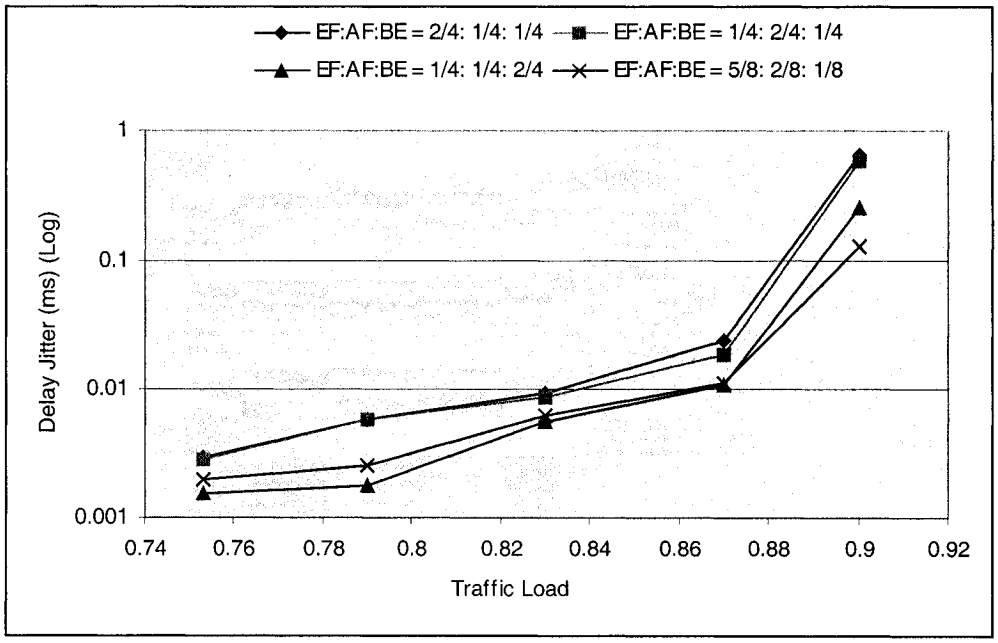


Figure 6.124 BE Traffic Delay Jitter vs. Traffic Load (with WRR scheduling under the different proportions of traffic volume, EF: AF: BE = 2/4 : 1/4 : 1/4, EF: AF: BE = 1/4 : 2/4 : 1/4, EF: AF: BE = 1/4 : 1/4 : 2/4, and EF: AF: BE = 5/8 : 2/8 : 1/8)

The delay jitter (or standard deviation of delays, is calculated by $\sqrt{\frac{1}{N}\sum_{i=1}^{N}(D_i - \overline{D})^2}$ where D_i is the delay of i_{th} packet, $\overline{D}$ is the mean delay, N is the number of packets) is also one of the key metrics that affects the quality of service of the traffic. Figures 6.116, 6.117, and 6.118 illustrate the EF, AF, and BE traffic delay jitters in QoS-Aware PetaWeb with different scheduling disciplines and in Non-DiffServ capable PetaWeb network. We can find in the general PetaWeb network, due to tandem paths exist, without quality of service functionality, the amount of packets, which are transmitted through tandem paths in EF and AF traffic, are considerable larger than in the QoS-Aware PetaWeb network. These packets encounter much longer propagation delay (200 milliseconds from source to destination in the experiments); as a result, the EF and AF traffic will encounter much higher delay jitter in Non-DiffServ capable PetaWeb network. Conversely, for Best Effort traffic, the amount of tandem packets is considerably higher in QoS-Aware PetaWeb with PQ and WRR scheduling disciplines, so the delay jitter of best effort traffic is higher as well. As expected, in the Non-DiffServ capable PetaWeb network the EF traffic delay jitter is much higher than that in QoS-Aware PetaWeb network with any scheduling, due to the fact that some packets are transmitted through tandem paths; also using Priority Queuing has smaller average delay jitter than using Weighted Round Robin. The reason is that in the PQ, the EF packet is always serviced before the AF and Best Effort packets; the EF packets are inserted into buffer, only when the server is servicing other EF packets; however, in the WRR, the EF, AF, and BE packets are serviced according to their predetermined weights, thus, no full priority is given to the EF traffic.

