DYNAMIC ROUTING AND LOAD BALANCING IN IP-OVER-WDM NETWORKS

LI JING

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2003

Acknowledgements

I would like to thank my supervisors, Dr. Mohan Gurusamy and Dr. Kee Chaing Chua, for their valuable guidance and encouragement, which light the way to the interesting research area for me.

I also would like to thank my parents, who are always besides me no matter when and where. Their endless love gives me the courage to face many difficulties.

Table of Contents

Summary

In this thesis, dynamic routing and load balancing issues in IP/WDM networks are studied. We first investigate the problem of dynamically routing bandwidth-guaranteed LSPs in an integrated IP/WDM network with inaccurate link state information. Then we address the issue of dynamic load balancing in IP-over-WDM optical burst switching networks.

Dynamic routing in an integrated IP/WDM network has been receiving more attention with the emergence of the GMPLS mechanism. Since dynamic integrated routing takes into consideration the network topology and resource usage information at the IP and optical layers, it makes better use of the network resources. This is a topic which has not been studied extensively. We consider dynamic integrated routing of bandwidth-guaranteed LSPs where the link state information is updated and the routing mechanism uses this information to select paths for each LSP request. In an integrated IP/WDM network, the link state information includes not only the residual bandwidth of a logical link (IP layer) but also the free wavelengths on a physical link (optical layer). A central routing server is assumed if real time update of the link state information is needed to achieve accurate information. Such routing schemes based on accurate link state information are therefore suitable for only small networks and are not scalable to large networks. From the practical point of view, in order to avoid extensive overheads in advertising and processing the link state

information, a threshold-trigger based link state update model is considered. This leads to inaccuracies in the link state information. Consequently, uncertainties arise from the inaccuracies—bandwidth and wavelength inaccuracy. In order to minimize the impact of the inaccurate information so that the blocking probability as well as setup failures are reduced, the routing problem needs to take into consideration the uncertainties of link state parameters. Based on the threshold-triggered update scheme, we present a probabilistic method to model the uncertainties in the link state parameters. We then define a cost function that reflects the uncertainties which are considered as a cost metric. Depending on the different cost metrics chosen to be optimized, we propose two routing algorithms considering the uncertainties in the link state parameters: *most probable path* (MPP) and *minimum hops most probable path* (MHMPP). MPP uses the uncertainties as the cost metric and tries to find a path which is the most probable to satisfy the bandwidth requirement of the LSP request. MHMPP considers both hops and the uncertainties as the cost metrics and tries to find a path which is the most probable path among all the shortest-hop paths.

The explosive growth of Internet traffic and the advances in WDM technology have led to IP-directly-over-WDM optical Internet networks. In order to efficiently utilize the bandwidth in the optical layer, *optical burst switching* (OBS) is considered as a promising switching technology. Load balancing is an important issue in OBS networks due to the unique features of OBS networks such as no electronic buffering and no/limited optical buffering. We propose a load balancing scheme based on adaptive alternate routing whose objective is to reduce burst loss through load balancing. The key idea of adaptive alternate routing is to reduce network congestion by adaptively balancing the load between two pre-determined link-disjoint alterna-

tive paths based on the measurement of the impact of traffic load on each of them. We present a time-window-based measurement mechanism in conjunction with the adaptive alternate routing algorithm. Also we present two alternative path selection schemes based on shortest-hop path routing and widest-shortest-hop path routing, respectively.

List of Figures

List of Symbols

ATM: asynchronous transfer mode

AARA: adaptive alternate routing algorithm

CR-LDP: constraint-based routing label-distributed protocol

CSPF: constrained shorted path first

FDL: fiber delay line

FFUC: first fit unscheduled channel

FAR: flow arrival rate

GMPLS: generalized multi-protocol label switching

IS-IS: intermediate system to intermediate system

IETF: Internet Engineering Task Force

JET: Just-Enough-Time

LSR: label switched router

LSP: label switched path

LMP: link management protocol

LSA: link state advertisement

LAUC: latest available unscheduled channel

LAVF: latest available void filling

LAUC-VF: latest available unused channel with void filling

MPLS: multi-protocol label switching

MOCA: maximum open capacity routing algorithm

MATE: multipath adaptive traffic engineering

MHMPP: minimum hop most probable path

MPP: most probable path

MH-A: minhop-accurate

MH-I: minhop-inaccurate

NNI: network-network interface

OADM: optical add/drop multiplexer

OXC: optical cross connect

OSPF: open shortest path first

OEO: opto-electronic-opto

OCS: optical circuit switching

OBS: optical burst switching

OPS: optical packet switching

QOS: quality of service

RSVP: resource reservation protocol

RWA: routing and wavelength assignment

RTT: round trip time

SONET/SDH: synchronous optical network/synchronous digital hierarchy

SLA: service level agreement

SHPR: shortest-hop path routing

SPR: shortest path routing

SAR: static alternate routing

TE: traffic engineering

UNI: user-network interface

VPN: virtual private network

VF: void filling

WDM: wavelength division multiplexing

WSHPR: widest-shortest-hop path routing

Chapter 1

Introduction

1.1 Background

Recently, there has been a dramatic increase in data traffic, driven primarily by the explosive growth of the Internet as well as the proliferation of *virtual private networks* (VPNs). The demand for bandwidth is growing at a rapid speed and the data traffic is expected to dominate the voice traffic in the near future. The emergence of *wavelength-division multiplexing* (WDM) transmission technology is catering to the massive bandwidth requirement in a cost-effective way. WDM eliminates the electronic bottleneck by dividing the optical transmission spectrum into a number of non-overlapping wavelength channels, each operating at the rate of a few gigabits per second [1], [2].

The early deployment of WDM technology was in a point-to-point manner to ease fiber exhaustion. As more advanced systems, such as *optical add/drop multiplexers* (OADMs) and *optical cross-connects* (OXCs) (capable of routing and wavelength switching), mature, WDM has become a network-level technology.

OADMs and OXCs are introduced into the WDM networks to add/drop traffic (wavelengths) at intermediate points along the route between the end points [3]. A 2-wavelength OADM as shown in Fig. 1.1 can be realized using a demultiplexer, 2×2 switches — one switch per wavelength , and a multiplexer. If a 2×2 switch

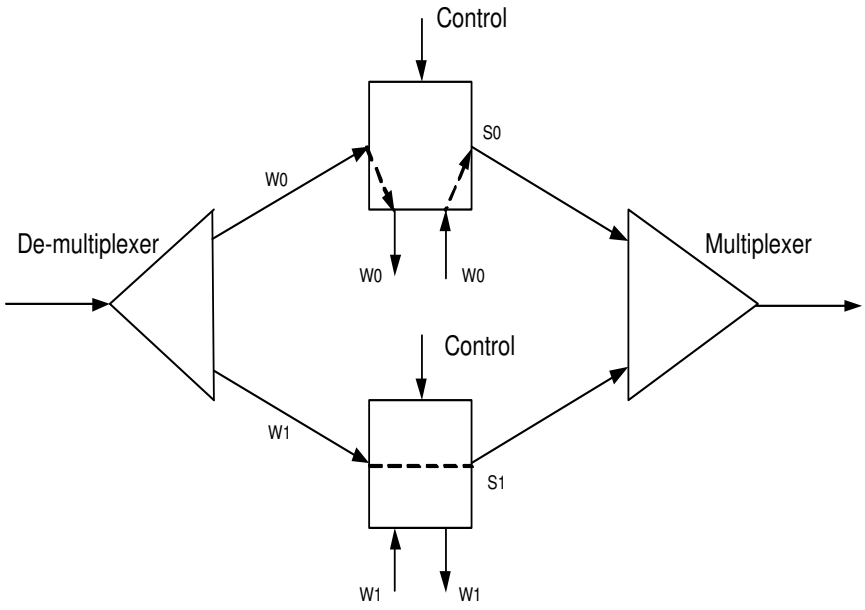

Figure 1.1: Optical add/drop multiplexer.

($S1$ in the figure) is in 'bar' state, then the signal on the corresponding wavelength passes through the OADM. If the switch ($S0$ in the figure) is in 'cross' state, then the signal on the corresponding wavelength is 'dropped' locally, and another signal can be 'added' on to the same wavelength at the OADM location. OADMs are commonly used in networks that follow the logical-ring structure. Functionally, OXCs are quite similar to OADMs, differing mainly in the ability to connect any input wavelength channel from an input fiber to any one of the output fibers. Fig. 1.2 shows a 2×2 2-wavelength OXC which can be realized by demultiplexers, optical switches, and multiplexers.

With the emergence of OADM and OXCs, one can build a flexible multi-point WDM optical network. An attractive WDM optical network architecture widely studied is wavelength-routed WDM network, which are built on the concept of circuit switching technology. A wavelength routed network consists of OXCs interconnected by fiber links in a general mesh topology. Lightpaths are set up between

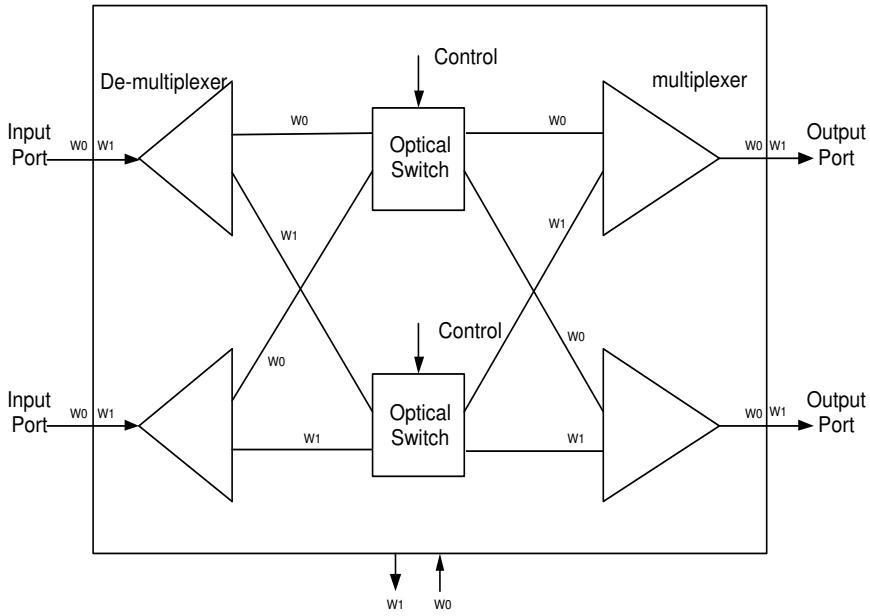

Figure 1.2: Optical cross-connect.

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 network/synchronous digital hierarchy* (SONET/SDH). A lightpath is an all-optical communication path between two nodes without requiring any *optical-electronic-optical* (conversion) in between. The setup of a lightpath is subject to the *wavelength continuity constraint*, i.e., the same wavelength must be used on all links along the route. This constraint is relaxed if wavelength converters are placed at the OXCs.

Today's data networks typically have four layers: IP for carrying applications and services, ATM for traffic engineering, SONET/SDH for transport, and WDM for capacity [1]. This architecture has drawbacks such as inscalability and cost-ineffectiveness. Any one layer can limit the scalability of the entire network, as well as add to the cost of the entire network. As a result, there arises a need for a simpler and cost-effective network that will transport a wide range of data streams and very

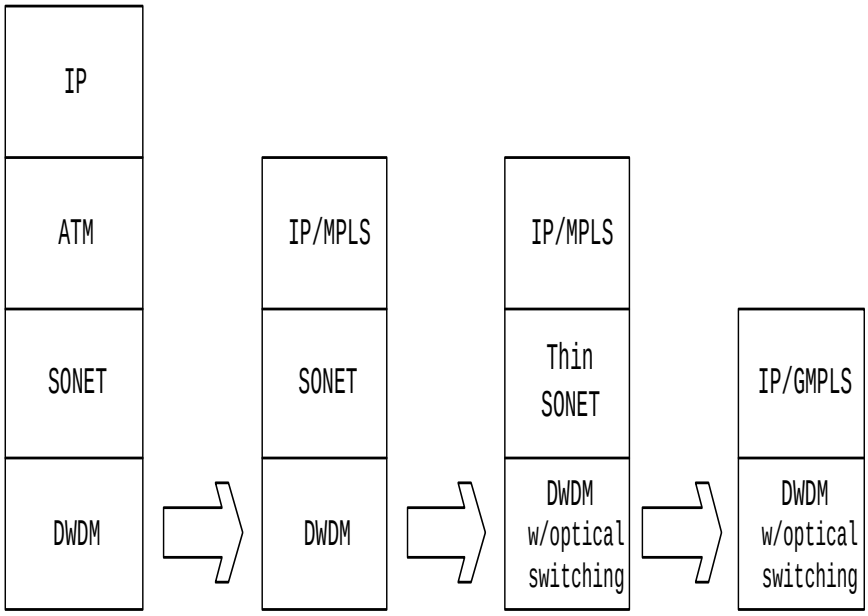

Figure 1.3: Merging network layers.

large volumes of traffic. Furthermore, due to the predominance of IP-based traffic a simpler IP-directly-over-WDM architecture as shown in Fig. 1.3 will allow bypassing the SONET/SDH and ATM layers.

Once the view about network topology has changed, one will have to re-think routing as well [2]. For example, initially there was fixed routing over fixed circuits (PSTN), and next came dynamic routing over fixed circuits (IP-over-SONET). Subsequently there was a move towards dynamic routing over virtual circuits (i.e., IP over ATM) [5]. Now, with recent advances in *multi-protocol label switching* (MPLS), we have label swapping over virtual circuits. Furthermore, industry organizations are now extending the MPLS framework called *generalized multiprotocol label switching* (GMPLS) to support not only devices that perform packet switching (routers), but also those that perform switching in time (SONET), wavelength (OXCs), and space. Therefore it is most likely that the next evolution will be label swapping over dynamic circuits or lightpaths [4], [6], [7], [8], [9], [10].

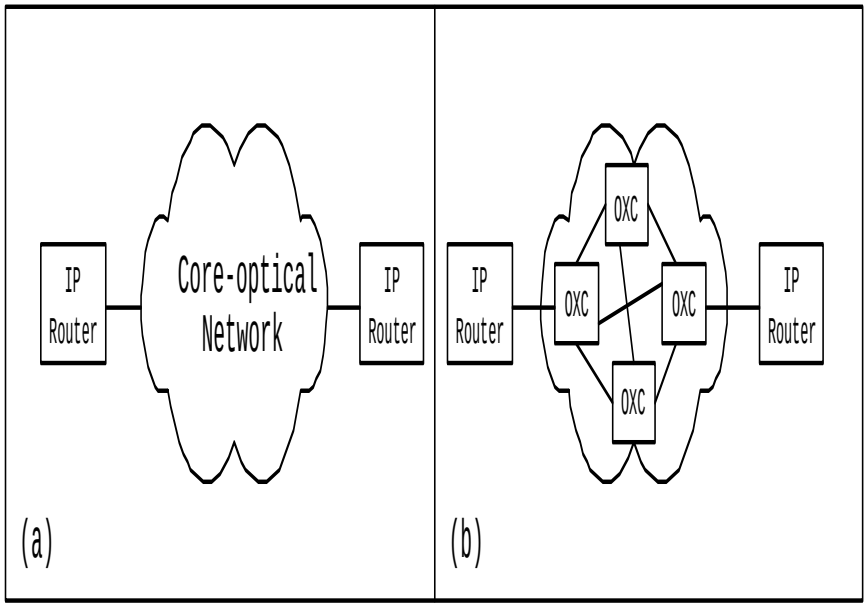

Figure 1.4: Network level abstraction: (a) overlay (b)peer.

1.2 IP-over-WDM Network Architecture

The IP-over-WDM architecture may use an overlay model or a peer model. In the overlay model (Fig. 1.4(a)), there are two separate control planes: one operates within the optical domain, and the other between the optical domain and the IP domain (called the *user-network interface*, UNI). The IP domain acts as a client to the optical domain. The IP/MPLS routing and signaling protocols are independent of the routing and signaling protocols of the optical layer. In this model, the client routers request lightpaths from the optical network through the UNI with no knowledge of the optical network topology or resources. Likewise, the optical network provides point-to-point connections to the IP domain. The overlay model may be statically provisioned using a network management system or may be dynamically provisioned.

In the peer model(Fig. 1.4(b)), a single instance of the control plane spans an administrative domain consisting of the optical and IP domains. Thus, the OXCs

are treated just like any other routers (IP/MPLS routers and OXCs act as peers) and there is only a single instance of routing and signaling protocols spanning them. Thus, from a routing and signaling point of view, there is no distinction between the UNI and the NNI (*network-network-interface*). This allows the IP routers to have full access to the topology of the optical network [2].

1.3 An Overview of GMPLS Framework

In a traditional IP network, each IP packet is transmitted across the network through hop-by-hop routing and forwarding. This kind of layer-3 packet forwarding is slow due to the long packet processing time. The *multiprotocol label switching* (MPLS) framework enables layer-2 forwarding and thus speeding up the IP packet forwarding. In IP/MPLS networks, a router capable of MPLS is called a *label switched router* (LSR).

In IP/MPLS, the control plane and data plane are separated. A label containing the forwarding information is separated from the content of the IP header. An LSR forwards the IP packet using the label carried by the packet. This label, combined with the port on which the packet was received, is used to determine the output port and outgoing label for the packet. Therefore, the MPLS control plane operates in terms of label swapping and forwarding paradigm abstraction.

Constraint-based routing is a significant feature of MPLS which allows to explicitly route and create *label switched paths* (LSPs). Constraint-based routing is a combination of extensions to existing IP link-state routing protocol (e.g., *Open Shortest Path* (OSPF) and *Intermediate System to Intermediate System* (IS-IS)) with the *Resource Reservation Protocol* (RSVP) or the *Constraint-Based Routing Label-*

Distributed Protocol (CR-LDP) as the MPLS control plane, and the *Constrained Shortest-Path-First* (CSPF) heuristic. The extensions of OSPF and IS-IS allow nodes to exchange information about the network topology, resource availability and even policy information. This information is used by the CSPF [11] heuristic to compute paths subject to specified resource and/or policy constraints. Then, either RSVP or CR-LDP is used to establish the label forwarding state along the routes computed by a CSPF-based algorithm. This creates the LSPs. The MPLS data plane is used to forward the data along the established LSPs.

The Internet Engineering Task Force (IETF) is taking efforts to standardize GMPLS as the common control plane not only in the IP domain but also in the optical domain [1], [12], [13], [14]. Some modifications and additions to the MPLS routing and signaling protocols required in support of GMPLS are summarized as follows:

1. *Link Management Protocol* (LMP) addresses the issues related to link management in optical networks using photonic switches.
2. *Enhanced OSPF/IS-IS* routing protocols advertise the availability of optical resources in the network.
3. *Enhanced RSVP/CR-LDP* signaling protocols for traffic engineering purposes allow an LSP to be explicitly specified across the optical network.

1.4 IP/WDM Routing

In an IP/WDM network as shown in Fig. 1.5, OXCs are interconnected by fiber links and IP routers are optionally connected to OXCs through wavelength ports comprising optical transmitters and receivers. A lightpath originating and terminating at IP

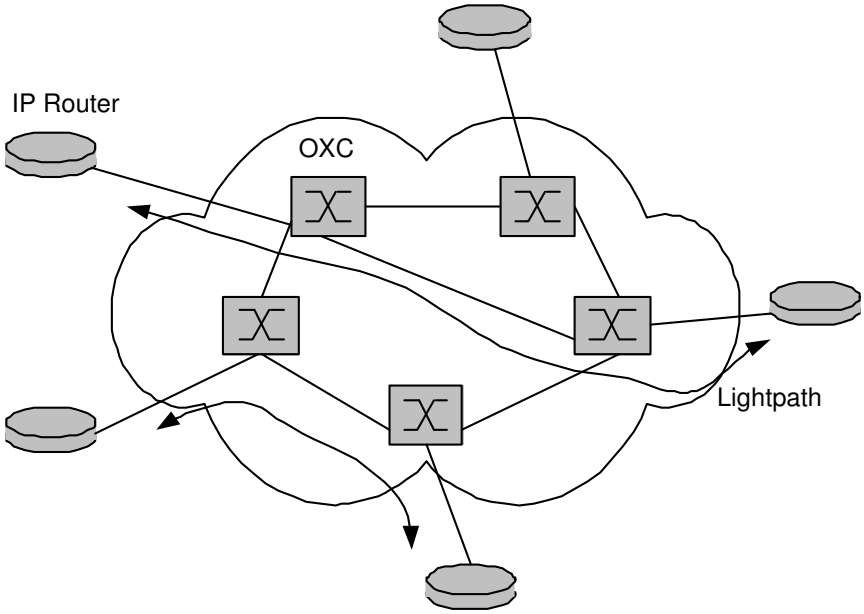

Figure 1.5: An example of IP/WDM network.

routers is subject to the wavelength continuity constraint. If a sequence of more than one lightpath is required to transmit the message from an ingress router to an egress router, optical switching occurs within a lightpath and *opto-electronic-opto* (O-E-O) switching takes place between two consecutive lightpaths. For the routing problem in such a network, two approaches can be used. In the first approach, the routing in the optical layer is solved apart from the IP layer routing. The second approach is to develop an integrated IP/WDM solution that simultaneously addresses the routing issue in both the IP and WDM networks.

1.4.1 Separate Routing for IP and WDM Networks

In this approach, routing in IP-over-WDM network has been separated into routing at the IP layer taking only IP layer information into account, and wavelength routing at the optical layer taking only optical network information into account. Routing solutions, such as OSPF, have already been implemented in IP routers.

In this 'overlay' model, the optical layer acts like the server and the IP layer acts like the client. The IP layer treats a lightpath (in the optical layer) as a link between two IP routers. The topology perceived by the IP layer is the virtual topology wherein the IP routers are interconnected by lightpaths. The IP layer routing is running on this virtual topology. On the other hand, routing in the optical layer establishes lightpath connections on the physical topology. The optical layer manages wavelength resources and chooses the route and wavelength for each of the lightpaths in an optimum way. The two layers may interact and exchange information through the UNI to attempt performance optimization globally.

