ADAPTIVE TRAFFIC DISTRIBUTION IN OPTICAL BURST SWITCHING NETWORKS

LU JIA

(B.Eng.(Hons.), NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2005

Acknowledgements

I would like to thank the following personnel for their invaluable advice and help during the whole course of the project.

First and foremost, Dr Mohan Gurusamy and A/Prof Chua Kee Chaing, project supervisors, for their invaluable guidance, advice and understanding throughout the project.

Mr. Liu Yong, Research Fellow at the Optical Network Laboratory for his helpful advice and in-depth discussions on the key issues related to my research work.

I would also like to thank my parents for their love, care and encouragement which have given me the courage to overcome the difficulties all along the way.

Contents

Summary

In this thesis, the problem of adaptive traffic distribution in optical burst switching (OBS) networks has been studied. We first focus on the problem of dynamically and efficiently routing the incoming traffic at the flow level in OBS networks. Then we address the issue of online multi-path traffic routing in OBS networks based on a theoretical optimization framework.

Load balancing and multi-path traffic routing are important issues in OBS networks due to their unique features such as no electronic buffering and no/limited optical buffering at the core nodes. In the first part of the thesis, we introduce a scheme called adaptive proportional flow routing algorithm (APFRA), which performs traffic routing and adjustment at the flow level. The key idea of APFRA is to reduce network congestion by adaptively adjusting the traffic flow proportion assigned to each pre-determined link-disjoint path between each node pair based on the measurement of the impact of traffic load on each path. The algorithm works in a time-window-based manner and within each time window, a path is selected to route new incoming flows with a prescribed frequency determined by its assigned flow proportion. Once the assignment for a new flow is made, the flow will be transmitted using the same path until its departure and will not be shifted between different paths. Based on the measured “quality” at the end of each time window as well as the hop length factors of the paths, the set of assigned flow proportions for the paths between each source and destination node will be adjusted accordingly and applied to route new incoming flows in the next time window. However, the existing flows being transmitted are not affected. The packet out-of-sequence arrival problem resides in previous proposed load balancing algorithms in OBS networks since traffic flows can be disrupted and shifted between different paths. By performing traffic routing at the flow level, in effect our proposed algorithm is packet re-ordering free. Furthermore, the routing and adjustment at the flow level and in a proportional manner helps to improve the routing stability in the network. Through extensive simulations, we show that our proposed algorithm works well in practice and achieves significant burst loss improvement over the static alternate flow routing algorithms.

In the second part of the thesis, we propose a new online multi-path traffic routing scheme which is based on the gradient projection optimization framework to determine the traffic splitting or mapping among the multiple paths between each source and destination pair. The key idea is to let each source node periodically measure the offered load on the links that are traversed by the alternative paths between the source and destination pair. Then at the end of each time window, the source node calculates each path’s first derivative length to evaluate the impact of the offered burst traffic on the path. Based on the above information, we apply the gradient projection algorithm to obtain the amount of burst traffic that will be distributed to each alternative path for the next time period. Traffic flows may be shifted between different paths during transmission in order to implement the calculated traffic rate assigned to each path. Hence, packet out-of-sequence arrival may occur when a flow is shifted from a longer path to a shorter path. However, the proposed algorithm has the following attractive features. Firstly, it achieves very good performance in reducing burst loss and minimizing congestion in the network. Secondly, it exhibits good routing stability in adapting to traffic variations in the network. Finally, the proposed algorithm only uses a simple measurement mechanism which does not incur much signaling and processing overhead. Through extensive simulations under different traffic scenarios, we show that our proposed algorithm performs well in minimizing congestion in the network as well as exhibits good routing stability.

List of Tables

List of Figures

List of Abbreviations

WDM	Wavelength Division Multiplexing
DWDM	Dense Wavelength Division Multiplexing
IP	Internet Protocol
O-E-O	Optical to Electrical to Optical
WADM	Wavelength Add Drop Multiplexer
OXC	Optical Cross-Connect
OCS	Optical Circuit Switching
OPS	Optical Packet Switching
OBS	Optical Burst Switching
SD	Source and Destination
ATM	Asynchronous Transfer Mode
SONET	Synchronous Optical Network
SDH	Synchronous Digital Hierarchy
RAM	Random Access Memory
FDL	Fiber Delay Line
MPLS	Multiprotocol Label Switching
GMPLS	Generalized Multiprotocol Label Switching
LSP	Label Switched Path
VC	Virtual Circuit

TCP	Transport Control Protocol
JIT	Just In Time
JET	Just Enough Time
LAUC	Latest Available Unscheduled Channel
LAUC-VF	Latest Available Unscheduled Channel with Void Filling
APFRA	Adaptive Proportional Flow Routing Algorithm
EPMR	Equal-Proportion Multi-path Routing
HLMR	Hop-Length Based Multi-path Routing
FAR	Flow Arrival Rate
MATE	Multi-path Adaptive Traffic Engineering
RTT	Round Trip Time
GPMR	Gradient Projection Based Multi-path Routing
AARA	Adaptive Alternate Routing Algorithm

Chapter 1
Introduction

1.1 Background in Optical Networking

The Internet has grown exponentially in usage during recent years. As the World Wide Web and corporate intranets continue to grow, applications that require large bandwidth such as voice over IP and video-on-demand are emerging. There is thus an urgent need for new technologies to increase the bandwidth or the data carrying capacity of the network. The industry believes that optical network is a key solution to keep up with the growing bandwidth demands of the Internet. As a result, massive interest has been focused on optical networking in recent years.

Wavelength Division Multiplexing (WDM) [1] has emerged as a core transmission technology for the next-generation Internet backbone networks to cater for the massive bandwidth requirement. WDM gets rid of the electronic bottleneck by dividing the optical transmission spectrum into a number of non-overlapping wavelength channels, each operating at the rate of several gigabits per second [2, 3].

The early deployment of the WDM technology lies in the point-to-point WDM links in optical network architectures. Such networks are comprised of several point-to-point links at which traffic arriving at a node needs to undergo opto-electronic-opto (O-E-O) conversion for every wavelength. The traffic will be dropped, converted from optical signal to electronic signal, processed electronically, and then converted back to optical signal before exiting from the node. The processing at every node in the network will incur significant overhead in terms of switch complexity, large buffer and high electronic processing capacity. It will also slow down the transmission of the traffic since electronic processing is done at a much slower speed than the optical transmission rate.

In order to reduce the overheads and network cost, wavelength add-drop multiplexers (WADM) come into picture for the second generation optical network architecture [4], where traffic can be added and dropped at WADM locations. WADMs allow selected wavelength channels on a fiber to be terminated, while other wavelengths pass through untouched, and they are primarily used to build optical WDM networks, which are expected to be deployed in metropolitan area markets [5].

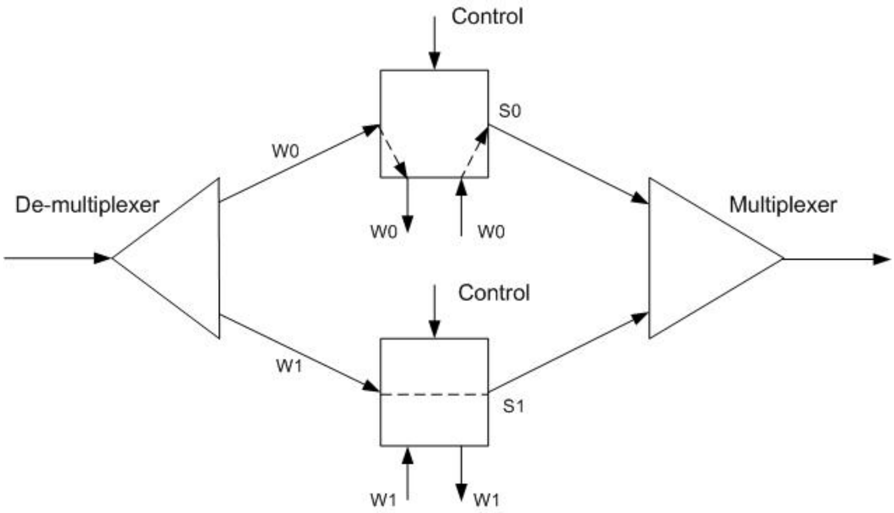

Figure1.1: Optical add/drop multiplexer

The structure of a 2-wavelength WADM is shown in Figure 1.1 and it can be realized using a de-multiplexer, 2×2 switches - one switch per wavelength, and a multiplexer. If the 2×2 switch (S1 in the figure) is in “bar” state, then the signal on the corresponding wavelength passes through the WADM. If the switch (S0 in the figure) is in “cross” state, then the signal on the corresponding wavelength will be “dropped” locally, and another signal can be “added” on the same wavelength at the WADM location.

Third-generation optical network architecture is based on all-optical interconnection devices to build mesh networks that consist of multi-wavelength fiber links. An example of such devices is the Optical Cross Connect (OXC) [6]. It can selectively add and drop wavelengths and also optically switch signals from any input fiber to any output fiber. The OXC can also be equipped with wavelength conversion capability so that it can be configured to change the interconnection pattern of incoming and outgoing wavelengths. The following figure 1.2 shows a 2×2 2-wavelength OXC which can be implemented by demultiplexers, optical switches, and multiplexers. Hence, in third-generation optical networks, data is allowed to bypass intermediate nodes without undergoing O-E-O conversion, thereby reducing the cost and overheads associated with providing high-capacity electronic switching and routing capability at each node. It is possible that data can be switched entirely in the optical domain during transmission between any node pair.

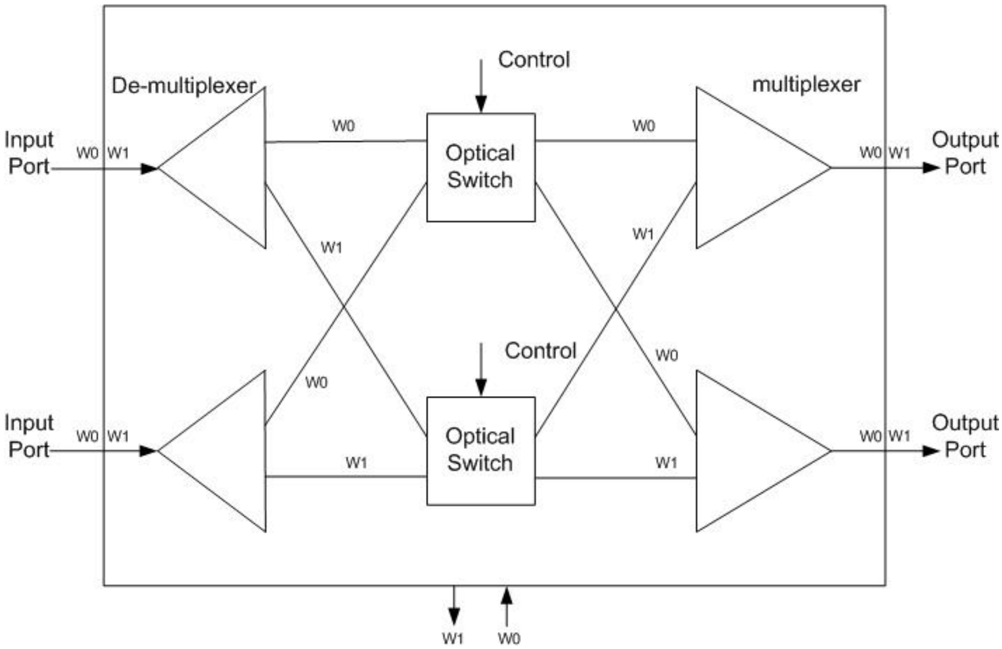

Figure1.2: Optical cross-connect

To date, there are three main all-optical switching technologies proposed for the optical transport networks. They are wavelength routing (or the optical circuit switching, OCS), optical packet switching (OPS) and optical burst switching (OBS) technologies. They are described in detail below.

Wavelength routing [7] or OCS is built on the concept of circuit switching technology and it has been widely studied in the literature. In this approach, lightpaths are set up between two nodes serving as optical circuits to provide connection-oriented transmission to the higher layer protocols such as IP, asynchronous transfer mode (ATM) and synchronous optical networks/synchronous digital hierarchy (SONET/SDH). A lightpath is an all-optical communication channel between two nodes without any O-E-O conversion involved along the way. If wavelength converters are available in the network, the lightpah can use different wavelengths at different links along the route. Otherwise, the same wavelength must be used on all links along the route and this property is known as the wavelength continuity constraint.

The wavelength routing approach is mature and has achieved a great improvement over the early point to point optical network architecture; however, it has some drawbacks. First, the circuit switched connections in OCS are fairly static, they may not be able to accommodate the bursty nature of Internet traffic in an efficient manner. Furthermore, WDM networks based on OCS technologies use lightpaths as the optical circuits, and being a circuit, a lightpath does not allow statistical multiplexing among different connections, which will result in inefficient utilization of network resources.

Optical Packet Switching (OPS) [8, 9] is a new optical switching paradigm in which the basic switching entity is a packet. Packets are switched and routed independently through the network entirely in the optical domain without conversion back to electronics at each node. The header and payload of a packet are sent out together, and upon reaching a node the header will be extracted and processed electronically. The payload is optically delayed by using a fiber delay line (FDL) and then optically switched from the input port to the selected output port. A connection between the input port and the output port is set up for the transmission of that optical packet and will be released immediately afterwards.

OPS allows a great degree of statistical multiplexing of packets onto WDM wavelength channels and this results in improved utilization of the resources in the network. It will also make OPS more suitable for handling bursty traffic than OCS. However, OPS faces some challenges which involve some optical technologies which are still immature or expensive at the current stage. One of the important challenges lies in the lack of optical Random Access Memories (RAMs) for buffering and processing. The optical buffer employed at the current stage is the simple fiber delay lines (FDLs), which are not fully functional as the RAMs in the electronic domain. Some other challenges involve the need of packet synchronization, extraction of headers of optical packets and fast optical switching, whose technologies are still at an immature stage [5].

Optical Burst Switching (OBS) [10, 11, 12, 13] is a recently proposed switching paradigm in optical networks which appears as a promising alternative to OCS and OPS. In OBS, the basic switching entity is a burst which can be thought of as a large container of a number of IP packets with common source and destination nodes. OBS employs a one-way reservation scheme whereby a control packet is sent ahead of the data burst to reserve the wavelength channels and configure the switches along the path. The corresponding data burst will be sent out after a certain period of time without waiting for the acknowledgement for the connection establishment. If some of the switches along the path cannot accommodate the burst due to lack of channel resources, the burst will be simply dropped.

OBS is designed to avoid the challenges faced by OPS while keeping the advantage of statistical multiplexing of network resources. It removes the needs for optical buffering, optical synchronization and optical header extraction technologies which are essential in OPS. At the same time, OBS ensures efficient resource utilization on fiber links as in OPS by reserving bandwidth on a link only when there are data needed to be transmitted along the links. The following table 1.1 from [5] summarizes the features of all the three switching technologies in WDM networks. We can see that OBS has combined the advantages of both OPS and OCS, while avoiding their constraints and shortcomings.

Switching Technology	Bandwidth Utilization	Setup Latency	Switching Speed Requirement	Processing /Synchronization Requirement	Traffic Adaptivity
OCS	Low	High	Slow	Low	Low
OPS	High	Low	Fast	High	High
OBS	High	Low	Medium	Low	High

Table 1.1: Comparison of the different all-optical switching technologies

1.2 Overview of Optical Burst Switching Technology

OBS is designed to achieve a balance between OCS and OPS. A block diagram of a typical OBS network is shown in Figure 1.3. It consists of a meshed network of core nodes and edge nodes interconnected by WDM links. Depending on whether an edge node is a source or destination for traffic transmission, it may be called ingress or egress node, respectively. In an OBS network, IP packets are assembled into data bursts at the network ingress nodes and dissembled back into IP packets at the network egress nodes. Data bursts are switched through the network all optically in dedicated data wavelength channels. A control packet is transmitted along the separate control channel ahead of the data burst in order to reserve the channel and configure the switches along the burst's route. The data burst will only be sent out after a period of time which is called the offset time. The offset time is set to be at least equal to the sum of the header processing time at all intermediate nodes to ensure sufficient time for header processing and the switch to be set up before the burst arrives at the intermediate node. The physical separation of transmission and switching of data bursts and their headers helps to facilitate the electronic processing of headers at optical core routers and provide end-to-end transparent optical paths for transporting data bursts [13].

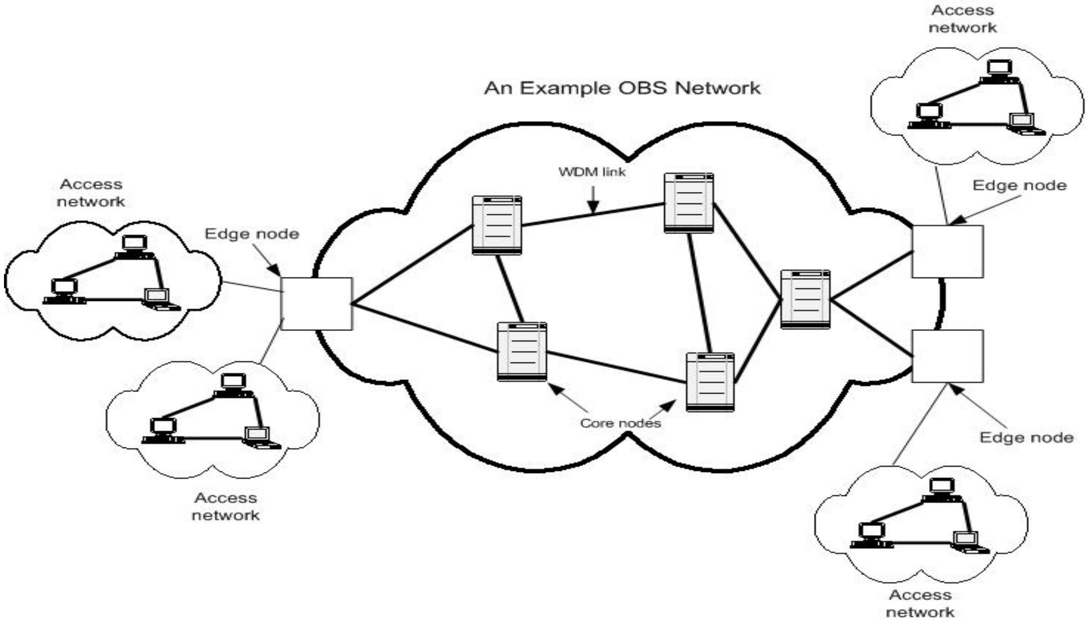

Figure1.3: OBS network architecture

In the literature, it is usually assumed that the core nodes in an OBS network are equipped with full wavelength conversion capability [10, 13] which means they can convert the arriving bursts from any input wavelength to any output wavelength. Furthermore, depending on the choice of the switch architecture, the core nodes may be equipped with optical buffering capacity, which is in the form of fiber delay lines (FDLs). FDLs can only provide deterministic delay and cannot be considered as the full functional optical memory.

In an OBS network, a signaling scheme is required for reserving resources and configuring switches for an arriving burst. Several signaling and reservation protocols have been proposed in the literature. The Just-In-Time(JIT) scheme has been proposed in [14, 15] in which an output wavelength is reserved as soon as the control packet arrives at a node and will only be released after the transmission of the data burst completes and a release message is received. A similar scheme proposed in [12] works in the same way except that the burst length information is carried in the control packet to enable automatic release of wavelength after burst transmission instead of waiting for the release message. These two schemes are simple for implementation, but they cannot make use of the channel resources during the time between the arrivals of the control packet and its associated data burst, which may incur inefficiency in network resource utilization. Another scheme called Just-Enough-Time (JET) was proposed in [10, 13]. In JET, an output wavelength is reserved for a data burst for a fixed duration whose length is specified by the burst length information carried in the control packet. The offset time information is also carried in the control packet. When the control packet reaches a node, it will reserve a wavelength channel on the output link for the duration of the data burst starting from the arrival time of the data burst. The offset time is chosen properly to ensure that when the data burst reaches the node, the resource reservation and switch configuration have been made and the wavelength channel on the output link is available for use. Hence, under JET, there is no bandwidth wastage for the period between the arrivals of the control packet and its corresponding data burst. This will lead to much better bandwidth utilization and a significant performance improvement in OBS networks.