Also figures 6.119 to 6.124 show the effect of the different proportions of EF, AF, and BE traffic on the delay jitter. In figure 6.119 and figure 6.120, we can see that in the PQ scheduling scenario, increasing the AF and BE traffic load, there is no much different influence on the EF traffic delay jitter; this is because the EF packets are always serviced first; however, as expected, increasing the EF traffic load, even the EF traffic load over reserved bandwidth, the EF traffic encounters less delay jitter, compared with increasing AF traffic load and BE traffic load. In WRR scheduling scenario, increasing the AF traffic

load, the EF traffic encounters larger delay jitter when compared with increasing the EF and BE traffic load; while increasing the BE traffic load, the EF traffic encounters less delay jitter when compared with increasing the EF and AF traffic load. This is because the increasing AF traffic, the more AF packets would be serviced, so the EF packets queuing delay will be higher. If the EF traffic load over reserved bandwidth, the delay jitter of EF traffic is considerably higher as expected, due to some of EF traffic are transmitted through tandem path. From figures 6.121 to 6.124, we see that in PQ, increasing AF traffic load, the delay jitter of AF traffic is lower than increasing EF traffic load, but higher than increasing BE traffic load, due to the AF traffic has middle priority; however, in the WRR, increasing AF traffic load, the AF traffic encounters considerably higher delay jitter than increasing EF and BE traffic load, due to the amount of AF traffic transmitted through tandem path is much higher. As expected, increasing the BE traffic load, the BE traffic encounters less delay jitter than increasing EF and AF traffic load, no matter which kind of scheduling schemes are used; but while increasing EF and AF traffic load, the BE traffic encounters nearly same influence. This is because increasing EF and AF traffic load, the proportion of BE tandem traffic is higher than increasing BE traffic, so the delay jitter is higher.

6.9 The TCP Performance over PetaWeb Network

As we know, the TCP end-to-end congestion control mechanisms, force the TCP source to back off whenever congestion is detected in the network. However, on the other hand, non-adaptive sources such as those running on User Datagram Protocol (UDP) continue to send traffic into network at the same traffic rate during congestion. It will starve well-behaved TCP sources and get greater shares of the network bandwidth. Considerable research has been performed on TCP. A number of TCP schemes and a variety of control mechanisms have been proposed in order to optimize TCP behavior [74] [75].

However, just as we demonstrated before, due to its agile nature, the PetaWeb network can provide higher network performance. The channel connectivity between core nodes and edge nodes would be changed dynamically at specific time interval based on the traffic load at each edge nodes. The channel capacity at a pair of edge nodes varies dynamically. As result, the network congestion problem is almost diminished. Thus, we would expect

that TCP traffic would be able to take advantage of this. However, as PetaWeb network core nodes dynamically reconfigure channel connectivity, they may affect the TCP traffic behavior. In this section, we investigate the TCP performance over a PetaWeb network, and compare with the performance of TCP over a static core network.