1.4.2 Integrated Routing for IP/WDM networks

In this approach, the IP and optical layers provide a single unified control plane for efficient management and usage of the network resources, which corresponds to the 'peer' model. The topology perceived by the network nodes (either OXCs or IP routers) is the one where physical fiber links and *logical links* (lightpaths) co-exist. Such a topology contains complete information with regards to wavelength usage on physical links and bandwidth usage on logical links in the network. Integrated routing runs on such a topology to route lightpaths between two network nodes such as IP routers established via the OXCs. Since integrated routing takes into account the combined knowledge of resource and topology information in both the IP and optical layer, it can manage resources more dynamically and respond faster to the traffic changes than the separate routing.

In IP/MPLS networks, LSPs are established between two IP routers, providing a notion of connection-oriented service. Here, network resource information is updated

and the network state is maintained periodically by the routers. An ingress router can use the information to determine routes for explicit routing of LSPs. Recently, proposals have been made to use OSPF-like link state discovery and enhanced MPLS signaling (RSVP or LSP), in the optical networks, to dynamically set up lightpaths [6]. Also, proposals have been made to define a standard interface permitting routers to exchange information and to dynamically request lightpaths from the optical network [15]. This makes it feasible to consider integrated routing in IP/WDM networks, wherein sub-lambda LSPs (IP-LSPs or LSPs) are routed over a sequence of lambda-LSPs (lightpaths) to carry IP traffic. Routing of a sub-lambda LSP may require openning up a new lambda-LSP in addition to the existing ones for better performance and resource utilization. An existing lambda-LSP may be removed when it no longer carries any sub-lambda LSP.

1.4.3 Static versus Dynamic Traffic Demand

The connection requests (traffic demand) can be either static or dynamic. In case of a static traffic demand, connection requests are known a priori. The demands may be specified in terms of source-destination pairs. These pairs are chosen based on an estimation of long term traffic requirements between the node pairs. The objective is to assign routes and wavelengths to all the demands so as to minimize the number of wavelengths used.

In case of a dynamic traffic demand, connection requests arrive to and depart from a network one by one in a random manner. The lightpaths once established remain for a finite time. It may become necessary to tear down some existing lightpaths and establish new lightpaths in response to changing traffic patterns. When a new request

arrives, a route and wavelength need to be assigned to the request with the objective of maximizing the number of connection requests honored (equivalently minimizing the number of connection requests rejected).

1.4.4 Topology and Resource Discovery

In integrated IP/WDM networks with GMPLS capabilities, dynamic integrated routing is allowable. To support dynamic integrated routing, the information about topology and resource usage of all the links in a network should be determined and advertised to the whole network to facilitate the path selection. With the support of GMPLS, extensions to the existing routing protocols such as OSPF allow the exchange of topology and resource usage information among network nodes (IP-MPLS routers and OXCs) in IP-over-WDM networks. For the IP layer, OSPF extensions can be used to distribute bandwidth usage information. For the optical layer similar extensions can be used to distribute wavelength usage information for each link. This information is advertised to the whole network by the opaque *link state advertisements* (LSAs) by using OSPF extensions. The link state information is stored in a link state database at each node, and based on this the source node of a connection request can compute an explicit route by using constraint-based routing schemes. Then, by using the extensions of signaling protocols such as RSVP/LDP, signaling messages (setup messages) are sent along the explicit route to configure the intermediate nodes so that the required resources to support the connection request are reserved. Similarly, signaling messages are sent to release the resources when a connection terminates.

1.5 WDM Switching Technologies

IP-over-WDM networks may use *optical circuit switching* (OCS), *optical burst switching* (OBS), or *optical packet switching* (OPS) technology. The OCS technology is mature and WDM networks based on OCS technology use lightpaths as the optical circuits. Being a circuit, a lightpath does not use statistical multiplexing, which results in inefficient use of resources.

WDM technology is evolving into OBS and OPS. These technologies are expected to support direct integration of IP and WDM layers in the future. In OPS networks, the basic switching entity is a packet. Here, the header and payload are sent together. Upon reaching a node, the header is extracted and processed electronically. The payload is optically delayed by using *fiber delay lines* (FDLs) and then optically switched from the input port to the selected output port. Apart from speeding up the packet switching, optical packet switching supports statistical multiplexing of packets onto WDM wavelength channels. This results in improved bandwidth utilization. However, OPS has the drawbacks such as the need of synchronization of packets and the expensive cost of the switching hardware. Furthermore, since the switching entity operates on a per-packet basis, bottlenecks of electronic processing of the header are introduced into the network.

OBS appears as the promising solution to circumvent the limitations of OPS while keeping the advantage of statistical multiplexing. OBS combines the advantages of optical circuit switching and optical packet switching. There is no need for buffering and electronic processing of data in burst switching. At the same time, optical burst switching ensures efficient bandwidth utilization on a fiber link as in packet switching

by reserving bandwidth on a link only when data is actually required to be transferred through the links.

1.6 Optical Burst Switching

Recently, OBS as a new optical switching technology is receiving more attention for building terabit optical routers and realizing IP-over-WDM [16]. In OBS networks, the basic switching entity is a burst which can be thought of as a large-container containing a number of IP packets with common ingress and egress edge nodes. A block diagram of an OBS network is shown in Fig. 1.6, which consists of optical core routers and electronic edge routers connected by WDM links. Packets are assembled into data bursts at the network ingress nodes and disassembled back into packets at the network egress nodes. A data burst is switched through the network all-optically along a path on *data wavelength channels* which are dedicated to data bursts. A control message (or header) is transmitted on a separate wavelength called *control wavelength channel* ahead of the data burst by an *offset* time to ensure sufficient time for header processing at the intermediate nodes. The header is electronically processed to schedule a data channel for the associated data bursts. This coupled-overlay architecture ideally combines the mature electronic control technologies and promising optical transport technologies.

Several burst switching protocols such as the *Just-Enough-Time* (JET) protocol have been proposed in the literature [17]. In JET protocol, a wavelength on an outgoing link is reserved for a data burst for a fixed duration specified by the corresponding control packet. The source node first sends a header on a control channel. It then sends the corresponding data bursts on a data channel with a time delay equal to

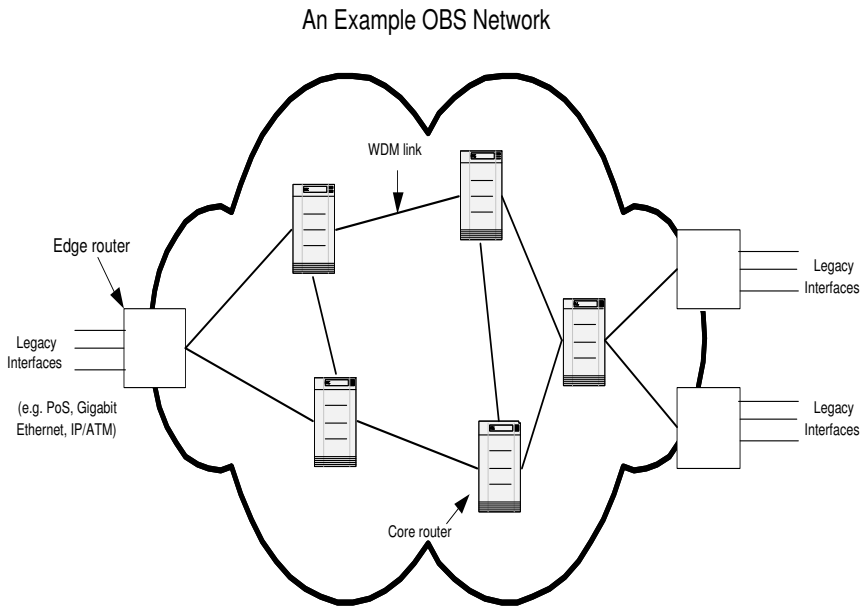

Figure 1.6: An optical burst switching network.

the burst offset time. When the header reaches a node, it reserves a wavelength on the outgoing link for the duration of the data burst starting from the time of arrival of the data burst. The offset time is chosen such that when the data burst arrives at a node, the reservation has already been made and a wavelength on the outgoing link is readily available for onward transmission. Therefore, a data burst needs not be buffered at a node avoiding the need for FDL buffers. Also, there is no bandwidth wastage as it is reserved for the duration of the data burst only.

When a control packet arrives at a node, a *wavelength channel scheduling* algorithm is used to determine the wavelength channel on an outgoing link for the corresponding data burst. The information required by the scheduler such as the arrival time of the data burst and its duration are obtained from the header. The scheduler keeps track of the availability of time slots on every wavelength channel. It selects one among several idle channels for the data burst. If FDLs are available at the node, the scheduler selects one or more FDLs to delay the data burst, if neces-

sary. Several scheduling algorithms have been proposed in the literature to achieve a high bandwidth utilization, such as *First Fit Unscheduled Channel* (FFUC), *Latest Available Unscheduled Channel* (LAUC), and *Latest Available Void Filling* (LAVF) [18], [19], [20].

Contention is considered a major problem in OBS networks since it directly influences the burst loss performance. Contention occurs when several bursts contend for the same data channel at the same time and the contended bursts except one are dropped. In the literature, several issues such as data channel scheduling, offset time management and contention resolution have been extensively studied. The common objective of these issues is to reduce burst loss caused by contention.

1.7 Contributions

In this thesis, we address the problems of dynamic routing and load balancing in IP-over-WDM networks. In the first part, we investigate the problem of dynamically routing bandwidth-guaranteed LSPs in an integrated IP-over-WDM network with inaccurate link state information. To select a good path, a routing algorithm needs up-to-date link state information. This leads to excessive update overhead and scalability problems. In real networks, from the practical point of view, in order to avoid extensive overhead of advertising and processing link state information, updates need to be made periodically or based on a threshold trigger. This leads to inaccuracies in the link state information. We consider the routing problem taking into consideration the uncertainty of link state parameters arising due to the wavelength inaccuracy in addition to bandwidth inaccuracy. Based on the threshold-triggered update scheme, we present a probabilistic method to model the uncertainty of link state parameters.

We then define a cost function reflecting the uncertainty. Depending on different cost metrics chosen to be optimized, we propose two routing algorithms considering the uncertainty of link state parameters. The objective is to minimize the impact of inaccurate information so that the blocking probability as well as setup failures are reduced. We use various performance metrics such as the total blocking probability, blocking probability due to setup failures, blocking probability due to routing failures, bandwidth update frequency, and wavelength update frequency to evaluate the effectiveness of the proposed algorithms. Through extensive simulation experiments, we show that our algorithms can significantly reduce the impact of inaccurate link state information and perform very well.

In the second part, we deal with the problem of dynamic load balancing in IP-over-WDM OBS networks using adaptive alternate routing. Contention is a major problem in OBS networks since it directly influences the burst loss performance. To date, most reported works use burst-centric approaches to deal with the contention problem [16], [21],[22], [23] . However, from the whole network point of view, contention can be reduced by avoiding network congestion through load balancing. Load balancing is an important traffic engineering issue in OBS networks. This is because the lack of optical memory devices renders contention resolution schemes used in traditional IP networks such as buffering and deflection routing inappropriate. Besides the dearth of load balancing mechanisms, another limitation in OBS networks is the use of fixed shortest path routing. Multiple paths may exist between every *ingress-egress* (IE) node pair. Fixed shortest path routing fails to take advantage of these multiple paths, thus causing the network to operate inefficiently. We propose an adaptive alternate routing based load balancing scheme whose objective is to reduce burst

loss through load balancing. The key idea of adaptive alternate routing is to reduce network congestion by adaptively balancing the load between two pre-determined link-disjoint alternative paths based on the measurement of the impact of traffic load on each of them. Through extensive simulation experiments for different traffic scenarios, we show that the proposed dynamic load balancing algorithm outperforms the shortest path routing and static alternate routing algorithms.

1.8 Organization of the Thesis

The thesis is organized into five chapters.

In this introductory chapter, we have given a overall picture of IP-over-WDM networks. Specifically, a brief review of the network architecture, GMPLS support, switching technologies, routing solutions, and OBS technology in IP-over-WDM networks have been provided. Also, we have given a brief introduction of our contributions in this thesis.

Chapter 2 reviews the earlier work on dynamic routing in IP/WDM networks. Works related to load balancing in IP/MPLS networks and contention problem in OBS networks are presented.

In chapter 3, we present the proposed dynamic routing algorithm in integrated IP-over-WDM networks with inaccurate link state information. A graph representation of the integrated IP/WDM network is presented and the bandwidth and wavelength update models developed by us are explained. Also the routing algorithms proposed by us are described. Simulation results are then presented and discussed.

The load balancing issue in WDM-based OBS networks is studied in chapter 4.

The details of the proposed load balancing algorithm based on adaptive alternate routing are presented, followed by a discussion on the results of the performance study.

In chapter 5, the work in this thesis is summarized.

Chapter 2

Related Work

In this chapter, we describe earlier works on the routing and load balancing problems in IP and WDM networks. In particular, we look into several issues: lightpath routing, integrated routing of LSPs, routing with inaccurate information, non-real time update model in WDM networks, load balancing in IP/MPLS networks, and contention in OBS networks.

2.1 Lightpath Routing in WDM Networks

Lightpath routing in WDM networks refers to the *routing and wavelength assignment* (RWA) problem. RWA solves the problem of selecting a physical route and wavelength for a lightpath connection request. Typically, connection requests may be of three types: static, incremental and dynamic [2]. With static traffic, the entire set of connections is known in advance, and the lightpaths are set up on a global basis to optimize use of the network resources. The RWA problem can be formulated as a mixed-integer linear program [24]. In the incremental case, the connection requests arrive sequentially and the lightpaths remain in the network indefinitely.

For dynamic traffic, a lightpath is set up as the connection request arrives, and the lightpath is released after some finite random amount of time. The objective is to set up lightpaths in a manner that minimizes connection blocking.

In the literature, the dynamic routing problem in the optical layer (corresponding to the dynamic traffic case) to route lightpaths has been studied extensively [3], [25], [26], [27]. In [28] a protection scheme that considers routing at the optical layer and client layer is proposed. However, the routing instances at both layers are separate.

A decentralized path selection with on-demand wavelength channel provisioning in WDM networks with multiple constraints such as transmission degradation and delay is presented in [29]. In this work, to select a wavelength path satisfying the given constraints, the ingress node *floods* the network by sending the *wavelength probe* messages. The path is determined based on the availability of a local, rather than the accurate or inaccurate global, network state information. However, it does not use the two-layer integrated routing. Further, the flooding mechanism to obtain the path information for each service request leads to the scalability problem. This problem becomes even worse in integrated IP/WDM networks when the routing constraint is bandwidth requirement since the LSP requests are more frequent and dynamic when compared to the lightpath requests and the granularity of the LSP bandwidth requests is a fraction of a wavelength.

2.2 Integrated Routing of LSPs in IP/WDM Networks

Recently, the problem of dynamic integrated routing of bandwidth-guaranteed LSPs in integrated IP/WDM networks taking into account the link capacities and ingress-egress node information has been considered in [30]. The bandwidth-guaranteed LSPs considered here are MPLS LSPs. The bandwidth requirement of each LSP is some fraction of the capacity of a wavelength. The bandwidth may be used as the *quality of service* (QoS) metric; if any other metric such as delay is specified by the

service level agreement (SLA) then it is assumed to be translated into an effective bandwidth requirement (with the queueing delay primarily restricted to the edge router and with a predictable or negligible queuing delay at the core routers). Such a delay-to-bandwidth translation has also been used for the QoS routing problem in IP networks [32]. Algorithms for routing bandwidth guaranteed LSPs considering only the IP layer topology and resource information have been extensively studied in [31], [32], and [33]. Different from lightpath routing, which is independent of the IP layer routing, and LSPs routing in the IP layer, integrated routing of LSPs in a IP/WDM network integrates the IP layer and the optical layer routing instances into a single one. And routing takes into account the combined topology and resource usage information at the IP and optical layers.

In [30], an expanded network model that allows for the representation of different wavelengths carried by each physical optical link is introduced. Such a model enables the direct application of Dijkstra's algorithm on the network graph. Also, lightpaths is modelled using *cut-through arcs* that replace traversed physical links. Thus, the topology of the graph is dynamic, and may change with each accepted request.

In [30], the Maximum Open Capacity Routing Algorithm (MOCA), which determines routes that minimize interference with future requests, is also developed. This is achieved by identifying the critical links in the network, using the maxflow-mincut principle. First, the maxflows between all possible ingress-egress nodes, excluding the pair currently requested, is determined using the Goldberg-Tarjan highest label perflow push algorithm [34]. Computation of the maxflow values allows edges in the mincut to be found, due to linear programming duality [35]. Such edges are deemed to be critical, and are reflected as weights in the network graph. Thus, by choosing

the shortest path with the least cost in terms of criticality, the route determined is the least likely to interfere with future requests.

In [30], the presence of a centralized route server which keeps the up-to-date link state information in the form of a graph is assumed. When an LSP-request arrives at an ingress router, it queries the route server, which then computes the explicit route to satisfy the request by using a path selection algorithm on the graph representing the current network state. A major drawback of using a centralized route server is that it requires accurate link state information to compute paths. The routing scheme is therefore only suitable for small networks and is not scalable. Instead of using a centralized route server, one possible alternative is to let every ingress router maintain the topology information based on the optical- and IP-LSAs generated by the OXCs and IP routers. Such a topology is constructed based on the previously received link state updates. Hence uncertainty exists in the resource availability information related to both the IP and optical layers. When an LSP-request arrives at an ingress router, it uses the topology information stored within it to select a path after modelling the uncertainties. Thus, this solution is amenable to distributed implementation and is also scalable to large networks.

2.3 Routing with Inaccurate Link State Information

QoS routing in non-WDM networks in the presence of inaccurate link state information has been studied where the link state information is related to bandwidth and delay. In particular, several routing algorithms have been presented that choose paths which are most likely to satisfy the specific QoS requirements of either bandwidth or delay in [32], [36],and [37]. In [38], the impact of stale link state information on

QoS routing in non-WDM networks is evaluated. These approaches and algorithms cannot be directly extended to WDM networks because wavelength inaccuracy is a unique feature of WDM networks.

In [32], the path selection algorithm focuses on selecting a path that is capable of satisfying the bandwidth requirement of a flow, while at the same time trying to minimize the amount of network resources that need to be allocated to support the flow. Instead of real time update of link state information in terms of available bandwidth of each link, a simple hybrid update mechanism, which attempts to reconcile accuracy of link state information with the need for the smallest possible overhead, is used. In this update mechanism, each node sends an LSA only when the ratio between a current bandwidth value, b_w, of a link and the last reported value is above (or below) a threshold, say 2. This implies that when a path with some b units of bandwidth is sought, links with advertised bandwidth values above $2b$ are 'safe bets' and those with values below $\frac{b}{2}$ should be excluded, and all the rest may supply the required bandwidth with various degrees of certainty. By incorporating the certainty of each link in the path selection process, a probabilistic approach is proposed to choose a path with the maximum certainty to support the bandwidth requirement b as follows:

Based on the hybrid update mechanism, the bandwidth value of a link l is a random variable that takes value in $(\frac{b_l}{2}, 2b_l)$, where b_l is the last advertised value. Assuming these values are uniformly distributed, One can compute for each bandwidth requirement b the success probability of a link l, say $p_l(b)$, and then run a standard shortest path algorithm on the metric w_l, where $w_l = -log(p_l(b))$.

2.4 Non-real Time Update in WDM Networks

The control channel bandwidth requirements have been studied for a threshold-triggered wavelength update model in WDM optical networks in [39]. OSPF's opaque LSA mechanism is used to extend OSPF to disseminate optical resource related information through optical LSAs. Standard link-state database flooding mechanisms are used for distribution of optical LSAs. In the absence of any change in the network state, the optical LSAs are refreshed at regular refresh intervals of 30 min. In addition to regular refreshes, LSAs need to be updated to reflect changes in the network state. In order to reduce the number of optical LSA updates, the paper presents two configurable update mechanisms: relative change based triggers and absolute change based triggers.

In relative change based triggers, an update is triggered when the relative difference between the current and previously advertised link states exceeds a certain threshold. In absolute change based triggers, the measure of change is absolute, i.e., an update is triggered when the link state reaches a certain configurable constant.

However, the routing problem is not studied in [39] . Further, it considers only the optical layer where link state information corresponds to only wavelengths but not both bandwidth and wavelengths as in integrated IP/WDM networks.

2.5 Load Balancing in IP/MPLS networks

The load balancing issue has been studied in IP/MPLS networks. In [40], a multipath adaptive traffic engineering mechanism, called MATE, is presented which is targeted for switched networks such as MPLS networks. The main goal of MATE is to avoid

network congestion by adaptively balancing the load among multiple paths based on measurements and analysis of path congestion. MATE uses a state-dependent mechanism which deals with adaptive traffic assignment according to the current state of the network which may be based on utilization, packet delay, packet loss, etc.

MATE's operational setting assumes that several explicit LSPs (typically range from two to five) between an ingress node and an egress node in an MPLS domain have been established using a standard protocol such as CR-LDP or RSVP, or configured manually. The goal of the ingress node is to distribute the traffic across the LSPs so that the loads are balanced and congestion is thus minimized.

Figure 2.1 shows a functional block diagram of MATE located at an ingress node. Incoming traffic enters into a filtering and distribution function whose objective is to facilitate traffic shifting among the LSPs in a way that reduces the possibilities of having packets arrive at the destination out of order. The traffic engineering function decides on when and how to shift traffic among the LSPs. This is done based on LSP statistics which are obtained from measurements using probe packets. The role of the measurement and analysis function is to obtain one-way LSP statistics such as packet delay and packet loss. This is done by having the ingress node transmit probe packets periodically to the egress node which returns them to the ingress node. In [40], packet delay is used as it can be reliably measured by transmitting a probe message from the ingress node to the egress node.

However, due to the unique features of OBS networks such as no electronic buffering and no/limited optical buffering, the algorithm proposed for MPLS-based IP networks cannot be directly extended to OBS networks.