Wavelength channel scheduling is another important issue that has been widely studied in the OBS literature. When a control packet arrives at a node, a wavelength channel scheduling algorithm is needed to decide which wavelength channel on the output link will be allocated to the corresponding data burst. The arrival time and the duration of the data burst can be extracted from its control packet and based on this the scheduling algorithm will select one of the idle channels on the output link to transmit the burst. If FDLs are available at the node, the scheduling algorithm will select one or more FDLs to delay the data burst until the busy wavelength channels become available. Some scheduling algorithms have been proposed in the literature to achieve high bandwidth utilization as well as low burst loss probability, such as First Fit Unscheduled Channel (FFUC) [12], Latest Available Unscheduled Channel (LAUC) [16], and Latest Available Unscheduled Channel with Void Filling (LAUC-VF) [17].

Due to its unique features such as no/limited optical buffering at the core nodes and its one-way reservation scheme, burst delay in OBS network is predictable. Queuing and assembly delay is primarily restricted to the edge nodes in OBS network. Burst delay is predominantly determined by the propagation delay, which is fixed for a specific path. Hence, unlike traditional IP networks where delay is an important performance metric for study and research, delay is not as appropriate a performance metric in OBS networks. Instead, burst loss is considered as the main performance metric of interest in OBS networks.

The primary cause of burst loss in OBS networks lies in the wavelength channel contention. Contention occurs when the total number of overlapping burst reservations at the output link of a core node exceeds the number of available wavelength channels. Contention is aggravated when the traffic becomes bursty and the data burst duration varies and becomes longer. Contention resolution is an important issue in OBS networks and has been extensively studied in literature. Some approaches such as wavelength channel scheduling, deflection routing and load balancing have been proposed to reduce the burst loss due to contention in OBS networks. We will give a detailed introduction in Chapter 2 of the thesis.

1.3 Contributions

In this thesis, we consider the problem of adaptive traffic distribution in OBS networks. In the first part, we introduce a scheme designed for OBS networks which is called adaptive proportional flow routing algorithm (APFRA). The objective of the proposed algorithm is to reduce burst loss and minimize congestion in the network, at the same time avoid the packet reordering at the destination node which is a common problem in previous proposed load balancing algorithms in OBS networks. In our proposed algorithm, it is assumed that multiple link-disjoint shortest paths have been set up between each source and destination (SD) pair. A set of flow proportions will be assigned to the paths between each SD pair. A path is selected to route the new incoming flows with a prescribed frequency determined by its assigned flow proportion within each time window. Once the path assignment for a flow is made, all the packets belonging to the flow will be transmitted using the same path until the flow exits the network. At the end of each time window, based on the measured “quality” and hop length factors of the paths between each SD pair, the set of assigned flow proportions for the paths will be adjusted accordingly and applied to route new incoming flows in the next time window. However, the existing flows being transmitted are not affected by the traffic proportion adjustment and they will not be shifted from one path to another. Hence, the proposed algorithm retains the entirety of traffic flows and waives the need for packet re-ordering at the destination node. Packet reordering is known to have an adverse effect on the application level performance for some services [18]. Since over 90% of the current Internet traffic is TCP traffic [19], care must be taken to maintain the integrity of the TCP flow status when we exercise traffic engineering. As mentioned earlier in the chapter, the bursts in OBS networks are assembled from the packets in TCP/IP flows that are aggregated into the OBS-based backbone network. Hence, if the flow transmission is disrupted, packets from the same TCP/IP flow might reach the destination in a highly disordered manner. This is undesirable for TCP applications as this not only causes excessive reordering burden but also renders a wrong impression to TCP that congestion occurs. TCP will consequently decrease the size of the congestion window of the TCP/IP flows, which leads to degradation in performance. Due to its flow-based nature, our proposed algorithm is effectively packet re-ordering free. Furthermore, the routing and adjustment at the flow level and in a proportional manner will also help to improve the routing stability and reduce traffic fluctuation in the network [22, 43]. Through extensive simulations, we investigate the performance of our proposed adaptive flow routing algorithm under different traffic conditions. The results show that our approach behaves well in practice and achieves a significant performance improvement over the static alternate flow routing algorithms such as the equal-proportion and hop-length based flow routing.

In the second part of the thesis, we deal with the problem of adaptively and efficiently mapping the offered burst traffic into multiple end-to-end paths between each SD pair in OBS networks based on the theoretical optimization framework. Burst contention is a major problem in OBS networks since it directly influences the burst loss performance. Some works employing load balancing and multi-path adaptive routing techniques have been proposed in the literature to reduce burst contention in OBS networks [20, 21, 22]. Their main ideas are to balance a certain amount of traffic from the heavily-loaded paths to the lightly-loaded paths. However, their ways to determine the amount of traffic that needs to be adjusted are only based on some simple heuristic algorithms which are based on some link load or link congestion status information. Although they are working well in reducing the burst loss in the network compared with the simple shortest path routing, we can achieve further performance improvement considerably if we can determine the amount of traffic for adjustment based on some well-known network optimization frameworks. Furthermore, there is no guarantee that the load balancing schemes proposed in [20, 21, 22] can converge to a stable routing state. They may suffer from the routing instability problems, such as traffic fluctuations and route oscillations which are common in link-state based load balancing algorithms. To overcome the above mentioned shortcomings, we propose a new multi-path traffic routing scheme in OBS networks which is based on the gradient projection optimization algorithm [23] to determine the traffic splitting or mapping among the multiple paths between each SD pair. In this scheme, traffic flows may be shifted between different paths during transmission in order to implement the calculated traffic rate mapped to each path. Hence, the packet out-of-sequence arrival problem may occur when a flow is shifted from a longer path to a shorter path. However, the proposed gradient projection based multi-path traffic routing algorithm has the following attractive features. Firstly, it achieves very good performance in reducing burst loss and minimizing network congestion. Secondly, it exhibits good routing stability in adapting to traffic variations in the network. Finally, the proposed algorithm only uses a simple measurement mechanism which does not incur much signaling and processing overhead. Through extensive simulations under different traffic scenarios, we show that our proposed algorithm performs well in minimizing congestion in the network as well as exhibits good routing stability.

1.4 Thesis Organization

The thesis is organized as follows:

Chapter 1 gives the overview of optical networking technology as well as the background in the optical burst switching. Also, we give a brief summary of our contributions in this thesis.

Chapter 2 provides a survey of the current literature on contention resolution in OBS networks. Works related to multi-path traffic routing and load balancing in MPLS-based IP networks as well as in OBS networks are also presented.

Chapter 3 describes the proposed adaptive proportional flow routing algorithm in OBS networks. Simulations results are presented and discussed.

Chapter 4 presents the proposed gradient projection based multi-path optimal traffic routing algorithm. The details of the algorithm are illustrated, followed by a discussion on the simulations results.

Chapter 5 – summarizes the thesis with some concluding remarks.

Chapter 2

Related Work

In this chapter, we will describe the early works related to contention resolution and avoidance policies in OBS networks. We will also touch on the related works on multi-path traffic routing and load balancing in MPLS-based IP networks as well as in OBS networks.

2.1 Contention Problem in OBS Networks

A major concern in OBS networks is high contention and burst loss due to output wavelength channel contention. Contention and burst loss can be reduced by having efficient data channel scheduling algorithms at the core nodes, as well as implementing contention resolution and avoidance policies in the network.

2.1.1 Wavelength Channel Scheduling Algorithms

In [12, 13], several wavelength channel scheduling algorithms have been proposed to schedule bursts efficiently while achieving high resource utilization at the same time. Latest Available Unscheduled Channel (LAUC), and Latest Available Unscheduled Channel with Void Filling (LAUC-VF) are among the most popular algorithms. LAUC maintains the unscheduled time for each wavelength channel and tries to schedule the arriving burst using the unused channel that becomes available at the latest time. When void filling (VF) is allowed, gaps/voids between two scheduled data bursts are recorded and they can also be utilized to transmit bursts. LAUC-VF is to schedule each arriving data burst using the latest available unused data channel to minimize the starting time of the void and the arrival time of the data burst. In both of the above algorithms, JET is employed as the resource reservation scheme.

2.1.2 Contention Resolution Policies in OBS Networks

One of the primary design issues in OBS is to minimize the burst loss in the network. Burst loss occurs primarily due to the contention of bursts in the bufferless core nodes. In the literature, some approaches have been extensively studied to resolve the burst contention problem in OBS, such as wavelength conversion, optical buffering and space deflection. In wavelength conversion, if multiple bursts contend for the same wavelength at the same time, the bursts will be shifted to another wavelength on the same link using wavelength converters [24]. In optical buffering, fiber delay lines (FDLs) are used to provide the necessary delay for data bursts in order to resolve the contentions [25]. However, FDLs are expensive in cost and large volume of FDLs is needed if we want to provide enough optical buffers at each core node to resolve burst contentions. In the space deflection approach, deflection routing is employed to deflect the burst to an alternate port or channel if the primary port or channel is occupied [26, 27, 28]. However, deflection routing may cause some side effects such as burst looping and burst out-of-order arrival at the destination.

When there is no unscheduled channel, the contention cannot be resolved by any one of the above techniques and some of the bursts must be dropped. The policy for selecting which bursts to drop is referred to as the soft contention resolution policy and is used to reduce the overall burst loss rate [29]. Some soft contention resolution algorithms have been proposed in earlier literature, including the burst segmentation [30] and look-ahead window contention resolution [31]. In burst segmentation, in case of contention, instead of dropping the entire burst, only the overlapping segments are dropped. It is useful for those applications which have stringent delay requirements but relaxed packet loss requirements. In [31], a look-ahead window with a size of *W* time units is constructed which consists of multiple control packets arrivals. The decision on which incoming data bursts should be reserved or discarded is based on the collective view of multiple control packets. At each hop, the control packets must be stored for a duration of *W* time units before they are retransmitted and FDLs are used on each hop to delay the data bursts by *W* time units to maintain the original offset time. Although this algorithm can achieve improved performance for burst dropping, it introduces high end-to-end delay for bursts and has high requirement for FDLs at core nodes.

2.1.3 Contention Avoidance Policies in OBS Networks

The above contention resolution policies are considered as reactive approaches in the sense they are only triggered after contention occurs. Another way to reduce contention in the network is by proactively attempting to avoid network overload through some traffic management policies at the system level. Consequently, contention avoidance policies attempt to prevent a network from entering the congestion state in which considerable burst loss occurs.

In general, contention avoidance policies can be implemented in either non-feedback-based or feedback-based manner. In a non-feedback-based approach, the ingress nodes do not have knowledge of the network states and they cannot react to the network load changes accordingly. Without requiring any additional information from the control plane, each node regulates its own offered traffic load into the network through traffic shaping and regulation. One way to implement the traffic shaping is through the burst assembly techniques such as the schemes proposed in [32, 33, 34]. In [35], the authors proposed the regulation of burst traffic by combining the periodic traffic reshaping at the edge node and a proactive reservation scheme. The main challenge in implementing the contention avoidance policies in non-feedback-based OBS networks lies in the definition of traffic parameters, such as peak and average traffic rate at each edge node, in order to avoid or minimize link congestion.

In a feedback-based manner, congestion reduction is achieved by adaptively adjusting the offered traffic load at the source to match the latest status of the network and its available resources. Hence, as the available resources in the network change, the source should vary its offered load or burst rate to in adaptation to the network situation accordingly. The main design issues in feedback-based manner lie in defining the feedback mechanism, the parameters needed to be measured, as well as how the designed schemes interpret the feedback information and react to the current network state [36].

In [29], the authors have proposed a contention avoidance policy designed for the feed-back based OBS networks where explicit feedback signaling is sent to each source indicating the required reduction in the burst flow rate going to congested links. Hence, the edge node attempts to avoid or minimize contention by adjusting its data burst flow rate to the required level through admission control. In [37], the authors proposed a contention avoidance policy by implementing the TCP-like congestion avoidance mechanism to regulate the load offered to an OBS switch. In the approach, a TCP decoupling virtual circuit (VC) is set up for each pair of source and destination nodes. The VC uses TCP congestion control to control the burst sending rate of its source node. Under TCP congestion control, the total sending rates of contending source nodes will not exceed the bandwidth of the bottleneck links too much. This can effectively control the load offered to an OBS switch and avoid high burst/packet drop rate while keeping the link utilization high.

Some other proposed contention avoidance schemes are based on load-balancing and traffic re-routing between alternative paths such as the ones presented in [20, 21, 22] and they will be given more illustrations in the subsequent section of this chapter.

2.2 Load Balancing and Multi-path Traffic Routing in IP/MPLS Networks

The problem of multi-path traffic routing and load balancing has been extensively studied in IP networks. In [23], the multi-path optimal traffic routing has been generalized as a constraint optimization problem. Analytical models have been built and a set of classical optimization algorithms such as Frank-Wolfe and Gradient Projection algorithms have been used by the authors to solve the problem. The proposed solution works under the assumption that the traffic demand for each source and destination pair is known beforehand and it is an offline optimization problem.

In [38], an online multi-path adaptive traffic engineering algorithm, called MATE, is proposed for switched networks such as MPLS networks. The objective of MATE is to reduce network congestion by adaptively balancing the traffic load among multiple paths between each SD pair based on the measurement and analysis of the path congestion metric. MATE uses a state-dependent mechanism which adjusts the traffic assignment based on the current state of the network, which can be reflected as some performance metrics like link utilization, packet delay and packet loss etc.

In MATE, it is assumed that several explicit LSPs (Label Switched Path) have been pre-established between the ingress and egress nodes in an MPLS-enabled domain. The role of the ingress node is to distribute the traffic across the LSPs so that the loads are balanced and the congestion in the network is minimized. Traffic routing in MATE has also been modeled as a constraint optimization problem, and the authors adopt the gradient projection algorithm to solve the problem. Since it is an online optimization problem, the new traffic rates calculated by the SD pairs may only be reflected in the link flows after certain delays and SD pairs may update their rates asynchronously and in an uncoordinated manner. Hence, the authors propose the first derivative length of a path to be estimated empirically by averaging several past measurements over a period of time in the update algorithm.

In [39], the authors propose a similar distributed optimal routing algorithm based on stochastic approximation theory, using local network state information. The paper proposes a different measurement-based algorithm which is derived from the idea of simultaneous perturbation stochastic approximation [40, 41] to estimate the required first derivative length of each path. The paper claims that the proposed approach can greatly reduce the number of measurements required to estimate the first derivative lengths at the same time the approximately the same level of accuracy can be retained at each iteration. By reducing the number of measurements, a better overall convergence rate will be achieved due to the fact that a non-negligible amount of time is required for each measurement in a realistic networking environment.

2.3 Load Balancing and Multi-path Traffic Routing in OBS Networks

Although multi-path traffic routing and load balancing have been extensively studied in traditional MPLS-based IP networks, little attention has been paid to the case of OBS networks. Due to the unique features of OBS, such as no electronic buffering and no/limited optical buffering at the core nodes, we can have different or better ways to solve this online multi-path traffic engineering problem. As has been mentioned before, delay in OBS networks is predictable and is predominantly determined by the propagation delay, hence delay in OBS networks is not as appropriate a performance metric as in MPLS-based IP networks to implement the load balancing and multi-path traffic routing schemes. Instead, in literature, the burst loss probability is the most widely used performance metric since the link burst loss probability is directly related to the traffic load offered to the link in OBS networks.

To date, some schemes have been proposed to tackle the above-mentioned problem in OBS networks. In [20], the author has proposed a dynamic congestion-based load balanced burst routing scheme. The scheme statically computes link-disjoint alternate paths between each SD pair and dynamically selects one of the paths based on the collected path congestion information to route the incoming bursts. In the scheme, whenever the offered load on a link exceeds a maximum threshold value, it will signal a congestion status. Then once the congestion status of all the links at a core node has been determined, this information is sent to all edge nodes in the network. Based on this information, ingress nodes calculate the load status of the paths. Then whenever there is a burst ready to be sent, the edge node sends the burst through the primary or alternate path whichever is the least congested in terms of the load status. In [21], a similar idea has been presented to transmit bursts along the least congested path between each SD pair. A suite of path selection strategies, each utilizing a different type of information regarding the link congestion status has been presented. In the paper, the authors also present the idea of hybrid path selection strategies, which makes routing decisions based on a weighted combination of the decisions taken by several independent path selection strategies. In [22], a similar approach has been proposed where the authors consider balancing the traffic load by shifting the traffic flows between primary and alternative paths periodically. For each time window, the ingress node will send out probe packets to get the burst loss information along the primary and alternative paths between the SD pair. Then based on this burst loss as well as hop length information, traffic flows will be shifted between the paths in order to balance traffic load and reduce congestion in the network.

For the above proposed schemes, the rerouting of traffic between different paths will introduce considerable packet out-of-sequence arrival problems at the destination since data traffic in OBS networks are assembled from different IP flows. It incurs high buffering capacity and processing power at the destination side to do the packet re-ordering. It also has an adverse effect on the application level performance for some services. We can have better ways to do the traffic routing and avoid the above mentioned problem which will be discussed in Chapter 3 of the thesis. Furthermore, as has been mentioned in Chapter 1, in the above schemes, the way to determine the amount of traffic that needs to be adjusted is only based on some simple heuristic algorithms. Although they out-perform the simple shortest path routing, we can achieve further performance improvement and better routing stability if we can determine the amount of traffic for adjustment based on some well-known network optimization models and theories. We will work on this issue in Chapter 4 of the thesis.

Chapter 3

Adaptive Proportional Flow Routing in IP-over-WDM OBS Networks

3.1 Introduction

In this chapter, we introduce a scheme designed for the feedback-based OBS networks which is called adaptive proportional flow routing algorithm (APFRA). The algorithm works in a distributed manner, which means the algorithm is run for each individual node pair independent of other node pairs. In the algorithm, multiple link-disjoint shortest paths are pre-selected between each SD pair using Dijkstra's algorithm. Each path will be assigned a traffic flow proportion at the beginning of each measurement period and the flow proportions here are obtained on the basis of flow numbers. Within a measurement period, a path is selected to route new incoming flows with a prescribed frequency determined by its assigned flow proportion. Once the assignment for a new flow is made, all packets belonging to the flow will be transmitted using the same path. Probe packets will be sent out from the ingress periodically to measure the burst loss performance along each path between the SD pair. Based on the measured "quality" as well as the hop length factors of the paths, the set of assigned flow proportions for the paths between the SD pair will be altered accordingly and applied to route new incoming flows. However, existing flows on transmission will not be affected by the change and they will not be shifted between different alternative paths. In the mean time, the burst assembly time for each of the path between the SD pair may also be varied based on the measured “quality” of the path to further enhance the performance of the proposed algorithm.

From the above description, we can see that the flow-based nature of the proposed algorithm strictly controls the probability of packet re-ordering and in effect the algorithm can be made effectively packet re-ordering free. The adjustment in the algorithm will not disturb the entirety of the existing traffic flows and waives the need for high processing power and overhead for packet re-ordering at the destination node. It also helps to avoid the adverse effect brought about by packet re-ordering which will cause performance degradation at the higher-layer applications. Furthermore, the proposed algorithm performs traffic routing and adjustment at the flow level and in a proportional manner with a proportion assigned to a path reflecting its quality. Instead of picking just one “best” path to route the traffic as in the case of “best-path” routing schemes like shortest-path or widest-shortest-path routing, in the proposed algorithm a better path is favored by assigning a larger flow proportion to it and a worse path is assigned a smaller flow proportion. In this manner, it helps to improve the routing stability and reduce traffic fluctuation in the network [43].

