ATTACK MITIGATION FOR SECURE OPTICAL BURST SWITCHED NETWORKS

A Thesis

submitted to Pondicherry University in partial fulfillment of the requirements for the award of the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE AND ENGINEERING

by

K. MUTHURAJ

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PONDICHERRY ENGINEERING COLLEGE

PUDUCHERRY - 605 014

INDIA

MARCH 2015

Dr. N. Sreenath ***B.Tech., M.Tech., Ph.D (IIT-M)***
Professor,
Department of Computer Science and Engineering,
Pondicherry Engineering College,
Puducherry - 605 014.

CERTIFICATE

Certified that this thesis entitled **"ATTACK MITIGATION FOR SECURE OPTICAL BURST SWITCHED NETWORKS"** submitted for the award of the degree of **DOCTOR OF PHILOSOPHY** in **COMPUTER SCIENCE AND ENGINEERING** of the Pondicherry Engineering College, Pondicherry University, Puducherry is a record of independent research work done by **Mr. K. MUTHURAJ** during the period of study (November 2008 - January 2014) under my supervision.

Place: Puducherry

Date: 25.03.2015 **(Dr. N. SREENATH)**

Phone: 91-9443289642
Email: nsreenath@pec.edu

DECLARATION

I hereby declare that this thesis entitled **"ATTACK MITIGATION FOR SECURE OPTICAL BURST SWITCHED NETWORKS"** submitted to the Department of Computer science and Engineering, Pondicherry Engineering College, Pondicherry University, Puducherry, India for the award of the degree of **DOCTOR OF PHILOSOPHY** in **COMPUTER SCIENCE AND ENGINEERING** is a record of bonafide research work carried out by me under the supervision of **Dr. N. SREENATH**, Professor, Department of Computer Science and Engineering, Pondicherry Engineering College, Puducherry and has not formed the basis for the award of any other degree by any university/institution before.

(K. MUTHURAJ)

Place: Puducherry

Date: 25.03.2015

ABSTRACT

Optical networks are considered as the potential candidates for providing high bandwidth requirement and high transmission rate beyond electronic router's capability. Since, they combine the optimistic optical packet switching and wavelength routing together and an optical burst switched network possesses many limitations or vulnerabilities that are quite natural to sustain in case of the security attacks.. In addition, OBS achieves high traffic throughput and high resource utilization by aggregating multiple packets like IP Packets, Ethernet frames or ATM Cells into a single burst which is consider as the primary data transmission unit in an OBS network. Further, within a pre-determined offset time, the burst is always transmitted without waiting for an acknowledgement from the receiver and also it configures every intermediate node before the arrival of actual burst which is highly venerable factor. Furthermore, it is also found that there is only a limited amount of work base been carried out towards the provision of security in the optical switched networks.

Hence, this research has been proposed to tackle the security issues both at the OBS layer and physical layer. It is also found that the attacks like burst hijacking attack, fake spectral attack, burst flooding attack, timeout attack, burst circulating attack and land attack are investigated based on ns2 simulator with the modified nOBS patch to quantify the impact and the effectiveness of the proposed solutions. The statistical approach for detecting and preventing attacks are also proposed for the normal scenario, attack scenario and attack solution scenario.

In addition, the mitigation mechanisms proposed for each of the potential attacks infers that the Cornbach Alpha Reliability Coefficient Based Mitigation mechanism for Burst Hijacking attack in an average decreases the burst block probability by 23%, the variations alpha coefficient approach of Fake Spectral Attack mitigation in an average decreases the burst block probability by 18%, the Kappa Coefficient based mitigation model for burst Flooding attack decreases the Burst block probability by 34%, the Optimal Time Threshold based mitigation approach for Timeout Attack decreases the Burst block probability by 21%, the contention detection based mathematical model for burst circulating attack decreases the burst loss probability by 16% and the Stratified Alpha Reliability Coefficient based Mitigation mechanism for Land attack decreases the burst block probability by 24%.

The possible detection and countermeasures for each and every attack are also dealt with the mathematical model for reliable identification and isolation. Finally, the results can be enhanced through Real Test bed simulation, which is not done for the work due to unavailability of optical nodes and links and more statistical mathematical mitigation mechanism may be formulated and analyzed for the identified potential attacks.

ACKNOWLEDGEMENTS

This is certainly the section of the thesis that I enjoyed writing the most. First and foremost I would like to thank my supervisor Professor **Dr. N. Sreenath** for giving me the privilege of being one of his Ph.D students and for his constant advice and support throughout this enduring period of my life. Thank you for believing in me and basically for changing my life by accepting me for this research adventure.

My eternal gratitude goes to **Dr. P. Dananjayan** and **Dr. G. Aghila,** dissertation committee members generously offering their time, support, guidance and good will throughout the research work. The good advice, support and kindness of **Dr. V. Prithiviraj**, **Dr. D. Govindarajulu, Dr. K. Vivekanandan** and **Dr. D. Loganathan** have been invaluable on the evolution of this thesis for which I am extremely grateful.

I am most grateful to Pondicherry Engineering College for giving me the opportunity to research in the Information Security Lab and CASIT Lab while earning my Ph.D and I would like to thank **Dr. V. Kamakoti** for providing the research assistance through the ISEA project grants.

I would like to thank all the past and present members of the Lab for all their infinite support, friendship and for the FUN (!) particularly A. Amuthan, P. Boobalan, S. Mourouga Pragash, N. Sivakumar, M. Thirumaran and V. Govindasamy.

I wish to express my thanks to Teaching, Non-Teaching and other Supporting Staff of the Department of Computer Science and Engineering, Pondicherry Engineering College for their help during the research work.

These six years in Pondicherry gave me the opportunity to meet unforgettable marvelous people that are impossible to list and this thesis is dedicated to all of them. A special dedication goes to my best friends Anbumani, Rajavalu, Thirumalaivasan, Mahesh, Robinson, Marimuthu, Sivasubramaniyan, Prasad, Vinoth, Ramkumar, Brabagaran, Sengathir and Abdulrahman. Thank you for your precious friendship and the infinite indelible great memories.

Last but not the least, I would like to thank the most important people in my life without whom it would not have been possible to write this doctoral thesis. My Eternal Gratitude goes to my parents G. Kaluvan and K. Mahalakshmi and my father in law Mr. A.M. Mohan and Mother in law Mrs. A.M. Geetha to my sisters R. Amutha and V. Geetha and Brother in laws Mr. P. Rengaraj, Mr. A. Vijay and Mr. A.M. Thiyagarajan and childrens R. RaGav, V. Hasini, R. Harish and V. Janavi. And my behalf Hemalatha @ Papu and my daughter M. sanJana for their infinite patience support and love.

Above all, I owe the credit to the Almighty for blessing, wisdom and everything

MUTHURAJ KALUVAN

TABLE OF CONTENTS

CHAPTER **PAGE**

APPENDIX

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVATIONS

AON	All Optical Network
ATM	Asynchronous Transfer Mode
BD	Burst Data
BEB	Best Effort Bursts
BER	Bit Error Rate
BCH	Burst Control Header
BHP	Burst Header Packet
BS	Branching Switch
CBQ	Class Based Queuing
CC	Candidate Connector
CD	Candidate Destination
CDN	Content Distribution Network
CONUS	Continental US
DIS	Distributed Interactive Simulations
DoC	Drop or Continue
DoS	Denial of Service
DP	Distance Priority
DSB	Delay Sensitive Bursts
EC	Equal Coverage
FDL	Fiber Delay Line

FFUC	First Fit Unused channel
FFUC-VF	First Fit Unused channel – Void Filling
FTP	File Transfer Protocol
FWC	Fixed Wavelength Conversion
HCF	Hop Count Factor
IP	Internet Protocol
JET	Just Enough Time
JIT	Just In Time
LAUC	Latest Available Used Channel
LAUC-VF	Latest Available Used Channel – Void Filling
LED	Light Emitting Diode
MC	Multicast Capable node
MF	Member First
MI	Multicast Incapable node
MO	Member Only
MSC	Multicast Shared Class
MST	Multicast Shared Tree
NCF	Nearest Connector First
NSFNET	National Science Foundation Network
OBS	Optical Burst Switching
OC	Overlapping Coverage
OCS	Optical Circuit Switching