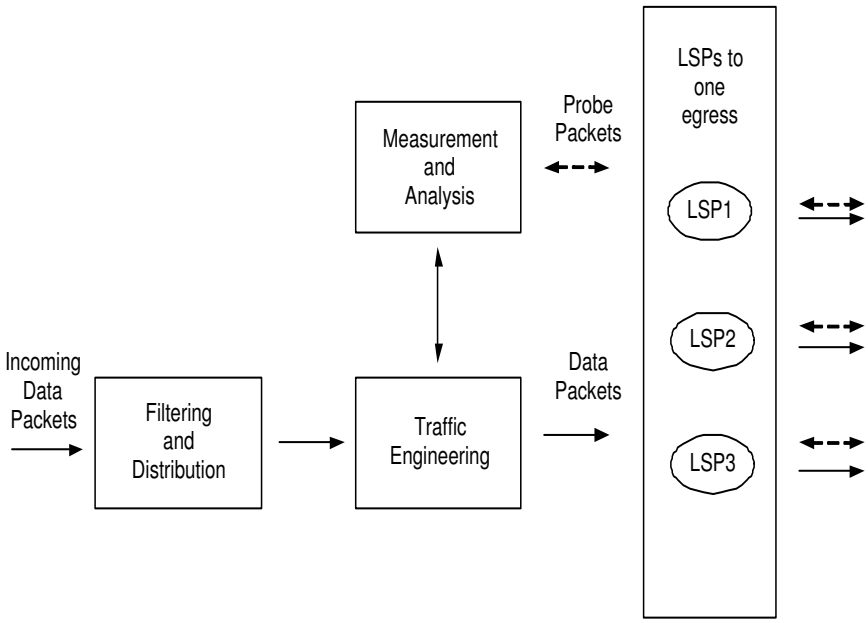

Figure 2.1: MATE functions in an Ingress node.

2.6 Contention Problem in OBS Networks

In [16], several data channel scheduling algorithms have been presented to schedule bursts efficiently while achieving a high bandwidth utilization at the same time. Data channel scheduling algorithms can be classified into two categories: without and with *void filling* (VF). A typical scheduling algorithm without void filling is the *latest available unscheduled channel* (LAUC) algorithm. In the LAUC algorithm, only one value—*the unscheduled time*—is maintained for each data channel. The basic idea of the LAUC algorithm is to minimize gaps/voids by selecting the latest available *unscheduled* data channel for each arriving data burst. LAUC can be extended to a more sophisticated scheduling algorithm by incorporating void filling, which is called *latest available unused channel with void filling* (LAUC-VF). Different from LAUC, LAUC-VF records the void/gap between two data bursts and the void can be filled by new data bursts. The basic idea of the LAUC-VF algorithm is to minimize voids

by selecting the latest available *unused* data channel for each arriving data burst.

In [21], an offset time based scheme is proposed to solve the inter-class contention problem. An extra offset time is set for a high priority class in order to isolate it from a low priority class. To achieve 100 percent class isolation, the extra offset time assigned to the high priority class needs to be longer than the maximum duration of a burst belonging to the low priority class. The limitation of this scheme is that the delay of high priority bursts is dramatically increased and more storage capacity is required at the edge nodes. Furthermore, it has the tendency to prefer short bursts than long bursts for low priority class especially when the traffic load becomes heavy [41].

Several intra-class contention resolution schemes are presented in the literature [22], [23]. In [22], a burst segmentation scheme to reduce packet losses is presented. With burst segmentation, in case of contention, instead of dropping the entire burst only the overlapping segments are dropped. This scheme is useful for certain kinds of applications which have stringent delay requirements but relaxed packet loss requirements. A contention resolution scheme using enhanced alternative routing is investigated in [23]. Here, bursts are transmitted along the shortest path determined at the ingress nodes in the normal condition. When contention occurs at a core node, the burst encountering contention is rerouted through an alternative route from that core node and downstream nodes perform hop-by-hop routing until the burst does not encounter any contention.

The above reported works use burst-centric approaches to deal with the contention problem. However, from the point of view the whole network, contention can be reduced by avoiding network congestion through load balancing. Therefore, in this

thesis, we consider the problem of contention in OBS networks from the perspective of load balancing.

Chapter 3

Dynamic Routing in Integrated IP-over-WDM Networks with Inaccurate Link State Information

3.1 Introduction

In this chapter, we consider the problem of dynamically routing bandwidth-guaranteed LSPs in an integrated IP-over-WDM network with inaccurate link state information. To select a good path, a routing algorithm needs up-to-date link state information. This leads to excessive update overhead and scalability problems. In real networks, from the practical point of view, in order to avoid excessive overhead of advertising and processing link state information, updates need to be made periodically or based on a threshold trigger. This leads to inaccuracies in the link state information. We consider the routing problem taking into consideration the uncertainty of link state parameters due to wavelength inaccuracy (*optical layer*) in addition to bandwidth inaccuracy (*IP layer*). Based on the threshold-triggered update scheme, we present a probabilistic method to model the uncertainty of link state parameters. We then define a cost function reflecting the uncertainty. Depending on different cost metrics chosen to be optimized, we propose two routing algorithms considering

the uncertainty of link state parameters. The objective is to minimize the impact of inaccurate information so that the blocking probability as well as setup failures are reduced. We use various performance metrics such as the total blocking probability, blocking probability due to setup failures, blocking probability due to routing failures, bandwidth update frequency, and wavelength update frequency to evaluate the effectiveness of the proposed algorithms. Through extensive simulation experiments, we show that our algorithms can significantly reduce the impact of inaccurate link state information and perform very well.

The rest of this chapter is organized as follows. In section 3.2, we give the motivation for our work. In section 3.3, a graph representation of the integrated IP/WDM network is presented. The bandwidth and wavelength update models developed are then explained. The proposed routing algorithms are described in Section 3.4. The results of a performance study via simulation are presented in section 3.5.

3.2 Motivation

In the integrated IP/WDM network model described in [30], the link state information includes the residual bandwidth of the logical link (*IP layer*) and free wavelengths on the physical link (*optical layer*). A path chosen by an integrated routing algorithm may traverse either of or both the logical links (existing lightpaths) and the physical links. The traversal of physical links results in the creation of new lightpaths changing the IP layer topology. By making use of the combined network state with the resource and topological information at both the IP and optical layers, LSPs can be routed in a resource-efficient way. The network state is built and updated by the optical- and IP- LSAs by using OSPF extensions. Each node can obtain up-to-date link state

information if updates occur immediately whenever there is a change in the link state, i.e., whenever there is a change in bandwidth usage on a logical link or wavelength usage on a physical link. However, in real networks, instead of immediate updates, periodic updates or threshold-triggered updates are preferred in order to reduce the update overhead and improve scalability [32], [36], [38], [39]. As a result, the link state information used by the routing algorithm may be stale or inaccurate. This leads to inefficient resource utilization and other problems as explained below.

When non real-time updates are used, blocking of LSP connections are caused by *routing failures* or *setup failures*. When an LSP request arrives, the ingress node uses a routing algorithm to compute a feasible path based on the stored link state information. If the path is not available, the request is rejected. We refer to this failure as routing failure. If a feasible path is returned by the routing algorithm, signaling messages are sent along the chosen path to reserve resources to set up the LSP connection. In this phase, if resources are not available the setup fails. We refer to this failure as setup failure. Due to the outdated update, the difference between the current link state information and that stored at each node results in either some available resources being regarded as unavailable or some occupied resources being regarded as available. The former contributes to routing failures while the latter causes setup failures.

A routing failure can be decided locally at the ingress node. In contrast, a setup failure is reported by the first node (along the chosen path) which has insufficient resources to honor the request. Since signaling is needed in this setup process extra signaling overhead is introduced into the network. It is therefore imperative that the uncertainty associated with the stored link state be modelled as accurately as

possible so that routing of LSPs uses resources efficiently and at the same time the setup failures are reduced.

Our work here considers inaccuracies in both the bandwidth and wavelength link state information in integrated IP/WDM networks with GMPLS capabilities, which, to the best of our knowledge, has not been studied previously. Since uncertainty arises from inaccurate information, we develop a easy-to-implement but practical and effective probabilistic method to define the uncertainty of link state parameters. We then propose routing algorithms that use uncertainty as a cost metric.

3.3 Network and Update Model

We consider an integrated IP/WDM network with n nodes, m links, and w wavelengths per fiber. Each node comprises an optical OXC and an IP router connected through ports. The optical OXC performs pure optical switching without optical wavelength conversion capability and the granularity of switching is a whole wavelength, while the IP router has O-E-O conversion and sub-wavelength multiplexing capabilities. We assume that there are enough ports between an OXC-router pair. LSP requests can be generated and terminated at each router. Each LSP request can be represented by the triple (source node-s, destination node-d, bandwidth request-b). We define the full capacity of a wavelength as some integer c, while the requested bandwidth is a fraction of c. We first present a network model to represent the integrated IP/WDM network. Then we present an update model which describes how the link state information about the bandwidth, wavelength and logical edge are updated and how the uncertainty of the link state parameters is represented.

3.3.1 Network Model

In this section, we describe an integrated wavelength-layered graph to model the integrated IP/WDM network. The network model is similar to that presented in [30]. This graph representation is used by the routing algorithms to compute paths for LSP requests.

The network can be represented by a wavelength-layered graph, which consists of w layers with each wavelength layer corresponding to the physical topology on a particular wavelength. In each wavelength layer λ_i, we refer to the nodes as sub-nodes and each sub-node corresponds to an OXC. A physical edge corresponds to a wavelength on a physical link. A super-node is introduced into the graph which functions like a router performing O-E-O conversion. All the sub-nodes in the wavelength layers corresponding to the same node in the physical topology are connected to a super-node. The links between sub-nodes and super-nodes are referred to as O-E-O edges. An O-E-O edge models O-E-O wavelength conversion. Through a super-node, two lightpaths on the same/different wavelengths can be connected. We use Fig. 3.1 to illustrate the wavelength-layered graph representation. Figure 3.1(a) shows a physical network with 4 nodes interconnected by directed optical links. Each link is assumed to have two wavelengths, λ_1 and λ_2. Figure 3.1(b) shows an instance of the wavelength-layered graph.

When there is no traffic in the network, all the edges in the graph are physical edges with full capacity. After routing some LSPs, lightpaths are set up in the network. A lightpath is represented as a logical edge with residual bandwidth in the graph. Once the logical edge is formed, all the physical edges constituting the

lightpath are removed from the graph. We use Fig. 3.1(b) to illustrate the logical edge. Assume that a lightpath with route $B_0 \rightarrow B_1 \rightarrow C_1 \rightarrow D_1 \rightarrow D_0$ using wavelength λ_1 is established to route an LSP request with the bandwidth requirement of b units from B to D. A logical edge corresponding to the newly opened lightpath is created in the graph with the residual bandwidth of $(c-b)$ units. Note that the physical edges used by the logical edge have been deleted. Now, the graph consists of both physical edges and logical edges which can be used by the routing algorithm to compute paths for LSP requests that arrive later. For example, in Fig. 3.1(b), there are four candidate paths to route an LSP request from A to D. The first possible path traverses the physical edge between node A and B followed by the logical edge between node B and D using wavelength λ_1. The second possible path traverses three consecutive physical edges between node A and D using wavelength λ_2. The third possible path traverses the physical edge between node A and B using wavelength λ_2 and through wavelength conversion at node B traverses the logical edge between B and D using wavelength λ_1. The fourth possible path traverses the physical edge between node A and B using wavelength λ_1 and through wavelength conversion at node B traverses the two consecutive physical edges between node B and D using wavelength λ_2. Since OXCs do not support sub-wavelength multiplexing, node C cannot use this logical edge to add or drop LSPs. Each time when a logical edge is used, the corresponding amount of bandwidth of the routed request is reduced from the residual bandwidth. When the residual bandwidth of a logical edge returns to the full capacity upon the termination of LSPs, the logical edge is torn down and the constituent physical edges are restored in the graph.

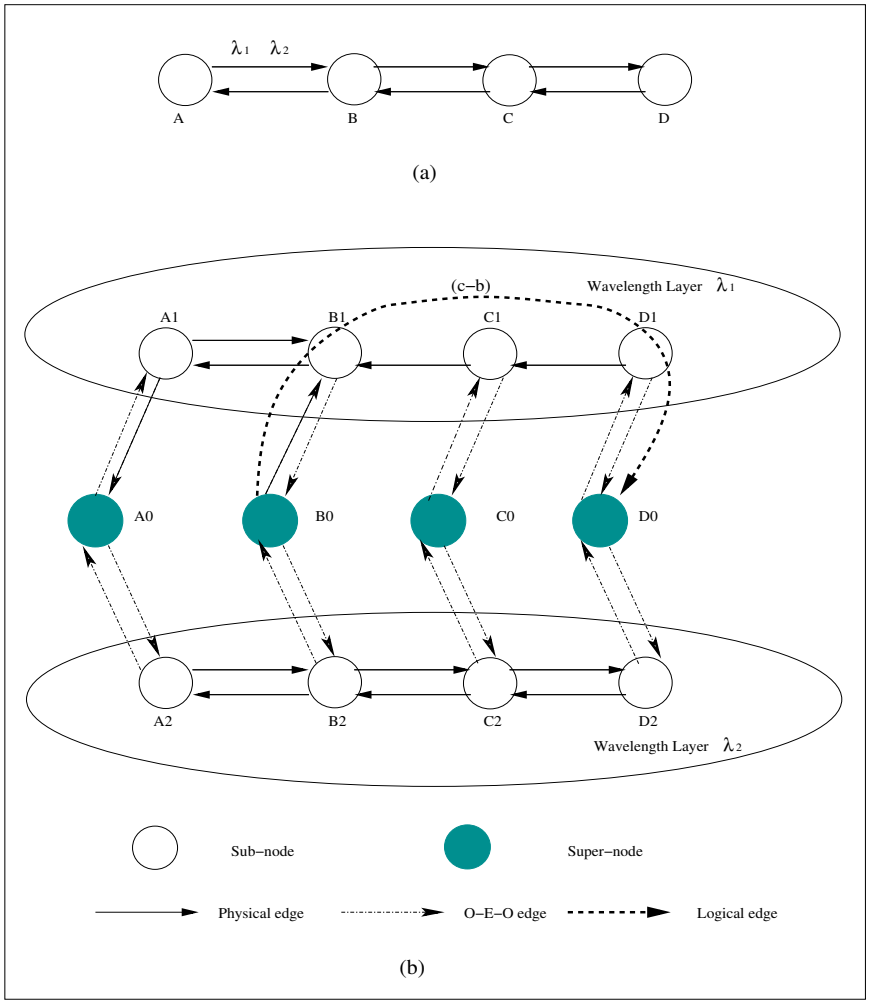

Figure 3.1: (a) A physical network (b) An instance of the wavelength-layered graph

3.3.2 Update Model

In an integrated IP/WDM network where logical edges (IP layer) and physical edges (Optical layer) co-exist, three kinds of link state information need to be announced to the whole network through link state information updates. These are the residual bandwidths of logical edges, free wavelengths on the physical links and the IP layer topology changes due to setup & tear-down of logical edges.

In a GMPLS network, both the logical links and physical links are referred to as *Traffic Engineering* (TE) links with TE properties [42]. The TE metrics include the maximum bandwidth and unreserved bandwidth. The head-end node of a TE link is responsible for collecting and advertising the information about the link through OSPF/ISIS flooding. As a result of this flooding, the information stored in the link state database at each node includes not only the physical links but also the logical links.

In an IP/WDM network, a logical edge starts and ends at routers while a physical edge starts and ends at OXCs. The IP routers are responsible for advertising the information about the IP layer such as IP layer topology changes and the residual bandwidths of the logical edges. Information about the availability of wavelengths on the physical links is advertised by the OXCs. All this information can be carried in the opaque LSA and are advertised to all the network nodes. Extensions to OSPF in support of GMPLS are currently being standardized by the IETF to support the new features of opaque LSA [42],[43].

3.3.2.1 Update Schemes

In order to avoid excessive overheads due to link state information updates, especially in large networks, link state information can be updated periodically or based on a threshold trigger [32],[36],[37],[38],[39]. We adopt the threshold-triggered update method. We describe below the updates dealing with different kinds of link state information.

Bandwidth of a logical edge (IP layer): A threshold B is set so that when the difference between the current and previously advertised values of the bandwidth exceeds B, the current value is advertised by the head-end router.

IP layer topology change (IP layer): Once a logical edge is set up (due to the setting up of an LSP connection) or torn down (due to the release of an LSP connection), a direct edge between the two end routers is introduced to or removed from the IP layer topology. Since the change of topology is critical to the network, the head-end router advertises the topological change information immediately to the whole network. Note that the content of LSA does not include the information about the intermediate nodes (OXCs) constituting an edge in the IP topology [42], [43].

Wavelength on a physical link (optical layer) : A physical link consists of w wavelengths, each corresponding to a physical edge in the wavelength layered graph. At any time instant, the w wavelengths can be divided into two different sets: a *free* (idle) wavelength set F and a *used* (busy) wavelength set U. The status of a wavelength on a physical link can become busy (idle) from idle (busy) due to the setting up (tearing down) of a logical edge using this wavelength on the physical link. We set a threshold K such that when the number of wavelengths that change status

in both sets exceeds K, the current link state of this physical link is advertised and the two sets are updated accordingly.

3.3.2.2 Modelling the Uncertainty

We now show how to represent the uncertainty of link state parameters (both bandwidth and wavelength) in a probabilistic manner based on the update schemes presented and the independence model that is assumed.

Bandwidth Uncertainty: We use a method similar to the one presented in [32] to represent the bandwidth uncertainty. Consider a logical edge l. Let the value of available bandwidth stored in the link state database based on the last advertisement be B_a. Let the actual value of the available bandwidth be B_c. We assume B_c is uniformly distributed within the region determined by the update threshold B. The region varies with different values of B_a. If $B_a \leq B$, the region is $(0, B_a + B)$; if $B < B_a < c - B$, the region is $(B_a - B, B_a + B)$, where c is the full capacity of the logical edge; and if $B_a \geq c - B$, the region is $(B_a - B, c)$. Assume that an LSP requesting bandwidth b arrives. Let $P(b, l)$ be the probability that bandwidth b is available in logical link l. Three cases are identified to calculate $P(b, l)$ as given below:

Case 1: if $B_a \leq B$ then

$$P(b,l) = \begin{cases} 0 & \text{if } b \geq B_a + B; \\ \frac{B_a + B - b}{B_a + B} & \text{if } 0 < b < B_a + B. \end{cases}$$

Case 2: if $B < B_a < c - B$ then

$$P(b,l) = \begin{cases} 0 & \text{if } b \geq B_a + B; \\ \frac{B_a + B - b}{2B} & \text{if } B_a - B < b < B_a + B; \\ 1 & \text{if } b \leq B_a - B. \end{cases}$$

Case 3: if $B_a \geq c - B$ then

$$P(b,l) = \begin{cases} \frac{c-b}{c-B_a+B} & \text{if } B_a - B < b \leq c; \\ 1 & \text{if } b \leq B_a - B. \end{cases}$$

Wavelength Uncertainty: In our wavelength update scheme, a wavelength update is triggered when the total number of wavelengths that change status exceeds a predefined threshold K. Recall that a wavelength may belong to the *free* set F or *used* set U. Consider a physical link l. Let X be the number of wavelengths in set F and Y be the number of wavelengths in set U associated with link l based on the last advertised wavelength update. Note that the sum of X and Y equals to the total number of wavelengths w. At a given instant, it is possible that some wavelengths in F and U have changed their status. Let C_x be the possible number of wavelengths in set F and C_y the possible number of wavelengths in set U that change status in the given time before the next wavelength update. Note that the sum of C_x and C_y is at most K. Let the pair (C_x, C_y) represent a status-changed state, where $0 \leq C_x \leq K_1$, $0 \leq C_y \leq K_2$, and $K_1 + K_2 \leq K$. Let S be the set of all possible status-changed states.

Consider wavelength λ that belongs to set F. This wavelength λ was free at the time of the last wavelength update. Now, we calculate the probability that wavelength λ is no longer free, i.e. the probability that λ has changed its status. First, we determine set S identifying all possible status-changed states (C_x, C_y). Note that set S depends on X, Y, and K. We assume that all the status-changed states in set S are equally probable with probability $\frac{1}{|S|}$, where $|S|$ denotes the size of S. We also assume that the change of state on a link is independent of that in other links. These assumptions are made for the purpose of simple and practical implementa-

tion enabling fast online path selection to handle the dynamically arriving requests. Complex algorithms without making independence assumptions could be more accurate but they might become impractical since their computational requirement is too high for network scalability. We therefore develop a simple and reasonably effective algorithm by using independence assumption on the change of link state. We note that this assumption applies only between two consecutive link state updates. Therefore, the inaccuracy caused by this assumption is not expected to be significant. Further, through extensive simulation experiments, we demonstrate the applicability and effectiveness of this model.

Let S_i be a subset of S comprising of all the status-changed pairs wherein the value of C_x is i. We denote the size of S_i as $|S_i|$. Let C_t^s denote the combination $C(s,t)$ whose value is the number of ways s objects can be selected from t objects.

A status-changed state (i,j) means that i wavelengths out of X wavelengths in set F and j wavelengths out of Y wavelengths in set U have possibly changed their status. Let q_i be the probability that wavelength $\lambda \in F$ has changed its status for the given state (i,j). Then, q_i is calculated as $\frac{C_{X-1}^{i-1}}{C_X^i}$. It may be noted that there are $|S_i|$ possible states of the form $(i,-)$ each of which occurs with probability $\frac{1}{|S|}$.

Let p_i be the probability that wavelength $\lambda \in F$ has changed its status given that i wavelengths in set F have changed their status. Then, p_i is calculated as $q_i \times \frac{1}{|S|} \times |S_i| = \frac{C_{X-1}^{i-1}}{C_X^i} \times \frac{1}{|S|} \times |S_i|$. Let $P_u(\lambda, l)$ be the probability that wavelength λ on physical link l has changed to *busy* status from *idle* status. The value of $P_u(\lambda, l)$ can be calculated in a general way as $\sum_{i=1}^{min(X,K)} p_i$.

Let $P(\lambda, l)$ be the probability that wavelength λ on physical link l in set F does not change its status. The value of $P(\lambda, l)$ can be calculated as $1 - P_u(\lambda, l)$.

In order to support fast selection of paths for LSPs, the values of $P(\lambda, l)$ can be computed off-line for different possible values of X and Y for the predefined value of K.