By simulation, we investigate the burst loss performance of our proposed adaptive proportional flow routing algorithm under different traffic conditions. We compare our results with the static alternate flow routing algorithms such as equal-proportion and hop-length based flow routing. We show that our approach works well and achieves significant improvement in terms of burst loss performance.

The rest of the chapter is organized as follows. In section 3.2, the overall picture of the proposed flow routing algorithm is presented. The implementation details of the algorithm are presented in section 3.3. The operation of each functioning unit will be described. In section 3.4, the performance study of the proposed algorithm is presented. Finally, a summary of the results is presented in section 3.5.

3.2 An Overview of the Proposed Adaptive Proportional Flow Routing Algorithm

In this section, we will briefly describe the basic operation of the proposed adaptive proportional flow routing algorithm. For each source and destination pair, a set of multiple link-disjoint paths are used to transmit data bursts and control packets. Label switched paths (LSP) could be established to facilitate the transmission of control packets with reduced signaling and processing overhead. For a given source and destination pair, individual traffic flows arriving at the ingress node are identified and adaptively assigned to one of the paths between the node pair based on the set of computed flow proportions. This set of flow proportions are computed based on the measured burst loss performance of various paths in the previous time windows. Once a flow has been assigned to a specific path, all packets of the same flow will be forwarded to the same path. The flow will not be disturbed or shifted from one path to the other in the midway. The goal of the algorithm is to reduce the network congestion and achieve improved overall network performance.

A time-window-based measurement and feedback mechanism has been adopted in the algorithm. The set of flow proportions are periodically evaluated in each measurement time window. If congestion occurs in some of the paths between a node pair, the node pair’s set of flow proportions will be adjusted based on the traffic statistics measured in the previous time windows. If none of the paths between the node pair becomes congested, its set of flow proportions will remain the same as the previous time window in order to minimize the unnecessary traffic adjustment.

The time-window-based mechanism is based on the assumption that traffic condition remains relatively stable. It is reasonable and feasible here due to the following reasons. Firstly, the number of traffic flows on a link changes relatively slowly in the scale of few minutes [38]. Hence, traffic situation in a future time window is predictable based on the traffic statistics measured in previous time windows. Secondly, recent studies show that IP traffic often exhibits long-range dependence, with the implication that congestion period may be long and predictable [42]. Since bursts in OBS network are assembled from IP flows, we expect the congestion situation in the network to be predictable.

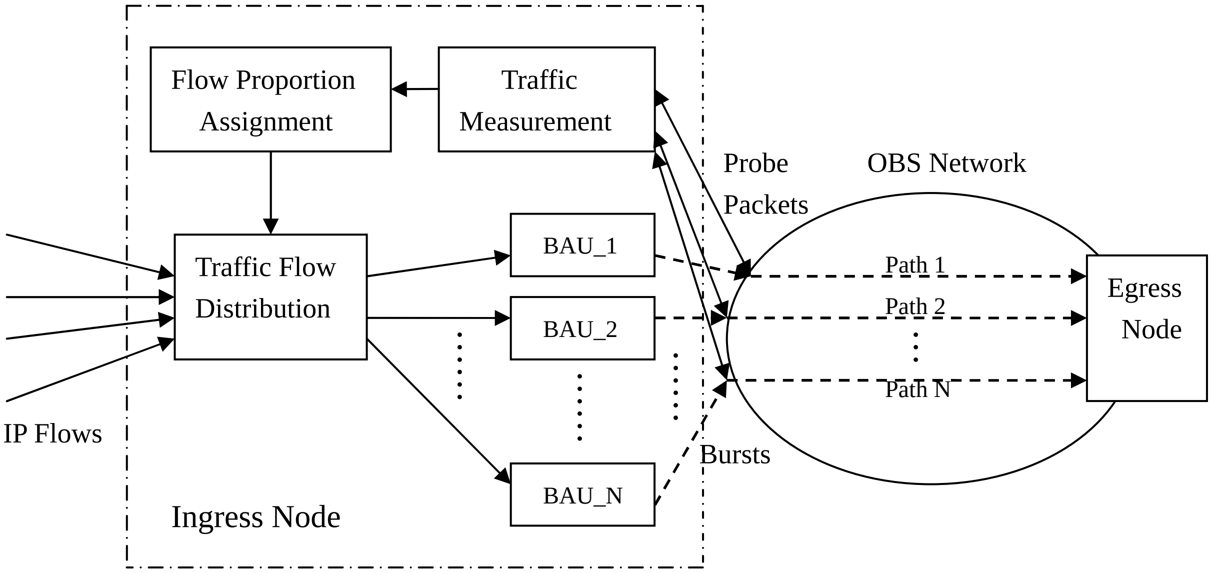

Figure3.1: Functional units of the proposed flow routing algorithm

Figure 3.1 shows the basic functional blocks of the architecture for the proposed flow routing algorithm for a specific node pair. At the ingress node, four functional units - traffic measurement, flow proportion assignment, traffic flow distribution and burst assembly units (denoted by BAU in the figure) work together to perform the multi-path proportional flow routing algorithm. Traffic measurement is in charge of measuring the traffic statistics along each path between the node pair by sending probe packets periodically. The collected information will be used to evaluate the burst loss performance of the paths under the previous flow proportion assignment. Based on the burst loss performance and the hop length of each path, the flow proportion assignment unit determines the new flow proportion that will be assigned to each path at the next measurement period. The set of new flow proportions will be used to make the routing decisions for new incoming traffic flows. The traffic flow distribution unit plays the role of distributing the IP flows that arrive at the ingress node to the multiple paths between the node pair according to the decision made by the flow proportion assignment unit. Since the traffic assignment here is flow-based, the traffic flow distribution unit has to maintain the mapping information between the flows and the path. Finally, each path will have its own burst assembly unit to assemble the packets from the traffic flows which have been assigned to it for transmission.

3.3 Adaptive Proportional Flow Routing Algorithm

In this section, the details of the proposed multi-path adaptive proportional flow routing algorithm (APFRA) will be presented. The algorithm can be dissected into several functional units which have been introduced in the previous section, i.e. traffic measurement, flow proportion assignment, traffic flow distribution and burst assembly units. As mentioned in the previous section, each node pair performs their own flow routing independent of other node pairs. Hence, without loss of generality, we describe the working details of APFRA for a specific node pair *s*. *N* link-disjoint shortest paths are pre-selected between the node pair.

3.3.1 Notations

For the ease of exposition, we define the following notations,

$r_1, r_2, ...r_N$: multiple link-disjoint paths between a node pair

$length_1, length_2, ...length_N$: hop counts of the paths

$T(i)$: i th measurement window

$loss_k(i)$: mean burst loss probability on the k th path in time window *T(i)*

P_k^i : flow proportion assigned to the k th path in time window *T(i)*

$\{P_1^i, P_2^i, ...P_N^i\}$: the set of flow proportions assigned to the paths between the node pair in time window *T(i)* which is used to determine the path selection sequence to route new incoming traffic flows

Note that $length_1 \le length_2, ... \le length_N$ and $P_1^i + P_2^i + ... + P_N^i = 1$

3.3.2 Traffic Measurement

In the algorithm, the traffic measurement is carried out on a per-path basis. The purposed is to collect traffic statistics for each path by sending probe packets periodically and calculate the mean burst loss probability to evaluate the impact of assigned flow proportions. The measurement mechanism in one specific time window *T(i)* is illustrated as follows.

At each node, a counter is used to record the number of bursts dropped since the last probe was made. At the beginning of *T(i)*, the ingress node starts to record the total number of bursts sent to the different paths between the node pair, $total\_burst_s(r_1), total\_burst_s(r_2), ..., total\_burst_s(r_N)$, respectively. At the end of each T(i), the ingress node will send out the probe packets along each of the path separately to collect the recorded number of lost bursts at each intermediate node. When the probe packet traverses all the way down to the egress node and then comes back to the ingress node, the total number of dropped bursts along the path r_k can be obtained as $dropped\_burst_s(r_k)$. It is the sum of the number of dropped bursts at each link along the path. Then the mean burst loss probability of each path in time window $T(i)$ can be calculated as follows:

$$loss_k(i) = \frac{dropped\_burst_s(r_K)}{total\_burst_s(r_K)} \tag{1}$$

With the support of GMPLS, LSPs could be setup for each path. This can help to reduce the complexity and overhead incurred in the operation. One important point to take note here is that the size of the measurement time window *T(i)* should be set sufficiently larger than the longest propagation round trip time (RTT) in the network. This is to reduce the impact of the probe packet propagation delay on the accuracy of measurement and hence the performance of the algorithm.

3.3.3 Flow Proportion Assignment

Flow proportion assignment adaptively determines the flow proportion allocated to each of the paths between the node pair in each measurement time window. The flow proportion here is calculated based on the number of flows that have been assigned to the paths. The flow proportion assignment is determined by two parameters: the measured mean burst loss probability on the paths and the hop length of the paths. The measured mean burst loss probability in a number of previous time windows is used to estimate the impact of offered flow proportions on the paths as well as predict the future traffic condition along the paths. In order to defend our prediction from the bursty nature of the network and reflect the “quality” of a path more objectively and completely, we will integrate several history burst loss probability values of the paths into the metric for decisions. Instead of using the mean burst loss probability in the latest measurement time window only, we use the weighted sum of the mean burst loss probabilities of a path in the past few time windows in the expression for determining its new flow proportion. Let the size of the sampling size be *W*, which means we will use the mean burst loss probability of a path in the past *W* time windows to form the combined weighted sum. The sum for a specific path r_k at the end of the time window $T(i)$ can be expressed as follows:

$$S_k = \sum_{m=0}^{W-1} \alpha_m loss_k(i-m) \text{, for } k = 1,2,...,N, \qquad (2)$$

where $\alpha_0, \alpha_1, ...\alpha_{W-1}$ are the weights assigned to each of the burst loss probabilities in the *W* time windows and $\alpha_0 + \alpha_1 + ... + \alpha_{W-1} = 1$. The values of the $\alpha_i's$ are chosen such that the more recent mean burst loss probability is given a lager weight, whereas the older mean burst loss probability is given a smaller weight.

Another parameter for determining the flow proportions is the hop length of the path. The following reasons may account for the importance of the hop length factors in OBS networks.

1. Since burst scheduling is done at each intermediate node traversed along the path, a longer path means a higher possibility for burst contentions to happen along the way as well as higher burst processing overhead.
2. A longer path consumes more network resources. Congestion that occurs in a longer path will cause more links in the network to be over-loaded. It will bring about a more adverse effect than the congestion that happens in a shorter

path and results in a lower network performance.

Hence, network performance may become poorer when excessive traffic flows are routed through the longer paths even though they are lightly loaded. Therefore, in the algorithm, we will incorporate the hop count factor into the decision metric such that the relatively shorter paths will be made more favorable to route more traffic flows. Specifically, a factor b_k will be given to path r_k such that it is inversely related to the path's hop count and $\sum_{i=0}^{N} b_k = 1$. One possible way of assignment for this factor can be implemented as follows,

$$b_k = \frac{\frac{1}{length_k}}{\sum_{i=1}^{N} \frac{1}{length_i}} \quad \text{for} \quad k = 1,2,..., N \tag{3}$$

Frequent and excessive adjustment of traffic flow proportions when the traffic conditions in the network are relatively light and stable may bring about some undesirable effects. For example, it will be prone to causing traffic fluctuation and introducing routing instability in the network. In addition, it will also incur higher routing and processing overhead for the algorithm.

In order to avoid this problem, we will set a so-called congestion threshold value for each path between the node pair. When the measured $loss_k(i)$ for path r_k in time window *T(i)* exceeds its threshold value $Threshold_k$, the path is considered as congested. At the end of each time window, the ingress node will check whether any of the paths between the node pair has become congested. If none of them is congested, the ingress node will not trigger the algorithm to make adjustment to the flow proportion assignment. The set of flow proportions to route new incoming flows for the next time window will remain the same as the previous time window. However, if one or more paths between the node pair become congested, the proposed algorithm will be triggered to make appropriate adjustment to the flow proportions.

We illustrate the detailed flow proportion assignment process in a specific time window *T(i)*. Initially, the flow proportion is distributed in the following way,

$$P_k^0 = \frac{\frac{1}{length_k}}{\sum_{i=0}^{N} \frac{1}{length_i}}, \quad \text{for} \quad k = 1,2,...N \tag{4}$$

Let the mean burst loss probability of the *N* paths returned by the traffic measurement window *T(i-1)* be $\{ loss_1(i-1), loss_2(i-1),..., loss_N(i-1) \}$. Let the flow proportion assignment in time window *T(i-1)* be $\{ P_1^{i-1}, P_2^{i-1},.......P_N^{i-1} \}$. If $loss_k(i-1) < Threshold_k$ for all $k = 1,2,...N$, the set of flow proportions for the next time window *T(i)* will remain the same as that of *T(i-1)*, i.e. $\{ P_1^i, P_2^i,.......P_N^i \}$= $\{ P_1^{i-1}, P_2^{i-1},.......P_N^{i-1} \}$. If $loss_k(i-1) \geq Threshold_k$, for any $k = 1,2,...N$, the algorithm will be triggered to make adjustment to the flow proportions as follows.

For the mean burst loss probability parameter, first we associate a proportional factor with each path between the node pair such that its value is inversely related to its S_k, the combined weighted sum of its mean burst loss probability in the past. For a specific path r_k, the factor can be expressed as

$$f_k = \frac{\frac{1}{S_k}}{\sum_{k=1}^{N} S_k} \quad \text{for all} \quad k = 1,2,...,N \tag{5}$$

where the value of S_k is given by Equation (2).

On the other hand, for the hop count parameter, as has been mentioned above, the factor b_k can be expressed as Equation (3) for path r_k. Hence the flow proportion that is assigned to path r_k in the coming time window *T(i)* will be as follows.

$$P_k^i = \frac{b_k * f_k}{\sum_{k=1}^{N} b_k * f_k} \tag{6}$$

The flow proportions assigned to the other paths between the node pair will be calculated in the same way, and take on the same expression as Equation (6) above for all $k = 1,2,......,N$. In the end, the set of new proportions will be obtained as $\{ P_1^i, P_2^i, ...P_N^i \}$, where $P_1^i + P_2^i + ... + P_N^i = 1$. The flow proportion assignment unit will then pass this information to the traffic flow distribution unit for selecting the paths to route the new incoming flows in the coming time window *T(i)*.

3.3.4 Traffic Flow Distribution

The traffic flow distribution unit chooses the path from the path set between the node pair to route the new incoming IP flows based on the flow proportion assignment decision. The traffic distribution here is on a per-flow basis and once a flow has been assigned to a path, all of its packets will follow the same path until the departure of the flow. We will not shift any portions of an existing flow from one path to the other in the midway.

The flow routing is performed in the weighted-round-robin manner. When a new traffic flow arrive at the node pair, a path will be selected to route the flow using the weighted-round-robin-selector [43], which is formed based on the set of given flow proportions. The basic principle works such that if a path r_k 's assigned flow proportion is $1/n$, then in every n times of the path selections, there must be one occurrence of path r_k. For example, we have four paths r_1, r_2, r_3, r_4 between the node pair, and their assigned flow proportions are $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}$, respectively. Then a typical path selection sequence will be $\{ r_1, r_2, r_1, r_{3,} r_1, r_{2,} r_1, r_4 \}$. The frequency of occurrence for a specific path in the path selection sequence is proportional to its assigned flow proportion. Hence, whenever a new traffic flow arrives, based on the path selection sequence, a path will be chosen to route the flow and the path selection sequence will repeat in cycles.

The flow-based routing algorithm will have to maintain the mapping information between the flow and its assigned path. With the support of GMPLS, LSPs can be set up for each path. Hence, only the ingress node needs to maintain this mapping information. In the proposed algorithm, since all the packets of a flow will follow the same path once a flow has been assigned, the ingress only has to maintain one piece of mapping information per entire flow. On the contrary, if shifting of existing flows is allowed, the ingress node will have to maintain the mapping information between different shifted segments of a flow and the paths they followed for each flow. It will inevitably incur higher signaling and processing overhead during the operation. Furthermore, when flows are shifted from a longer path to a shorter path, considerable burst and packet out-of-sequence arrival problems will arise in the network. Hence, it will require high buffering capacity and processing power at the egress node to perform the packet reordering. Significant packet re-ordering will also bring about some adverse effect to some services standing at the higher application level. These drawbacks are inherent in the previous algorithm proposed in [22] which relies on flow shifting to achieve load balancing in the network. In contrast, the proposed flow routing algorithm, avoids the above-mentioned problems by performing the routing on the per-flow basis and maintaining the entirety of a flow during its transmission along a path.

3.3.5 Burst Assembly Unit

After the paths have been selected to route the incoming traffic flows, the packets from the flows will go through the burst assembly units for their assigned paths. They will be assembled into larger data bursts and transmitted through the paths. The burst assembly technique employed here is the time-threshold based burst assembly. In this burst assembly technique, packets from the arriving flows are queued at the burst assembly unit of the path. A clock will be set up to keep the timing of the packet queuing. When the time elapses beyond a certain prescribed threshold value $T_{burst\_assembly}$, the packets accumulated during this period will be packed into the data burst and sent out for transmission.

The reservation scheme used here is JET and the burst scheduling technique employed here is LAUC-VF in which a burst chooses the unused channel that becomes available at the latest time and the gaps between two scheduled data bursts can also be utilized. Under such circumstances, the bursts with shorter length will get a higher chance in finding suitable voids to be scheduled when compared to the burst with longer length. It will be helpful to reduce the length of the bursts that are assigned to a path in order to improve its overall burst loss performance. Hence, in addition to adjusting the flow proportion, we can also adjust the burst assembly time threshold $T_{burst\_assembly}$ for a path based on the path's measured "quality". This can further enhance the performance of the proposed flow routing algorithm which can be seen from the simulation results presented in the subsequent performance study section.

The basic principle of varying the burst assembly time is that if there are N paths between the node pair, each of the path's burst assembly unit will have its own assigned burst assembly time threshold. The burst assembly time thresholds will be chosen between the prescribed upper bound and lower bound values. The upper bound is set to reduce the burst loss along the paths. The lower bound is set to avoid generating unduly high number of bursts which will incur high signaling and processing overhead in the network. Based on the measured burst loss performance of a path, or, specifically, its combined weighted sum of its mean burst loss in the past $S_k = \sum_{m=0}^{W-1} \alpha_m loss_k (i-m)$, a burst assembly time threshold will be assigned to it accordingly in the next time window. In addition to reducing its assigned flow proportion, a path with a worse measured burst loss performance will be assigned a shorter burst assembly time threshold relative to the path with better measured burst loss performance. With reduced flow proportion, the packet arrival rate of the path will also be reduced. Hence, together with its shorter assigned burst assembly time threshold, the bursts transmitted through the path with a worse measured “quality” will have a shorter average length and can be scheduled more easily. This can help to alleviate the burst loss situation along the path. On the contrary, the less congested path with a better measured “quality” will be assigned a longer burst assembly time threshold, and the flows assigned to the path will be assembled into bursts with a longer average length.

3.4 Performance Study

In this section, we will show the performance of our proposed Adaptive Proportional Flow Routing Algorithm (APFRA) through extensive simulations in the optical network shown in Figure 3.2. The network comprises 10 nodes and 20 bi-directional links. A bi-directional link consists of two unidirectional fibers in opposite direction. Each fiber has 4 data channels at 1 Gb/s transmission capacity. The data channel scheduling algorithm employed here is LAUC-VF.

