A Technique to Minimize Contention in Optical Burst Switching Networks

Mrinal Nandi

Department of Computer Science and Engineering
National Institute of Technology, Rourkela
Rourkela-769 008, Orissa, India

A Technique to Minimize Contention in Optical Burst Switching Networks

Thesis submitted in partial fulfillment

of the requirements for the degree of

Master of Technology

(Research)

in

Computer Science and Engineering

by

Mrinal Nandi

(Roll: 60606005)

Department of Computer Science and Engineering

National Institute of Technology, Rourkela

Rourkela-769 008, Orissa, India

Department of Computer Science and Engineering
National Institute of Technology, Rourkela
Rourkela-769 008, Orissa, India.

Certificate

This is to certify that the work in the thesis entitled **A Technique to Minimize Contention in Optical Burst Switching Networks** submitted by **Mrinal Nandi** is a record of an original research work carried out by him under our supervision and guidance in partial fulfillment of the requirements for the award of the degree of *Master of Technology (Research)* in *Computer Science and Engineering* during the session 2006 – 2008 in the department of *Computer Science and Engineering, National Institute of Technology, Rourkela.* Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Bibhudatta Sahoo
Senior Lecturers
Department of CSE
National Institute of Technology
Rourkela-769008

Ashok Kumar Turuk
Assistant Professor
Department of CSE
National Institute of Technology
Rourkela-769008

Place: NIT, Rourkela
Date:

Acknowledgment

There are many people to thank for making this thesis possible. First and foremost, I would like to thank my supervisor, Prof. Ashok Kumar Turuk for giving me the guidance, encouragement, counsel throughout my research and painstakingly reading my reports. His undivided faith in this topic and ability to bring out the best of analytical and practical skills in people has been invaluable in tough periods. Without his invaluable advice and assistance it would not have been possible for me to complete this thesis.

Secondly, I would like to thank my co-supervisor Prof. B. D. Sahoo for his invaluable suggestions, and encouragements during this research period. I also thank Prof. Banshidhar Majhi, Head CSE, Prof. Durga P. Mahapatra, Prof S. Ghosh for serving on my Masters Scrutiny Committee. A special thanks goes out to Prof. Pankaj Kumar Sa for his time to time suggestions from the first day of my joining in NIT, Rourkela.

I wish to thank all faculty members and secretarial staff of the CSE Department for their sympathetic cooperation.

Further, I would like to thank my seniors M.Tech students, especially R. K. Bind, Aser A. Ekka for sharing their knowledge at the initial stage of my research work. Thanks to my all fellow researchers, especially Dillip K. Puttal, Subrajeet Mohapatra and Kumar Dhiraj for sharing their knowledge and being good friends.

Finally, I would like to thank my family for academic guidance, love and support, and for implicitly and explicitly showing their belief on me.

Mrinal Nandi

Abstract

Optical burst switching (OBS) is the new switching technique for next generation optical networks. However, there are certain issues such as burst aggregation, scheduling, contention resolution and QoS that needs to be addressed in OBS. This thesis is an attempt to address the burst scheduling and burst contention in OBS networks.

Several scheduling algorithms have been proposed in the literature, which can be categorized into *Horizon* and *Void filling* scheduling algorithm. Void filling algorithms perform better because they exploits void within a channel for scheduling. Reported void filling algorithms - Latest available unscheduled channel with void filling (LAUC-VF) and Minimum end void (Min-EV) - do not consider the void duration in scheduling. In this thesis we propose a new scheduling algorithm called Best Fit Void Filling (BFVF), which consider both void duration and incoming burst length to find an optimal void channel. We simulate our proposed scheme using obs-ns simulator and compared with LAUC-VF and Min-EV algorithm. Result shows that the burst loss ratio is lower and channel utilization is higher in our proposed scheme.

Burst loss due to contention is another important issue in optical burst switching networks. A number of techniques have been proposed in the literature to resolve contention. However, none of these techniques tries to minimize the occurrences of contention. In this thesis, we proposed a cluster based scheme, which reduces the burst loss by minimizing the occurrences of contention. In our proposed scheme, a given network is logically divided into number of clusters. One of the node within each cluster is selected as a cluster head, which maintains the status of resources in the network. Cluster heads exchange the status of network resources among themselves to maintain an up-to-date information about the network resources. Prior transmission of an OBS control packet, a node request its cluster head for an available wavelength channel. Cluster head sends a positive reply with the identity of wavelength channel or negative reply depending

upon whether a wavelength channel is available or not. A node on receiving positive reply sends an OBS control packet to reserve the wavelength channel whose identity it has received from the cluster head. For a negative reply the burst is dropped. Our proposed scheme is compared with no-deflection and deflection routing scheme. We found that burst loss ratio is higher in no-deflection routing and lower in our proposed scheme.

List of Acronyms

Acronym	Description
ATM	Asynchronus Transfer Mode
BFVF	Best Fit Void Filling
CWDM	Coarse Wavelength Division Multiplexing
CRP	Channel Request Packet
CBRT	Control Burst Router
DWDM	Dense Wavelength Division Multiplexing
EDFA	Eerbium-doped Fiber Amplifier
FDM	Frequency Division Multiplexing
FDL	Fiber Delay Line
FFUC	First Fit Unscheduled Channel
FFUC-VF	First Fit Unscheduled Channel With Void Filling
IM	Input Module
IP	Internet Protocol
JIT	Just-In-Time
JET	Just-Enough-Time
LAUC	Latest Available Unscheduled Channel
LAUC-VF	Latest Available Unscheduled Channel With Void Filling
Min-EV	Minimum End Void
NAK	Negative Acknowledgment
NCR	Negative Channel Reply
OBS	Optical Burst Switching
OXC	Optical Cross Connect
O-E-O	Optical to Electronic to Optical Conversion
OSN	Optical Switching Network
OM	Output Module
PCR	Positive Channel Reply
RWA	Routing and Wavelength Assignment
RUP	Resource Update Packet
TAG	Tell-And-Go
WDM	Wavelength Division Multiplexing

List of Symbols

Symbol	Description
T_b	Burst assembly period in timer-based schemey
L_b	Burst length in threshold-based scheme
t_{CB}	Arrival time of a control packet in an optical burst switched node
t_{setup}	Time taken by an OBS node to process the control packet
t_{offset}	Offset value
t_{oxc}	Time taken by an OXC to configure its switch fabric to setup a connection
t_{burst}	Is the length of the payload
H	Is number of hops from source to destination

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, the demand for network bandwidth is growing due to increase in global popularity of Internet and variety of applications. Optical data communication has been acknowledged as the best solution to meet the present bandwidth requirement of users and supporting future network services. This is because theoretically optical fiber has the ability to support bandwidth demand up to 50 THz. Light wave has higher frequency and hence shorter wavelength, therefore more bits of information can be contained in a length of fiber versus the same length of copper. Apart from this, optical fiber provides extremely low bit-error rate of the order of 10^{-12}. Optical signals are immune to electrical interferences. Fiber cables are much more difficult to tap than copper wires, so there is a security advantage in optical communication. All these factors make optical networks as the future networks.

In first generation optical networks, fibers were used as point-to-point connections. The entire bandwidth available for transmission was not fully exploited. This is because electronic equipments operate at an order of gigabits per second, whereas the fiber has a bandwidth of terabits per second. This mismatch between electronic speed and the optical bandwidth is called electronic bottleneck. Representative of first generation optical networks are SONET/SDH.

In second generation *Wavelength Division Multiplexing (WDM)* technology were deployed to overcome the problem of electronic bottleneck. WDM is the optical version of frequency division multiplexing (FDM). WDM divides the available bandwidth of a single fiber into a number of non-overlapping wavelength channels. Each of the wavelength channels operate at the electronic speed. Several signals are transmitted at different wavelengths in a single fiber at the same time. Thus, WDM encapsulates many virtual fibers in a single fiber. The main advantages of WDM technology are transparency, scalability and flexibility. Transparency refers to the fact that the wavelengths can carry data at a range of bit rates through a variety of protocols. Some wavelengths could carry SONET data, whereas others carry ATM cells, and all operating at different bit rates. In a WDM network, though the number of wavelengths available in the network is limited, network offers an enormous capacity by spatial reuse of wavelengths in the network. This spatial reuse of wavelengths in WDM networks makes the network scalable. WDM is flexible too; an existing optical network can be upgraded to a WDM based optical network.

WDM systems can be classified as dense wavelength division multiplexing (DWDM) systems and coarse wavelength division multiplexing (CWDM) systems. In DWDM, the bandwidth of the fiber is divided into more than *eight* wavelengths. CWDM refers to the systems where the fiber bandwidth is divided into less than *eight* wavelengths.

Another remarkable technological development, which makes optical network a reality, is *erbium-doped fiber amplifier (EDFA)*, which amplifies signals at many different wavelengths simultaneously, regardless of their individual bit rates, modulation scheme or power levels. Before invention of EDFAs, the effect of optical loss were compensated for every few tens of kilometers by an electronic regenerator, which required the optical signals be converted to an electrical signals and then back to optical.

As the IP aware traffic is increasing at a faster rate, it is envisaged that the future networks will predominantly carry traffic of IP based applications. The commonly used IP-backbone architecture of today is built on a network protocol stack, which uses ATM, SDH, and WDM, and referred to IP/ATM/SDH/WDM protocol stack. This architecture has traditionally been used to provide assured levels of performance and reliability for the predominant voice and leased-line services. However, this multilayer protocol stack has a number of redundant functionalities and is associated with high capital and running costs. Thus, it is not suitable to provide *data*-optimized *packet-switched* services for the transport of rapidly growing IP based traffic. Therefore, to carry the IP based data traffic on WDM networks, a two-layer architecture – IP/WDM – is emerging as the de-facto standard, where the WDM layer is used for bandwidth provisioning. This two-layered architecture is destined to eliminate redundant functionalities, reduce the protocol overheads, simplify the network management, and transport IP traffic as efficiently as possible over WDM based optical networks. All-optical WDM layers will ideally enable a huge amount of traffic to be switched in the optical domain overcoming the potential bottleneck in the electronic router. Thus, it will provide direct high speed/high bandwidth communication pipes as well as transparency to bit rate and coding formats.

Besides multiplexing, the network needs switching techniques to carry traffic from source to destination. There exists many switching techniques. In the following section we briefly discuss switching techniques commonly used in optical networks.

1.1 Switching Techniques for Optical Networks

Three switching techniques that are well studied to carry IP traffic over WDM networks are – optical circuit switching, packet switching and burst switching. Accordingly, WDM networks can be classified as wavelength routed networks, optical packet switched networks and optical burst switched networks, respectively. In all such networks, routing information is provided by the network. Besides these networks, there is another category of networks called broadcast-and-select network, where no routing information is provided by the network.

1.1.1 Broadcast-and-Select Networks

In a broadcast-and-select network, all input signals are combined at a passive star coupler and is broad cast to all nodes. The intended destination has to select the corresponding wavelength by tuning its receiver accordingly. Most local area networks (LANs) of today, for example Ethernet, token ring and FDDI networks belong to broadcast-and-select type of networks. Such networks need a MAC protocol to resolve contentions and to avoid/minimize collision in the network while sharing the media. The design of a MAC protocol is guided by the node architecture. Depending upon the tunability of the transceiver used, four types of node architectures are possible. The possible combinations are: (i) fixed transmitter(s) and fixed receiver(s) (FT-FR), (ii) fixed transmitter(s) and tunable receiver(s) (FT-TR), (iii) tunable transmitter(s) and fixed receiver(s) (TR-FR) and (iv) tunable transmitter(s) and tunable receiver(s) (TT-TR). A node is equipped with any of the above four combinations.

A node architecture configured with fixed components cannot adapt to the variations in number of nodes in the network. Thus, in a broadcast-and-select network, where nodes are equipped with fixed transmitters and/or receivers, the network is not scalable. We define network scalability in two ways. We call a network scalable, if its performance does not degrade *drastically* with increase in number of nodes. However, the performance does degrade with increase in load and nodes, a desirable feature of the network is that such degradation should be *graceful*. Additionally, we call a network scalable if a change in the number of nodes does not necessitate for additional resource requirements at each other node. A network, where nodes are equipped with tunable transceivers, is scalable because it does not necessitate additional resources for a change in the number of nodes in the network. Moreover, for nodes with tunable transceivers, it takes a single hop for traffic to flow between a source-destination pair.

Besides scalability of the node architecture, the MAC protocol for network should be collision free and satisfy QoS demands from different applications.

1.1.2 Wavelength Routed Networks

Wavelength routed networks consist of optical-corssconnects interconnected by point-to-point fiber links in an arbitrary mesh topology. Connection between any two nodes in the network is established by setting up a lightpath. A lightpath is a circuit established between any two nodes in the network and is uniquely identified by a route and a wavelength associated with it. The algorithms used for selecting the route and wavelength to establish lightpath are known as routing and wavelength assignment (RWA) algorithms. Once lightpath is established between source-destination pair, data is transmitted between the end points of the lightpath without processing, buffering or optical-electronic-optical (O-E-O) conversion at intermediate nodes. There can be single hop or multihop lightpaths between a source-destination pair. In a single hop lightpath, the wavelength on all links between source-destination pair remains the same. This constraint, that, the wavelengths on all links in a lightpath remain the same is known as wavelength continuity constraint. In a single hop lightpath, the traffic remains in optical domain between any source-destination pair. However, it is not possible between every source-destination pair to setup a single hop lightpath because of the scarcity of the available wavelengths. If it is not possible to setup a single hop lightpath, there can be a multi-hop lightpath between the source-destination pair. In such a multihop lightpath, traffic undergoes O-E-O conversion at intermediate nodes.

Wavelength routed networks do not use statistical sharing of resources, and therefore provides lower bandwidth utilization.

1.1.3 Optical Packet Switched Networks

In packet switched networks, IP traffic is processed at every router on a packet-by-packet basis. An IP packet contains payload and a header. Header contains the routing information and the payload, the actual data. In optical packet switching, a packet is sent along with its header. While the header is being processed at an intermediate node, either all-optically or electronically (after an O/E conversion), the packet is buffered at the node in optical domain. The availability of optical buffers is a major constraint in this approach. Fiber delay lines have been proposed as an alternative to electronic buffers. However, the buffers provided by fiber delay lines are of limited capacity and cater to delays of fixed duration only. For synchronization, the packet size is kept fixed.

The packet switched network uses statistically sharing/multiplexing of resources, and therefore, increases the bandwidth utilization.

1.1.4 Optical Burst Switched Networks

Recent studies have established that the Internet traffic as well as the LAN traffic is bursty in nature. In optical circuit switching (also know as wavelength routing), a lightpath needs to be established between the source-destination pair using a *dedicated* wavelength on each link along a physical path. Bandwidth would be efficiently utilized if the duration of transmission is relatively longer then the lightpath setup time. Moreover, as the number of wavelengths is limited, every source-destination pair cannot have a dedicated lightpath in the network. As a result, some traffic have to go through O-E-O conversion. On the other hand, optical packet switching has its own limitations due to the non-availability of optical buffers and processing in the optical domain.

In such a technological scenario, optical burst switching (OBS) is emerging as the preferred switching paradigm, which is expected to provide high-bandwidth transport services at optical layer for bursty traffic in a flexible, efficient and feasible way. OBS combines the advantages of both circuit and packet switching while overcoming their limitations. It is envisaged that OBS *may* provide an efficient integration for IP-over-WDM framework in comparison to today's multiprotocol stack. However, optical burst switched networks are inherently buffer-less, they need some mechanism for contention resolution at the network core. With increasing demand for QoS from different applications, OBS networks should also support QoS.

1.2 Motivation

It is envisaged that, traffic at the backbone of next-generation optical network will remain in optical domain. In such all-optical network, buffering, switching and routing within the network nodes will be performed optically. Network elements such as optical cross-connects and optical add/drop multipliers will have full control of all wavelengths. Additionally, they are expected to have full knowledge of the traffic carrying capacity and the status of each wavelength. With such intelligence, these networks are envisioned as being self-connecting and self-regulating.

The main problem in such all-optical network is the unavailability of optical RAM and technology is not matured for optical processing. As a result of which packet switching is not an appropriate switching technique in all-optical network. Circuit switching is not appropriate because of inefficient bandwidth utilization and is costly in maintaining a circuit. In between the extremes of circuit and packet switching, optical burst switching (OBS) is emerging as the new switching paradigm for next generation optical networks. In OBS, control part is separated from data part. Control part is called control packet and data part is called data burst. A control packet is sent in advance to configure intermediate switches. Data burst follows the control packet after a pre-determined amount of time called the offset time. At intermediate nodes, control packet configure switches using a wavelength scheduling algorithm. Different wavelength scheduling algorithms such as *Horizon* and *Void Filling* algorithm have been proposed to schedule an incoming data burst to an outgoing wavelength channel. First fit unscheduled channel (FFUC) and latest available unscheduled channel (LAUC) are the candidate of horizon algorithms. First fit unschedule channel with void filling (FFUC-VF), latest available unschedule channel with void filling (LAUC-VF) and minimum end void (Min-EV) are the candidate of void filling algorithms. Horizon algorithms are easy to implement but have higher burst loss. Void filling algorithms are more complex and gives better performance in terms of burst loss.

Scheduling of multiple data burst at same outgoing link results in contention. Different contention resolution techniques such as deflection routing, wavelength conversion, buffering and burst segmentation are proposed in the literature. These techniques do not try to minimize the occurrences of contention. They try to resolve contention.

In this thesis we proposed a channel scheduling algorithm and a technique to minimize contention in OBS network.

1.3 Objective

OBS is a one-way reservation scheme. Source has no way of knowing whether resource reservation is successful for the data burst it sent. Scheduling of more than one data burst on the same outgoing link results in contention. Scheduling algorithm should be able to minimize contention at the outgoing links. In this thesis we propose a new channel scheduling algorithm and a technique to reduce contention in OBS networks. Objectives of the thesis are enumerated as below:

- to study the existing channel scheduling algorithms,
- to propose a new channel scheduling algorithm,
- to propose a technique to minimize contention in OBS networks,
- to study through simulation the performance of the above propose scheme.

We have used obs-ns simulator (which runs on the top of ns2 simulator) and self-similar traffic in our simulation.

1.4 Organization of the Thesis

Rest of the thesis is organized into the following chapters :

A brief discussion on the scheduling algorithms and contention resolution schemes are given in **Chapter 2**.

A scheduling algorithm call best fit void filling (BFVF) is proposed in **Chapter 3**. Simulation result shows that burst loss ratio is lower in proposed scheme.

A scheme for contention minimization is proposed in chapter **Chapter 4**. The proposed scheme divides a given network into number of clusters. A node within each cluster is selected as cluster head, which keeps track of the status of resources available in the network. A node before sending a data burst, requests its cluster head for an available wavelength channel and schedule the data burst on the channel. Simulation result shows burst loss ratio is lower in proposed scheme.

In **Chapter 5** we summarized the work done and suggest directions for possible future work.

Chapter 2

Optical Burst Switching

There has been a phenomenal increase in the demand of bandwidth over the years due to rapid growth in the number of Internet users and increase in bandwidth intensive applications such as voice-over-IP, video conferencing, interactive video-on-demand, and many other multimedia applications [1]. To meet the ever growing demand of bandwidth, copper cables were replaced by optical fibers in both the access networks as well as in the backbone networks [2]. Optical fiber not only supports huge bandwidth but also have other advantages too such as low bit-error rate, no interference problem and security advantage [3].