We illustrate the above probabilistic expressions through a simple example. Consider the case of $X = 3$, $Y = 3$, and $K = 2$. In this case, all possible status-changed states (C_x, C_y) in set S are $(0,0)$, $(0,1)$, $(0,2)$, $(1,0)$, $(1,1)$, and $(2,0)$ with $|S| = 6$. Each state (C_x, C_y) has the same probability $\frac{1}{|S|} = 1/6$. Since only at states $(1,-)$ and $(2,-)$, the wavelengths in set F have possibly changed their status, the cases of $i = 1$ and $i = 2$ need to be considered when calculating $P(\lambda, l)$. For $i = 1$, the possible states $(1,-)$ in subset S_1 are $(1,1)$ and $(1,0)$ with $|S_1| = 2$. For $i = 2$, the possible state $(2,-)$ in subset S_2 is $(2,0)$ with $|S_2| = 1$. The values of q_i, p_i, $P(\lambda, l)$, and $P_u(\lambda, l)$ can be calculated as follows:

$$
\begin{aligned}
q_1 &= \frac{C_{3-1}^{1-1}}{C_3^1} = 1/3. \\
p_1 &= q_1 \times \frac{1}{|S|} \times |S_1| = 1/3 \times 1/6 \times 2 = 1/9. \\
q_2 &= \frac{C_{3-1}^{2-1}}{C_3^2} = 2/3. \\
p_2 &= q_2 \times \frac{1}{|S|} \times |S_2| = 2/3 \times 1/6 \times 1 = 1/9. \\
P_u(\lambda, l) &= \sum_{i=1}^{min(X,K)} p_i = p_1 + p_2 = 1/9 + 1/9 = 2/9. \\
P(\lambda, l) &= 1 - P_u(\lambda, l) = 1 - 2/9 = 7/9.
\end{aligned}
$$

Depending on the values of X, Y, and K we identify the following cases to calculate $P_u(\lambda, l)$. The value of $P(\lambda, l)$ can be calculated as $1 - P_u(\lambda, l)$.

Case 1: $X \geq K$ and $Y \geq K$.

For this case, $|S| = \frac{(K+1)(K+2)}{2}$ and $|S_i| = K - i + 1$. Therefore,

$$
P_u(\lambda, l) = \frac{2}{(K+1)(K+2)} \sum_{i=1}^{K} (K - i + 1) \times \frac{C_{X-1}^{i-1}}{C_X^i}
$$

Case 2: $0 < X < K$ and $Y \geq K$.

For this case, $|S| = \frac{(K+1)(K+2)-(K-X)(K-X+1)}{2}$ and $|S_i| = K - i + 1$. Therefore,

$$P_u(\lambda, l) = (\frac{2}{(K+1)(K+2)-(K-X)(K-X+1)}) \times (\sum_{i=1}^{X}(K-i+1) \times \frac{C_{X-1}^{i-1}}{C_X^i})$$

Case 3: $X \geq K$ and $0 < Y < K$.

For this case, $|S| = \frac{(K+1)(K+2)-(K-Y)(K-Y+1)}{2}$, $|S_i| = Y + 1$ if $1 \leq i \leq K - Y$, and $|S_i| = K - i + 1$ if $K - Y + 1 \leq i \leq K$. Therefore,

$$P_u(\lambda, l) = (\frac{2}{(K+1)(K+2)-(K-Y)(K-Y+1)})$$
$$\times(\sum_{i=1}^{K-Y}(Y+1) \times \frac{C_{X-1}^{i-1}}{C_X^i} + \sum_{i=K-Y+1}^{K}(K-i+1) \times \frac{C_{X-1}^{i-1}}{C_X^i})$$

Case 4: $Y = 0$.

For this case, $|S| = K + 1$ and $|S_i| = 1$. Therefore,

$$P_u(\lambda, l) = \frac{1}{(K+1)} \sum_{i=1}^{K} \frac{C_{X-1}^{i-1}}{C_X^i}$$

3.4 Proposed Routing Algorithms

In this section, we propose two routing algorithms, *minimum hop most probable path* (MHMPP) and *most probable path* (MPP). These algorithms consider the uncertainty of link state parameters. They use the integrated graph described in section 3.3. The edges are assigned cost values to optimize certain cost metrics. The algorithms use different cost metrics to edges and hence to a path. A shortest-path selection algorithm is used on the graph to select a minimum-cost path. As we see, there are $O(nw)$ nodes in the graph. Application of Dijkstra's algorithm to select a minimum-cost path requires a worst case running time of $O(n^2w^2)$. The complexity could be reduced to $O(nw(n+w))$ if the Dijkstra-like algorithm developed in [26] is used. Note

that the algorithm developed in [26] is used to select a minimum-cost path on the optical layer. However, it can be suitably modified for use in integrated networks.

When an LSP request with bandwidth requirement b arrives, the following actions take place at the source node.

1. All the edges that cannot satisfy the bandwidth requirement from the graph are removed (by assigning infinite edge cost) to form a residual graph with logical edges whose residual bandwidth are at least b and physical edges whose bandwidth are the full wavelength.
2. Costs of the edges on the residual graph are assigned.
3. A minimum-cost path on the residual graph is chosen.
4. If a new lightpath needs to be established, the logical edge is created in the graph.
5. Signaling messages are sent to the downstream nodes to set up the LSP.

3.4.1 Cost Metrics

We use hops and uncertainty as cost metrics in our routing algorithms. A hop refers to a wavelength on a physical link. The importance of hops is more pronounced in WDM networks than in other networks because of the unique feature that traffic can enter and leave only at the end nodes but not at an intermediate node of a lightpath. Given a certain resource requirement, the uncertainty reflects the probability that the required resource can be guaranteed. A smaller uncertainty implies a larger probability of resource availability. Therefore, the most probable path is the path with the smallest uncertainty. We translate the probability of resource availability

into uncertainty and use the uncertainty to choose the minimum-cost path. This helps us to develop a Dijkstra-like shortest path selection algorithm as the uncertainty value is additive non-decreasing when more and more edges are traversed by a path. We show below that the least uncertain path (a path with the minimum uncertainty) is the most probable path (a path with the highest probabilistic guarantee).

Given a bandwidth requirement b and a path constituting s logical links and t physical links, the probability of resource availability associated with each link can be calculated using the formulae developed in Section 3.3. For a logical link l and bandwidth b, the probability is given by $P(b, l)$. For a physical link l and wavelength λ, the probability is given by $P(\lambda, l)$. With the independence assumption on the change of link state, the probabilistic guarantee of the path P can be calculated as the product of the probability of each link (physical link or logical link) constituting it:

$$P = \prod_{i=1}^{s} P(\lambda, i) \times \prod_{i=1}^{t} P(b, i)$$

The problem of finding the most probable path is to find the path with the largest probability: $maxP$. Since the probability associated with a path decreases as and when a new edge is added to it by a Dijkstra-like shortest-path finding algorithm, the probability values cannot be directly used. To overcome this, we define an appropriate cost function so that the probability associated with each link is translated to the cost of uncertainty. By doing so, we ensure that the cost associated with each link is non-negative and the cost of the path is the sum of the cost of each link constituting it.

Given a probability function $P(x)$, we define a corresponding cost function $C(x) = -\log P(x)$. For the logical edge l and bandwidth b we define $C(b, l) = -\log P(b, l)$.

For the physical edge l and wavelength λ we define $C(\lambda, l) = -\log P(\lambda, l)$. The cost of uncertainty of the path P is given as

$$C = \sum_{i=1}^{s} C(\lambda, i) + \sum_{i=1}^{t} C(b, i)$$

Therefore, when a shortest path algorithm is used with uncertainty as the cost metric, the minimum-cost path returned by it must be the path with $min\ C$. The transformation of a multiplicative path cost to an additive path cost is well-known and is given below for the purpose of clarity and completeness. This transformation implies that a $min\ C$ path is equivalent to the $max\ P$ path:

$$\begin{aligned}
\min\ C &\equiv min(\sum_{i=1}^{s} C(\lambda, i) + \sum_{i=1}^{t} C(b, i)) \\
&\equiv min(\sum_{i=1}^{s} -logP(\lambda, i) + \sum_{i=1}^{t} -logP(b, i)) \\
&\equiv min(log(\prod_{i=1}^{s} \frac{1}{P(\lambda, i)} \times \prod_{i=1}^{t} \frac{1}{P(b, i)})) \\
&\equiv min(\prod_{i=1}^{s} \frac{1}{P(\lambda, i)} \times \prod_{i=1}^{t} \frac{1}{P(b, i)}) \\
&\equiv max(\prod_{i=1}^{s} P(\lambda, i) \times \prod_{i=1}^{t} P(b, i)) \\
&\equiv max\ P.
\end{aligned}$$

We now describe the two algorithms developed by us. They are based on different cost metrics.

3.4.2 Algorithm MPP

Algorithm MPP considers only the uncertainty. It does not consider any other metrics such as hops. It selects the most probable path among all possible paths. The link cost function used for an edge has only one component corresponding to uncertainty. For a logical edge l traversing any number of hops and a requested bandwidth of b,

the link cost function is given by $C(b, l)$. For a physical edge l on wavelength λ the link cost function is given by $C(\lambda, l)$.

3.4.3 Algorithm MHMPP

Algorithm MHMPP considers both hops and uncertainty. A hop refers to a wavelength on a physical link. Hops are an important resource in WDM networks because of several reasons: (i) increased number of hops means that a large number of wavelength resources are used (ii) an LSP can enter or leave only at the end nodes of a lightpath and it cannot enter into or leave from an intermediate node. The algorithm selects the most probable path among the minimum-hop paths. The number of hops of a physical edge is 1. The number of hops of a logical edge is the number of physical edges traversed by it. Two physical edges are differentiated based on the uncertainty associated with the wavelength available on it. On the other hand, two logical edges are first differentiated based on their (physical) hop length and then based on the uncertainty associated with the bandwidth availability. The link cost function used for an edge has two components. The first component corresponds to hops while the second component corresponds to uncertainty. For a logical edge l traversing h hops and a requested bandwidth of b, the link cost function is given by $k_1 \times h + k_2 \times C(b, l)$. For a physical edge l on wavelength λ the link cost function is given by $k_1 + k_2 \times C(\lambda, l)$. We choose constants $k_1 \gg k_2$ such that the chosen path has the minimum number of hops and the second component in the link cost function (i.e. uncertainty) is used to break the tie.

3.5 Performance Study

In this section, we study the performance of our proposed algorithms through extensive simulations. The network used for simulation is randomly generated in a way that the node pairs are considered one by one. For a node pair, link is introduced with a certain probability. Finally, it is checked if the network is connected. The generated network shown in Fig. 3.2 has 32 nodes, 50 bidirectional links and 8 wavelengths per fiber. A bidirectional link comprises two unidirectional fibers in opposite directions. The LSP requests arrive at a (source) node as a Poisson process with an exponentially distributed mean holding time. Every node other than the source node is equally probable to be the destination node. An LSP requests low bandwidth with probability 0.8 and high bandwidth with probability 0.2. We assume that bandwidth of a full wavelength is 100. The low bandwidth requests are uniformly distributed in the range $[1, 10]$ whereas the high bandwidth requests are uniformly distributed in the range $[10, 100]$. We have chosen these values because many LSPs require relatively low bandwidth when compared to a full wavelength bandwidth [30].

We use the total blocking probability, blocking probability due to routing failures (the fraction of requests rejected by the routing algorithm) and setup failures (the fraction of requests that fail during setup), and bandwidth and wavelength update frequency as performance metrics. We note that both routing failures and setup failures contribute to the total blocking probability. The bandwidth (wavelength) update frequency is measured as the number of bandwidth (wavelength) updates generated in the entire network per holding time of an LSP connection.

We also implemented two other algorithms, *minhop-accurate* (MH-A) and

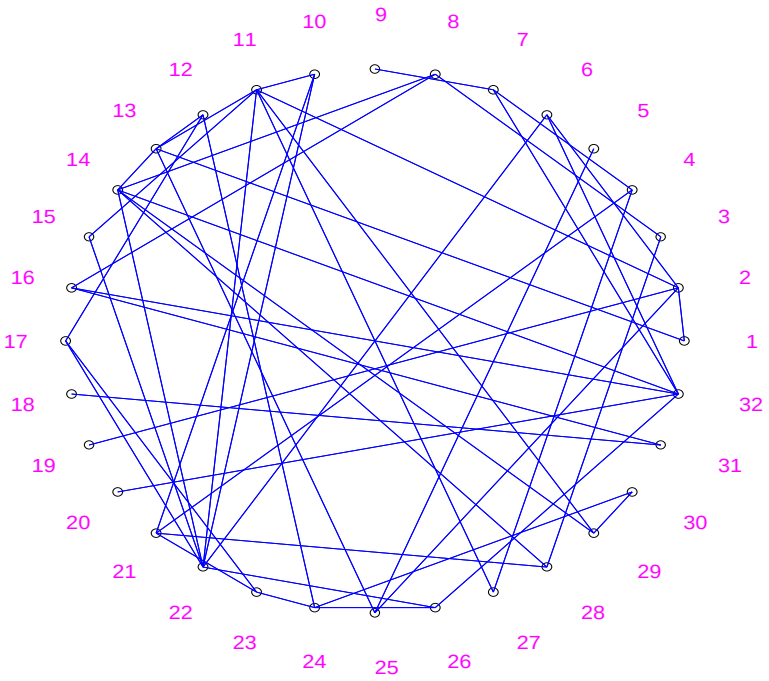

Figure 3.2: 32-node randomly generated network.

minhop-inaccurate (MH-I) to demonstrate the effectiveness of our algorithms in dealing with inaccurate link state information. Algorithm MH-A uses accurate state information for computing paths based on hops. It assumes immediate update of bandwidths and wavelengths whenever changes occur. The performance of this algorithm in terms of blocking probability is expected to be better than any of the algorithms using inaccurate link state information. On the other hand, algorithm MH-I uses inaccurate state information for computing paths based on hops. It does not consider the uncertainty factor. Instead, it assumes that the information available is correct. Hence, the performance of this algorithm in terms of blocking probability is expected to be the worst. When probability values are set to 1, algorithm MHMPP reduces to MH-A and MH-I for the accurate and inaccurate link state information, respectively. Therefore, the performance of MH-A and MH-I serve as bounds to verify

the effectiveness of our methods which consider uncertainty factor to select paths.

First, we evaluate the performance under different loading conditions for a specific combination of bandwidth and wavelength update threshold values. We then study the impact of bandwidth and wavelength update threshold on the performance by varying one threshold parameter while fixing the other one. Finally, we present a possible way of choosing appropriate update threshold values based on the simulation results.

3.5.1 Effect of Traffic Loading

We fix the bandwidth threshold $B = 20$ and wavelength threshold $K = 1$. Note that for the case of K=1 updates are made when the number of status-changed wavelengths is 2. We study the performance of our algorithms MHMPP and MPP and compare them with the algorithms MH-A and MH-I. We consider a wide range of traffic intensity values to account for low, medium and high traffic loading scenarios.

Fig. 3.3 shows the total blocking probability with varying traffic intensity per node. The traffic intensity is measured as the number of LSP requests arrived per holding time. We observe that our algorithms MHMPP and MPP perform much better than MH-I. At the same time their performance is very close to that of MH-A. This proves that our algorithms are able to model the uncertainty accurately. Now we compare the performance of MHMPP and MPP. They perform differently under different loading conditions. We recall that MHMPP considers both hop and uncertainty while MPP considers only uncertainty. Under light load conditions, sufficient resources (wavelengths and equivalently hops) are available and hence uncertainty becomes dominant over hops. Therefore, MPP performs better than MHMPP un-

der light load conditions (in the figure, it is below 5 Erlangs). On the other hand, MHMPP performs better than MPP when the traffic load increases.

The performance of the algorithms in terms of setup failures is depicted in Fig. 3.4. There is no setup failure reported by MH-A as it works with accurate information. We observe that the performance of our algorithms MHMPP and MPP is much better than that of MH-I. The performance trends of MHMPP and MPP are similar to the case of total blocking probability. We make an important and useful observation from Fig. 3.3 and Fig. 3.4. A *trough* appears in the graphs corresponding to MHMPP and MPP. When the traffic load is low (below 5 Erlangs in this case), blocking probability and setup failures show a decreasing trend with increasing traffic intensity. The reason for this *trough* behavior can be explained as follows.

We recall that in integrated networks, the path chosen may traverse logical edges and (new) physical edges. When the load is very light, a logical edge carries little load and is highly likely to be released before the load builds up. As a result, a new LSP is likely to choose more physical edges. Since wavelengths are distinguishable (unlike the bandwidth resource) uncertainty associated with the chosen path increases when more (new) physical edges (hence more wavelengths) are traversed. This leads to more setup failures and hence a higher total blocking probability. When the traffic load increases, logical edges tend to stay for longer time and a new LSP is more likely to use logical edges than physical edges. This results in reduced setup failures and hence reduced total blocking probability. When the traffic intensity increases beyond a certain value, the blocking probability and setup failures increase because of resource shortage.

The above argument is supported by Fig. 3.5 and Fig. 3.6. Fig. 3.5 shows the mean

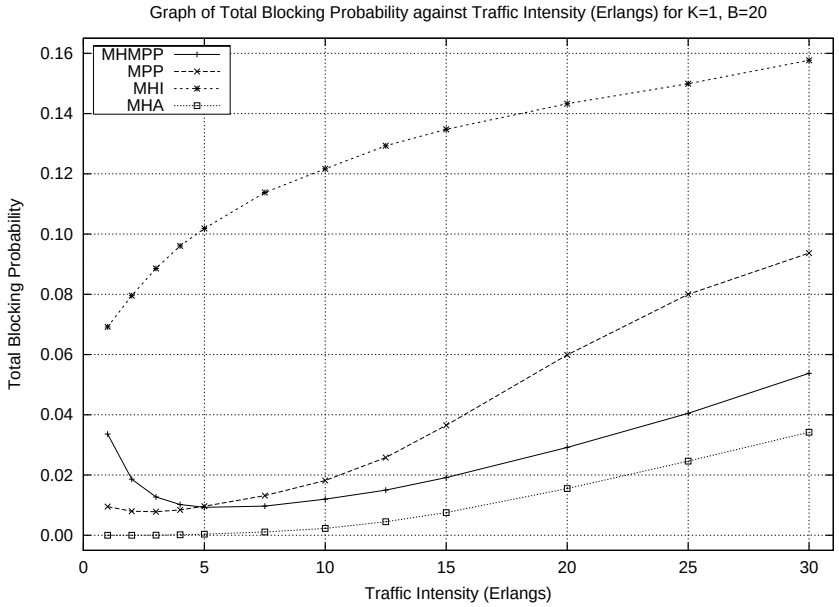

Figure 3.3: Graph of total blocking probability against traffic intensity(Erlangs) for K=1, B=20.

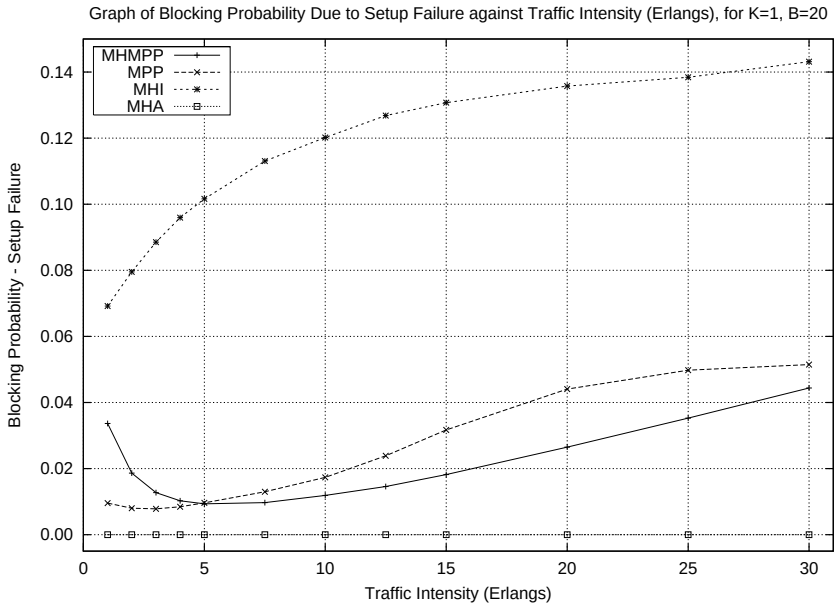

Figure 3.4: Graph of blocking probability due to setup failures against traffic intensity(Erlangs)for K=1, B=20.

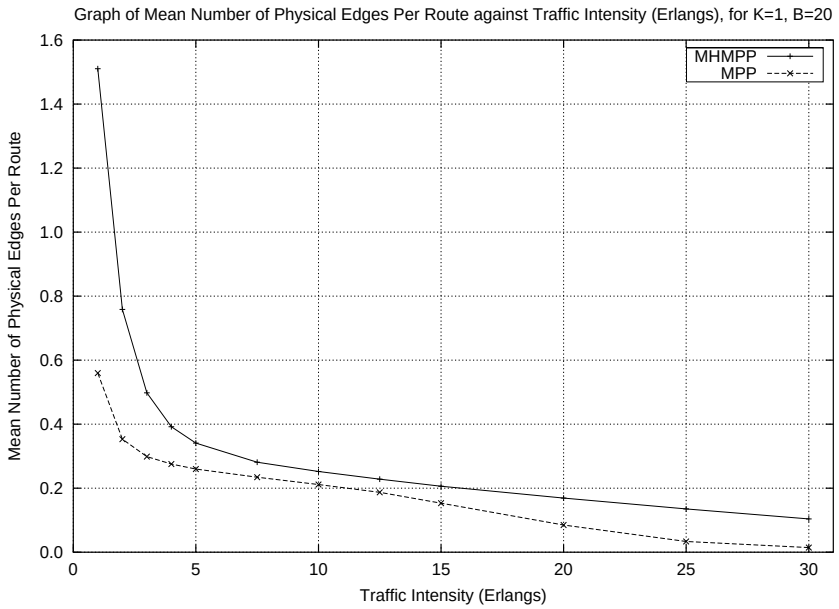

Figure 3.5: Graph of mean number of (new) physical edges per route against traffic intensity(Erlangs)for K=1, B=20.

number of (new) physical edges used by the path selected by the routing algorithm. Fig. 3.6 shows the mean path probability, i.e., the success probability associated with a path when selected by the path selection algorithm. When the traffic load is low (below 5 Erlangs), the mean path probability increases and the mean number of physical edges decreases with the increasing load due to the reasons stated above.