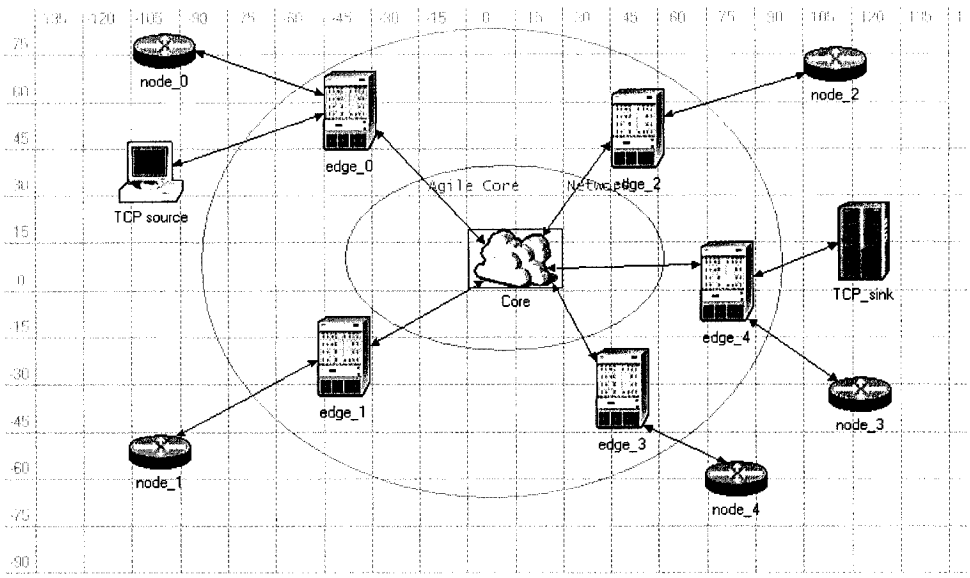


Figure 6.125 PetaWeb network with TCP connection Simulation Model

We use combined alpha-stable and sinusoidal traffic model (see Figure 5.4, traffic pattern in source node) as UDP traffic source, also defined as background traffic, and added one TCP traffic source at Source 0 in the PetaWeb network (see Figure 6.125, PetaWeb simulation model), and which destination is TCP sink. Currently, WWW traffic is the dominant traffic in the Internet [76]; WWW data is transmitted using HTTP (Hypertext Transfer Protocol) [77], which uses TCP as transport protocol. TCP connections are established by the WWW client prior to each request (e.g., to access a WWW site) and closed by the WWW server after sending the requested data (e.g., HTML document).

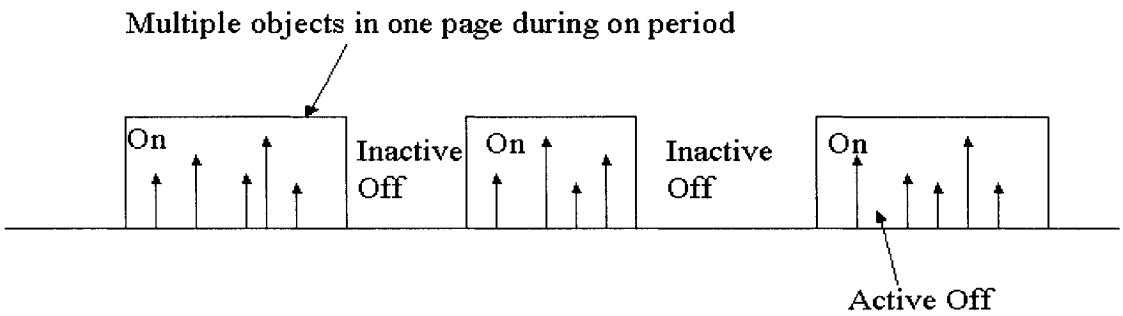


Figure 6.126 Web User Equivalents (WUE) Model

In the simulation, WWW application is used as the TCP traffic source. The Web User Equivalent (WUE) model [78] is used to generate the WWW traffic (illustrated in Figure 6.126). Typically when a user sends a request for a new web page, multiple objects embedding in the page are transmitted back to the user, which corresponds to an ON period. After the burst of object files, a user will spend some think (OFF) time to study the downloaded web page without action. A new ON period starts when the user requests another new page. This behavior is completely captured by the distribution of the requested object file sizes, the embedded references in the ON periods and the OFF periods.

The workload generated by this model is roughly correspond to that generated by a population of some known number of users. In the experiment, the number of users is set to 300, the transmission rate of each individual user is set close to 33 Kbps, then the total traffic average rate is nearly 10 Mbps. TCP Reno is used at the ends of the TCP connection in the simulation, which is one of the most common TCP flavors in the Internet. The TCP segment size is set to 512 bytes.

