A NEW FEEDBACK-BASED CONTENTION AVOIDANCE ALGORITHM FOR OPTICAL BURST SWITCHING NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HADİ ALPER TOKU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

DECEMBER 2008

Approval of the thesis:

A NEW FEEDBACK-BASED CONTENTION AVOIDANCE ALGORITHM FOR OPTICAL BURST SWITCHING NETWORKS

submitted by HADİ ALPER TOKU in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İsmet Erkmen _________
Head of Department, **Electrical and Electronics Engineering**

Assist. Prof. Dr. Şenan Ece Güran Schmidt _________
Supervisor, **Electrical and Electronics Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Semih Bilgen _____________
Electrical and Electronics Engineering Dept., METU

Assist. Prof. Dr. Şenan Ece Güran Schmidt _____________
Electrical and Electronics Engineering Dept., METU

Assist. Prof. Dr. Cüneyt Bazlamaçcı _____________
Electrical and Electronics Engineering Dept., METU

Assist. Prof. Dr. İlkay Ulusoy _____________
Electrical and Electronics Engineering Dept., METU

Emin İlker Çetinbaş _____________
Test Engineering Dept., ASELSAN

Date: _____________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Hadi Alper TOKU

Signature :

ABSTRACT

A NEW FEEDBACK-BASED CONTENTION AVOIDANCE ALGORITHM FOR OPTICAL BURST SWITCHING NETWORKS

TOKU, Hadi Alper
M.Sc., Department of Electrical and Electronics Engineering
Supervisor: Asst. Prof. Dr. Şenan Ece Güran SCHMIDT

December 2008, 90 pages

In this thesis, a feedback-based contention avoidance technique based on weighted Dijkstra algorithm is proposed to address the contention avoidance problem for Optical Burst Switching networks.

Optical Burst Switching (OBS) has been proposed as a promising technique to support high-bandwidth, bursty data traffic in the next-generation optical Internet. Nevertheless, there are still some challenging issues that need to be solved to achieve an effective implementation of OBS. Contention problem occurs when two or more bursts are destined for the same wavelength. To solve this problem, various reactive contention resolution methods have been proposed in the literature. However, many of them are very vulnerable to network load and may suffer severe loss in case of heavy traffic. By proactively controlling the overall traffic, network is able to update itself in case of high congestion and by means of this method; contention avoidance can be achieved efficiently.

The performance analysis of the proposed algorithm is presented through network simulation results provided by OMNET++ simulation environment. The simulation results show that the proposed contention avoidance technique significantly reduces the burst loss probability as compared to networks without any contention avoidance techniques.

Keywords: Optical Burst Switching, Feedback Control, Contention Avoidance, Dynamic Routing

ÖZ

OPTİK ÇOĞUŞMA ANAHTARLAMALI AĞLAR İÇİN GERİBESLEME TABANLI YENİ BİR ÇEKİŞME ÖNLEME ALGORİTMASI

TOKU, Hadi Alper
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Yrd. Doç. Dr. Şenan Ece Güran SCHMIDT

Aralık 2008, 90 sayfa

Bu tezde optik çoğuşma anahtarlamalı ağlardaki çekişme önleme sorununa yönelik ağırlıklı Dijkstra algoritması temelli geribesleme tabanlı bir çekişme önleme yöntemi önerilmiştir.

Optik Çoğuşma Anahtarlama, yeni nesil optik internetin yüksek bant genişliği ve çoğuşmalı veri trafiğini desteklemesi için gelecek vaat eden bir yöntem olarak önerilmiştir. Bununla beraber Optik Çoğuşma Anahtarlama yönteminin etkili bir şekilde uygulamaya geçirilmesini sağlamak için çözülmeyi bekleyen çeşitli zorlu sorunlar bulunmaktadır. Optik çoğuşma anahtarlamalı ağlarda 2 veya daha fazla sayıdaki çoğuşma, aynı dalgaboyuna yönlendiklerinde çekişme problemi ortaya çıkmaktadır. Literatürde çekişme çözümüne yönelik çok çeşitli tepkisel yöntemler önerilmiştir. Ancak bunların çoğu ağ yüküne karşı savunmasız kalmakta ve yoğun trafik durumlarında şiddetli kayıplarla karşılaşmaktadırlar. Yoğun trafik durumunda ağ, genel trafiği inisiyatifi ele alacak şekilde kontrol ederek kendini güncelleme yeteneğine sahip olabilir ve bu yöntemle çekişme önleme verimli bir şekilde sağlanabilir.

Önerilen algoritmanın performans analizi, OMNET++ benzetim ortamı tarafından sağlanan ağ benzetim sonuçlarından yararlanılarak yapılmıştır. Benzetim sonuçları önerilen çekişme önleme yönteminin çekişme önleme yöntemi kullanılmamış ağlara oranla paket kayıp oranını önemli ölçüde düşürdüğünü göstermiştir.

Anahtar Kelimeler: Optik Çoğuşma Anahtarlama, Geribeslemeli Denetim, Çekişme Önleme, Dinamik Yol Atama

To My Dear Family & Lale

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor Asst. Prof. Dr. Şenan Ece Schmidt for her valuable guidance, criticism and support during this thesis study and my whole graduate life.

I would like to thank to my colleagues at ASELSAN Inc. for their support and valuable contribution to this thesis work. I am grateful to ASELSAN Inc. for providing tools and other facilities for the completion of this thesis work.

I would also like to thank my parents, Sermin and Bilgin TOKU and my brother, Berkay TOKU for giving me encouragement during this thesis and all kind of supports during my whole education.

Finally, I have very special thanks for my dear Lale, for her endless love, encouragement and patience during this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACK	Acknowledgement
AON	All Optical Network
ATM	Asynchronous Transfer Mode
AWG	Arrayed Waveguide Gratings
BER	Bit Error Rate
BHP	Burst Header Packet
BN	Basic Network
CCG	Control Channel Group
COST	European Cooperation in the field of Scientific and Technical Research
DCG	Data Channel Group
DFB	Distributed Feedback
DWDM	Dense Wavelength Division Multiplexing
EDFA	Erbium-doped Fiber Amplifier
FDL	Fiber Delay Lines
FDM	Frequency Division Multiplexing
FFUC	First Fit Unscheduled Channel
FFUC-VF	First Fit Unscheduled Channel with Void Filling
HDTV	High Definition Television
IP	Internet Protocol
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
JET	Just-Enough-Time
JIT	Just-In-Time
LAUC	Latest Available Unscheduled Channel

LAUC-VF	Latest Available Unscheduled Channel with Void Filling
LAUT	Latest Available Unscheduled Time
LED	Light Emitting Diode
Min-SV	Minimum Starting Void
MSM	Metal-Semiconductor-Metal
NED	Network Description
nfp	Negative Feedback Packet
NSF	National Science Foundation
NSFNET	National Science Foundation Network
OADM	Optical Add-Drop Multiplexer
OBS	Optical Burst Switching
OCS	Optical Circuit Switching
OE	Optoelectronic
O/E/O	Optical/Electronical/Optical
OPS	Optical Packet Switching
OXC	Optical Cross Connect
QoS	Quality of Service
RAM	Random Access Memory
SCU	Switch Control Unit
ST	Switching Time
TAG	Tell and Go
TAW	Tell and Wait
TCP	Transmission Control Protocol
US	United States
UTP	Unshielded Twisted Pair
WDM	Wavelength Division Multiplexing

CHAPTER 1

INTRODUCTION

In Optical Communication Systems, due to the high bandwidth capacity of the fiber optic links and recent improvements on dense wavelength division multiplexing (DWDM) technology, raw bandwidth available on fiber optic links has increased significantly. However, this high capacity cannot be fully utilized because of the inadequate optical network architectures. Optical Burst Switching (OBS) [1]-[2] has been proposed as a new paradigm to provide the flexible and dynamic bandwidth allocation required to support bursty traffic.

In OBS Networks, incoming data packets are assembled into basic units, called data bursts. Each data burst has its own control packet, that carries the length, the destination address, and the QoS requirements of this data burst. In OBS networks that adopt Just Enough Time (JET) reservation model [1], control packets carry extra offset time information, which informs the traversed OBS node about the expected arrival time of the corresponding data burst. When the control packet visits an OBS node, the burst length and the arrival time are extracted from the control packet and the data burst is scheduled in advance to an available outgoing wavelength.

A major concern in OBS networks is high contention and burst loss due to output data channel contention, which occurs when multiple bursts contend for the same outgoing wavelength at the same time. Contention is inevitable for an OBS network, which assumes that there is no optical buffer in the core nodes. The contention can affect the network performance dramatically in terms of loss ratio and delivery rate. Each discarded burst causes a wasted bandwidth and decreased throughput. Contention is worsened when the traffic becomes bursty and when the data burst duration varies and becomes longer.

There are several proposed solutions about contention resolution issue in the literature. Most solutions up to present can be classified as reactive and proactive approaches. Reactive approaches are provoked after contention occurs. Most promising reactive approaches are deflection routing [3], buffering by using fiber delay lines [4], wavelength conversion [5], segmentation [6], look-ahead contention resolution [7] and shortest drop policy [8]. All of these policies attempt to improve the contention resolution in OBS networks. However, they cannot resolve the problem efficiently because of their reactive approach. Wavelength conversion is a good method with optical buffering technique, however buffering technology in optical routers is not mature at present. Using fiber delay lines increases data latency and implementation complexity and is not commercially viable. Segmentation also increases complexity of control. Deflection routing increases burst latency and a burst can loop in the network and waste resources.

There are alternative proactive approaches to reduce network congestion, which are controlling the rate of traffic injection into the network [9] or changing the route of the burst [10] so that congestion does not arise. In [11], the core nodes broadcast the burst loss information to all edge nodes so that they can adjust their burst injection rates to control the network load. In [12], a feedback-based OBS network is proposed by using explicit feedback signaling to each source, thus the data burst flow rate going to congested links is controlled. These approaches may reduce the burst loss probability, but they are very vulnerable to network load and may suffer from high data losses in heavy traffic load situations.

In this thesis, congestion avoidance method, namely proactively controlling the network traffic and rerouting of the data bursts is adopted as a first priority to reduce congestion and the data burst losses. In order to achieve the control over the network, feedback based approach is preferred. Along with data burst and control packets, a third type of packet, namely, “negative feedback packet” is introduced. Negative feedback packets are formed and sent from the core nodes to the source nodes in the reverse direction when the control packet is unable to schedule an available time for the incoming data burst on an output wavelength of the core node due to congestion. Negative feedback packets have the information of the congested link and the number of forwarded and discarded packets on every output link of the visited nodes. According to the information coming from negative feedback packets, core nodes calculate the load density of every possible core node and they will introduce new routing paths to the incoming bursts in order to avoid congestion. In the rerouting process of the burst, the core node evaluates the possible congested links and chooses the most appropriate path that new bursts will follow using weighted Dijkstra algorithm. In the simulation stage of this proposed algorithm, OMNET++ simulation environment is used.

An OBS simulation environment using JET signaling technique is designed and created for this thesis. The performance and results of the simulation environment is evaluated in comparison to the results of the simulations of a previously published OBS technology study. It is concluded that simulation results are nearly matched, revealing similar burst loss probabilities. The proposed congestion avoidance algorithm is implemented on the simulation environment and results are evaluated due to network delay and burst loss probability metrics. We observed that the ratio of the discarded packets on congested links is decreased by 10-15 percent with the introduction of our approach. While achieving this enhancement, delay parameters like propagation delay, offset time and end-to-end delays are affected on a limited scale. In addition to this, several simulations are implemented with various data burst sizes and link transmission rates and similar results are obtained.

This thesis is organized as follows:

The second chapter first gives a brief introduction to optical communication systems and WDM technology. In the next chapter, OBS network architecture and the functional components needed for OBS technology are explained. The fourth chapter presents the new feedback based contention avoidance technique based on weighted Dijkstra algorithm proposed in this thesis and its performance evaluation. Last chapter concludes the thesis and outlines future work.

CHAPTER 2

OPTICAL COMMUNICATION SYSTEMS AND WDM

In this chapter, firstly the optical communication system components and superiorities of this technology over existing electronic communication systems are presented. Then the chapter continues with the main multiplexing technique, which makes this technology feasible among all other existing communication solutions: Wavelength Division Multiplexing.

Following the brief information about the WDM technology, the chapter is completed with the introduction of three main switching solutions in Optical Communication Systems, which constitute our primary interest in this thesis.

2.1 OPTICAL COMMUNICATION SYSTEMS

Optical Communication is any form of telecommunication that uses light as the transmission medium. Optical fiber is the most common type of channel for optical communications. Modern fiber-optic communication systems need an optical transmitter to convert an electrical signal into an optical signal to send into the optical fiber, a cable containing bundles of multiple optical fibers that is routed through underground conduits and buildings, various kinds of amplifiers, and an optical receiver to recover the signal as an electrical signal. The information transmitted is typically digital information generated by computers, telephone systems and cable television companies [13].

2.1.1 SYSTEM COMPONENTS

2.1.1.1 OPTICAL TRANSMITTERS

The transmitters in optical communication systems are mostly light-emitting diodes (LEDs) or laser diodes. These two types of diode have very much in common although LEDs produce incoherent light, while laser diodes produce coherent light. Infrared light, rather than visible light is used more commonly in optical systems because optical fibers can transmit infrared wavelengths with less attenuation and dispersion.

For use in optical communications, optical transmitters must be designed to be reliable, compact, efficient and directly modulated at high frequencies [14].

2.1.1.2 OPTICAL FIBERS

Optical fiber consists of a core, cladding, and a protective outer coating, which guides light along the core by total internal reflection (Figure 2-1). The core and the cladding parts are mainly made of high-quality silica glass or plastic. An optical fiber is very susceptible of any kind of pressure and it can be broken easily if bent too sharply. Special skills and advanced interconnection technology is needed to align the fiber cores and connect two optical fibers. Microscopic precision can be provided by the two basic methods: Fusion splicing or mechanical splicing,

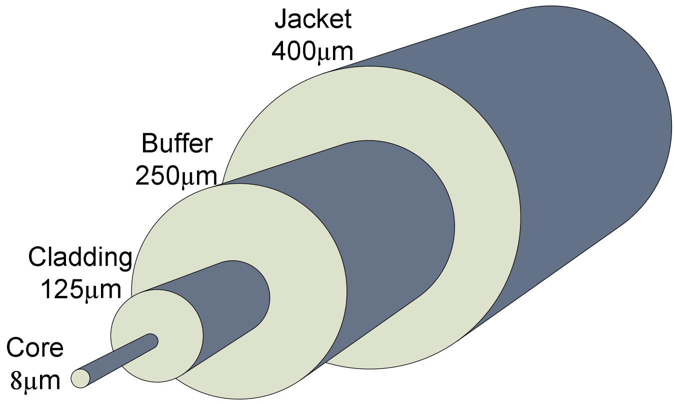


Figure 2-1 A typical single-mode optical fiber

Single-mode and multi-mode optical fibers are the two main categories of optical fiber used in fiber optic communications. Multi-mode fiber has a larger core (≥ 50 micrometers) and it allows less precise, cheaper transmitters and receivers to connect to it. But, multi-mode fiber usage causes multimode distortion limiting the bandwidth and length of the link. In addition to this, multi-mode fiber is usually more expensive due to its higher dopant content and exhibits higher attenuation. Single-mode fiber's smaller core (<10 micrometers) offers higher bandwidth capabilities and more efficient links but its equipment is more expensive than the equipment of multi-mode fiber. This again is no big drawback for single-mode fibers though the single mode fiber itself is usually cheaper in bulk [15].

2.1.1.3 OPTICAL AMPLIFIERS

The transmission distance of a fiber-optic communication system is limited dramatically because of the fiber attenuation and fiber distortion in the transmission links. Firstly, opto-electronic repeaters are developed in order to extend this distance to higher values. These repeaters first convert the signal to an electrical signal then use a transmitter to send the signal again at a higher intensity. Because of their high complexity and the need for the installation about every 20 km, the cost for these repeaters was very high.

As an alternative to opto-electronic repeaters, optical amplifying technology is investigated and after doping a length of fiber with the rare-earth mineral erbium, and pumping it with light from a laser with a shorter wavelength than the communications signal (typically 980 nm), optical amplifiers are introduced to the optical communications area. These devices amplify an optical signal directly, without the need to first convert it to an electrical signal. Because of their simplicity and ease of use, they become widespread in newly installed optical communication systems and replaced the opto-electronic repeaters [13].

2.1.1.4 OPTICAL RECEIVERS

Optical receivers perform simply the opposite task of what optical transmitters do while optical signaling is carried on. An optical receiver extracts the information that has been placed on the modulated light carrier and restores it to its original electrical form. This task is mainly performed by a photo detector which is the main component of an optical receiver. A photo detector converts light into electricity through the photoelectric effect and it is typically a semiconductor-based photodiode, such as a p-i-n photodiode, a p-n photodiode, or an avalanche photodiode. Metal-semiconductor-metal (MSM) photo detectors are also used but they are not the design of choice for high-quality communication systems. However, their ease of fabrication and integration with other components makes them desirable for some low-cost applications [13].

2.1.2 COMPARISON WITH OTHER EXISTING TECHNOLOGIES

Explosive growth of the Internet traffic makes all-optical networks a promising solution. That is not only because of the huge bandwidth that optical networks promise but also because of the several characteristics of the optical fibers which makes it the sweetheart of present and future networking:

1. Bit Error Rate (BER) of a fiber is very low
2. The material used for currently commercialized fibers (core and cladding) include pure glass (SiO2) which is made of sand. Hence manufacturing fibers is cheap.
3. Due to its low attenuation characteristic, repeaters are rarely needed.
4. Fibers are immune to power surges, electromagnetic interference or power failures
5. Optical fiber cable installation is simple and much easier than installing coaxial or UTP copper cable because they are thin and lightweight.
6. In computer networks, optical fibers are commonly used because they are very suitable for carrying digital information.
7. Optical fibers are very flexible and can transmit and receive light. Hence, they are used in many flexible digital cameras for the medical imaging, mechanical imaging and plumbing purposes.

Despite of all these eye-catching superiorities of fibers over copper wires, fiber-optic communication has some limitations on the application. Infrastructure development within cities is relatively difficult and time-consuming, and fiber-optic systems were complex and expensive to install and operate. Due to these difficulties, fiber-optic communication systems have primarily been installed in long-distance applications, where they can be used to their full transmission capacity, offsetting the increased cost [14].

2.2 WAVELENGTH DIVISION MULTIPLEXING

2.2.1 WDM TECHNOLOGY

Wavelength Division Multiplexing technology enables different optical signals to be transmitted by a single fiber. Main principle of WDM is very similar to the principle of frequency division multiplexing (FDM). Several signals can be transmitted using different carriers and occupying non-overlapping parts of a frequency spectrum. In WDM case, the spectrum band used is in the region where optical fibers have very low signal loss. (1300 or 1500 nm)

Years ago, each window was used to transmit a single digital signal. With the latest technological improvements in optical networks, such as erbium-doped fiber amplifiers (EDFAs), distributed feedback (DFB) lasers, and photodetectors, it was quickly discovered that each transmitting window could be used by many optical signals. Each of these optical signals occupies a small portion of the total wavelength window available. In reality, only the precision of these components can limit the number of optical signals multiplexed within a window. With current technology, over 100 optical channels can be multiplexed into a single fiber. This technology is called dense WDM (DWDM).

DWDM’s most outstanding characteristic is its potential to increase the optical fiber bandwidth cost effectively. The large network of fibers that is used around the world has suddenly their capacity multiplied, without any need to lay new fibers, which is a high-priced process. Although, new DWDM equipment must be connected to these fibers and optical regenerators might be needed, this technology got a warm welcome from the researchers and of course users all over the world.

ITU-T is responsible for the standardization procedure of the number and the frequency of wavelengths to be used. For interoperability and avoidance of the destructive interference between optical signals, this assignment process of wavelength is quite important matter for the technology [16].

2.2.2 WDM COMPONENTS

In a simple WDM system (Figure 2-2), the main components that form a simple WDM system are:

1. Transmitter (DFB lasers),
2. Multiplexer or Combiner,
3. High bandwidth optical fiber,
4. Optical Switch,
5. Wavelength converter,
6. Demultiplexer or Splitter,
7. Tunable optical fiber.

DFB lasers are used as transmitters, one for each wavelength. Then the emitted light signals are multiplexed together onto a single optical fiber by the multiplexer or combiner. The combined optical signals have to demultiplexed at the receiving end after they are transmitted through a high-bandwidth optical fiber. This demultiplexing process is achieved by distributing the total optical power to each output port and then requiring that each receiver recover only the desired wavelength by using a tunable optical filter.

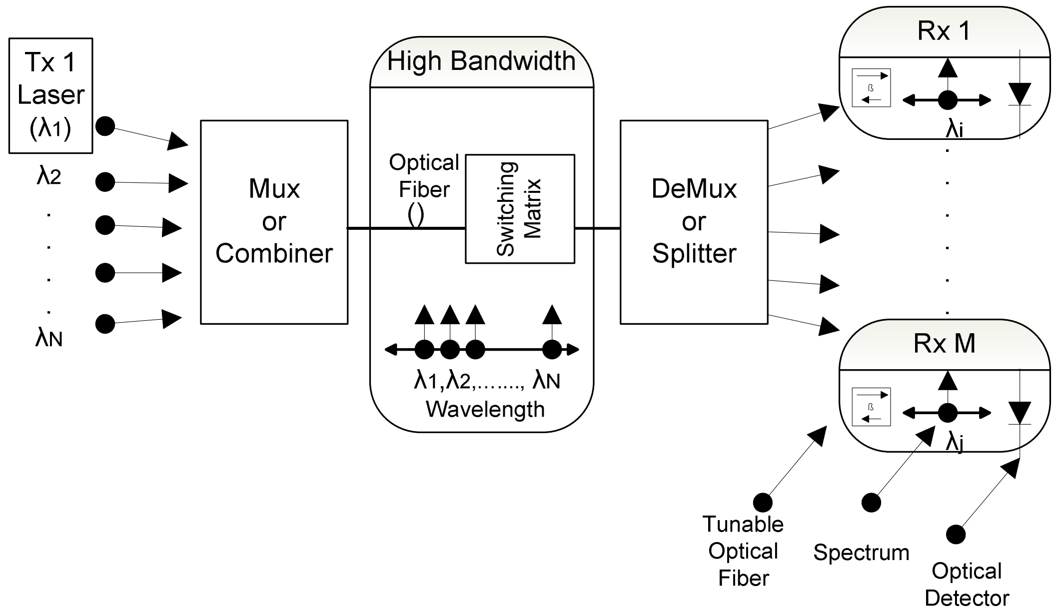


Figure 2-2 Diagram of a simple WDM system [22]

Multiplexers and demultiplexers in the system are named as optical ADMs (Add-Drop Multiplexers) (OADMs). These optical devices are very similar to digital ADMs but grooming and splitting optical signals along the transmission path is the distinctive property of these devices. OADMs are usually made of arrayed waveguide gratings (AWG). Fiber Bragg gratings are also used in the OADMs.