The graphs depicted in Fig. 3.7 show the blocking probability due to routing failures. Since algorithm MH-A is able to correctly decide if a path is available at the stage of path selection itself, the routing failure is high and is the same as the total blocking probability. Since other algorithms make incorrect decisions (i.e. they may choose a path guessing that it is feasible which may not actually be available) the routing failures may be low. While the performance of MPP is good when the traffic load is low, it degrades with increasing load. This is because it does not optimize the hops and consequently it tends to use more hops leading to poor performance.

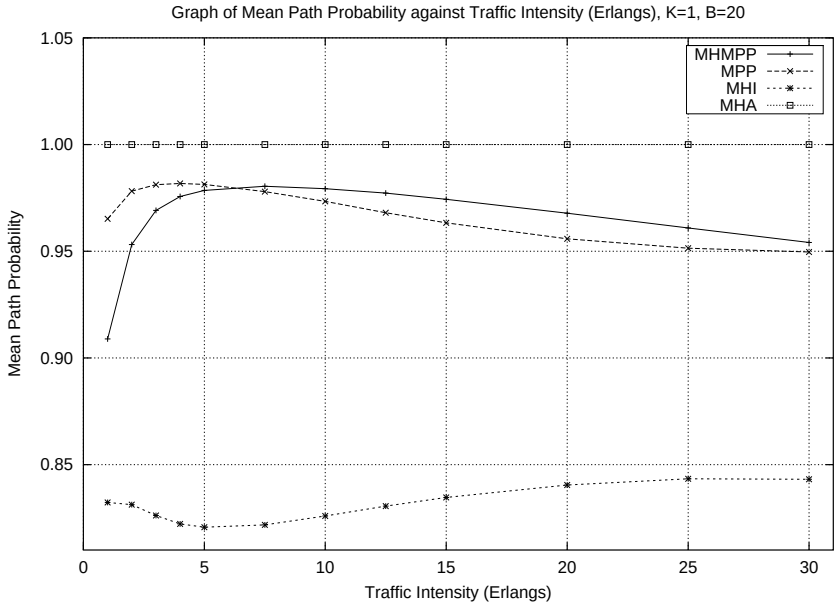

Figure 3.6: Graph of mean path probability against traffic intensity(Erlangs)for K=1, B=20.

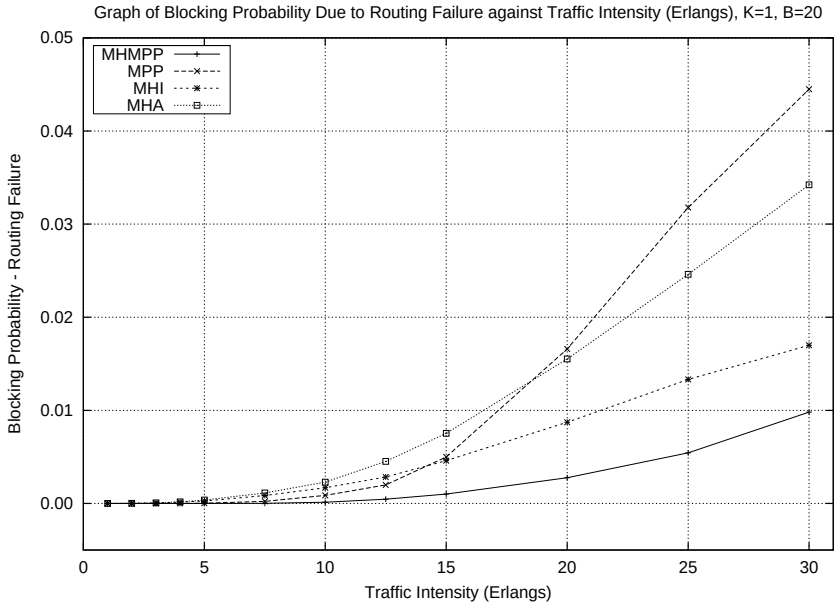

Figure 3.7: Graph of blocking probability due to routing failures against traffic intensity(Erlangs)for K=1, B=20.

3.5.2 Bandwidth and Wavelength Update Frequency

Fig. 3.8 plots the bandwidth update frequency in the entire network per connection holding time for varying traffic load per node. It can be observed that the update frequency and hence the messages generated by our algorithms are significantly lower than that of MH-A. When the traffic load increases, the update frequency for our algorithms increases slowly when compared to MH-A.

Fig. 3.9 shows the graph of wavelength update frequency for varying traffic load per node. In general, the wavelength update frequency of our algorithms is much lower than that of MH-A and also MH-I. Initially, when the traffic load increases, update frequency increases because wavelengths are used more frequently. But the update frequency either decreases or levels off when the traffic load becomes higher. This is because, at high load, the logical links are highly loaded and are less likely to be released. Also, as shown in Fig. 3.10, the number of free wavelengths on a link becomes low at heavy traffic load conditions. Therefore, the number of changes in wavelengths becomes low and hence the wavelength update frequency decreases.

3.5.3 Effect of Update Threshold

We evaluate the impact of varying update thresholds on the performance of our algorithms MHMPP and MPP for three traffic intensity values, 5, 15 and 30 Erlangs per node.

Fig. 3.11 through Fig. 3.13 show the effects of varying bandwidth update threshold for a fixed wavelength update threshold $K = 1$. From Fig. 3.11, we observe that the total blocking probability increases slowly with the increasing bandwidth

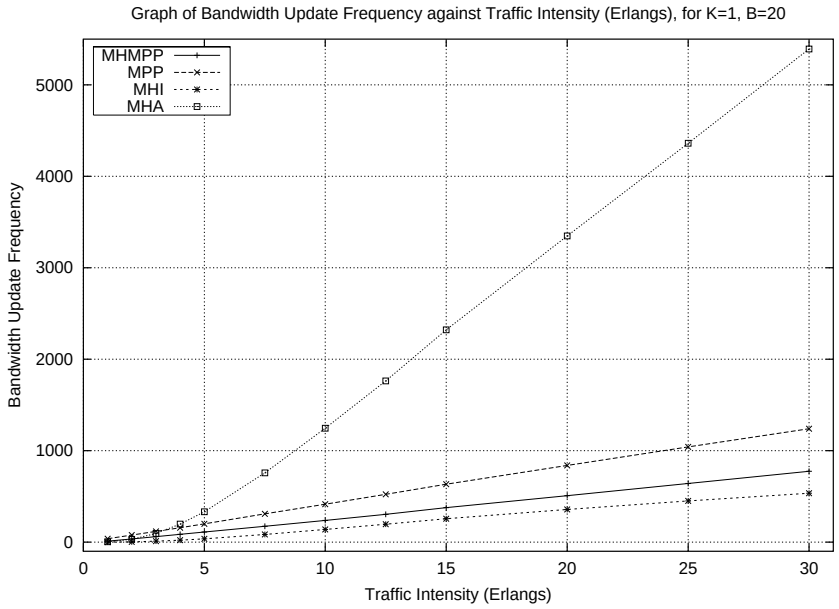

Figure 3.8: Graph of bandwidth update frequency against traffic intensity(Erlangs)for K=1, B=20.

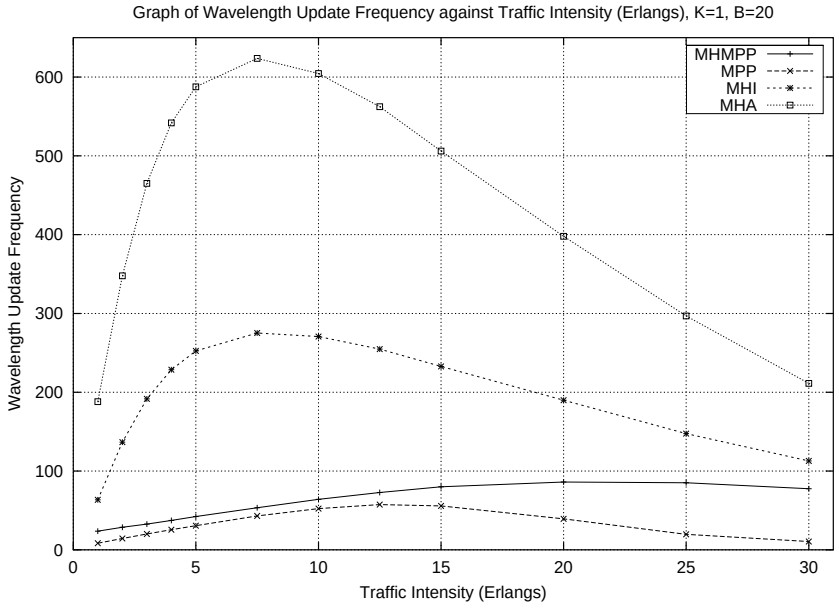

Figure 3.9: Graph of wavelength update frequency against traffic intensity(Erlangs)for K=1, B=20.

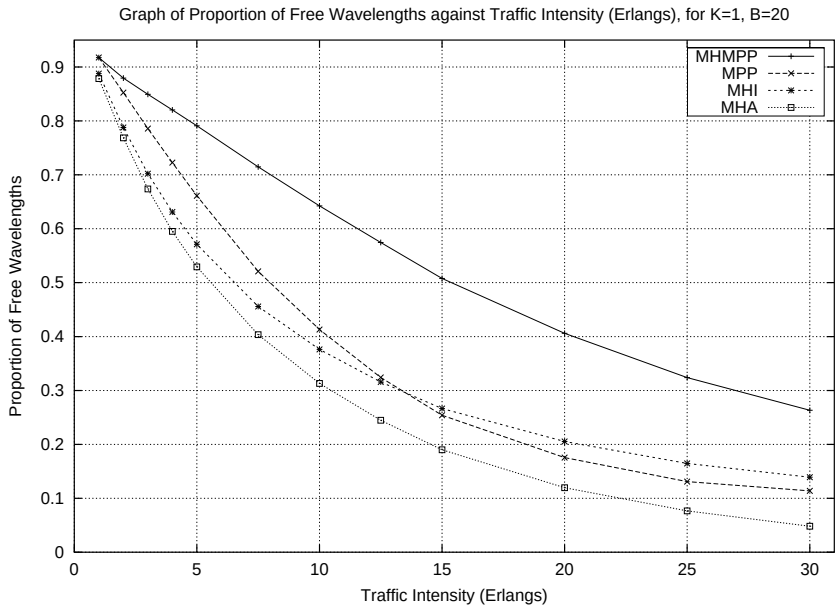

Figure 3.10: Graph of proportion of free wavelength against traffic intensity(Erlangs)for K=1, B=20.

threshold for both MHMPP and MPP. From Fig. 3.12, we observe that the bandwidth update frequency can be reduced significantly especially when the traffic load is high. From Fig. 3.13, we observe that in general the setup failure increases with the increasing bandwidth update threshold. From the above figures, we note that the larger the update threshold, the lower the update frequency (equivalently smaller number of update messages) and the larger the setup failures (equivalently larger number of signaling messages). By choosing an appropriate value for the bandwidth threshold, we can achieve a tradeoff between the blocking probability and the number of update and signaling messages. Through the above observation, we note that the overhead due to update and signaling messages can be reduced considerably without influencing the total blocking probability by appropriately choosing the bandwidth update threshold.

Fig. 3.14 through Fig. 3.16 show the effects of varying the wavelength update

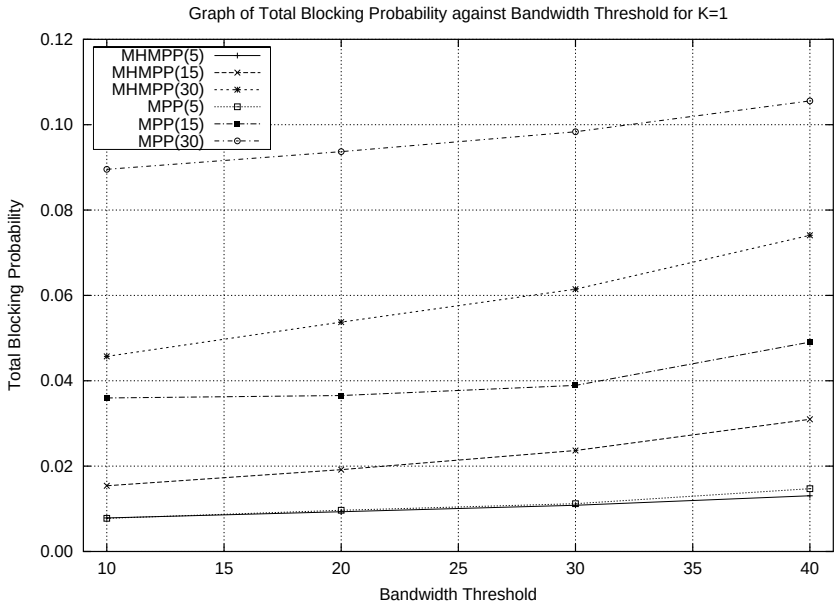

Figure 3.11: Graph of total blocking probability against bandwidth threshold for K=1.

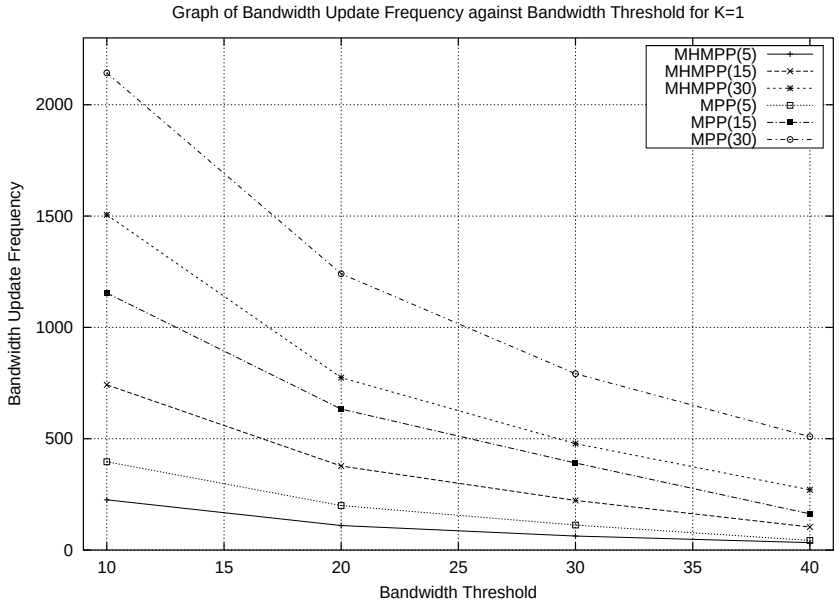

Figure 3.12: Graph of bandwidth update frequency against bandwidth threshold for K=1.

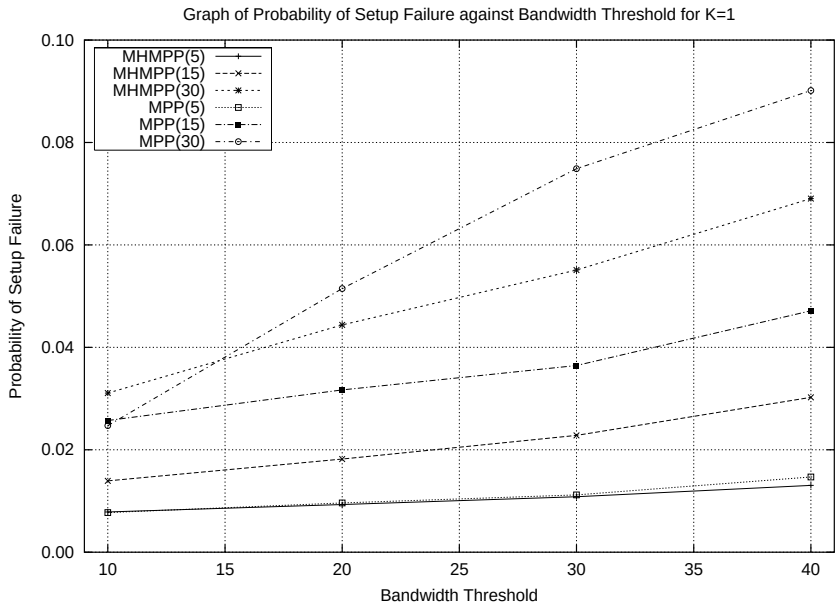

Figure 3.13: Graph of blocking probability due to setup failure against bandwidth threshold for K=1.

threshold for a fixed bandwidth update threshold $B = 20$. From Fig. 3.14, we observe that the total blocking probability increases slowly as the wavelength update threshold increases for both MHMPP and MPP. From Fig. 3.15, we observe that the wavelength update frequency can be lowered by increasing the wavelength threshold. From Fig. 3.16, we observe that the setup failure increases with increasing wavelength update threshold. Further we note that the wavelength update frequency is much smaller than the bandwidth update frequency because lightpaths tend to stay longer since they are used by several LSPs. By choosing an appropriate value for the wavelength update threshold, we can achieve a tradeoff between the blocking probability and the number of update and signaling messages.

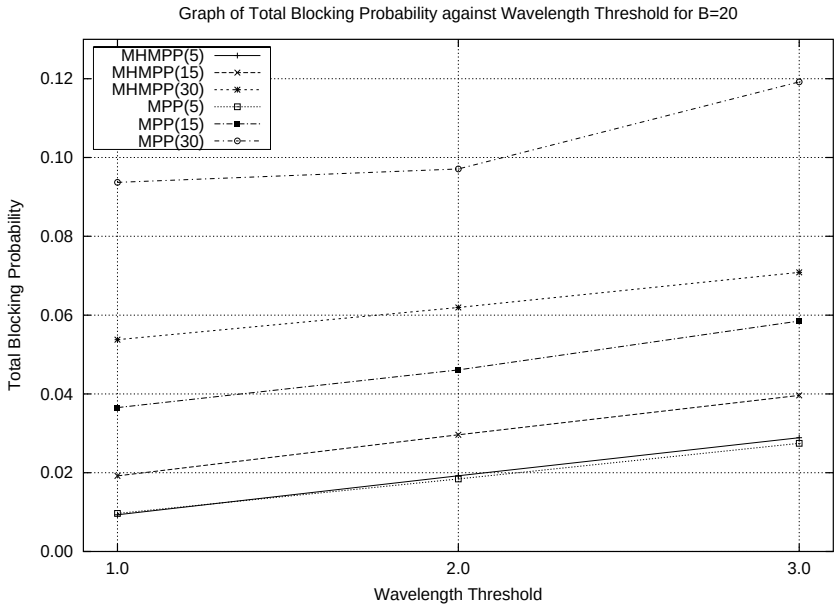

Figure 3.14: Graph of total blocking probability against wavelength threshold for B=20.

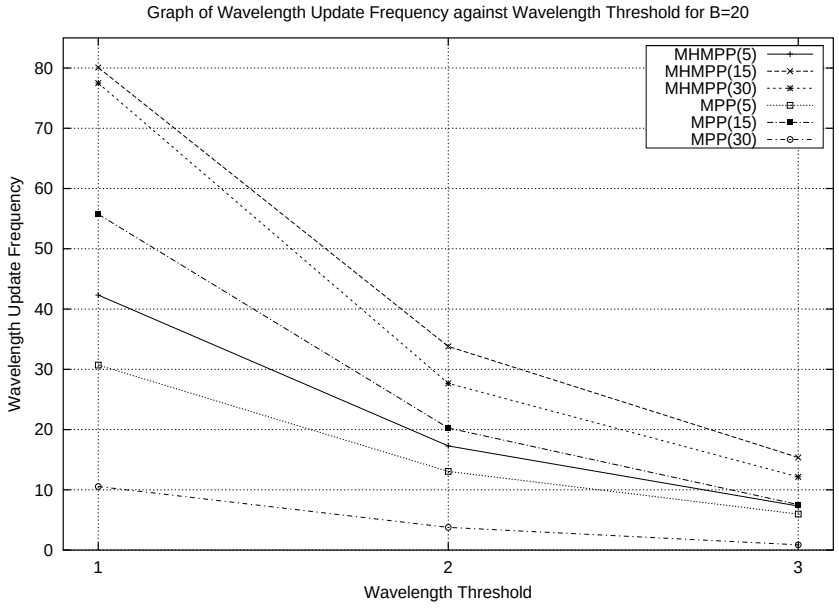

Figure 3.15: Graph of wavelength update frequency against wavelength threshold for B=20.

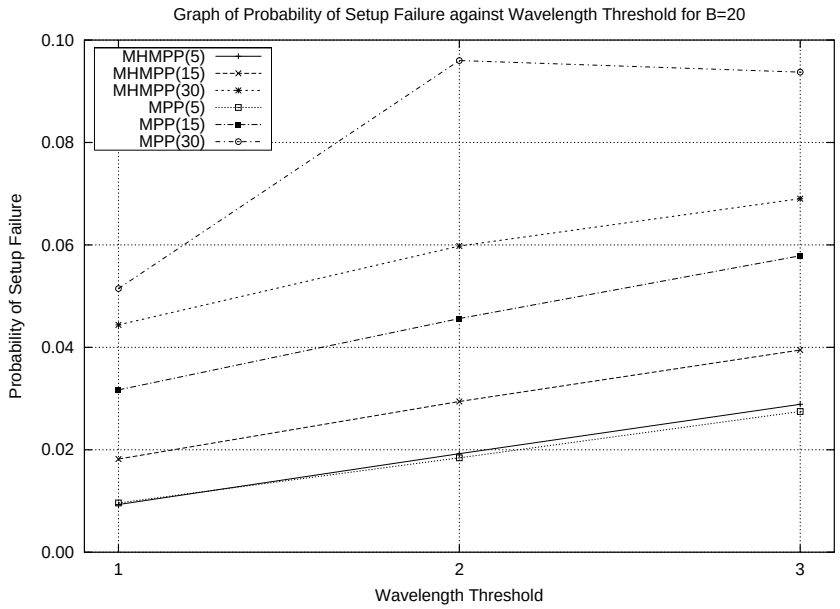

Figure 3.16: Graph of blocking probability due to setup failure against wavelength threshold for B=20.

3.5.4 Selection of update threshold

Determining the appropriate threshold values for a good network performance is an important issue. In the literature, analytical models have been developed to predict the blocking performance of the networks for some specific route and wavelength assignment methods in the optical layer independent of the client layer routing [44, 45, 46]. Such analytical models are usually developed for path networks and are then extended to mesh networks by making a number of assumptions. They do not consider the two-layer integrated routing problem with constraints such as bandwidth guarantees. Further, they do not consider any update schemes. Predicting the network performance through analytical models is extremely difficult when dynamic integrated constrained-routing is used on arbitrary mesh networks. It becomes much more difficult when dynamic adaptive routing is carried out at the optical and IP lay-

ers without using any predetermined path or wavelength selection policy. It is hard to estimate the distribution of traffic load over the network. Another difficulty is that the network topology changes whenever a lightpath is set up or torn-down. Considering the difficulty in developing analytical models without unreasonable assumptions, we use simulation results as a guideline to select threshold values.