In first generation optical network, optical fibers provide only point-to-point connections. Entire potential of the fiber could not be utilized, because the electronic routers operate at a much lower speed than the fiber capacity. Wavelength division multiplexing (WDM) technology, were deployed in the second generation optical networks. WDM divides the available bandwidth of the fiber into number of non-overlapping wavelength channels each operating at electronic speed. To carry IP traffic over WDM networks three switching technologies have been studied: optical circuit, packet switching and burst switching. Optical circuit switching and packet switching have their own limitations when applied to WDM networks [4]. Circuit switching is not bandwidth efficient unless the duration of transmission is greater than the circuit establishment period. It is shown that establishment of circuits (lightpaths) in optical networks is an NP-hard problem [5–7]. On the other hand packet switching is hop-by-hop store and forward scheme and needs buffering and processing at each intermediate node [8]. It is flexible and bandwidth efficient. However, technology for buffering and processing in optical domain is yet to mature for this scheme to be commercialize [9, 10]. Fiber delay lines (FDL) have been proposed in literature to provide buffering. However, FDL have limited buffering capability and support only for a fixed duration [11].

In this context optical burst switching (OBS) [12–17] is emerging as the alternative switching techniques, which combines the advantages of both circuit switching and packet switching. OBS needs no buffering and ensures efficient bandwidth utilization on a fiber link by reserving bandwidth only when data is actually required to be transferred through the link.

In OBS, a burst is the basic switching entity. Burst is a variable length data packet, assembled at an ingress router by aggregating a number of IP packets, which may be received from a single host or from multiple hosts belonging to the same or different access networks. A burst has two components: control and payload [18, 19]. The control packet carries the header information. Thus, the control component incurs an overhead, referred to as control overhead. Payload is the actual data transmitted.

In OBS control and payload is decoupled. Control is sent on a control channel and payload/ data on data channels. Control packet is sent first followed by the payload on a separate wavelength channel after an offset time equal to the processing time of control packet at intermediate node. Control packet is processed electronically at each intermediate node and reserves resources for a period starting from the time the payload/ data burst is expected to arrive at the node until the transmission is completed. If reservation is successful the control packet is transmitted to the next node on the path, else it is dropped at the node. For a successful reservation, switches are configured by the time payload/ data burst arrive at the node. Hence the data burst remains in optical domain from source to destination. OBS uses one-way reservation schemes. We summarized below, the important properties of OBS.

- Payload (data burst) and header (control packet) are transmitted on different channels.

- Data bursts are of variable length.

- Payload follows the header after an offset time equal to the sum of processing delay of control packet at each intermediate nodes.
- Header undergoes optical-electronic-optical (O-E-O) conversion at each intermediate nodes.
- Payload remains in optical domain from source to destination.
- Uses one-way reservation scheme.
- Resources are reserved for a fixed duration and are release implicitly.
- No buffering of payload at intermediate nodes.

Comparison of three switching technology is given in Table 2.1 [20].

Table 2.1: Comparison of Switching Technologies

Switching	Bandwidth Utilization	Latency	Optical Buffering	Overhead	Adaptively
Circuit	Low	High	Not required	Low	Low
Packet	High	Low	Required	High	High
OBS	High	Low	Not required	Low	High

2.1 Architecture of OBS

An architecture of OBS network is shown in Figure 2.1. OBS network consists of two types of nodes: edge node and core node [2, 21, 22]. Edge nodes are at the interface between electronic and optical domain. Edge nodes can be an ingress or egress node. Packets are assembled into bursts at ingress edge node, which are then routed through the OBS network and disassembled back into packets at egress edge node. A core node is mainly composed of an optical switching matrix and a switch control unit which are responsible to forward payload/ data burst.

A node in OBS network consists of both optical and electronic components. The optical components are multiplexers (Mux), demultiplexers (Demux) and an

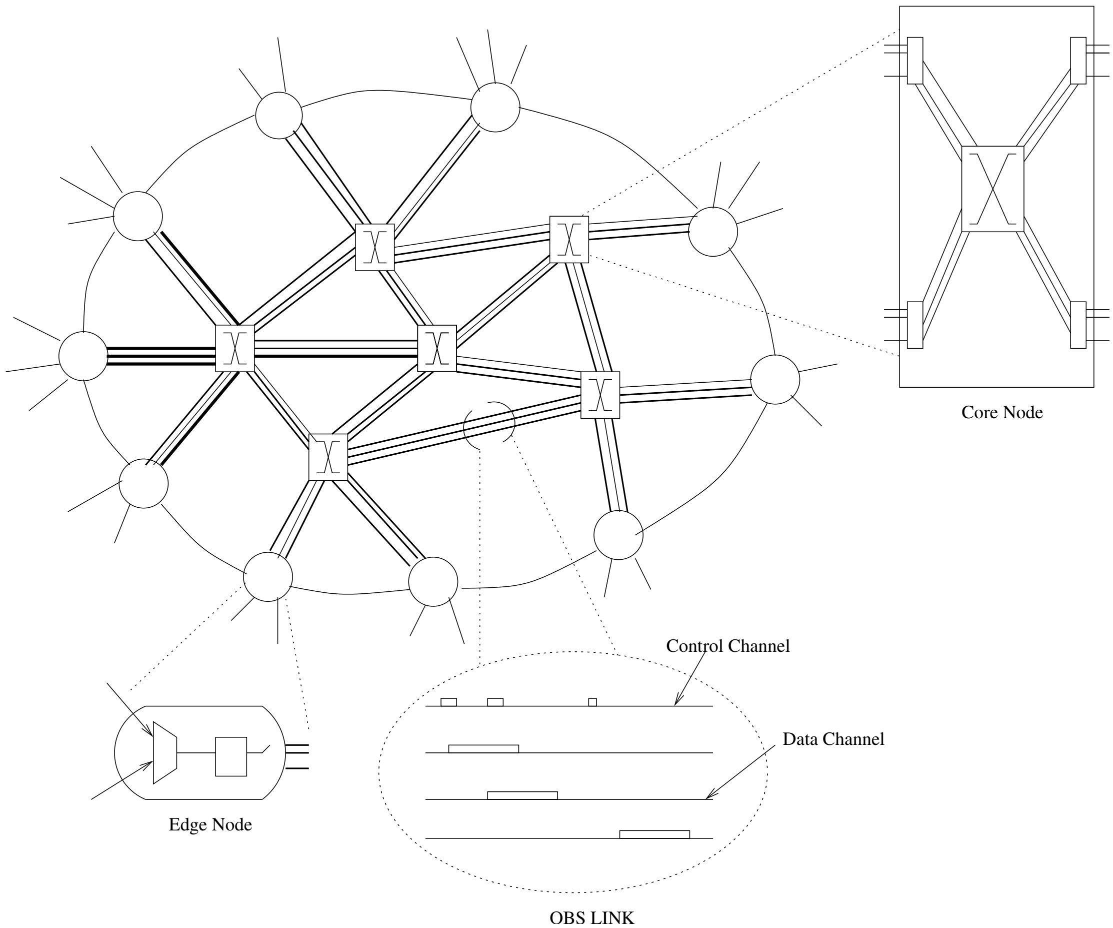

Figure 2.1: Architecture of OBS network

optical switching network (OSN). The electronic components are input modules (IM), output module (OM), a control burst router (CBRT), and a scheduler [21]. An optical burst switch control unit transfers an incoming data burst from an input port to its destination output port.

When an edge node intends to transmit a data burst, it sends a control packet on the control wavelength to a core node. At core node, the control packet on the control channel is input to the corresponding IM, which converts the control packet into electronic form. The control fields are extracted from the control packet. The CBRT uses these control fields to determine the next outgoing fiber for the corresponding payload by consulting a routing table maintained locally. The control packet is scheduled for transmission onto the selected outgoing link by the scheduler and the control packet is buffered until the scheduled time. The scheduler maintains a control packet queue. The scheduler also reserves wavelength on the determined links for the upcoming payload. The control packet is then forwarded on the OM, which updates its control fields and transmits it to the selected outgoing fiber using the optical transmitter. Just before the payload arrives, the switching element in the node is configured to connect the input port to the corresponding output port for the entire duration of the burst transmission. If the control packet is unable to reserve the wavelength then the control packet as well as payload is dropped.

2.2 Burst Assembly Schemes

In OBS network, packets are assembled into burst at edge node. There exists two burst assembly schemes: threshold-based and timer-based [23, 24]. In a timer-based scheme, a timer is started at the initialization of burst assembly. A data burst containing all packets in the buffer is generated when the timer exceeds the burst assembly period T_b. A large time-out value T_b results in a large packet and higher buffering delay at the edge node. On the other hand, a too small T_b results in too many small bursts and a high electronic processing load.

In a threshold-based scheme, a burst is created and sent into the OBS network when the total size of the packets in the queue reaches a threshold value L_b. The shortcoming of the threshold-based scheme is that it does not provide any guarantee on the assembly delay that packets will experience.

The choice of burst assembly algorithms depends on the type of traffic being transmitted. Timer-based algorithms are suitable for time-constrained traffic such as real-time applications because the upper bound of the burst assembly delay is limited. For a time-insensitive application such as file transmission, to reduce the overhead of control packets and increase OBS transmission efficiency, a threshold based scheme may be more appropriate.

2.3 Wavelength Reservation Schemes

Wavelength reservation refers to when and how the bandwidth is reserved and release. The reservation schemes in OBS network is adopted from ATM block transfer (ABT) [25]. There are two versions of ABT: ABT with delayed transmission and ABT with immediate transmission.

In an immediate transmission reservation scheme, an output wavelength is reserved for a payload immediately after the arrival of the corresponding control packet; if a wavelength cannot be reserved at that time, then the setup message is rejected and the corresponding data burst is dropped [16]. In a delayed reservation scheme, the control packet and the payload are separated in time by an offset value in order to accommodate the processing of the control packet. An output wavelength is reserved for a data burst just before the arrival of the first bit of the data burst. If, upon arrival of the setup message, it is determined that no wavelength can be reserved at the appropriate time, then the setup message is rejected and the corresponding data burst is dropped [16].

These two techniques have been adopted in OBS. Depending on bandwidth reservation, offset time and control management, three schemes for OBS implementation have been proposed: Tell-and-go (TAG) [16], Just-in-time (JIT) [15,26] and Just-enough-time (JET) [27].

2.3.1 Tell-And-Go (TAG)

This is an immediate reservation scheme. In TAG, the control packet is transmitted on a control channel followed by a payload, on a data channel with zero or negligible offset. The payload is buffered using fiber delay line (FDL) while the control packet is processing at each intermediate node. If wavelength reservation is successful then the payload is transmitted along the reserved channel else the data burst is dropped and a negative acknowledgment (NAK) is sent to the source. The source node sends a control packet after transmitting the payload to release the reserved resources along the path.

The drawback of this scheme is availability of optical buffer. FDL can hold

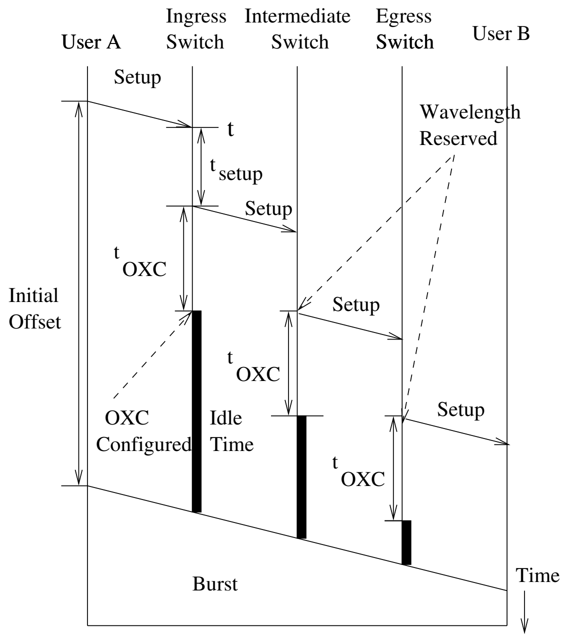

Figure 2.2: Just-In-Time Reservation Scheme

data only for a fixed duration and can not accommodate data burst of variable size. Furthermore, loss of control packet to release reserved resources result in wastage of bandwidth [21, 26].

2.3.2 Just-In-Time (JIT)

This is also an immediate reservation scheme. Here, nodes reserve the resources as soon as the control packet is processed. Source transmits the payload after an offset time which is greater than the total processing time of control packet at intermediate nodes. If the resource is not available, the data burst is dropped. The difference between JIT and TAG is that in JIT the buffering of the payload at each node is eliminated by inserting a time slot between the control packet and the payload. The time slot is equal to the offset time. Since the bandwidth is reserved immediately after processing the control packet, the wavelength will be idle from the time the reservation is made till the first bit of the payload arrives at the node. This is because of the offset between the control packet and the payload. An in-band-terminator is placed at the end of each data burst, which is used by each node to release the reserved wavelength after transmitting the payload [6,26].

Working of JIT is shown in Figure 2.2. In this figure user **A** send a data burst to user **B**. Let t be the time a control packet arrives at some OBS node along the path to the destination. Let t_{setup} be the amount of time it takes an OBS node to process the control packet and t_{offset} be the offset value. The offset value depends on (i) the wavelength reservation scheme, (ii) number of nodes the control packet has already traversed, and (iii) other factors, such as whether the offset is used for service differentiation [16]. t_{oxc} is the amount of time it takes the OXC to configure its switch fabric to set up a connection from an input port to an output port. Once, the processing of the control packet is complete at time $t + t_{setup}$, a wavelength is immediately reserved for the upcoming data burst, and the operation to configure the OXC fabric to switch the data burst is initiated. When this operation completes at time $t + t_{setup} + t_{oxc}$, the OXC is ready to carry the data burst.

Note that the data burst will not arrive at the OBS node until the time is $(t + t_{offset})$. As a result, the wavelength remains idle for a period of time equal to $(t_{offset} - t_{setup} - t_{oxc})$. Since the offset value decreases along the path to the destination, deep inside the network for an OBS node, will have shorter idle time between the instant OXC is configured and the arrival of first bit of payload [3, 16, 26].

2.3.3 Just-Enough-Time (JET)

JET is a delayed reservation scheme. Here, the size of the data burst is decided before the control packet is transmitted by the source. The offset between control packet and payload is also calculated based on the hop count between the source and destination. At each node, if bandwidth is available, the control packet reserves wavelength for the upcoming data burst for a fixed duration of time. The reservation is made from the time when the first bit of payload reaches the node till the last bit of payload is transmitted to the output port. This eliminates the wavelength idle time. This is the basic difference between JET and JIT. Since the wavelength is reserved for a fixed duration, there is no need for explicit release of reserved resources along the path. Since there is no wastage of bandwidth in this

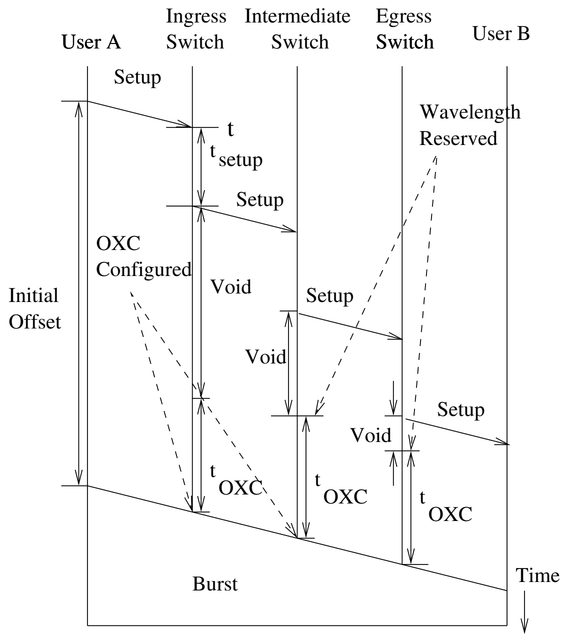

Figure 2.3: Just-Enough-Time Reservation Scheme

scheme, channel utilization is higher than other schemes. However, scheduling process is complex compared to other schemes.

The operation of delayed reservation in JET is shown in Figure 2.3. Let assume that a control packet has arrived at an OBS node at time t. Let the offset time is t_{offset} and the length of the payload is t_{burst}. The first bit of the corresponding data burst is expected to arrive at time $t + t_{offset}$. After processing the control packet, the node reserves a wavelength for the payload starting at time $t+t_{offset} - t_{OXC}$ and ending at time $t + t_{offset} - t_{OXC} + t_{burst}$. At time $t + t_{offset} - t_{OXC}$, the OBS node instructs its OXC fabric to configure its switching elements to carry the payload, and this operation completes just before the arrival of the first bit of the data burst. Immediate reservation protocols only permit a single outstanding reservation for each output wavelength, whereas delayed reservation schemes allow multiple setup messages to make future reservations on a given wavelength (provided these reservations, do not overlap in time). A void is created on the output wavelength between the time slot $t + t_{setup}$ to $t + t_{offset} - t_{OXC}$. In an attempt to use the voids created by the earlier setup messages, void filling algorithms are employed in JET [16].

TAG and JIT schemes are significantly simpler than JET since they do not involve complex scheduling or void-filling algorithms. Previous studies have shown that JET performs better than either JIT or TAG in terms of burst loss probability [3, 16, 27–29].

2.4 Burst Scheduling Algorithms

When a control packet arrives at a core node, a wavelength channel scheduling algorithm is used to determine a wavelength channel on an outgoing link for the corresponding data burst. The information required by the scheduler such as the expected arrival time of the data burst and its duration are obtained from the control packet. The scheduler keeps track of the availability of time slots on every wavelength channel. It selects one among several idle channels. The selection of wavelength channel needs to be done in an efficient way so as to reduce the burst loss. At the same time, the scheduler must be simple and should not use any complex algorithm, because the routing nodes operate in a very high-speed environment handling a large amount of burst traffic. A complex scheduling algorithm may lead to the early data burst arrival situation wherein the data burst arrives before its control packet is processed and eventually the data burst is dropped [3].

In this section we discuss various scheduling algorithms proposed in literature [30, 31]. These algorithms differ in their complexity and performance in terms of burst loss. A wavelength channel is said to be unscheduled at time t when no data burst is using the channel at or after time t. Algorithms which consider unscheduled channels are called Horizon algorithm. A channel is said to be unused for the duration of voids between two successive data bursts and after the last data burst assigned to the channel. Algorithms which consider voids within channels are called void filling algorithm. According to scheduling strategy used scheduling algorithms can be classified as follows:

- *Horizon or Without void filing [30].*
- *With void filling [31].*

Representative of Horizon algorithms are: First Fit Unscheduled Channel (FFUC) [30–33], Latest Available Unused Channel (LAUC) [6,33] and that of void filling algorithms are: First Fit Unscheduled Channel with Void Filling (FFUC-VF) [31], Latest Available Unused Channel with Void Filling (LAUC-VF) [32,34,35] and Minimum End Void (Min-EV) [35].

Working of algorithms is illustrated with the help of Figure 2.4. In Figure 2.4, control packet arrive at a node at time t_{CB}. Duration of payload is t_{burst} and the offset time for the data burst is t_{offset}. The offset time is calculated as:

$$t_{offset} = H * t_{setup} \tag{2.1}$$

where H is number of hops from source to destination and t_{setup} is the time required for processing and switching the control packet. The time at which the first bit of payload arrive at the node is $t_{CB} + t_{offset}$ and the last bit arrive at $t_{CB} + t_{offset} + t_{burst}$.