OEO	Optical- Electronic-optical
OPS	Optical Packet Switching
OT	Offset Time
OTWC	Optical Tunable Wavelength Converter
RED	Random Early Detection
RPB	Reverse Path Broadcast
RPM	Reverse Path Multicasting
RTA	Reroute to Any
RTS	Reroute to Source
SAN	Storage Area Network
SC	Super Coverage
SN	Splitter Node
TAG	Tell And Go
TAW	Tell And Wait
TCD	Time domain Cloning Differentiation
TDM	Time Division Multiplexing
TRPB	Truncated Reverse Path Broadcasting
VINT	Virtual Inter Network Test-bed
VS	Virtual Source
WC	Wavelength Conversion
WDM	Wavelength Division Multiplexing
WR-WDM	Wavelength Routed WDM

CHAPTER 1

INTRODUCTION

1.1 Optical Networking

Optical networks are potential networks developed for meeting the exploding bandwidth requirements of existing and emerging communications applications. These networks have a tremendous bandwidth of around 50 terabits per second. However, the demand for point to point communication per application is not typically enough. Therefore, for utilizing the capabilities of optical networks, the bandwidth of the optical fiber is divided into multiple communication channels. Each channel corresponds to a unique wavelength. In other words, these optical networks employ Wavelength Division Multiplexing (WDM) [1].

The user of an optical network wants the significant data to be sent from a source node to a destination node. These demands must be routed in the most efficient way over the network. First, the router needs to find uncongested paths between the source and destination. Furthermore, in optical networks the router assigns a wavelength for each and every data traveling in a link. This optical path generally known as a lightpath enables the routing and the wavelength assignments on the route and this lightpath is reserved for a session time until the demand is satisfied. When the demand is satisfied, all the assigned wavelengths become available on the lightpath. An example of an optical network is shown in Figure 1.1.

This network typically consists of a collection of edge nodes and core nodes connected to each other using WDM links. In this network, the traffic is often originated at the ingress node and terminated at one or more destination nodes called egress edge nodes. The ingress node receives the incoming IP traffic from multiple client networks, such as SONET, ATM, or Gigabit Ethernet, and transmits it through high capacity DWDM links [2][3][4]. Each core node is connected to one or more edge nodes. Depending on the capability, the core node can either pass the incoming optical signal to the next node or terminate it.

The optical signal eventually gets terminated at the egress edge node after travelling through multiple core nodes. On receiving the data, the egress edge node sends the data to the corresponding client network [5][6].

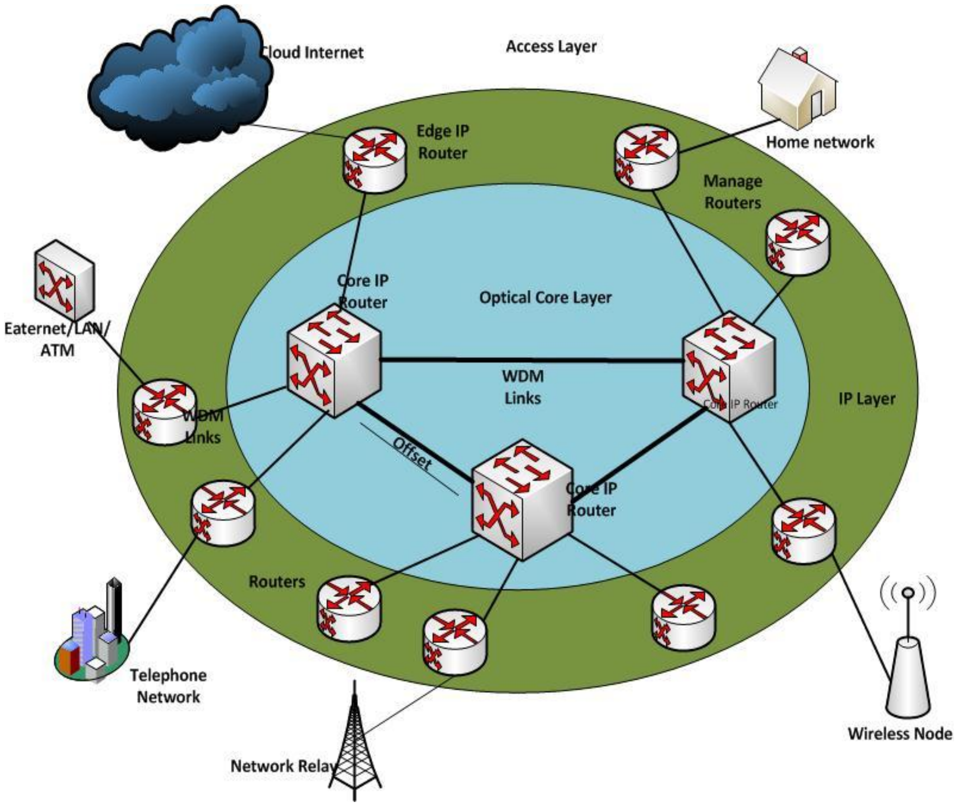


Figure 1.1 An Optical Network

1.2 Multiplexing Technologies

The multiplexing in optical networks is always beneficial since, it is always more economical to transmit data at higher rates over a single fiber than it is to transmit at lower rates over multiple fibers, in most applications. There are fundamentally two ways of increasing the transmission capacity on a fiber, as shown in Figure 1.2. The first is to increase the bit rate. This requires higher-speed electronics. Many lower-speed data streams are multiplexed into a higher-speed stream at the transmission bit rate by means of electronic time division multiplexing (TDM) [7]. The multiplexer typically interleaves the lower-speed streams to obtain the higher-speed stream. For example, it could pick 1 byte of data from the first stream, the next byte from the second stream, and so on.

As an example, 64 155 Mb/s streams may be multiplexed into a single 10 Gb/s stream. Today, the highest transmission rate in commercially available systems is around 10 Gb/s; 40 Gb/s TDM technology will be available soon. To push TDM technology beyond these rates, researchers are working on methods to perform the multiplexing and demultiplexing functions optically. This approach is called optical time division multiplexing (OTDM) [8]. Laboratory experiments have demonstrated the multiplexing/demultiplexing of several 10 Gb/s streams into/from a 250 Gb/s stream, although commercial implementation of OTDM is still several years away.

However, multiplexing and demultiplexing high-speed streams is not sufficient to realize practical networks. The higher the bit rate, the more difficult is to engineer around the impairments. So, we expect the transmission bit rates to monotonically increase. Another way to increase the capacity is through a technique called wavelength division multiplexing (WDM) [9][10].

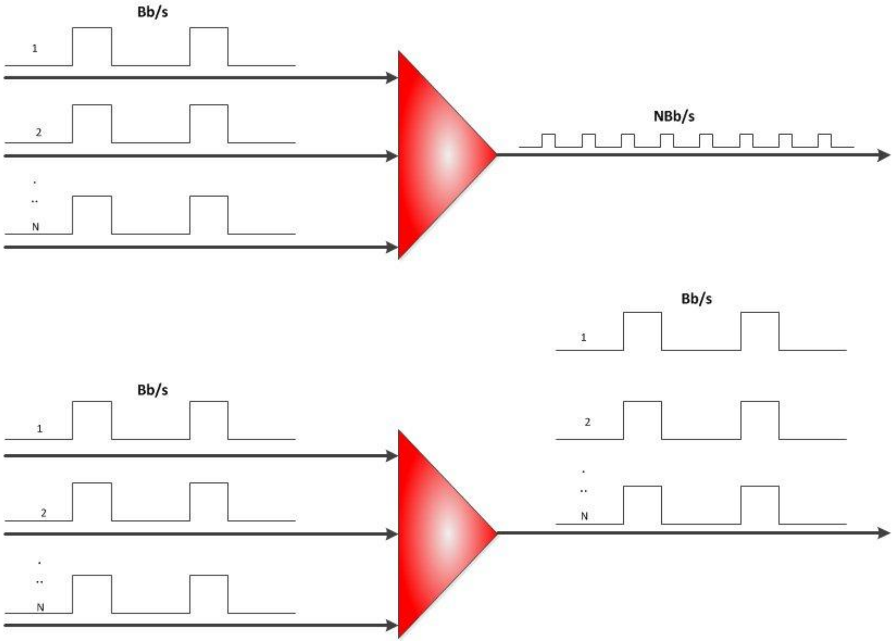


Figure 1.2 TDM / WDM

WDM is essentially the same as frequency division multiplexing (FDM), which is used in radio systems for more than a century. For the same reason, the term FDM is used widely in radio communication, but WDM is used in the context of optical communication, perhaps because FDM was studied first by communications engineers and WDM by physicists.