In general, the network performance degrades and network overhead due to link state update decreases when the update threshold increases. An appropriate update threshold can be chosen by considering the tradeoff between network performance and network overhead. The tradeoff can be set by considering several constraints and requirements. In this section, we present a possible way of choosing the threshold values. We use the simulation results of algorithm MHMPP and MH-A. We choose these two algorithms because MHMPP reduces to MH-A when the probability values are set to 1 and $K = B = 0$. Here, *total blocking probability* (TBP) and *link state update frequency* (LSUF) are considered as the parameters of network performance and network overhead, respectively. Note that LSUF is the sum of both wavelength and bandwidth update frequencies.

To facilitate the update threshold selection procedure, from the practical point of view, we define some constant network parameters which may be determined by policy or estimated from real data. These parameters include the *mean traffic intensity* (TI_{mean}), *maximum traffic intensity* (TI_{max}), and *maximum total blocking probability* TBP_{max}. Let the total blocking probability and link state update frequency of algorithm MH-A be TBP_{mha} and $LSUF_{mha}$ respectively for traffic intensity value TI_{mean}. Since MH-A is the case when both K and B values for MHMPP are set to zero, TBP_{mha} and $LSUF_{mha}$ can be used to evaluate the effect of the update

thresholds on the network performance and network overhead.

Let S_K be the set of all possible wavelength update thresholds and S_B be the set of all possible bandwidth update thresholds under consideration. Let the triple $(K, B, TI\_)$ represent a combination of specific wavelength update threshold K, bandwidth update threshold B, and traffic intensity $TI\_$ values. The TBP and LSUF values of MHMPP for a given triple can be obtained from simulation results. We introduce a new metric called *loss factor*, which is defined as given below:

$$loss\ factor(K, B, TI_{mean}) = \frac{relative\ TBP\ loss\ (K, B, TI_{mean})}{relative\ LSUF\ gain\ (K, B, TI_{mean})} \tag{3.1}$$

where,

$$relative\ TBP\ loss\ (K, B, TI_{mean}) = \frac{TBP\ (K, B, TI_{mean}) - TBP_{mha}}{TBP_{mha}} \tag{3.2}$$

$$relative\ LSUF\ gain\ (K, B, TI_{mean}) = \frac{LSUF_{mha} - LSUF(K, B, TI_{mean})}{LSUF_{mha}} \tag{3.3}$$

For a particular update threshold pair (K, B), expression (2) reflects the loss of network performance in terms of TBP while expression (3) reflects the gain of network overhead in terms of LSUF when the network is operated at the load of mean traffic intensity. Hence, the value of loss factor obtained from (1), which is the ratio between relative TBP loss and relative LSUF gain, depicts the degree of loss when trading off network performance for network overhead. A small loss factor is preferred to achieve a good tradeoff.

From the perspective of loss factor, the best update threshold (K_a, B_a) can be determined by satisfying the following two conditions:

1. The loss factor associated with (K_a, B_a) should be the smallest among all possible (K, B) pairs so that $loss\ \ factor(K_a, B_a, TI_{mean}) = min\ (loss\ factor(K, B, TI_{mean})), \quad for\ all\ K \in S_K, B \in S_B.$

2. $TBP(K_a, B_a, TI_{max}) < TBP_{max}$. This condition limits the maximum total blocking probability in the network to a predetermined value TBP_{max}.

Now, we illustrate the above update threshold selection method. By observing the simulation results we set TI_{mean} and TI_{max} to 15 and 40 Erlangs, respectively. TBP_{max} is set to 0.1. Let S_B be [5,40] and S_K be [1,3]. Fig. 3.17 shows the graph of loss factor for varying update threshold pairs for the mean traffic intensity. From Fig. 3.17 we observe that the lowest loss factor is obtained when $(K = 1, B = 10)$ and from Fig. 3.11 we verify that the TBP for $(K = 1, B = 10)$ corresponding to the maximum traffic intensity of TI_{max} is below 0.1. Therefore, the update threshold values recommended by our method are $(K_a = 1, B_a = 10)$.

The update threshold values chosen by the above approach ensures an acceptably good tradeoff between network performance and network overhead. However, from the implementation point of view, the update threshold may be determined by several other factors. In practice, the two most significant ones are *service quality* and *network cost*, which are reflected by the network performance and network overhead, respectively. If network cost is considered as more important than quality, then the update threshold may be adjusted so that the network overhead is reduced to the required level at the cost of service quality, and vice versa.

3.5.5 Summary of Results

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

1. In the presence of inaccurate information, our algorithms MHMPP and MPP perform well in terms of the total blocking probability, blocking probability

due to routing failures and setup failures, bandwidth update frequency and wavelength update frequency.

2. Algorithms MHMPP and MPP perform very close to algorithm MH-A which uses accurate information. At the same time, bandwidth and wavelength update frequencies and hence the amount of update messages generated by our algorithms are significantly lower than that of MH-A.

3. When the traffic intensity is very low MPP performs better than MHMPP. When the traffic intensity increases MHMPP performs better than MPP in terms of the total blocking probability and blocking probability due to setup failures.

4. Increasing the update threshold increases the total blocking probability slowly. By carefully choosing bandwidth and wavelength update threshold values, the update overhead can be reduced significantly with a marginal increase in total blocking probability as well as setup failures.

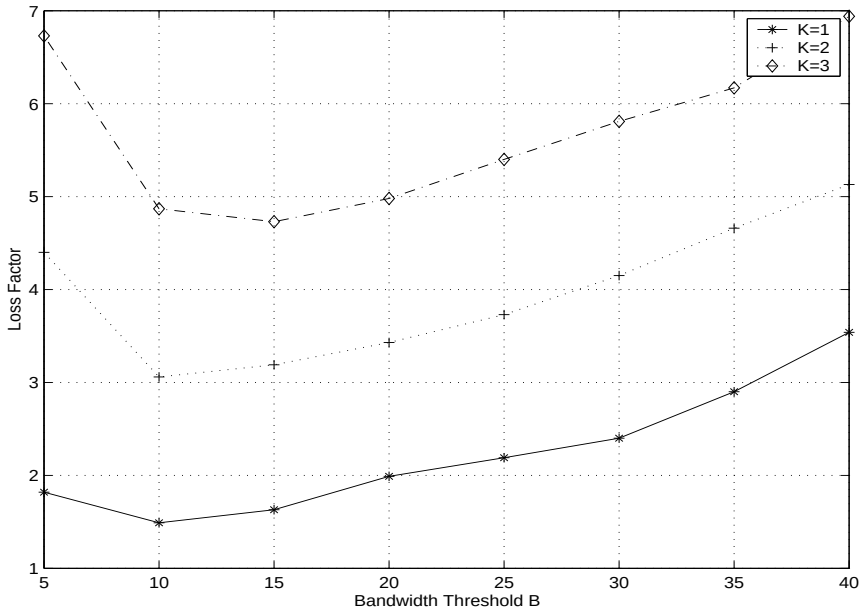

Figure 3.17: Graph of loss factor against bandwidth threshold for different wavelength threshold values for the traffic intensity of 15 Erlangs.

Chapter 4

Load Balancing Using Adaptive Alternate Routing in IP-over-WDM OBS Networks

4.1 Introduction

In this chapter, we investigate the problem of load balancing in *optical burst switching* (OBS) networks which pro-actively avoids network congestion by balancing the load between a node pair among multiple paths. This leads to reduced data channel contention. Although multipath load balancing has been studied in conventional MPLS-based IP networks [40], it has not been investigated in OBS networks. Due to the unique features of OBS networks such as no electronic buffering and no/limited optical buffering, the algorithms proposed for MPLS-based IP networks cannot be directly extended to OBS networks. For the same reason, delay in OBS networks is predictable. Since queuing delay is primarily restricted to the edge nodes, burst transfer delay is predominantly determined by propagation delay, which is fixed for a path. Therefore, delay is not as appropriate a performance metric as in MPLS-based IP networks to implement multipath load balancing. Instead, burst loss probability is a more appropriate metric to use in evaluating the impact of traffic load.

In OBS networks, the length of a path (in terms of hop count) is a critical factor which influences the burst loss probability. Since scheduling is required at each inter-

mediate node, bursts traversing a longer path will have a higher risk of encountering more contentions. Therefore, in this chapter, we consider particularly two-path load balancing in OBS networks wherein two link-disjoint paths are chosen as the alternative paths. We consider two path selection schemes to choose the alternative paths for a node pair. In the first scheme, the first two link-disjoint shortest-hop paths are chosen for a node pair. In the second scheme, the two link-disjoint widest shortest-hop paths are chosen for a node pair. The key idea is to let each ingress node periodically measure the impact of traffic load on the two alternative paths and dynamically decide the proportion of traffic load assigned to the paths based on these measurements. The proposed load balancing scheme has several attractive features. First, it operates dynamically without any prior knowledge of the traffic load distribution; second, it works in a distributed manner. All node pairs do load balancing independent of each other. Finally, it uses a simple measurement mechanism which does not incur much signaling and processing overhead. Through extensive simulation experiments, it is shown that the network performance in terms of burst loss probability is significantly improved by using the proposed load balancing scheme.

The rest of the chapter is organized as follows. In section 4.2, the overall picture of the proposed load balancing scheme is presented. The details of the proposed load balancing algorithm based on adaptive alternate routing are presented in section 4.3. The path selection schemes to select the alternative paths are described in section 4.4. The functions of traffic measurement, traffic assignment, and traffic distribution are described. The performance study is presented in section 4.5.

4.2 An Overview of the Proposed Load Balancing Scheme

In this section, we briefly describe the functioning of the proposed load balancing scheme based on adaptive alternate routing. For each node pair, two link-disjoint alternative paths are used for sending data bursts and control messages. With the use of GMPLS as the control plane in OBS networks, Label switched paths for the above pre-determined paths could be set up to facilitate transmission of control messages with reduced signaling and processing overhead. Each header containing the information about data channel scheduling is forwarded along the LSP by label swapping. For a given node pair, traffic loads which are the aggregation of IP flows arrive at the ingress node and are adaptively assigned to the two paths so that the loads on the paths are balanced. The objective of adaptive alternate routing is to avoid network congestion in order to achieve better network performance.

A time-window-based mechanism is adopted in which adaptive alternate routing operates in cycles of specific time duration which is called time window. Traffic assignment on the two paths are periodically adjusted in each time window based on the statistics of traffic measured in the previous time window. The time-window-based mechanism assumes that the traffic condition is predictable. This is reasonable and feasible for the following reasons. First, IP traffic changes relatively slowly in the scale of few minutes [48]. Therefore, the traffic condition is more likely to be predictable based on the traffic statistics measured in the previous time window. Second, recent studies have shown that traffic in IP networks often exhibits *long-range dependence* (LRD), with the implication that congested periods can be quite long [49]. Since bursts in OBS networks are assembled from IP flows, we expect that

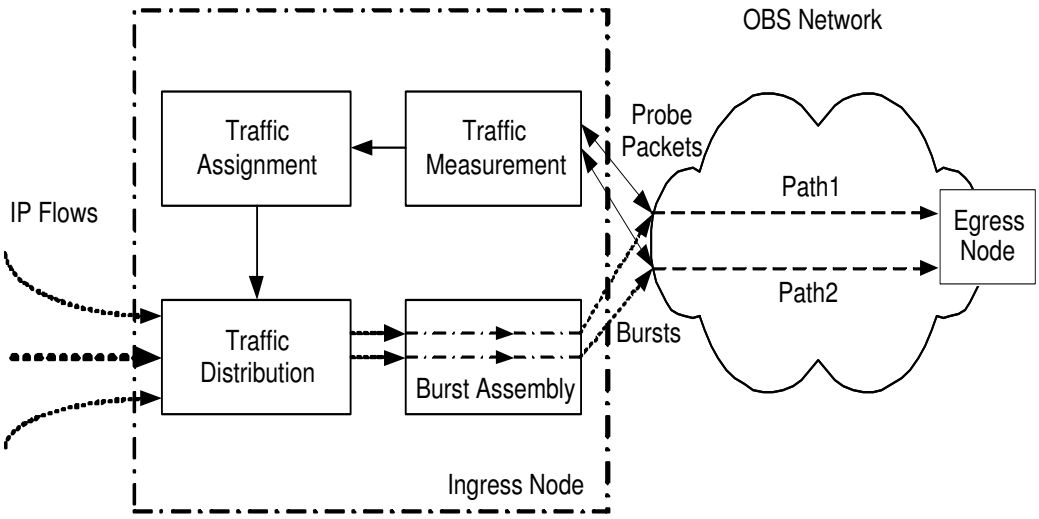

Figure 4.1: Functional units of the load balancing scheme.

the traffic congestion condition is predictable.

Figure 4.1 shows the functional block diagram of the proposed load balancing scheme for a specific node pair. At the ingress node, four functional units — traffic measurement, traffic assignment, traffic distribution and burst assembly — work together to achieve load balancing. Traffic measurement is responsible for collecting traffic statistics by sending probe packets to each of the two paths periodically. The collected information is then used to evaluate the impact of traffic load on the two paths. Based on the measurements and the hop difference between the two alternative paths, traffic assignment determines the proportion of traffic allocated to each of the two paths in order to balance the traffic loads on the two paths by shifting a certain amount of traffic from the heavily-loaded path to the lightly-loaded path. Traffic distribution plays the role of distributing the IP traffic that arrives at the ingress node to the two paths according to the decisions made by traffic assignment. Finally, bursts are assembled from packets of those flows assigned to the same path.

4.3 Adaptive Alternate Routing Algorithm

In this section, we explain in detail the proposed two-path load balancing algorithm —*adaptive alternate routing algorithm* (AARA), which performs several functions such as traffic measurement, traffic assignment and traffic distribution. As stated earlier, AARA is run for each node pair independent of other node pairs. Without loss of generality, we explain the working of AARA for a specific node pair s. Two link-disjoint paths are pre-determined as the alternative paths. The details of the alternative-path selection schemes are described in section 4.4.

4.3.1 Notations

For ease of explanation, we define:

$path_p$: primary path.

$path_a$: alternate path.

$length_p$: hop count of the primary path.

$length_a$: hop count of the alternate path.

$T(i)$: ith time window.

$loss_p(i)$: mean burst loss probability on the primary path in time window $T(i)$.

$loss_a(i)$: mean burst loss probability on the alternate path in time window $T(i)$.

P_p^i: proportion of traffic load assigned to the primary path in time window $T(i)$

P_a^i: proportion of traffic load assigned to the alternate path in time window $T(i)$.

$(P_p, P_a)^i$: combination of P_p^i and P_a^i which represents the traffic assignment in time window $T(i)$.

We note that $length_p \leq length_a$ and $P_p^i + P_a^i = 1$.

4.3.2 Traffic Measurement

The traffic measurement process is invoked periodically in each time window. We use 'mean burst loss probability' as the measured performance metric. The purpose of traffic measurement is to collect traffic statistics for each path by sending probe packets and then calculate the mean burst loss probability to evaluate the impact of traffic load. To achieve this, we set a counter at each node in the network. The counter is used to record the number of bursts dropped at the node since the last probe was made. The recorded data is classified on a per-path basis (e.g., based on a 2-tuple $\langle$ node pair, path $\rangle$). Since the traffic measurement process is similar in each time window, we illustrate the whole process for a specific time window $T(i)$. Let p_i be the i^{th} node on $path_p$. Let the recorded value of burst loss at node p_i be $count_{p_i}(s, path_p)$. Let a_j be the j^{th} node on $path_a$. Let the recorded value of burst loss at node a_j be $count_{a_j}(s, path_a)$. Let the probe messages for the two paths be $probe(s, dropped_p)$ and $probe(s, dropped_a)$, where $dropped_p$ and $dropped_a$ record the number of bursts dropped on the two paths since the last probe was made. Initially, both $dropped_p$ and $dropped_a$ are set to zero. Let probe-success-acknowledgement messages for the two paths be $probe_{success}(s, dropped_p)$ and $probe_{success}(s, dropped_a)$.

The ingress node, egress node and all the intermediate nodes participate in the traffic measurement process. The actions performed by each of the above nodes are given below.

Ingress node:

1. At the beginning of $T(i)$, the ingress node starts to record the total number of bursts sent to the two paths, $total(s, path_p)$ and $total(s, path_a)$, respectively.

2. At the end of $T(i)$, the ingress node sends out probe packets, $probe(s, dropped_p)$ and $probe(s, dropped_a)$, to $path_p$ and $path_a$ separately to collect the record at each intermediate node.

3. After receiving the probe-success-acknowledgement messages, $probe_{success}(s, dropped_p)$ and $probe_{success}(s, dropped_a)$, sent from the egress node, the mean burst loss probability on the two paths, $loss_p(i)$ and $loss_a(i)$, in time window $T(i)$ are calculated using the following formulas:

$$loss_p(i) = \frac{dropped_p}{total(s, path_p)};$$
$$loss_a(i) = \frac{dropped_a}{total(s, path_a)}.$$

Intermediate node:

1. At the beginning of $T(i)$, i.e., immediately after the last "probe", the counters are reset to zero at each node such that

$$count_{p_i}(s, path_p), count_{a_j}(s, path_a) = 0.$$

2. When a burst is dropped at node p_i that belongs to $path_p$, update

$$count_{p_i}(s, path_p) = count_{p_i}(s, path_p) + 1.$$

3. When a burst is dropped at node a_j that belongs to $path_a$, update

$$count_{a_j}(s, path_a) = count_{a_j}(s, path_a) + 1.$$

4. When the probe packet $probe(s, dropped_p)$ arrives at node p_i, update

$$dropped_p = dropped_p + count_{p_i}(s, path_p).$$

5. When the probe packet $probe(s, dropped_a)$ arrives at node a_j, update

$$dropped_a = dropped_a + count_{a_j}(s, path_a).$$

Egress node: After receiving the probe packets, the egress node sends out the probe-success-acknowledgement messages, $probe_{success}(s, dropped_p)$ and $probe_{success}(s, dropped_a)$, to the ingress node.

The traffic measurement is carried out on a per-path basis. With the support of GMPLS, LSPs could be setup for each path. The intermediate nodes perform simple operations for each LSP passing through them whenever bursts are dropped. Therefore, such operations do not add much complexity to the nodes. The time window size should be properly chosen so that it is considerably larger than the longest *round trip time* (RTT) in the network. This will help reduce the impact of the probing delay on the accuracy of measurements and hence the performance.

4.3.3 Traffic Assignment

Traffic assignment adaptively determines the proportion of traffic allocated to each of the two paths in each time window. The traffic assignment decision is determined by two parameters: measured value of the mean burst loss probability on the two paths and the hop difference between the two paths. The measured mean burst loss probability returned by traffic measurement in the previous time window is used to estimate the impact of traffic loads on the two paths. These loads are balanced in the current time window. The basic idea is to shift a certain amount of traffic from the heavily-loaded path to the lightly-loaded path so that traffic loads on the two paths are balanced.

Hop count is an important factor in OBS networks for the following two reasons:

1. Since burst scheduling is required at each intermediate node traversed, a longer path means a higher possibility that a burst encounters contention.
2. A longer path consumes more network resources which results in low network efficiency.

Thus, network performance may become poorer if excessive traffic is shifted from the shorter path to the longer path even though the longer path is lightly loaded. To avoid this, we set a protection area PA which is used to determine when traffic should be shifted from the shorter path ($path_p$) to the longer path ($path_a$). Let the measured mean burst loss probability difference between the two paths ($loss_p(i) - loss_a(i)$) be Δp. If and only if Δp is beyond PA, traffic can be shifted from the shorter path ($path_p$) to the longer path ($path_a$). Let the hop difference between the two paths ($length_a - length_p$) be Δh. PA is given by $PA = \Delta h \times \tau$, where τ is a system control parameter. By doing so, a good tradeoff is achieved between the benefit of using a lightly-loaded path and the disadvantage of using a longer path.

We illustrate the traffic assignment process in a specific time window $T(i)$. Initially, in time window $T(0)$, the traffic is equally distributed to the two paths. Let the mean burst loss probability of the two paths returned by traffic measurement in time window $T(i-1)$ be $loss_p(i-1)$ and $loss_a(i-1)$, respectively. Then $\Delta p = loss_p(i-1) - loss_a(i-1)$. Let the traffic assignment in time window $T(i-1)$ be $(P_p, P_a)^{i-1}$. The following procedure is used to determine $shiftP$ (the amount of traffic to be shifted) and the new traffic assignment $(P_p, P_a)^i$ in time window $T(i)$.

1. if $\Delta p \geq PA$, then traffic is shifted from $path_p$ to $path_a$,

$$shiftP = P_p^{i-1} \times (\Delta p - PA);$$

$$P_p^i = P_p^{i-1} - shiftP;$$

$$P_a^i = P_a^{i-1} + shiftP;$$

else if $\Delta p < PA$ and $\Delta p \geq 0$, then traffic assignment remains the same, i.e., $(P_p, P_a)^i = (P_p, P_a)^{i-1}$;

else if $\Delta p < 0$, then traffic is shifted from $path_a$ to $path_p$,

$$shiftP = P_a^{i-1} \times \mid \Delta p \mid;$$

$$P_p^i = P_p^{i-1} + shiftP;$$

$$P_a^i = P_a^{i-1} - shiftP;$$

end if.

2. Send the new traffic assignment information to the traffic distribution unit.

3. At the end of time window $T(i)$, receive the values of $loss_p(i)$ and $loss_a(i)$ from the traffic measurement unit.

4. Let $i = i + 1$ and go to step 1.

4.3.4 Traffic Distribution

The traffic distribution function distributes IP flows arriving at the ingress node to the two paths based on the traffic assignment decision. One way to distribute the traffic is on a per-packet basis. Each packet is distributed to $path_p$ with probability P and to $path_a$ with probability $1 - P$. This, however, may cause the reordering of packets for each flow which is undesirable for TCP connections. Another way 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. Unlike per-packet based traffic distribution, the

order of packets is preserved in this approach. Reordering of packets is needed only if the flows are shifted from a longer path to a shorter path when traffic assignment is adjusted. Further, although the mapping information between flow and path needs to be maintained, it is restricted to the ingress node. Therefore, from the perspective of buffer requirements at the egress nodes, per-flow based traffic distribution is more suitable than per-packet based traffic distribution in OBS networks.