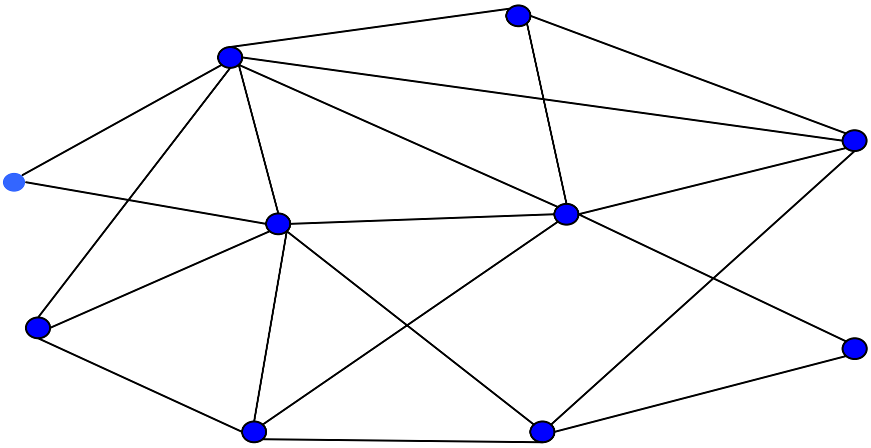

Figure3.2: Simulation network

In the simulations, the program will randomly choose the edge nodes to form source and destination pairs among the 10 nodes. There are altogether 42 source and destination pairs that have been formed.

In order to get more realistic results, the long range dependent traffic model is employed in our study. In this traffic model, traffic that arrives at each node pair in the network is the aggregation of multiple IP flows. Each IP flow is an ON/OFF process with Pareto distributed ON and OFF times. During each ON period of the Pareto-ON/OFF model, a Pareto distributed number of packets, with mean N and Pareto shape parameter β, are generated at the peak rate p packets/sec. The OFF times are also Pareto distributed with mean I and γ. The following set of values is used for the Pareto-ON/OFF flows in the simulations: N=5, $\beta = 1.2$, $I = 56000us$, γ=1.1, $p = 640$. The packet length is set to be 100 bytes. The transmission rate per flow r is fixed at 20kb/s. The burst assembly time is fixed at 120us.

Flows arrive at a source and destination pair according to a Poisson process with mean λ. The holding time of a flow is exponentially distributed with mean $\frac{1}{\mu}$. In the simulations, the traffic load is expressed as the number of traffic flows that arrive per second (flow arrival rate (FAR)). In the simulation, the average flow holding time is set to be *5s*. The values of the above parameters for our simulations are chosen based on the traffic distribution settings in the simulations in [22] since we are using the same traffic distribution and arrival model as that in [22].

We set the congestion threshold of a path to be 0.01 in terms of burst loss probability, which is used to trigger the algorithm to make adjustment of flow proportions when the measured burst loss probability of certain paths exceeds the threshold. If none of the paths' measured burst loss probability between a node pair exceeds the congestion threshold, the set of flow proportions will remain unchanged. In this way, the algorithm will try to avoid frequent adjustment if the traffic situation in the network is relatively light and stable. It will help to reduce the overhead of adjustment as well as the traffic load fluctuation in the network.

The values of the parameters used in the combined weighted sum of burst loss probability for a path are as follows: $\alpha_1 = 0.1, \alpha_2 = 0.2, \alpha_3 = 0.3, \alpha_4 = 0.4$. In the simulation, we use the measured burst loss probability of a path in the past four measurement time windows and get a weighted sum of them as the path's "quality" metric. The weights are given such that the most current measured burst loss probability will be given a higher weight, whereas the older measured burst loss probability will be given a lower weight. In this way, we will be able to quantify a path’s “quality” in a more objective and complete manner.

In the simulation, the burst loss probability and mean hop-length have been used as performance metrics. The burst loss probability is measured as the fraction of bursts dropped. The mean hop-length is measured as the average number of hops traversed by bursts.

We also implemented two other flow routing algorithms; equal proportion multi-path routing (EPMR) and hop-length based multi-path routing (HLMR). For EPMR, traffic flows that arrived at a source and destination pair will be distributed evenly among the multiple paths between the node pairs. For HLMR, flows will be routed through a path between a source and destination pair with a probability which is inversely proportional to the hop-length of a path. This probability will remain unchanged throughout the whole course.

We will first consider identical traffic demand to verify the performance improvement achieved by the proposed adaptive routing algorithm. In the case of identical traffic demand, the traffic that arrives at each node pair is homogeneous, i.e., all flows arrive at the same rate for different source and destination pairs and the rate is derived from the same Poisson process with a fixed mean value.

Since there are 42 node pairs in the network and the traffic is generated from the flow level, and each flow is the aggregate of many IP flows, a huge number of packets and bursts will be generated and processed in each experiment. Hence a

20,000,000*us* of total simulation time has been employed here in order for the network to be sufficiently loaded. We expect that similar performance may be achieved in a large time scale also.

First we consider different loading conditions for a specific time window size with identical traffic demand and a fixed burst assembly time threshold value for various paths between a node pair. Next we study the impact of time window size and congestion threshold values, followed by the performance enhancement achieved by varying the burst assembly time thresholds for different paths. The results are obtained based on the setup that there are two paths between any source and destination pair. The paths are the shortest link-disjoint paths between any source and destination pair obtained from Dijkstra's algorithm.

3.4.1 Identical Traffic Demand

3.4.1.1 Effect of Traffic Loading

Figure 3.3 shows the burst loss probability of the three flow routing methods with varying traffic load per node pair for time window size *T*=500,000us and the burst loss threshold value set to *Threshold*=0.01. Figure 4 shows the percentage of burst loss performance improvement achieved by the proposed adaptive proportional flow routing algorithm (APFRA) in comparison with EPMR and HLMR. We can observe that APFRA performs much better than the EPMR and HLMR. The performance is improved by up to 36 percent when compared to EPMR and 29 percent when compared to the HLMR. Since APFRA performs adaptive flow routing based on the periodic measurement of the "quality" of various paths in the network, congestion as well as burst loss due to contention has been reduced. Although both EPMR and HLMR distribute traffic flows across multiple paths, they perform worse than APFRA because they fail to keep track of the varying congestion situation on the different link-disjoint paths.

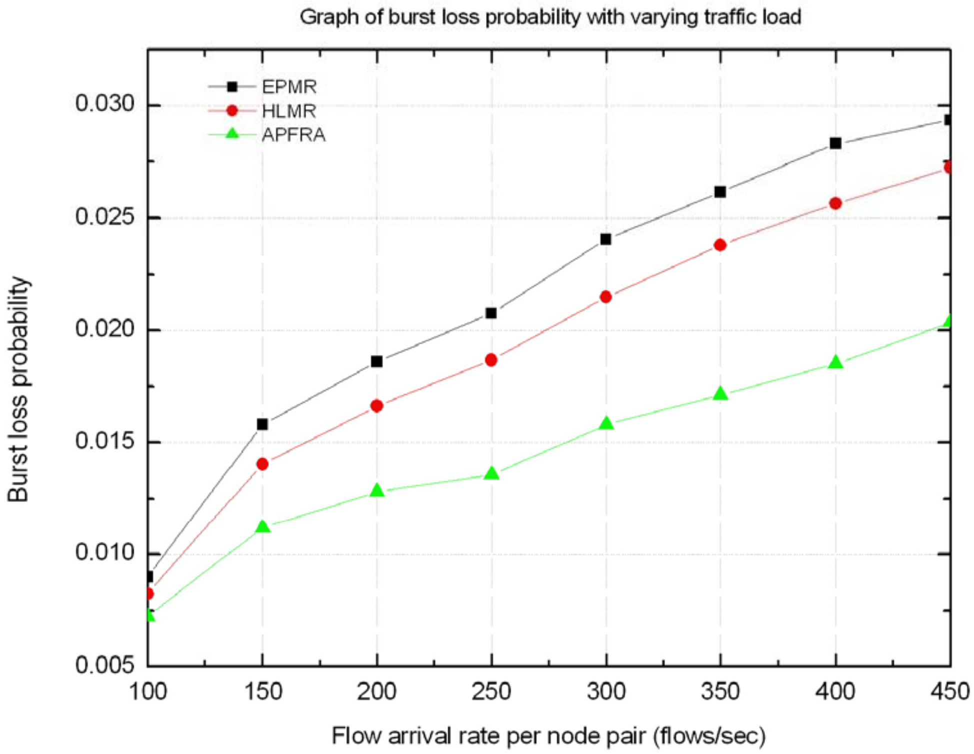

Figure3.3: Burst loss probability vs traffic load

From Figure 3.4, we observe that the percentage of performance improvement of the APFRA versus EPMR and HLMR first increases when traffic load increases. However beyond a certain traffic load, it starts to decrease. The reason is that when the traffic load is light, network resources are abundant and they are available for use to adjust the traffic flows among the multiple paths. In this case, the performance of APFRA increases with the increased traffic load as the adaptive routing becomes more useful in mitigating the congestion. However when the traffic load increases beyond a certain value, the performance improvement of APFRA decreases with the increased traffic load due to the shortage of network resources for traffic adjustment.

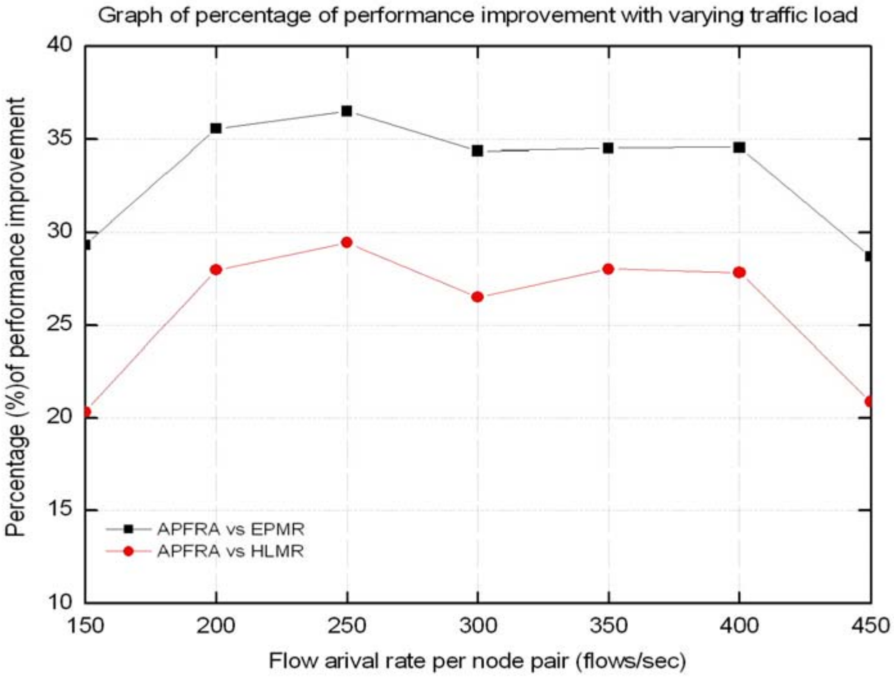

Figure3.4: Graph of performance improvement against traffic load

Figure 3.5 shows the mean hop-length traversed by a burst with varying traffic load per node pair for the three routing methods. The mean hop-length can reflect the delay, initial offset time and control burst signaling overhead in the network in an indirect manner. From the graph, we can see that APFRA has achieved a shorter mean hop-length when compared to the EPMR and HLMR. The reason is that EPMR and HLMR treat all the paths in a pre-set and fixed manner. Flows are routed through different paths based on the initial flow proportion settings throughout the whole course. Therefore on average they use the longer paths more often than APFRA since they do not adapt to the varying traffic conditions in the network.

We also observe that the mean hop-length for APFRA increases when the traffic load increases. This is because when the traffic load increases and the primary shortest path becomes more congested, the longer alternative path will be utilized more and given a higher proportion of the traffic flows.

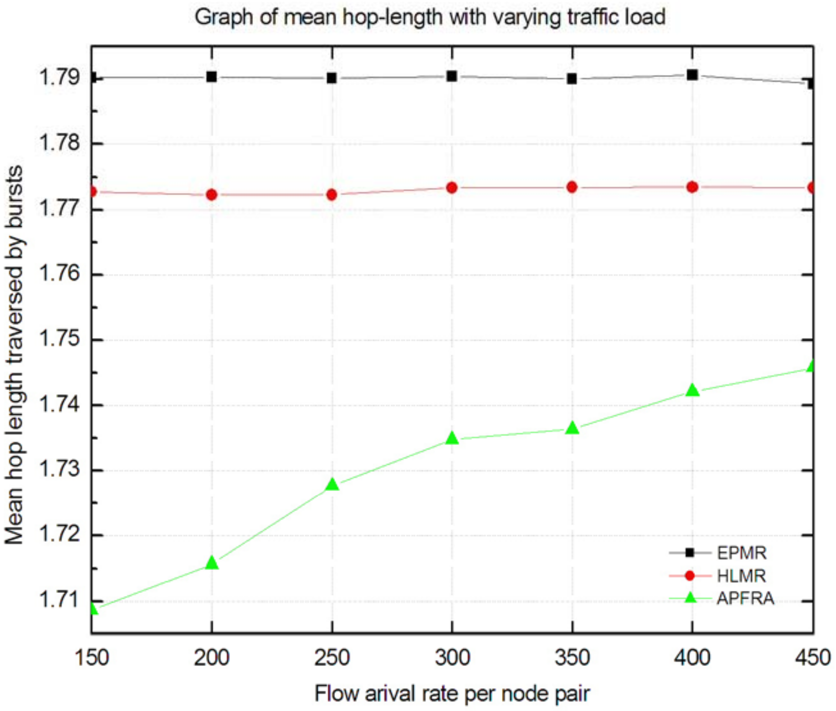

Figure3.5: Graph of mean hop-length against traffic load

3.4.1.2 Effect of Measurement Time Window Size

We evaluate the impact of varying the time window size on the performance of APFRA for two traffic load values, 300 flows/sec and 400 flows/sec.

Figure 3.6 plots the burst loss probability with varying time window size. We observe that the burst loss probability first decreases with then increases with increasing window size. The efficiency of the proposed adaptive routing algorithm depends on the accuracy of the traffic measurement. When the time window size is small, the collected traffic statistics in the network reflect only the short-term traffic load conditions and may not be accurate and objective enough. As a result, the proposed adaptive routing algorithm based on it does not work as well. Furthermore, a small time window size also results in frequent flow proportion adjustments, which will induce fluctuations and instability in the network. On the other hand, when the window size becomes too large, the algorithm will become incapable of tracking the dynamics of the changing traffic load in the network. The routing decisions will be based on stale traffic load information and this will render the algorithm inefficient.

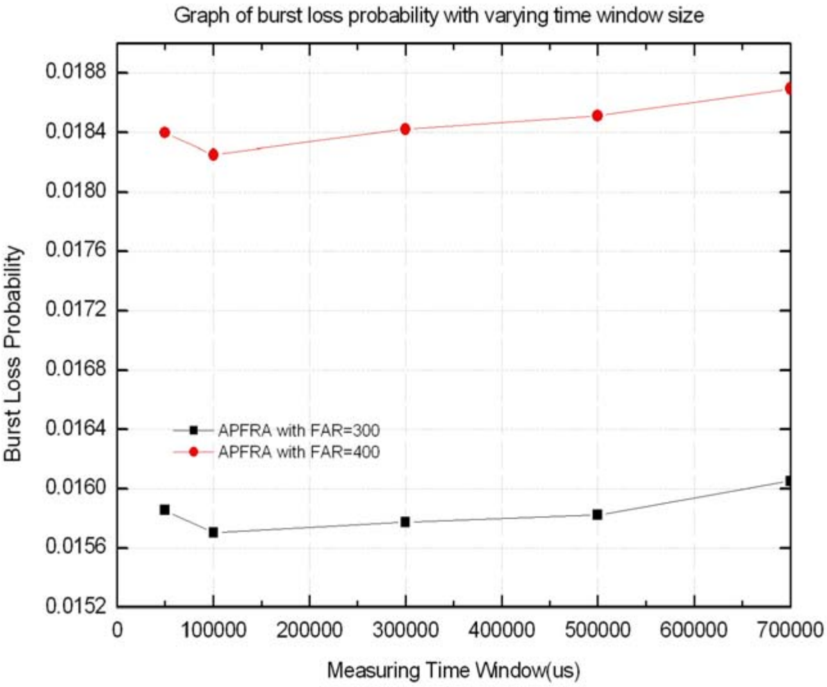

Figure3.6: Graph of burst loss probability against time window size

3.4.1.3 Effect of Congestion Threshold Values

Figure 3.7 shows the impact of varying congestion threshold values on the performance of the proposed flow routing algorithm. We vary the congestion threshold values to see how it affects the performance of APFRA under different offered traffic load situation. In the figure, each curve represents one offered traffic load scenario, with the flow arrival rate of 150 flows/sec, 250 flows/sec and 350 flows/sec per node pair, respectively. From the figure, we can see that for all the three cases, when the congestion threshold values decrease from 0.05 to 0.015, the proposed algorithm has achieved a considerable improvement in its burst loss performance. However, when the congestion threshold value decreases below 0.015, the results manifest a diminishing return effect. The proposed algorithm only achieves a negligible performance improvement even when the congestion threshold values decrease further below 0.015.

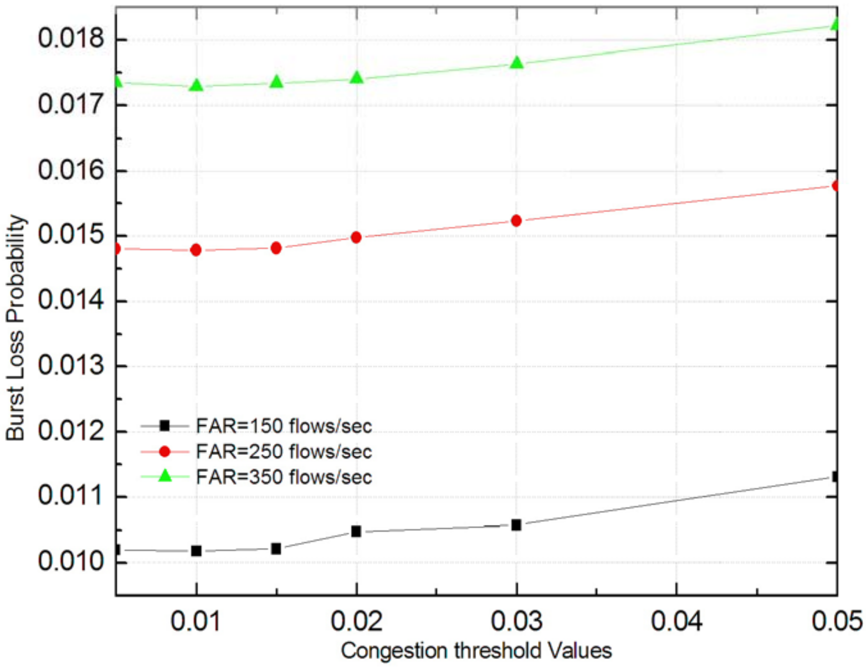

Figure3.7: Graph of burst loss probability against congestion threshold values

The congestion threshold values in the algorithm are set to control the extent and frequency in triggering the algorithm to make flow proportion adjustment. When the congestion threshold value is too high, it renders the algorithm incapable of tracking the congestion situation in the network and making necessary adjustment promptly to alleviate the congestion. Hence, under such circumstances, decreasing the triggering congestion threshold value can make the algorithm more capable of tracking and tackling the congestion situation in the network. However, when the threshold value decreases below a certain value, its effect on the performance improvement is diminishing or negligible. Furthermore, when the threshold value has decreased to too small a value, it will result in frequent adjustment of traffic assignment and defeat the original purpose for setting the congestion threshold values. It will also make the network unstable and degrade the performance of the algorithm.

3.4.2 Adaptive Proportional Flow Routing Algorithm with Varying Burst Assembly Time (APFRA-VBA)

For the multiple paths between a source and destination pair, a set of different burst assembly time values will be assigned to them. The basic principle is that a path with a better measured performance metric will be assigned the longer burst assembly time in the set. A path with a worse measured performance metric will be assigned a shorter burst assembly time so that it will be easier to find the available time slots to schedule the incoming bursts. The burst assembly time assignment will adapt to the dynamic traffic situation in the networks and change according to the measured “quality” of different paths periodically. In the simulation, the set of burst assembly time is chosen such that their average value is 120us, which is the same as the burst assembly time employed in the basic APFRA previously.