The idea is to transmit data simultaneously at multiple carrier wavelengths (or, equivalently, frequencies or colors) over a fiber. These wavelengths do not interfere with each other provided they are kept sufficiently far apart. Thus WDM provides a virtual fiber that makes a single fiber look like multiple "virtual" fibers, with each virtual fiber carrying a single data stream. WDM systems are widely deployed today in long-haul and undersea networks. WDM and TDM provide a number of ways to increase the transmission capacity that are complementary to each other. Therefore networks use a combination of TDM and WDM. The question of combining TDM and WDM is an important that one faces in their carriers [11-15]. Using a combination of WDM and TDM, systems with transmission capacities of around 1 Tb/s over a single fiber are becoming commercially available, and no doubt these systems with higher capacities operating over longer distances emerge in the future.

1.3 Wavelength Division Multiplexing

A key technology in developing optical networks is the Wavelength Division Multiplexing (WDM) technology. WDM technology exploits the wide communication bandwidth in optical fiber that enables each fiber to carry multiple optical signals, each at a different wavelength. In this way, WDM transforms a fiber into multiple virtual fibers. Using WDM technology, it is possible to maintain low bit rate and multiply the number of wavelengths. This approach is particularly attractive to overcome technological challenges currently confronting 40 Gbps TDM systems. Implementing WDM systems also results in reducing the number of required regenerators and hence, dramatically lowers the cost. As an example, consider transmitting a 40 Gbps signal over 600 km.

Using a traditional system, optical networking requires 16 separate fiber pairs with regenerators placed every 35 km for a total of 272 regenerators. A 16 channel WDM system, in which each wavelength transmits at a rate of 2.5 Gbps, on the other hand, uses a single fiber pair and 4 amplifiers positioned every 120 km for a total of 600 km [16]. A unique property of a WDM optical network is the ability to do wavelength routing. Here, the path of the signal through the network is determined by the wavelength and origin of the signal, as well as the states of the network switches and wavelength changers.

Unlike other optical approaches, wavelength routing provides a transparent lightpath between network terminals. A lightpath is a path an optical signal traverses in the network from a source to a single destination which may include optical wavelength changers. Similarly, a wavelength path is the lightpath without wavelength changers. This transparency provides a simple way for heterogeneous users to share network resources [17][18].

For example, certain wavelengths could carry analog signals with other wavelengths simultaneously being used for digital. Moreover, different network terminals may use different modulation formats and terminals may be upgraded without any network reconfiguration. As a philosophy, the network will provide bandwidth on demand and let the users determine their individual hardware requirements.

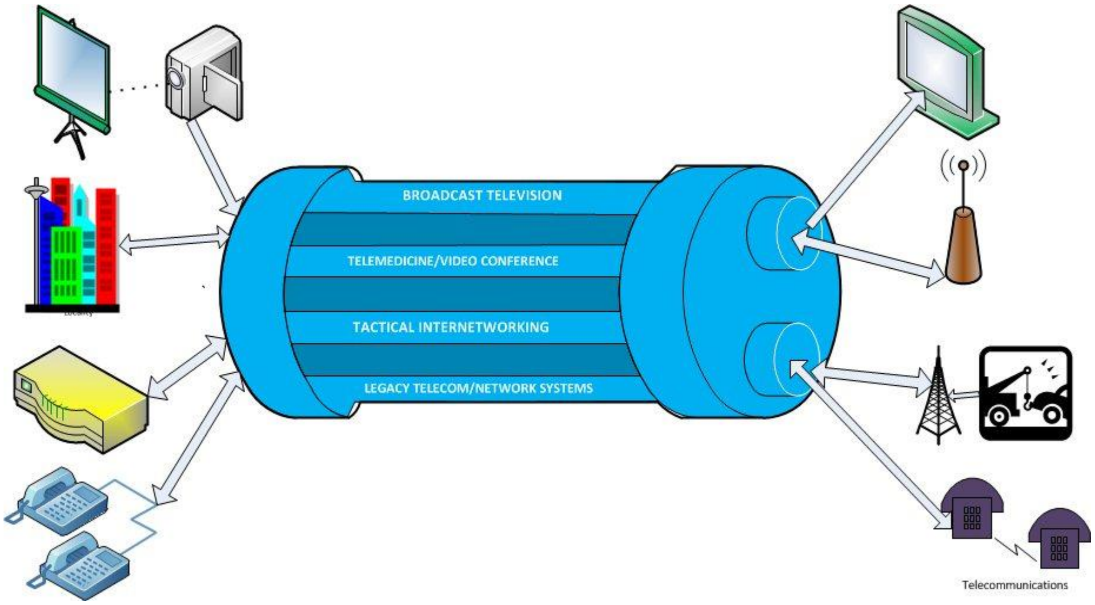


Figure 1.3 Wavelength Division Multiplexing

Figure 1.3 represents the Wavelength Division Multiplexing and the Internet Protocol (IP) which is the predominant protocol in WDM technology that yields maximum bandwidth. This IP/WDM technology becomes the best choice for upcoming generation in Internet [19].

1.4 Optical Circuit Switching Networks

The optical data along an optical fiber needs to be switched through the intermediate nodes. Traditional telecommunication networks centered on voice traffic apply a circuit-switching model. The communication between end users is achieved through the assistance of dedicated channels that are established in the connection-setup phase. These dedicated circuits cannot be used by other users during an active connection even if the communication is not taking place at that particular moment. The circuit-connections are simple when compared to other switching patterns employed in electronic networks and so the adaptation of circuit-switching model to optical circuit switching (OCS) networks are considered .

Adapting the above model, a virtual circuit is set up between the source and destination pair until the whole optical data is transmitted, when the circuit is torn. This architecture is called as Optical Circuit Switching (OCS) networks. This switching technique is an optical data transmission from the source to destination is done on pre-established all-optical communication paths called as lightpaths. The OCS architecture is given in Figure 1.4 [20].

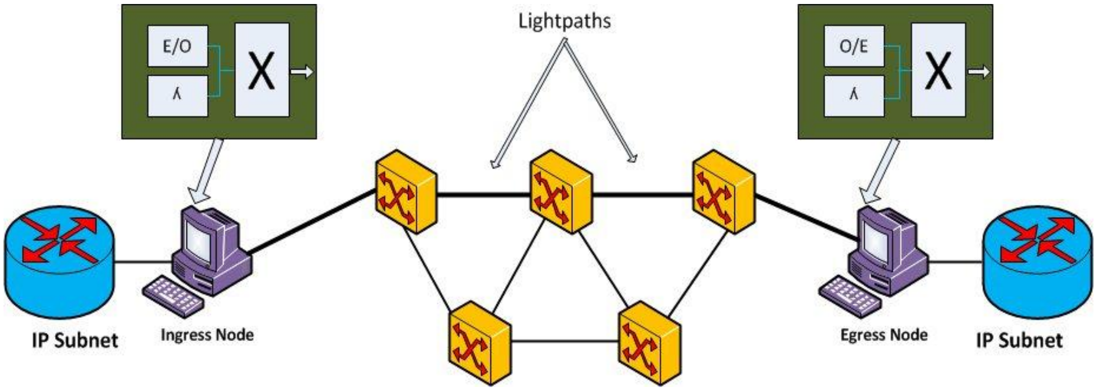


Figure 1.4 Optical Circuit Switching Network

1.5 Optical Packet Switching Networks

Optical Packet switching model was mainly developed for data-centric applications. This end-end communication is established by splitting the packets into fixed sized chunks and sending them to the destinations. At the intermediary, the packets are stored using buffering technique and there they are processed.