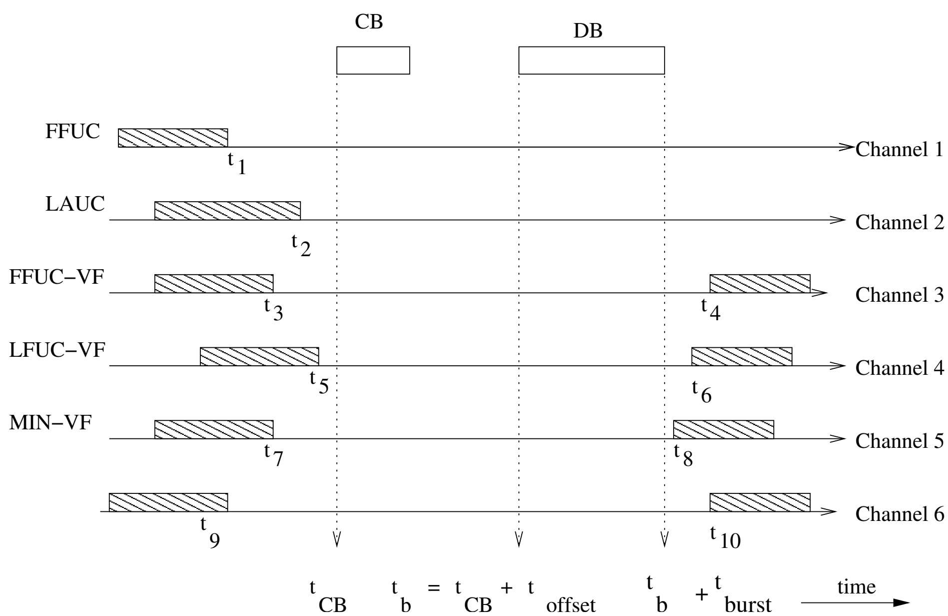

Figure 2.4: Illustration of Burst Scheduling Algorithms

We define unscheduled channel and void channel as following:

unscheduled channel: A wavelength channel is said to be unscheduled at time t when no data burst is using the channel at or after t.

void channel: If a channel is unused for a duration between two successive data bursts.

2.4.1 First Fit Unscheduled Channel (FFUC)

First fit unscheduled channel (FFUC), selects an unscheduled channel for an incoming payload/ data burst [31, 32, 36]. FFUC, keeps the unscheduled time for each data channel. When a control packet arrives, the FFUC algorithm searches all data channels in a fixed order and assigns the data burst to the first channel that is available at or after the arrival time of the payload.

In Figure 2.4, when a control packet arrive at a time t_{CB}, the scheduling algorithm searches for all unused channels. Available unscheduled channels are channel 1 and 2. FFUC selects channel 1, since this is the first available channel. And the channel is reserved for the duration

$$T_{duration} = [t_{CB} + t_{offset}, t_{CB} + t_{offset} + t_{burst}] \tag{2.2}$$

Advantage of the algorithm is speed due to the relatively small number of channels that it checks. The best implementation of the FFUC scheduling algorithm takes $O(log\ n)$ time to schedule a data burst, where n is the number of data channels [21, 37].

Disadvantage of the algorithm is low network resource utilization due to following reasons:

i. does not consider voids that may appear between two already scheduled data bursts as a possible place for fitting the incoming data burst.

ii. stops after first available channel.

2.4.2 Latest Available Unscheduled Channel (LAUC)

Latest available unscheduled channel (LAUC), selects an unscheduled data channel where the void created between consecutive scheduling of data bursts is minimum [31, 36]. In Figure 2.4, channel 1 and 2 are two unscheduled channel at t_b. Scheduling on channel 1 creates a void $(t_b - t_1)$ and in 2 is $(t_b - t_2)$. Since $(t_b - t_1) > (t_b - t_2)$, LAUC selects channel 2 for scheduling.

LAUC has the same complexity as that of FFUC. In addition, LAUC utilizes the network resources better than FFUC.

2.4.3 First Fit Unscheduled Channel With Void Filling (FFUC-VF)

In First fit unscheduled channel with void filling (FFUC-VF) [36], all possible voids are found and the payload is scheduled on the first available void that is suitable for transmission.

In Figure 2.4, voids are available on the channel 3, 4, 5 and the duration of voids are $(t_4 - t_3)$, $(t_6 - t_5)$ and $(t_8 - t_7)$. FFUC-VF selects the channel 3 to schedule the data burst, because channel 3 is the first available void channel.

If n is the number of data bursts currently scheduled on every data channel, then a binary search algorithm takes $log\ n$ time to check that the data channel is eligible or not. Thus the time complexity of the FFUC-VF algorithm is $O(w\ log\ n)$, where w is the number of data channels [21].

2.4.4 Latest Available Unscheduled Channel With Void Filling (LAUC-VF)

Latest available unscheduled channel with void filling (LAUC-VF) [34,36], searches all data channels to find an available void channel for the time interval $(t_b + t_{offset})$ and $(t_b + t_{offset} + t_{burst})$. Then select a channel, such that placement of new data burst create minimal void between newly arrival data burst start time and previous scheduled data burst end time.

In Figure 2.4, channel 3, 4, 5, 6 has such void. The difference between start time of newly arrival data burst and already scheduled data burst whose end time is prior to the start time of new data burst on the channels 3, 4, 5 and 6 are: $(t_b + t_{offset} - t_3)$, $(t_b + t_{offset} - t_4)$, $(t_b + t_{offset} - t_5)$ and $(t_b + t_{offset} - t_6)$ respectively. LAUC-VF select channel having minimum of the above time difference. So it selects channel 4 to schedule the incoming data burst.

To implement LAUC-VF, switching control unit have to store usage information of all data channels. That makes LAUC-VF more complex compared to that of FFUC and LAUC. But it has higher network resource utilization.

2.4.5 Minimum End Void (Min-EV)

A variation of LAUC-VF algorithm is Minimum end void (Min-EV) [35]. It searches all data channels to find an available void channel to schedule the newly arrival data burst. Then, select a channel, such that placement of new data burst create minimal void between already scheduled data bursts start time and newly arrival data bursts end time.

In Figure 2.4, channel 3, 4, 5, 6 has such void. The difference between start time of already scheduled data burst and end time of newly arrival data burst on channel 3, 4, 5 and 6 are: $(t_4 - (t_b + t_{burst}))$, $(t_6 - (t_b + t_{burst}))$, $(t_8 - (t_b + t_{burst}))$ and $(t_{10} - (t_b + t_{burst}))$ respectively. Min-EV selects a channel having minimum of the above value. Therefore, channel 5 is selected.

2.5 Contention Resolution Techniques

Contention occurs when more than one data burst try to reserve the same wavelength channel on an outgoing link. In electronic network, contention is resolve by buffering the contending packets. In OBS network when contention occur one of contending data burst is allowed to reserve the channel, for other data bursts one or a combination of the following contention resolution technique can be applied. If contention can not be resolved then one of the contending data burst is dropped.

Using FDL: In optical network, fiber delay line (FDL) is currently the only way to implement optical buffering. To resolve contention using FDL, one of the contending data burst is passed through FDL.

But it has several limitations. FDL are bulky and require over a kilometer of fiber to delay a single packet for 5 $\mu sec.$ [11], provide only a fixed delay [25] and data leave the FDL in the same order in which they entered [38]. Delay lines are commercially not viable due to the above drawbacks. In generally, FDL can be used with other schemes to improve the performance.

Wavelength Conversion: Wavelength conversion is the process of converting a wavelength on an incoming channel to another wavelength on an outgoing channel [39–41]. To resolve contention using this method, a contending data bursts wavelength is shifted to another wavelength on the designated output link. Thus it increases wavelength re-usability. The concept of wavelength conversion is illustrated in Figure 2.13. Assume that connections are required to be established between node pairs (C, D) and (A, D). Suppose both connections select the wavelength W1 for lightpath establishment. At node B, both connections try for wavelength W1 on link BD. Only one of the connections can be accepted. Let that the connection be (C, D). Since the wavelength W_1 is already used, the connection (A, D) would be dropped in case of wavelength continuity constraint. However in wavelength conversion, node B would convert an incoming wavelength W_1 to an available wavelength W_2 on the link B→D and the connection (A, D) would be established.

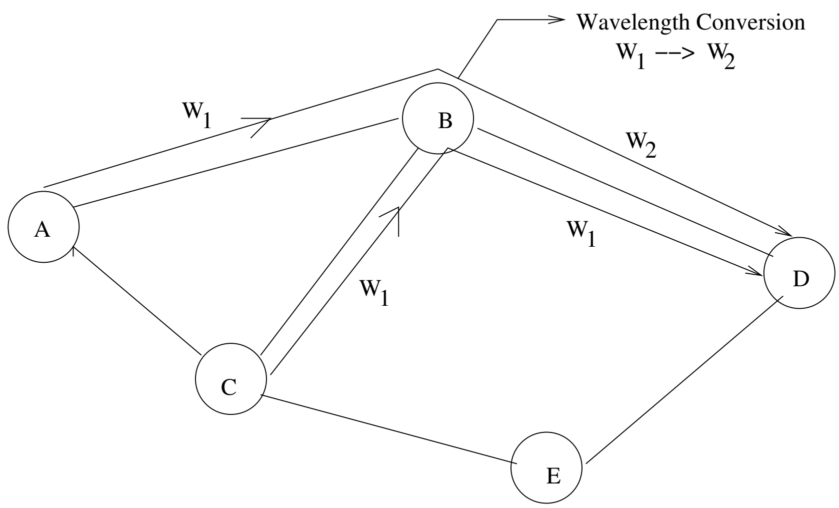

Figure 2.5: Wavelength conversion.

But, the technology is so far immature and is highly expensive for deployment in real network [3].

Deflection Routing: Deflection routing is another approach to resolve contention in OBS networks. In deflection routing one of a contending data burst is sent to a different output port and then follow an alternative route to the destination [38, 40, 42]. Working of deflection routing is explained below. We consider Figure 2.6 for explanation. Suppose both nodes A and B are sending data bursts to node E. Before sending data bursts, nodes A and B send control packets (denoted as C(A, E) and C(B, E)) on control channels for bandwidth reservation for their respective data bursts. Assume, C(B, E) arrives at node C earlier than C(A, E). In this case, the output link CE is reserved by C(B, E). When C(A, E) arrives at node C, the link CE is not available. Without deflection, this data burst will be dropped. In deflection routing, node C checks other output links and selects the deflection link CD which is idle at that time. Then node D forwards B(A, E) on the link D→E and the connection between node A and E would be established.

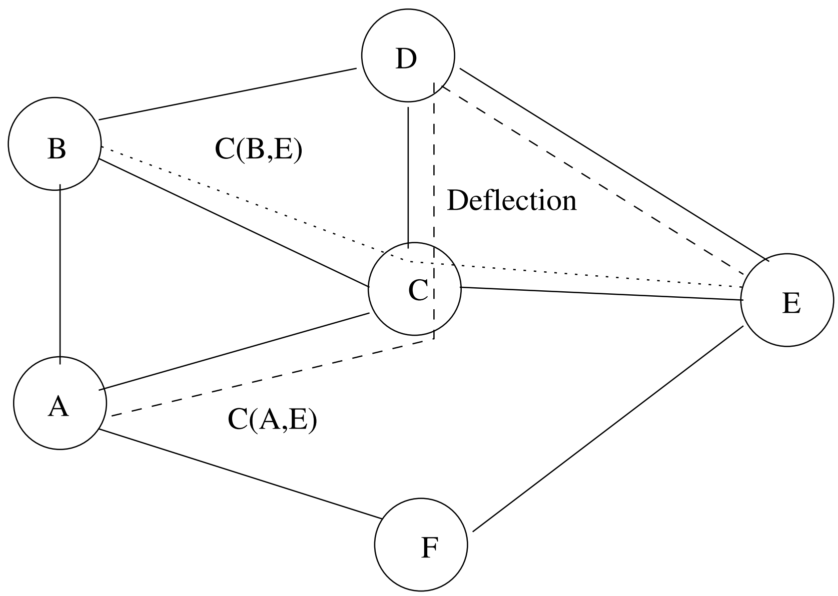

Figure 2.6: Deflection routing.

It has several advantages. Like, it does not require any additional hardware so it can be easily implemented in existing network. But also has some drawbacks. Like end-to-end delay is high, due to follow deflected route which may not be always shortest route [43].

Burst Segmentation: Burst segmentation is a technique to reduce packet loss rather than burst loss [25]. A data burst is composed of a number of segments. When two data bursts are contending, the overlapping segments of one of the contending data burst is dropped rather than the entire data burst. The concept of burst segmentation is shown in Figure 2.7. Burst segmentation gives good performance in terms of packet loss. But it requires a complex control handling to make it reality.

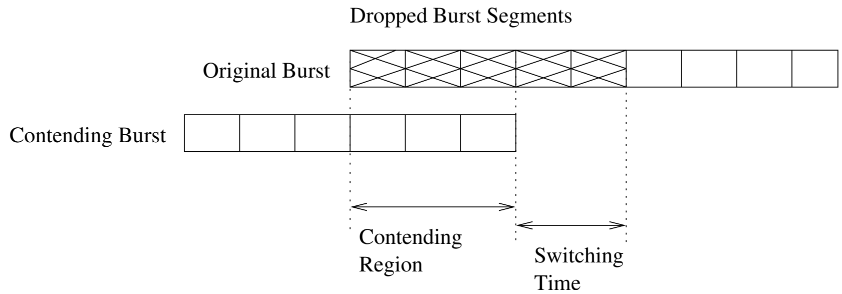

Figure 2.7: Burst Segmentation.

Table 2.2: Comparison of different contention resolutions techniques

Contention Resolution	**Advantages**	**Disadvantages**
Wavelength Conversion	The most efficient solution	Immature and expensive
FDL Buffering	Simple	Increasing end-to-end delay
Deflection Routing	No extra hardware requirement	Out of order arrival
Burst Segmentation	Lower packet loss ratio	Complicated Control handling requirement

Table 2.2 [25] gives a comparison between four contention resolution techniques. When there is no available unscheduled channel, and a contention cannot be resolved by any one of the above techniques, one or more of the contending data bursts are dropped. The policy for selecting which data bursts to drop is referred to as the soft contention resolution policy and is aimed at reducing the overall burst loss rate (BLR), and consequently enhancing link utilization. Several soft contention resolution algorithms have been proposed in [44], including the shortest-drop policy [45] and look-ahead contention resolution [46]. These contention resolution policies are considered as reactive approaches in the sense that they are invoked after contention has occurred.

An alternative approach to reduce network contention is by proactively attempting to avoid network overload through traffic management policies [44].

2.6 Simulation and Results

We evaluated the performance of above mentioned scheduling algorithms through simulation. We compared the performance of different scheduling algorithms reported in the literature and present the result below. For simulation we used ns2 [47] and obs-ns [48] simulator. We consider a network having *two* core node and *fourteen* edge node as shown in Figure 2.8. Simulation parameters are are given in TABLE 2.3. Self-similar traffic as mentioned [48] was consider for trafic generation.

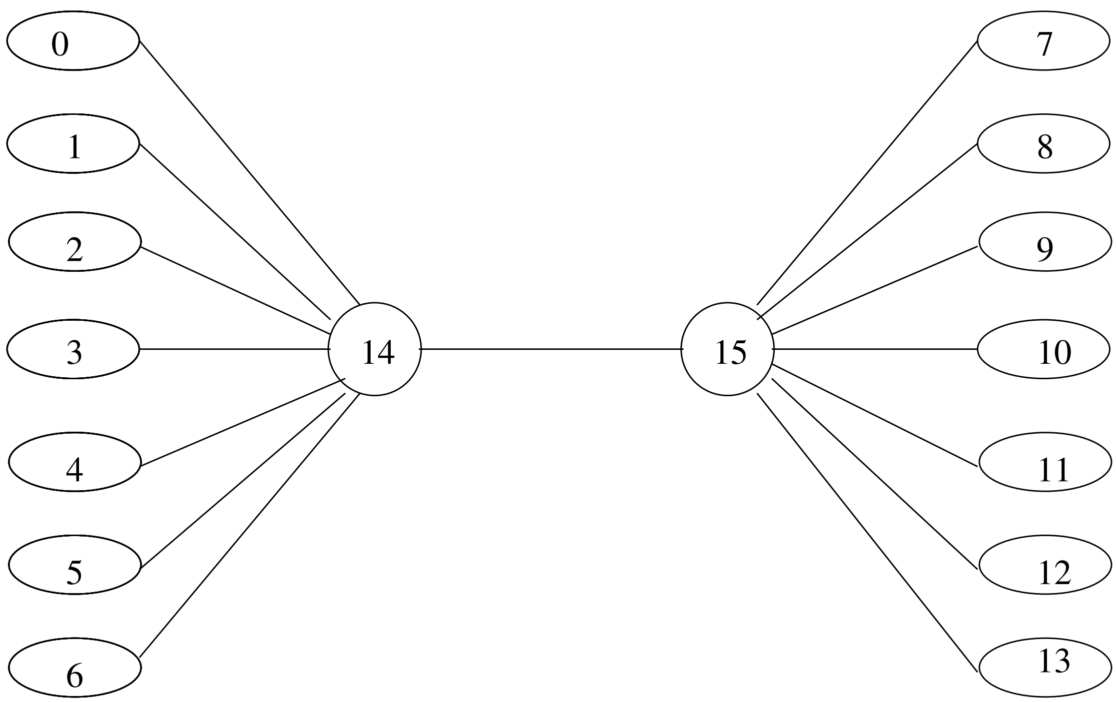

Figure 2.8: Simulated topology.

The comparison of the horizon algorithms (considering the number of channel/link is 8) is shown in Figure 2.9, 2.10 and 2.11. It is observed from Figure 2.9 that burst loss is lower in LAUC. This is due to selection of latest horizon channel.

Figure 2.10 shows the comparison between void filling algorithms. It is observed from the figure that burst loss is higher in FFUC-VF and lower in LAUC-

Table 2.3: Simulation parameters

Parameter	Value
Maximum Burst Size	40 KB
Processing time of control packet	1.5 μs
Number of FDL	0
Edge Node	14
Core Node	2
Bandwidth/channel	5 Gbit/s
Delay on fiber line	1 ms
Total No. of channels/link	8, 5, 3
No. of data channels/link	7, 4, 2
No. of control channels/link	1
Reservation Protocol	JET
Traffic	Self-similar
Mean batch size	2000
Shape parameter for batch size distribution	0
Batch size process Hurst exponent	-0.5
Mean arrival rate	10000.0
Arrival process Hurst exponent	0.5
Std. Dev. interarrival time	1.0e-5

VF. In FFUC-VF data bursts are scheduled to a void channels in a fixed order. Due to this fixed order scheduling few data bursts may get block. In LAUC-VF data bursts are scheduled to a void channel, such that placement of new data burst create minimal void between newly arrival data bursts start time and previous scheduled data bursts end time.

The comparison between horizon and void filling algorithms is shown in Figure 2.11. It is observed from figure that void filling algorithms outperformed the horizon scheduling algorithms in terms of burst loss ratio. This is due to selection of void channels in void filling algorithms.

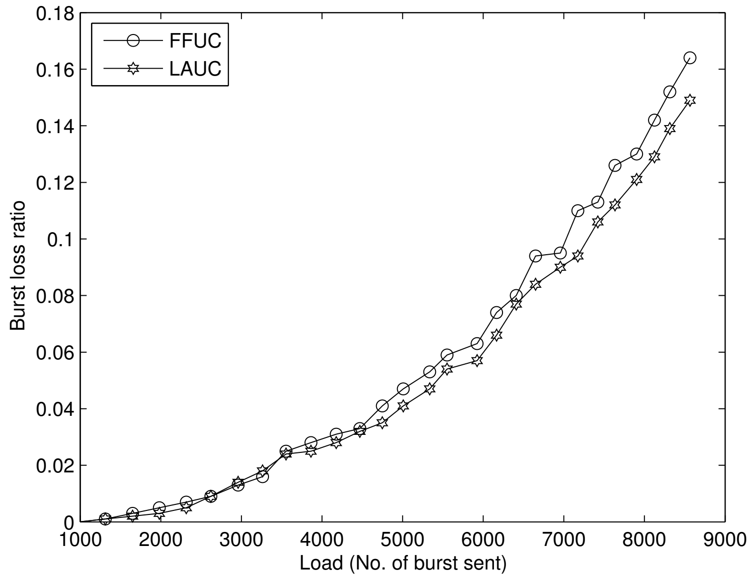

Figure 2.9: Burst loss ratio *vs.* Load in FFUC and LAUC algorithms.