In Figure 3.8, the performance of APFRA-VBA is compared with the basic APFRA, EPMR and HLMR. From the graph, we can see that APFRA-VBA has achieved a considerable performance improvement when compared to the basic APFRA with fixed burst assembly time. On average APFRA-VBA has achieved 21 percent of performance improvement in terms of the burst loss probability as compared to the APFRA with fixed assembly time for every path. It is shown to be an effective way to further enhance the routing performance of the proposed adaptive proportional flow routing algorithm.

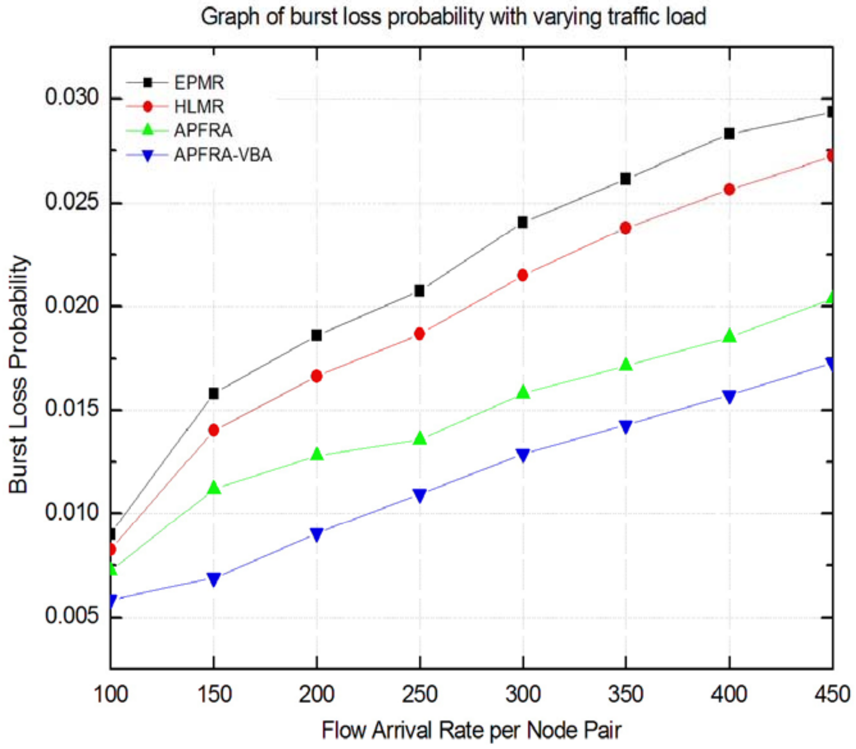

Figure3.8: Burst loss performance of various flow routing algorithms

3.4.3 Non-identical Traffic Demand

In this section, we will investigate the applicability of APFRA and APFRA-VBA in routing non-identical traffic demand. In a non-identical traffic demand, the flow arrival rate for a node pair is randomly selected from a set of flow arrival rates $\{r_1, r_2, r_3, r_4, r_5\}$ with equal probability. The traffic load is measured as the mean flow arrival rate which is given by the average of the flow arrival rates.

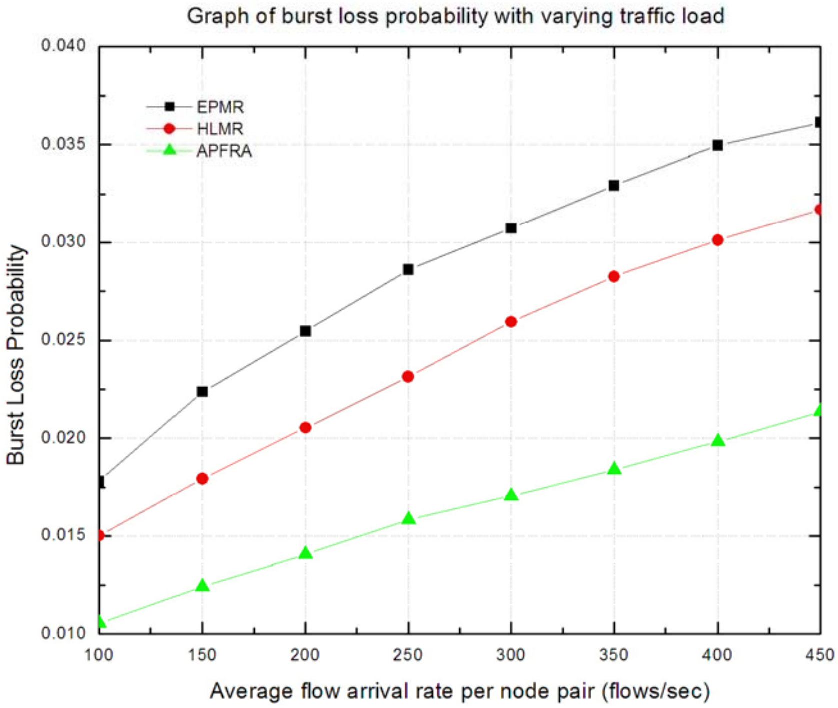

Figure3.9: Graph of burst loss probability for various non-identical traffic demands

Figure 3.9 shows the mean burst loss probability achieved by EPMR, HLMR and APFRA for various non-identical traffic demands. Figure 3.10 shows the percentage of burst loss performance improvement of APFRA in comparison with HLMR and EPMR. Figure 3.11 shows the mean hop-length traversed by bursts in EPMR, HLMR and APFRA. From the figures, we make similar observations as in the case of identical traffic demand. Hence, it shows that APFRA works well even under the situation when traffic is distributed unevenly over the whole network. It verifies that APFRA can be applied to different traffic scenarios.

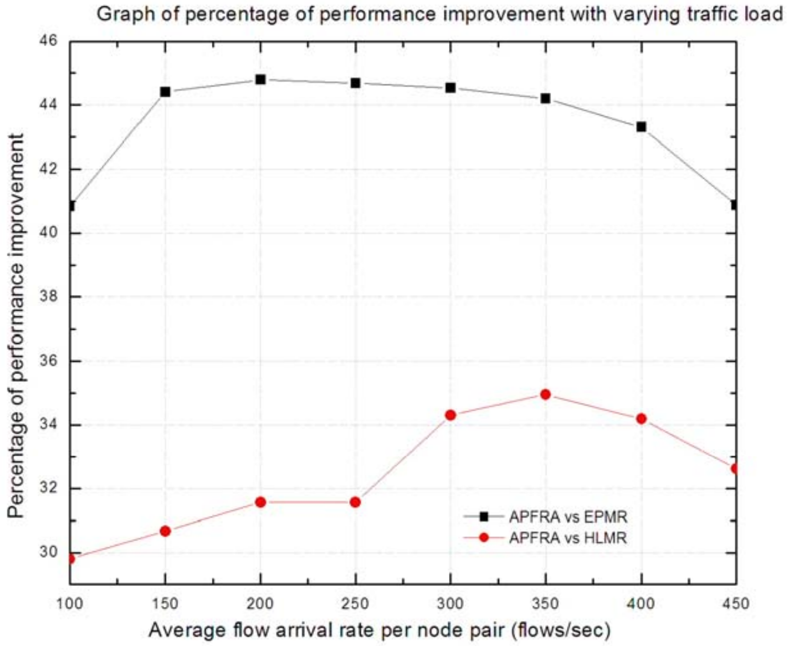

Figure3.10: Graph of percentage of performance improvement for various non-identical traffic demands

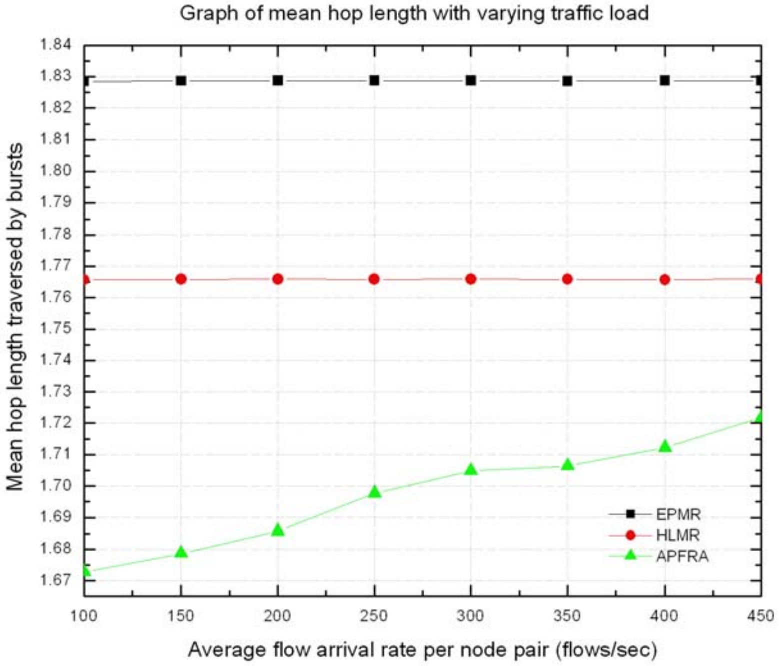

Figure3.11: Graph of mean hop-length for various non-identical traffic demands

3.4.4 Performance of APFRA-VBA under Non-identical Traffic Demand

In Figure 3.12, the performance of APFRA-VBA is compared with the basic APFRA, EPMR and HLMR under the non-identical traffic demands. We make similar observation as in the case of identical traffic demand and APFRA-VBA has achieved a considerable performance improvement when compared to the basic APFRA with fixed burst assembly time. Hence it verifies that it is an effective way to further enhance the performance of the proposed adaptive flow routing algorithm even in the case when traffic is not evenly distributed in the network. It is applicable to different traffic scenarios.

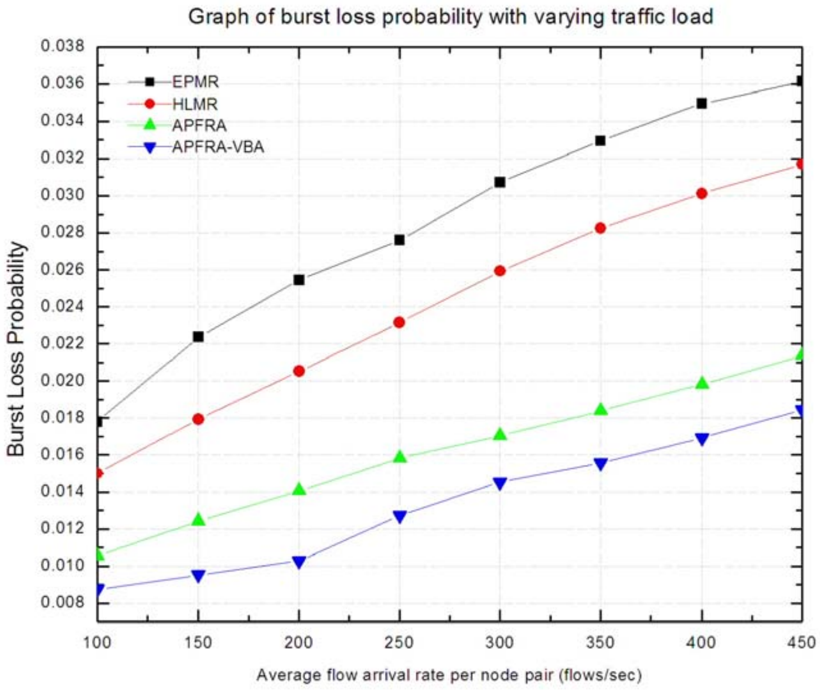

Figure3.12: Graph of burst loss performance various non-identical traffic demands

3.4.5 Summary of Results

We now summarize the important observations made from the simulations results.

1. APFRA significantly improves the performance in terms of burst loss reduction over EPMR and HLMR. Furthermore by routing the traffic at the flow level, we can avoid the burst arrival out-of-sequence problem which is significant in the previously proposed load balancing schemes.
2. APFRA performs well under different traffic situations.
3. By carefully choosing the time window size as well as the congestion threshold value, burst loss probability can be reduced significantly and effectively with lower overhead.
4. APFRA-VBA is shown to be an effective way in further enhancing the performance of the APFRA with fixed burst assembly time.

Chapter 4
Gradient Projection based Multi-path Traffic Routing in OBS Networks

4.1 Introduction

In this chapter, we consider the problem of adaptively and efficiently mapping the offered burst traffic into multiple end-to-end paths between each ingress and egress pair in OBS networks. The main goal is to minimize the overall burst loss in the network by adaptively balancing the burst traffic among multiple paths based on the measurement and analysis of path congestion. As has been mentioned in Chapter 2 of the thesis, delay in OBS networks is predictable and predominantly determined by the propagation delay, which is fixed for a specific path. Hence, delay in OBS networks is not an appropriate performance metric to implement the multi-path traffic routing. Instead, the burst loss probability is a more appropriate performance metric to evaluate the impact of the traffic load since the burst loss on a link in OBS networks is directly impacted by its offered traffic load.

We propose a new multi-path routing scheme in OBS networks which is based on the gradient projection optimization algorithm [23] to determine the traffic splitting or mapping among the multiple paths between each SD pair. In the scheme, we assume that N_s paths are pre-established between each SD pair *s*. They are the first N_s link-disjoint shortest-hop paths between the SD pair. The key idea is to let the source node of each SD pair periodically measure the offered load on the links that are traversed by its N_s alternative paths. Then at the end of each time window, the source node calculates each path's first derivative length to evaluate the impact of the offered burst traffic on the path. Based on the above information, we will apply the gradient projection algorithm to obtain the amount of burst traffic that will be offered to each alternative path for the next time period.

Our proposed multi-path routing algorithm in OBS networks works under the assumption that the traffic in the network is quasi-stationary which means that the network traffic statistics change relatively slowly (much longer than the round-trip delay between the ingress and egress nodes). Recent measurements of Internet traffic indicate that the aggregate load on links changes relatively slowly in the scale of minutes [19]. Since bursts in OBS networks are assembled from IP flows, we expect that the traffic exhibits similar behaviors and thus it makes our assumption reasonable.

The proposed algorithm works in a distributed manner in which all SD pairs perform their traffic routing independently. Furthermore, the proposed algorithm only uses a simple measurement mechanism which does not incur much signaling and processing overhead. Through extensive simulations, it is shown that our proposed algorithm significantly improves the network performance in terms of burst loss probability when compared to the load balancing scheme proposed in

[22]. Furthermore, our proposed algorithm exhibits good routing stability and good capability in adapting to traffic variations in the network.

The rest of the chapter is organized as follows. Section 4.2 formulates an analytical optimization model for the multi-path routing and load-balancing problem. Section 4.3 describes the implementation details of the proposed multi-path routing scheme. Section 4.4 presents the simulation setup and the performance study.

4.2 The Optimization Problem

4.2.1 The Analytical Model

In this section, we formulate the optimization problem of interest, illustrate the network model for the analysis, and list the assumptions we make.

The network consists of L unidirectional links. It is shared by a set ***S*** of source and destination (SD) pairs, denoted by ***S***={1,2,……S}. Each of this SD pair *s* has a set of N_s link-disjoint alternative paths available to it. We let $P_s = \{1,2,......,N_s\}$, and define the set of all paths as $P = \cup_{s \in S} P_s = \{1,2,......,N\}$ where $N = \sum_{s \in S} N_s$. By definition, no two distinct SD pairs use the same path, but some of their paths may share links.

The total input traffic rate between an SD pair *s* is given by r_s and it routes x_{sp} amount of traffic on path $p \in P_s$ such that

$$\sum_{p \in P_s} x_{sp} = r_s \text{ , for all } s \tag{7}$$

Let $x_s = (x_{sp}, p \in P_s)$ be the traffic rate vector of SD pair *s*, and let

$x = (x_{sp}, p \in P_s, s \in S)$ be the vector of all rates. This traffic rate can be interpreted as the offered burst arrival rate under the circumstances of OBS. Then, the offered burst rate on a link $l \in L$ has a value that can be approximated by the sum of the source burst arrival rates on all paths that traverse link l:

$$x^l \approx \sum_{s \in S} \sum_{l \in p, p \in P_s} x_{sp} \tag{8}$$

It is an approximation because there may be some burst loss that happens before the traffic arrives at link l and the direct summation of source traffic rate may not be accurate.

For each link l, let $C_l(x^l)$ represent the link cost as a function of the link burst rate x^l. We assume that, for all l, $C_l(.)$ is convex and continuously differentiable. The objective is to minimize the total cost $C(x) = \sum_l C_l(x)$ by optimally mapping the traffic on paths in P_s:

$$\min_x C(x) = \min_x \sum_l C_l(x^l) \tag{9}$$

subject to $$\sum_{p \in P_s} x_{sp} = r_s \,, \quad \forall s \in S \tag{10}$$

$$x_{sp} \geq 0, \forall p \in P_s, s \in S \tag{11}$$

A vector is called a feasible rate if it satisfied conditions *(9)-(11)*. A feasible rate x is called optimal if it is a minimizer to equations *(8)-(11)*.

The derivative of the objective function with respect to x_{sp} is [23]:

$$\frac{\partial C}{\partial x_{sp}}(x) = \sum_{l \in p} C_l^{'}(x^l)$$

We will interpret $C_l^{'}(x^l)$ as the cost derivative of link *l*, and $\frac{\partial C}{\partial x_{sp}}(x)$ as the first derivative length of path p .

4.2.2 The Distributed Algorithm

We can use the gradient projection algorithm to solve the above constraint optimization problem, where the constraint set θ is defined by *(4)* and *(5)*. Each iteration of the algorithm takes the following form:

$$x(t+1) = \prod{}_{\theta}[x(t) - a(t)\nabla C(t)] \qquad (12)$$

where $\nabla C(t)$ is the gradient vector whose $(s, p)^{th}$ element is the first derivative length of path $p \in P_s$ at iteration t ($[\nabla C(t)]_{sp} = \partial C / \partial x_{xp}$), $a(t) > 0$ is the step size, and $\prod{}_{\theta}[\vartheta]$ is the projection of a vector ϑ onto the feasible set with respect to the Euclidean norm. The algorithm terminates when there is no appreciable change, i.e. $\|x(t+1) - x(t)\| < \in$ for some predefined $\in$.

The above iteration can be carried out in a distributed manner independently by each SD pair *s* without coordination with other SD pair [23, 45]. Hence, for each individual SD pair *s*, the iteration can take on the following form:

$$x_s(t+1) = \prod{}_{\theta_s}[x_s(t) - a_s(t)\nabla C_s(t)] \qquad (13)$$

where $\nabla C_s(t) = (\partial C / \partial x_{sp}(x(t)), p \in P_s)$ is the vector of first derivative lengths of paths in P_s , and $\prod{}_{\theta_s}$ denotes a projection onto the feasible space of SD pair *s*.

The challenging part of this problem lies in the online solution of optimally mapping/splitting traffic since the first derivative length of a path $\nabla C_s(t) = (\partial C / \partial x_{sp}(x(t)), p \in P_s)$ may not be available immediately and can only be estimated empirically by averaging several measurements over a period of time.

Some methods have been proposed in [38, 39] to solve this problem of multi-path routing in traditional IP networks to deal with the dynamic changing of input traffic. Similarly, for the multi-path traffic routing problem in OBS networks, we can also treat it as a constraint optimization problem. We can extend the gradient projection algorithm to tackle this problem in OBS networks with careful selection of performance metrics as well as implementation methods to achieve a considerable performance improvement over the previously proposed heuristic-based multi-path routing or load balancing algorithms in [20, 21, 22].