In Optical Packet Switching model, the data is divided into fixed sized packets and at each switching node, the same is converted electronically for processing. Processing is done based upon the information contained in the packet headers attached to the packet. The research on OPS was started at the mid 1990s and is still under progress due to the research inadequacy on potential optical buffers. Unlike electronic buffers, optical data cannot be stored for longer intervals in an optical buffer. The Optical Packet Switching architecture is shown in Figure 1.5 [21-25].

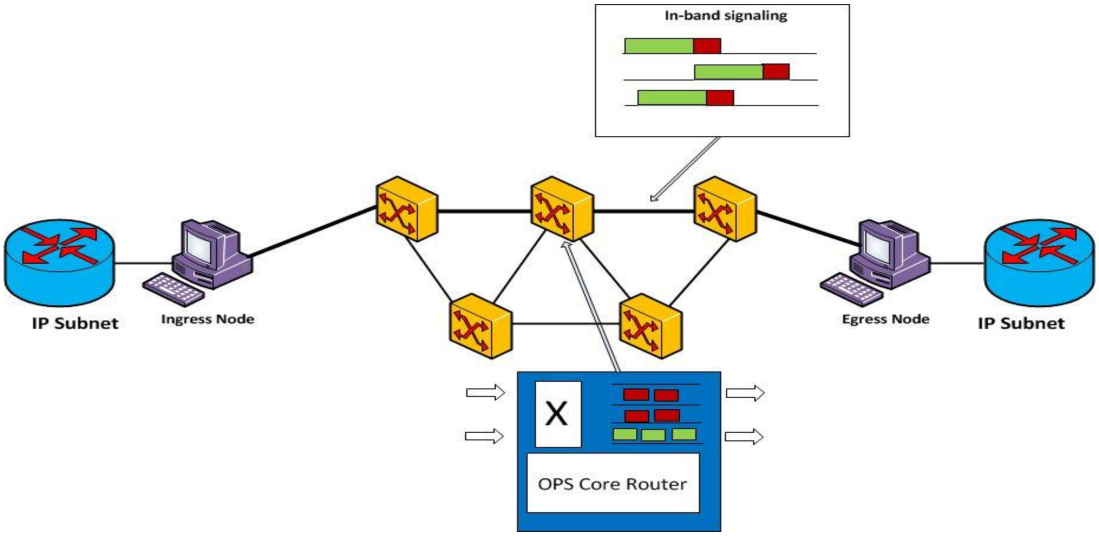


Figure 1.5 Optical Packet Switching Network

1.6 Optical Burst Switching Networks

In late 1990s, a novel switching was employed for the block transfer, which serves as another optical switching architecture. This is called as Optical Burst Switching (OBS). OBS shares the merits of OCS and OPS. OBS is advantageous over OCS with better bandwidth utilization and over OPS with less processing requirements and core network energy consumption [26][27].

The chief characteristics of OBS are:

- Packets are aggregated at the ingress nodes/switches to form bursts which are the smallest switchable units
- Wavelengths are reserved only for the fraction of time a burst is switched
- Thus, finer granularity than OCS and hence higher utilization of bandwidth

- Absence of optical buffer at the switches or possibly very small amount of FDL buffering at the switches
- Separation of control and data planes, packets in the control plane going through O/E/O conversion, and data bursts in the data plane always remain in the optical domain.
- Bursts coarser than packets in OPS, and OBS does not require any rigid synchronization between bursts and their control headers
- OBS better implementable with the current state of physical devices than OPS.

In OBS, the burst assembler present at the ingress assembles stream of packets belonging to a common destination end point i.e. forming a variable sized burst. In contrast to other switching paradigms, the header of a data packet is sent before the transmission of data, which is called as Burst Control Header (BCH) or control header. Whereas the payload is called as Data Burst (DB).The time interval between a BCH and its corresponding DB is termed as Offset Time (OT). The OT aids the BCH to configure the intermediate switch fabric for its corresponding DB. The Burst assembly is done at the ingress node and the disassembly is done at the egress node. The BCH sent along a different wavelength, is processed electronically at intermediate switches whereas the Data Burst is transmitted all-optically. The OBS architecture is given in Figure 1.6 [28][29].

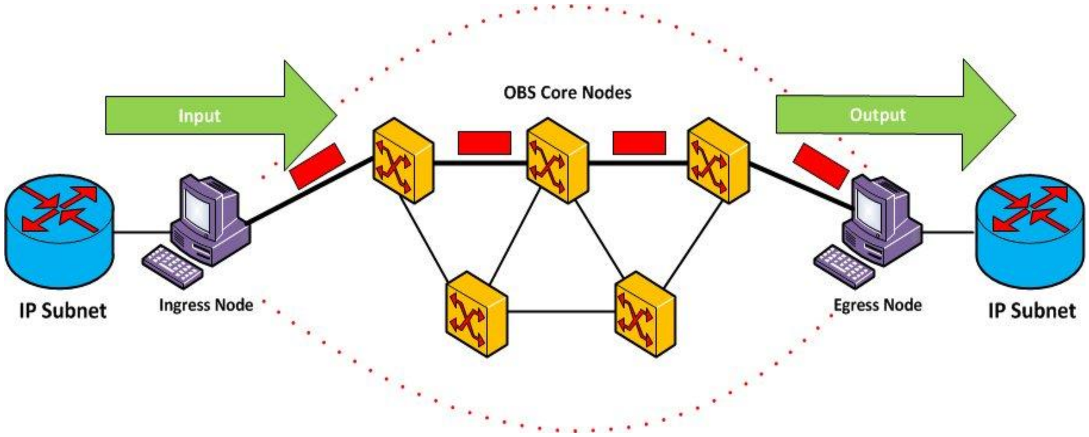
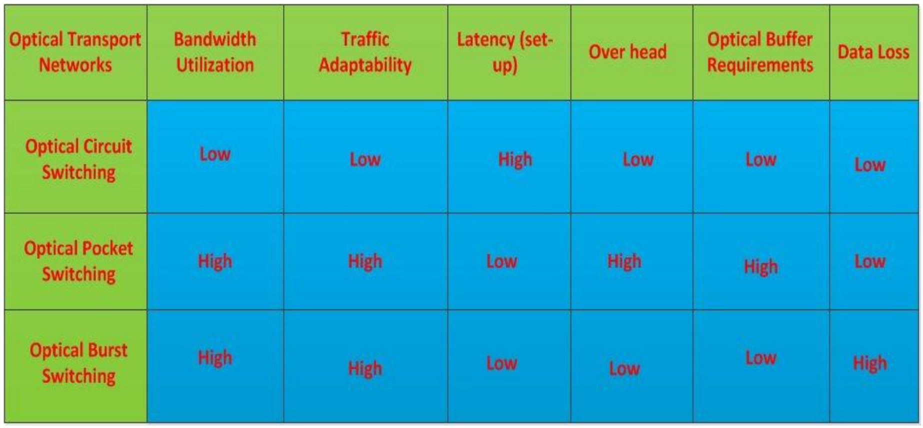


Figure 1.6 Optical Burst Switching Network

1.6.1 Scope of Optical Switching Networks

The switching speeds of electronics cannot cope up with the transmission capacity offered by optics and thus electronic switches have speed limitations. Power consumption for optical switches is comparatively lower compared to the electronic ones and higher heat dissipation complements to power consumption. Optical switches handles large number of switching ports when compared with electronic switches. Electronic switch architecture is not scalable enough and will suffer from technological limitations when trying to reach the multi-terabit throughput range. If the density of integration of electronic circuits is increased to cater higher bit rates, the system may compliment unwanted capacitances and impedances at small dimensions [30-34].