Optical switch is one of the key components of WDM technologies. It is capable of switching optical signals from a given input port to a given output port and the equivalent of an electronic crossbar. Optical switches play an important role in optical networks because a given optical signal can be routed toward its appropriate destination due its existence.

Another important optical component is the wavelength converter. A wavelength converter is a device that converts an optical signal coming at a given wavelength into another signal on a different wavelength, without ever changing the digital information on the signal. This capability of the wavelength converters provides more flexibility in routing optical signals across the network [16].

Optical switching technology can be categorized into Optical Circuit Switching (OCS), Optical Packet Switching (OPS) and Optical Burst Switching (OBS). The following sections describe and discuss these different optical switching technologies.

2.3 OPTICAL CIRCUIT SWITCHING

The Optical Circuit Switched networks are built on the concept of lightpaths. An optical circuit switching (OCS) node shown in Figure 2-3 consists of optical cross-connects (OXCs) connected by a set of optical fibers. A lightpath, an all-optical path that spans over multiple links in the optical network, is setup between a source and destination for transferring data. The establishment of lightpaths involves several tasks. These tasks include topology and resource discovery, routing, wavelength assignment, and signaling and resource reservation. Once a lightpath is setup, all data carried by one input λ will flow from a specific output λ. This specific wavelength can not be used for any other connection, because it is particularly reserved for only this connection.

During the setup process of the lightpaths between clients, the OXCs perform switching and routing functions. However, because of the limited number of wavelengths per fiber, it is difficult to create a full mesh of lightpaths between all end users in a large OCS networks.

In addition to this, wavelength-routed lightpath connections are fairly static and may not be able to accommodate the highly variable and bursty nature of Internet traffic in an efficient manner. Sending this traffic over static lightpaths results in the inefficient utilization of bandwidth [17]. In order to achieve better wavelength utilization, traffic grooming has to be employed to support statistical multiplexing of data from different users [23].

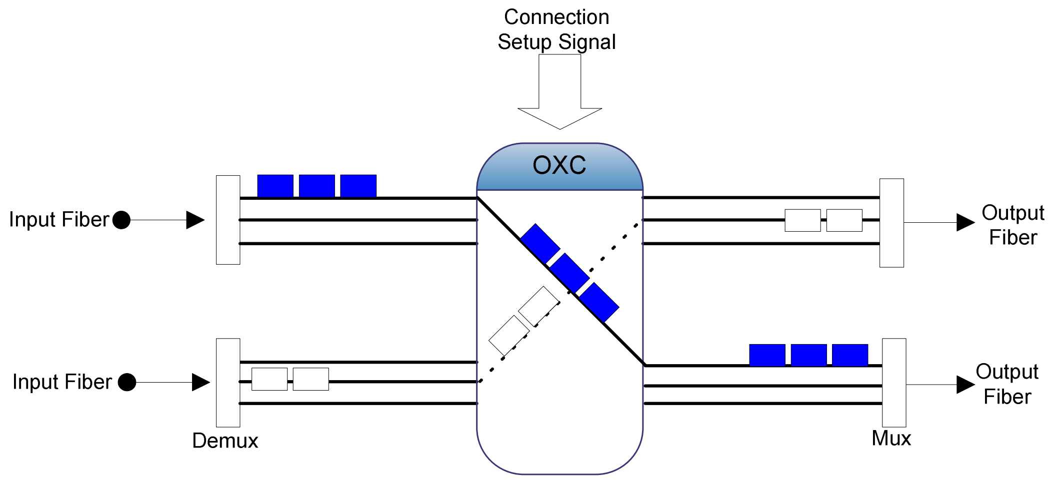


Figure 2-3 An OCS Node [18]

2.4 OPTICAL PACKET SWITCHING

In OPS architecture, the data packets along with the control information are transmitted over optical communication links. As you can see in Figure 2-4, at each core node, the control information is extracted and processed in the electronic domain and then the packet is switched to the next node. The user data is transmitted in optical packets and these optical packets are never processed in the electronic domain. They are switched within each optical switch entirely in an optical domain. No optical-to-electrical and electrical-to-optical conversions are required so the user data remains as an optical signal for the entire path from source to destination.

In addition to the lack of unnecessary optoelectronic (OE) conversions in the core nodes, OPS provides high efficiency, full transparency, increased scalability and flexibility for future growth of networks. Because of these benefits of OPS, it is considered to be one of the most promising solutions for end-to-end delivery of high bitrate data, video, and voice signals across optical networks of the future [24].

However, for the current optical networking technology, the main problem with OPS technology is the lack of optical buffers that are necessary to resolve output port contention. Any packet switch, no matter whether it is optical or electronic needs buffering. Although optical buffering technology is still immature, Fiber Delay Lines (FDL) can be used to delay packets for a fixed time. An FDL is a long fiber through which a packet when traversed will suffer propagation delay in the range of a few nanoseconds. The delay experienced in configuring the switch for each incoming packet is an additional problem for OPS networks [19].

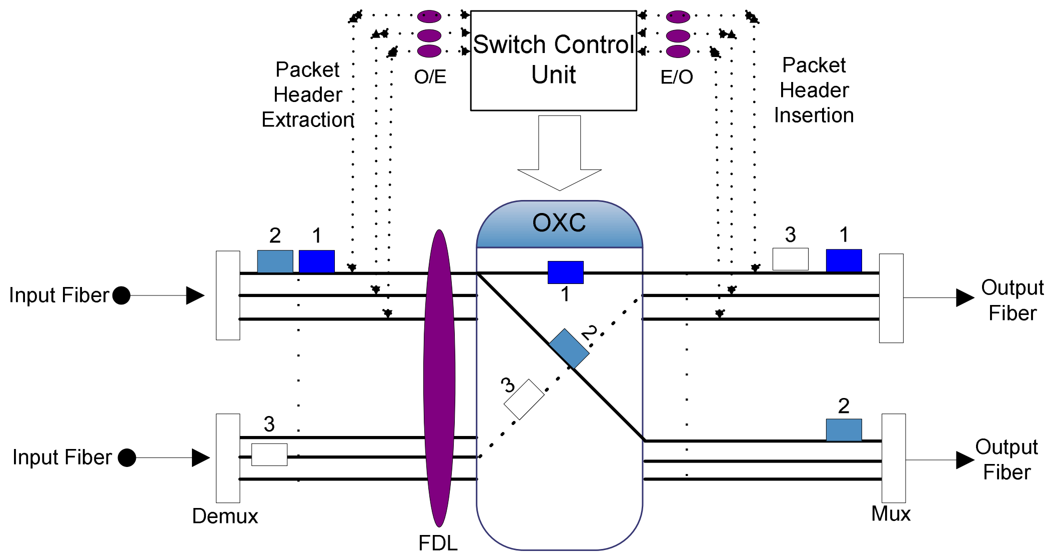


Figure 2-4 An OPS Node [18]

2.5 OPTICAL BURST SWITCHING

Optical Burst Switching was presented as an alternative to Optical Packet Switching and it did not last long for OBS to receive considerable attention in optical networking community. For handling bursty traffic in optical networks, OBS is the most promising technique in the literature up to now and it is based on ATM block transfer (ABT) which is an ITU-T standard for burst switching in ATM networks [25].

In OBS technology, transmission units are called bursts and they are mainly composed of several IP packets, ATM cells, an HDTV frame or raw bit streams. In OBS, main idea is to set up a bufferless optical connection all the way from the source to the destination, and in order to fulfill this objective, a control packet is fed into the network, before transmitting its corresponding data burst. After a predetermined offset delay time, the data burst is transmitted optically. Similar to OCS, the connection is set up uniquely for the transmission of a single data burst, and it stays still until the burst has been transmitted successfully [19].

The OBS network has the most practical All Optical Network (AON) architecture because it combines the best features of circuit switching and packet switching [20]. The main characteristics of OBS are:

1. Packets are assembled at the ingress nodes to form data bursts which are the smallest transmission units
2. Unlike OCS technology, wavelengths are reserved only for a small amount of time a burst is switched. This leads to finer granularity than OCS and utilization of bandwidth
3. No necessity of optical buffers at the intermediate nodes or possibly very small amount of FDL buffering at the switches. Optionally FDLs and wavelength converters can help in reducing burst losses.
4. Control and data packets are carried on different wavelengths, packets in the control plane go through O/E/O conversion, and data bursts in the data plane always remain in the optical domain. Hence data transparency and statistical multiplexing can be achieved simultaneously.
5. Bursts are larger than packets in OPS, and OBS does not require any synchronization process between bursts and their control packets.
6. OBS technology implementation is easier with the current state of physical devices than OPS.
7. Dynamic nature of OBS, allows for network adaptability and scalability, this feature makes it very favorable for the transmission of Internet traffic [21].

In the next chapter, OBS technology and different proposed approaches for implementing OBS are discussed and compared.

CHAPTER 3

OBS NETWORKS

In this chapter, in order to give a clear understanding of the basic technology that lies behind Optical Burst Switching systems, first we introduce the OBS network architecture.

Then we present several fundamental techniques that are needed for the practical implementation of the technology and discuss various proposed approaches for the techniques that are mentioned. These techniques inspired us within the development process of our simulation environment.

3.1 OBS NETWORK ARCHITECTURE

An Optical Burst Switching (OBS) network is consisted of core nodes and end-devices interconnected by WDM fibers as shown in Figure 3-1.

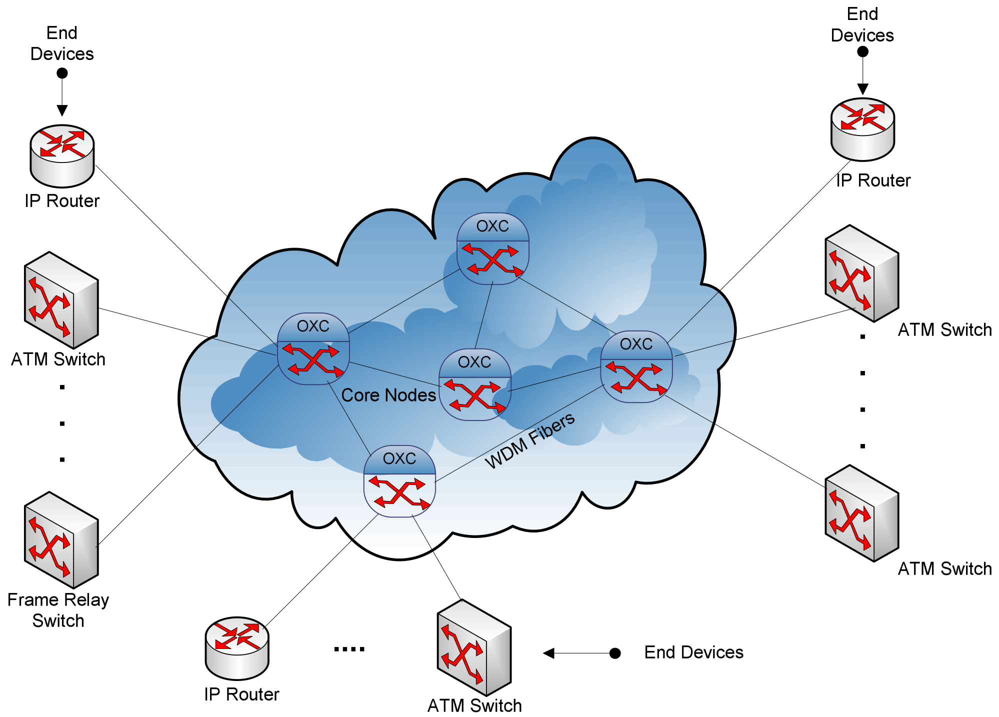


Figure 3-1 OBS Network Architecture [20]

An OBS core node mainly consists of input FDLs (fiber delay lines), an optical switching matrix or an optical cross connect (OXC), an electronic switch control unit (SCU) and routing and signaling processors [26] as seen in Figure 3-2. Data channels are connected to the optical switching matrix and control channels are terminated at the SCU. (DCG: Data Channel Group, CCG: Control Channel Group)

An OXC can switch an optical signal from an input port to an output port without any need to convert the signal to electronics. An OBS interface is present at every OBS edge node. Each OBS edge node is connected to an ingress OBS core node and could be an ATM switch, electronic IP router, frame relay switch, etc.

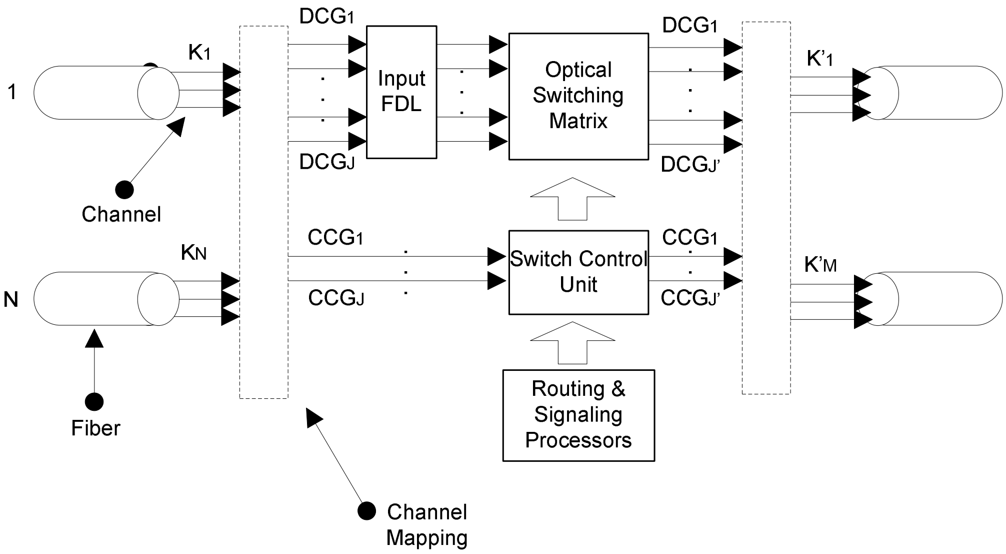


Figure 3-2 An OBS Core Node

End-devices in an OBS network collect traffic from different electronic networks (such as ATM, IP, frame relay, etc.). They classify the incoming traffic per destination OBS end-device address and assemble the packets into larger variable-size units, called data bursts. For each data burst, end-devices also generate a control packet that contains some basic information about the burst, such as burst destination address, burst length, etc. This control packet is immediately sent along the route of the burst and it is electronically processed at each node. Informing the intermediate nodes of the incoming data burst and setting up an end-to end optical path between the source and the destination is the main function of the control packet. After an offset time, i.e., the delay time between the control packet and the burst, the edge node transmits the data burst itself. It travels as an optical signal over the end-to-end optical path which is set up by its control packet. This optical path is disposed after the burst transmission is completed.

This separation of the control information and the burst data is one of the main superiorities of OBS. With the help of this separation, the bursts are transmitted entirely as an optical signal, which remains transparent throughout the network. This yields efficient electronic control and a great flexibility in the format and transmission rate of the user data.

Generally, the time it takes the control packet to reach the destination node is equal to the end-to-end propagation delay plus the sum of all the processing delays at all the intermediate core nodes. On the other hand, the burst goes through the OBS switches without any processing or buffering delays, so the time needed for a burst to reach the destination node is only equal to the end-to-end propagation delay. The transmission of a burst is delayed by an offset time. During this offset delay time, switch control units at the intermediate nodes has the chance to process the control packet associated with the burst and configure their optical switch fabric. The offset time is a function of the number of hops that the control packet has to traverse from source to destination [20].

3.2 BURST ASSEMBLY ALGORITHMS

Burst assembly is the process of assembling incoming data from various sources, such as an IP router, into bursts at the edge of an OBS network. As packets arrive from the higher layer, they are stored in electronic buffers according to their destination and class. The burst assembly mechanism must then place these packets into bursts based on some assembly policy.

The architecture of a typical OBS ingress node is shown in Figure 3-3. The switching unit forwards incoming data packets to the burst assembly units. The data packets that go to the same OBS egress node are classified and processed in one burst assembly unit. Mostly, there is one assigned assembly queue for each traffic class. The burst scheduler is in charge of many tasks in order to maintain a flawless OBS network. These tasks are;

- Creating data bursts and their corresponding control packets
- Adjusting the offset delay time for each data burst
- Scheduling bursts on each output link
- Forwarding the control packets and data bursts to the intermediate nodes [26]

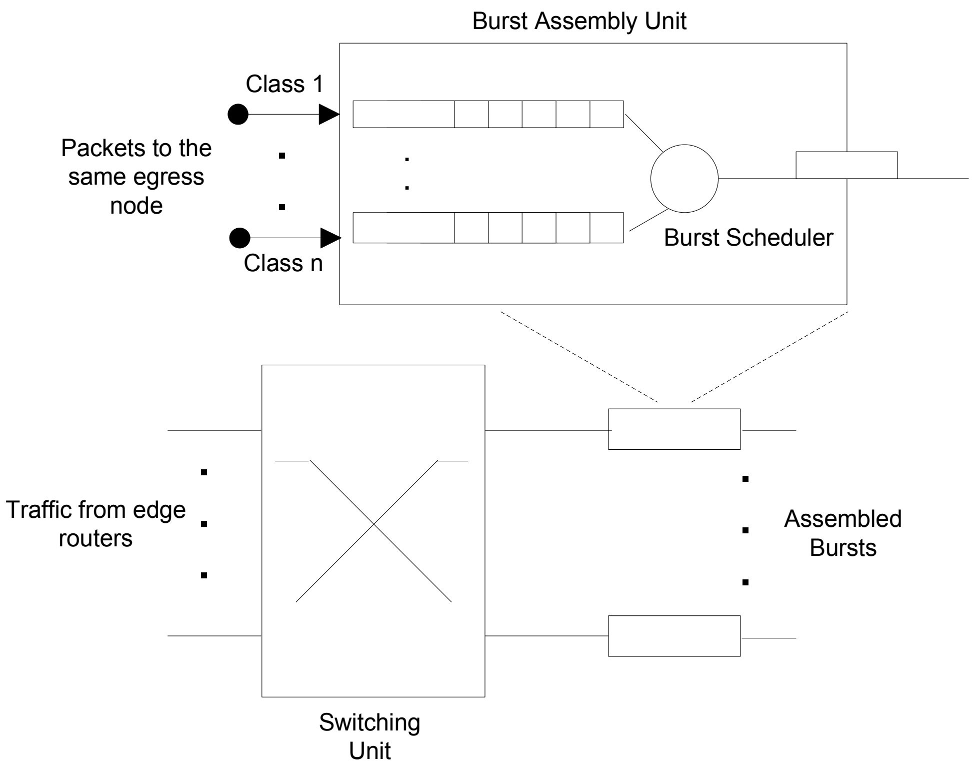


Figure 3-3   Architecture of an OBS Ingress Node

The key parameter in burst assembly is the trigger criterion for determining when to create a burst and send the burst into the network. The trigger criterion for the creation of a burst is very crucial, because it controls the characteristic of the burst arrival into the OBS core. There are several types of burst assembly techniques adopted in the current OBS literature. The most common burst assembly techniques are timer-based and threshold-based ones [17].

In the timer-based scheme, a timer starts at the beginning of each new assembly event. After a predetermined time *T*, all data packets that arrived within this period are assembled into a burst. In timer-based burst assembly algorithm the bursts' inter-arrival time is a constant value.

In threshold-based burst assembly approaches, a limit is assigned to the maximum number of packets contained in each burst. Hence, fixed-size bursts will be generated at the network edge. However, a threshold-based burst assembly approach will generate bursts at non-periodic time intervals [18].

3.3 SIGNALING SCHEMES IN OBS

While a burst is in transmission over an OBS core network, a signaling scheme has to be carried out in order to allocate resources and to configure optical switches for the burst at each core node. This signaling scheme in an optical burst-switched network is generally implemented using out-of-band control packets. In an out-of-band signaling scheme, the control packet associated with a data burst is transmitted on a different wavelength from the burst itself. The out-of-band control packet travels along the same route as the burst and gives information to each core node along the route about the configuration in the optical cross connect to accommodate the arriving burst at the appropriate time.

In the following chapter, several OBS signaling protocols that have been proposed in the research literature will be described in detail.

3.3.1 JUST-ENOUGH-TIME (JET)

In the JET signaling technique as shown in Figure 3-4, a control packet, i.e., burst header packet (BHP) is sent from the source towards the destination on a control channel [27]. The BHP is processed in the core nodes in order to establish an all-optical data path for the corresponding data burst. If the resources are available and the reservation is successful, the optical switch will be configured before the burst arrives. At this point, the burst waits at the source in the electronic domain for a predetermined offset time. After this delay time, the data burst is sent optically from the source [1]. Offset time is calculated as:

$$OT = h\delta + ST \tag{3-1}$$

Where h is the number of hops between the source and the destination, δ is the per-hop burst header processing time, and ST is the switching reconfiguration time. The burst will be discarded in the case of unsuccessful reservation at the core node. The main difference of JET signaling technique from other one-way signaling mechanisms is delayed reservation and implicit release.