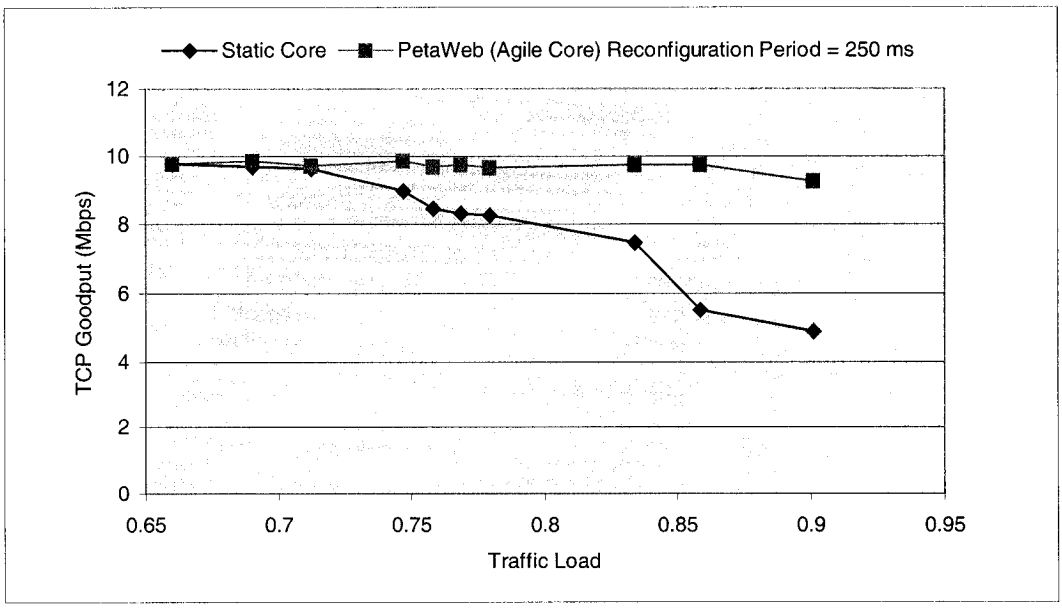


Figure 6.127 TCP Performances for the PetaWeb vs. the Static Core Network

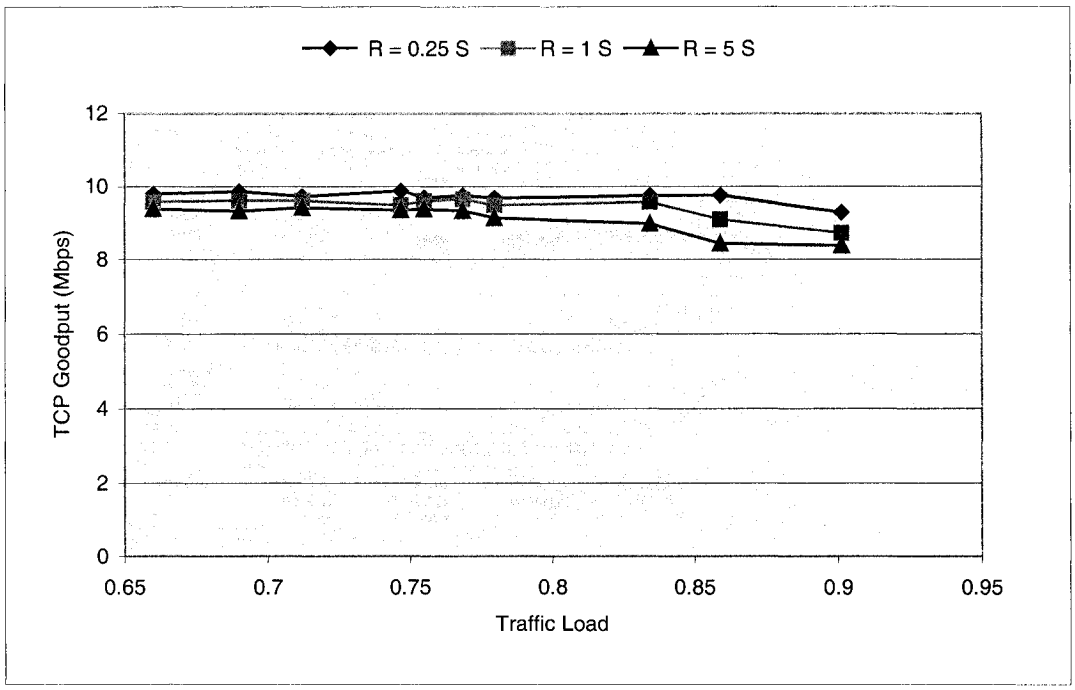


Figure 6.128 TCP Goodput vs. Reconfiguration Period

In the experiments, we investigate the TCP performance through changing background traffic load (normalized to the link capacity) under static core network and PetaWeb network; the results are shown in Figure 6.127. TCP goodput is used as one parameter to evaluate the TCP performance. The definition of goodput is the effective bandwidth delivered to the receiver, excluding overhead and retransmission packets. From Figure 6.127 we can see that as the traffic load increases, the TCP goodput is not affected too much in the PetaWeb network. Conversely, the TCP goodput decreases sharply in the static core network. This demonstrates the benefits of PetaWeb, seeing from the point of view of a very common and popular application.