Optical Transport Networks	Bandwidth Utilization	Traffic Adaptability	Latency (set-up)	Over head	Optical Buffer Requirements	Data Loss
Optical Circuit Switching	Low	Low	High	Low	Low	Low
Optical Pocket Switching	High	High	Low	High	High	Low
Optical Burst Switching	High	High	Low	Low	Low	High

Figure 1.6.1 Scope of Optical Switched Networks

The Figure 1.6.1 summarizes the scope of optical switching networks technologies like optical circuit switching (supported by wavelength routed networks), optical packet switching, and optical burst switching. OCS is employed when the traffic is constant as in case of voice traffic, at the same time OCS cannot be utilized for highly dynamic traffic. Moreover, the lightpath building makes a round-trip delay, the high operating expense for the establishment of connection looks inefficient in case of shorter burst traffic. In OPS data is communicated forming optical packets which are transported across the optical core without transition to electrical form at intermediate core nodes. OPS can afford dynamic bandwidth allocation on packet-by-packet basis.

This dynamic allotment leads to a high degree of statistical multiplexing which enables the network to attain a higher degree of usage when the traffic is capable of change and bursty. But still, there exist lots of technical challenges to impose a practical OPS system. Implementation of optical buffer poses a major difficulty in OPS networks. The call to haste header processing and rigorous synchronization lays down OPS impractical employing current technology. They are having some limitations when applied to optical networks [35].

The main advantage of OBS networks is its low requirement for optical buffering and low average setup latency. Although, the burst latency setup is low packets must be delayed until the burst is ready to be transmitted, on average, packets experience longer average end-to-end delay when compared to packet switching. Furthermore, OBS tends to reduce the total overhead as well as the processing power requirement. These are mainly due to the fact that fewer individual packets are transmitted in OBS for the same number of incoming IP packets. On the other hand, the main concern in OBS networks is high loss rate. A practical approach to reduce high rate of loss in OBS networks is by using fiber delay lines. However, the tradeoff will be high cost and complexity [36].

1.6.2 Architecture of OBS Networks

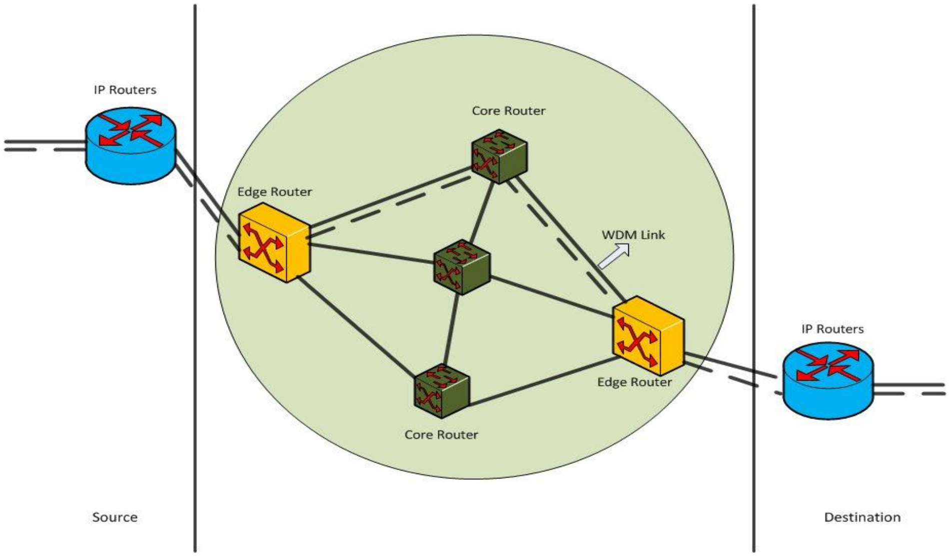


Figure 1.6.2 Architecture of OBS Networks

An optical burst-switched network consists of optical burst switching nodes that are interconnected via fiber links. Each fiber link capable of supporting multiple wavelength channels using wavelength division multiplexing (WDM). Nodes in an OBS network can either be edge nodes or core nodes as shown in Figure 1.6.2. Edge nodes are responsible for assembling packets into bursts, and scheduling the bursts for transmission on outgoing wavelength channels. The core nodes are primarily responsible for switching bursts from input ports to output ports based on the burst header packets, and for handling burst contentions [37][38].

The ingress edge node assembles incoming packets from the client terminals into bursts. The assembled bursts are transmitted all-optically over OBS core routers without any storage at intermediate nodes within the core. The egress edge node, upon receiving the burst, disassembles the bursts into packets and forwards the packets to the destination client terminals. In the network architecture, it can be assumed that each node can support both new input traffic as well as all-optical transit traffic. Hence, each node consists of both a core router and an edge router, as shown in Figure 1.6.2.1 and Figure 1.6.2.2.

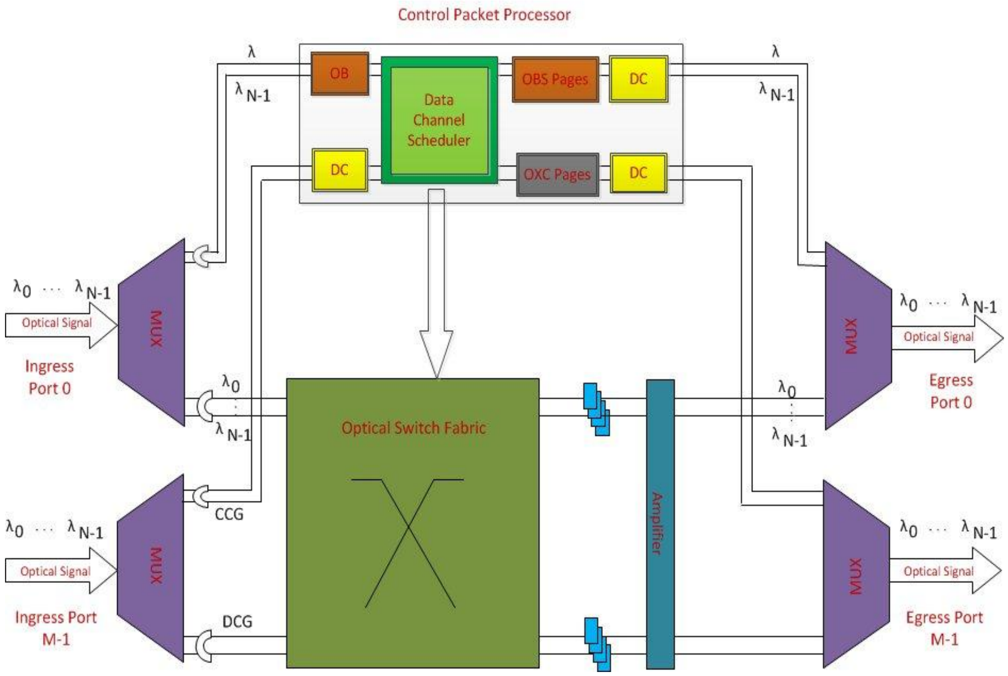


Figure 1.6.2.1 Architecture of Core Router

The core routers consist of an optical cross connect (OXC) and a switch control unit (SCU). The SCU creates and maintains a forwarding table and is responsible for configuring the OXC. When the SCU receives a burst header packet, it identifies the intended destination and consults the router signaling processor to find the intended output port. If the output port is available when the data burst arrives, the SCU configures the OXC to let the data burst pass through. If the port is not available, then the OXC is configured depending on the contention resolution policy implemented in the network. In general, the SCU is responsible for header interpretation, scheduling, collision detection and resolution, forwarding table lookup, switching matrix control, header rewrite, and wavelength conversion control. In the case of a data burst entering the OXC before its control packet, the burst is simply dropped [39][40].