As seen in Figure 3-4, in a delayed reservation technique, the channel is reserved from the exact arrival time of the data burst at that node using offset time information coming from the control packet. In an implicit release technique, intermediate node needs extra information from the control packet such as burst length in order to presume the exact time when to release an existing reservation. Because of the need for extra information on the control packets, systems which implement JET algorithms are rather complicated than the others [17].

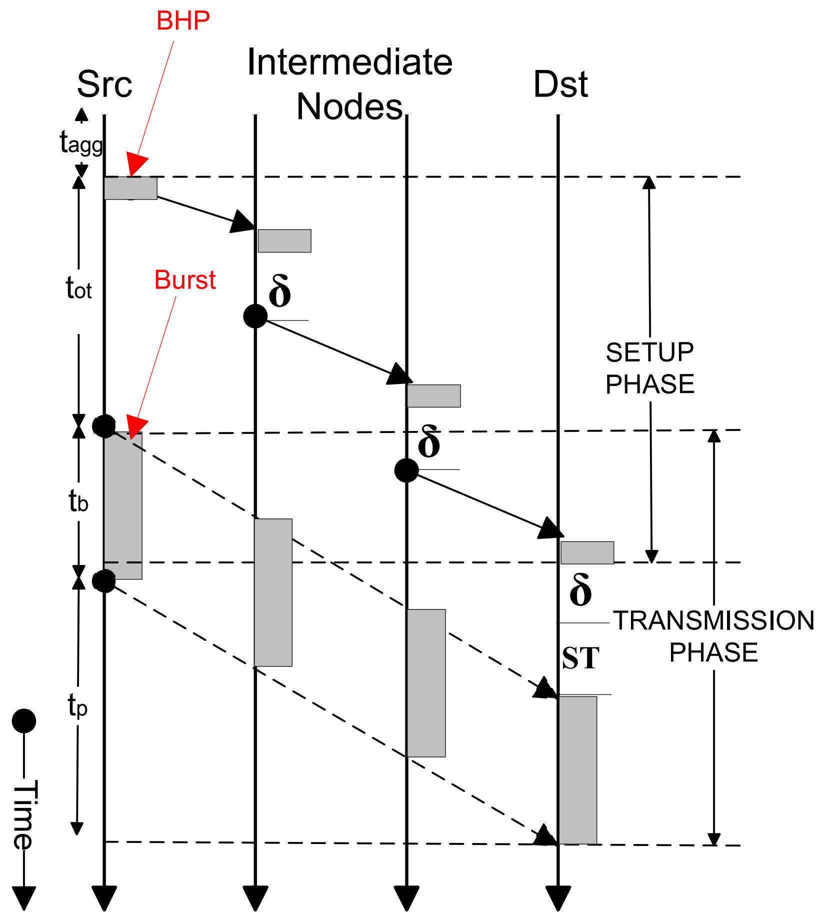


Figure 3-4 Just-Enough-Time (JET) Signaling Technique

Another important aspect of JET signaling technique is its void filling characteristic [28]. A burst can be transmitted in between two already reserved bursts, so bursts are accepted with a higher probability in JET. In addition to this, JET has a better utilization of bandwidth and lower blocking than the other existing signaling techniques.

There are other similar one-way based signaling techniques, such as Tell-and Go (TAG) and Just-in-Time (JIT).

In the TAG approach, unlike from JET algorithm, offset delay time is not determined at the source beforehand [29]. The data burst is delayed at each core node in order to allow time for the control packet to be processed and for the switch to be configured. But this delay requires the use of fiber delay lines (FDL), which is not the desired way for solution.

JIT is similar to JET except that JIT uses immediate reservation and explicit release approach instead of delayed reservation and implicit release [30]. As seen in Figure 3-5, in this method, the channel is reserved immediately after the control packet reaches the immediate node and an additional control message is sent after the data burst, from the source towards the destination, in order to release an existing reservation.

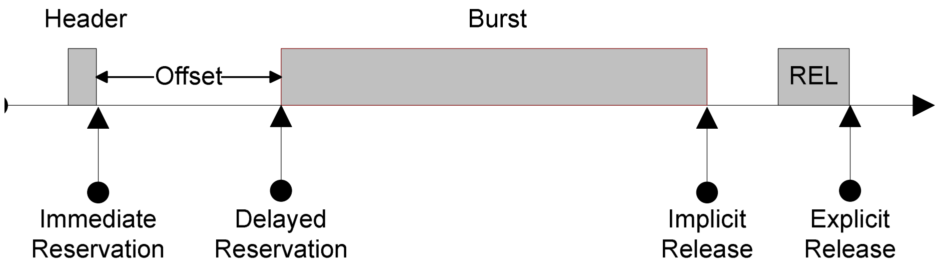


Figure 3-5 Reservation and Release Mechanisms in OBS

Figure 3-6 compares JET and JIT signaling scenarios respectively. The main advantage of using these one-way based techniques is the low end-to-end delay for data transmission over an optical network but it should be mentioned that there is higher packet loss due to data burst contentions for resources at the bufferless core network.

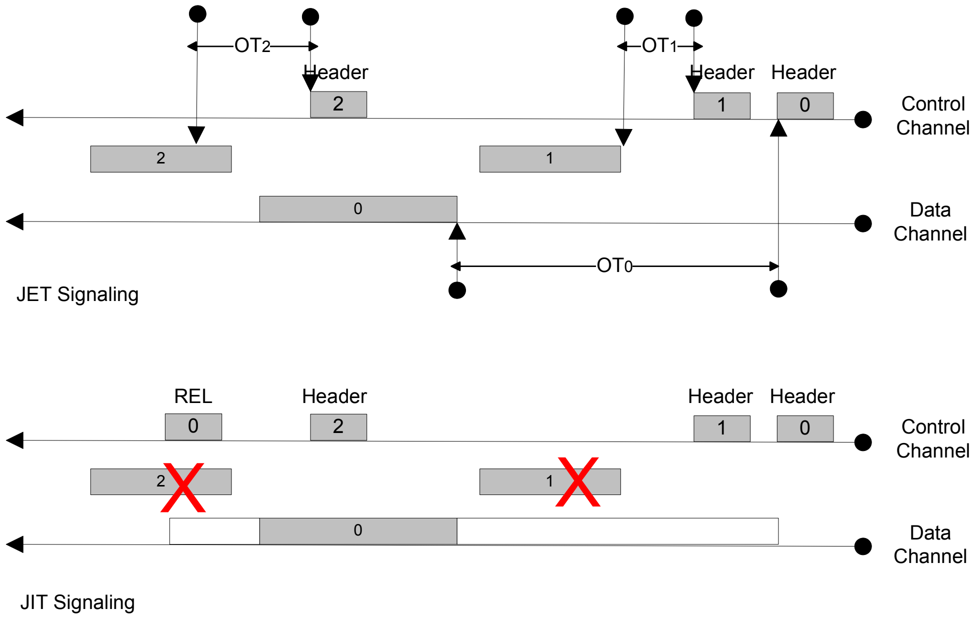


Figure 3-6 Comparison of JET and JIT based signaling

3.3.2 TELL-AND-WAIT (TAW)

In the TAW signaling technique as shown in Figure 3-7, there are three types of BHP, which have various tasks in order to manage the signaling process. A “SETUP” BHP is sent along the burst's route to collect wavelength availability information at every intermediate node along the path. At the destination, a wavelength assignment algorithm is executed, and the reservation period on each link is determined based on the earliest available channel times of all the intermediate nodes. A "CONFIRM" BHP is generated and sent in the reverse direction (from destination to source). This packet reserves the wavelength for the needed duration at each intermediate node. At any node along the path, if the required wavelength is not available, a "RELEASE" BHP is sent to the destination to release the previously reserved resources. If the "CONFIRM" packet arrives the source node successfully, then the data burst is sent into the core network [29].

TAW is the only two-way signaling technique among all other protocols and it has a disadvantage which is the round-trip setup time, i.e., the time taken to set up the channel. It takes three times the one-way propagation delay from source to destination for the burst to reach destination. However, the data loss is very low compared to other techniques. Hence, TAW algorithm can be used in the networks with loss-sensitive traffic.

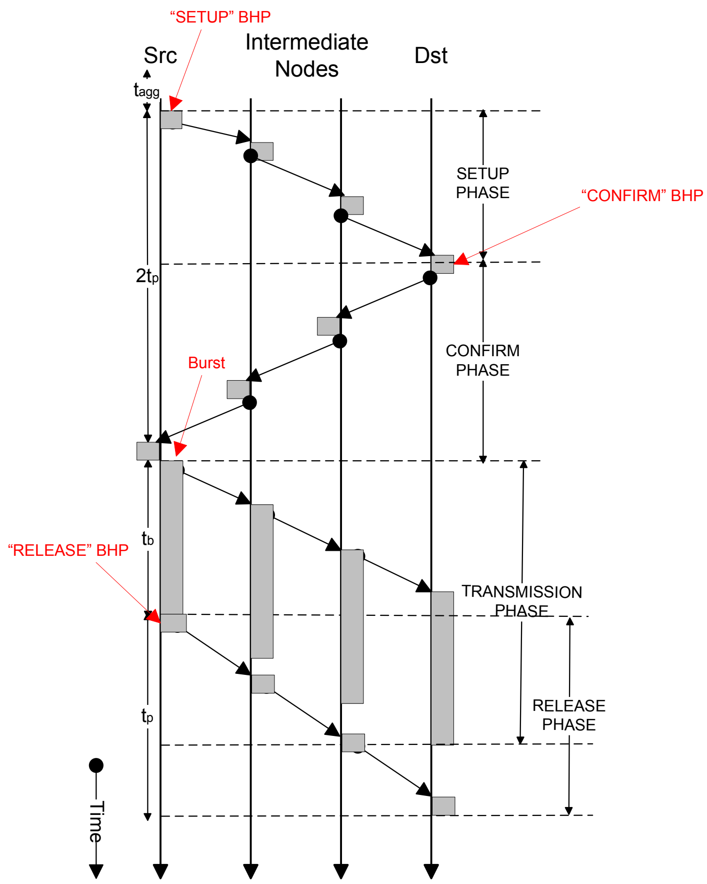


Figure 3-7 Tell-and-Wait (TAW) Signaling Technique

Table 3-1 summarizes the all four signaling techniques explained in Chapter 3.3.

Table 3-1 Summary of the different OBS Signaling Techniques

Signaling	Direction	Initiation	Reservation	Release	Delay	Loss
TAW	two-way	src./dest.	immediate	explicit	high	low
TAG	one-way	source	delayed	implicit	least	high
JET	one-way	source	delayed	implicit	low	high
JIT	one-way	source	immediate	explicit	low	high

3.4 SCHEDULING ALGORITHMS IN OBS

In OBS, when a data burst arrives to an intermediate node, it must be sent to the next node without being stored, because of the lack of the optical buffers. Hence, a wavelength must be assigned to it on a suitable outgoing link. In order to achieve this goal, the scheduling algorithm has to be implemented at intermediate core nodes as well as ingress nodes.

Following the arrival of the control packet at a core node, a channel scheduling algorithm is initiated to assign the unscheduled burst to a data channel on the outgoing port. In the core node, there exists a channel scheduler that extracts the burst arrival time and duration of the unscheduled burst from the control packet. The algorithm usually needs to maintain the latest available unscheduled time (LAUT) or the horizon, gaps, and voids on every outgoing data channel. The LAUT of a data channel is defined as the earliest time at which the data channel is available for an unscheduled data burst to be scheduled [2]. A gap is the time difference between the arrival of the unscheduled burst and ending time of the previously scheduled burst. A void is the unscheduled duration between two scheduled bursts on a data channel.

We can classify data channel scheduling algorithms into two main categories: with and without void filling. Basically these algorithms differ from each other due to the type and amount of channel information that is stored at each intermediate node about every channel. In data channel scheduling algorithms without void filling, the $LAUT_i$ on every data channel is stored by the channel scheduler [26]. In void filling algorithms, the starting time and ending times are maintained for each burst on every data channel.

In the following sections, traditional non-void filling scheduling algorithms, First Fit Unscheduled Channel (FFUC) and Latest Available Unscheduled Channel (LAUC), and traditional void-filling scheduling algorithms, First Fit Unscheduled Channel with Void Filling (FFUC-VF) and Latest Available Unscheduled Channel with Void Filling (LAUC-VF) will be described.

3.4.1 WITHOUT VOID FILLING CHANNEL SCHEDULING ALGORITHMS

3.4.1.1 FIRST FIT UNSCHEDULED CHANNEL (FFUC)

The FFUC scheduling algorithm keeps track of the LAUT (or horizon) on every data channel. When the burst arrival time of the incoming burst is larger than the unscheduled time (LAUT) of the data channel, this means that there is an available place on this data channel The FFUC algorithm tracks all the channels in a static order and assigns the first available channel for the new incoming burst. FFUC algorithm implementation is a very simple because it only needs to maintain one value ($LAUT_i$) for each channel. The FFUC algorithm can be illustrated in Figure 3-8 (a)

$LAUT_i$ values for data channels D_0 and D_3 are less than the burst arrival time of the unscheduled burst so only data channels D_1 and D_2 are available for the duration of the unscheduled burst. If the channels are ordered based on the index of the wavelengths (D_0, D_1 ..., D_w), we can say that FFUC algorithm will schedule the arriving burst on outgoing data channel D_1. The time complexity of the FFUC algorithm is only O(log W). The primary drawback of FFUC is the high burst dropping probability as a trade-off for simplicity in scheduling.

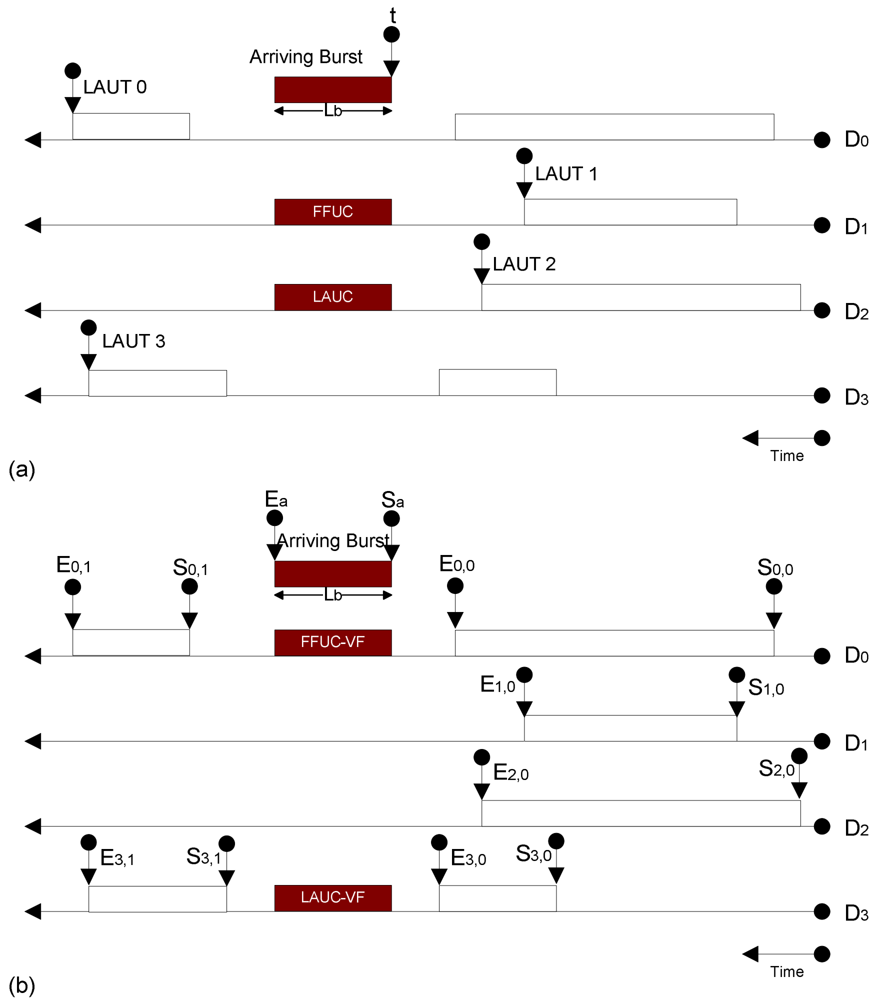


Figure 3-8 Channel Assignment after (a) nonVF (b) VF algorithms

3.4.1.2 HORIZON OR LATEST AVAILABLE UNSCHEDULED CHANNEL (LAUC)

The LAUC or Horizon [2] scheduling algorithm keeps track of the LAUT (or horizon) on every data channel and assigns the data burst to the latest available unscheduled data channel. The LAUC algorithm can be illustrated in Figure 3-8 (a). As mentioned earlier, only data channels D_1 and D_2 are available for the duration of the unscheduled burst. In LAUC algorithm, the latest available unscheduled data channel is preferred so because $LAUT_2$ is larger than $LAUT_1$, it can be said that LAUC algorithm will schedule the arriving burst on outgoing data channel D_2. The time complexity of the LAUC algorithm is O(log W).

3.4.2 WITH VOID FILLING CHANNEL SCHEDULING ALGORITHMS

3.4.2.1 FFUC WITH VOID FILLING

The FFUC-VF scheduling algorithm keeps track of the starting and ending times for each scheduled data burst on every data channel. Only with the help of this additional information, the utilization of the voids between two data burst assignments is possible. In FFUC with void filling algorithm, the first channel with a suitable void is chosen.

The FFUC-VF algorithm is illustrated on Figure 3-8 (b). Based on the starting and ending times of the data bursts, all the data channels D_0, D_1 and D_2 and D_3 are available for the duration of the unscheduled burst. If the channels are ordered based on the index of the wavelengths (D_0, D_1,..., D_w), we can say that FFUC-VF algorithm will schedule the arriving burst on outgoing data channel D_0. If N_b is the number of bursts currently scheduled on every data channel, then a binary search algorithm can be used to check for a suitable data channel. Hence, the time complexity of the FFUC-VF algorithm is $O(\log(WN_b))$.

3.4.2.2 LAUC WITH VOID FILLING

The LAUC-VF [26] scheduling algorithm keeps track of the starting and ending times for each scheduled data burst on every data channel. The objective of this algorithm is to utilize voids between two data burst assignments and the channel with a void that minimizes the gap is chosen.

The LAUC-VF algorithm is illustrated on Figure 3-8 (b). Based on the the starting and ending times of the data bursts, all the data channels D_1, D_2, D_3

and D_4 are available for the duration of the unscheduled burst. It is observed that D_3 had the least gap thus; so LAUC-VF algorithm will schedule the arriving burst on outgoing data channel D_3. If N_b is the number of bursts currently scheduled on every data channel, then a binary search algorithm can be used to check for a suitable data channel. Hence, the time complexity of the LAUC-VF algorithm is $O(\log(WN_b))$.

Lately, researchers proposed several algorithms which optimize the previously described scheduling algorithms. One of the most intriguing algorithms is the Minimum Starting Void (Min-SV) algorithm for selecting channels for incoming data bursts [28]. Although Min-SV has the same scheduling criteria as LAUC-VF, its data structure is different, it is constructed by implementing a balanced binary search tree. By constructing this tree, Min-SV achieves a loss rate as low as LAUC-VF and processing time as low as Horizon (LAUC).

3.5 CONTENTION RESOLUTION

In optical burst-switched networks, transmission of the data burst is not implemented on the routes totally reserved for this particular packet like OCS networks. Hence, there is always the possibility of contention between data bursts at intermediate nodes. Contention will occur if multiple data bursts from different input ports are destined for the same output channel at the same time. Generally, contention in traditional electronic packet switching networks is managed by using electronic buffers; but, in the optical networks, it is more difficult to implement buffers, since optical buffering technology is still immature.

By implementing contention resolution policies, such as optical buffering, wavelength conversion and deflection routing, contention and packet loss can be reduced. Dropping bursts is inevitable when there is no available unscheduled channel, and a contention cannot be resolved by any one of the contention resolution techniques. The policy for choosing which bursts to drop, is called the soft contention resolution policy. This policy reduces the overall burst loss rate and enhances link utilization. Several soft contention resolution algorithms have been proposed in the literature, including the shortest-drop policy [8], segmentation [6], and look-ahead contention resolution [7].

The contention resolution policies are referred as reactive approaches because they are initiated after contention occurs. These policies attempt to resolve contentions rather than avoiding the contentions. In addition to this, aim of these contention resolution techniques is to minimize the packet loss based on the local information at that node.

In this section, several possible methods for resolving contention in OBS networks will be discussed.

3.5.1 OPTICAL BUFFERING

Contention in traditional electronic packet-switching networks is solved by storing packets in electronic buffers (namely RAMs); but, RAM-like buffering is not yet implementable in the optical domain.

In optical networks, delaying packets for a fixed amount of time is possible with the use of fiber delay lines (FDLs) [31]. These fiber delay lines can be implemented in stages [32] or in parallel [33], in order to hold a burst for a variable amount of time.

3.5.2 WAVELENGTH CONVERSION

Wavelength conversion is the process of converting the wavelength of an incoming channel to another wavelength at the outgoing channel. Wavelength converters are responsible for this operation converting an incoming signal's wavelength to a different outgoing wavelength. With the help of wavelength converters, reuse of the same wavelength can be accomplished in order to carry different connections in different fiber links in the network. Wavelength converters offer a 10%-40% increase in reuse values when wavelength availability is low [34].

In optical burst switching with the wavelength conversion capability, contention is reduced by utilizing the multiple wavelengths per link. A contending burst may be switched to any of the idle wavelengths on the outgoing link.