We also investigate how the reconfiguration period affects TCP behavior. Figure 6.128 shows that the larger the reconfiguration period is, the lower the TCP throughput becomes. However, the TCP performance is not changed a lot under different reconfiguration period.

6.10 Summary

In this chapter, we have evaluated the PetaWeb network performance by investigating some factors, which may impact PetaWeb network operations. Through compared with static core network, the advantages of the PetaWeb networks are presented through a set of numerical results. In brief, the PetaWeb network can provide higher network throughput, better transport resource utilization, lower traffic delay, and have resilience to traffic fluctuations. Furthermore, we demonstrated that the QoS-Aware PetaWeb network could achieve end-to-end quality of service requirements. Finally, by studying TCP performance over the PetaWeb network, we find that the TCP connection can perform very well over the PetaWeb network, but in static core network TCP goodput is considerably lower. It demonstrates, once more, that the PetaWeb networks gain considerable of benefits due to its adaptability to the traffic demand.

Chapter 7

CONCLUSION AND FUTURE WORK

7.1 Conclusions

This dissertation concentrates on the issues associated with the architecture of Next Generation Optical Network, which should be a wide-coverage high capacity and scalable data network. We are focusing on a possible architecture for the next generation network, called PetaWeb network. The functionality of the PetaWeb network has been verified, and its performance also has been quantified. Some issues impacting the performance benefits of the PetaWeb network have been investigated, such as reconfiguration period, the number and the granularity of optical channels, and traffic characteristics. The QoS-Aware PetaWeb network has been proposed in order to provide quality of service over the network. A variety of parameters have been investigated including packet lose ratio, packet-forwarding delay, as well as delay jitter. Different scheduling policies are used while studying the network performance, and the impacts to the network performance are also investigated.

In the simulator, we implement the main logic components of the PetaWeb network: the edge node, the agile core node, as well as the channel allocation algorithm. The simulator also includes two kinds of traffic models. The sinusoidal traffic model is used to model the macroscopic traffic variations; the self-similar (alpha-stable) traffic model is used to represent the microscopic behavior. The fundament of these is based on the daily traffic pattern. However, prediction of future Internet traffic distribution is quit difficult. It is precisely for this reason that an adaptive network like the PetaWeb network is so attractive. Through reconfiguring the network core nodes, the PetaWeb can adapt traffic

demand by re-allocating optical channels between edge nodes. It allows most of the traffic to be transmitted through direct channels, without suffering O-E-O conversions. From the performance point of view, this offers some very attractive advantages. It provides low packet latency, only limited by the propagation delay. It ensures resilience to spatial and temporal traffic changes. It simplifies traffic engineering, and the hub-and-spoke architecture brings scalability up to multiples of Petabits per second. In addition to these, the PetaWeb can provide better transport resource utilization and higher network throughput than the static core network, due to the fact that the tandem traffic is eliminated or reduced significantly. This can translate into cost savings. The simulation results also indicate that a high reconfiguration frequency, as well as a fine granularity of the optical channels is desirable for the PetaWeb performance.

Furthermore, in this dissertation, we proposed and implemented a QoS-Aware PetaWeb network. We map the DiffServ to the PetaWeb. The PetaWeb edge nodes are DiffServ-enabled, and a QoS-Aware channel allocation algorithm is proposed. The QoS-Aware channel allocation algorithm has been assessed through simulation. By comparison with the traditional algorithm, the packet loss ratio for high priority traffic is sharply reduced; the end-to-end delay and delay jitter also are much lower.

7.2 Future Work

While considering the effort we made and the results we presented in this dissertation, it is no doubt that what we did here is limited, just the beginning of research in the network architecture of next generation optical networks. We provide below a number of suggestions for future research work.

In the reported work, the network model we developed is used to understand the performance of the PetaWeb network, so the features that have the largest effect on agility and the quantification of the results, and only a single core node in agile core were implemented. A natural extension of the model to multiple core nodes deserves to be considered and implemented in the future work, even though it does not improve significantly the accuracy of the results. The interest lies more in the demonstration and the prototyping than in the performance analysis.

Also, the results presented in this work and the analysis was based on the use assumed traffic patterns. The research to predict the evolution of traffic characteristics on large backbone networks while considering the future applications such as optics to the desktop, tele-immersive virtual reality, 3D interactive gaming, distance education and tele-medicine, is desirable.