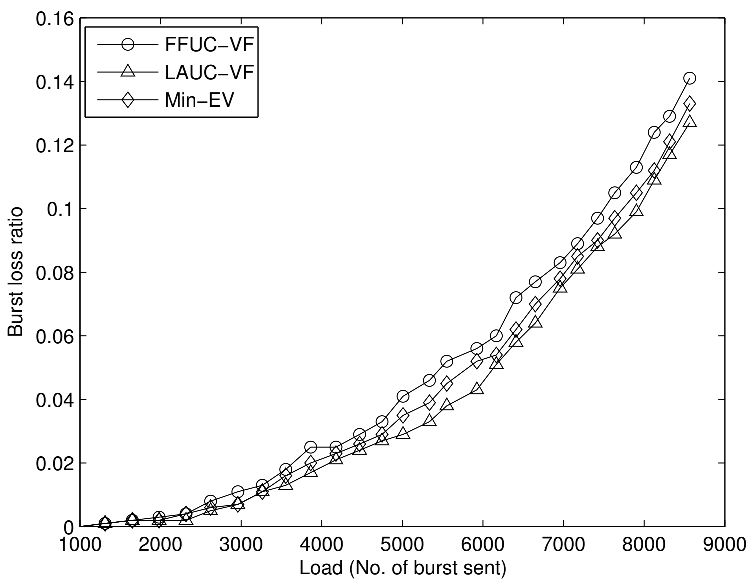

Figure 2.10: Burst loss ratio *vs.* Load in FFUC-VF, LAUC-VF and Min-EV algorithms.

We compared the scheduling algorithms with varying the number of channel/link as 5, 3. The plots are given in 2.12 and 2.13. It is observed from Figure 2.12 and Figure 2.13, that burst loss ratio increases with the decreases of data channels. This is due to unavailability of network resources i.e., data channels.

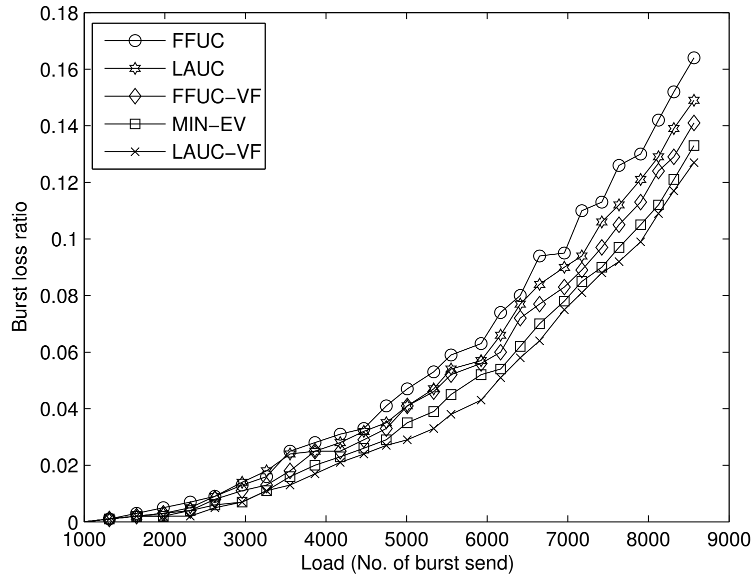

Figure 2.11: Burst loss ratio *vs.* Load of FFUC, LAUC, FFUC-VF, Min-EV and LAUC-VF algorithms, taking number of wavelength *eight*.

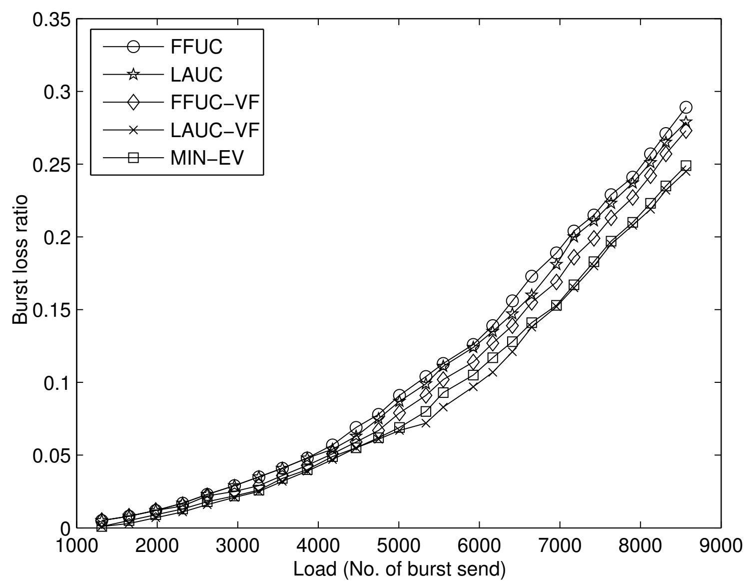

Figure 2.12: Burst loss ratio *vs.* Load of FFUC, LAUC, FFUC-VF, Min-EV and LAUC-VF algorithms, taking number of wavelength *five*.

It is also interesting to see that the performance gap between the algorithms becoming very closer with decreases of number of channels. Still the void filling algorithms give better performance rather than horizon algorithms.

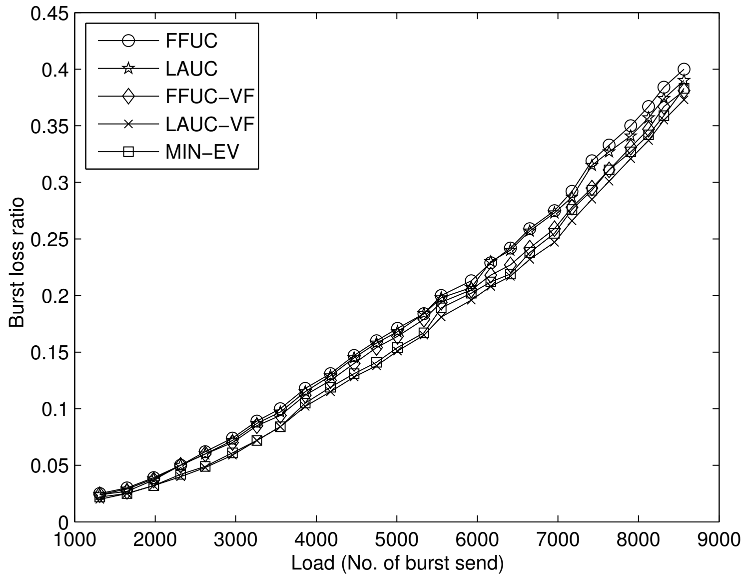

Figure 2.13: Burst loss ratio *vs.* Load of FFUC, LAUC, FFUC-VF, Min-EV and LAUC-VF algorithms, taking number of wavelength *three*.

2.7 Summary

In this chapter we discussed briefly about the issues in optical burst switching, and made a comparison with optical circuit switching and optical packet switching. Different issues, like burst assembly, disassembly, wavelength scheduling algorithms and contention resolution techniques are discussed. We made a comparison among the existing scheduling algorithms through simulation. Our simulation result shows that the burst loss ratio is lower in LAUC among the horizon algorithms and overall burst loss ratio is lower in LAUC-VF among both the horizon and void filling algorithms.

In the next chapter we propose a new channel scheduling algorithm.

Chapter 3

Best Fit Void Filling Algorithm

Optical burst switching (OBS) is emerging as the switching technology for next generation optical networks. Advantages of optical packet switching and circuit switching are combined in OBS and overcoming their limitations. Data (or payload) is separated from control packet. A control packet is sent before the payload to reserve the resources on the path to the destination of payload. When a control packet arrives at an intermediate node a wavelength scheduling algorithm is used by the scheduler to schedule the data burst on an outgoing wavelength channel. The required information to schedule a data burst are arrival time and duration of data burst, which are obtained from the control packet. On the other hand, scheduler keeps availability of time slots on every wavelength channel and schedule a data burst in a channel depending upon the scheduling algorithm it uses. Different scheduling algorithms have been proposed in literature to schedule payload/ data burst. They differ in burst loss and complexity. Depending upon the channel selection strategy, they can be classified as *Horizon* and *Void filling* algorithm. *Horizon* algorithm consider the channels which has no scheduled data burst at or after current time t and the channels are called *Horizon* channels. *Void filling* algorithms consider the channels which have unused duration in between two scheduled data bursts. These are called *Void* channels. The example of *Horizon* algorithms are FFUC, LAUC and *Void filling* algorithms are FFUC-VF, LAUC-VF and Min-EV. *Horizon* algorithms are easy to implement and burst loss ratio is high. Where as burst loss ratio is lower in *Void filling* algorithms but complex switching is required to implement. Among the *void filling* algorithms, burst loss ratio is lower in LAUC-VF and Min-EV. LAUC-VF schedule a data burst in a *void* channel such that the time difference between arrival data bursts starting time and previous scheduled data bursts end time is minimum. Where as Min-EV schedule a data burst in a *void* channel, such that the time difference between a scheduled data bursts start time and arrival data bursts end time is minimum. Both, LAUC-VF and Min-EV consider only one side of a void. There may be a possibility, in which a smaller data burst will be scheduled in a larger void where as a bigger data burst will be dropped. This will lead to higher burst blocking and lower channel utilization.

In this chapter we propose a new channel scheduling algorithm, which attempts to make efficient utilization of existing void within a channel. Thus, giving rise to higher channel utilization and lower blocking probability.

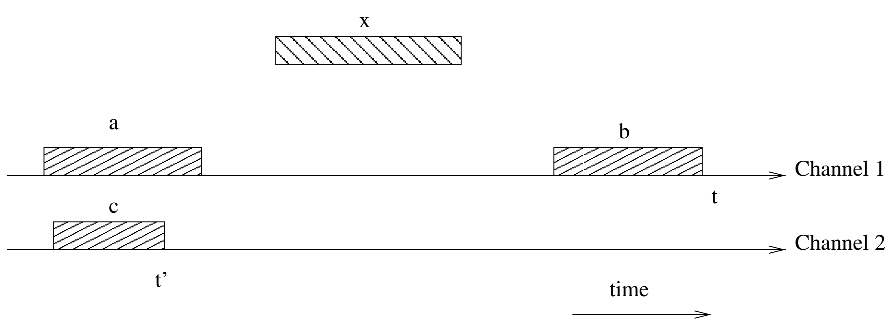

Figure 3.1: A scheduling scenario

3.1 Limitations of Existing Scheduling Algorithms

Horizon scheduling algorithms consider the unscheduled channels to schedule a data burst. It does not consider the availability of void within a channel, which could otherwise be used in channel scheduling. For example consider the Figure 3.1. In this figure there two data bursts a and b are scheduled on channel *1* and data burst c on channel *2*. For horizon scheduling algorithms, channel *1* is available at time instant t and channel *2* is at t'. Suppose a data burst x arrives. Horizon scheduling algorithms will schedule the data burst x on channel *2* as shown in Figure 3.2. They do not consider the voids within a channel. In channel *1* there exist a void between data bursts a and b within which the data burst x

could have been scheduled. Thus, horizon scheduling algorithms are not efficient in terms of channel utilization and gives rise to higher burst loss.

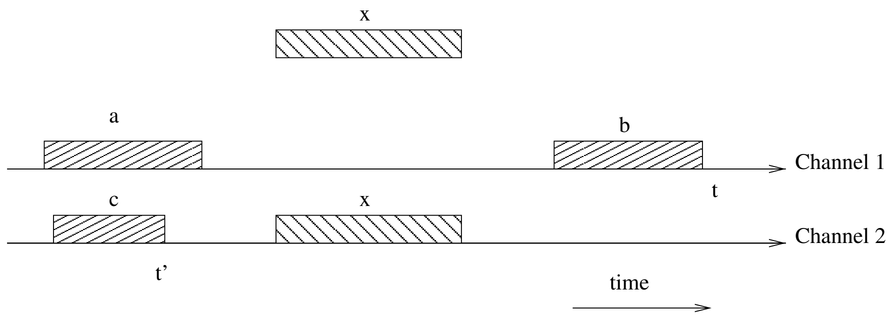

Figure 3.2: Scheduling by horizon algorithms

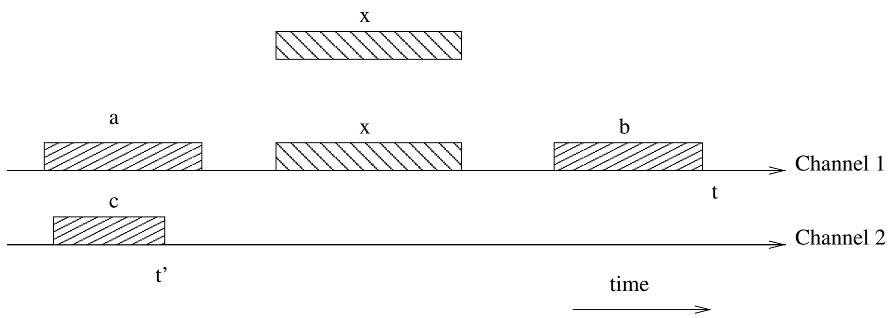

Figure 3.3: Scheduling by void filling algorithms

On the other hand, void filling algorithms consider both unscheduled and void channel to schedule data bursts. For the scenario as shown in Figure 3.1, void filling algorithms will schedule data burst x on channel *1*. Thus, increases the channel utilization. Any data burst arriving between t' and t could be schedule on channel *2*, which otherwise could have been dropped in horizon algorithms. Thus, horizon scheduling algorithms are not efficient in terms of burst loss and channel utilization in comparison to void filling algorithms.

Though void filling algorithms are efficient than horizon scheduling algorithms, but they are not the optimal scheduling algorithms. The limitations of the void filling algorithms such as LAUC-VF and Min-EV algorithms lies in the fact that they consider only one side of a void. LAUC-VF, consider the void created between incoming data bursts start time and previous scheduled data bursts end time. Whereas Min-EV, consider the void created between scheduled data bursts start time and incoming data bursts end time. Due to this smaller size data bursts may be scheduled in a larger void whereas bigger size data bursts may get blocked. In the following subsection a brief description of the limitations in terms of blocking and channel utilization of LAUC-VF and Min-EV void filling algorithms is presented.

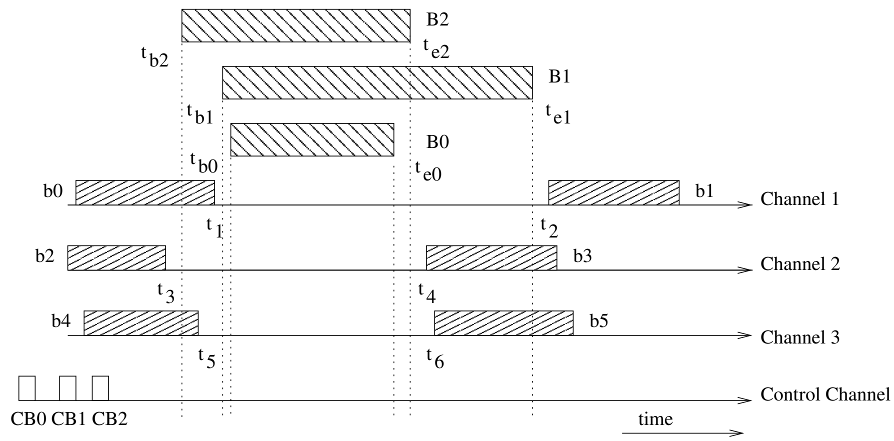

Figure 3.4: Failure of LAUC-VF and Min-EV Algorithms

3.1.1 Blocking in LAUC-VF and Min-EV

In OBS data bursts are of variable lengths. If a smaller data burst arrive earlier than a larger size data burst then void filling algorithm may schedule the smaller data burst on a larger void and the larger size data burst may be dropped due to unavailability of data channel. This can happens in void filling algorithms due to their consideration of one side of a void.

For example consider the Figure 3.4. In this figure data burst *b0* and *b1* are schedule on channel *1*, *b2* and *b3* on channel *2* and *b4* and *b5* on channel *3*. On channel *1* the end time of data burst *b0* is t_1 and start time of data burst *b1* is t_2. Data burst *b2* has end time of t_3 and data burst *b3* has start time of t_4 on channel *2*. Similarly, for data burst *b4*, *t5* is the end time and for data burst *b5*, *t6* is the start time.

Suppose three data bursts *B0*, *B1* and *B2* arrive at a node. Arrivals of control packet for data bursts are shown in control channel. Control packet *CB0* for data burst *B0* has arrived first then *CB1* for data burst *B1*, and finally *CB2* for data burst *B2* arrived in that order. Start time and end time of data burst *B0* is t_{b0} and t_{e0}, for data burst *B1* is t_{b1} and t_{e1} and for data burst *B2* is t_{b2} and t_{e2}.

Scheduling of the data burst onto a channel depend on the type of scheduling algorithm node is using. That is, whether node is using LAUC-VF or Min-EV algorithm. We present below two different cases: (i) Scheduling with LAUC-VF, and (ii) Scheduling with Min-EV algorithms. Since the data burst *B0*, *B1* and *B2* arrive in that order, the scheduler will schedule data burst *B0* first, then *B1* and followed by *B2* in that order.

Case 1: Scheduling using LAUC-VF

LAUC-VF algorithm tries to schedule a data burst on a void, such that difference between the start time of a new data burst and the end time of a previous scheduled data burst whose end time is prior to the new data burst start time will be minimum.

Data burst *b0*, *b2* and *b4* have their end time prior to data burst *B0*'s start time. Differences between the start time of *B0* and end time of *b0*, *b2* and *b4* are $(t_{b0} - t_1)$, $(t_{b0} - t_3)$ and $(t_{b0} - t_5)$ respectively. Of this LAUC-VF, schedule the data burst on a channel, that has the minimum difference. Difference between the start time of data burst *B0* and end time of data burst *b0* is minimum. That is $(t_{b0} - t_1)$ is the minimum value of the three values $(t_{b0} - t_1)$, $(t_{b0} - t_3)$ and $(t_{b0} - t_5)$. So LAUC-VF schedule the data burst *B0* on channel *1*. When the request *CB1* for data burst *B1* arrives, there is no available channel to schedule the data burst *B1*, hence *B1* is dropped. Data burst *B2* can be schedule in channel *2*.

Case 2: Scheduling using Min-EV

In Min-EV scheduling algorithm, an incoming data burst is schedule on a channel, such that the start time of a already scheduled data burst and end time of an incoming data burst is minimum. Here we consider only those schedule data bursts whose start time is after the end time of the incoming data burst. In Figure 3.4, data bursts *b1*, *b3* and *b5* have start time after the end time of data burst *B0*.

Difference between the end time of data burst *B0*, and the start time of data burst *b1*, *b3* and *b5* are $(t_2 - t_{e0})$, $(t_4 - t_{e0})$ and $(t_6 - t_{e0})$ respectively. Of these $(t_4 - t_{e0})$ is the minimum. So the data burst *B0* is schedule on channel *2*. Similarly data burst *B1* is schedule on channel *1*. However, data burst *B2* can not be schedule as there is no wavelength channel is available.

3.1.2 Channel utilization in LAUC-VF and Min-EV

In Figure 3.4 the duration of void in channel *1*, *2* and *3* are $(t_2 - t_1)$, $(t_4 - t_3)$ and $(t_6 - t_5)$ respectively. Higher the fraction of void utilized higher will be channel utilization. Fraction of void utilized is the ratio of the data burst duration scheduled on the void to the void duration.