4.4 Alternative-Path Selection Scheme

In this section, we present two different kinds of alternative-path selection schemes which are based on *shortest-hop path routing* (SHPR) and *widest-shortest-hop path routing* (WSHPR). The purpose of alternative-path selection is to choose two link-disjoint paths, primary path $path_p$ and alternate path $path_a$, for each node pair. The alternative paths for each node pair are pre-determined based on the network topology and estimated traffic demands. The traffic demands may be estimated from the long-time traffic statistics. These alternative paths are used by the adaptive alternate routing algorithm to distribute traffic load on them. Here, we assume a central server operates off-line to choose the alternative paths a priori.

Consider a network with a set E of links and a set S of node pairs. We model the network as a directed graph comprising vertices and edges where vertices correspond to network nodes and edges correspond to network links. We assume traffic demands for each node pair $s \in S$ is known a priori. Each of the node pair s has a total input traffic load r_s. We let the path returned by the shortest path selection algorithm consist of I links. We define positive constants k_1 and k_2, $k_1 >> k_2$, such that $k_1 x_1 > k_2 y_2$ where x_1 is the smallest possible non-zero value and y_2 is the largest

possible non-zero value in a function of the form $k_1x + k_2y$.

By considering different cost metrics, different weights are assigned to each link and the minimum cost path generated using a Dijkstra-like shortest path selection algorithm will have varying performance and optimize a certain path-cost metric. In our schemes, we use hops and link load as cost metrics. Hops refer to the number of optical links traversed by a path. Fewer network resources are consumed if a path traverses fewer links. The link load on a link $l \in E$ is the sum of traffic demands on all paths that traverse link l. By properly choosing the paths to distribute the traffic load in the network, congestion of certain links may be avoided. The link load for link $l \in E$ is denoted by $t(l)$.

4.4.1 SHPR Based Alternative-Path Selection

In this scheme, both primary and alternate paths are chosen by shortest-hop path routing. In SHPR, only hops are considered as the cost metric and the path with the minimum number of physical links is chosen. The cost of the path returned is given by

$$path\_cost = I$$

where I is the length of the path.

For a node pair, the primary path is chosen first followed by the alternate path. By assigning a weight of 1 to all the links and running a shortest-path selection algorithm such as Dijkstra algorithm, the primary path $path_p$ is chosen. Let $path_p$ consists of a sequence of links $(l_1, l_2, ..., l_I)$ such that $l_i \in E$, $i = 1, 2, ..., I$. After that, we remove every link l_i constituting the primary path from link set E and get a residual link set E'. By running shortest-path routing algorithm on the residual link

set E', an alternate path $path_a$ is chosen such that each constituted link of the path belongs to E'.

4.4.2 WSHPR Based Alternative-Path Selection

In this scheme, both the primary and alternate paths for every node pair $s \in S$ are chosen by WSHPR. In WSHPR, both hops and link load are considered as the cost metrics. The WSHPR chooses a path that minimizes the maximum link load on it among all the shortest hop paths. The maximum link load on a path is the maximum of the load on the links traversed by it. We define the path cost as

$$path\_cost = \sum_{i=1}^{I} k_1 + max_{i=1}^{I} k_2 t(i)$$

The WSHPR-based alternative-path selection is carried out in two phases: primary path selection and alternate path selection. In each phase, the paths are selected between node pairs one by one in decreasing order of input traffic load r_s. In the primary path selection phase, primary paths for each node pair $s \in S$ are selected first in the pre-determined order. Then the alternate path selection phase is invoked for selecting the alternate paths for all node pairs. Like SHPR-based alternative-path selection, the primary path is chosen by running WSHPR on the whole set of links E while the alternate path is determined by running WSHPR on the residual set of links E' which excludes the links on the corresponding primary path. In both phases, link loads are updated once a path is determined. When a path between node pair s consisting of a set L of links is selected, the link load of $t(l)$ of each link $l \in L$ belonging to this path is updated such that

$$t(l) = t(l) + r(s), \text{ for all } \ l \in L,$$

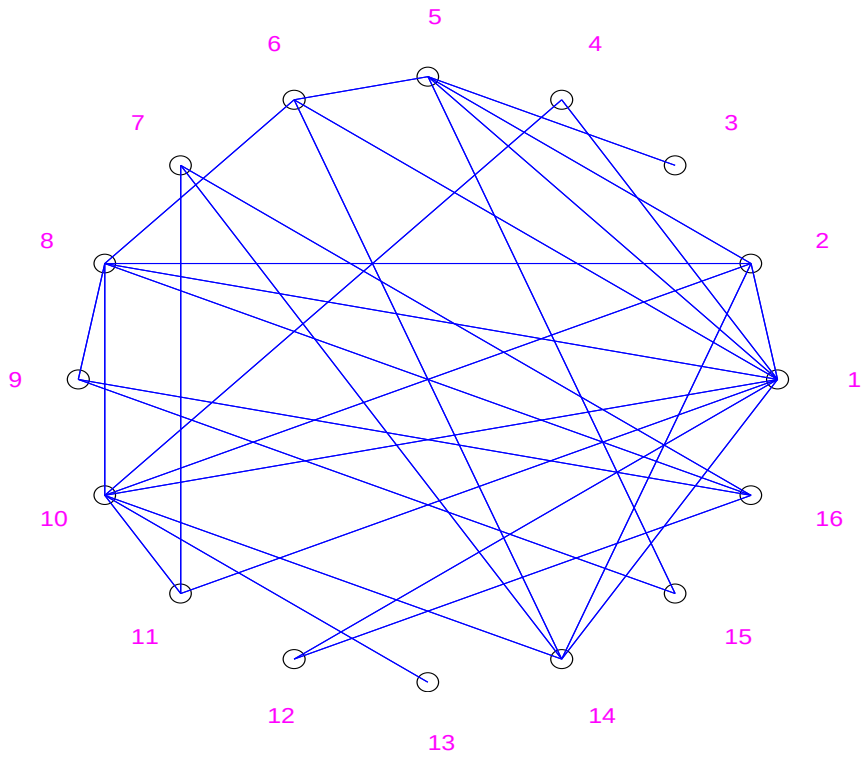

Figure 4.2: 16-node randomly generated network.

where $r(s)$ is the total input traffic load for node pair s as mentioned before.

4.5 Performance Study

In this section, we study the performance of our proposed AARA algorithm through extensive simulations on a randomly generated network shown in Fig. 4.2. The network has 16 nodes, 31 bidirectional links and 8 data channels per fiber. The transmission rate per data channel is $1Gb/s$. A bidirectional link comprises two unidirectional fibers in opposite direction. We use the basic void filling data channel scheduling algorithm presented in [16] to schedule the data channels for the data bursts. The burst assembly time is fixed at $60\mu s$.

In order to get more realistic results, we use a LRD traffic model in our study. In this LRD traffic model, traffic that arrives at each node pair in the network is the aggregation of multiple IP flows [51]. 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 some peak rate p packets/sec. The OFF times are also Pareto distributed with mean M and shape parameter γ. The following values are used for the Pareto-ON/OFF flows in our simulations: $N = 20$, $\beta = 1.2$, $M = 32,000\mu s$, $\gamma = 1.1$, $p = 640$. The packet length is assumed to be $400bytes$. The transmission rate per flow r is fixed at $1Mb/s$.

Flows arrive at a node pair according to a Poisson process with mean λ. The holding time of a flow is exponentially distributed with mean $1/\mu$. In our simulations, traffic load is measured as the number of flows that arrive per second (*flow arrival rate* (FAR)). Following [50], the offered network traffic load ρ can be approximately computed using the following formula:

$$\begin{aligned} \rho &= (N1 \times N2 \times h \times r)/(C \times 2 \times L \times w) \\ &= \lambda \times (\frac{N2 \times h \times r}{\mu \times C \times 2 \times L \times w}). \end{aligned}$$

$N1$ is the mean number of active flows per node pair at steady state, which is given by λ/μ according to Little's theorem; $N2$ is the total number of node pairs in the network; h is the mean number of hops per path; L is the total number of links in the network; C is the total capacity per data channel; and w is the total number of data channels per fiber. The values for the above parameters used in the simulations are: $N2 = 120$, $r = 1Mb/s$, $L = 31$, $\mu = 1s$, $C = 1Gb/s$ and $w = 8$. h is taken as

2.005, which is the mean shortest path length in the simulation network.

We set the system control parameter τ to be 0.01, which is used to avoid the excessive traffic shifting from the shorter path to the longer path that may result in performance degradation due to increased consumption of network resources and increased chance of burst dropping on a longer path.

We use burst loss probability and mean hop-length 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 a burst.

We also implemented two other algorithms, *shortest path routing* (SPR) and *static alternate routing* (SAR) to demonstrate the effectiveness of our algorithm in dealing with load balancing. For algorithm SPR, all the bursts are transmitted along the fixed shortest path. For algorithm SAR, bursts are transmitted along the two pre-determined link-disjoint alternative paths. It differs from AARA in that the traffic flows are equally distributed between the two paths and remain unchanged. Since the choice of alternative path selection scheme impacts on the performance of load balancing algorithm, we apply different alternative path selection schemes to AARA and SAR. Therefore, we study two AARA algorithms based on the two different alternative path selection schemes, *SHPR-based AARA* (SHPR-AARA) and *WSHPR-based AARA* (WSHPR-AARA) and two SAR algorithms, *SHPR-based SAR* (SHPR-SAR) and *WSHPR-based SAR* (WSHPR-SAR) in the simulations.

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 conditions. In an identical traffic demand, the traffic that arrives at each node pair is homogeneous, i.e., all flows arrive at the same rate and the flow

arrival rate is derived from the same Poisson process with a fixed mean value. In a non-identical traffic demand, the traffic that arrives at each node pair is heterogeneous where an individual flow arrival rate is derived from the Poisson process with a different mean value. Since there are 240 node pairs in the simulation network with 8 wavelengths per link and the traffic is generated from the flow level, a huge number of packets and bursts need to be processed in each simulation experiment. Therefore, we provide the results for $4,000,000\mu s$ of simulated time. We expect that the 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. Next we study the impact of the time window size. Then we evaluate the impact of the alternative path selection. Finally, we study the performance with non-identical traffic demand.

4.5.1 Identical Traffic Demand

4.5.1.1 Effect of Traffic loading

In this section, SHPR-based alternative path selection scheme is applied to all simulations. Figure 4.3 shows the burst loss probability with varying traffic load per node pair for time window size $T = 100,000\mu s$. Figure 4.4 shows the percentage of burst loss performance improvement achieved by the proposed load balancing algorithm, SHPR-AARA, in comparison with SPR. We observe that the proposed algorithm performs much better than both SPR and SHPR-SAR. The performance improves by up to 68% in comparison with SPR. Since SHPR-AARA performs load balancing in the network, congestion is reduced. As a result, the bursts dropped due to contention

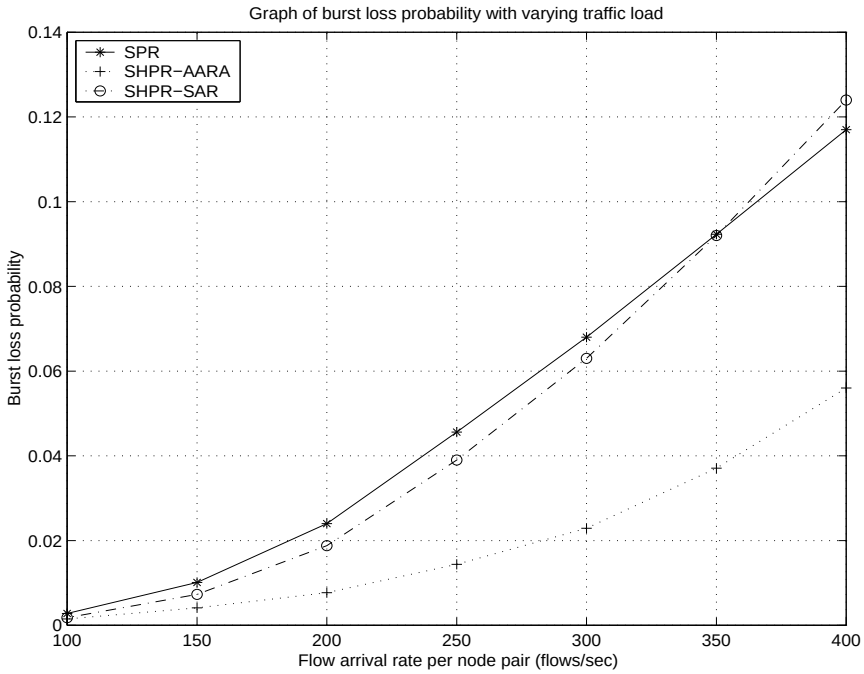

Figure 4.3: Graph of burst loss probability against traffic load.

are reduced. On the other hand, although algorithm SPR always chooses the shortest paths to use, the number of dropped bursts is larger than that of SHRP-AARA which may choose longer paths. Although SHPR-SAR distributes traffic to the two paths, it performs worse than SHPR-AARA because it fails to keep track of the differing congestion states on the link-disjoint paths.

From Fig. 4.4, we observe that the percentage of performance improvement first increases when traffic load increases but beyond a certain point it decreases instead. When the traffic load is light, network resource is abundant and is available for balancing the traffic load. In this case, the performance of SHPR-AARA increases with increased traffic load as load balancing becomes more useful in mitigating the effects of congestion. But when the traffic load increases beyond a certain value, the performance improvement achieved by SHPR-AARA decreases because of resource shortage.

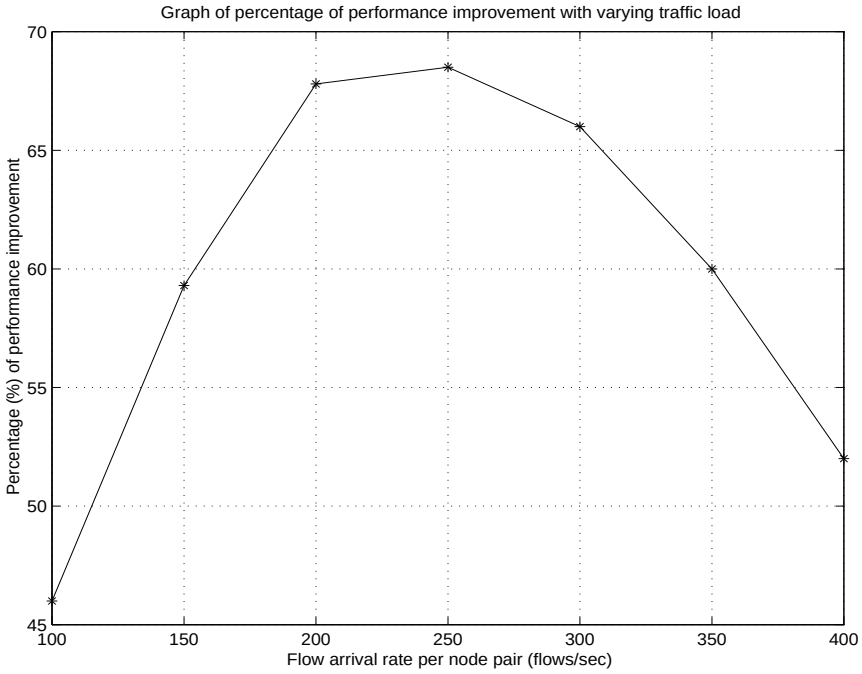

Figure 4.4: Graph of percentage of performance improvement against traffic load.

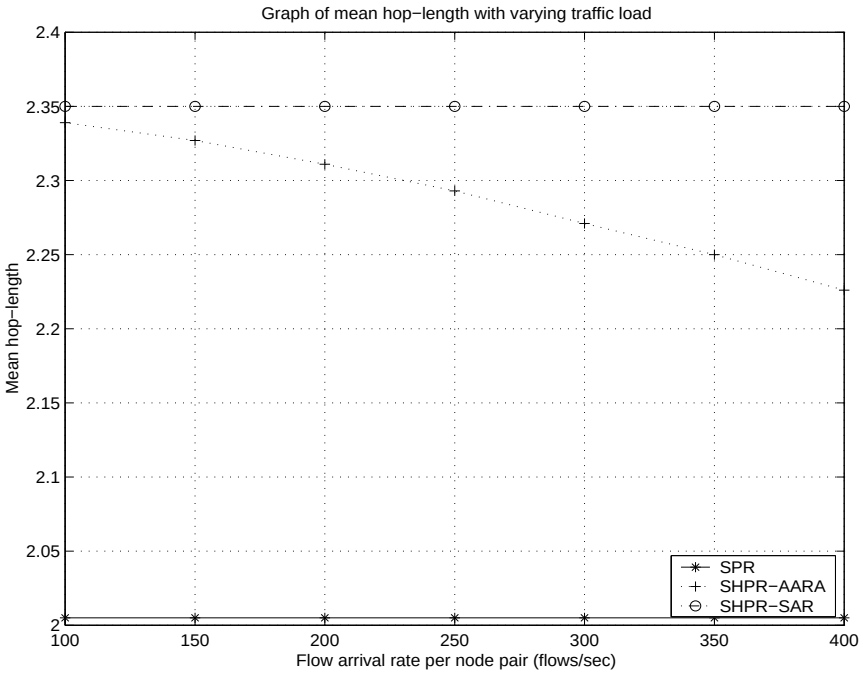

Figure 4.5: Graph of mean hop-length against traffic load.

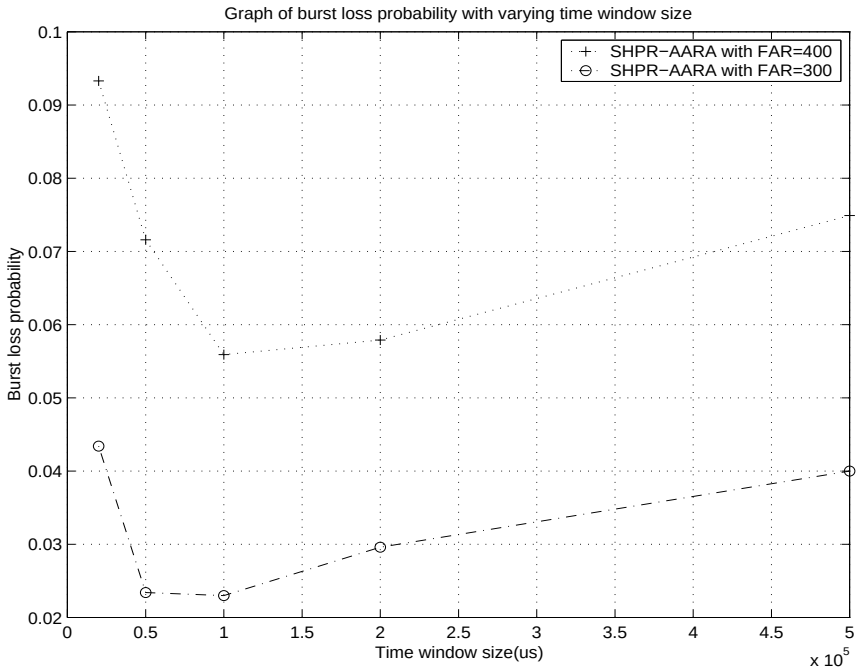

Figure 4.6: Graph of burst loss probability against time window size (μs).

Figure 4.5 shows the mean hop-length traversed by a burst with varying traffic load per node pair. The mean hop-length may reflect delay, signaling overhead, and initial offset time in the network. We observe that the mean hop-length for SHPR-AARA is slightly larger than that for SPR. This implies that the additional delay, signaling overhead, and initial offset time introduced by SHPR-AARA is rather low when compared to the performance improvement achieved. We also observe that the mean hop-length for SHPR-AARA decreases when traffic load increases. This is because SHPR-AARA tends to prefer the shorter path when traffic load increases. The mean hop-length for SHPR-SAR is larger than the other two since it treats the two paths equally. Therefore on average it uses the longer path more often than SHPR-AARA since it does not adapt to the traffic load.

4.5.1.2 Effect of Time Window Size

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

Figure 4.6 plots the burst loss probability with varying time window size. We observe that the burst loss probability first decreases and then increases with increasing time window size. The efficiency of SHPR-AARA directly depends on the accuracy of the traffic measurements and when the time window size is small, the collected traffic statistics reflect only the short-term traffic load conditions and may not be accurate. As a result, load balancing based on it does not work very well. Further, a small time window size results in frequent adjustments of traffic assignments, which may make the network unstable. When the time window size becomes too large, the performance also starts to degrade. Here, the large window size renders the algorithm incapable of tracking dynamic changes in traffic loads and yields smoothed out traffic loads which are not reflective of the actual load conditions.

4.5.1.3 Effect of Alternative Path Selection

We evaluate the impact of different alternative path selection schemes —SHPR-based and WSHPR-based— on the performance of the load balancing algorithm.