3.5.3 DEFLECTION ROUTING

In deflection routing, contention is resolved by forwarding data burst from an output port other than the intended output port. In electronic packet-switched networks, deflection routing is not a favorable method because looping and out-of-sequence delivery of packets is very likely. On the other hand, it may be inevitable to implement deflection routing in all-optical burst-switched networks, where buffer capacity is very low. However, there is limited number of related work about deflection routing in optical burst-switched networks.

In deflection routing, a deflected data burst takes a longer route to its destination, leading to a degradation of the signal quality and increased delay. In addition to this, it is very likely that the data burst will loop indefinitely within the network.

Another problem in deflection routing is the maintenance of the offset time between control packet and corresponding deflected burst. The deflected burst must traverse a greater number of hops than if the burst had not been deflected, so it is possible that at some future time the initial offset time may not be sufficient for the control packet to be processed and for the switch to be reconfigured before the data burst arrives to the switch [17].

3.5.4 BURST SEGMENTATION

In existing optical burst switching approaches, when contention between two bursts cannot be resolved through other contention resolution algorithms, one of the bursts will be dropped entirely, even the overlap between the two bursts is a small fraction of the bursts. Losing a few packets from the contended burst rather than losing the entire burst is more desirable in certain networks with strict delay requirements but relaxed packet loss requirements. A soft contention resolution policy, burst segmentation [6] minimizes packet losses by partitioning the burst into segments and dropping only those segments, which contend with another burst.

In burst segmentation, the burst is consisted of a number of basic transmission units called segments. Each segment includes a segment header and a payload. The segment header contains fields for synchronization bits, source and destination information, error correction information and the length of the segment in the case of variable length segments. The segment payload may carry any type of data, such as IP packets, ATM cells, or Ethernet frames.

Burst segmentation allows the data bursts to be preempted by other contending bursts. This preemption ability of contending bursts enables contentions to be handled in a prioritized manner.

3.6 CONTENTION AVOIDANCE

An alternative approach to handle network contention is using traffic management policies in order to avoid network overload in a proactive manner. Contention avoidance policies try to prevent a network from entering the congestion state before any contention occurs. An ideal contention avoidance policy must fulfill several goals: minimize the packet loss rate, minimize the average end-to-end packet delay, guarantee fairness among all users and operate with minimum additional signaling requirements [35].

Generally, contention avoidance policies can be implemented in either non-feedback-based or feedback-based networks.

3.6.1 NON-FEEDBACK BASED ALGORITHMS

In a non-feedback-based network, the ingress nodes do not have any information about the network state and they cannot respond to changes in the network load. Each ingress node can adjust its own load into the network through traffic shaping or traffic rerouting and load balancing based on a predetermined traffic description. Traffic shaping is implemented by regulating the rate of the data bursts entering the OBS network. Regulating data bursts can be achieved by reshaping the traffic at the edge node periodically and using a proactive reservation scheme [36]. Traffic rerouting on alternative shortest paths (or load splitting) can also be implemented. In order to avoid or minimize link contention in non-feedback-based networks, it is very crucial to define the traffic parameters such as peak rate and average rate at each ingress node precisely.

3.6.2 FEEDBACK-BASED ALGORITHMS

In a feedback-based network, network contention can be avoided by dynamically varying the data burst flows at the edge routers according to the latest network status and its available resources. When the state of the network and the availability of the resources are changed, the edge router should regulate its data burst transmission rate to the network, accordingly. In OBS networks, it is generally assumed that the edge routers have sufficient buffering capacity.

The critical issues in feedback-based networks are defining the feedback mechanism and determining what type of information must be delivered to the edge router [37]. When the edge router receives the necessary information, the main design issues include how to interpret the delivered information and how to implement the interpreted information on the current network state.

Figure 3-9 summarizes the two key elements and their related algorithms in feedback-based contention avoidance networks: control and signaling strategies [35]. The feedback control strategy is related with the actions the node performs after the arrival of the feedback messages to the node. For example, an edge node can regulate the transmission rate through admission control strategies or reroute data burst flows going through the congested link. The feedback signaling strategy is related with the measure of the current network state and its communication with the edge and core nodes.

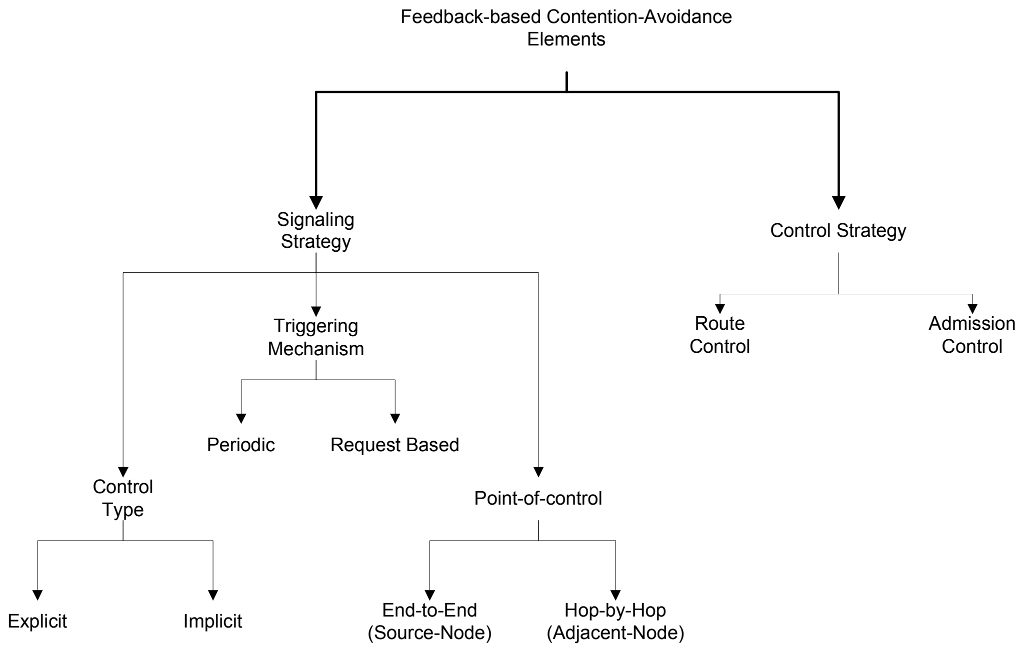


Figure 3-9 Feedback-based Contention Avoidance Elements

The feedback signaling strategy can be classified into three main mechanisms:

(a) Feedback control type: determines the type of the control messaging which is responsible for the deliverance of the current network state to the source. The signaling type can be explicit or implicit. In the explicit signaling type, the feedback signal explicitly gives information about the congestion state and the requested transmission rate. In the implicit signaling type, the feedback signal indicates the rate of the packet loss on a particular link or in a node.

(b) Feedback triggering mechanism: indicates how often the feedback signals will be transmitted to source nodes. For example, the feedback signals can be transmitted periodically or based on the request of the other nodes. After the feedback signal is triggered, it is optional to be broadcasted to all sources or sent to specific nodes.

(c) Feedback point-of-control: refers to the nodes, which respond to the feedback messages and initiate congestion avoidance algorithms. These nodes can be the edge nodes or the adjacent core nodes. These are referred as end-to-end and hop-by-hop signaling, respectively.

There are several proposed solutions about contention avoidance issue in the literature. Rerouting some of the traffic from heavily loaded paths to underutilized paths is one of the solutions in order to avoid contention in feedback-based OBS networks [10]. In this solution, an intermediate node forwards feedback messages containing the load information of its overloaded output links to the edge nodes. Another similar approach is balancing the data burst traffic between predetermined alternative routes [38].

A different contention avoidance algorithm is to implement a TCP-like contention avoidance approach by regulating the burst transmission rate. In this approach, TCP ACK packets are sent from egress edge nodes to the ingress edge nodes. Ingress edge nodes calculate the most congested links, and reroute their traffic accordingly.

A potential drawback of these algorithms is that rerouting the data bursts to alternative links can possibly cause congestion on a different link and a network instability problem can be inevitable. In addition to this, when the network operates at a very high speed and the round trip delay is large, the response of the edge nodes to the network change, tend to be slow.

CHAPTER 4

IMPLEMENTATION OF A NEW FEEDBACK-BASED CONTENTION AVOIDANCE ALGORITHM: DYNAMIC REROUTING BASED ON WEIGHTED DIJKSTRA ALGORITHM

In this thesis, we propose a feedback-based contention avoidance scheme in which, the core nodes will communicate with each other for congestion information by generating and sending explicit feedback signals. The source nodes can reroute the traffic based on these feedback signals to avoid or minimize congestion. Rerouting will be implemented by dynamically computing new routes for new bursts. During the computation process of a least congested route, weighted Dijkstra algorithm will be implemented using predefined cost parameters.

Feedback-based contention avoidance elements that are mentioned in Chapter 3 are selected carefully in order to achieve lower burst loss rate and end-to-end delay.

In our proposed approach, feedback control strategy is mainly facilitated in route control. The idea is to route the data bursts to the less congested links and reduce the high load on the links that discard packets intensively.

The feedback signaling strategy in this thesis, involves the following approaches to the control elements.

(a) Feedback control type:

In our approach, explicit type of feedback control messaging is selected. Feedback messaging is very crucial and feedback packets carry critical information about the current state of the network, such as the number of discarded and forwarded burst on each link of the network.

(b) Feedback triggering mechanism:

In our approach, the feedback signals will be generated when a burst is discarded at the core node. Hence, it cannot be said that they are transmitted periodically or based on a request. After the feedback signal is triggered, it is sent to all previously visited nodes.

(c) Feedback point-of-control:

In our approach, the nodes that forwarded the discarded burst, respond to the feedback messages and initiate congestion avoidance algorithms.

4.1 ROUTE CALCULATION TECHNIQUES

Route calculation techniques are implemented extensively in telecommunication network systems in order to determine the most suitable path that is traversed by the communication packets according to a chosen metric.

In feedback-based congestion avoidance algorithms, route calculation techniques play an important role in a similar manner. In these algorithms, rerouting process is initiated after the congestion information which is extracted from the feedback packets, is evaluated in the source and core nodes in the network.

In an OBS network, the route calculation can be categorized into two basic types, namely static and dynamic calculations. In static-route calculation, one or more routes are calculated beforehand due to some static metric, like physical distance or number of hops. For example, one or more routes can be computed using Dijkstra's shortest-path algorithm in fixed alternate path routing. For the networks with a steady traffic, these static techniques are mostly favorable; however such a mechanism might be inefficient to accommodate dynamic bursty traffic since the congestion could be high on some links when data bursts try to travel on the shortest routes which pass through these links.

In dynamic-route calculation, the routes are calculated periodically based on certain dynamic traffic information. This information can be about link congestion or number of contentions. The information necessary to make the route computation can be acquired in two ways, they are probe-based and broadcast approaches.

In the probe-based approach, after the source node transmits the control packet into the network, it can also send a probe packet following the control packet. The probe collects the desired information from the core nodes, and returns to the sender with the collected network information.

In the broadcast approach, the core nodes periodically transmit relevant congestion information to all edge nodes. However, in this method, packets carrying the congestion information to the core nodes can constitute undesirable high packet traffic. To overcome this deficiency, the feedback information about a link can be sent to all edge nodes only if there is a change in the congestion status of the link from the previous value [10].

In our proposed algorithm, we present a new route-computation mechanism using Dijkstra's shortest-path algorithm, which the routing table is dynamically adapted to the current network state. Necessary congestion information will be obtained via feedback packets that are broadcasted from congested nodes and destined to source nodes.

4.2 IMPLEMENTATION OF WEIGHTED DIJKSTRA ALGORITHM

At the initialization of the network, Dijkstra's shortest-path algorithm using physical distance metric is implemented at the core nodes in order to carry out the static routing of the data bursts in the network. In this algorithm, physical distance for every source-destination pair is calculated and the path with the minimum physical distance related to the specific source-destination pair is selected. When a negative feedback packet arrives to a core node, the core node implements weighted Dijkstra algorithm to apply dynamic route calculation.

Weighted Dijkstra algorithm is a way of dynamically rerouting traffic from heavily loaded paths to under-utilized paths based on the feedback information of network links. According to the data accumulated from the negative feedback packets, the core nodes will introduce new routing paths to the data bursts in order to avoid congestion.

The interpretation of the information coming from the feedback packets is the essential part of this thesis work.

4.2.1 DEFINITION AND USAGE OF THE NEGATIVE FEEDBACK MESSAGES

Each core node in an OBS network is equipped with a burst generator that generates data bursts with exponentially distributed burst lengths. These data bursts are assumed to have been already assembled at the ingress nodes of the core nodes; hence no burst assembly algorithm is implemented in our study. Control packets are formed immediately after the generated data bursts are forwarded from the burst generators to the core nodes. Along with data burst and control packets, a third type of packet, namely, "negative feedback packet" is introduced in our proposed algorithm.

The negative feedback packet is generated in the core node when a control packet of an incoming data burst, is discarded due to contention at one of the links of the core nodes.

Each node keeps the number of forwarded or discarded packets on its each output link. After the generation of the feedback packet on the congested node, the congested link information is acquired by the feedback packet.

Negative feedback packets follow exactly the same path with respect to its corresponding discarded packet but in a reverse direction and on the same wavelength where control packets are transmitted (Figure 4-1). While control packets are on transmission during their route from source to destination, they store the necessary output port information of the links on a reverse path table for every visited hop. Reverse path information is extracted from control packets after the generation of negative feedback packets and negative feedback packets are able to follow the same route as the control packets with the help of these reverse path tables.

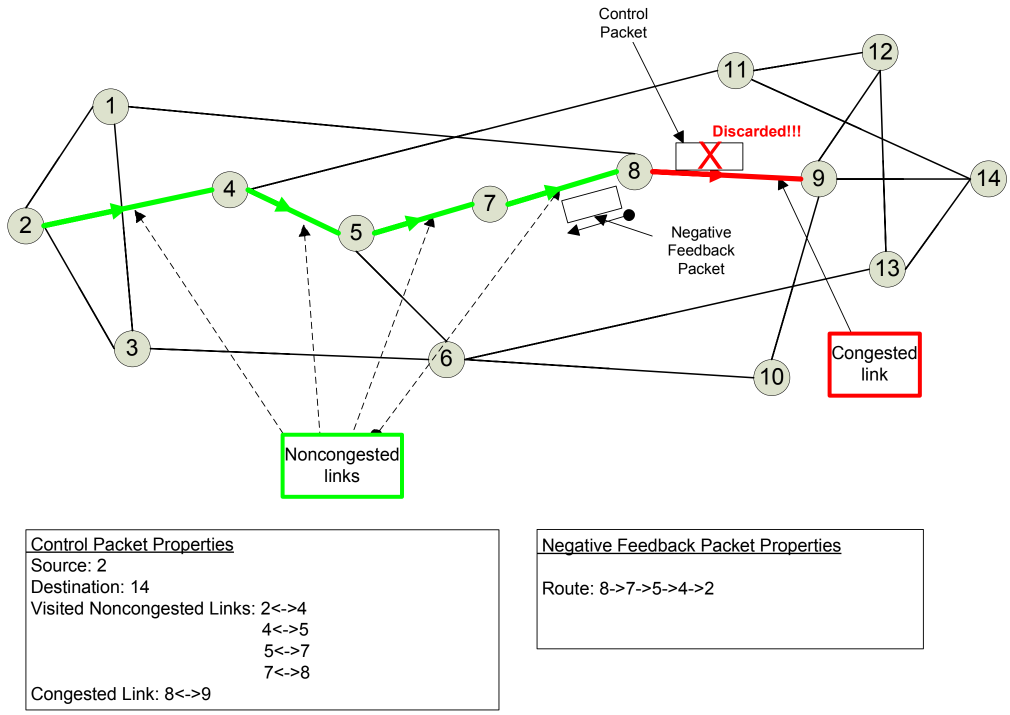


Figure 4-1 Routes of Control Packet and Negative Feedback Packet

During the route of the feedback packet, it starts to collect the information of the number of forwarded or discarded packets on each output link of the visited nodes. In addition to this, it shares with the newly visited nodes, the information of the number of forwarded or discarded packets of the previously visited nodes.

By this method, with the help of negative feedback packets, core nodes have knowledge about the drop rate of every output link of other core nodes in the same network.

The algorithm works as seen in Figure 4-2:

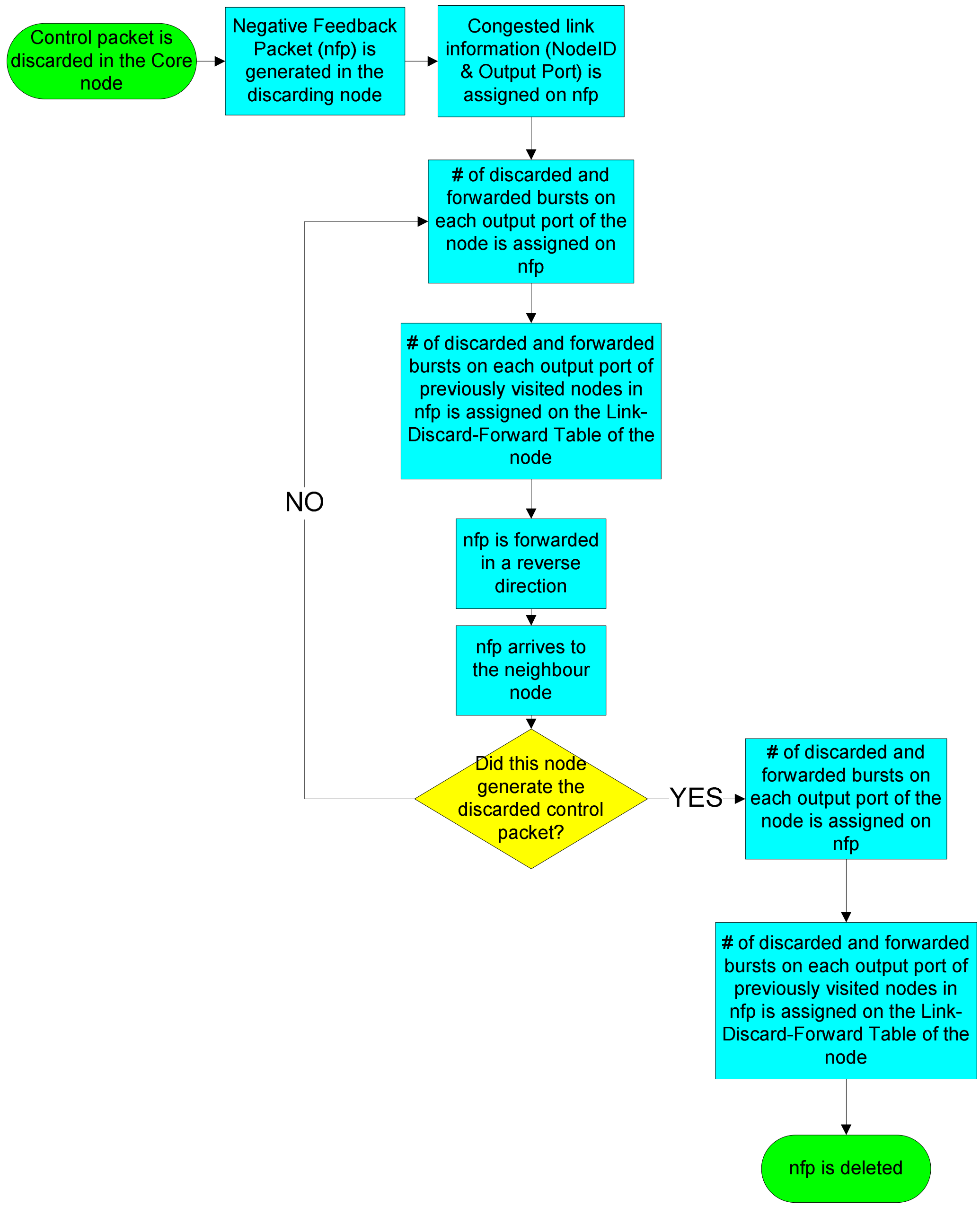


Figure 4-2 Flow Chart of the Usage of the Negative Feedback Packet

4.2.2 INTERPRETATION OF THE NEGATIVE FEEDBACK PACKETS

When a negative feedback packet arrives to a core node, the information about the number of forwarded or discarded packets on each output link of the previously visited nodes are extracted from the feedback packet (Table 4-1). This information is kept in the link-discard-forward tables of the core nodes (Table 4-2).

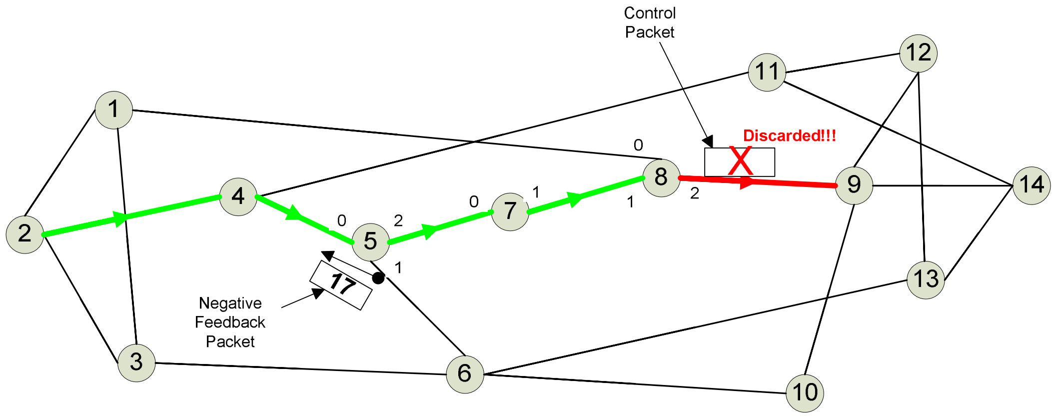


Figure 4-3 Interpretation of Negative Feedback Packet

Table 4-1 Sample Link-Discard-Forward Information in nfp(17)

Forwarding Node No	Node Output Port No	Discarded Packets No	Forwarded Packets No
8	0	7	49
8	1	3	74
8	2	12	61
7	0	6	47
7	1	2	21

Table 4-2 Sample Link-Discard-Forward Table in Core5

Node No	Node Output Port No	Discarded Packets No	Forwarded Packets No
*8	0	7	49
*8	1	3	74
*8	2	12	61
*7	0	6	47
*7	1	2	21
**5	0	7	72
**5	1	4	18
**5	2	15	86
***9	0	13	43

Table 4-2 (cont'd) Sample Link-Discard-Forward Table in Core5

***9	1	5	52
***9	2	3	27

*Information extracted from nfp (17)

**Information already available on Core5

*** Information extracted from previous nfp's

According to Dijkstra's shortest-path algorithm using physical distance metric, the weight of each link is assigned according to the actual physical length of the link. Figure 4-4 shows the 14-node NSFNET on which the simulation is implemented. The distances shown are in km. Using 14-node NSFNET network topology, the initial weights in Table 4-3 are assigned at the very beginning of the simulation.