4.3 Gradient Projection based Multi-path Traffic Routing in OBS Networks

In this section, we explain in detail the proposed Gradient Projection based Multi-path routing (GPMR) algorithm in OBS networks, which performs several functions such as traffic measurement, traffic assignment and traffic distribution. As stated earlier, GPMR is run for each individual SD pair independent of other SD pairs. Without loss of generality, we explain the working of GPMR for a specific SD pair s. N_s link-disjoint paths are pre-established by a variant of Dijkstra’s algorithm. It works in the way that each time when the shortest path is found between an SD pair, the links traversed by the paths will be removed from the topology. Then the next shortest path between the SD pair will be found by

Dijkstra's algorithm only based on the updated topology with the removed links to ensure that all the alternative paths between the SD pair are link-disjoint.

4.3.1 Notations

For the ease of exposition, we define the following notations:

$p_1, p_2, ...p_{N_s}$: multiple link-disjoint paths between the SD node pair

$T(i)$: ith measuring window

$load_l(T(i))$: measured normalized offered load on the link *l* in time window *T(i)*

x_k^i : burst rate assigned to the k th path in time window *T(i)*

r_s *:* total burst rate offered to SD pair *s*

{ $x_1^i, x_2^i, ...x_{N_s}^i$ }: the set of burst rates distributed to the paths between the SD pair in time window *T(i)* and $x_1^i + x_2^i + ... + x_{N_s}^i = r_s$.

4.3.2 Path First Derivative Length Estimation

The traffic rate adjustment process is invoked periodically for every time window. *T(i)*. Let $T_s \subseteq \{T(1), T(2), ..., \}$ be a set of times at which SD pair s adjusts its burst rate allocation based on its current knowledge of the first derivative lengths of paths $p \in P_s$. At a time $t \in T_s$, s calculates a new rate vector,

$$x_s(T(i+1)) = \prod\nolimits_{\theta_s}[x_s(T(i)) - a_s(T(i))\nabla C_s(T(i))] \qquad (14)$$

and splits its burst rate r_s starting from time $T(i+1)$, along its paths in P_s according to $x_s(T(i+1))$ until after the next update time in T_s . Here, $\nabla C_s(T(i))$ is the estimate of the first derivative length vector at time *T(i)*, and is calculated as follows.

The new rates calculated by the SD pair may only be reflected in the link flows after certain delays. We model this by getting the weighted average of past traffic rates of a link instead of using its instantaneous measured value [38].

$$\overset{\wedge}{x^l} = \sum_{m=0}^{W-1} \beta_m x_l(T(i-m)), \text{ for } l = 1,2,...,L \tag{15}$$

In the above, $\overset{\wedge}{x^l}(T(i))$ represents the estimated burst rate available at link *l* at time T(i) . $\beta_0, \beta_1, ...\beta_{W-1}$ are the weights assigned to each of the measured link rates in the previous W time windows and $\beta_0 + \beta_1 + ... + \beta_{W-1} = 1$. The weights are assigned such that the more recent time windows' measured rates are given higher weightage, and that of older time windows are given lower weightage.

For the ease of implementation, instead of measuring directly the burst rates on a link, we measure the sum of the length of all the bursts offered to the link within *T(i).* Based on this statistics, we approximate the measured normalized offered load to link *l* by the following:

$$load_l(T(i)) = \frac{BL_l(T(i))}{Cap_l * T(i)} \tag{16}$$

where $BL_l(T(i))$ is the total burst length offered to link *l* within the time window *T(i)* and Cap_l is the capacity of link *l*. Note that, in essence, $BL_l(T(i))$ is proportional to $x_l(T(i))$, which is the aggregate offered burst rate to the link *l*. Approximately, $BL_l(T(i)) \approx Mean\_burst\_length * x_l(T(i)) * T(i),$ assuming that the burst length has the same average value over the whole network. Then the relationship between $x_l(T(i))$ and $load_l(i)$ can be approximated by

$$load_l(T(i)) = \frac{Mean\_burst\_length * x_l(T(i))}{Cap_l} = K_l * x_l(T(i)) \qquad (17)$$

where $K_l = \dfrac{Mean\_burst\_length}{Cap_l} > 0$ is a coefficient constant for link *l*.

We form the estimate of the normalized offered load to link *l* in *T(i)* by getting the weighted average as follows:

$$\overset{\wedge}{load}_l(T(i)) = \sum_{m=0}^{W-1} \beta_m load_l(T(i-m)) = \sum_{m=0}^{W-1} \beta_m K_l x_l(T(i-m)) \qquad (18)$$

for $l = 1,2,...,L$. From the above expression, we can obtain

$$\overset{\wedge}{load}_l(T(i)) = K_l \hat{x}_l((T(i)) \qquad (19)$$

Let C_l be the cost function of link *l* and we assume that it is a convex function of the estimated normalized load on link *l*. Then from the above expression, we can have the expression for C_l in terms of the estimated burst rate on a link as follows:

$$C_l(\overset{\wedge}{load}(T(i))) = C_l(K_l \hat{x}_l(T(i))) \qquad (20)$$

It is also a convex function of the estimated burst rate on the link.

Each SD pair *s* will estimate the first derivative length of a path $p \in P_s$ by collecting a certain number of measurements in the previous time windows and forming their weighted average as follows:

$$FDL_{sp}(T(i)) == \sum_{n=0}^{N-1} \sum_{l \in p} \mu_n C_l^{'}(K_l \hat{x}_l(T(i-n))) \qquad (21)$$

where $u_0 + u_1 + ... + u_{N-1} = 1$. Again the estimated FDL is obtained by averaging over the past values of the path's first derivative lengths. The weights are assigned such that the more recent time windows' values will be given higher weightage, whereas the older time windows' values will be given lower weigtage.

4.3.3 Traffic Measurement

Traffic measurement is invoked periodically in each time window. The purpose of traffic measurement is to collect traffic statistics for each path by sending probe packets and then calculate the first derivative length of the path to evaluate the impact of traffic load. We collect the normalized traffic load offered to the links along each path within a time window to perform the multi-path routing algorithm. To achieve this, we set a counter at each link in the network. The counter is used to record the number of bursts as well as the sum of burst length sent through the link within each time window. At the end of *T(i),* the ingress sends out probe packets, along paths $p_1, p_2, ...p_{N_S}$ separately to collect the record from the counter at each intermediate link. When the probe packet propagates all the way down to the egress node and then comes back to the ingress node, the total burst length offered to each of the path's links within *T(i)* can be obtained by the ingress node as $BL_l(T(i))$. Then the normalized load offered to each link l can be approximated as equation (16).

Based on the above information, we can form the estimate of the first derivative length along each path based on the description in part *4.3.2* and carry out the optimization process.

One important point that should also be noted here is that the size of the measurement time window *T(i)* should be set sufficiently larger than the longest propagation round trip time (RTT) in the network. This is to reduce the impact of the probe packet propagation delay on the accuracy of measurement as well as the performance of the proposed routing algorithm.

4.3.4 Cost Function and Convergence

The choice of cost function determines the parameters to be measured and equalized in carrying out the proposed multi-path routing algorithm. From [23], if the chosen cost function can be proven to be strictly convex, the convergence of the Gradient Projection based constraint optimization algorithm can be guaranteed. Hence, the choice of cost function is vital in ensuring the convergence of our proposed algorithm as well as achieving our goal in reducing overall burst loss in the networks.

We assume that each unidirectional link in the network consists of K wavelengths with the same capacity. We assume that the arrival process of bursts to each link in the network is Poisson. This is clearly an approximation since, even if arrivals to the network are Poisson, burst arrivals to a given link are reduced due to loss in previous links and hence are not Poisson. However, whenever the burst loss is small, we can assume that the thinned process remains Poisson. Therefore, under the Poisson arrival assumption, the link can be approximated by an M/G/K/K queue. The burst loss probability on the link can be approximated by the Erlang-B loss formula as a function of its normalized offered load $load_l$ as follows:

$$Er_l(load_l, K) = \frac{(load_l)^K / K!}{\sum_{n=0}^{k} (load_l)^n / n!} \tag{22}$$

We choose the cost function in our proposed multi-path routing algorithm as follows,

$$C_l(load_l) = load_l * Er_l(load_l, K) \tag{23}$$

which is the amount of burst loss at link l. Hence, the objective of the algorithm is to minimize the overall burst loss in the network which is approximated by the sum of cost functions of all the links in the network.

As has been mentioned above, to guarantee the convergence of our proposed scheme, the chosen cost function must be convex. In the regime of interest (e.g. with link normalized load or utilization level being greater than zero percent and K>0 in OBS), it is well known that the Erlang-B is a strictly convex function of $load_l$ [44]. Now we prove the convexity of our chosen cost function given by Equation (23). For this, we consider the second derivative of the cost function.

$$\begin{aligned} C''(load_l) &= (load_l * Er_l(load_l))'' \\ &= (E_r(load_l) + load_l * Er_l'(load_l))' \\ &= 2Er'(load_l) + load_l * Er''(load_l) \end{aligned}$$

Due the convexity of the Erlang-B function, $Er''(load_l) > 0$ for $load_l > 0$. In addition, the Erlang-B is also monotonically increasing, hence we have $Er_l'(load_l) > 0$ for $load_l > 0$. Hence, $C''(load_l) > 0$ for $load_l > 0$. It is known that a function with monotonically increasing first order derivatives, i.e. with positive second derivatives, is a convex function [23]. Hence, our chosen cost function is a convex function of the normalized offered load $load_l$. Since the offered burst rate to link *l* is given by $x_l = load_l / K_l$, our chosen cost function is a strict convex function of x_l in the regime of interest as well. Therefore, the chosen cost function satisfies the convexity condition for our proposed gradient projection based multi-path routing algorithm to converge to the optimal solution.

Although the gradient projection based optimization framework is also adopted in [38], there are some major differences between our work and [38]. In [38], the performance metrics/cost function chosen is the average end-to-end delay experienced by each probe packet. The FDL of the cost function obtained by empirical measurements due to the difficulty in obtaining the explicit close form formula of the cost function. Therefore in [38], it is hard to theoretically provide the guarantee on the convergence of their scheme under different traffic conditions and network scenarios. However the convergence can be guaranteed in our proposed scheme as has been proved above under chosen cost function given by Equation (23) by considering the bufferless property of OBS networks.

4.3.5 Traffic Assignment

From the above description, in the gradient projection algorithm, each iteration takes the following burst rate vector form for each SD pair *s*:

$$x_s(T(i+1)) = \prod_{\theta_s} [x_s(T(i)) - a_s(T(i)) \nabla C_s(T(i))] \tag{24}$$

We have described the way to obtain the estimate of the first derivative length vector $\nabla C_s(T(i))$ in the previous section. Then we can follow the method described in [23] to obtain the traffic rate assigned to each path between SD pair *s* at each iteration. Then we can obtain $\{x_1^{i+1}, x_2^{i+1}, ... x_{N_S}^{i+1}\}$ through the above rate vector $x_s(T(i+1))$ at the end of each time window.

It should be noted that the step size $a_S(T(i))$ has to be chosen carefully. If we choose too small a value, it will slow down the convergence speed of the proposed routing algorithm and affect the overall performance. If we choose too large a value, it will make the proposed algorithm overshoot beyond the optimal solution quickly and the iterations keep on crossing over the optimal solution point. In this case it will cause unnecessary traffic fluctuations and induce overall performance degradation. According to [23], if we have the step size given by the inverse of the Hessian matrix, $H(x_s(T(i)) = \nabla^2 C_s(T(i))$, we will achieve super-linear convergence rate. However, it makes the method impractical since the inverse of $H(x_s(T(i))$ is required at every iteration. Hence, instead of using $H^{-1}(x_s(T(i))$, it is found that using $a_s(T(i)) = H_{Diagonal}{}^{-1}(x_s(T(i))$ given as follows is good enough for achieving desirable convergence performance.

$$H_{Diagonal}{}^{-1}(x_s(T(i)) = \begin{bmatrix} (\frac{\partial^2 C(x_s(T(i))}{\partial(x_1^i)^2})^{-1} & Zeroes & 0 \\ Zeroes & ... & Zeroes \\ 0 & Zeroes & (\frac{\partial^2 C(x_s(T(i))}{\partial(x_{N_S}^i)^2})^{-1} \end{bmatrix} \qquad (25)$$

Now, the improved iteration becomes:

$$x_s(T(i+1)) = \prod\nolimits_{\theta_s}[x_s(T(i)) - H_{Diagonal}{}^{-1}(T(i)) * \nabla C_s(T(i))] \qquad (26)$$

and $x_1^{i+1}, x_2^{i+1}, ...x_{N_s}^{i+1}$ will be given by the above rate vector $x_s(T(i+1))$ at the end of each time window.

4.3.6 Traffic Distribution

The traffic distribution function distributes traffic arriving at the ingress node to the multiple paths between the SD pair based on the traffic splitting proportions from the proposed routing algorithm. The way in which the traffic will be distributed is similar to that presented in [22]. The proposed algorithm calculates the rate at which the traffic should be distributed along the alternative paths between the SD pairs. The traffic assignment is on a per-flow basis. Once a flow is distributed to a path, the packets belonging to the flow should be transmitted on this path. However, flows can be shifted from one path to the other in different time windows in order to make the traffic rate adjustment along the paths between the SD pairs. Reordering of packets will occur only if flows are shifted from a longer path to a shorter path when the traffic assignment is adjusted.

4.4 Experimental Setup and Simulation Results

In this section, we show the performance of our proposed gradient projection based multi-path routing algorithm (GPMR) through extensive simulations on the Pan-European optical network shown in Figure 4.1.

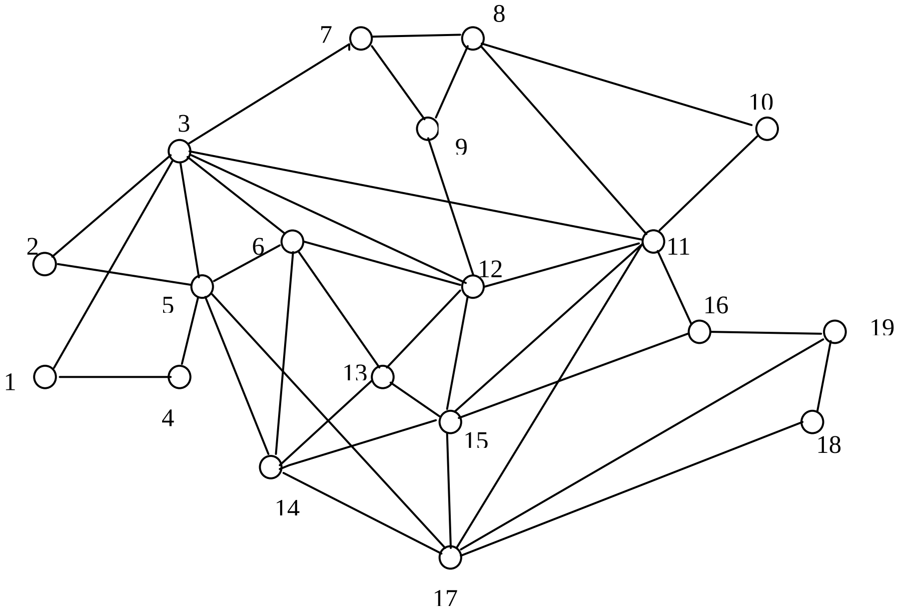

Figure4.1: Pan European optical network

The network consists of 19 nodes and 38 bidirectional links. A bidirectional link comprises two unidirectional fibers in opposite directions. Each fiber has 4 wavelengths at 1 Gb/s transmission rate. The reservation scheme employed here is JET. We use the basic LAUC-VF to schedule the bursts in the data channel. Altogether there are 342 source and destination pairs in the network.

In order to get more realistic results, the long range dependent traffic model is employed in our study. In this model, traffic that arrives at each node pair in the network is the aggregation of multiple IP flows. Each IP flow is an ON/OFF process with Pareto distributed ON and OFF times. During each ON period of the Pareto-ON/OFF model, a Pareto distributed number of packets, with mean N and Pareto shape parameter β, are generated at the peak rate p packets/sec. The OFF times are also Pareto distributed with mean I and γ. Since there are 342 node pairs in the network and the traffic is generated from the flow level, a huge number of packets and bursts will be generated and processed in each experiment. In such a large-scale optical backbone network, the offered traffic load on each link is the aggregation of a large number of independent traffic flows. Hence, the long range dependence within the aggregate traffic will be reduced to zero or to very short range dependent and the traffic arrival at each link in the network can be approximated by the Poisson process [46]. This justifies the use of Erlang-B loss formula in our cost function in our simulations.

The results are obtained based on the setup that there are two link-disjoint shortest paths between each SD pair in the network. The paths are formed based on the variant of Dijkstra's algorithm.

We use the mean burst loss probability and the mean hop length as the performance metrics. The burst loss probability is measured as the portions of bursts dropped. The mean hop length is measured as the average number of hops traversed by a burst.

We have also implemented the load balancing scheme proposed by Li Jing in [22], which is called adaptive alternate routing algorithm (AARA), to demonstrate the effectiveness and advantages of our proposed gradient projection based multi-path routing algorithm (GPMR) in distributing and load balancing the traffic in the networks. From the simulation results, we will show that the proposed GPMR in OBS achieves a considerable performance improvement over AARA, in terms of burst loss, mean hop length and also routing stability.

The following set of values is used for the Pareto-ON/OFF flows in the simulations: N=5, $\beta = 1.2$, $I = 56000us$, γ =1.1, $p = 640$. The packet length is set to be 100 bytes. The transmission rate per flow r is fixed at 20kb/s. In the simulations, traffic load is measured as the number of flows that arrive per second (flow arrival rate (FAR)). The values of the above parameters for our simulations are chosen based on the simulation setting in [22] for a better comparability since we are using the same network topology and traffic distribution model as those in [22].

We consider two traffic scenarios in our simulations, identical traffic and non-identical traffic demands, to verify the effectiveness of our proposed algorithm under different traffic situations. In an identical traffic demand, the traffic for all SD pairs arrives at the same rate and the flow arrival rate is derived from the same Poisson process with the fixed mean value. In a non-identical traffic demand scenario, the traffic flow arrival rate for different SD pairs is derived from the Poisson process with a different mean value.

First, we study the burst loss performance of the proposed GPMR for different loading conditions with identical traffic demand. Then, the link load and the network burst loss dynamics will be investigated to verify the convergence property of GPMR. Finally, we will study the performance with non-identical traffic demand.

4.4.1 Identical Traffic Demand

4.4.1.1 Effect of Traffic Loading

Figure 4.2 shows the burst loss probability with varying traffic load per node pair for time window size *T=200,000us*. The total simulation time employed here is *20,000,000us*. The initial traffic proportion assigned to each path between every SD pair in the network is chosen randomly. Then the GPMR and AARA algorithms come into picture and make traffic adjustment accordingly. Figure 4.3 shows the percentage of burst loss performance improvement achieved by GPMR in comparison with AARA. We observe that the proposed GPMR has achieves a considerable burst loss performance over AARA. Under the low-load region, the burst loss probabilities under GPMR scheme is up to 60-70 percent lower than that of AARA. While at moderate load, the decrease in burst loss probability remains significant (30-40 percent); even at high loads, the improvement is still considerable (around 20 percent). GPMR helps to reduce the network burst loss more effectively and it tends to more evenly and better balance the load among the multiple paths between the SD pairs.

From Figure 4.3, we can see that the performance improvement for GPMR decreases with the increase of the traffic load. When the traffic load is light, network resource is abundant and is available for adjusting the traffic load. In this case, the performance improvement by GPMR is more significant and it is more effective in mitigating the congestion and balancing the load in the network. However, when the traffic load increases, the percentage of performance improvement for GPMR decreases due to the shortage of network resources.