Figure 4.7 plots the burst loss probability with varying traffic load per node pair for time window size $T = 100,000\mu s$. We observe that in general the algorithms based on WSHPR perform better than the algorithms based on SHPR. The performance of WSHPR-SAR even outperforms that of SHPR-AARA. The reason behind this is that WSHPR considers the bottleneck link in the network and avoids congesting such links

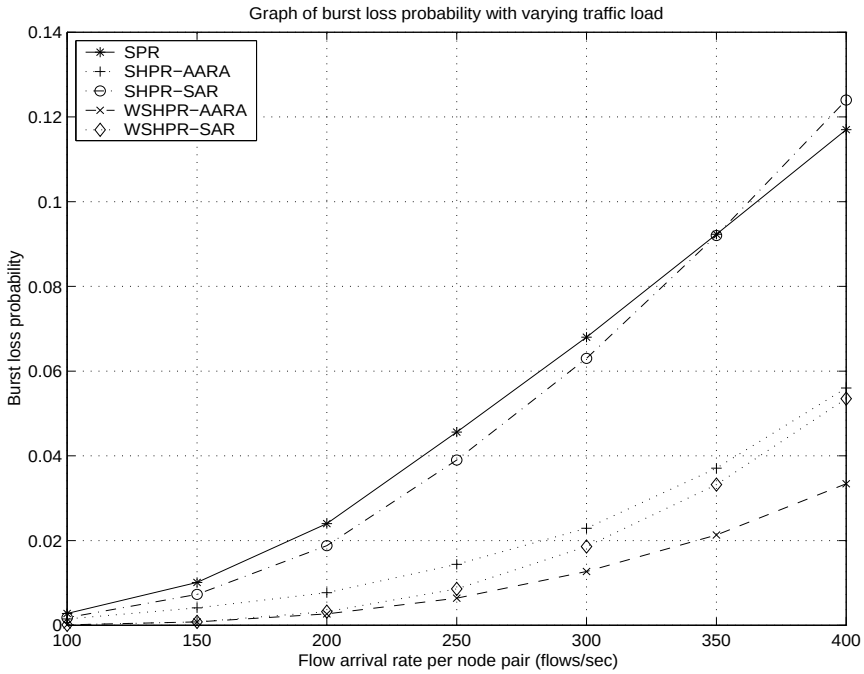

Figure 4.7: Graph of burst loss probability against traffic load.

while determining the alternative paths. However, WSHPR requires prior knowledge of the traffic demands of the whole network which may not adapt to the dynamic traffic demands in the real network. WSHPR-AARA performs best among all the algorithms since it considers load balancing through path selection as well as dynamic load balancing through adaptive load distribution. However, WSHPR-AARA may not perform well when the traffic demands vary from time to time deviating from the estimated long-term average demands. On the other hand, SHPR-AARA preserves the advantage of dynamic adaptation without requiring a prior traffic load information and while having reasonably good performance.

4.5.2 Non-identical Traffic Demand

In this section, we study the applicability of AARA in balancing non-identical traffic demand. This set of simulations are based on SHPR. We study the performance

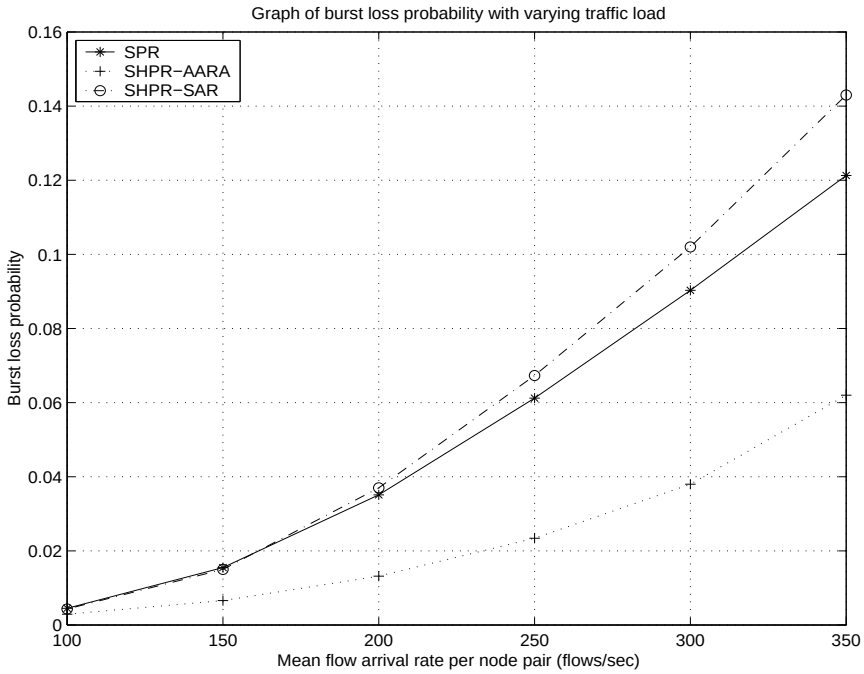

Figure 4.8: Graph of burst loss probability for various non-identical traffic demands.

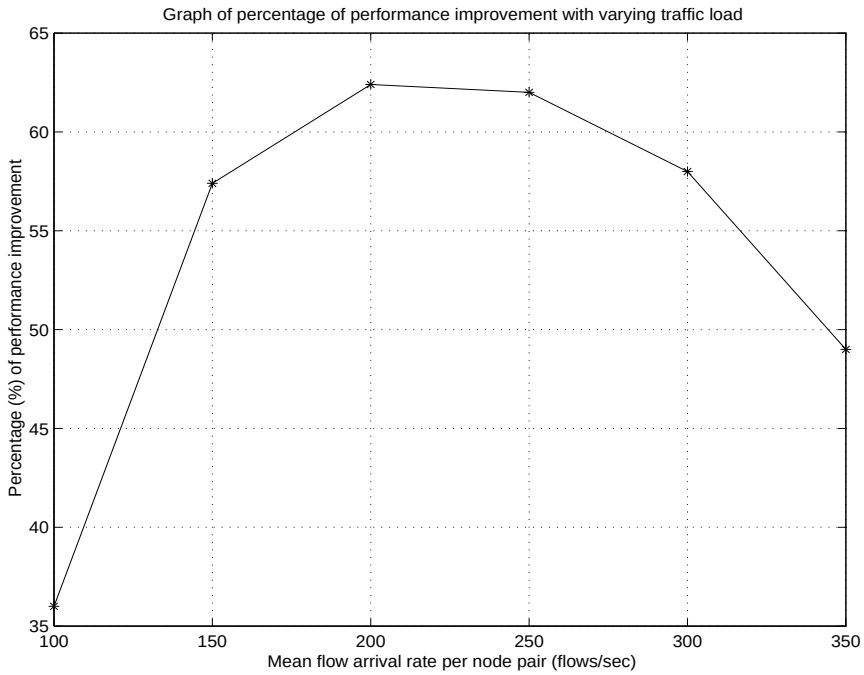

Figure 4.9: Graph of percentage of performance improvement for various non-identical traffic demands.

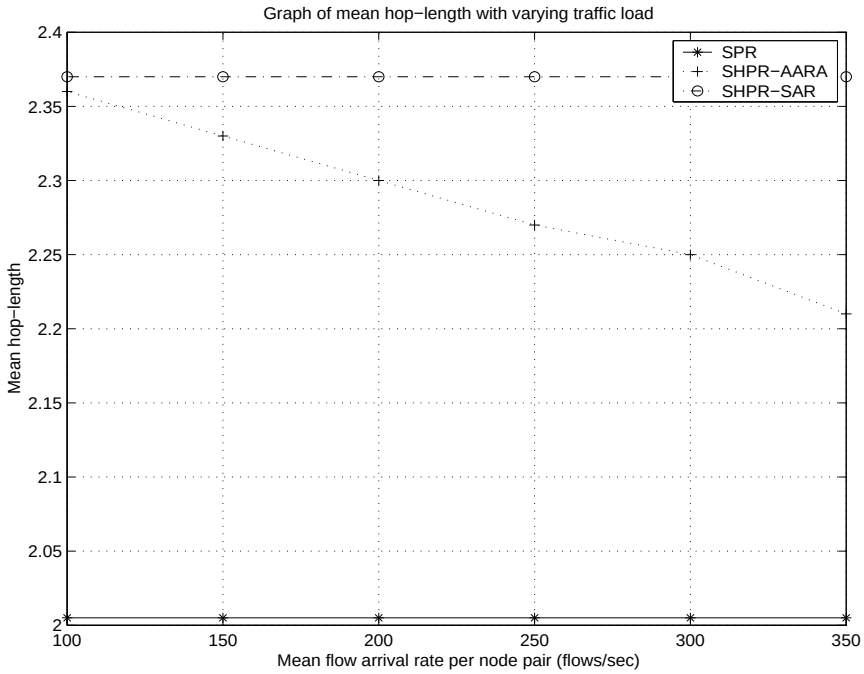

Figure 4.10: Graph of mean hop-length for various non-identical traffic demands.

for six different non-identical traffic demands. In a non-identical traffic demand, the flow arrival rate for a node pair is randomly selected from a set of flow arrival rates $\{r1, r2, r3, r4, r5\}$ with equal probability. The traffic load is measured as the mean flow arrival rate which is given by the average of the five flow arrival rates. The values for r1, r2, r3, r4, and r5 are chosen such that the mean flow arrival rate per node pair ranges from 100 to 350 flows per second.

Figure 4.8 shows the burst loss probability for six different non-identical traffic demands. Figure 4.9 shows the percentage of burst loss performance improvement in comparison with SPR. Figure 4.10 shows the mean hop-length. From these figures, we make similar observations as in the case of identical traffic demand. This proves that SHPR-AARA works well even when the traffic loads are not evenly distributed in the network and verifies that SHPR-AARA is applicable to different traffic scenarios. In the same way we expect that the performance trend of WSHPR-AAPR is similar

to the case of identical traffic demand.

4.5.3 Summary of Results

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

1. By doing load balancing, AARA can significantly improve performance in terms of burst loss probability with a marginal increase in mean hop length (in terms of delay), signaling overhead and initial offset time.
2. AARA performs well under different traffic conditions.
3. By carefully choosing the time window size the burst loss probability can be reduced significantly.
4. WSHPR based algorithms perform better than SHPR based algorithms in general. However, from the practical point of view, SHPR-AAPR is better than others since it keeps the dynamic property without needing a prior traffic load information.

Chapter 5

Conclusions

In this thesis, dynamic routing and load balancing issues in IP-over-WDM networks have been studied. Firstly, the problem of dynamically routing bandwidth-guaranteed LSPs in integrated IP-over-WDM network with inaccurate link state information has been investigated. Then the dynamic load balancing scheme based on adaptive alternate routing has been proposed for IP-over-WDM OBS networks.

Due to the non-real-time update of the link state information, inaccuracies exists in both the bandwidth and wavelength link state information in integrated IP/WDM networks. We have presented a practical bandwidth and wavelength update model wherein the update of such link state information is based on certain threshold instead of any change in the link state. Since uncertainty arises from inaccurate information, we have developed a probabilistic method to represent the uncertainty of the link state parameters, namely bandwidth and wavelengths. Uncertainty reflects the probability that the required resource can be guaranteed. Then, we have defined cost metrics based on uncertainty in addition to hops. We have developed two routing algorithms, namely MPP and MHMPP, considering the uncertainty of the link state parameters, each of which optimizes a certain cost metric. For MPP, only the uncertainty is considered and the most probable path is selected among all possible paths. For MHMPP, both hops and uncertainty are considered and the most proba-

ble path among the minimum-hop paths is selected. We have demonstrated that both algorithms can effectively reduce the impact of inaccurate link state information and perform very well through extensive simulation experiments. From the simulation results we have made several useful observations. We have also presented a method to choose appropriate threshold values to achieve a desired tradeoff between network performance and overhead.

Load balancing is more important in OBS networks due to the lack of optical memory devices. Also fixed shortest path which is currently widely adopted in OBS networks for transporting bursts between a node pair limits the possibility that network resources are efficiently used since it fails to take advantage of the multiple paths existing between node pairs. Based on the above consideration, we have investigated dynamic load balancing in OBS networks. We have developed a load balancing algorithm called adaptive alternate routing which dynamically balances the traffic load on two paths based on measurements. In our scheme, load balancing is achieved by the cooperation of four functional units—traffic measurement, traffic assignment, traffic distribution and burst assembly. We have presented a time-window-based mechanism which works in conjunction with adaptive alternate routing. In the time-window-based mechanism, adaptive alternate routing works in cycles of time duration called time windows. The proposed load balancing scheme has several attractive features. First, it operates dynamically without any prior knowledge of the traffic load distribution. Second, it works in a distributed manner. Finally, it uses a simple measurement mechanism which does not incur much signaling and processing overhead. We have demonstrated that our algorithm can effectively balance the traffic load and reduce burst loss significantly through extensive simulation experiments. Also we

have verified that our algorithm is applicable to different traffic scenarios. From the simulation results we have made several useful observations.

We now present the possible research directions for future investigation. Developing routing algorithms based on inaccurate state information in networks with limited number of ports and O-E-O constraints is an important problem to be studied. Further, a more realistic update model considering the wavelength correlation instead of the wavelength independence assumption could be developed to get more accurate results. The multipath load balancing problem to support multiple classes of services with different QoS requirements is a challenging problem to be studied. Analysis of buffer requirements and admission control at the edge nodes are also important problems to be studied. Due to arbitrary and irregular network topology, different paths belonging to different node pairs may share links. Then congestion may occur when several paths shift traffic simultaneously to a shared link. On the other hand, it is highly possible that there may be some undesirable synchronized switching between alternative paths of traffic in different node pairs, analogous to the synchronization problem in TCP. Further study could consider shared link as a factor when doing load balancing.

Bibliography

[1] A. Banerjee *et. al.*, "Generalized Multiprotocol Label Switching: An overview of Routing and Management Enhancements," *IEEE Communications Magazine*, pp. 144-150, January 2001.

[2] C. Assi, Abdallah Shami, and M. A. Ali, "Optical Networking and Real-Time Provisioning: An Integrated Vision for the Next-Generation Internet," *IEEE Network*, vol. 15, no. 4, pp. 36-45, July-Aug. 2001.

[3] C. Siva Ram Murthy and G. Mohan, "WDM Optical Networks: Concepts, Design, and Algorithms," *Prentice HALL PTR*, NJ, USA, November 2001.

[4] N. Ghani, " Lambda-labelling: A Framework for IP-over-WDM using MOLS," *Optical Networks Magazine*, vol. 1, pp. 45-58, Apr. 2000.

[5] R.Callon, " A Framework for Multi-Protocol Label Switching," *work in progress*, Sept., 1999.

[6] D. O. Awduche, Y. Rekhter, J. Drake, and R. Coltun, "Multi-protocol Lambda Switching: Combining MPLS Traffic Engineering Control with Optical Crossconnects," *Internet Draft draft-ieft-awduche-mpls-te-optical-00.txt*, October 1999.

[7] N. Chandhok *et. al.* , " IP-over-optical Networks: A Summary of Issues," *draft-osu-ipo-mpls-issues-02.txt*, work in progress, internet draft.

[8] B. Rajagopalan *et. al.* , " IP-over-optical Networks: A Framework," *draft-many-*

optical-framework-02.txt, work in progress, internet draft.

[9] P. Ashwood-smith *et. al.* , " Generalized MPLS, Signaling Functional Description ," *draft-ieft-mpls-generalized-signaling-01*, work in progress, internet draft.

[10] B. Rajagopalan *et. al.*, " IP-over-Optical Networks: Architecture Aspects," *IEEE Communication Magazine*, pp. 94-102, Sept. 2000.

[11] D. Awduche et al., "Requirements for Traffic Engineering over MPLS," *IETF*.

[12] A. Banerjee *et. al.*, "Generalized Multiprotocol Label Switching: An overview of Signalling Enhancement and Recovery Techniques," *IEEE Communications Magazine*, pp. 144-151, July 2001.

[13] N. Ghani, "Lambda-Labelling: A Framework for IP-over-WDM using MPLS," *Optical Networks Magazine*, pp. 45-58, April 2000.

[14] D. Awduche and Y. Rekhter, "Multiprotocol Lambda Switching: Combining MPLS Traffic Engineering Control with Optical Crossconnects," *IEEE Communications Magazine*, pp. 111-115, March 2001.

[15] *http://www.sycamorenet.com/solutions/technology/frame.dsi.html*

[16] 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, October 2000.

[17] L. Baldlne *te. al.* , "JumpStart: A Just-in-Time Signaling Architecture for WDM Burst-Switched Networks," *IEEE Communications Magazine*, pp. 82-89, February 2002.

[18] J. S. Turner, "Terabit Burst Switching," *Journal of High Speed Networks*, vol. 8,

no. 1, pp. 3-16, 1999.

[19] L. Tancevski *te. al.* , "A new scheduling algorithm for asynchronous, bariable length IP traffic incorporating void filling," In Proc. of *OFC'99*, 1999.

[20] Y. Xiong, M. Vandenhoute, and H. C. Cankaya, "Design and analysis of optical burst-switched networks," In Proc. of *SPIE'99*, vol. 3843, no. 10, pp. 112-119, Sept. 1999.

[21] M. Yoo, C. Qiao, and S. Dixit, "Optical Burst Switching for Service Differentiation in the Next-Generation Optical Internet," *IEEE Communications Magazine*, pp. 98-104, February 2001.

[22] V. M. Vokkarane, J. P. Jue, and S. Sitaraman, "Burst Segmentation: An Approach For Reducing Packet Loss In Optical Burst Switched Networks," In Proc. of *IEEE ICC 2002*, pp. 2673-2677, 2002.

[23] S. Kim, N. Kim, and M. Kang, "Contention Resolution for Optical Burst Switching Networks Using Alternative Routing," In Proc. of *IEEE ICC 2002*, pp. 2678-2681, 2002.

[24] R. Ramaswami and K. N. Sivarajan, " Dynamic Allocation in All-Optical Ring Networks," In Proc. of *IEEE ICC 97*, vol. 1, pp. 432-436, June 1997.

[25] H. Zang, J. P. Jue, L. Sahasrabuddhe, R. Ramamurthy, and B. Mukherjee, "Dynamic Lightpath Establishment in Wavelength-routed WDM Networks," *IEEE Communications Magazine*, pp. 100-108, November 2001.

[26] I. Chlamtac, A. Farago, and T. Zhang, "Lightpath (Wavelength) Routing in Large WDM Networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 909-913, June 1996.

[27] L. Li and A. K. Somani, "Dynamic Wavelength Routing Using Congestion and Neighborhood Information," *IEEE/ACM Transactions on Networking*, vol. 7, no. 5, pp. 779-786, Oct. 1999.

[28] Y. Ye, C. Assi, S. Dixit, and M. A. Ali, "A Simple Dynamic Integrated Provisioning/Protection Scheme in IP over WDM Networks," *IEEE Communications Magazine*, pp. 174-182, November 2001.

[29] A. Jukan and G. Franzl, "Constraint-based Path Selection Methods for On-demand provisioning in WDM Networks," In Proc. of *IEEE INFOCOM 2002*, 2002.

[30] M. Kodialam and T. V. Lakshman, "Integrated Dynamic IP and Wavelength Routing in IP over WDM networks," In Proc. of *IEEE INFOCOM 2001*, pp. 358-366, 2001.

[31] M. Kodialam and T. V. Lakshman, "Minimum Interference Routing with Application to MPLS Traffic Engineering," In Proc. of *IEEE INFOCOM 2000*, pp. 884-893, 2000.

[32] R. A. Guerin, A. Orda, and D. Williams, "QoS Routing Mechanisms and OSPF Extensions," In Proc. of *IEEE GLOBECOM 1997*, vol. 3, pp. 1903-1908, 1997.

[33] S. Plotkin, " Competitive Routing of Virtue Circuits in ATM Network," *IEEE J. Selected Areas in Communications*, pp. 1128-1136, June 1997.

[34] A. V. Goldberg and R. E. Tarjan, "Solving Mimimum Cost Flow Problem by Successive Approximation," In Proc. of *the 19th ACM Symposium on the Theory of Computing*, pp. 7-18, 1987.

[35] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, "Network Flows: Theory, Algo-

rithms, and Applications," *Prentice Hall*, 1993.

[36] D. H. Lorenz and A. Orda, "QoS Routing in Networks with Uncertain Parameters," *IEEE/ACM Transactions on Networking*, vol. 6, no. 6, pp. 3-10, December 1998.

[37] R. A. Guerin and A. Orda, "QoS-based routing in Networks with Inaccurate Information: Theory and Algorithms," In Proc. of *IEEE INFOCOM 1997*, pp. 75-83, 1997.

[38] A. Shaikh, J. Rexford, and K. G. Shin, "Evaluating the Impact of Stale Link State on Quality-of-Service Routing," *IEEE/ACM TRANSACTIONS ON NETWORKING*, vol. 9, pp. 162-175, April 2001.

[39] S. Sengupta, D. Saha, and S. Chaudhuri, "Analysis of Enhanced OSPF for Routing Lightpaths in Optical Mesh Networks," In Proc. of *IEEE ICC 2002*, vol. 5, pp. 2865-2869, 2002.

[40] A. Elwalid, C. Jin, S. Low and I. Widjaja, "MATE: MPLS Adaptive Traffic Engineering," *In Proc. of IEEE INFOCOM 2001*, pp. 1300-1309, 2001.

[41] Y. Chen, M. Hamdi, and D. H. K. Tsang, "Proportional QoS over OBS Networks," In Proc. of *IEEE ICC 2001*, pp. 1510-1514, 2001.

[42] K. Kompella *et. al.*, "OSPF Extensions in Support of Generalized MPLS," *Internet Draft draft-ietf-ccamp-ospf-gmpls-extensions-07.txt*, May 2002.

[43] K. Kompella *et. al.*, "LSP Hierarchy with Generalized MPLS TE," *Internet Draft draft-ietf-mpls-lsp-hierarchy-05.txt*, April 2002.

[44] T. Tripathi and K. N. Sivarajan, "Computing Approximate Blocking Probabilities in Wavelength Routed ALL-Optical Networks with Limited-Range Wave-

length Conversion," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, October 2000.

[45] S. Subramaniam, M. Azizoglu, and A. K. Somani, "All-Optical Networks with Sparse Wavelength Conversion," *IEEE/ACM Transactions on Networking*, vol. 4, no. 4, pp. 544-557, August 1996.

[46] Y. Zhu, G. N. Rouskas and H. G. Perros, "A Path Decomposition Approach for Computing Blocking Probabilities in Wavelength-Routing Networks," *IEEE/ACM Transactions on Networking*, vol. 8, no. 6, pp. 747-762, Dec. 2000.

[47] C. Qiao, "Labeled Optical Burst Switching for IP-over-WDM Integration," *IEEE Communications Magazine*, pp. 104-114, September, 2000.

[48] K. Thompson, G. J. Miller, and R. Wilder, "Wide-Area Internet Traffic Paterns and Characteristics," *IEEE Networks*, vol.6, no. 6, Dec. 1997.

[49] V. Paxson and S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling," In Proc. of *ACM SIGCOMM 1994*, pp. 257-268, Aug. 1994.

[50] A. Shaikh, J. Rexford, and K. Shin, "Load-Sensitive Routing of Long-Lived IP Flows," In Proc. of *ACM SIGCOMM 1999*, 1999.

[51] S. Jamin, S. J. Shenker, and P. B. Danzig, " Comparison of Measurement-based Admission Control Algorithms for Controlled-Load Service," In Proc. of *IEEE INFOCOM*, 1997.