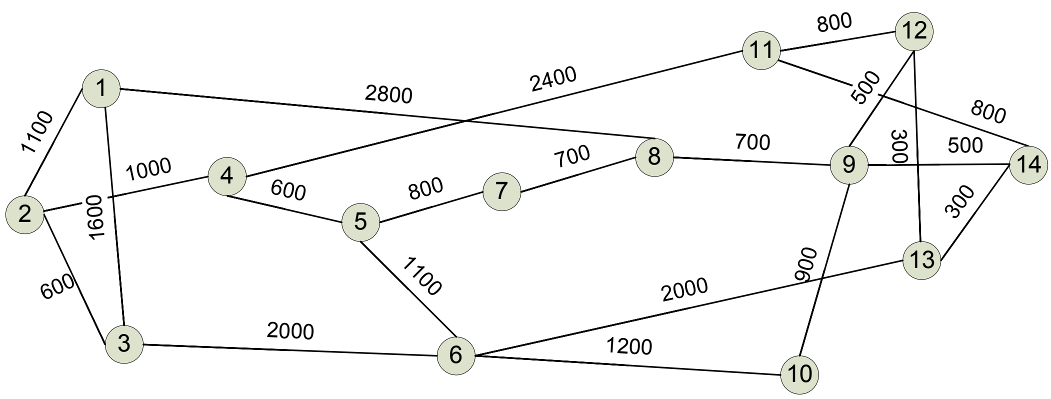


Figure 4-4 OBS network under study [39]

Table 4-3 Initial weights of the links in the network topology

Links		Weight of the link
Core-1	Core-2	1100
Core-1	Core-3	1600
Core-1	Core-8	2800
Core-2	Core-3	600
Core-2	Core-4	1000

Table 4-3 (cont'd) Initial weights of the links in the network topology

Core-3	Core-6	2000
Core-4	Core-5	600
Core-4	Core-11	2400
Core-5	Core-6	1100
Core-5	Core-7	800
Core-6	Core-10	1200
Core-6	Core-13	2000
Core-7	Core-8	700
Core-8	Core-9	700
Core-9	Core-10	900
Core-9	Core-12	500
Core-9	Core-14	500
Core-11	Core-12	800
Core-11	Core-14	800
Core-12	Core-13	300
Core-13	Core-14	300

We consider two different approaches while interpreting the number of discarded and forwarded packets in each core node.

In order to initiate the rerouting process using updated link weights, one of these interpretation approaches has to be implemented. Prior to the implementation of these approaches, each core node first checks if a previous nfp about that specific link arrived at the node within a period which is ten times of the mean interarrival time of the Poisson process. This period is dependent on the traffic load and we chose it to be ten times of the mean interarrival time of the Poisson process because we do not want it to be significantly small or large compared to the burst arrival rate of the network. If it is chosen too small, the effect of our proposed algorithm will be very limited. On the contrary, if it is chosen too large, the weight of almost every link will be updated by taking some obsolete data into account. Simulations implemented with various period times showed us that ten times of the mean interarrival time of the Poisson process is a reasonable value for this period. If the previous negative feedback packet is old enough, then none of the interpretation approaches is implemented assuming the corresponding link is not fully congested.

To achieve this type of control on the incoming nfp, core nodes keep the arrival time of the latest negative feedback packet for each link. When a negative feedback packet arrives to the core node, its arrival time is compared with the arrival time of the latest negative feedback packet. After the calculation and evaluation of this period, it will be decided whether interpretation process is needed to be done. If the nfp about the specific link arrives to the node within this period time, then the interpretation process is initiated.

The first approach is related with the average drop rate of the links, about which we have knowledge. Each core node updates its Link-Discard-Forward Table after the nfp of the discarded control packet arrives to the node. Drop rate of every link in the Link-Discard-Forward Table can be calculated easily. Nevertheless, in order to get up-to-date information, we take into account of the discarded and forwarded packets, which are accumulated between two consecutive feedback messages related to specific link.

$$diff\_discard\_no = link\_discard\_no - prev\_link\_discard\_no \quad (4\text{-}1)$$

$$diff\_forward\_no = link\_forward\_no - prev\_link\_forward\_no \quad (4\text{-}2)$$

$$droprate = \frac{diff\_discard}{diff\_discard + diff\_forward\_no} * 100 \quad (4\text{-}3)$$

is calculated for each link.

If drop rate of this specific link is more than %30, then the weight of this link is increased by an additional cost value of +100. This additional cost value is chosen after the minimum physical distance for every source-destination pair is calculated. It is seen that the longest distance between any core nodes in the network is 4500 km which are the pairs Core1-Core11 and Core2-Core13. The shortest distance between any core nodes in the network is 300 km, which are the pairs Core12-Core13 and Core13-Core14. Moreover while setting the additional cost value, we took into consideration that after the weight of the link is incremented by this value multiple times, the total weight of a path involving this link should be higher than the weight of a less congested path. Also if we set this value too high, then the congestion on previously idle links will be too high so this feedback algorithm may not return better results at all. Using these assumptions, the additional cost value is set to 100 and we observed that the results of our simulations are better than other selected values as seen in Figure 4-5.

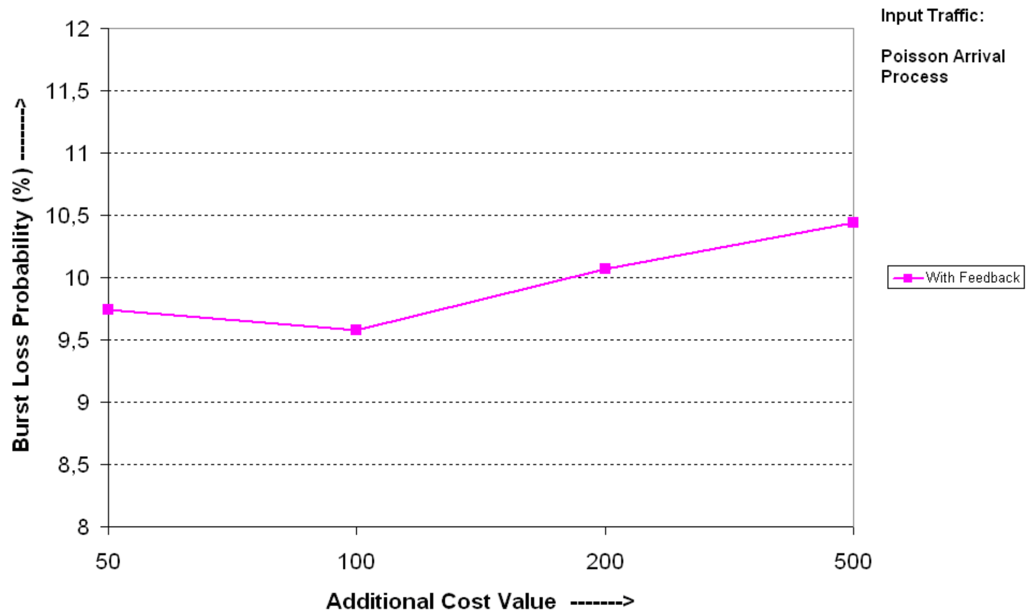


Figure 4-5 Effect of the Additional Cost Value on the Proposed Algorithm

We selected the threshold drop rate of a link as %30 because according to the assumptions we made in our simulations, we saw that burst drop probabilities does not exceed %20 of total packets and we thought that %30 threshold is appropriate for a burst drop rate of a link.

The second approach is about the congested link, which discarded our corresponding control packet. While the negative feedback packet is visiting every core node on the way, the information about the congested link is extracted in every visited node. In each node, the weight of this congested link is increased by an additional cost value of +100.

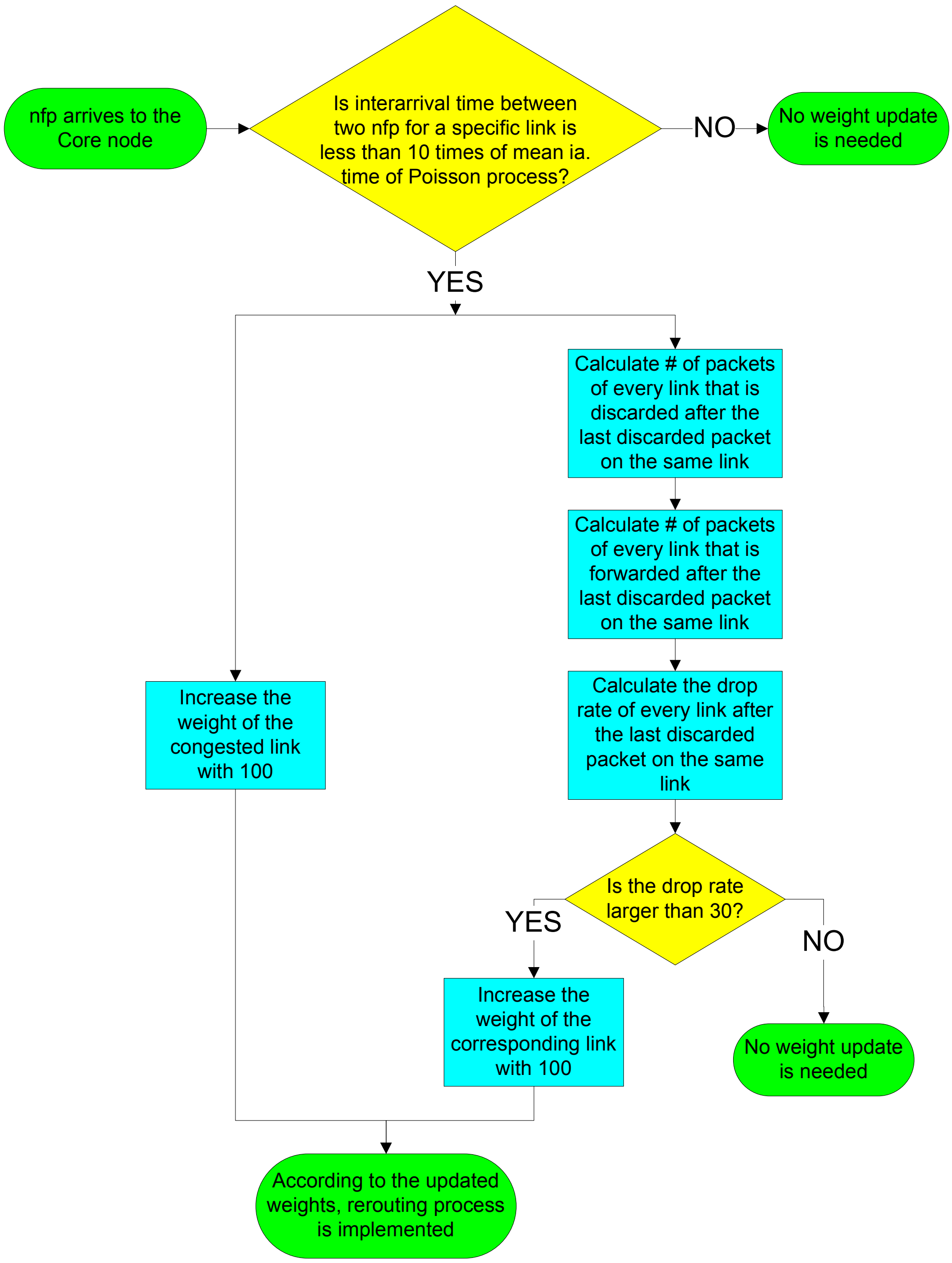


Figure 4-6 Flow Chart of the Interpretation of the nfp

4.3 SIMULATION MODEL AND NETWORK SETUP OF THE PROPOSED ALGORITHM

In order to evaluate the performance of our proposed algorithm, first we needed to create a network simulation environment which can be easily implemented for OBS technology. After choosing the appropriate network simulator, software design process was initiated and a network model which is widely studied by the photonic network researchers is chosen. Throughout the design process, the fundamental characteristics of OBS that are explained in Chapter 3 are taken as basis for our simulation model.

In the following sections, the details of our network simulation environment will be given.

4.3.1 RUNNING SIMULATION IN OMNET++

In the simulation stage of this proposed algorithm, OMNET++ [47] simulation environment is used. OMNET++ is a C++-based discrete event simulator for modeling communication networks, multiprocessors and other distributed systems. It is an object-oriented program, which the programming codes are written in C++.

OMNET++ is fully programmable and modular, and it was designed to support modeling very large networks built from reusable model components. It is open source, free for non-profit use, and it has a fairly large user community.

There are many features of OMNET++, which make it a great candidate for both research and educational purposes:

- Simulation models that are easy to trace and debug
- Powerful graphic interface that makes the internals of a simulation model fully visible to the person running the simulation
- Strong network graphics (animations of message flow, etc.)

An OMNeT++ model is composed of modules, which communicate with message passing. These basic modules are named as "simple modules" and they are written in C++, using the simulation class library. Simple modules can be grouped into compound modules unlimitedly.

Simple modules basically send messages via gates which are the input and output interfaces of modules. The messages are sent out through output gates and arrive through input gates. An input and an output gate can be linked with a connection. It is possible to assign properties to the connections such as propagation delay, data rate and bit error rate [41].

4.3.2 CLASSES IMPLEMENTED IN SIMULATION ENVIRONMENT

There are two main classes, which are written in C++ for our simulation environment, namely Generator, and CRouter classes. Figure 4-7 shows these classes modeled by OMNET++'s topology description language, NED.

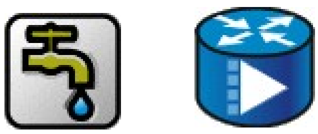


Figure 4-7 Models of Generator and CRouter classes in OMNET++

Generator and CRouter classes and its functions are listed in Table 4-4. Each function of these classes is defined in order to perform tasks required for the realistic simulation of an Optical Burst Switched network and implementation of our proposed feedback-based contention avoidance algorithm.

Table 4-4 Classes and its functions implemented in simulation environment

Class Name	Class Function
Generator	initialize()
Generator	handleMessage()
CRouter	initialize()
CRouter	handleMessage()
CRouter	genControlMessage()
CRouter	forwardControlMessage()
CRouter	forwardDataMessage()
CRouter	gennegFeedbackMessage()
CRouter	forwardnegFeedbackMessage()
CRouter	handleFeedbackMessage()
CRouter	topologyUpdate()
CRouter	topologyUpdateNULL()
CRouter	DistanceAssign14()
CRouter	finish()

The tasks performed by the Generator and CRouter classes are described in detail in the following sections.

4.3.2.1 TASKS PERFORMED BY THE "GENERATOR" CLASS

1. Generates data bursts with Poisson arrivals and exponentially distributed burst lengths. (handleMessage())
2. Assigns destination router information with equal probability to the every generated data burst. (handleMessage())

4.3.2.2 TASKS PERFORMED BY THE "CROUTER" CLASS

1. Implements Dijkstra's shortest-path algorithm using initial link weights during the initialization of the network. (initialize(), topologyUpdateNULL(), DistanceAssign14())
2. Generates control packets for every data burst coming from Generator module. (genControlMessage())
3. Sets the offset time between the burst and its corresponding control packet according to JET signaling technique explained in Chapter 3. (genControlMessage())
4. Assigns destination, burst length, offset time information of the burst to its corresponding control packet. (genControlMessage())
5. Checks the destination information of the incoming control packet with its own module ID and if there is a match, increments its "arrivedpacket_no" information for informative purposes and delete the control packet. (forwardControlMessage())
6. If there is no match, calculates and stores the arrival and departure times of the data bursts using burst length and offset time information acquired from the incoming control packet. (forwardControlMessage())
7. Compares the arrival time of the latest burst with the latest available unscheduled time (LAUT) of the desired link. (forwardControlMessage())
8. If the arrival time of the burst is larger than the LAUT of the desired link, forwards the control packet to the neighbour node from a suitable output port according to predetermined routes after "control packet processing time".(forwardControlMessage())
9. If the arrival time of the burst is smaller than the LAUT of the desired link, increments its "droppacket_no" information for informative purposes and discards the most recent control packet (forwardControlMessage())
10. After discarding the control packet, generates negative feedback packet. (gennegFeedbackMessage())

11. Assigns destination and congested link information to the negative feedback packet. (gennegFeedbackMessage())
12. Forwards the negative feedback packet to the node, which generated the corresponding control packet. (forwardnegFeedbackMessage())
13. Forwards data bursts to the neighbour nodes from the same output port as its corresponding control packet if it has not discarded its corresponding control packet. (forwardDataMessage())
14. Discards data bursts whose control packet is already discarded. (forwardDataMessage())
15. Implements weighted Dijkstra algorithm using updated link weights. (forwardnegFeedbackMessage(), topologyUpdate())

4.3.3 THE OBS NETWORK TOPOLOGY UNDER STUDY

Performance evaluation and network traffic studies on computer networks are carried out by different research groups worldwide, and results of these studies are comparable only if they are based on the same or at least similar network scenarios.

There are two main optical transport network reference scenarios that serve as basis for a great variety of studies in optical networks area. These scenarios are based on the following networks [44]:

- a pan-European network defined in European COST 266 project and denoted as "basic network" (BN) [43].
- a US network based on a former NSF network topology [40] which has been used in many studies that have been published over the last couple of years.

We preferred to implement our simulations on an optical transport network which is studied before by other researchers in order to have a comparison base for our results. Thus, US network based on a former NSF network topology is selected. This is the T3 NSFNET backbone architecture which replaced the earlier T1 architecture in 1992 [40]. Topology of this US network is shown in Figure 4-8.

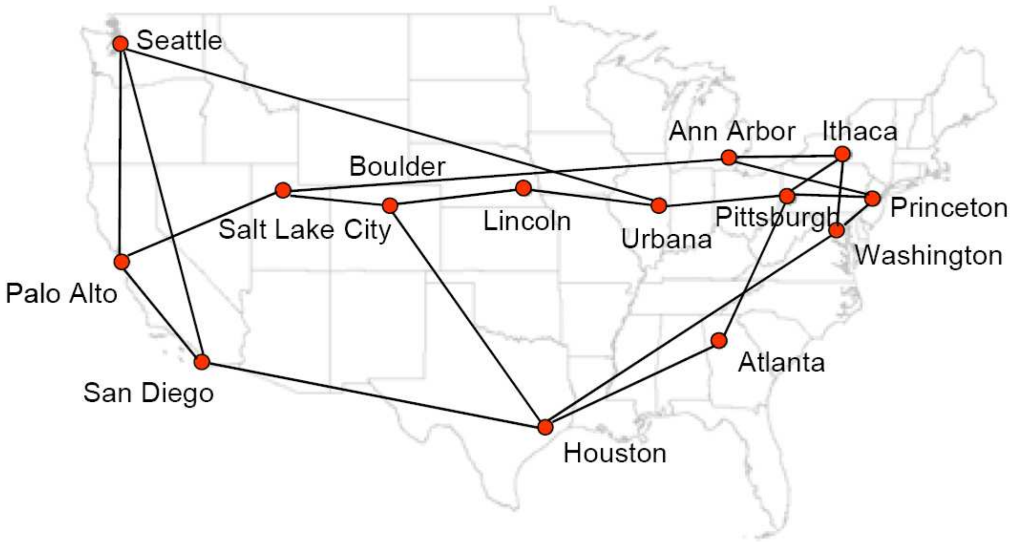


Figure 4-8 14 node NSFNET Topology [44]

The main topological parameters of the network are listed in Table 4-5.

Table 4-5 Simulation parameters and assumptions [42]

	US Network
number of nodes n	14
number of links	21
average node degree	3
average distance (hops)	2,14286

4.3.4 SIMULATION MODEL

The simulation model with core nodes and their burst generator modules can be seen in Figure 4-9. Simulation events are logged and all activities of modules and messages can be tracked in OMNET++ (Figure 4-10).