In Figure 3.4 LAUC-VF schedule data burst *B0* in the void of channel *1*. The fraction of void utilized is $(t_{e0} - t_{b0})/(t_2 - t_1)$. Of these the fraction $(t_{e0} - t_{b0})/(t_2 - t_1)$ is smaller. Scheduling data burst *B0*, in channel *1*, *2* and *3*, the fraction of void utilized will be $(t_{e0} - t_{b0})/(t_2 - t_1)$, $(t_{e0} - t_{b0})/(t_4 - t_3)$ and $(t_{e0} - t_{b0})/(t_6 - t_5)$ respectively. This is because $(t_2 - t_1) > (t_4 - t_3) > (t_6 - t_5)$. Thus scheduling data burst *B0* in channel *1*, gives rise to inefficient channel utilization. Moreover, this creates a void $(t_2 - t_{e0})$ of considerable duration .

Min-EV algorithm schedule data burst *B0* in channel *2*. Fraction of void utilized is higher than that of scheduling on channel *1* and lower than scheduling on channel *3*. Scheduling *B0* in channel *2*, void of channel *3* remains utilized.

Thus, it is observed that the channel utilization is lower in both LAUC-VF and Min-EV. This is because both algorithm consider only one side of a void i.e., either the start or end side of a void. Next we propose a new channel scheduling algorithm which considers both end of a void in scheduling and it utilizes void efficiently.

3.2 Best Fit Void Filling Algorithm

In this section we propose a new scheduling algorithm called Best Fit Void Filling (BFVF), which attempts to maximize the channel utilization and minimize the burst loss. Our propose algorithm first selects all possible void channels, on which the data burst can be scheduled. Then selects one of the possible void channel such that the void utilization factor is maximum. We calculate the void utilization factor as:

$$utilization = (a * 100)/x \tag{3.1}$$

where a is the data burst length and x is the void length.

In Figure 3.4, data burst *B0* can be schedule any one of the channel *1*, *2* and *3*. Void utilization factor for *B0* on channel *1*, *2* and *3* are $(t_{e0} - t_{b0})/(t_2 - t_1)$, $(t_{e0} - t_{b0})/(t_4 - t_3)$ and $(t_{e0} - t_{b0})/(t_6 - t_5)$ respectively. Void utilization factor for channel *3* is maximum, since $(t_6 - t_5) < (t_4 - t_3) < (t_6 - t_5)$. So BFVF algorithms selects channel *3* to schedule the data burst *B0*. Similarly data burst *B1* is schedule on channel *1* and *B2* on channel *2*. In our propose algorithm all three data burst *B0*, *B1* and *B2* can be scheduled on channel *3*,*1* and *2* respectively as shown in Figure 3.5. Thus the channel utilization is higher and burst loss ratio is lower in our propose scheme than in LAUC-VF and Min-EV.

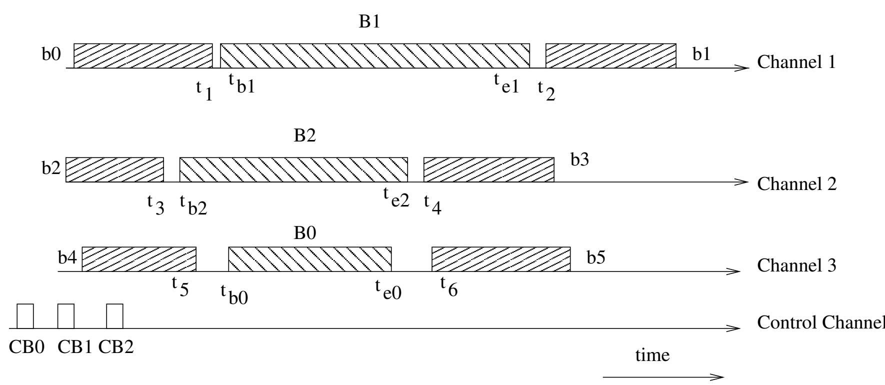

Figure 3.5: Scheduling by BFVF algorithm

We workout an example to show the void utilization in LAUC-VF, Min-EV and our proposed BFVF algorithm. We assume the following numerical values:

$t2 - t1 = 12\mu s$

$t4 - t3 = 10\mu s$

$t6 - t5 = 8\mu s$

and length of data burst B0 is $= 5\mu s$

Void utilization in

LAUC-VF, $utilization = (5 * 100)/12 = 41.67\%$

MIN-EV, $utilization = (5 * 100)/10 = 50\%$

BFVF, $utilization = (5 * 100)/8 = 62.5\%$

This shows that void utilization is higher in our proposed BFVF algorithm.

Formally, we describe BFVF algorithm below. The following notations are used in our algorithm:

$length_b$: Length of the incoming data burst,

$length_v(i)$: Void length in channel i,

$start_b$: Start time of a data burst,

$start_v(i)$: Start time of void in channel i and

data channel : Data channel selected by the algorithm to schedule the data burst.

Best Fit Void Filling Algorithm

Input: $start_b$, $length_b$

Output: *data channel*

Step 1: Select all possible schedulable void channels. A void channel i is said to be schedulable if $start_b > start_v(i)$ and $length_b < length_v(i)$. If no schedulable void channel exists then goto Step 4.

Step 2: Calculate the channel utilization factor for all schedulable void channel found in Step 1.

Step 3: Find a channel j such that it has the maximum channel utilization factor as found in Step 2. Output channel j as the required data channel. Stop.

Step 4: Schedule the data burst according to LAUC algorithm.
Stop.

Step 1 of the algorithm is to find a schedulable *void* channel. If no such void channel is available then the data burst is scheduled as in LAUC algorithm.

3.3 Simulation and Results

We compare the performance of our proposed BFVF algorithms with that of LAUC-VF and Min-EV algorithm through simulation. For simulation, we have considered, obs-ns simulator that runs on the top of ns2 simulator. Performance metrices considered for comparison are: (i) burst loss ratio *vs.* load (in number of data burst sent) (ii) link utilization *vs.* load (in number of data burst sent). Topology considered for simulation is shown in Figure 2.8. Parameters considered for simulation is shown in TABLE 3.1.

Table 3.1: Simulation parameters

Parameter	**Value**
Maximum Burst Size	40 KB
Processing time of control packet	1.5 μs
Number of FDL	0
Edge Node	14
Core Node	2
Bandwidth/channel	5 Gbit/s
Delay on fiber line	1 ms
Total No. of channels/link	8, 5, 3
No. of data channels/link	7, 4, 2
No. of control channels/link	1
Reservation Protocol	JET
Traffic	Self-similar

We plot the burst loss ratio *vs.* load in Figure 3.6. Burst loss ratio is calculated as number of burst loss divided by number of burst sent. It is observed from the Figure 3.7 that the burst loss ratio increases with increases in load, in all the three schemes. However the increase in our proposed BFVF scheme is lower than that of LAUC-VF and Min-EV algorithm. This is due to the efficient utilization of void channels in our BFVF scheme, which is not the case in LAUC-VF and Min-EV algorithm. In LAUC-VF and Min-EV algorithm a smaller data burst may be scheduled to a larger void, which is not the case in of BFVF algorithm.

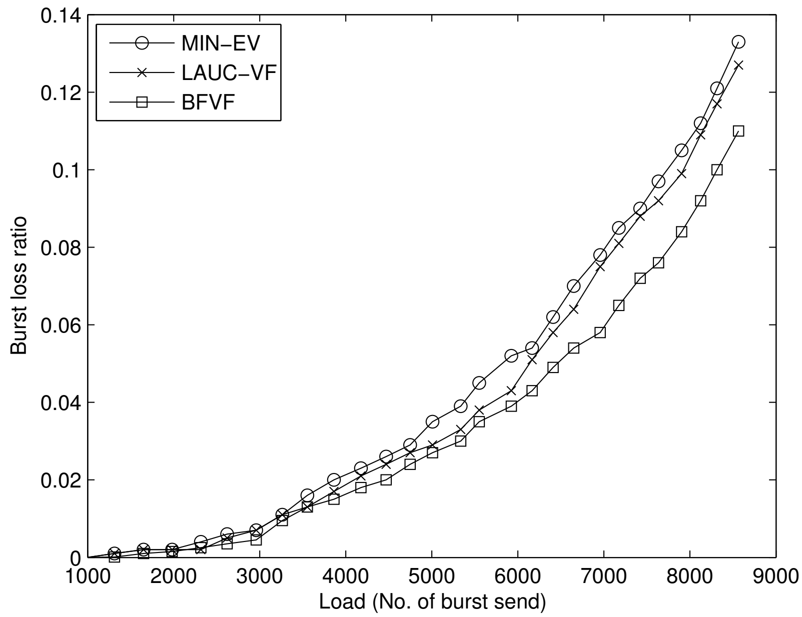

Figure 3.6: Burst loss ratio *vs.* Load in BFVF, LAUC-VF and Min-EV algorithm for *eight* taking number of wavelength channel.

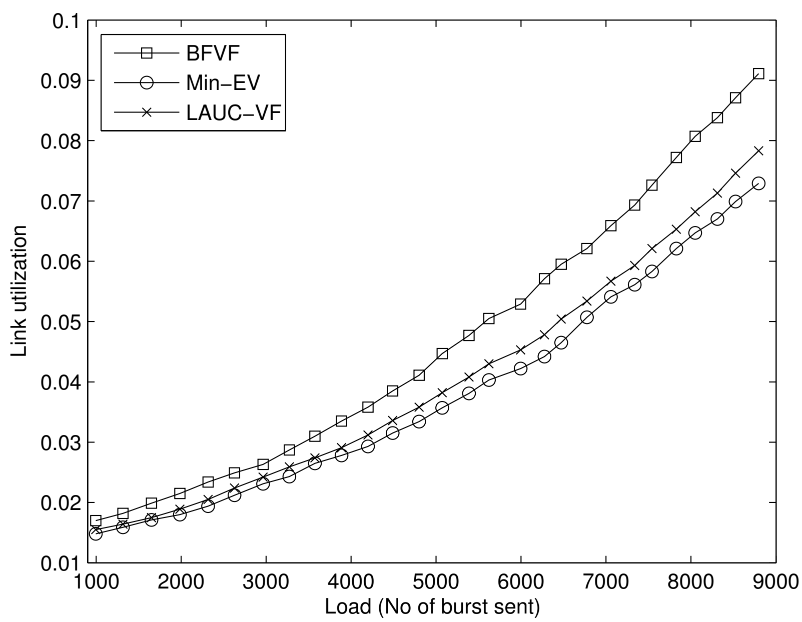

Figure 3.7: Link utilization *vs.* Load in BFVF, LAUC-VF and Min-EV algorithm for *eight* taking number of wavelength channel.

We varied the number of wavelength channel to *five* and *three* to study the effect of decreased wavelength channel on burst loss ratio. Plot for burst loss ratio *vs.* load for *five* number of wavelength in Figure 3.8 and for *three* wavelength channel in Figure **??**

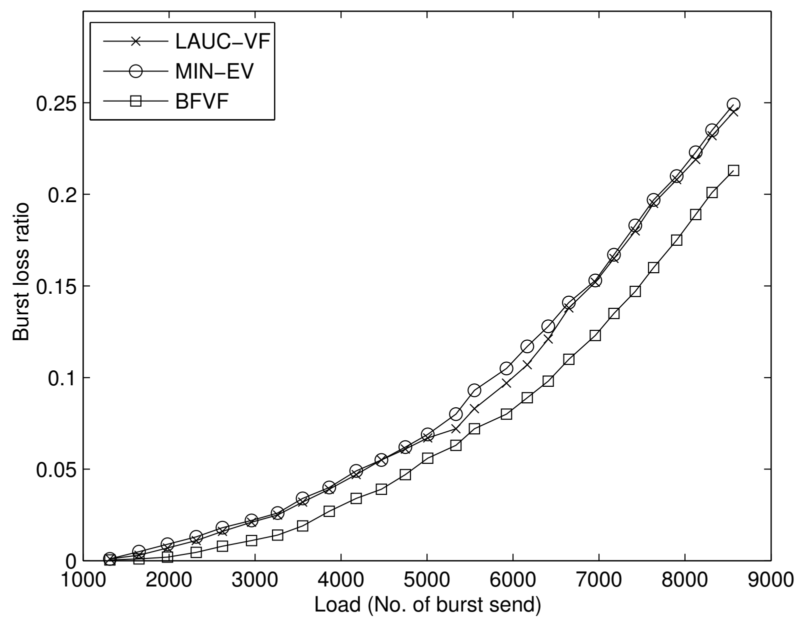

Figure 3.8: Burst loss ratio *vs.* Load in BFVF, LAUC-VF and Min-EV algorithm, for *five* number of wavelength channel.

We plot the graph for link utilization in Figure 3.10. It is observed from the figure, link utilization in BFVF algorithm is more than sixty percent whereas in LAUC-VF and Min-EV it is just fifty percent.

Time complexity of our proposed algorithm is same as in LAUC-VF and Min-EV and is equal to $O(w\ log\ n)$, where n is the number of data bursts currently scheduled on every data channel and w is the number of data channels in a link.

3.4 Summary

In this chapter we discuss performance of horizon and void filling scheduling algorithm. It is found that the void filling scheduling algorithm performs better than the horizon scheduling algorithms. However, there are limitations to the existing

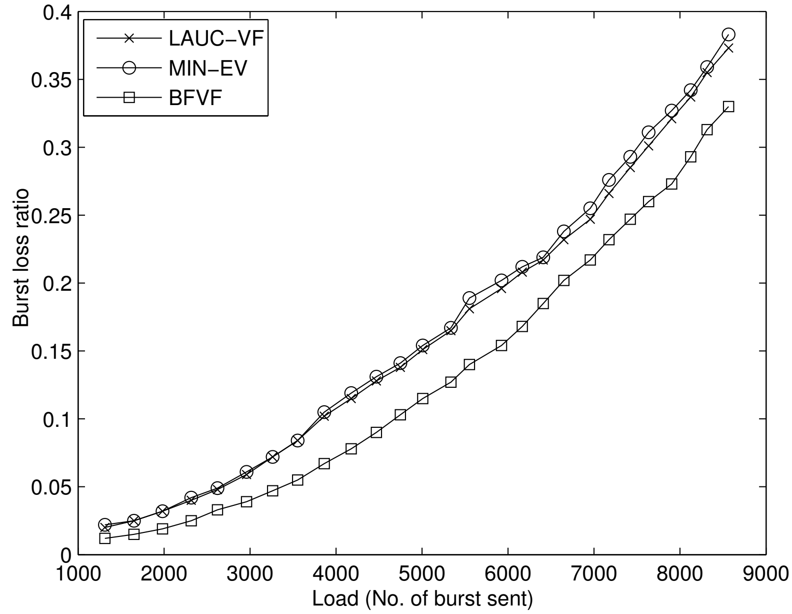

Figure 3.9: Burst loss ratio *vs.* Load in BFVF, LAUC-VF and Min-EV algorithm, for *three* number of wavelength channel.

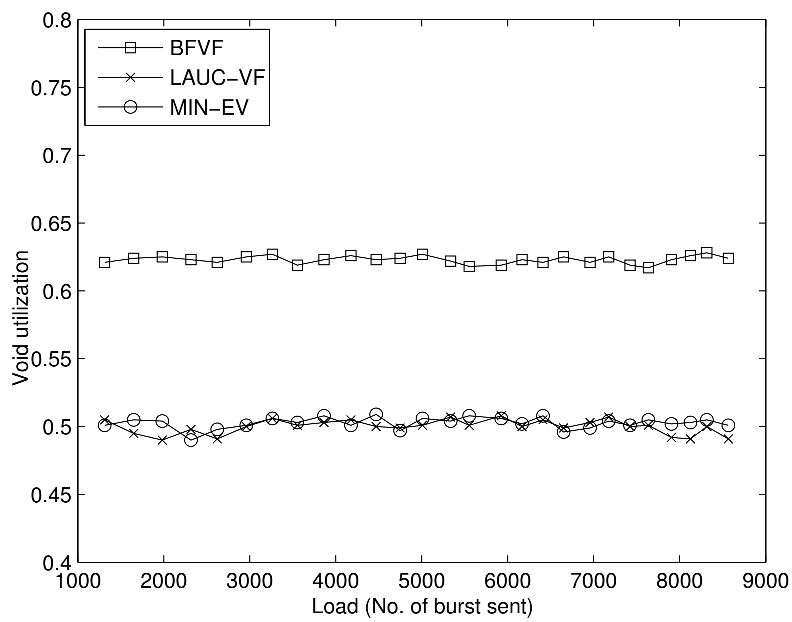

Figure 3.10: Void Utilization *vs.* Number of Voids in BFVF, LAUC-VF and Min-EV algorithm

void filling scheduling algorithms. This limitation is mainly due to that; the existing schemes consider either the start time of the new data burst and end time of the previously scheduled data burst or start time of previously scheduled data burst and the end time of the new data burst. They do not take into account the data burst length and void length.

We proposed an algorithm called BFVF, which takes the arrival data burst length and void length into account in scheduling. Proposed scheme calculates the void utilization factor, and schedule the new data burst into a void channel having maximum void utilization factor. For non-availability of void channel, scheduling takes place as in LAUC scheduling algorithm.

The proposed scheme is compared with LAUC-VF and Min-EV. It is found that the proposed scheme perform better in term of burst loss ratio and channel utilization.

In the next chapter we propose a cluster-based technique to minimize the occurrence of contention in OBS networks.

Chapter 4

Cluster Based Contention Minimization

Contention is an important issue in OBS networks. Contention occurs when more than one data burst try to use a same wavelength channel at an outgoing link. In packet switching network contention is solved by storing the contending packets in a buffer and forwarding other. Buffering of signal in optical domain difficult. Though, fiber delay lines (FDL) are proposed to used as buffer, there are many limitations when deployed in real network. Such as they are bulky, can not be access randomly as in electronics domain and provide delay only for a fixed duration. A number of techniques have been proposed in the literature to resolve contention [25, 38, 40]. But none of these technique try to reduce the occurrences of contention in the network.

In this chapter we propose a technique to minimize the occurrence of contention in OBS networks. In the propose scheme a given network is logically divided to a number of clusters. A node within each cluster is selected as cluster head, which keeps track of the resources available in the network. Cluster head exchange the status of the resources among themselves to maintain an up-to-date information. A node within a cluster that wishes to send a data burst make request it's cluster head for an available wavelength channel on the path of the data burst. A cluster head send a positive or negative reply depending on the availability of wavelength channel. A node on receiving positive reply transmit OBS control packet followed by the data burst on that channel else drop the data burst.

4.1 Cluster Formation

In this section we explain the partition of a given network into number of sub-networks called clusters. To explain our cluster formation we consider a *fourteen* node NSFNET [49] as shown in Figure 4.1. We use the following notations:

N : A set representing the number of nodes in the network,

D : A set representing the degree of each nodes in the network. Each element of the set D is represented as n^d, where d is the degree of node n in the given network, and

$cluster_i$: A set representing the number of elements in the i^{th} cluster.

For forming a cluster, first a cluster head is selected. Nodes are then added to the cluster head progressively to form a cluster around it. A node that has the maximum degree in the set D is selected as cluster head. Nodes that are one-hop distance away from the cluster head is added to it to form a cluster. Addition of all nodes which are one-hop distance away from the cluster head forms the first cluster. Nodes which are included in the first cluster are deleted from the set N and their degree from the set D. After the first cluster is formed a node with highest degree in set N is selected as cluster head for the second cluster. Nodes in the set N which are one-hop distance away from this cluster head are included to from the second cluster. Nodes which are included in the second cluster are deleted from the set N and their degree from the set D. Above process is repeated until the set N becomes empty.

Clusters formed in the above process may contain a single node. Thus, we consider number of nodes in a cluster as the parameter in cluster formation. This is a tunable parameter. For our case we have consider minimum number of nodes in a cluster to be *four*. Nodes in a cluster having less than *four* nodes, are attached to other clusters depending on their hop distance from remaining cluster heads. We have assume minimum hop distance as the criteria to add node in cluster.