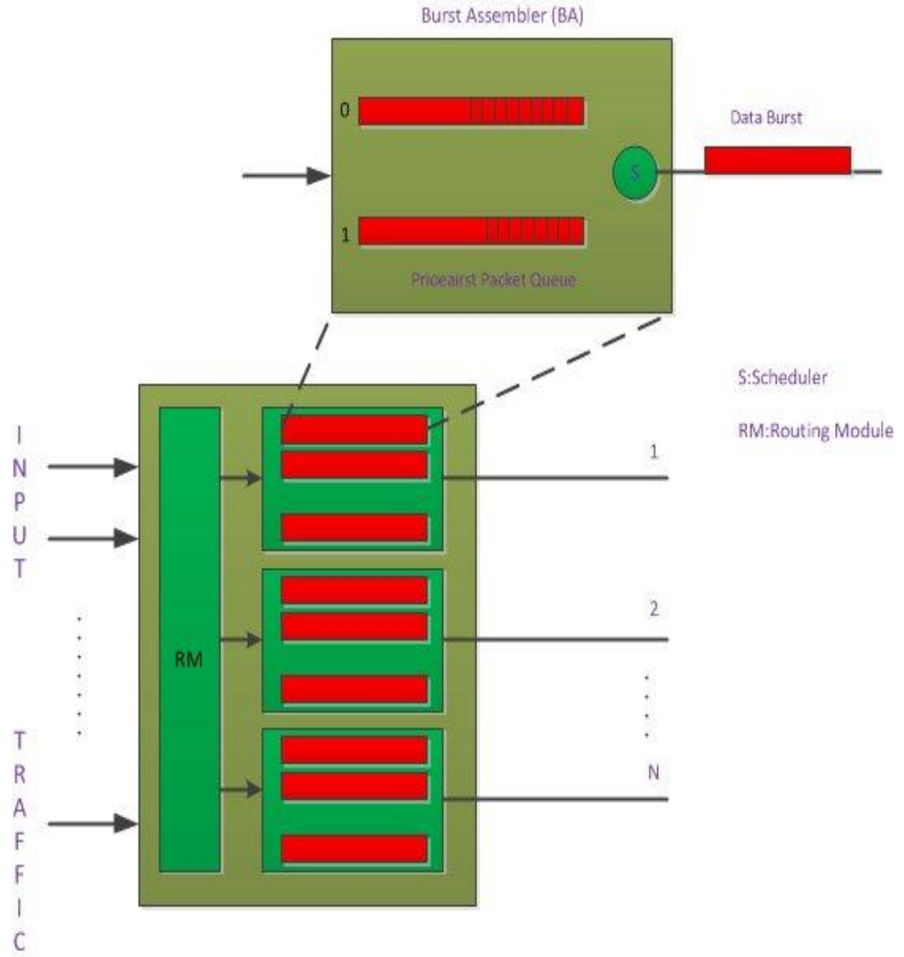


Figure 1.6.2.2 Architecture of Edge Router

The edge router performs the functions of pre-sorting packets, buffering packets, assembling packets into burst, and disassembling bursts into its constituent packets. Different burst assembly policies, such as a threshold policy or a timer mechanism can be used to aggregate bursty data packets into optical bursts and to send the bursts into the network. The architecture of the edge router consists of a routing module (RM), a burst assembler, and a scheduler. The routing module selects the appropriate output port for each packet and sends each packet to the corresponding burst assembler module. Each burst assembler module assembles bursts consisting of packets which are headed for a specific egress router. In the burst assembler module, there is a separate packet queue for each class of traffic. The scheduler creates a burst based on the burst assembly technique and transmits the burst through the intended output port. At the egress router, a burst disassembly module disassembles the bursts into packets and sends the packets to the upper network layers [41][42].

1.6.3 Functional Model of OBS Networks

The Figure 1.6.3 illustrates the various functionalities that are implemented within an optical burst-switched network. The ingress edge node is responsible for burst assembly, routing, wavelength assignment, and scheduling of bursts. The core node is responsible for signaling, scheduling bursts on core links, and resolving contention. The egress edge node is primarily responsible for disassembling the burst and sending the packets up to the higher network layer [45 - 47].

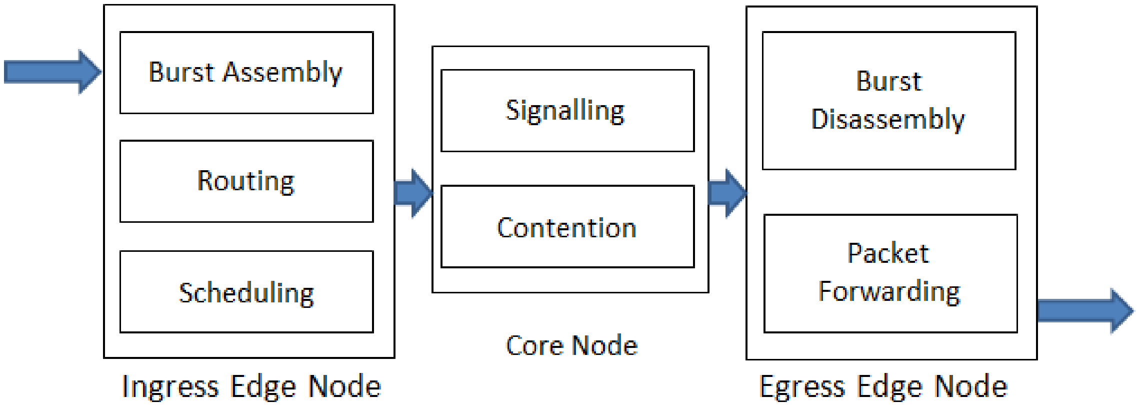


Figure 1.6.3 OBS Functional Diagram

1.6.4 Burst Assembly

Burst assembly is the process of aggregating and assembling input packets from the higher layer into bursts at the ingress edge node of the OBS network. The trigger criterion for the creation of a burst is very important, since it predominantly 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. In timer-based burst assembly approaches, a burst is created and sent into the optical network at periodic time intervals. A timer-based scheme is used to provide uniform gaps between successive bursts from the same ingress node into the core networks. Here, the length of the burst varies as the load changes. In threshold-based burst assembly approaches on the other hand, a limit is placed on the maximum number of packets contained in each burst [48].

Hence, fixed-size bursts will be generated at the network edge. If the packet arrival rate is very high, a threshold-based burst assembly approach will generate bursts at non-periodic time intervals. More efficient assembly schemes can be achieved by combining the timer-based and threshold-based approaches.

A major problem in burst assembly is the process of choosing the appropriate timer and threshold values for creating a burst in order to minimize the packet loss probability in an OBS network. The selection of such an optimal threshold (or timer) value is still under investigation. If the threshold is too low, the bursts become very short and more bursts will be generated in the network. The higher number of bursts leads to a higher number of contentions, but the average number of packets lost per contention is less. Also, there will be increased pressure on the control plane to process the control packets of each data burst in a quick and efficient manner. If the switch reconfiguration time is non-negligible, shorter bursts will lead to lower network utilization due to the high switching time overhead for each switched (scheduled) burst [49].

On the other hand, if the threshold is too high, then bursts will be long, which may reduce the total number of bursts injected into the network. Hence, the number of contention in the network reduces compared to the case of having shorter burst, but the average number of packets lost per contention will increase.

Thus, there exists a tradeoff between the number of contentions and the average number of packets lost per contention. Hence, the performance of an OBS network can be improved if the incoming packets are assembled into bursts of optimal length. The same argument is true in a timer-based assembly mechanism [50].

1.6.5 Burst Signaling

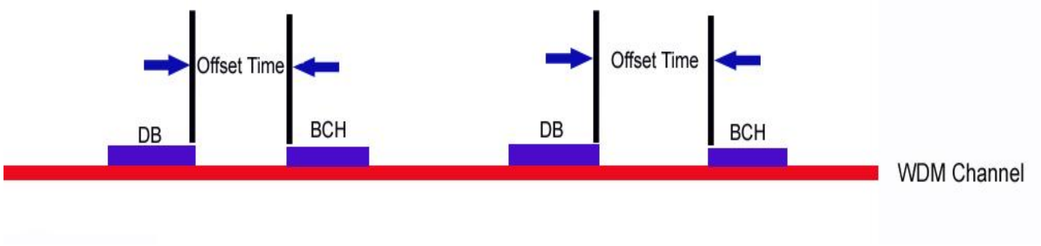


Figure 1.6.5.1 In – band signaling

In OBS, the basic switching entity is burst, which consists of a number of encapsulated packets. When a burst is transported over the optical core, a signaling scheme must be implemented in order to allocate resources and to configure optical switches for the burst at each node. For every burst there is a corresponding Burst Control Header (BCH) to establish a path from source to destination. BCH of a connection is sent prior to the transmission of Data Bursts (DB) with specific offset time on the same wave length is termed as In – Band Signaling shown in Figure 1.6.5.1. All BCH's of various connections are sent on the same control channel and the corresponding DB's will be sent on the different channels with specific offset time named as Out – of – Band signaling is shown in Figure 1.6.5.2 [51] [52].