In this dissertation, we proposed and implemented the QoS-Aware PetaWeb network model to support the quality of service. We map the DiffServ to the PetaWeb and the PetaWeb edge nodes are DiffServ-enabled. However, the QoS architecture for the PetaWeb network also requires further study, such as Multi-Protocol Label Switching (MPLS) and RSVP interworking with the PetaWeb.

Furthermore, the different switching granularities can be exploited in the PetaWeb network. The different switching modes are associated with different granularities. In this dissertation, the channel switching is used. In the future, the PetaWeb with other switching modes, such as link switching, channel-band switching, time division multiplexed (TDM) switching, and burst switching, or even composite multiple grained switching modes, can be considered.

REFERENCES

[1] J. E. Smith and F. W. Weingarten, Eds., "Research Challenges for the Next Generation Internet", Computing Research Association, May 1997, Report from the Workshop on Research Directions for the Next Generation Internet.

[2] http://ietf.org/

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

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

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

[6] http://www.isc.org/ds/

[7] http://glreach.com/globstats/

[8] http://caida.org

[9] http://www.abilene.iu.edu/

[10] C. Siva, R. Murthy, M. Gurusamy, "WDM Optical Networks: Concepts, Design, and Algorithms," Prentice Hall PTR 2001.

[11] S. Banerjee and C. Chen," Design of wavelength-routed optical networks for circuit switched traffic," In Proc. IEEE Globecom, London, November 1996.

[12] N. Wauters and P. Demeester, "Design of the optical layer in multiwavelength cross-connected networks," IEEE J. Select. Areas Commun., vol. 14, pp. 881–892, 1996.

[13] L. Wuttisittikulkij and M. J. O'Mahony, "Design of efficient and practical algorithm for wavelength assignment in multi-wavelength ring transport networks," in Proc. IEEE GLOBECOM'97, Nov. 1997, pp. 571–575.

[14] S. Subramaniam and R. A. Barry, "Wavelength assignment in fixed routing WDM networks," in *Proc. IEEE Int. Conf. Commun.*, June 1997, pp. 406–410.

[15] Birman, "Rougin and wavelength assignment methods in single-hop all-optical networks with blocking," In Proc. IEEE Infocom, Boston, April 1995.

[16] S. Subramaniam, M. Azizoglu, and A. K. Somani, "All-optical networks with sparse wavelength conversion," IEEE/ACM Trans. Networking, 4(4): 544-557, 1996.

[17] E. Hyytia and J. Virtamo, "Wavelength assignment in multifiber networks," COST257 TD (99) 04, January 1999.

[18] K. C. Lee and V. O. K. Li, "A Wavelength Rerouting Algorithm in Wide-Area All-Optical networks," IEEE J. Lightwave Technology, Vol.14, Jun.1996.

[19] Mohan and C. Siva Ram Murthy, "A Time optimal Wavelength Rerouting Algorithm for Dynamic Traffic in WDM Networks," IEEE J. Lightwave Technology, Vol.17, Mar.1999.

[20] F. J. Blouin, A.. W. Lee, A. J.M. Lee, and M. Beshai, "Comparison of two optical-core networks," Journal of optical networking, Vol.1, No.1, January, 2002.

[21] R. Vickers and M. Beshai, "PetaWeb Architecture," presented at Networks 2000—Toward Natural Networks: 9th International Telecommunication Network Planning Symposium, Toronto, Canada, 10–15 Sept. 2000.

[22] M. Beshai, F. Blouin, and R. Krishnan, "PetaWeb—building block for a yottabit-per-second network," report to DARPA Next Generation Internet, Technology Investment Agreement TIA F30602-98-2-0194 (2001).

[23] http://www.nortelnetworks.com

[24] M. Erman, "Trends and evolution of optical networks and technologies," Alcatel Telecommunication Review - 3rd Quarter 2001.

[25] Lucent Technologies. Dense Wavelength Division Multiplexing (DWDM) Tutorial, www.iec.org/online/tutorials/dwdm.

[26] S. V. Kartalopoulos, "Introduction to DWDM Technology: Data in a Rainbow", IEEE Press 2000

[27] C. DeCusatis, "Fiber Optic Data Communication: technological Trends and Advances", Academic Press 2002

[28] International Telecommunications Union, www.itu.int

[29] M. Tachibana, R. I. Laming, P.R.Morkel, and D. N. Payne, " Gain-shaped erbium-doped fiber amplifier (EDFA) with broad spectrak bandwidth," Topical Meeting on Optical Amplifier Application, p. MD1, 1990.