We illustrate below, the partition of a *fourteen* node NSFNET [49] as shown in Figure 4.1 into clusters. Initially there are no clusters and the contents of set N and D is given below:

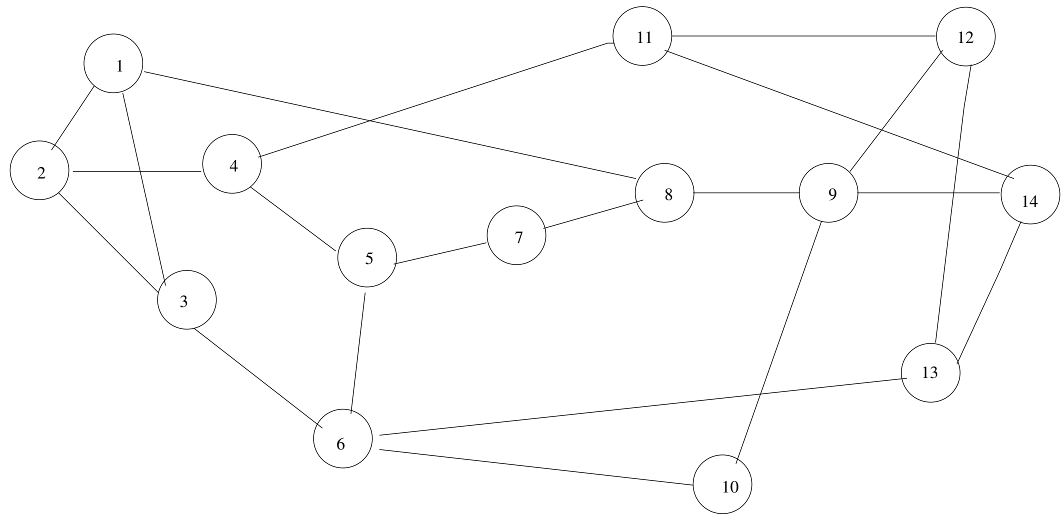

Figure 4.1: A 14 node NSFNET

$N = \{1,\ \ 2,\ 3,\ 4,\ 5,\ \ 6,\ 7,\ 8,\ 9,\ \ 10,\ 11,\ 12,\ \ 13,\ 14\}$
$D = \{1^3, 2^3, 3^3, 4^3, 5^3, 6^4, 7^2, 8^3, 9^4, 10^2, 11^3, 12^3, 13^3, 14^3\}$

To form the first cluster a cluster head is to be selected. From set D it is observed that, node having maximum degree in a *fourteen* node NSFNET is *four*. Both nodes 6 and 9 have the degree of *four*. So, either of the nodes can be selected as the cluster head. We select node 6 as the cluster head for the first cluster. Nodes which are adjacent to node 6 *viz*, 3, 5, 10, 13 are included in the first cluster. Nodes in the first cluster *viz* 3, 5, 6, 10 and 13 are deleted from the set N and their degree $3^3, 5^3, 6^4, 10^2, 13^3$ respectively from the set D. Elements in the set N, D and $cluster_1$ after formation of the first cluster is shown below:

$N = \{1,\ \ 2,\ 4,\ 7,\ 8,\ \ 9,\ 11,\ 12,\ 14\}$
$D = \{1^3, 2^3, 4^3, 7^2, 8^3, 9^4, 11^3, 12^3, 14^3\}$
$cluster_1 = \{6, 3, 5, 10, 13\}$

To form second cluster, we select the maximum degree node 9 in set N as shown above as the cluster head. Nodes which are at one hop distance from node 9 *viz.* 8, 12 and 14 are added to form the second cluster. Nodes in the second cluster *viz.* 8,9,12 and 14 are deleted from the set N and their degree $9^4, 8^3, 12^3, 14^3$ respectively from the set D. Elements of the set N, D and $cluster_2$ after *second* cluster is shown below:

$N = \{1,\ \ 2,\ 4,\ 7,\ 11\}$
$D = \{1^3, 2^3, 4^3, 7^2, 11^3\}$
$cluster_2 = \{9, 8, 12, 14\}$

Above procedure is repeated until N becomes an empty set. We have shown below the elements of set N, D and that of a cluster after each cluster formation starting from *third* cluster.

The elements of N, D and $cluster_3$ after *third* cluster is shown below:

$N = \{4,\ 7,\ \ 11\}$
$D = \{4^3, 7^2, 11^3\}$
$cluster_3 = \{1, 2\}$

The elements of N, D and $cluster_4$ after *fourth* cluster is shown below:

$N = \{7\}$
$D = \{7^2\}$
$cluster_4 = \{4, 11\}$

The elements of N, D and $cluster_5$ after *fifth* cluster is shown below:

$N = \{\phi\}$
$D = \{\phi\}$
$cluster_5 = \{7\}$

NSFNET of Figure 4.1 is partitioned into *five* clusters as shown in Figure

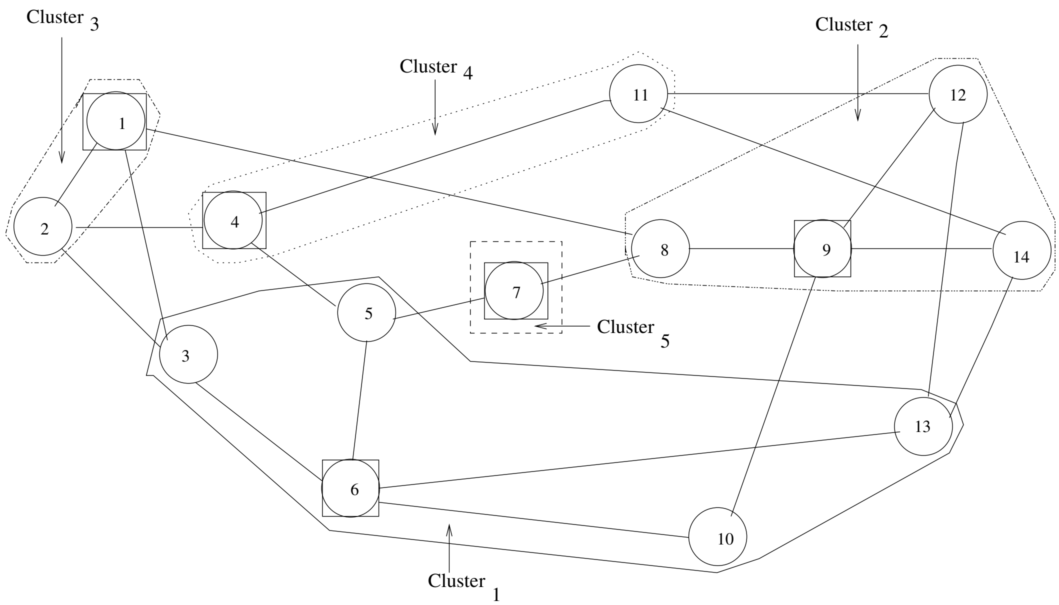

Figure 4.2: Intermediate clusters formed in NSFNET

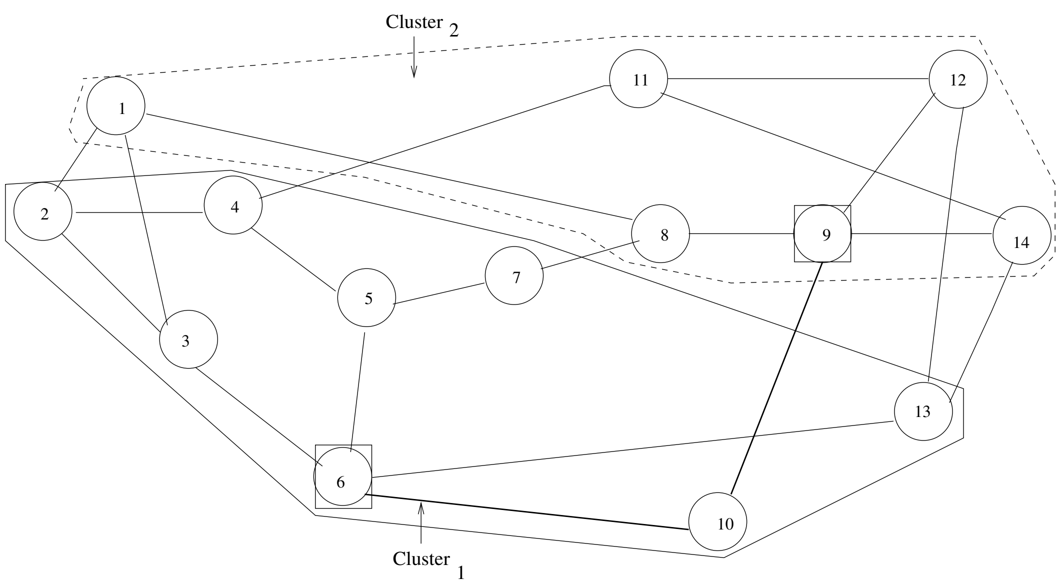

Figure 4.3: Final clusters formed in a *fourteen node* node NSFNET

4.2. In Figure 4.2 a node encircled within a square indicates a cluster heads. We are interested only in those clusters that have at least *four* nodes. From Figure 4.1, $cluster_3, cluster_4$ and $cluster_5$ have 2, 2, and 1 node respectively. Elements in these clusters are added either in $cluster_1$ or $cluster_2$ depending on their hop distance from the cluster head of $cluster_1$ and $cluster_2$. In case of a tie any one of them is selected randomly. Node 1 is *two* hop away from cluster head 6 and 9.

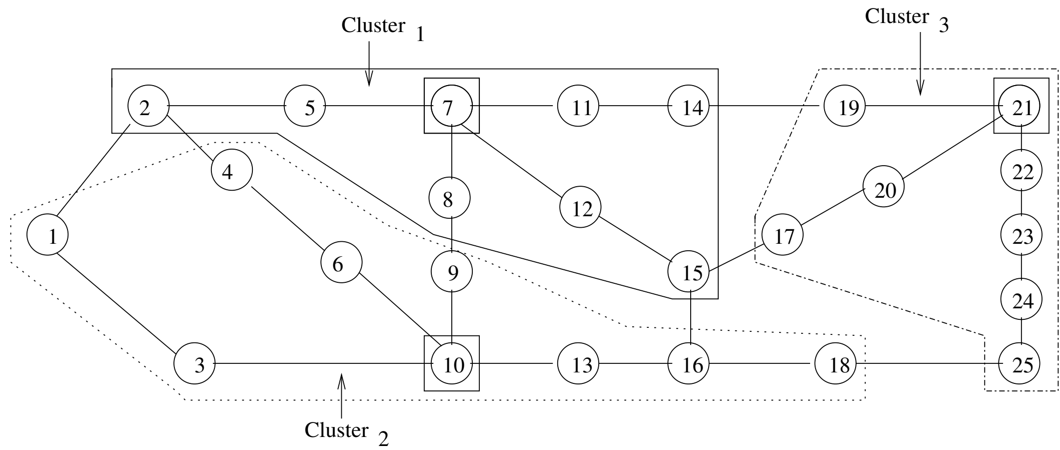

Figure 4.4: Clusters in a *twenty five* node ARPANET

We randomly add node 1 in $cluster_2$. Node 2 is *two* hop away from cluster head 6 and *three* hop away from cluster head 9. So we add node 2 to $cluster_1$. Similarly node 4 and 7 are added in $cluster_1$ and node 11 in $cluster_2$. Finally we obtain *two* clusters as shown in Figure 4.3. Elements in the final two clusters are shown below:

$cluster_1 = \{6, 2, 3, 4, 5, 7, 10, 13\}$

$cluster_2 = \{9, 1, 8, 11, 12, 14\}$

Next, we consider a *twenty five* node ARPANET [49] shown in Figure 4.4 and a *thirty three* node ARPANET [49] shown in Figure 4.5 to show cluster formation. First element in each cluster denotes the cluster head for that cluster.

Clusters for *twenty five* node ARPANET is shown in Figure 4.4. Elements of clusters are shown below:

$cluster_1 = \{7, 2, 5, 8, 11, 12, 14, 15\}$

$cluster_2 = \{10, 1, 3, 4, 6, 9, 13, 16, 18\}$

$cluster_3 = \{21, 17, 19, 20, 22, 23, 24, 25\}$

Clusters of *thirty three* node ARPANET is shown in Figure 4.5. Elements of the clusters are shown below:

$cluster_1 = \{1, 2, 3, 6, 8, 12\}$

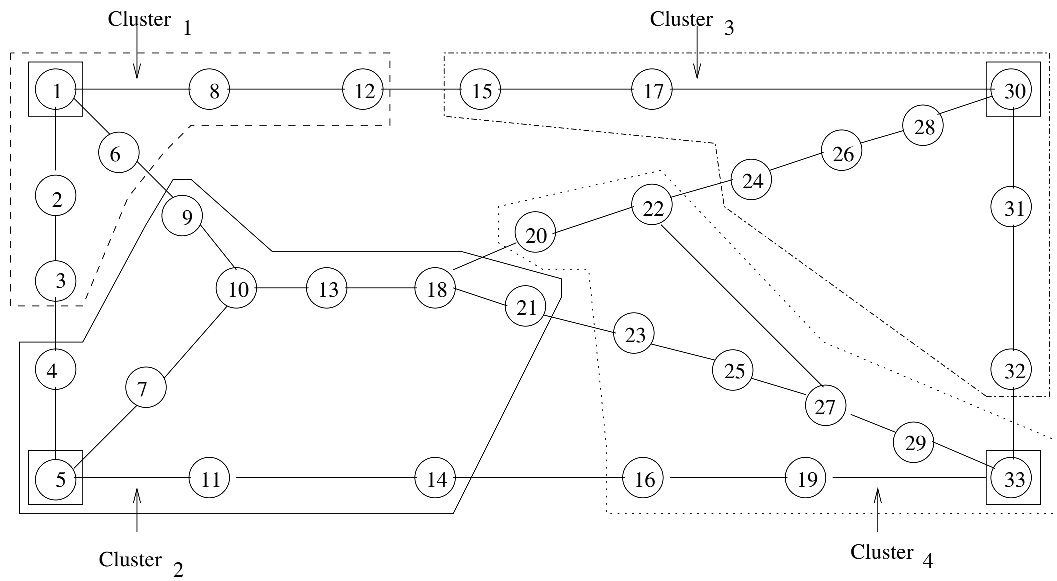

Figure 4.5: Clusters in a *thirty three* node ARPANET

$cluster_2 = \{5, 4, 7, 9, 10, 11, 13, 14, 18, 21\}$

$cluster_3 = \{30, 15, 17, 24, 26, 28, 31, 33\}$

$cluster_4 = \{33, 16, 19, 20, 22, 23, 25, 27, 29\}$

Step 2 of Table 4.1 algorithm partition a given network into number of clusters which may contain a single node. We are interested to form clusters with a certain minimum number of nodes. Minimum number of nodes in a cluster is a tunable parameter in our clustering algorithm. We have consider clusters with minimum of *four* number of nodes. Nodes in a cluster with lesser than the minimum desired numbers of nodes are to be added to other clusters having the desired number of nodes. Step 4, of the algorithm add the nodes in a cluster having less than the desired number to those clusters having the desired number of nodes.

After performing the step 4 we obtain clusters with desired minimum number of nodes. In next section we explain how contention is minimized in our propose technique.

Table 4.1: Cluster Formation Algorithm

1. Initialize N and D. Perform $i \leftarrow 1$.
2. While ($N \neq \phi$) do the following
   a. Find a node having maximum degree in the set D. Let this node be called *MaxDegree*. Mark the node *MaxDegree* as the cluster head of i^{th} cluster and add it to $cluster_i$. Initially $cluster_i$ is empty.
   b. Include all node k in $cluster_i$ such that $k \in N$ and node k is adjacent to cluster head *MaxDegree* of i^{th} cluster.
   c. Delete all nodes which are included in the i^{th} cluster, $cluster_i$ from set N and their corresponding degree from set D.
3. Sort the clusters formed in step 2 according to the number of elements in the cluster, such that number of elements in $cluster_i \leq cluster_{i+1}$.
4. For all cluster i whose number of elements in the $cluster_i \leq$ desired number of elements in a cluster do the following

   while ($cluster_i \neq \phi$), remove a node form $cluster_i$. Let this node be called z. Include node z in $cluster_j$ where $j \neq i$, such that

   i. number of nodes in $cluster_j \geq$ desired number of nodes in a cluster, and
   ii. the hop distance between node z and head node of $cluster_j$ is minimum.

      Delete the node z from $cluster_i$.

4.2 Propose Contention Minimization Technique

Our proposed cluster based contention minimization technique operates in two stages. In the first stage a given network is partitioned into number of subnetwork called clusters. In the second stage the data transmission takes places. Cluster formation is explained in Section 4.1. In this section, we explain our contention minimization technique. Our proposed technique is based on the observation that " more occurrence of contention gives rise to higher burst loss and less occurrence of contention gives rise to lower burst loss ". Burst loss in a network can be reduced by minimizing the occurrence of contention in the network. This work is an attempt to minimize the occurrence of contention in the network. In proposed scheme, *four* control packets in addition to OBS control packets are used. Purpose of additional control packets is explained below:

Channel request packet (CRP) : A node prior to sending OBS control packet, request its cluster head for an available wavelength channel on the path to destination by sending a channel request packet (CRP).

Channel Reply Packet This packet is sent by cluster head in response to CRP. Cluster head send a positive channel reply (PCR) if a wavelength channel is available on the path from the source to destination else a negative channel reply (NCR).

Resource Update Packet (RUP) Clusters head exchange resource information using resource update packet. If the response of a cluster head to channel request is positive then it send a resource update packet to other clusters head in the network. Clusters head update the usage of the wavelength channel in their database on receiving RUP packet which contains information on the usage of a wavelength channel sent in PCR packet.

We consider Figure 4.6 to explain the process of updating the usage of wavelength channels at a cluster head on receiving the RUP packet from another cluster head. A wavelength channel on a link may be free or it might have been already reserved for certain duration. Accordingly we identified three possible scenarios.

Scenario 1, a wavelength channel as seen by a cluster head is free *i.e.*, it is not being used. In this case the channel usage for the wavelength channel is updated to that received in RUP packet from other cluster head. This scenario is depicted

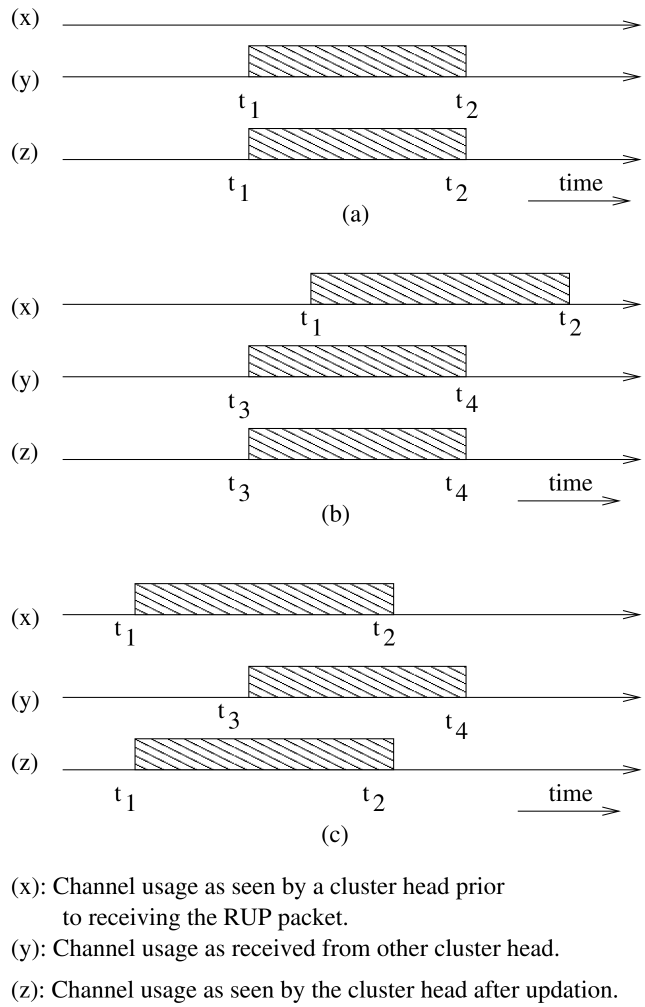

Figure 4.6: Update of channel usage information in a cluster head

in Figure 4.6(a).