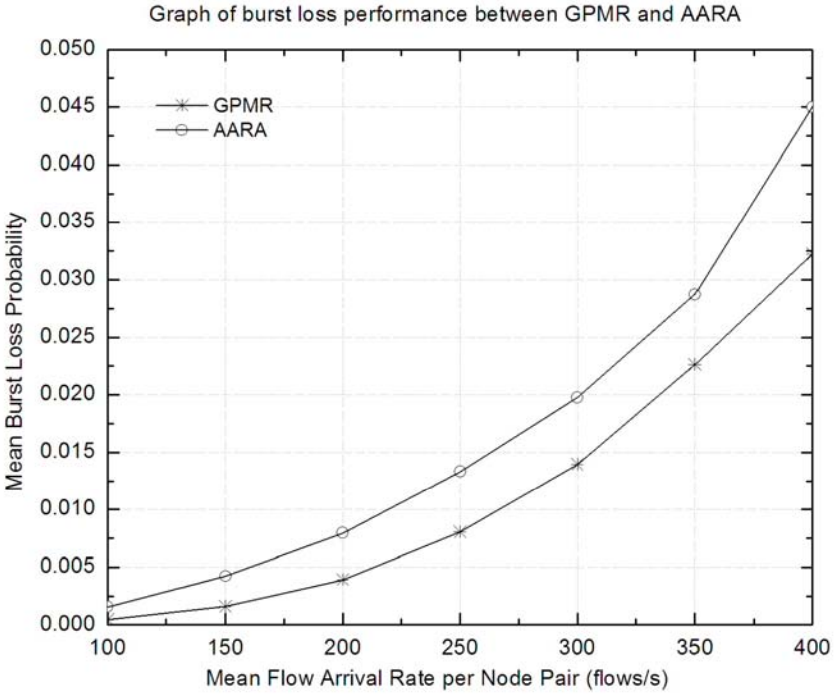

Figure4.2: Graph of burst loss probability against traffic load

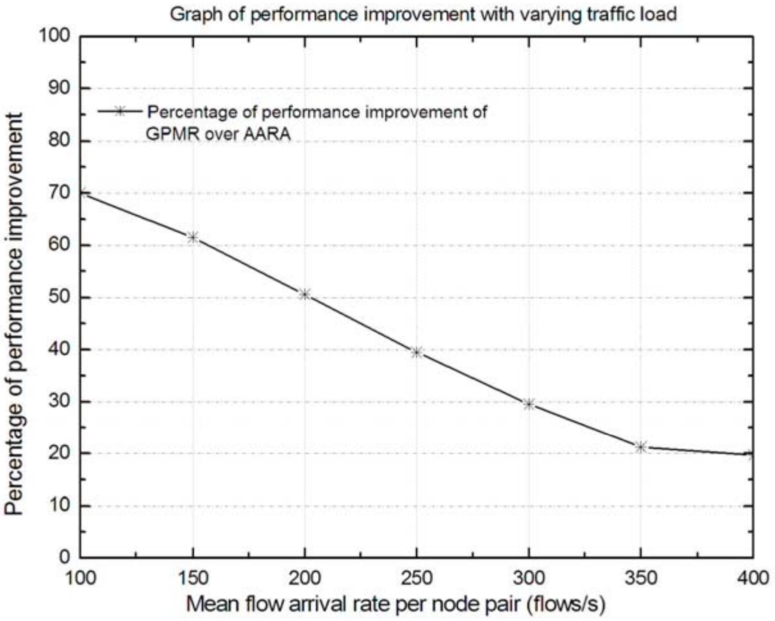

Figure4.3: Graph of percentage of performance improvement

Figure 4.4 shows the mean hop-length traversed by a burst with varying traffic load per SD pair. The mean hop-length could reflect delay, signaling overhead, and initial offset time in the network in an indirect manner. We can see that the proposed GPMR achieves a smaller mean hop length compared to AARA. This implies that on average, GPMR incurs less delay, signaling overhead, and initial offset time than AARA and routes the traffic load in a more effective manner. We can also see that the mean hop length for GPMR is slightly larger than that of the single shortest path routing. It shows that the additional delay and signaling overhead incurred by GPMR is rather low when compared to its significant performance improvement achieved. Furthermore, we also observe that the mean hop length for GPMR only varies within a small range when traffic load increases whereas in AARA, the mean hop-length decreases when traffic load increases. The proposed GPMR's routing decision is based on the first derivative length estimation of the paths between the SD pairs and does not purposely set the hop length of a path as a penalizing factor as in AARA. Hence, it will not favor the shorter paths to route traffic when the traffic load increases.

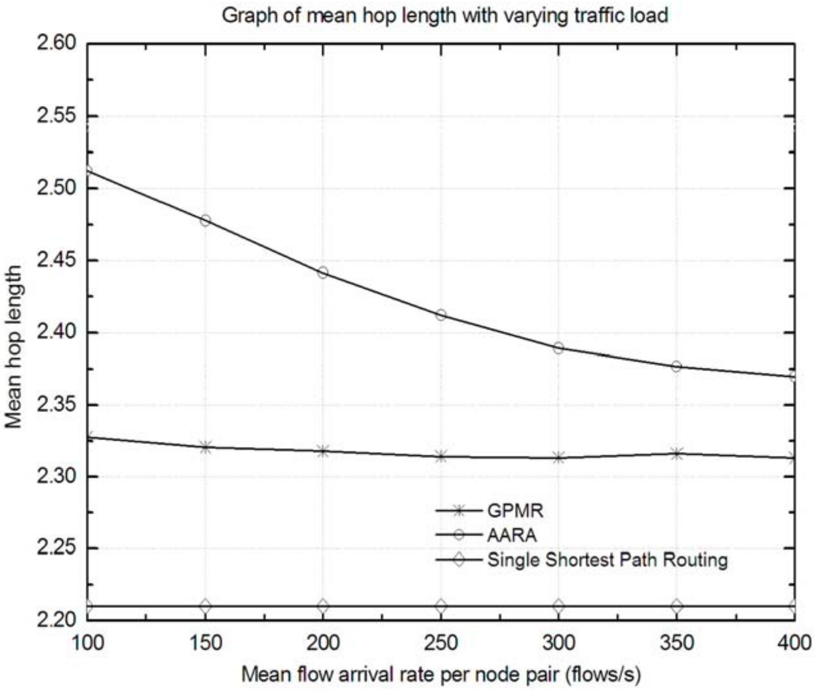

Figure4.4: Graph of mean hop length against traffic load

4.4.1.2 Dynamics of the GPMR Algorithm under Identical Traffic Demand

In this section, through the simulation results, we would like to show that the proposed GPMR algorithm is stable and robust in the sense that it minimizes congestion and quickly balances the load among multiple paths between SD pairs in a reasonable time under dynamic traffic situation. In the following simulation results of Figure 4.5 and 4.6, the measuring time window is selected at *200,000us*.

Figure 4.5 shows the network burst loss probability dynamics under the proposed GPMR and AARA. The network's overall burst loss probability is measured at the end of each time window. At the initial stage before time *20s,* the offered traffic load to the network is 200 flows/s per node pair. Then from time *20s* onwards, we increase the offered traffic load to 350 flows/s per node pair to see how the two algorithms react to the change. The initial traffic proportion for each alternative path between the SD pairs is set at a randomly chosen value. Then the two algorithms come into picture to make the appropriate adjustment.

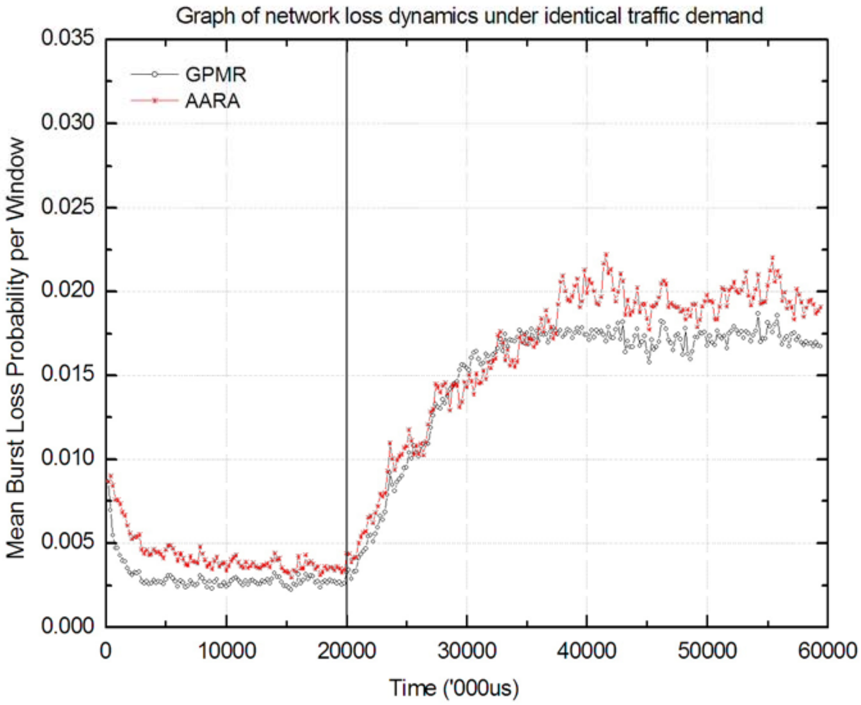

Figure4.5: Graph of network burst loss dynamics

From the figure, we can see that at the initial stage before *20s*, both GPMR and AARA can reduce the congestion effectively and bring down the network burst loss probability in a reasonably short period of time. However, the proposed GPMR out-performs AARA by successfully bringing the burst loss probability down to a lower level which can be seen from Figure 4.5. Then when the traffic load is increased to 350 flows/s at time *20s*, under the proposed GPMR, the network burst loss probability starts to climb up and reaches a reasonably stable level in about 10 seconds' time. It shows good performance in terms of convergence and stability, as well as the ability in adapting to the traffic variations in the network. However, for AARA, the network burst loss probability first climbs up to a level which is higher than GPMR's stabilized loss value. Then it keeps on fluctuating up and down around that burst loss level and cannot arrive at a reasonably converged and stable state even when the simulation lasts until the time of *60s*. Hence, from the above observation, we can see that, in addition to its better burst loss performance, GPMR also achieves better stability and robustness than AARA in adapting to the traffic variations in the network.

In Figure 4.6, we illustrate how the traffic load is distributed after the proposed GPMR starts. For the time before *20s*, the offered traffic load to the network is 200 flows/s per node pair. From the time *20s* onwards which is marked by the solid line in the figure, the offered traffic load is increased to 350 flows/s per node pair. The normalized link load is measured at the end of each time window. We can see that under the proposed GPMR algorithm, the link loads are quickly balanced and they converge to a stable value in a reasonably short period of time. For example, for the initially highly-loaded link 11->3, its link load is quickly brought down to a reasonable and stable level by GPMR to avoid congesting the link. However, for the initially very lightly loaded link 12->9, GPMR offers more load to it to avoid under-utilization of the link. From *20s* onwards where the offered traffic load increases to 350 flows/s, the link loads start to climb up, but stabilize at the new levels in a reasonably short period of time. It shows that the proposed GPMR has good capabilities in adapting to traffic variations in the network so that the routing can quickly arrive at a stable state again.

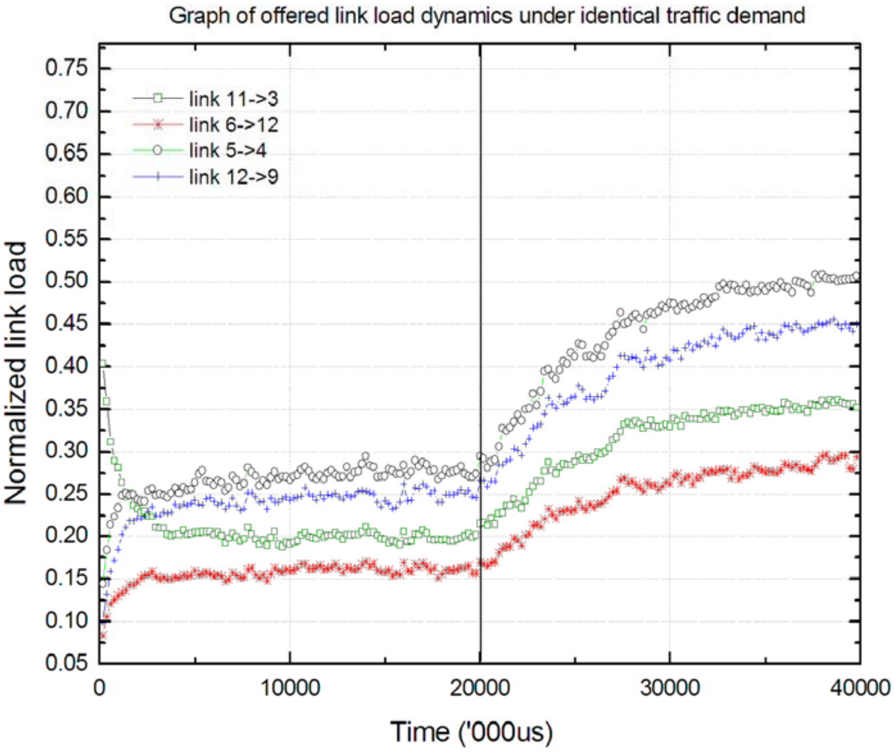

Figure4.6: Offered load of selected links

4.4.2 Non-identical Traffic Demand

In this section, we will investigate the performance of GPMR under non-identical traffic demand. In a non-identical traffic demand, the flow arrival rate for an SD pair is randomly selected from a set of flow arrival rates $\{r_1, r_2, r_3, r_4, r_5\}$ with equal probability. The traffic load is measured as the mean flow arrival rate which is given by the average of the flow arrival rates.

4.4.2.1 Effect of Traffic Loading

Figure 4.7 shows the burst loss probability of the proposed GPMR and AARA for six non-identical traffic demands. Figure 4.8 shows the percentage of burst loss performance improvement for GPMR in comparison with AARA. Figure 4.9 shows mean hop length comparison. From these figures, we make similar observations as in the case of the identical traffic demands.

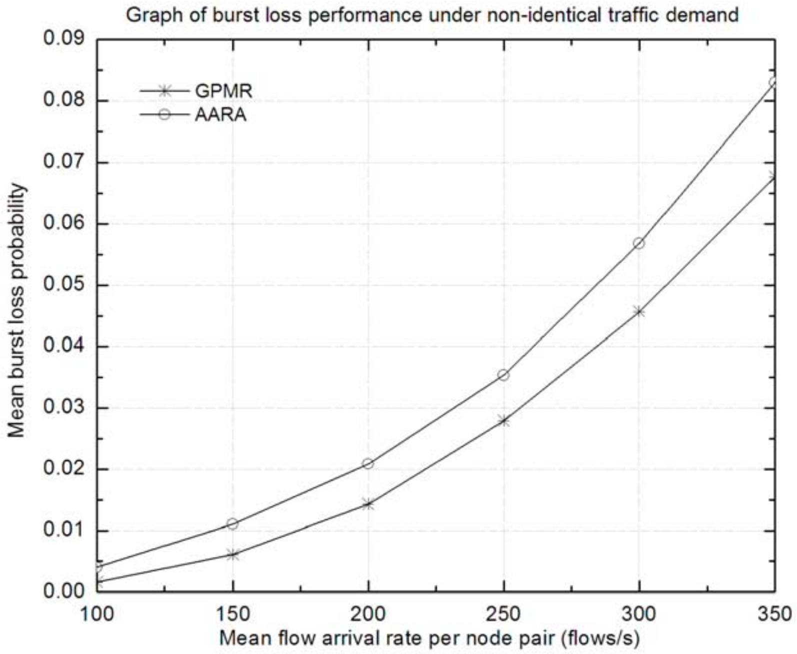

Figure4.7: Graph of burst loss probability under non-identical traffic demands

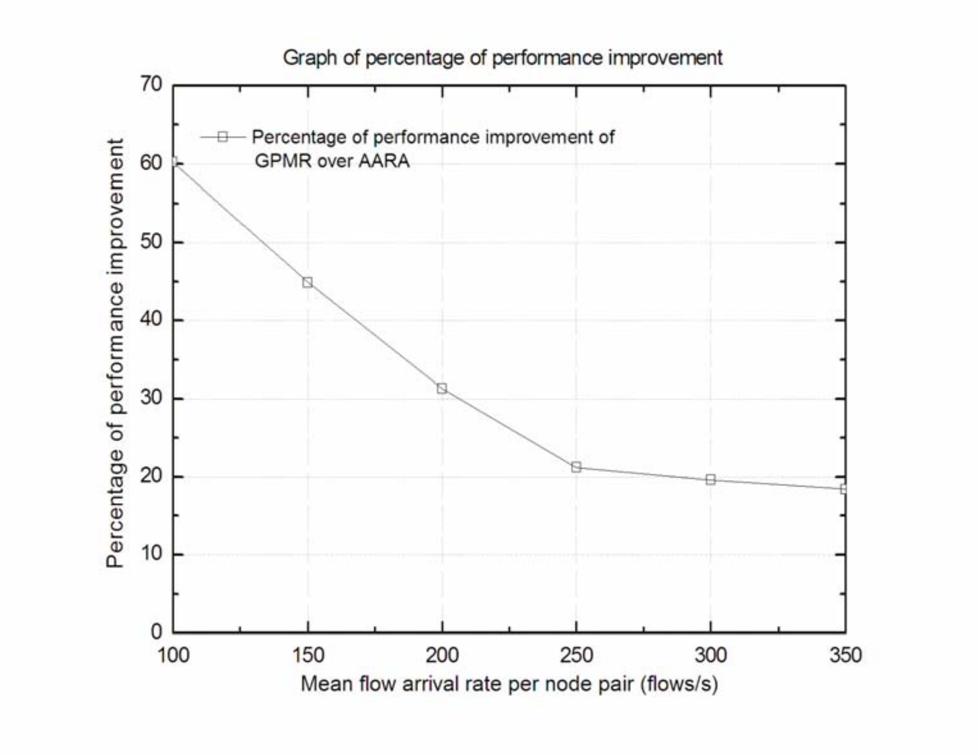

Figure4.8: Graph of percentage of performance improvement for non-identical demands

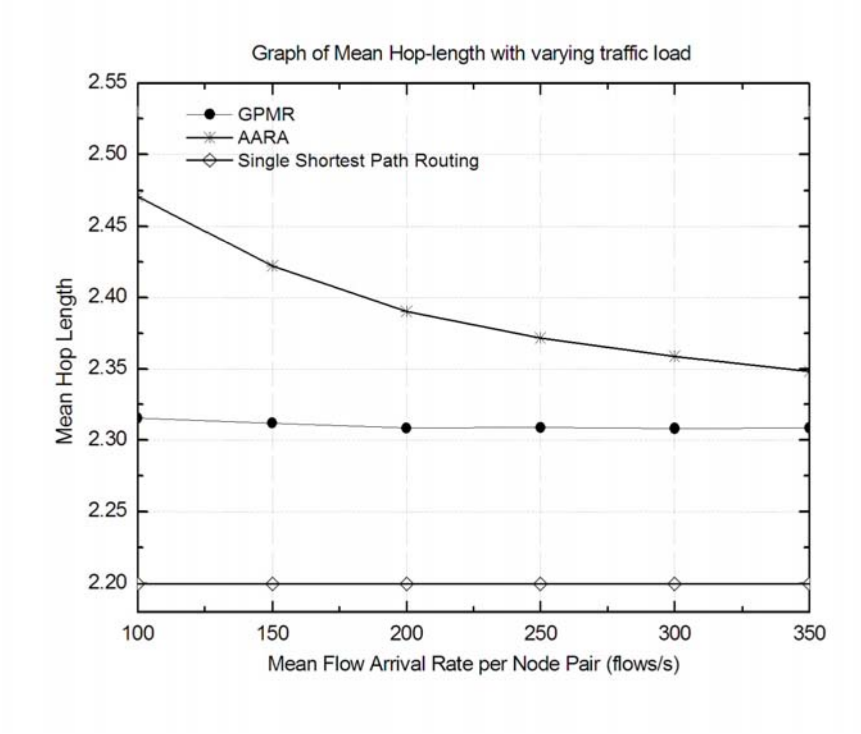

Figure4.9: Graph of mean hop length for non-identical traffic demands

4.4.2.2 Dynamics of the GPMR Algorithm under Non-identical Traffic Demand

In this section, through the simulation results, we would like to show that the proposed GPMR algorithm is also stable and robust under the non-identical traffic demand scenario. In the following simulation results of Figure 4.10 and 4.11, the measurement time window is selected at *200,000us*.