[30] E. Desurvire, C.R. Gies, J. L. Zyskind, J. R. Simpson, P.C. Becker, and N. A. Olsson, "Recent advances in erbium-doped fiber amplifiers at 1.5 μm," Proc. Optical Fiber Communcation Conference, San Francisco, CA, 1990

[31] M. J. O'Mahony, " Optical Amplifiers," in Photonics in Switching vol. 1, pp. 147-167, San Diego, CA: Academic Press, 1993

[32] G. Wilfong, B. Mikkelsen, C. Doerr, M. Zirngibl, "WDM Cross Connect Architectures with Reduced Complexity", Journal of Lightwave Technology, vol 17, pp. 1732, 1999.

[33] S. Johansson, M. Lindblom, M.Buhrgard. "Photonic Switching in High Capacity Transport Networks", 14th. International Switching Symposium, 1992, 2(B9.1)

[34] http://www.insight-corp.com/photonics.html

[35] H. A. Stavdas, E. N. Protonotarios, J. E. Midwinter, "An OXC Architecture Suitable for High Density WDM Wavelength Routed Networks", Photonics Network Communications, vol. 1, pp 77, 1999

[36] G. Wilfong, B. Mikkelsen, C. Doerr, M. Zirngibl, "WDM Cross Connect Architectures with Reduced Complexity", Journal of Lightwave Technology, vol 17, pp. 1732, 1999.

[37] R. Ramaswami and K. N. Sivarajan, ``Optical Networks: A Practical Perspective," Second Edition, Morgan-Kaufman, Nov. 2001

[38] B. Mukherjee, " Optical Communication Networks," McGraw-Hill, July 1997

[39] T. Stern and K. Bala, 'Multiwavelength Optical Network - A Layered Approach,' Addison Wesley Longman, Inc, May 1999.

[40] M. S. Goodman, H. Kobrinski, M. Vecchi, R. M. Bulley, and J. M. Gimlett. The LAMBDANET multiwavelength network: Architecture, applications and demonstrations. IEEE Journal on Selected Areas in Communications, 8(6):995-1004, August 1990.

[41] F. J. Janniello, R. Ramaswami, and D. G. Steinberg. A prototype circuit-switched multi-wavelength optical metropolitan-area network. IEEE/OSA Journal of Lightwave Technology, 11:777-782, May/June 1993

[42] N.R. Dono, P.E. Green, K. Liu, R. Ramaswami, and F.F.Tong. A Wavelength Division Multiple Access Network for Computer Communication. IEEE Journal on Selected Areas in Communications, 8:983-994, August 1990.

[43] P. Dowd, J. Perreault, J. Chu, D. Hoffmeister, R. Minnich, D. Burns, F. Hady, Y. Chen, M. Dagenais, and D. Stone. *LIGHTNING network and systems architecture.* IEEE/OSA Journal Of Lightwave Technology, 14(6):1371--1387, 1996

[44] T.K. Chiang, S. K. Agrawal, D. Mayweather, D. Sadot, C. F. Barry, M. Hickey, and L. G. Kazovsky, "Implementation of STARNET: a WDM computer communican," *IEEE JonSeleAreas in C*, **14**, No. 5, pp. 824-839, June 1996

[45] www.bell-labs.com/project/MONET

[46] http://www.ll.mit.edu/aon/

[47] R. Ramaswami, "Multi-wavelength lightwave networks for computer communication," IEEE Communication Magazine, 31 (2): 78-88, Feb, 1993

[48] I. P. Kaklamannis, et al., " A wideband all-optical WDM network," IEEE JSAC/JLT Special Issue on Optical Networks, 14 (5):780-799, June 1996

[49] S. Johansson et al. " A transport network involving a reconfigurable WDM network layer- a European demonstration," IEEE/OSA JLT/JSAC Special Issue on Multiwavelength Optical Technology and Networks, 14 (6): 1341-1348, June 1996

[50] A.M. Hill, and A. J. N. Houghton, "Optical networking in the European ACTS programme," In OFC'96 Technicla Digest, P238-239, San Jose, Feb. 1996

[51] G.R. Hill et al. "A transport network layer based on optical network elements," IEEE/OSA Journal on Lightwave Technology, 11: 667-679, May/June 1993

[52] M.E. Beshai and E.A. Munter, 'High-Capacity ATM Switch', United States Patent number 5745486, issued on April 28, 1998.