In **scenario 2** and **3** the wavelength channel under consideration is reserved for certain time duration. In these cases the channel update depends upon the time instant at which the channel is reserved. Scenario 2 is depicted in Figure 4.6(b). In this scenario, the channel usage as seen by the cluster head prior to receiving the RUP packet is shown in *(x)* of Figure 4.6(b). The channel usage as received by the cluster head in a RUP packet from another cluster head is shown in *(y)* of Figure 4.6(b). In this scenario a source has reserved the wavelength channel for the duration t_1 to t_2 [shown in *(x)* of Figure 4.6(b)] and another source trying to reserve the same wavelength channel for the duration t_3 to t_4 [shown in *(y)* of Figure 4.6(b)]. Since $t_3 < t_1$, cluster head will update the channel usage, to that it received in RUP packet as shown in *(z)* of Figure 4.6(b). We have assumed, a node that makes reservation request at an earlier time instant will succeed.

In scenario 3, the time instant t_1 at which the channel is reserved as shown in *(x)* of Figure 4.6(c) is earlier than the time instant t_3 for which the reservation request is made. Since $t_1 < t_3$, the RUP packet is dropped at the cluster head and the channel usage is not modified. This is shown in *(z)* of Figure 4.6(c).

In the proposed scheme, there exist a lightpath among cluster heads so that the RUP packet is not processed at any intermediate nodes. Cluster heads maintain up-to-date information about resources available in the network, by exchanging RUP packet among themselves. Initially a cluster head sees all resources as available. A cluster head sends a RUP packet to other cluster heads in the network only when it sends a PCR packet to the requesting node. Since a cluster head have latest information on the resources available in the network, the wavelength channel selected by it is most likely to be available on the path. Thus, minimizing the occurrence of contention in the network. The propose scheme is an attempt to reduce the occurrence of contention in OBS networks. Contention may occur in the network when two nodes of different clusters request their cluster heads at the same time. In this case cluster heads may select the same wavelength channel and send PCR packet to the respective node. If the data burst from two nodes share the same link on their path to destination (which may or may not be same) then a contention occurs in the shared link. Such a scenario is depicted in Figure 4.6(b) and 4.6(c).

However, two nodes of same cluster requesting the cluster head at the same time will never give rise to contention. This is because, a cluster head selects a wavelength channel on the basis of information it have on the availability of resources. A cluster head selects a wavelength channel only when it found that the selection of wavelength channel at that instant will not result into contention.

Each cluster head maintains the status of resources available in the network. Initially all resources are available as seen by the cluster heads. When a cluster head send a PCR packet to a requesting node, it also informs other cluster heads in the network about the usage of the wavelength channel. A cluster head informs other cluster heads about the usage of wavelength channel by sending a RUP packet. On receiving RUP packet, a cluster head updates the status of wavelength channel as explained.

We explain below transmission of data burst in our proposed scheme. A node that has a data burst to transmit, first request its cluster head for an available wavelength channel on the path of data bursts destination. Node request its cluster head by sending a CRP packet. Cluster head processes the CRP packet and sends PCR or NCR packet to the requesting node depending upon the availability of wavelength channel. It senda a PCR packet if a wavelength channel is available on the data burst path else send a NCR packet. A source after receiving PCR packet from its cluster head transmits OBS control packet requesting the intermediate node on the path to destination to reserve the wavelength channel. Then, send the data burst on the selected wavelength channel after an offset time as determined by the OBS scheduling algorithm. On receiving NCR packet, the source node drop the data burst. Wavelength channel requested for reservation is a guess by the node, which is more likely to be available on the path to destination.

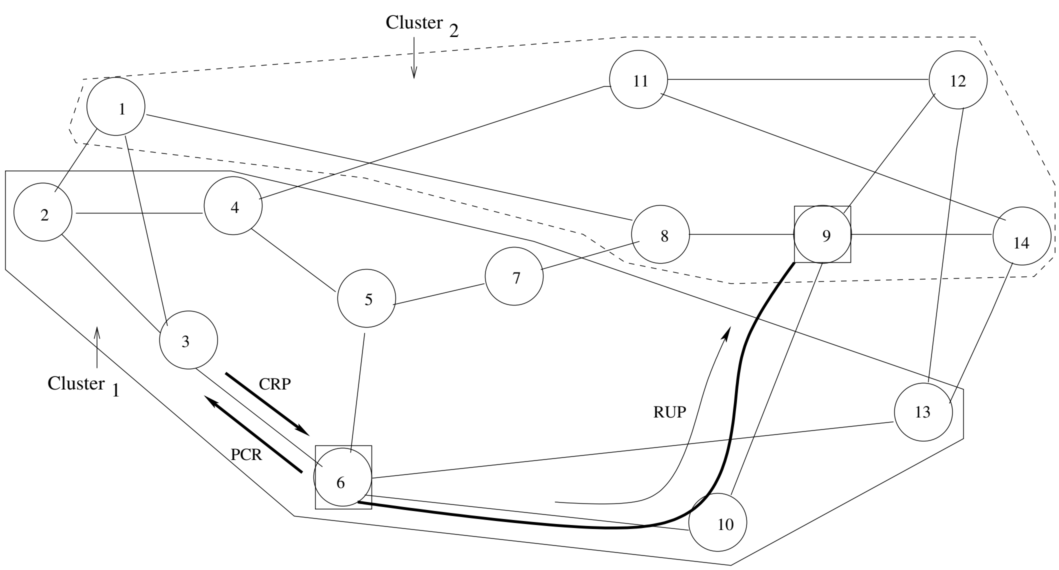

Figure 4.7: A *two* clustered NSFNET

We consider Figure 4.7, a *fourteen* node NSFNET partitioned into *two* clusters

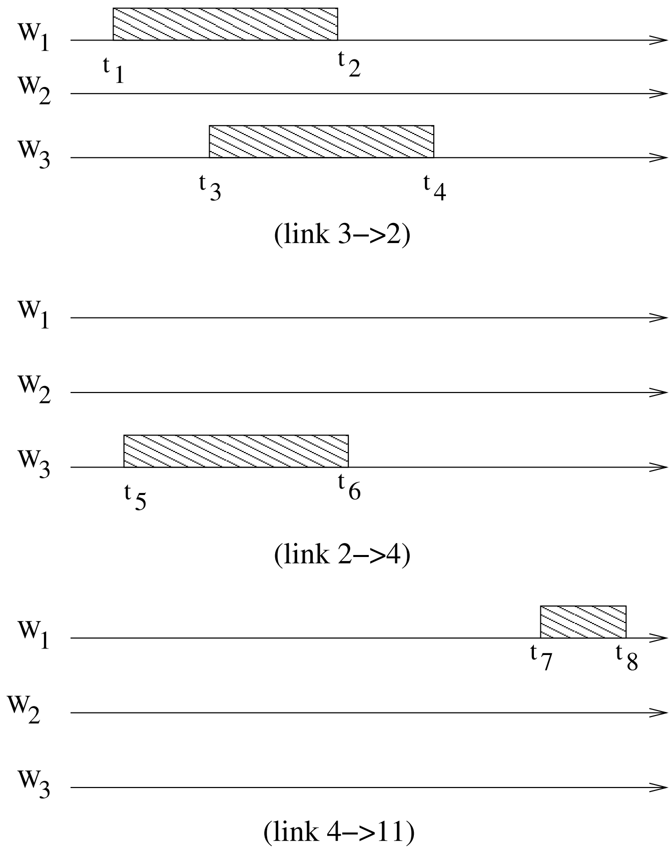

Figure 4.8: Status of wavelength channels as seen by cluster head node 6 on the links 3 → 2, 2 → 4 and 4 → 1

to illustrate communication between two nodes in our propose scheme. There exists a lightpath between the cluster head nodes 6 and 9 through node 10, shown in bold line. Suppose node 3 has a data burst destined to node 11. First, it sends a CRP packet to its cluster head node 6 which process it to find out an available wavelength channel on the path 3 → 2 → 4 → 11 from source 3 to destination 11. Cluster head selects a wavelength channel that is available on the links 3 → 2, 2 → 4 and 4 → 11. Suppose there are *three* wavelength channels (w_1, w_2 and w_3) on each links and the usage of wavelength channels on the links 3 → 2, 2 → 4 and 4 → 1 as seen by the cluster head node 6 is depicted in Figure 4.8. It is observed from Figure 4.8, wavelength channel w_2 is available in the link 3 → 2, w_1 and w_2 are available in the link 4 → 2, w_2 and w_3 are available in the link 4 → 11. A common wavelength channel w_2 is available on all the links 3 → 2, 2 → 4 and 4 → 11. So the cluster head selects the wavelength channel w_2 for this request and send a PCR packet to node 3 and RUP packet to cluster head node 9. Control packets PCR and RUP carry the information on wavelength channel w_2 to node 3 and node 9 respectively. Node 3 after receiving PCR packet, send a OBS control packet on the control channel to reserve wavelength channel w_2 on the path, followed by the data burst on the wavelength channel w_2 after an offset time determined by the OBS scheduling algorithm.

If no wavelength channel is available then the cluster head sends a NCR packet to the requesting source. In this scenario data burst is dropped at the source.

4.3 Simulation and Results

In this section we compare the performance of our propose scheme with deflection and no-deflection routing schemes [40]. For simulation we used obs-ns simulator that runs on the top of ns-2 simulator. For comparison we consider a *fourteen* node NSFNET and a *thirty three* node ARPANET as shown in Figure 4.3 and 4.5 respectively.

Parameters for simulation are given in Table 4.2. Traffic for simulation is generated using self-similar traffic [50].

Table 4.2: Simulation parameters

Parameter	**Value**
Maximum Burst Size	40 KB
Processing time of control packet	1.5 μs
Number of FDL	0
Bandwidth/channel	5 Gbit/s
Total No. of channels/link	8
No. of data channels/link	7
No. of control channels/link	1
Delay on fiber line	1 ms
Reservation Protocol	JET

We consider the following performance metric for comparison: (i) Burst loss ratio *vs.* Load (in number of burst sent) and (ii) End-to-end delay *vs.* Load (in number of burst sent).

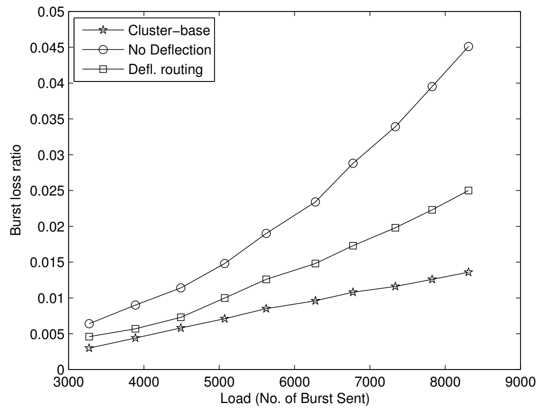

Figure 4.9: Burst loss ratio *vs.* Load in cluster-base, no-deflection and deflection routing for *eight* wavelength channel NSFNET.

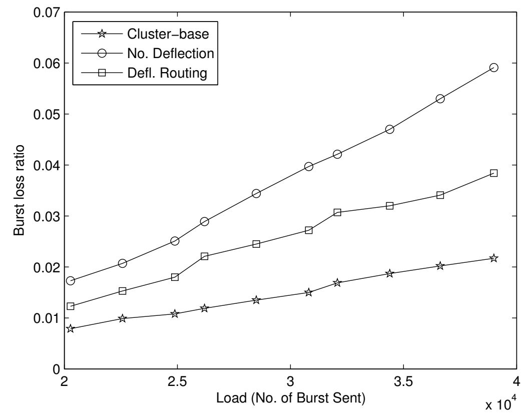

Figure 4.10: Burst loss ratio *vs.* Load in cluster-base, no-deflection and deflection routing for *eight* wavelength channel ARPANET.

Figure 4.9 and 4.10, shows the overall burst loss ratio *vs.* load in NSFNET and ARPANET respectively. Burst loss ratio is calculated as number of burst lost divided by number of burst sent. It is observed from Figure 4.9 and 4.10, burst loss ratio is higher in *no deflection* routing scheme and lower in our propose *cluster base* scheme. In *no deflection* routing contending data bursts are simply drop so burst loss ratio is higher. In *deflection* routing some of the contending data bursts follow an alternate route, so data bursts get a second chance to reach their destination. These deflected data bursts reach their destination depending on the network traffic. For lower network traffic, deflected data bursts get more chances to complete their journey. For this reason burst loss ratio is lower at low load in deflection routing. For higher network load, more deflected data burst gets blocked. Normal data bursts in the deflected route may be blocked by the deflected data bursts. This is the reason for increase burst loss ratio with load in deflection routing. Lower burst loss ratio in our proposed *cluster base* scheme is attributed to the selection of wavelength channel that is more likely to be available on the path to destination giving rise to lesser contention and lower burst loss.

Figure 4.11 and 4.12, shows the average end-to-end delay *vs.* load in NSFNET

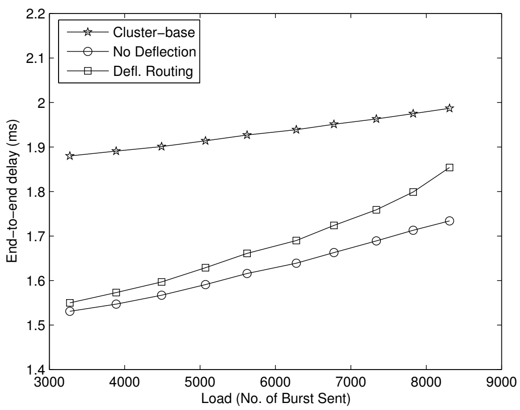

Figure 4.11: End-to-end delay *vs.* Load in cluster-base, no-deflection and deflection routing for *eight* wavelength channel NSFNET.

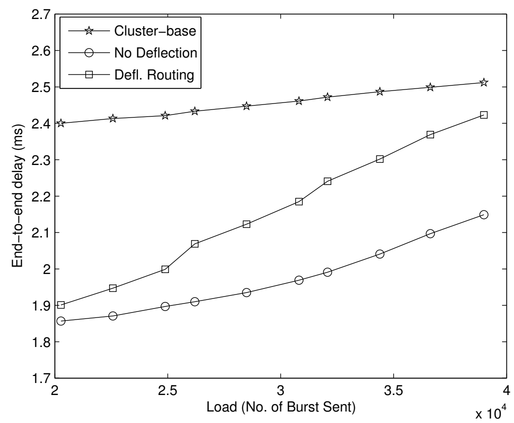

Figure 4.12: End-to-end delay *vs.* Load of cluster-base, no-deflection and deflection routing for *eight* wavelength channel ARPANET.

and ARPANET respectively. End-to-end delay is calculated as the total time taken by a successful data burst from source to destination. From Figure 4.11 and 4.12, it is observed that the end-to-end delay is higher in our *cluster base* scheme. This is because in the proposed scheme delay is calculated as the sum of the propagation delay between source to destination plus the round trip delay between a source and its cluster head. Higher delay in the proposed scheme is attributed to the addition of round trip delay between source and its cluster head. Though the delay is higher, increase in delay with load is only marginal. In deflection routing end-to-end delay is lower at low load but increases proportionately at higher load. This is because at higher load more data bursts follow the deflected route rather than normal route. As a result end-to-end delay increases with load in deflection routing. In no deflection routing end-to-end delay is lower, due to higher loss of the contending data bursts which do not contribute to delay calculation.

We varied the number of wavelength channel to *three* and *five* and show the effect on decreased number of wavelength channel in NSENET. The plot for burst loss ratio *vs.* load for *three* wavelength channel is shown in Figure 4.13 and for *five* wavelength channel in Figure 4.14.

Finally, the plot for burst loss ratio *vs.* load for *three, five* and *eight* number of wavelength channel in a fourteen node NSFNET is shown in Figure 4.15.

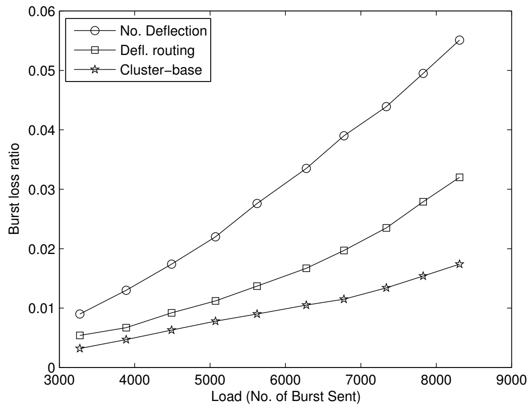

Figure 4.13: Burst loss ratio *vs.* Load in cluster-base, no-deflection and deflection routing for *five* wavelength channel NSFNET.

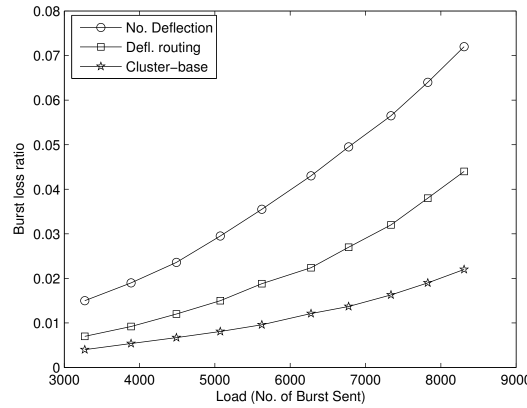

Figure 4.14: Burst loss ratio *vs.* Load of cluster-base, No Deflection and Deflection Routing for *three* wavelength channel NSFNET.

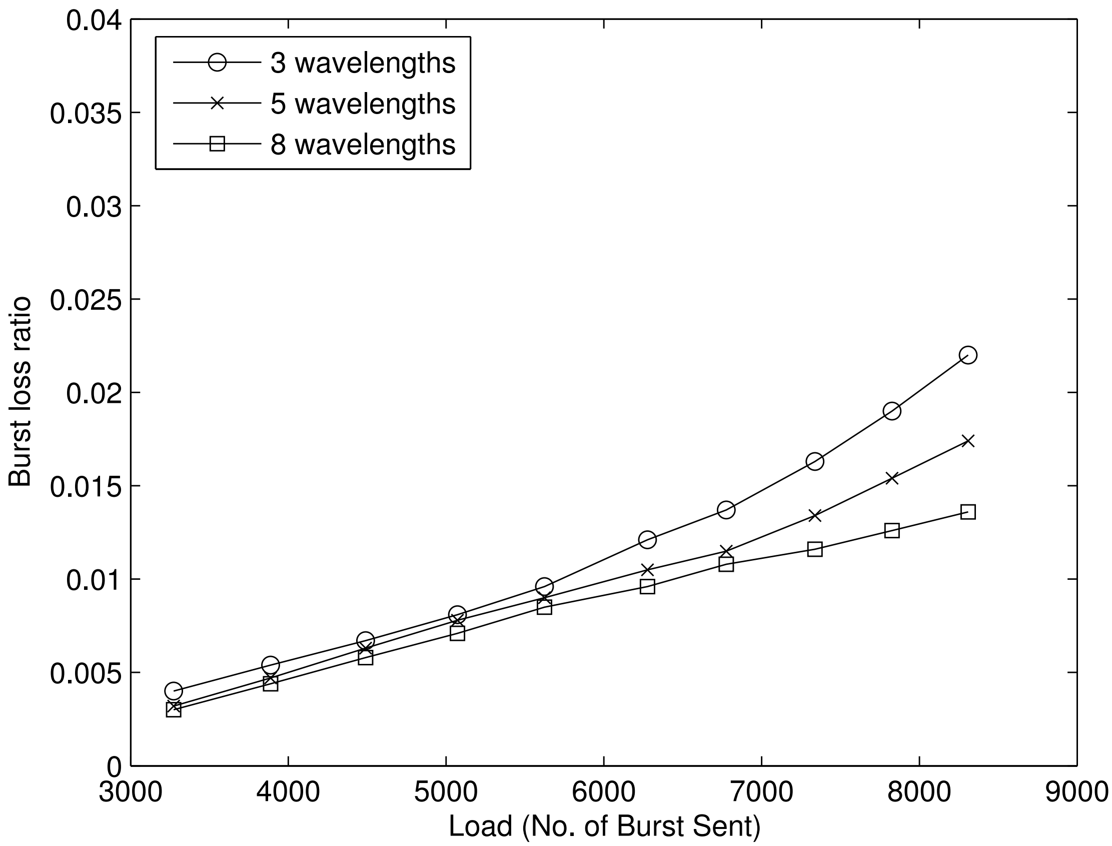

Figure 4.15: Burst loss ratio *vs.* Load for *three, five* and *eight* number of wavelengths in cluster-base routing of NSFNET.