Figure 4.10 shows the network burst loss probability changes under the proposed GPMR and AARA load balancing algorithms. The scenario is similar to that of the identical traffic demand. The network's overall burst loss probability is measured at the end of each time window. The average flow arrival rates vary at *20s* from 200 flows/s to 350 flows/s. The initial traffic proportion for each alternative path between the SD pairs is set at a randomly chosen value. Then the two algorithms come into picture to make the appropriate adjustment. From the figure, we make similar observations as in the case of identical traffic demand except that the proposed GPMR has even better stability and convergence performance over AARA when adapting to the traffic variations in the network. We can see from the figure that for GPMR, after the network burst loss probability climbs up to a new level due to the increase in the offered traffic load, it can stabilize at a relatively stable level within a reasonably short period of time. However, for AARA, the network burst loss probability still keeps on fluctuating up and down even when the simulation time lasts until *55s*, and we can see that the degree of fluctuation is larger than that in the case of identical traffic demands.

Hence, we can see that the GPMR has greater advantage over AARA in adapting to traffic variations under non-identical traffic demand.

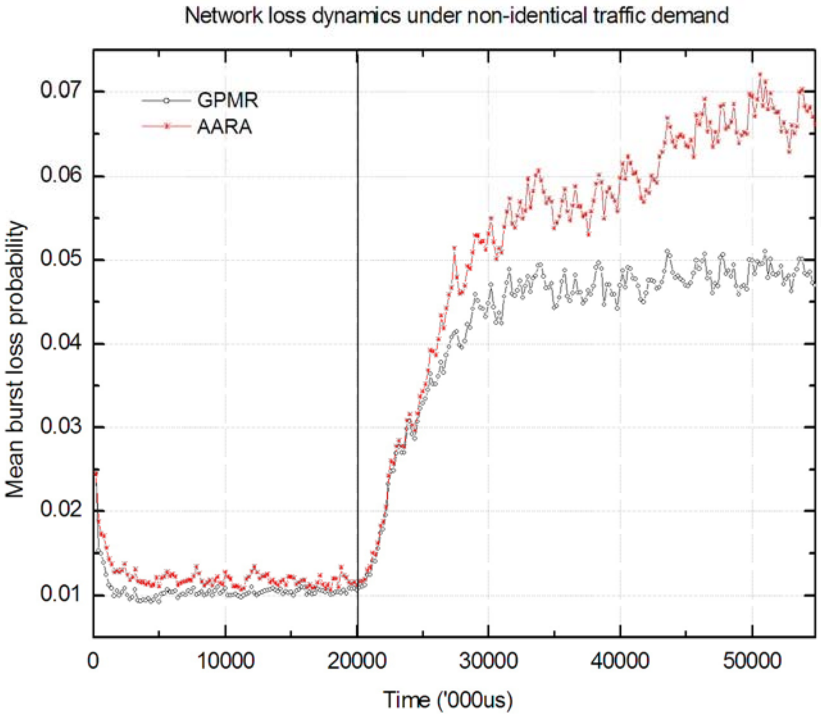

Figure4.10: Graph of network burst loss dynamics under non-identical traffic demands

In Figure 4.11, we illustrate how the traffic load is distributed after the proposed GPMR starts under the non-identical traffic demands. The traffic setting and scenario are the same as in the case of identical traffic demands and we also make similar observations. The proposed GPMR exhibits good capability in adapting to traffic variations even under the scenario of non-identical traffic demands.

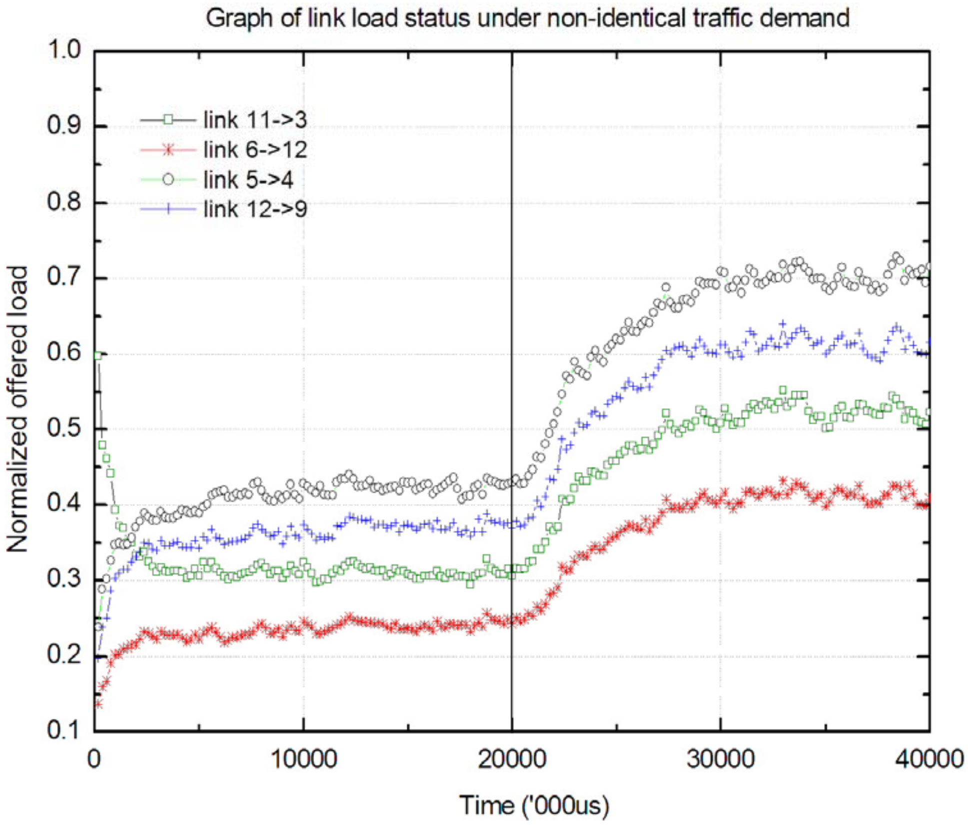

Figure4.11: Offered load of selected links under non-identical traffic demands

4.4.3 Summary of Simulation Results

We now summarize the important observations made from the simulations results.

1. GPMR can considerably improve the overall network performance in terms of burst loss reduction over AARA.
2. GPMR incurs lower processing and signaling overhead than AARA which is reflected as a shorter measured mean hop length traversed by bursts.
3. GPMR exhibits good routing stability and the routing in the network can

converge to a stabilized level in a reasonably short period of time.

4. GPMR performs well under different traffic situations and has good capability in adapting to traffic variations in the network.

Chapter 5
Conclusions

In this thesis, the problem of adaptive traffic distribution in OBS networks has been studied. Firstly, a scheme which dynamically routes the arriving traffic flows in an adaptive and proportional manner in OBS networks is presented. Then in the second part of the thesis, a gradient projection optimization framework based multi-path traffic routing scheme is proposed for OBS networks.

In the first part of the thesis, we have presented an adaptive proportional routing scheme which attempts to route the incoming traffic in OBS networks at the flow level. The proposed scheme avoids the problem of packet out-of-sequence arrival which is a common problem in previous proposed load balancing schemes. In our scheme, flow-based multi-path traffic routing is achieved by the cooperation of four functional units - traffic measurement, traffic assignment, traffic distribution and burst assembly units. We have presented a time-window-based mechanism which works in conjunction with adaptive proportional flow routing. In the time-window-based mechanism, adaptive proportional flow routing operates in cycles of time duration called time windows. From the simulation results, we have shown that our scheme can effectively balance the traffic load and improve the burst loss performance significantly over the equal-proportion and hop-length based flow routing schemes. We have also demonstrated that our scheme is applicable to different traffic scenarios.

Previous proposed load balancing algorithms in OBS networks make their traffic adjustment only based on some heuristic algorithms. Their performance may not be good enough or optimized and there is no guarantee that they can converge to a stable routing state. They may suffer from routing instability problems which are common in link-state based load balancing algorithms. To overcome the above shortcomings, in the second part of the thesis, we have developed an online adaptive multi-path traffic routing scheme in OBS networks. The proposed scheme works in a time-window manner and is based on the gradient projection optimization algorithm. It has several attractive features. First, it achieves very good performance in reducing burst loss and minimizing network congestion in the network. Second, it exhibits good routing stability and is capable of adapting to traffic variations in the network. Last but not least, it uses a simple measurement mechanism which does not incur much signaling and processing overhead. We have demonstrated that our proposed algorithm can effectively distribute the traffic load and further reduce burst loss significantly through extensive simulations. We have also verified that our algorithm can converge to a stable routing state and has good capability in adapting to traffic variations under different traffic loading conditions.

Finally, we present some possible research directions for future investigation.

The multi-path load balancing problem to support multiple classes of services with different QoS requirements is a challenging problem to be studied. It is interesting to achieve multi-path optimal routing and load balancing in the networks while providing differentiated services to different classes of traffic demands, which can be referred to as the differentiated traffic engineering problem. It has been recently studied in IP networks [47] but no work has been done yet in OBS networks. Besides that, buffer management, burst assembly and admission control policies at the edge nodes to implement traffic engineering are also important and interesting problems to be studied in the future.

Bibliography

[1] C. Bracket, “Dense Wavelength Division Multiplexing Networks: Principles and Applications,” *IEEE Journal on Selected Areas in Communications,* vol 8, no. 6, pp 948-964, August 1990.

[2] C. Assi, A. Shami, and M. A. Ali, “Optical Networking and Real-time Provisioning: An Integrated Vision for the Next-Generation Internet,” *IEEE Networks,* vol. 15, no.4, pp. 36-45, July-August, 2001.

[3] C. Siva Ram Murthy and G. Mohan, “WDM Optical Networks: Concepts, Designs, and Algorithms,” *Prentice Hall PTR, NJ,* USA, November 2001

[4] R.C. Alferness, H. Kogelnik, and T.H. Wood, “The Evolution of Optical Systems: Optics Everywhere,” *Bell Labs Technical Journal*, 5(1), Jan-Mar, 2000

[5] J.P. Jue, V. M. Vokkarane, “Optical Burst Switched Networks,” *Springer Science+Business Media, Inc*, Boston, USA, 2005

[6] R. Ramaswami and K. N. Sivarajan, “Optical Networks: A Practical Perspective”, 2nd ed., *Morgan Kaufmann Publishers*, 2002.

[7] I. Chlamtac, A. Ganz, and G. Karmi, “Lightpath Communications: An Approach to High Bandwidth Optical WANs,” *IEEE Transaction on Communications*, vol. 40, no. 7, pp. 1171-1182, Jul 1992

[8] T. S. El-Bawab and J.-D. Shin, “Optical Packet Switching in Core Networks: Between Vision and Reality,” *IEEE Communications Magazine*, vol. 40, no. 9, pp. 60-65, Sep, 2002

[9] M. J. O’Mahony, D. Simeonidou, D.K. Hunter, and A. Tzanakaki, “The Application of Optical Packet Switching in Future Communication Networks,” *IEEE Communication Magazine*, vol. 39, no. 3, pp. 128-135, Mar 2001

[10] C. Qiao and M. Yoo, “Optical Burst Switching- a New Paradigm for an Optical Internet,” *Journal of High Speed Network*, vol.8, no.1, pp69-84, Jan. 1999

[11] T. Battestilli and H. Perros, “An Introduction to Optical Burst Switching,” *IEEE Communication Magazines*, vol. 41, no. 8, pp. S10-S15, Aug, 2003

[12] J.S. Turner, “Terabit Burst Switching,” *Journal of High Speed Networks*, vol.8, no.1, pp.3-16, Jan. 1999

[13] Y. Xiong, M.Vandenhoute, and H.C. Cankaya, “Control Architecture in Optical Burst Switched WDM Networks,” *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 1838-1851, Oct 2000.

[14] J. Y. Wei and R. I. McFarland Jr., “ Just-In-Time Signaling for WDM Optical Burst Switching Networks,” *IEEE/OSA Journal of Lightwave Technology*, vol. 18, no. 12, pp. 2019-2037, Dec. 2000

[15] A. H. Zaim, I. Baldine, M. Cassada, G. N. Rouskas, H. G. Perros, and D. Stevenson, “Jumpstart Just-In-Time Signaling Protocol: A Formal Description Using Extended Finite State Machines,” *Optical Engineering*, vol.42, no.2, pp. 568-585, Feb. 2003

[16] L. Tancevski, te. Al., “A New Scheduling Algorithm for Asynchronous, Variable Length IP Traffic Incorporating Void Filling,” *In Proceedings of OFC*, 1999

[17] Y. Xiong, M. Vandenhoute, and H. C. Cankaya, “Design and Analysis of Optical Burst Switched Networks,” *In Proc. of SPIE*, vol. 3843, no. 10, Sep. 1999

[18] M. Laor and L. Gendel, “The Effect of Packet Re-ordering in a Backbone Link on Application Throughput,” *IEEE Network Magazine*, vol. 16, no.5, pp. 28-36, 2002

[19] K. Thompson, G. J. Miller, R. Wilder, “Wide-Area Internet Traffic Patterns and Characteristics”, *IEEE Networks*, Nov/Dec 1997

[20] G. Thodime, V. M. Vokkarane, and J. P. Jue, “ Dynamic Congestion-based Load Balanced Routing in Optical Burst Switched Networks,” *In Proceedings of IEEE GLOBECOM 2003*, San Francisco, CA. Dec 2003

[21] Y. Li, George N. Rouskas, “Path Selection in Optical Burst Switched Networks”, *In Proceedings of Networking 2005*, vol. 3462, Waterloo, Canada, May 2005

[22] J. Li, G. Mohan, and K. C. Chua, ``Dynamic Load Balancing in IP-over-WDM Optical Burst Switching Networks," Computer Networks Journal, vol. 47, no. 3, February 2005.

[23] D. Bersekas and R. Galleger, “Data Networks,” *Prentice Hall Inc.* 2nd Edition, 1992

[24] X. Wang, H. Morikawa, and T. Aoyama, “Photonic Burst deflection routing protocol for wavelength routing networks,” *SPIE Optical Networks Magazine*,vol3, no.6, pp.12-19, Nov-Dec, 2002

[25] S.Yao, B. Mukherjee, S.J.B. Yoo, and S. Dixit, “All-Optical Packet-Switched Networks: a Study of Contention Resolution Schemes in an Irregular Mesh Network with Variable-Sized Packets,” *Proceedings, SPIE Opticomm 2000,* Dallas, TX, pp 235-246, Oct. 2000

[26] S. Kim, N.Kim, and M. Kang, “Contention Resolution for Optical Burst Switching Networks Using Alternative Routing,” *Proceedings, IEEE International Conference on Communications (ICC),* New York, NY, April-May, 2002

[27] X.Wang, H. Marikawa, and T. Ayoma, “A Deflection Routing Protocol for Optical Bursts in WDM Networks,” *Proceedings, Fifth Optoelectronics and Communications Conference (OECC) 2000*, Makuhari, Jaan, pp. 94-95, July 2000.

[28] C. Hsu, T.Liu, and N. Huang, “Performance Analysis of Deflection Routing in Optical Burst-Switched Networks,” *Proceedings of INFOCOM 2002,* pages 66-73, Jun 2002

[29] F. Farahmand, Q. Zhang and J.P. Jue, “A Feedback-Based Contention Avoidance Mechanism for Optical Burst Switching Networks,” *In Proceedings of Workshop on OBS, Broadnets 2004* San Jose, CA, Oct 2004, .

[30] V. M. Vokkarane, J. P. Jue, and S. Sitaraman, “Burst Segmentation: An Approach for Reducing Packet Loss in Optical Burst Switched Networks,” Proceedings, IEEE International Conference on Communications (ICC) 2002, New York, NY, vol 5. pp. 2673-1677, April 2002

[31] F. Farahnamd and J.P. Jue, “Supporting QoS with Look-ahead Window Contention Resolution in Optical Burst Switched Networks,” *Proceedings, IEEE GLOBECOM 2003,* San Francisco, CA, Dec 2003

[32] V. M. Vokkarane, K. Haridoss, and J. P. Jue, “Threshold-based Burst Assembly burst Traffic in Optical Burst Switched Networks,” *Proceedings, SPIE Optical Networking and Communication Conference (OptiComm) 2002*, Boston, MA, vol. 4874, pp. 125-136, July-Aug. 2002

[33] X. Yu, Y. Chen, and C. Qiao, “A Study of Traffic Statistics of Assembled Burst Traffic in Optical Burst Switched Networks,” *Proceedings, SPIE Optical Networking and Communication Conference (OptiComm) 2002*, Boston, MA, pp. 149-159, July-Aug. 2002

[34] X. Cao, J. Li, Y. Chen and C. Qiao, “Assembling TCP/IP Packets in Optical Burst Switched Networks,” *Proceedings, IEEE GLOBECOM 2002,* vol. 3, pp 2808-2812, Nov. 2002

[35] M. Elhaddad, R. Melhem, T. Znati, and D. Basak, “Traffic Shaping and Scheduling for OBS-based IP/WDM Backbones,” *Proceedings, SPIE Optical Networking and Communication Conference (OptiComm) 2003,* Dallas, TX, vol. 5825, pp. 336-345, Oct. 2003

[36] R. Jain and K.K. Ramakrishnan, “Congestion Avoidance in Computer Networks with a Connectionless Network Layer: Concepts, Goals and Methodology,” *Proceedings, Computer Networking Symposium 1988,* pp. 134-143, Apr. 1988

[37] S. Y. Wang, “Using TCP Congestion Control to Improve the Performance of Optical Burst Switched Networks,” *Proceedings, IEEE International Conference on Communications (ICC) 2003*, vol. 2, pp. 1483-1442, May 2003.

[38] A. Elwalid, C. Jin, S. Low and I. Widjaja, “MATE: MPLS Adaptive Traffic Engineering,” *Proceedings, IEEE INFOCOM 2001*, pp. 1300-1309, 2001

[39] T. Guven, C. Kommareddy, R. J. La, M. A. Shayman, B. Bhattacharjee, “Measurement Based Optimal Multi-path Routing,” *Proceedings, IEEE INFOCOM 2004,* Hong Kong, Mar. 2003

[40] J. C. Spall, “Stochastic Optimization and the Simultaneous Perturbation Method,” *Proceedings, Winter Simulation Conference 1999.*

[41] J. C. Spall “Multivariate Stochastic Approximation using a Simultaneous Perturbation Gradient Approximation,” *IEEE Transaction on Automation and Control,* vol. 37, pp. 332-341, 1992

[42] V. Paxson and S. Floyd, “Wide-Area Traffic: The Failure for Poisson Modeling,” *Proceedings, ACM SIGCOMM 1999*, 257-268, Aug. 2004.

[43] S. Nelakuditi, Z.L. Zhang, R.P. Tsang, and David H.C. Du, “Adaptive Proportional Routing: A Localized QoS Routing Approach” Department of Computer Science, University of Minnesota, *Tech Rep.*, July 2002

[44] Messerli, E. J. (1972), “Proof of a Convexity Property of the Erlang-B Formula,” *Bell Systems Technical Journal,* vol. 51, pg. 951.

[45] J.N. Tsitsiklis and D.P. Bertsekas, “Distributed asynchronous optimal routing in data networks,” *IEEE Trans. Automat. Contr.* Vol. AC-31, no. 4, pp. 325-332, Apr. 1986

[46] Jin Cao, William S. Cleveland, Dong Lin, and Don X. Sun, "The effect of statistical multiplexing on Internet packet traffic: theory and empirical study," Tech. Rep., Bell Labs, 2001.

[47] V. Tabatabaee, B. Bhattacharjee, R. J. La, M. A. Shayman, “Differentiated Traffic Engineering for QoS Provisioning,” *Proceedings, IEEE INFOCOM 2005*, Miami, US, Mar. 2005