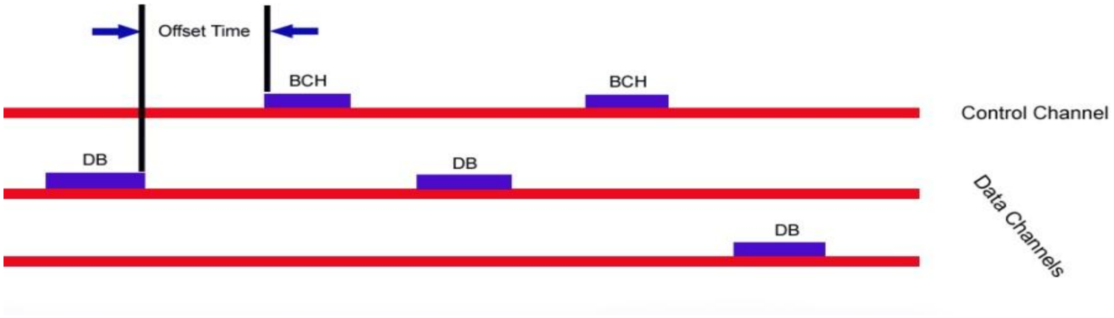


Figure 1.6.5.2 Out – of – band signaling

Offset time is the transmission time gap between the BCH and DB, which is used to allow the control part in intermediate core nodes to reserve the required resources for the onward transmission of bursts. The BCH fields are tabulated in Table 1.1.

Table 1.1 Information Fields in Burst Control Header

Information	Description
Burst ID	Identification ID of the Burst
Source IP	Source node address of the Burst
Destination IP	Destination node address of the Burst
Offset	Time gap between BCH and DB
Wavelength	Wave – channel through which DB arriving
Next port	Output port number through which DB to be sent
Arrival Time	Arrival time of the DB
Absolute Time	Departure time of BCH at each node

Several variations of optical burst switching signaling protocols are possible, depending on ways through which the resources along a route are reserved for a burst. The Figure.1.6.5.3 shows the signal classification schemes [53].

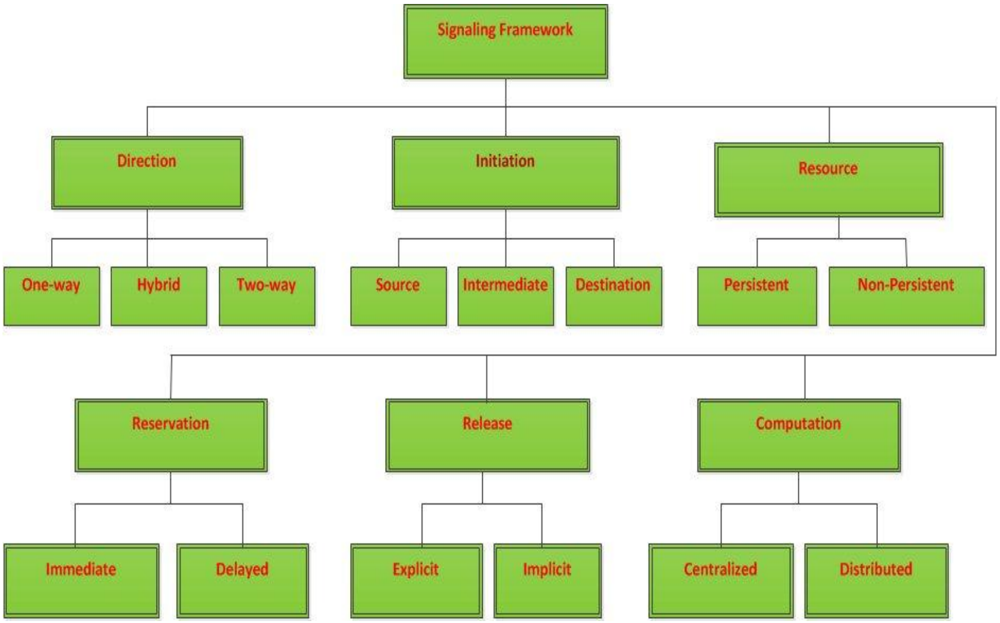


Figure 1.6.5.3 Signal Classification

In particular, a signaling scheme can be characterized by the following characteristics:

- One-way, two-way, or hybrid reservation.
- Source-initiated, destination-initiated, or intermediate-node-initiated reservation.
- Persistent or non-persistent reservation.
- Immediate or delayed reservation.
- Explicit or Implicit release of resources
- Centralized or distributed signaling.

1.6.6 OBS Applications

In OBS technology, burst switching concepts has been proposed as an extension to fast packet switching. Basic advantages of burst switching were reducing loop length and increasing data rate transmission. In optical burst switching the concept of burst switching is extended to optical networks. The main motivation for such a technology is to reduce (or eliminate) the need for optical buffering, as well as for minimizing the network overhead. Consequently, OBS technology has been considered as the underlying network technology for various applications with large data requests and sensitive to path delay. One such application is the distributed database. A distributed database is a collection of databases located at different geographic locations and connected through a network. In these networks, large pieces of data from different locations must be aggregated for computation. Hence, minimizing the delay in data aggregation is a key issue in improving the overall system throughput. In such applications, optical burst switching technology can achieve efficient data assembly and path setup while reducing network overhead. Another attractive area where OBS has been considered as an effective underlying technology is global Grid computing as a means of providing global distributed computing for applications with large bandwidth, storage, and computational requirements. A generic OBS-based architecture suitable to support Grid computing has been proposed in and key issues such as signaling issues, any cast routing, and transforming jobs into individual data bursts are discussed. Such areas are subjects of many ongoing research activities.

As a final remark, we emphasize that the developed concepts and protocols for OBS networks are not limited to optical networks. Many of the basic aforementioned techniques and models developed for OBS network can also be extended to sensor and satellite networks [54].

1.7 Research Issues in OBS Networks

When designing an optical burst network, many physical constraints must be taken into account. Some typical physical-layer issues include attenuation, dispersion, and fiber nonlinearities. While many of these issues apply to optical networks in general, several issues may raise particular concerns in optical burst switching networks. The main research issues are survivability, security, multicasting, Quality of Service and the interaction of optical burst switching with higher-layer protocols and applications.

1.8 Major contributions of the research

This research studies the possible vulnerabilities and impacts of different classes of attacks like espionage attacks, denial of service attacks and service disruption attacks. This research also quantifies the affect of these attacks on the optical burst switching networks analyzed through various performance metrics that are correlated to network reliability. It also investigates the proactive and reactive loss recovery mechanisms that are generally optimistic about the successful reception of the transmitted burst at the destination. Since, reactive mechanisms only attempt to recover when they receive an explicit failure message and they are generally pessimistic about the successful reception of the transmitted burst at the destination.

1.9 Structure of Thesis

This thesis is organized in to seven chapters. The present chapter elaborates on the introduction of optical networking background with multiplexing technologies and the need for wavelength division multiplexing. This chapter also elaborates on the various switching technologies with merits of optical networking.

It further, presents the state of the art of Optical Burst Switching networks followed by the discussion of main OBS architectures, including the edge node and core node representation of OBS networks and followed by the burst assembly, burst signaling and finalized with OBS applications.