4.4 Summary

In this chapter, we proposed a contention minimization scheme for OBS networks which partition a given network into number of sub networks called clusters. A node within a cluster is selected as cluster head which maintains the status of resources in the network. A node having data burst to transmit request its cluster head for a wavelength channel that is more likely to be available on the path to destination. Data burst is transmitted on this wavelength channel giving rise to lesser burst loss.

We compared our scheme with no deflection routing and deflection routing scheme. We found that burst loss ratio is higher in no deflection routing and lower in the proposed scheme. Lower burst loss in our proposed scheme is attributed to the selection of wavelength channel that is more likely to be available on the path to destination.

Lower burst loss in our proposed scheme comes with an additional delay. End-to-end delay in our proposed scheme is higher than deflection routing and no deflection routing schemes. Higher delay is due to the addition of round trip delay between a source and its cluster heads. However, delay increases only marginally with load in our proposed scheme.

Increase in the number cluster heads will lead to more signaling over head, and the increase in distance between a node and its cluster head adds to more end-end delay. We are investigating how to select a cluster head so that the end-end delay will be reduced without increase in burst loss ratio.

Chapter 5

Conclusions

Optical burst switching is the new switching paradigm for the next generation optical network. In OBS control packets are decoupled from the data packets and are sent in different channels. Control information is sent on control channel and data packets are sent on data channels. Some of the research issues identified in OBS network are - burst assembly and disassembly, burst scheduling, contention resolution and QoS. This thesis, focused on scheduling and contention in OBS networks. We have made a comparison between existing scheduling algorithms, propose a new scheduling algorithm and a technique to minimize contention in OBS networks.

In the reminder of this concluding chapter, we briefly summarize the original contributions of the study. Finally, some suggestions for future work are given.

5.1 Contributions

5.1.1 BFVF Algorithm

We proposed a channel scheduling algorithm called best fit void filling (BFVF). BFVF algorithm consider two parameters - (i) length of available void channel and (ii) length of incoming data burst to select an wavelength channel. Data burst is schedule on an wavelength channel, such that the void created will be minimal after scheduling of new data burst. If no void channel is available then BFVF uses LAUC algorithm.

We compared our proposed scheduling algorithm with LAUC-VF and Min-EV. It is observed that burst-loss ratio is lower and link utilization is higher in our proposed BFVF algorithm than LAUC-VF and Min-EV. Lower burst loss ratio in

BFVF algorithm is attributed to the selection of optimal void channel.

5.1.2 Cluster Based Contention Minimization

Next, we propose a contention minimization scheme called cluster-based contention minimization. This scheme tries to minimize the occurrences of contention in OBS network. In the proposed scheme a given network is logically divided into number of clusters (sub networks). A node within each cluster is selected as the cluster head, which maintains the status of resources available in the network. Cluster heads exchange the status of resources among themselves to maintain an up-to-date information about the network. A node within a cluster that wishes to transmit data, make request to its cluster head for an wavelength channel on the path to destination. If an wavelength channel is available on the path to the data burst destination, then the cluster head send a positive reply packet. Which contain the identity of the available wavelength channel. If no wavelength channel is available then send a negative reply. A node after receiving a positive reply from the cluster head send a OBS control packet followed by data burst on the selected wavelength channel. For a negative reply the data burst is dropped.

We compared our proposed scheme with no deflection routing and deflection routing scheme. A *fourteen* node NSFNET and *thirty three* node ARPANET is considered for simulation. We used obs-ns simulator for simulation.

It is observed that burst loss ratio is lower in our proposed scheme than no deflection routing and deflection routing scheme. This is due to Schelling a data burst in a wavelength channel that is more likely to be available on the path. In no deflection routing contending data bursts are dropped so burst loss ratio is higher. In deflection routing contending data bursts follow an alternative route so burst loss ratio is lower than no deflection routing.

The lower blocking in our proposed scheme comes with an additional delay. End-to-end delay in proposed scheme is higher than no deflection and deflection routing scheme, and depends on distance of a node from cluster head. Our proposed scheme experience an additional delay, that is equal to the round trip propagation delay between a node and it's cluster head besides the propagation delay between source and destination. However, the increase of end-to-end delay is marginal with load. In no deflection and deflection routing scheme the increase of end-to-end delay is significant with load.

Finally, we plot the graph for burst loss ratio *vs.* load in BFVF and cluster based algorithm in Figure 5.1.

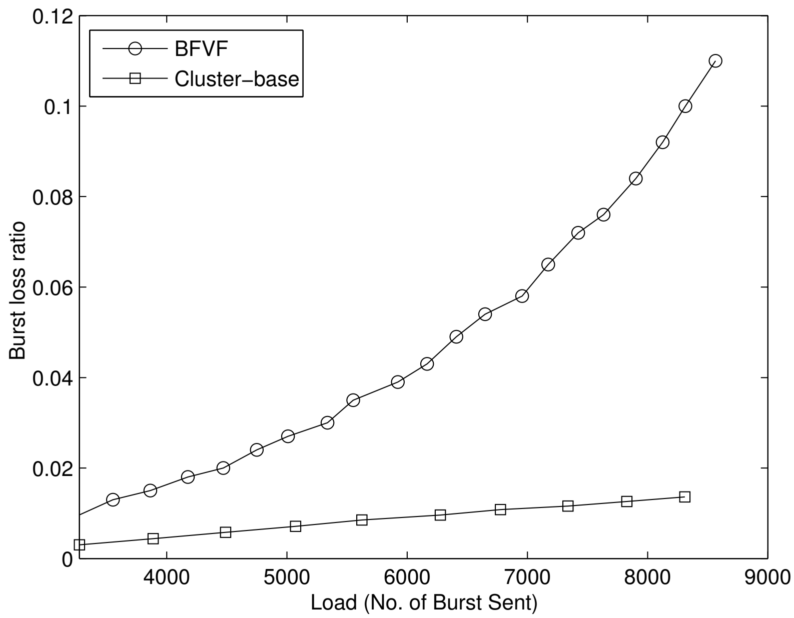

Figure 5.1: Burst loss ratio *vs.* Load in BFVF and Cluster-base algorithm for *eight*number of wavelength in *fourteen* node NSFNET.

5.2 Research Direction for Further Study

Area of future work includes a combination of existing contention resolution technique with our proposed scheme. If no wavelength channel is available on the shortest route then an alternative route can be used to send the data burst. Wavelength converters can be used to resolve contention.

Bibliography

[1] International Engineering Consortium. The Direction of Optical Networking Market. available at `http://www.iec.org/online/tutorials/`.

[2] A. K. Turuk and R. Kumar. A Novel Scheme to Reduce Burst-Loss and Provide QoS in Optical Burst Switching Network. In *Proceeding of HiPC-2004*, pages 19 – 22, 2004.

[3] Kishore Koduru. New Contention Resolution Techniques for Optical Burst Switching. Master's thesis, Louisiana State University, May 2005.

[4] S. Yoo, S. J. B. Yoo, and B. Mukherjee. All-Optical Packet Switching for Metropolitan Area Networks: Opportunities and Challenges. *IEEE Communications Magazine*, 39:142 – 148, 2001.

[5] Uyless D. Black. *Optical Networks: Third Generation Transport Systems*. Prentice Hall PTR, 2002.

[6] W. M. Golab and R. Boutaba. Resource Allocation in User-Controlled Circuit-Switched Optical Networks. *LNCS Spinger-Verlag*, 16(12):2081 – 2094, December 1998.

[7] R. Ramaswami and K.N. Sivarajan. Routing and Wavelength Assignment in All-Optical Networks. *IEEE/ACM Transactions on Networking*, 3(5):489 – 500, October 1995.

[8] M. J. O. Mahony, D. Simeonidou, D. Hunter, and A. Tzanakaki. The Application of Optical Packet Switching in Future Communication Networks. *IEEE Communications Magazine*, 39:128 – 135, March 2001.

[9] D. Blumenthal, P. Prucnal, and J. Sauer. Photonic Packet Switches Architectures and Experimental Implementations. In *Proceeding of IEEE*, pages 1650 – 1667, November 1994.

[10] S. Yao, B. Mukherjee, and S. Dixit. Advances in Photonic Packet Switching: An Overview. *IEEE Communications Magazine*, 38:84 – 94, February 2000.

[11] D.K.Hunter, M.C.Chia, and I.Andonovic. Buffering in Optical Packet Switches. *IEEE/OSA Journal of Lightwave Technology*, 16(12):2081 – 2094, December 1998.

[12] J. Turner. Terabit Burst Switching. *Journal of High Speed Networks*, 8:3 – 16, January 1999.

[13] M. Yoo and C. Qiao. A Novel Switching Paradigm for Buffer-Less WDM Networks. In *Optical Fiber Comm Conf (OFC), San Diego, CA*, pages 177 – 179, February 1999.

[14] M. Yoo and C. Qiao. Choices, Features and Issues in Optical Burst Switching (OBS). *Optical Networking Magazine*, 1(2):36 – 44, April 1999.

[15] S. Verma, H. Chaskar, and R. Ravikanth. Optical Burst Switching: A Viable Solution for Terabit IP Backbone. *IEEE Network*, pages 48 – 53, November/December 2000.

[16] Jing Teng and George N. Rouskas. A Comparison of the JIT, JET, and Horizon Wavelength Reservation Schemes on a Single OBS Node. In *Proceedings of the First Workshop on Optical Burst Switching*, October 2003.

[17] Vishwas S Puttasubbappa. *Optical Burst Switching: Challenges, Solutions and Performance Evaluation*. PhD thesis, North Carolina State University, 2005.

[18] C. Qiao and M. Yoo. Optical Burst Switching (OBS) - A New Paradigm for an Optical Internet. *Journal of High Speed Networks (JHSN)*, 8(1):69 – 84, 1999.

[19] Y. Chen, C. Qiao, and X. Yu. Optical Burst Switching: A New Area in Optical Networking Research. *IEEE Network Magazine*, 18(3):16 – 23, 2004.

[20] Bart Lannoo, Jan Cheyns, Erik Van Breusegem, Ann Ackaert, Mario Pickavet, and Piet Demeester. A Performance Study of Different OBS Scheduler Implementations. In *Proceeding of Symposium IEEE/LEOS Benelux Chapter, 2002, Amsterdam*, 2002.

[21] C. Siva Ram Murthy and Mohan Gurusamy. *WDM Optical Networks: Concepts, Design and Algorithms*. Prentice Hall PTR, November 2001.

[22] M. Yoo and C. Qiao. A New Optical Burst Switching Protocol for Supporting Quality of Service. In *Proceeding of SPIE, All Optical Networking: Architecture,Control, Network Issues*, pages 396 – 405, November 1998.

[23] Ping DU. *QoS Control and Performance Improvement Methods for Optical Burst Switching Networks*. PhD thesis, Department of Informatics, School of Multidisciplinary Sciences, The Graduate University for Advanced Studies (SOKENDAI), March 2007.

[24] V. Vokkarane, Q. Zhang, J.P. Jue, and B. Chen. Generalized Burst Assembly and Scheduling Techniques for QoS Support to Optical Burst-Switched Networks. *In Proceedings of GLOBECOM*, pages 2747 – 2751, November 2002.

[25] V. Vokkarane, J. P. Jue, and S. Sitaraman. Burst Segmentation: An Approach for Reducing Packet Loss in Optical Burst Switched Networks. In *Proceeding of IEEE ICC 2002*, pages 2673 – 2677, 2002.

[26] J. Y. Wei and R. I. McFarland. Just-In-Time Signaling for WDM Optical Burst Switching Networks. *Journal of Lightwave Technology*, 18(12):2019 – 2037, December 2000.

[27] M. Yoo and C. Qiao. Just-Enough-Time (JET): A High Speed Protocol for Bursty Traffic in Optical Networks. In *Proceeding of IEEE/LEOS*, pages 26 – 27, August 1997.

[28] G. C. Hudek and D. J. Muder. Signalling Analysis for A Multi-Switch All-Optical Network. In *Proceeding of IEEE International Conference on Communications, ICC 95*, pages 1206 – 1210, 1995.

[29] D. K. Hunter, W. D. Cornwell, T. H. Gilfedder, A. Franzen, and I. Andonovic. Slob: A Switch with Large Optical Buffers for Packet Switching. *IEEE/OSA Journal of Lightwave Technology*, 16(10):1725 – 1736, October 1998.

[30] K. Dozer, C. Gauger, J .Spath, and S. Bodamer. Evaluation of Reservation Mechanisms for Optical Burst Switching. *AEU International Journal of Electonics and Communications*, 55(1):2017 – 2022, January 2001.

[31] Marije Ljolje, Robert Inkret, and Branko Mikac. A Comparative Analysis of Data Scheduling Algorithms in Optical Burst Switching Networks. In *Proceeding of Optical Network Design and Modeling,2005*, pages 493 – 500, 2005.

[32] Yijun Xiong, Marc Vandenhoute, and Hakki C. Cankaya. Control Architecture in Optical Burst Switched WDM Networks. *IEEE JSAC*, 18(10):1838 – 1851, October 2000.

[33] M. Yoo, C. Qiao, and S. Dixit. QoS Performance of Optical Burst Switching in IP-Over-WDM Networks. *IEEE Journal on Selected Areas in Communications*, 18(10):2062 – 2071, October 2000.

[34] M. Yang, S. Q. Zheng, and D. Verchere. A QoS Supporting Scheduling Algorithm for Optical Burst Switching DWDM Networks. In *Proceeding of GLOBECOM 01*, pages 86 – 91, 2001.

[35] Jinhui Xu, Chunming Qiao, Jikai Li, and Guang Xu. Efficient Channel Scheduling Algorithms in Optical Burst Switching Networks. In *Proceeding of IEEE INFOCOM, 2003*, volume 3, pages 2268 – 2278, 2003.

[36] Vinod M. Vokkarane and Jason P. Jue. Segmentation-Based Nonpreemptive Channel Scheduling Algorithms for Optical Burst-Switched Networks. *Journal of Lightwave Technology*, 23(10):3125 – 3137, October 2005.

[37] B. Mukerjee. *Optical Communication Networks*. McGraw-Hill, 1997.

[38] S. Lee, K. Sriram, H. Kim, and J. Song. Contention-based Limited Deflection Routing in OBS Networks. In *Proceeding of IEEE Globecom 2003*, pages 2633 – 2637, 2003.

[39] B. Ramamurthy and B. Mukherjee. Wavelength Conversion in WDM Networking. *IEEE Journal on Selected Areas in Communications*, 16(7):1061 – 1073, September 1998.

[40] A. Zalesky, H. L. Vu, M. Zukerman, Z. Rosberg, and E. W. M. Wong. Evaluation of Limited Wavelength Conversion and Deflection Routing as Methods to Reduce Blocking Probability in Optical Burst Switched Networks. In *Proceedings of IEEE ICC 2004*, 2004.

[41] A. Bononi, G. A. Castanon, and O. K. Tonguz. Analysis of Hot-Potato Optical Networks with Wavelength Conversion. *IEEE Journal of Lightwave Technology*, 17(4):525 – 534, April 1999.

[42] A. Zalesky, H. L. Vu, Z. Rosberg, E. W. M. Wong, and M. Zukerman. OBS Contention Resolution Performance. *Performance Evaluation*, 64(4):357 – 373, 2007.

[43] R. Ramaswami and K. N. Sivarajan. *Optical Networks*. Morgan Kaufman, 2002.

[44] F. Farahmand, Q. Zhand, and J. P. Jue. A Feedback-Based Contention Avoidance Mechanism for Optical Burst Switching Networks. In *Proceeding of BROADNETS 2004*, 2004.

[45] Sungchang Kim, Namook Kim, and Minho Kang. Contention Resolution for Optical Burst Switching Networks Using Alternative Routing. In *Proceeding of IEEE, ICC 2002*, pages 1033 – 1038, April-May 2002.

[46] Ching-Fang Hsu, Te-Lung Liu, and Nen-Fu Huang. Performance Analysis of Deflection Routing in Optical Burst-Switched Networks. In *Proceeding of IEEE, INFOCOM 2002, New York, NY*, pages 1011 – 1016, June 2002.

[47] ns2 Simulator. http://www.isi.edu/nsnam/ns.

[48] obs-ns Simulator. http://www.wine.icu.ac.kr/ obsns/.

[49] D. Kratochvil and D. Sood. United State Computer Research Networking: Domestic and International Telecommunications Capacity Requirements. Technical report, NASA Lewis Research Center, December 1990.

[50] Will E. Leland, Murad S. Taqqu, Walter Willinger, and Deniel V. Wilson. On the Self-Similar Nature of Ethernet Traffic (Extended Version). *ACM Transactions on Networking*, 2(1):1 – 15, February 1994.

[51] R. Ramaswami and K. N. Sivarajan. *Optical Networks: A Practical Perspective.* Academic Press, 2001.

Dissemination of Work

Published

1. Mrinal Nandi and Ashok K. Turuk, **A Novel Channel Scheduling Algorithm in Optical Burst Switching Network**, *International Conference on Sensors, Signal Processing, Communication, Control and Instrumentation (SSPCCIN-2008), pp.610 - 614, 5 - 8 January, 2008, Pune, India.*

2. Mrinal Nandi and Ashok K. Turuk, **A Comparative Study of Different Optical Burst Switching Scheduling Algorithms Under Self-similar Traffic**, *2nd National Conference on. Recent Trends in Information Systems(ReTIS-08), pp.196 - 199, 7 - 9 February, 2008, Jadavpur, Kolkata, India.*

3. Mrinal Nandi and Ashok K. Turuk, **Analysis of Wavelength Conversion and Deflection Routing in Optical Burst Switched Network**, *1st International Conference Advance Computing (ICAC 2008), 21 - 23 February, 2008, Chikli, India.*

4. Mrinal Nandi, Ashok K. Turuk and B. D. Sahoo, **A New Contention Avoidance Scheme in Optical Burst Switch Network**, *16th IEEE International Conference on Networks (ICON 2008), 12 - 14 December, 2008, New Delhi, India.*

Communicated

1. Mrinal Nandi, Ashok K. Turuk and Bibhudatta Sahoo, **A Technique to Minimize Contention in Optical Burst Switching Networks**, *Communicated to Optics Communication - Elsevier.*

2. Mrinal Nandi, Ashok K. Turuk and Bibhudatta Sahoo, **Best Fit Void Filling Scheduling Algorithm in Optical Burst Switching Networks**, *Communicated to International Journal on Wireless and Optical Communication.*

3. Mrinal Nandi, Ashok K. Turuk and Bibhudatta Sahoo, **Cluster-Based Contention Minimization in Optical Burst Switching Networks**, *Communicated to Photonic Network Communications.*