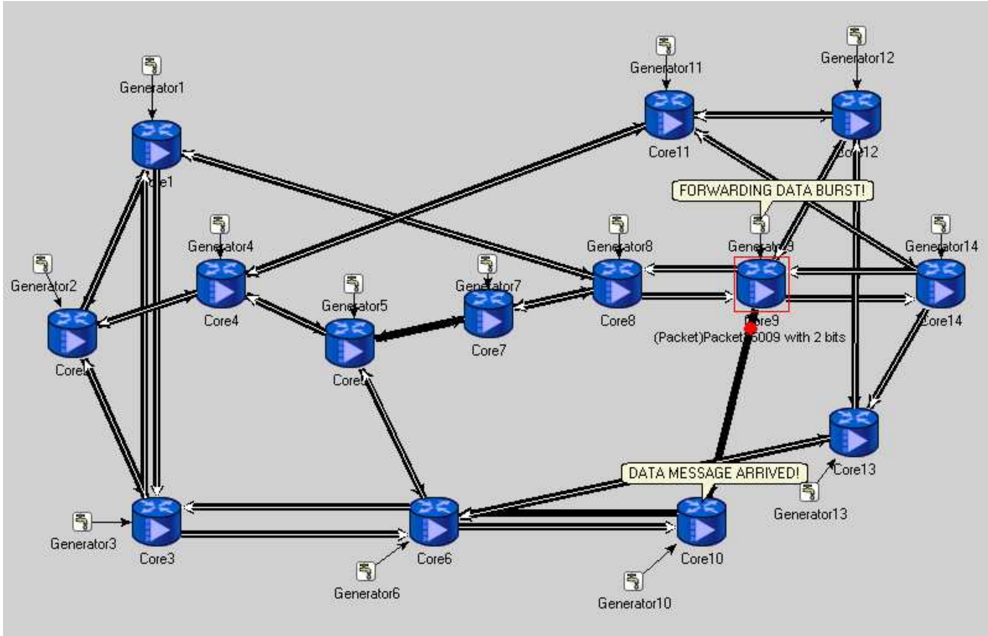


Figure 4-9 The Simulation Model

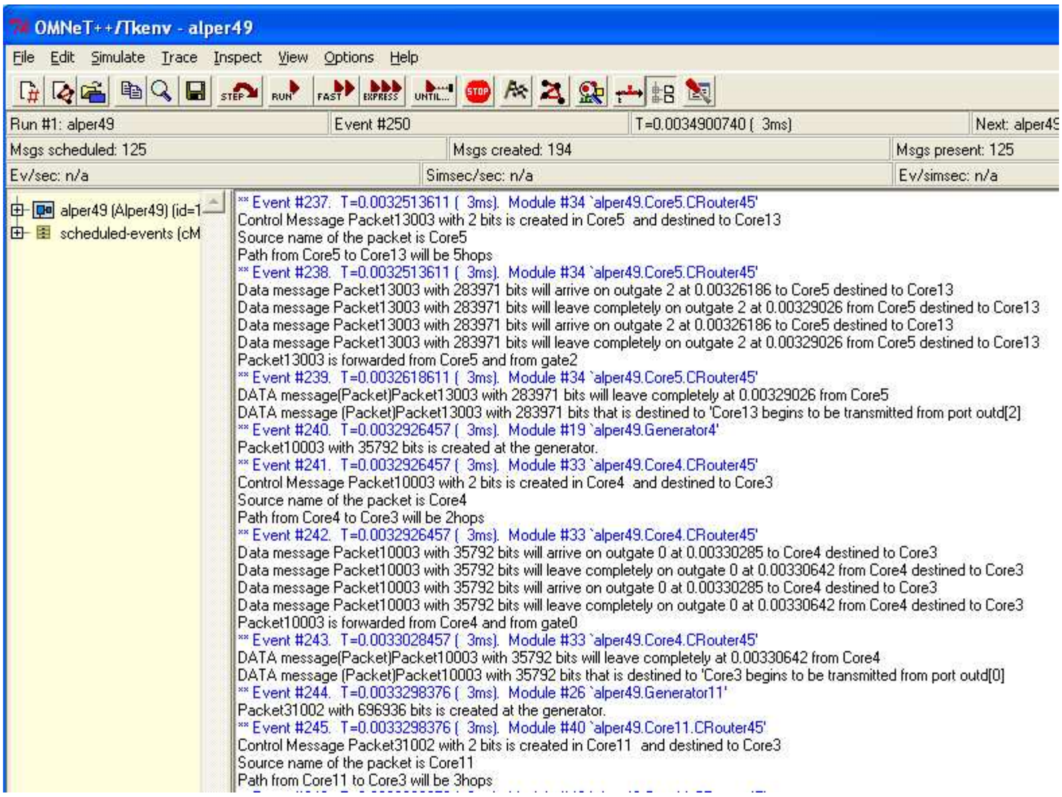


Figure 4-10 The Simulation Event Log

4.3.5 INPUT TRAFFIC ANALYSIS

In our simulations, three different burst arrival processes are implemented. This thesis is mainly interested in networks with Poisson arrivals which are frequently used in applications such as analyzing traffic flows, fault prediction on electric cables, the prediction of randomly occurring accidents, etc. It is one of the most widely used and oldest traffic models. In addition to this, in optical networks, this model is frequently adapted as a burst arrival model [6],[10],[18],[39]. Nevertheless, Bernoulli arrival process and On-Off source model are also implemented and the simulation results using these methods are included in our evaluation.

In networks using Poisson traffic model, burst arrivals into the network are assumed to be Poisson with exponentially distributed burst length. The rate λ

of a Poisson process is defined as the average number of events per unit time (over a long time) [45].

The random variables $\boldsymbol{\tau_1, \tau_2, \tau_3, \ldots, \tau_n}$ are called the interarrival times of the Poisson process. The interarrival times ($\boldsymbol{\tau_1, \tau_2, \tau_3, \ldots, \tau_n}$) are independent of each other and each have an exponential distribution with mean $1/\boldsymbol{\lambda}$. Mean values are set according to the desired traffic load in our simulations.

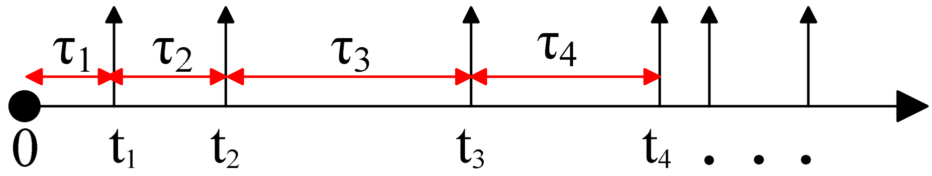


Figure 4-11 Poisson Arrival Traffic Model

In networks using Bernoulli traffic model, independent Bernoulli trials are implemented prior to the generation of the data bursts. Bernoulli trial is an experiment with only two possible outcomes that have positive probabilities p and q such that $p + q = 1$. The outcomes are said to be "success" and "failure" and the probability of success on each trial is a constant value, namely p.

Bernoulli process is a discrete time process, so the number of trials, failures and successes are integers. In Figure 4-12, we can see a sequence of n Bernoulli trials, each with a probability of success, p. Each trial is implemented on every time slot δ.

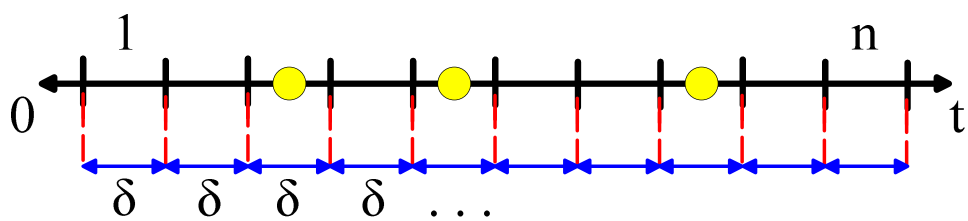


Figure 4-12 Bernoulli Arrival Traffic Model

In networks using On-Off model, a simple two-state Markov chain model (Figure 4-13) is adopted to represent the bursty packet arrival process for each burst generator [48]. A bursty traffic source alternates between active and idle periods. It generates packets back to back during the active periods and stays idle during silent periods. p is defined as the probability of terminating a burst and q is defined as the probability of starting a burst in each time slot.

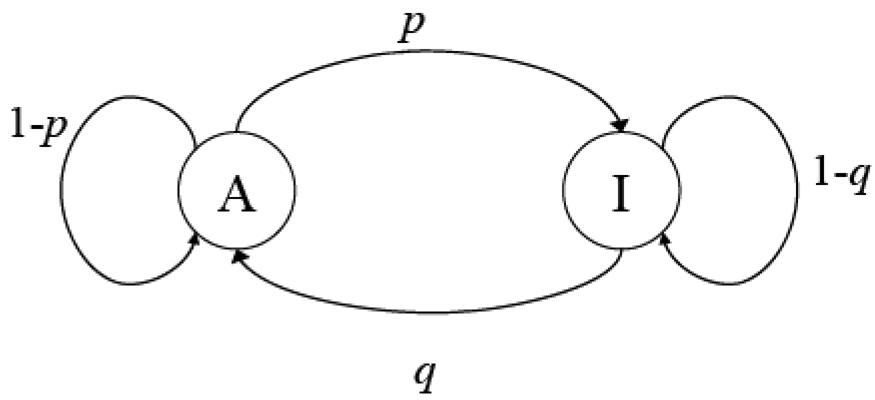


Figure 4-13 On-Off Source Model

In this model with the active period lasting for a duration of i time slots, the mean burst length (L), the mean idle period length and offered load are given by

$$L = \sum_{i=1}^{\infty} p(1-p)^{i-1} i = \frac{1}{p} \tag{4-4}$$

$$\text{Mean idle period length} = \sum_{i=1}^{\infty} q(1-q)^{i-1} i = \frac{1}{q} \tag{4-5}$$

$$\text{Offered load} = \rho = \frac{1/p}{1/p + 1/q} \tag{4-6}$$

In our simulations, the unit of our traffic load is Erlang (E). It is a dimensionless unit and it is used in computer networks as a statistical measure of the volume of the incoming traffic.

Traffic of one Erlang refers to a single resource being in continuous use, or two channels being at fifty percent use and so on. For example, if an office has three telephone operators who are both busy all the time, that would represent three Erlangs (3 E) of traffic, or a radio channel that is in use for thirty minutes during an hour is said to carry 0.5 E of traffic [46].

4.3.6 CHANNEL SCHEDULING ALGORITHM

In our simulations, FFUC scheduling algorithm is chosen for its simplicity. For every output link of a core node, latest available unscheduled time (LAUT) is maintained. The LAUT of an output link is the earliest time at which the output link is available for an unscheduled data burst to be scheduled. If the arrival time of a data burst is smaller than the LAUT of the desired output link, this data burst will be discarded due to the congestion on the link. If the arrival time of a data burst is larger than the LAUT of the desired output link, the data burst will be transmitted on that desired output link and the departure time of this data burst will be assigned as the new LAUT value of that specific link.

4.4 SIMULATION AND EVALUATION OF THE JET SIGNALING TECHNIQUE IN OUR SIMULATION MODEL

In this thesis, we take Just-Enough-Time (JET), which is the most common signaling technique in OBS architecture, as our starting point. As previously mentioned in "3.3 SIGNALING SCHEMES IN OBS", like other one-way based signaling techniques, JET algorithm mostly suffers from the high packet loss.

Due to its advantages like void filling capability, better utilization of bandwidth and lower blocking ability, JET algorithm acquired a solid reputation in the OBS literature. However, high packet loss rate still constitutes a big drawback for the developers who are engaged in bursty traffic and real time, error-free network environments. Thus, countless approaches are made in order to overcome this high packet loss problem.

In [39], various signaling techniques for optical burst-switched networks are discussed and a different signaling technique, called Intermediate-node-initiation (INI) signaling is proposed. Through simulations, it is shown that INI performs better than TAW in terms of average end-to-end packet delay and better than JET in terms of burst loss probability.

In this section, to verify the correctness of our simulation model, we first simulate JET algorithm proposed in [39], with the same simulation parameters that are used in this published work.

4.4.1 IMPLEMENTATION IN OMNET++

In order to evaluate the performance of the JET algorithm that is proposed in [39], a simulation model is developed. In this simulation, we consider a NSFNET topology with 14 nodes as shown in Figure 4-4.

In this model, a network with a single wavelength per fiber is investigated (additional one wavelength is used for the control channel). All links are bi-directional and wavelength channels are operating at 10 Gbps. We consider a simulation where each 14 node in the network is equipped with a burst generator and burst arrivals into the network are assumed to be Poisson with an exponentially distributed burst length of 10^6 bits. It should be noted that in our simulations, data bursts are assumed to have been already assembled at the ingress nodes of the core nodes; hence no burst assembly algorithm is implemented. The destination nodes of every generated burst are assigned with equal probability by the burst generator. Under 10Gbps link transmission rate, the average burst length is set to 0.1 ms. The switching reconfiguration time is 0.01 ms. The processing time of a control packet in each node is set to 100 ns. Simulation time is set to 2 secs. No optical buffering (FDLs) is supported at core nodes. Retransmission of the lost bursts is not considered. The fiber lengths are shown in Figure 4-4, the propagation delays between any two connected nodes range between 1.5ms and 14ms. A static route was chosen between each pair of nodes using Dijkstra's shortest-path algorithm with physical distance metric. In these simulations, only paths that are more than or equal to four hops are considered.

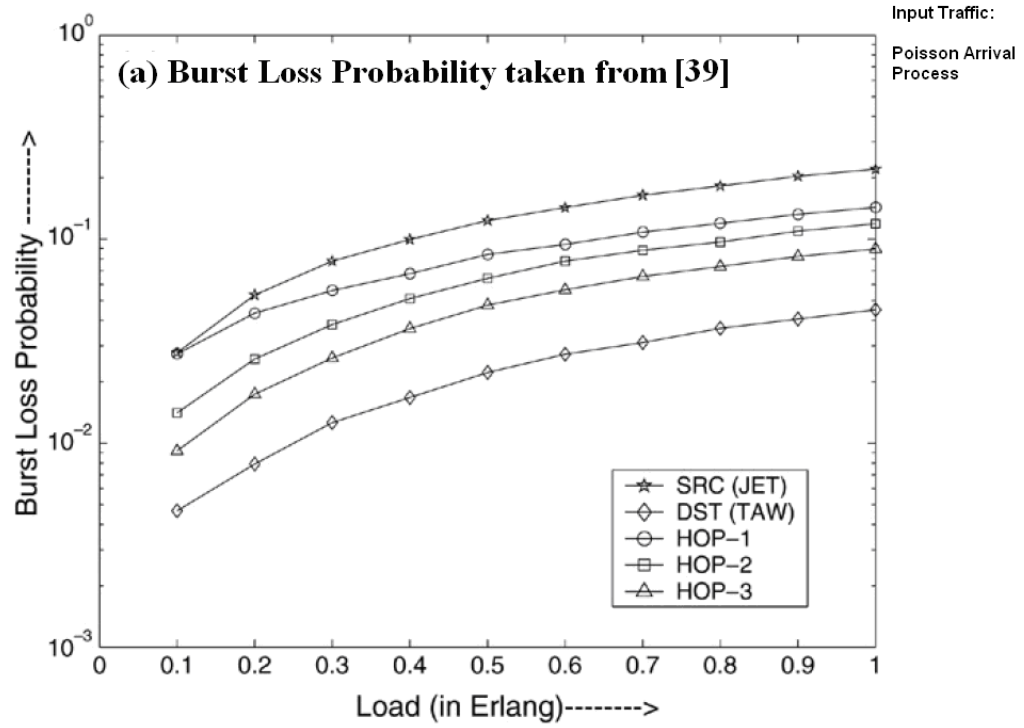
In our network topology, we have 14 nodes equipped with a burst generator. This implies that to achieve One Erlang of traffic in the network, every generator should be busy about %7.14 percent. An average burst is assumed to be 0.1 ms long. If a burst generator is in continuous use, it should generate 104 bursts in every second. To achieve one Erlang of traffic, every generator should generate 714 bursts in every second. We conclude that interarrival times of bursts should be set with mean value 1/714, which is 1400 μs. In Table 4-6, the mean values of the interarrival times of data bursts are listed with the corresponding Erlang value.

Table 4-6 Erlang and Interarrival Time values

ERLANG value	Interarrival Time (mean values)
0,1	14000μs
0,2	7000μs
0,3	4667μs
0,4	3500μs
0,5	2800μs
0,6	2333μs
0,7	2000μs
0,8	1750μs
0,9	1556μs
1	**1400μs**
1,1	1273μs
1,2	1167μs
1,3	1077μs
1,4	1000μs
1,5	933μs
1,6	875μs
1,7	824μs
1,8	778μs
1,9	737μs
2	700μs

The interarrival time is varied according to Table 4-6 and the burst loss probability is analyzed for each load.

Figure 4-14 shows the burst loss probability versus traffic load.



(b) Burst Loss Probability taken from OMNET Simulation

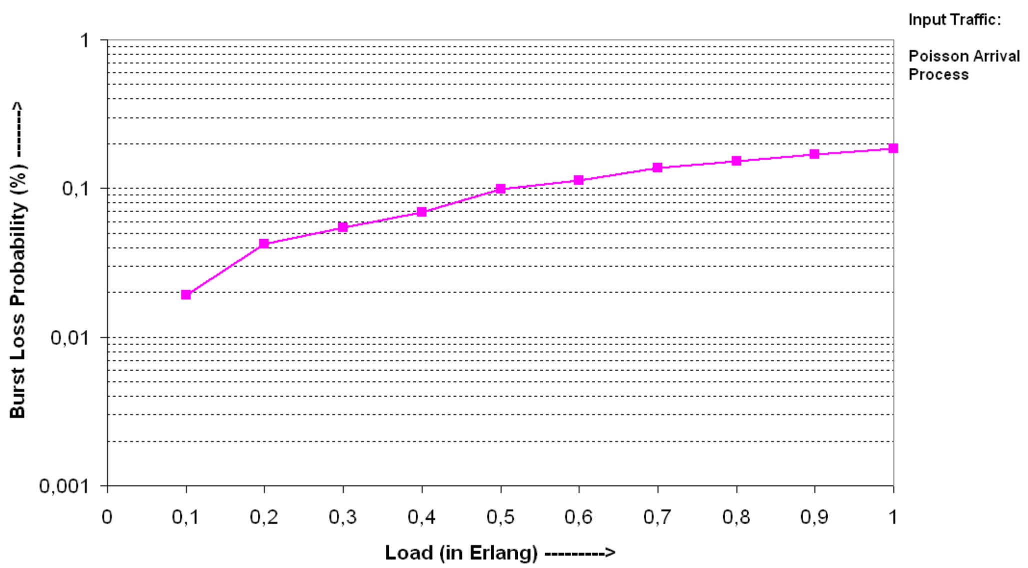


Figure 4-14 Burst Loss Probability Graphs of JET Signaling Technique

In Figure 4-14 (a), the graph line we are interested in is the uppermost one, which is the result of a JET signaling scheme. When we compare the two graphs above, we can clearly conclude that, our simulation results are nearly same with the simulation results in [39].

4.5 SIMULATION AND EVALUATION OF THE PROPOSED ALGORITHM

4.5.1 SIMULATION PARAMETERS AND ASSUMPTIONS

Simulation parameters and assumption values are listed in Table 4-7.

Table 4-7 Simulation parameters and assumptions

Parameter	**Assumption Value**
Control Packet Length	ignored
Feedback Packet Length	ignored
Data Burst Length	has exponential distribution with mean 10^6 bits
Capacity of channel	10 Gbps
Average burst length	0.1 ms (10^6 bits/ 10^{10} bits per sec)
Interarrival Time	has exponential distribution with mean 700-14000 μs
Offset Time	(hopcount * control packet processing time) + switching reconfiguration time
Control Packet Processing Time	100 ns
Switching reconfiguration time	10 μs
Simulation Time	800 miliseconds

4.5.2 DELAY CHARACTERISTICS AND PARAMETERS OF THE JET SIGNALING TECHNIQUE

In this section, we present analytical equations for evaluating the delay characteristics of our OBS signaling technique, JET. It is assumed that no optical buffering (FDLs) is supported at core nodes. In the following analysis, we ignore the delay resulting from the creation of control packets and the execution of the channel selection algorithm [39]. We define the following notation:

- R_{sd} : route from Source s to Destination d.
- t_{cpp} : control packet processing delay at each OBS node (core and edge). t_{cpp} is in the hundreds of ns range.
- t_{sw} : switching time needed for the reconfiguration of the optical cross-connect at each OBS node. t_{sw} is in the tens of μs range.
- t_{agg} : burst aggregation delay based on the assembly technique adopted at the ingress OBS node. In our simulation, delay caused by burst aggregation is ignored.
- t_b : data burst transmission time is directly proportional to the link transmission rate of the network and the burst size. In our simulations, link transmission rate is assumed as 10Gb/s. Burst arrivals into the network are assumed to be Poisson with an exponentially distributed burst length. Mean burst length is selected as 10^6 bits in order to satisfy the condition in [39], which implies that the average burst length is set to 0.1 ms.
- t_{ot} : offset time is calculated as $t_{ot} = ht_{cpp} + t_{sw}$, where h is the number of hops between the source and the destination, t_{cpp} is the per-node control packet processing time, and t_{sw} is the switching reconfiguration time [39]. In our 14-

node NSF backbone network topology, average hop distance is calculated as 2.14286 (Table 4-5). In our simulations, t_{cpp} is assumed as 100 ns and t_{sw} is assumed as 10µs. With a simple calculation, average offset time can be found as ~0.010214 ms.

• t_p^{ij} : propagation delay on the fiber link between nodes i and j. t_p^{ij} is 5 µs/km.

4.5.2.1 PROPAGATION DELAY TIME ANALYSIS

Propagation delay time is the time needed for a data burst between its transmission from the source node to destination node. Propagation delay time of the network is strictly proportional to the distance between the nodes in the network and the number of hops needed to be visited.

Propagation delay on any fiber link between nodes i and j, is 5 µs /km. In our 14-node NSF backbone network topology, the propagation delays between any two connected nodes range between 1.5 ms and 14 ms. This value comes from the fact that shortest link in the network is 300km and the longest link in the network is 2800km (Figure 4-4).

If we assume all links are used with equal share, it can be said that every link has an average 5.4 ms propagation delay. (This value is acquired by simply summing all propagation delays per link and dividing the sum with 21 (# of bidirectional links.) The average hop distance of the network is 2.14, so with a rough calculation, propagation delay of the network is expected to be around 11.5ms. Since shortest path algorithm is used during the static route calculation, shorter links are preferable so our achieved average value of 11 ms makes sense, which is a slightly lower value than the calculated one.