[53] D. B. Sarrazin, H. F. Jordan and V. P. Heuring, "Fiber-Optic Delay Line Memory," Applied Optics 29(5): 627-637, 1990.

[54] R. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, Sept. 1997

[55] G. Eichler, H. Hussmann, G. Mamais, I. Venieris, C. Prehofer, and S. Salsano, "Implementing Integrated and Differentiated Services for the Internet with ATM networks: A Practical Approach," *IEEE Communications Magazine*, Volume 38, Pages 132-141, January 2000

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

[57] V. Jacobson, K. Nichols, and K. Poduri, "An Expedited Forwarding PHB", RFC 2598, June 1999

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

[59] K. Nichols, V. Jacobson, and L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet", ftp://ftp.ee.lbl.gov/papers/dsarch.pdf

[60] T. Yang, D. Makrakis, "A New Service Model for Differentiated Services Architecture ", in proceedings of the 21st Biennial Symposium on Communications, Kingston, Canada, June, 2002

[61] C. S. Wang, and K. Shin, "Adaptive-Weighted Packet Scheduling for Premium Service", Proceedings of The IEEE International Conference on Communications, June 2001

[62] J. Banks, J. S. Carson, and B. L. Nelson, "Discrete-Event System Simulation," (2nd. ed.) Prentice-Hall, Upper Saddle River, NJ, (1996).

[63] http://www.opnet.com

[64] http://ncit.ca

[65] http://www.nortelnetworks.com

[66] V. Paxson, and S.Floyd, "Wide Area Traffic: The Failure of Possion Modeling," IEEE/ACM Trans. On Networking, Vol.3, NO. 3. June 1995

[67] B. Tsybakov, N. D Georganas, "On Self-Similar Traffic in ATM Queues: Definitions, Overflow Probability Bound, and Cell Delay Distribution", IEEE/ACM Transactions on Networking, Vol. 5, No. 3 (June), pp 397-408, 1997.

[68] W. Willinger, M. S. Taqqu, R. Sherman, D. V Wilson, "Self-Similarity Through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level", IEEE/ACM Transactions on Networking, Vol. 5, No. 1 (Feb.), pp 71-86, 1997.

[69] I. Norros, "On the Use of Fractional Brownian Motion in the Theory of Connectionless Networks", IEEE JSAC, Vol. 13, No. 6 (Aug.), pp 953-962, 1995.

[70] J.R. Gallardo, D. Makrakis, L. Orozco-Barbosa. " Use of alpha-stable self-similar stochastic process for modeling traffic in broadband networks". Performance Evaluation, 40(1-3), pp. 71-98, 2000

[71] D. Clark and W. Fang, " Explicit Allocation of Best Effort Delivery Service", IEEE/ACM Trans. on Networking, vol. 6, no. 4, pp. 362-373, August 1998.

[72] W. Fang, N. Seddigh, and B. Nandy, "Time Sliding Window Three Color Marker," RFC2859, IETF, March 2000.

[73] C. Semeria, "Supporting Differentiated Service Classes: Queue Scheduling Disciplines," White Paper, Juniper Networks, Inc. www.juniper.net.

[74] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP Selective Acknowledgement Options", Internet Request for Comments 2018 (rfc2018.txt) October 1996

[75] K. Fall, S. Floyd, "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP", ACM SIGCOMM Computer Communication, Vol. 26, Issue 3, July 1996

[76] K. Thompson, G. Miller, R. Wilder, "Wide-Area Internet Traffic Patterns and Characteristics", IEEE Network 2000; 11(6):10-23.

[77] T. Berners-Lee, R. Fielding, and H. Frystyk. "Hypertext Transfer Protocol -- HTTP/1.0", IETF RFC 1945, May 1996.

[78] P. Barford and M. Crovella, "Generating Representative Web Workloads for Network and Server Performance Evaluation", Proceedings of ACM SIGMETRICS Conference, Madison, WI, pp. 151-160, June 1998

[79] www.telcordia.com