Our proposed approach, "Dynamic Rerouting Based on Weighted Dijkstra Algorithm" routes the data bursts to the less congested links and reduces the high load on the links that discard packets. In our approach, during the routing process, the length of the link is a second priority because dropping the burst loss probability is the main goal to achieve. Because of the possible more visited hop number and larger link distances, propagation delay achieved by using our proposed approach is slightly larger than the usual value. This difference is averagely 1ms for the traffic load between 1-1.6 Erlang as seen in Figure 4-15. There are two reasons for this. Firstly, the number of packets that are affected from this rerouting process is limited. For every trigger coming from the feedback packets, it cannot be said that the route definitely will be changed. In addition to this, in our 14-node NSF backbone network, the lengths of alternative routes are also very close to the length of the routes determined without using our feedback approach. For example a packet generated at Core1 and destined to Core11, can follow three routes 1-8-9-12-11, 1-8-9-14-11 or 1-2-4-11 which has a propagation delay of 24ms, 24ms and 22.5ms respectively.

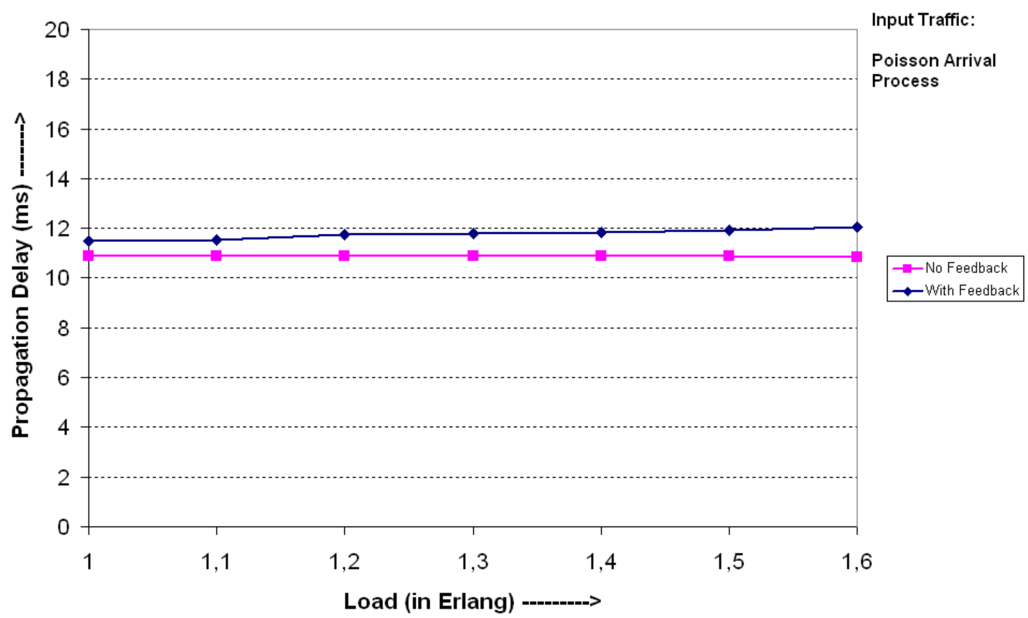


Figure 4-15 Propagation Delay versus Load

4.5.2.2 END TO END DELAY TIME ANALYSIS

We investigated the main delay parameters that constitute the total end-to-end delay, T_{JET} incurred by the signaling technique, JET. T_{JET} is the duration from the instant the first packet arrives at the ingress node to the instant the burst is completely received at the destination and the connection is completely released. Consider a route R_{sd} with h hops to the destination.

In JET, the end-to-end delay is given by the sum of the burst aggregation time, the offset time, the burst transmission time, and the data burst propagation time [39].

$$T_{JET} = t_{agg} + t_{ot} + t_b + \sum_{l^{ij} \in R_{sd}}^{h} t_p^{ij} \tag{4-7}$$

In Figure 4-16, we can see the sum of all previously calculated delay parameters with or without our proposed feedback approach. t_{ot} and t_b values are not affected from the traffic load values or congestion avoidance algorithms. The only parameter that causes difference between these two values is the propagation delay time, which is explained in the previous section.

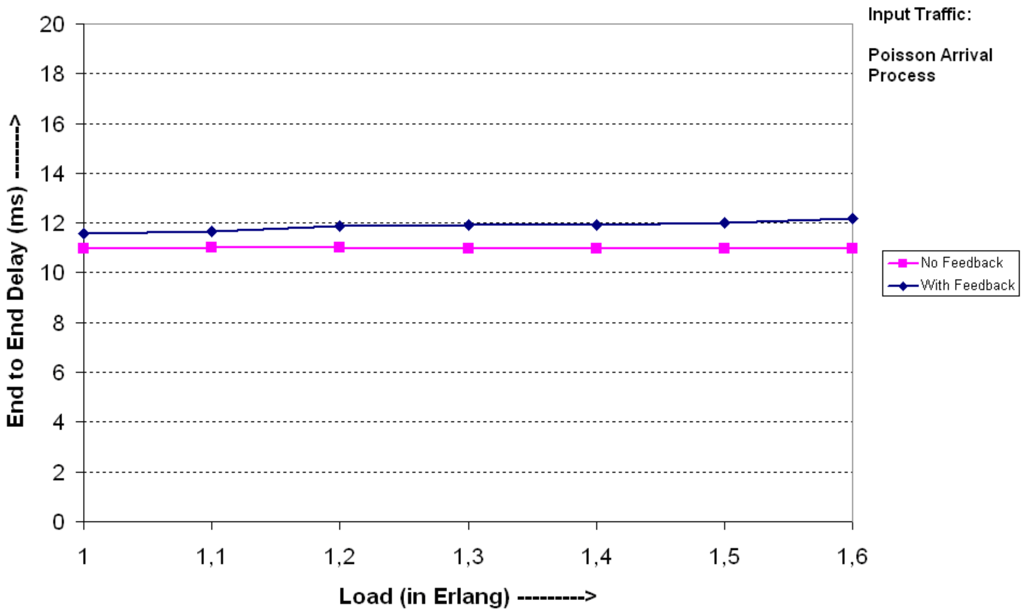


Figure 4-16 End to End Delay versus Load

4.5.3 BURST LOSS PROBABILITY

Reducing the burst loss probability on the highly congested links due to the bursty traffic is our main objective in this thesis. In our simulations, in order to determine the gain achieved with the introduction of our new feedback-based contention avoidance algorithm: Dynamic Rerouting based on Weighted Dijkstra Algorithm, we created "arrived packets" and "discarded packets" counters that are updated in the core nodes. With these parameters, we can calculate the burst loss probability of every link attached to our core nodes or the burst loss probability of every core node and the total network topology. In the simulations that are implemented to investigate the success rate of our new algorithm, all paths no matter their hop distances, are taken into consideration. Hence, burst loss probabilities in Figure 4-17 are far lower than the probabilities in Figure 4-14. This is an expected situation, because if we deal only with the longer routes, packets are more likely to be discarded due to their long journey to the destination.

$$BurstLossRate = \frac{discardedpackets}{discardedpackets + arrivedpackets} * 100 \qquad (4\text{-}8)$$

In Figure 4-17, we can see that an average %1.7 percentage of enhancement is achieved over %11-16 percentage burst loss probability between traffic loads of 1-1.6 E.

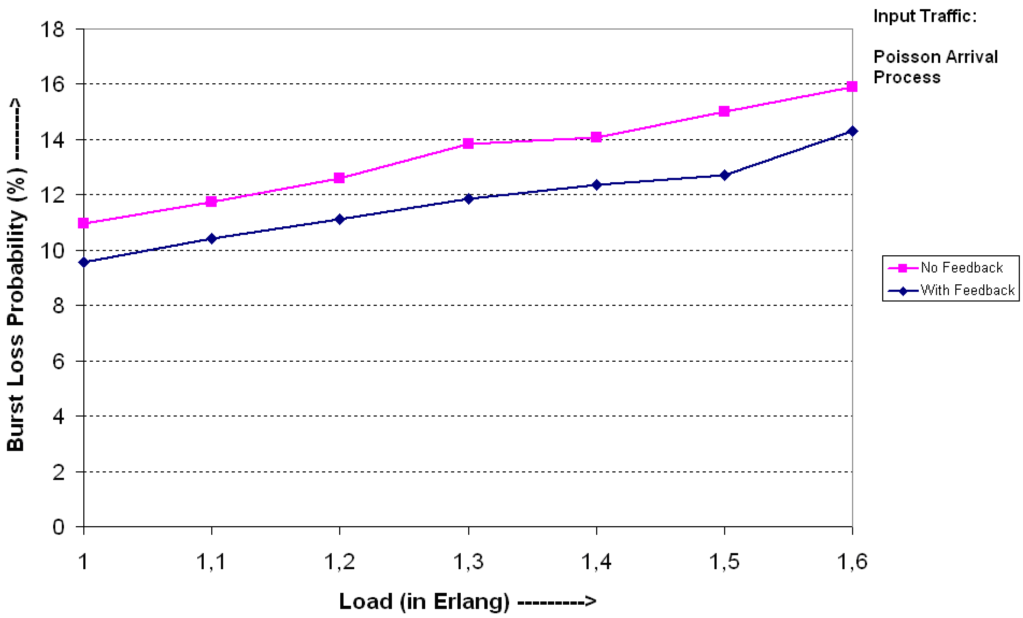


Figure 4-17 Burst Loss Probability versus Load

When we inspect the data collected from the network between 1-1.6E traffic (Table 4-8), although there seems to be a minor improvement on the burst loss probability, we achieved an enhancement of %10-%15 on the saved discarded packet probability (Figure 4-18). Saved discarded packet probability is defined as

$$SavedDisc.PacketRate = 100 * (\frac{disc.packets(nofeed) - disc.packets(withfeed)}{disc.packets(nofeed)}) \qquad (4\text{-}9)$$

Table 4-8 Datas collected from the network between 1.5E Traffic Load

Load (E)		# of Arrived Packets	# Discarded Packets	Burst Loss Probability (%)	Saved Discarded Packet Prob. (%)
1	No Feedback	7010	862	10,9502	12,64501
	With Feedback	7106	753	9,581372	
1,1	No Feedback	7640	1016	11,73752	11,41732
	With Feedback	7748	900	10,40703	
1,2	No Feedback	8282	1192	12,5818	11,91275
	With Feedback	8407	1050	11,10289	
1,3	No Feedback	8852	1422	13,84076	14,48664
	With Feedback	9034	1216	11,86341	
1,4	No Feedback	9510	1558	14,07662	12,45186
	With Feedback	9685	1364	12,34501	
1,5	No Feedback	10107	1786	15,01724	15,56551
	With Feedback	10366	1508	12,70002	
1,6	No Feedback	10723	2025	15,88484	10,2716
	With Feedback	10900	1817	14,28796	

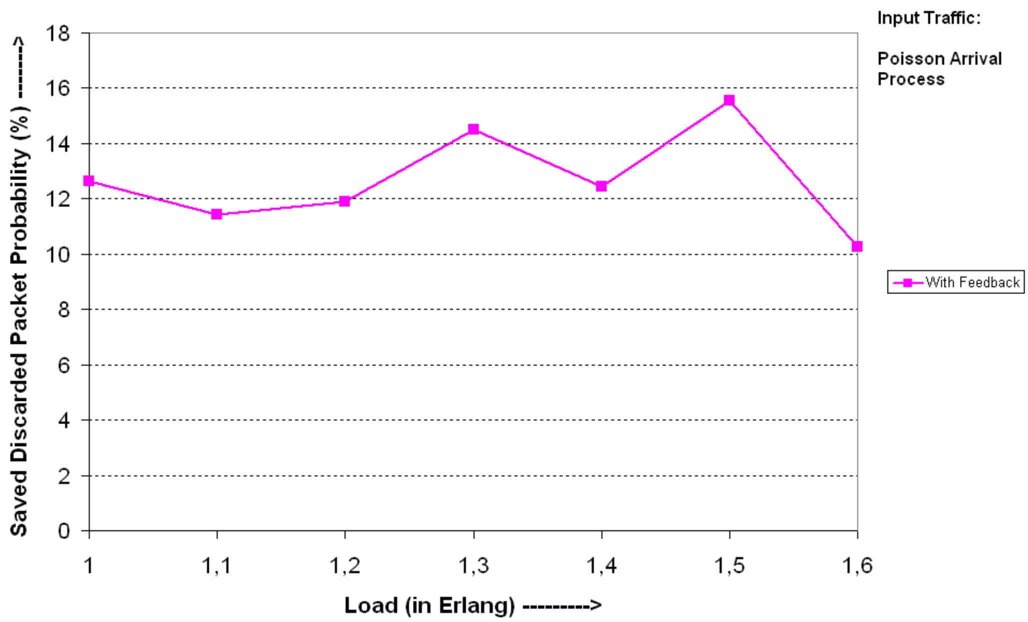


Figure 4-18 Saved Discarded Packet Probability versus Load

To have a better understanding of how every node is affected with our proposed algorithm, Table 4-9 shows the number of arrived and discarded packets and the drop rate of every node in 1.5 Traffic Load. We can instantly notice from Figure 4-19 that, the fluctuation caused from the inhomogeneous distribution of traffic is suppressed efficiently with the introduction of our proposed feedback algorithm. Without using feedback algorithms, routers drop data bursts with rates ranging between %5 and %26. This variance between the core nodes of the network implies that traffic load in the optical links between these nodes are highly unbalanced. This is one of the main causes of the high burst drop probabilities with bursty traffic.

However, with the introduction of our proposed feedback algorithm, routers drop data bursts with rates ranging between %9 and %18. The difference between the highest and lowest drop rates is decreased nearly up to %60.

Table 4-9 Data collected from every core node in 1.5E Traffic Load

	Without Feedback Packets			With Feedback Packets		
Node Name	**# of Arrived Packets**	**# of Discarded Packets**	**Router Drop Rate**	**# of Arrived Packets**	**# of Discarded Packets**	**Router Drop Rate**
Core1	721	40	5,25624	734	74	9,15841
Core2	717	139	16,23831	744	101	11,95266
Core3	773	59	7,09134	769	80	9,42285
Core4	689	156	18,46153	695	130	15,75758
Core5	681	162	19,21708	722	136	15,85082
Core6	816	78	8,72483	799	124	13,43445
Core7	710	165	18,85714	740	89	10,73583
Core8	702	222	24,02597	759	130	14,62317
Core9	704	248	26,05042	744	169	18,51041
Core10	733	69	8,60349	735	78	9,59409
Core11	711	104	12,76073	742	97	11,56138
Core12	710	202	22,14912	728	88	10,78431
Core13	706	111	13,58629	746	120	13,85681
Core14	734	31	4,05228	709	92	11,48564

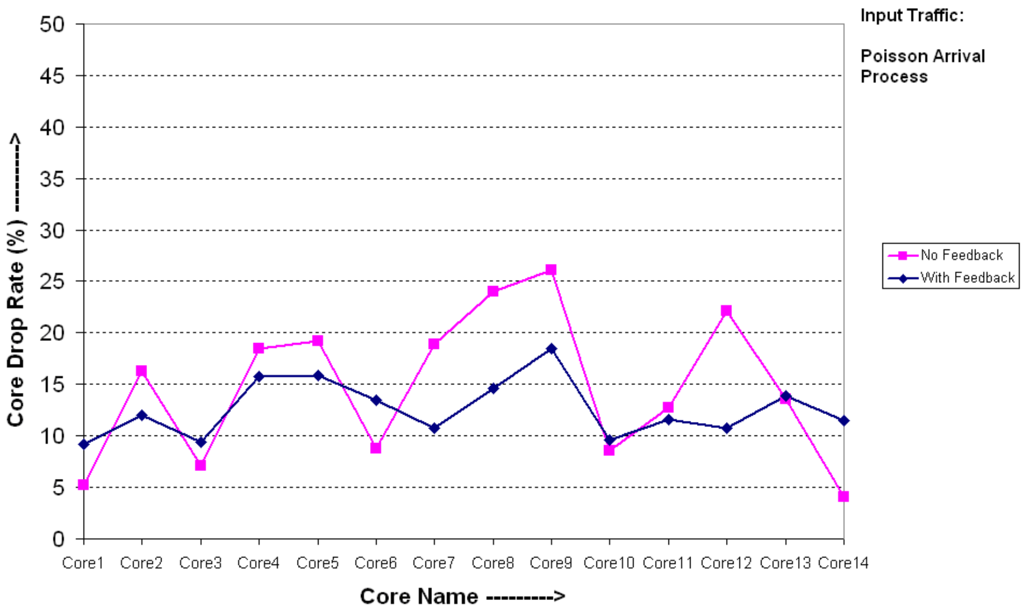


Figure 4-19 Core Drop Rate versus Core Name

When we investigate our network with various data burst lengths as seen in Figure 4-20, we observe that as the length of data burst increases, burst loss probability will increase proportionally. This is an expected result, hence longer data bursts will need longer transmission time. These bursts will occupy more space on the output links of the core nodes which leads more contention problems during the scheduling process of the data bursts. With our proposed feedback algorithm, we achieve an average %2 enhancement on the networks with data bursts whose lengths are varying between 1000k-2500k bits under one Erlang traffic load.

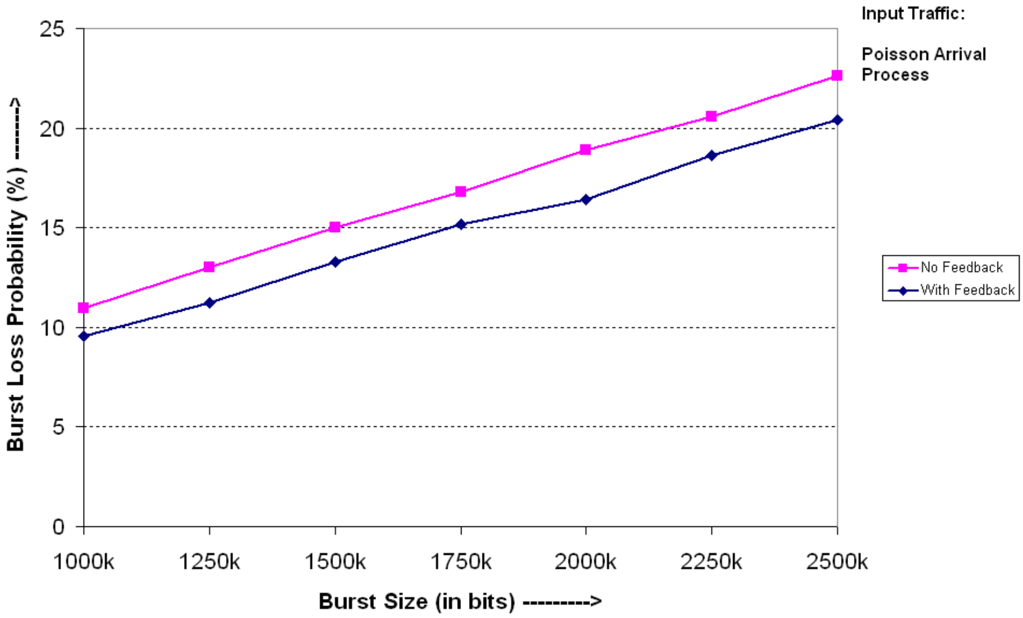


Figure 4-20 Burst Loss Probability versus Burst Size

A similar simulation can be implemented with varying link transmission rates. As the link transmission rate increases, the transmission time needed for a data burst on the optical links will be smaller. These bursts which are on the output links with higher transmission rates will occupy less space which leads less contention problems during the scheduling process of the data bursts. It is clearly concluded from Figure 4-21 that with increasing link transmission rates, the burst loss probabilities decreases significantly. A similar decline is experienced on the success of our proposed feedback algorithm due to the decrease of the discarded packets.

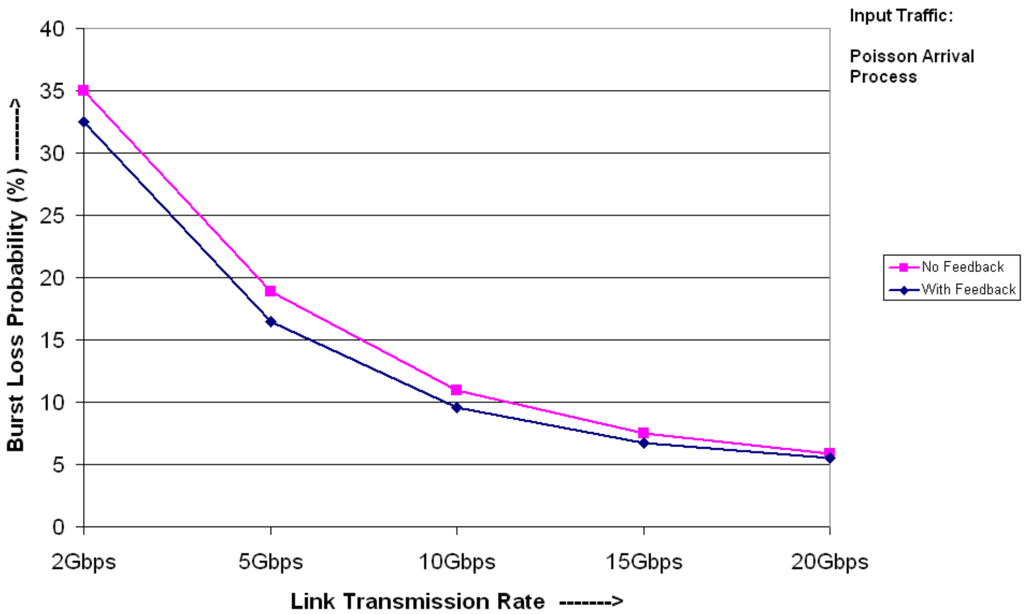


Figure 4-21 Burst Loss Probability versus Link Transmission Rate

We implemented further simulations to see if our proposed algorithm provides a similar enhancement on the networks that have a non-Poisson traffic models. As mentioned in 4.3.5 INPUT TRAFFIC ANALYSIS, Bernoulli arrival traffic model is simulated and the simulation results are collected as seen in Figure 4-22. With 800ms simulation time, time slot is selected as 1 ms, which implies 800 Bernoulli trials for each of the burst generator in our network. It is observed from Figure 4-22 that with increasing arrival probability (p), our proposed algorithm achieves better results and in a certain success state (p=1), the enhancement reaches around %4.

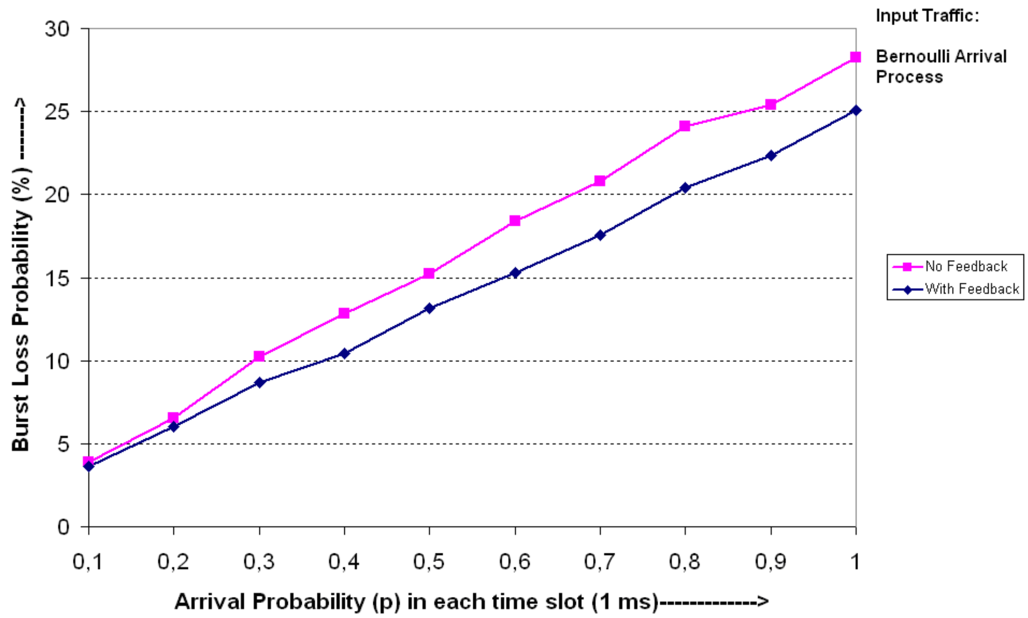


Figure 4-22 Burst Loss Probability versus Arrival Probability (p)

On-Off source model is also simulated by keeping either the probability of terminating a burst (*p*) or the probability of starting a burst (*q*) as constant values. In Figure 4-23 with 800ms simulation time, time slot is selected as 80µs and *p* is assigned as 0.8 implying a constant mean burst length of 0.1ms as in the previous simulations. In Figure 4-24 with 800ms simulation time, time slot is selected as 100µs and *q* is assigned as 0.1 implying a constant mean idle period length of 1ms. Mean interarrival time is assumed to be the sum of mean burst length and the mean idle period length. It is observed from Figure 4-23 that as the probability of starting a new burst increases, the mean idle period length decreases and this leads to the increase of the burst loss probability proportionally. In this situation, our proposed algorithm achieves an enhancement of average %2. When we investigate Figure 4-24, we conclude that as the probability of terminating a new burst increases, the mean burst size decreases and this leads to the decrease of the burst loss probability proportionally. In this situation, our proposed algorithm achieves an enhancement of average %2.5.

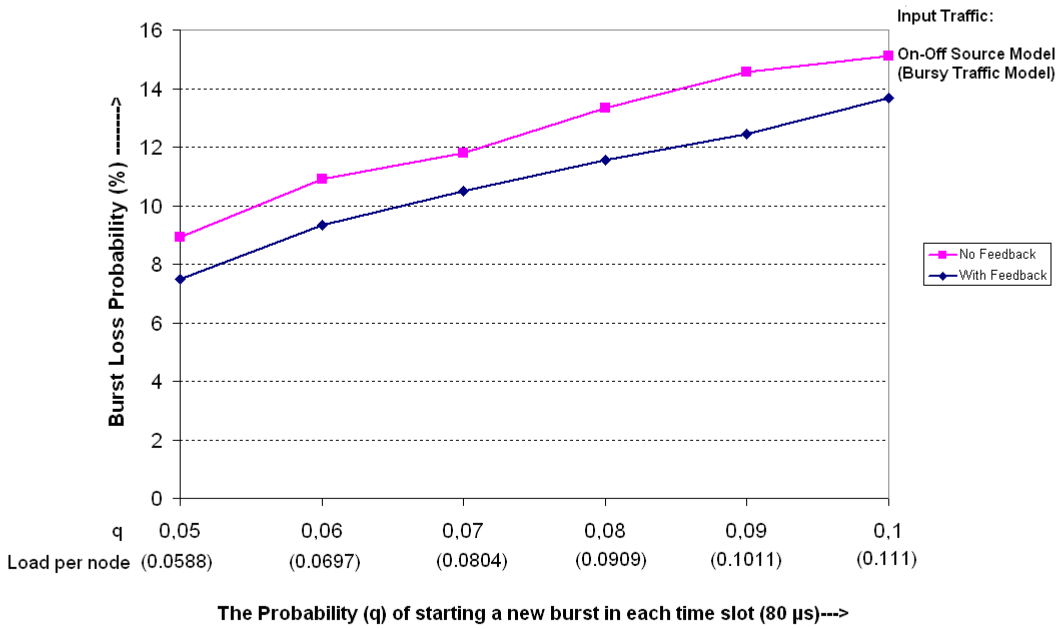


Figure 4-23 Burst Loss Probability versus Probability of starting new burst

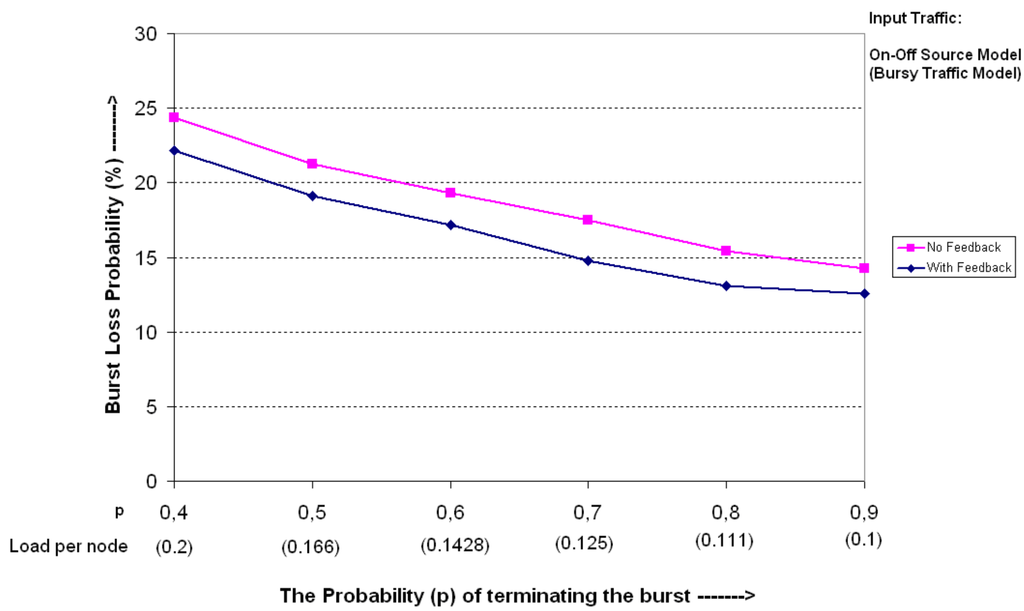


Figure 4-24 Burst Loss Probability versus Probability of terminating burst

4.5.4 CONCLUDING REMARKS

When we evaluate pros and cons of our algorithm, we can see that there are several advantages of our proposed feedback algorithm.

First of all we clearly see from the simulation results that our algorithm succeeds in rerouting the data bursts to the less congested links and reduce the high load on the links that discard packets intensively. While achieving this, data bursts suffer from a narrow increase of the end-to-end delays, which can be evaluated as a fine result.

In order to achieve better burst drop probabilities and a minor increase on the end-to-end delays of the data burst, the only extra overhead we introduce is the negative feedback packets. In our algorithm, we aim to benefit from our sole overhead, negative feedback packet as much as possible. From the generation of the feedback packet until its deletion in the source node, the main objective of the feedback packet is to get as much congestion information as from the visited core nodes and to share the collected information with each of them efficiently.

There are some disadvantages of the proposed algorithm, which we should not ignore. A potential disadvantage of the algorithm is that the rerouting the data bursts to the less congested links can possibly cause congestion on a different link and a network instability problem can be experienced.

While feedback packets are traversing on the core nodes, the congestion information extracted from the feedback packets can cause the initiation of a rerouting process as explained in Section 4.2.2. However, each core node calculates the routes in an “isolated” and distributed manner. So the information of a highly congested link can only be known by a limited number of core nodes. These nodes may not choose sending the new bursts through this link; however, other ones can still continue to forward the bursts through that highly congested links. Because of this, our algorithms may not work as efficient as it is presumed.

CHAPTER 5

CONCLUSION

OBS is a promising bufferless dense wavelength division multiplexing (DWDM) switching technology, which has a big potential to exploit the bandwidth provided by DWDM. OBS combines the advantages of packet switching and circuit switching in a single network and it is more bandwidth efficient for bursty traffic due to its nature of statistical multiplexing. However high congestion on the optical links of an OBS network is a crucial challenge and the high burst loss probability in the networks must be reduced to optimum levels without losing the genuine superiorities of the technology.

In this thesis, we proposed a feedback-based contention avoidance algorithm via dynamic rerouting based on weighted Dijkstra algorithm for optical burst switching networks. This algorithm is focused on decreasing the burst loss probability and keeping the end-to-end delay value in an acceptable region. To achieve this goal under bursty traffic, this algorithm aims to detect highly congested links on the network quickly and reroute new bursts from these links to uncongested, idle links efficiently.

First of all, a previous study using JET signaling technique in OBS networks is implemented in OMNET simulation environment in order to investigate the performance of our designed OBS simulation environment.

The results of the performance analysis for JET signaling technique in OBS networks show that high burst loss probability is a critical problem especially for the networks with bursty traffic and for the packets that has to traverse high number of intermediate nodes en route to destination. In comparison to the results presented in the paper studying JET algorithm, we nearly acquired the same results with our designed simulation environment.

Then the simulations for our proposed feedback-based contention avoidance algorithm are implemented on our verified simulation environment. The results of the simulations with a Poisson traffic model are evaluated according to burst loss probability and delay parameters such as propagation delay and end-to-end delay. It is seen that propagation delay and the total end-to-end delay values are increased slightly with the introduction of our proposed feedback algorithm. This slight increase is conceivable because some of the shortest paths that are calculated statically in the initialization process of the network can be rerouted dynamically in a congestion state.

The performance of the proposed algorithm in terms of burst loss probability is evaluated by simulation using three different burst arrival processes, namely Poisson, Bernoulli and On-Off arrival traffic models. The simulation results show that the proposed feedback-based algorithm has good performance characteristics, effectively distributes the traffic of bursts, and significantly reduces the burst loss probability as compared to networks without any contention avoidance techniques. In addition to this, the illustrative numerical results demonstrate that with increasing data burst size, burst loss probability increases proportionally, while with increasing link transmission rates, the burst loss probabilities decreases significantly. In the simulations with various burst sizes and link transmission rates, our proposed algorithm improves the network in a substantial amount. To sum up, we conclude that the proposed approach is a viable scheme to improve utilization for the OBS networks.

There are several future work areas that can be further investigated. One area of future work will be to regulate the rate of the data bursts entering the OBS network via traffic shaping method at the source nodes using the congestion information coming from the feedback packets. Another area of future work will be to extend the proposed feedback-based contention avoidance algorithm such that it can support service differentiation and QoS.

This thesis work can be further extended with the addition of burst assembly algorithm at the ingress routers, which is an interesting area of future work, wherein the burst length can be varied based on a timer or a length threshold to shape the traffic into the network.

The stability of our proposed algorithm should be investigated in order to make the algorithm more robust since it is based on dynamic rerouting of the bursts.

REFERENCES

[1] F. C. Qiao and M. Yoo, “Optical Burst Switching (OBS) - A New Paradigm for an Optical Internet,” *Journal of High Speed Networks*, vol. 8, no.1, pp. 69-84, Jan. 1999.

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

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

[4] C. Gauger, “Dimensioning of FDL Buffers for Optical Burst Switching Nodes,” *Proceedings, Optical Network Design and Modeling (ONDM)* 2002, Torino, Italy, 2002.

[5] R. A. Barry, P. Humblet, “Models of blocking probability in all optical networks with and without wavelength changers,” *IEEE J. Select. Areas Commun.,* vol. 14, pp. 852-857, June 1996.

[6] V. M. Vokkarane, J. P. Jue, and S. Sitaraman, “Burst Segmentation: An Approach for Reducing Packet Loss In Optical Burst Switched,” *Proceedings, IEEE International Conference on Communications (ICC)* 2002, New York, NY, vol. 5, pp. 2673-2677, April 2002.

[7] F. Farahmand, J. P. Jue, “Supporting QoS with look-ahead window contention resolution in optical burst switched networks,” *Proceedings, IEEE GLOBECOM* 2003, San Francisco, CA, USA, vol. 5, pp. 2699–2703, Dec. 2003.

[8] F. Farahmand, V. M. Vokkarane, J. P. Jue, "Practical priority contention resolution for slotted optical burst switching networks," *Proceedings, IEEE/SPIE First International Workshop on Optical Burst Switching* 2003, Dallas, TX, Oct. 16, 2003.

[9] S. Y. Wang, "Using TCP congestion control to improve the performances of optical burst switched networks", *Proc. IEEE ICC'03*, vol. 2, pp. 1438–1442, May 2003.

[10] G. P. Thodime, V. M. Vokkarane, J. P. Jue, "Dynamic congestion based load balanced routing in optical burst-switched networks," *IEEE GLOBECOM*, vol. 5, San Francisco, CA, pp. 2694–2698, December 2003.

[11] A. Maach, G. von Bochmann, H. T. Mouftah, "Congestion control and contention elimination in optical burst switching," *Telecommunication Systems*, vol.27, no.2–4, pp.115–131, Oct.–Dec. 2004.

[12] F. Farahmand, Q. Zhang, J. P. Jue, "A closed-loop rate based contention control for optical burst switched networks," *Proceedings, IEEE GLOBECOM,* St. Louis, Missouri, USA, pp.1989–1993, Nov.2005.

[13] E. W. Van Stryland, and M. Bass, *Fiber Optics Handbook: Fiber, Devices, and Systems for Optical Communications,* McGraw-Hill, New York, 2002.

[14] K. Thyagarajan, and A. Ghatak, *Fiber Optic Essentials,* Wiley-IEEE Press, 2007.

[15] D. Bailey, and E. Wright, *Practical Fiber Optics*, Elsevier, 2003.

[16] D. Cavendish, "Evolution of Optical Transport Technologies: From Sonet/SDH to WDM," *IEEE Comm.,* vol. 38, no. 6, pp. 164-172, June 2000.

[17] J. P. Jue and V. M. Vokkarane, *Optical burst switched networks*, Springer, Optical Networks Series, 2005.

[18] Y. Chen, C. Qiao, and X. Yu, "Optical burst switching (OBS): A new area in optical networking research," *IEEE Network Mag.*, vol. 18, no. 3, pp. 16-23, May-Jun. 2004.

[19] V. R. Jonnadula, "Performance Analysis of Congestion Control Schemes in OBS Mesh Networks," *MSc. Thesis*, Available: http://www4.ncsu.edu/~hp/VenkatThesis.pdf, (Last accessed 26 October 2008), NCSU, 2004.

[20] L. Battestilli, Harry G. Perros, "A performance study of an optical burst switched network with dynamic simultaneous link possession," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol.50 no.2, pp.219-236, 08 February 2006.

[21] V. Puttasubbappa, "Optical Burst Switching: Challenges, Solutions and Performance Evaluation," *PhD thesis*, Available: http://www4.ncsu.edu/~hp/VishwasThesis.pdf, (Last accessed 15 November 2008), NCSU, May 2005.

[22] A. Shtainhart, R. Segal, A. Tsherniak, "Wavelength Division Multiplexing", from http://www2.rad.com/networks/1999/wdm/wdm.htm, (Last accessed 21 November 2008).

[23] X. Huang, F. Farahmand, and J. P. Jue, "An Algorithm for Traffic Grooming in WDM Mesh Networks with Dynamically Changing Light-Trees," *Proceedings of the IEEE GLOBECOM'04*, Dallas, USA, pp. 1813-1817, 29 November to 3 December 2004.

[24] G. K. Chang, J. Yu, Y. K. Yeo, A. Chowdhury, and Z. Jia, "Enabling technologies for next-generation optical packet-switching networks," *Proceedings of the IEEE*, vol. 94, no. 5, pp. 892 – 910, May 2006.

[25] A. Abid, F. M. Abbou and H. T.Ewe, "Effective Implementation of Burst Segmentation Techniques in OBS Networks," *International Journal of Information Technology (IJIT)***,** vol. 3, no. 4, pp. 231-236, 2006.

[26] Y. Xiong, M. Vandenhoute, and H. Cankaya, "Control architecture in optical burst-switched WDM networks," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 1838 –1851, October 2000.

[27] M. Yoo and C. Qiao, "Just-enough-time (JET): a high speed protocol for bursty traffic in optical networks," *IEEE/LEOS Conf. on Technologies For a Global Information Infrastructure*, pp. 26-27, Aug. 1997.

[28] J. Xu, C. Qiao, J. Li, and G. Xu, "Efficient channel scheduling algorithms in optical burst switched networks," *Proc. IEEE Information Communications (INFOCOM)* San Francisco, CA, vol. 3, pp. 2268-2278, Mar. 2003.

[29] A. Detti, M. Listanti, "Application of tell and go and tell and wait reservation strategies in a optical burst switching network: A performance comparison," *Proceedings of IEEE International Conference on Telecommunication (ICT)*, vol. 2, Bucharest, Romania, pp. 540-548, June 4-7, 2001.

[30] J. Y. Wei and R. I. McFarland, "Just-in-time signaling for WDM optical burst switching networks (Invited Paper)," *J. Lightwave Technol., Special Issue on Optical Networks*, vol. 18, pp. 2019-2037, Dec. 2000.

[31] D. K. Hunter, M. C. Chia, and I. Andonovic, "Buffering in optical packet switches," *IEEE/OSA Journal of Lightwave Technology*, vol. 16 no. 12, pp. 2081-2094, December 1998.

[32] I. Chlamtac, A. Fumagalli, L. G. Kazovsky, and et al, "CORD: Contention resolution by delay lines," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 1014-1029, June 1996.

[33] Z. Haas, “The 'Staggering Switch': An electronically controlled optical packet switch,” *IEEE/OSA Journal of Lightwave Technology*, vol. 11 no. 5/6, pp. 925-936, May/June 1993.

[34] R. Ramaswami and K. N. Sivarajan, “Routing and wavelength assignment in all-optical networks,” *IEEE/ACM Transactions on Networking*, vol. 3, no. 5, pp. 489-500, October 1995.

[35] F. Farahmand, Q. Zhang, J. P. Jue, “A feedback-based contention avoidance mechanism for optical burst switching networks,” *Proc. 3rd International Workshop on Optical Burst Switching*, Oct. 2004.

[36] M. Elhaddad, R. Melhem, T. Znati, and D. Basak, “Traffic shaping and scheduling for OBS-based IP/WDM Backbones,” *Proceedings, SPIE Optical Networking and Communication Conference (OptiComm)* 2003, Dallas, TX, vol. 5285, pp. 336-345, October 2003.

[37] R. Jain and K. K. Ramakrishnan, “Congestion avoidance in computer networks with a connectionless network layer: concepts, goals, and methodology,” *Proceedings, Computer Networking Symposium* 1988, pp. 134 -143, 11-13 April 1988.

[38] J. Li, G. Mohan, and K. Chua, “Dynamic load balancing in IP-over-WDM optical burst switching networks,” *Computer Networks*, vol. 47, no. 3, pp. 393-408, 2005.

[39] Vinod M. Vokkarane, “Intermediate-Node-Initiation (INI): A Generalized Signaling Framework for Optical Burst-Switched Networks” *Elsevier Optical Switching and Networking (OSN)*, vol. 4, no. 1, pp. 20-32, Feb. 2007.

[40] K. C. Claffy, H.-W. Braun, and G. C. Polyzos, “Tracking Long-term Growth of the NSFNET,” *Communications of the ACM*, vol. 37, no. 8, pp. 34-45, August 1994. (Special Issue on Internet Technology.)

[41] A. Varga, “The OMNeT++ Discrete Event Simulation System,” *Proc. of European Simulation Multiconference* (ESM2001), June 2001.

[42] A. Betker, C. Gerlach, R. Hülsermann, M. Jäger, M. Barry, S. Bodamer, J. Späth, C. Gauger, and M. Köhn, "Reference Transport Network Scenarios, Technical report," *BMBF Multi Tera Net*, July 2003.

[43] S. De Maesschalck et al.: "Pan-European Optical Transport Networks: an Availability based Comparison," *Photonic Network Communications*, vol. 5, no. 3, pp. 203-226, May 2003.

[44] R. Hülsermann et al, "A Set of Typical Transport. Network Scenarios for Network Modelling," *5.th. ITG-Workshop on Photonic Networks*, pp. 65-72, 2004.

[45] R. B. Cooper, *Introduction to Queueing Theory*, Second Edition. New York: North Holland, 1981.

[46] I. Angus, "An introduction to Erlang B and Erlang C," *Telemanagement*, No.187, pp.6-8. (2001).

[47] *OMNET++* [Online], Available: http://www.omnetpp.org, (Last accessed 12 November 2008)

[48] Xuan-Hien Dang, Abdullah A. Abonamah, "An approximate analysis of shared-buffer channel-grouped ATM switches under imbalanced traffic," *Int. J. Communication Systems,* vol. 19, no. 5, pp. 567-584, 2006.