The remainder of this thesis is organized as follows: Chapter 2 deals with the literature survey that encompasses security attacks in regular networks, security attacks in optical networks and OBS security. It also concludes with the Research motivation, Problem description and the Research Objectives.

Chapter 3 enumerates on the simulation methodologies followed for simulating the proposed work. The architecture of network simulator is presented and a brief studies as well as comparison of various OBS frameworks are elaborated. The reasons for choosing the appropriate OBS framework and the modifications done on its architecture are further presented in the chapter.

Chapter 4 depicts the identified Espionage attacks for OBS networks like Burst Hijacking Attack and Fake Spectral Attack with their associated mitigation mechanisms, mathematical models are presented.

Chapter 5 presents the mathematical models and effective mitigation mechanism for encountering identified Denial of Service attacks for OBS networks named as Burst Flooding Attack and Timeout Attack. It also presents the simulation results and analysis about the performance of the optical network based on time and load.

Chapter 6 describes the innovated efficient mitigation mechanism with the mathematical model for the identified Service Disruption attacks of OBS networks like Burst Circulating Attack and Land Attack. In addition, the impact of the aforementioned attacks are studied and elaborated with the normal scenario, attack scenario and solution scenario separately with possible countermeasures.

Chapter 7 concludes the work by highlighting the algorithms that facilitated to accomplish the objectives. The limitations of the research work are also stated and the insights for the possible future research directions are also discussed.

Finally, the appendix includes installation procedure of nOBS patch, Screen shots for installation procedure of nOBS patch, TCL scripts for OBS networks, Modifications of nOBS patch, Basic Optical Components and QoS metric for OBS networks.

1.10 Summary

In this chapter, we discussed a survey on optical networks with multiplexing technologies and also describe the various optical switching networks. In each, we describe the working technologies and challenges and the scope of optical switching networks is explained and merits and demerits of optical switching networks are also analyzed. And also, we discussed the detailed survey about OBS network architectures and described the basic functional model of OBS networks with burst assembly and burst signaling schemes of edge nodes and core nodes and finalized with OBS applications.

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

The term network security and information security refer in a broad sense to the confidence that protects the information and services available on a network from being accessed by unauthorized users. Security implies safety, including assurance of data integrity, freedom from unauthorized access of computational resources, snooping or wiretapping, and from disruption of service. In this Chapter, we discuss about the security issues in regular networks, optical networks and OBS security.

2.2 Security Issues in Regular Networks

Providing security for information requires protecting both physical and abstract resources. Physical resources include passive storage devices such as magnetic tapes and disks as well as active devices such as users' computers. In a network environment, physical security extends to the cables, bridges, and routers that comprise the network infrastructure. Protecting an abstract resource such as information is usually more difficult than providing physical security because information is elusive. Data integrity (i.e., protecting information from unauthorized change) and data availability (i.e., guaranteeing that outsiders cannot prevent legitimate data access by saturating a network with traffic) are so crucial.

Since, information can be copied as it passes across a network, hence protection must also prevent unauthorized listening. That is, network security must include a guarantee of privacy because information can be accessed and transferred at a high speed, Further, it can be difficult to discern the difference between a legitimate and illegitimate access while a transfer is in progress [55].

Attack upon a network can be broadly categorized into seven areas based on the goal of the attacker:

- Traffic analysis
- Eavesdropping
- Data delay
- Service denial
- QoS degradation
- Spoofing
- Man-in-the-middle.

Likewise,self-similar traffic is efficiently handled which is essentially the de facto trend to characterize the multi-time scale burstiness of the internet traffic. It has become important to factor self-similarity property for system design and evaluate its performance of real networks which shows that the aggregated traffic exhibits long-range dependence, leading to substantially different performance evaluations from those based on the Poisson traffic model.A novel QoS provisioning mechanism for OBS networks namely Classified Cloning for carrying real-time applications (such as video on demand, Voice over IP, online gaming and Grid computing). A minimum loss rate by minimizing end-to-end delay and jitter is achieved unlike the classical schemes. Ingress node performance is investigated. QoS provisioning offers a guaranteed burst loss rate, and delay unlike existing QoS implementation in OBS which use the burst offset time to provide such differentiation. The burst loss rate is reduced by 50% reduced over classical cloning. The Classified Cloning scheme outperforms Basic Closing Scheme (BCS) and the classical QoS provisioning mechanisms in OBS for three reasons.

- It retains the same delay as without cloning because the Classified Cloning scheme does not use an extra offset time for class isolation
- It implements immediate loss recovery for real-time applications
- It does not need extra hardware or optical splitting because classified cloning is implemented in the ingress node.

In addition, the Load-Level based Admission Control Mechanism (LLAC) reserves a different amount of wavelengths in a link for each service class. The number of wavelengths that a class can conquer in a link is defined by a parameter called load level. LLAC discriminates the blocking probability practiced by a given class, admitting bursts based on the network load and the load level associated to this class. The performance of load-level-based technique is evaluated with a single-link model. The single-link model provides a good approximation for the blocking probability of each service class in a link, but this method does not consider that the blocking probability in a link is influenced by other links of the network. The single-link model is used to represent realistic network scenarios that have some shortcomings. This model does not take into account the wastage of capacity on links traversed before by blocked bursts. The single link model does not consider the reduction on the load offered to core nodes because of the blocking of bursts along source-destination path. In OBS networks, these two factors must be considered in the development of more realistic models.

In the next section, we discuss some basic class of attacks in regular networks which can be a cause for slow network performance, uncontrolled traffic, viruses etc.

2.2.1 Security Threats

There is a wide range of security threats that can exploit the vulnerability of a network through an attack. Main security threats are denial of service, distributed denial of service, viruses, Trojan horses, spywares, malwares, unauthorized access to the network resources and data, accidental deletion of the files and the uncontrolled internet access.

2.2.2 Virus Attack

A computer virus is a small program or an executable code that when executed and replicated, perform different unwanted and harmful functions for a computer and a network. Viruses can destroy the hard disks and processors, consume memory at a very large scale and destroy the overall performance of a computer or network. A Trojan is a malicious code that performs harmful actions but it cannot be replicated and they can destroy systems' critical data.

A computer worm is a program that replicates to all network and destroys useful data. The viruses, malware, adware and Trojan horses can be prevented by updating the antivirus program with the latest pattern files.

2.2.3 Unauthorized Access

Access to the network resources and data are allowed only to the authorized persons. Every shared folder and resources in your network can be accessed only by the authorized persons and should also be scanned and monitored regularly.

2.2.4 Information Theft and cryptography attacks

Another threat to a network is the loss of the important information and this loss of information can be prevented, if good encryption methods such as 128 bit security or 256 bit security encryption methods are incorporated. In this way, the data can be securely transferred through FTP programs.

2.2.5 Unauthorized application installations

Another type of virus and security attack prevention method is to install only the authorized software applications to the network server and the client computers. Unauthorized users are not allowed to install any kind of program which can cause security threats such as songs or video programs, codec, gaming software or other web based applications.

2.2.6 Application-Level Attacks

The attacker exploits the weakness in the application layer – for example, security weakness in the web server, or in faulty controls in the filtering of an input on the server side. Examples include malicious software attack (viruses, Trojans, etc.), web server attacks, and SQL injection.

2.3 Security Issues in All - Optical Networks

AONs are emerging as a promising technology for very high data rates due to its flexible switching and broadband application support. In particular, they provide transparency capabilities allowing routing and switching of traffic without regeneration of signals within the network.

Based on multiplexing techniques, they are being used to increase the transmission capacity in the optical fibre, AONs are divided into two types: time-division multiplexed (TDM) and wavelength-division multiplexed (WDM) networks [56]. In the recent years, WDM technology has been rapidly gaining acceptance as a significant technology employed for taking advantage of the enormous bandwidth in optical networks. WDM optical networks are increasingly built upon transparent optical nodes (TONs) such as Wavelength Selective Switches (WSSs) and all-optical amplifiers. Although transparency, in AONs, offers many advantages for high rate communications, it manifests new and still unstudied security vulnerabilities.

All-optical components are particularly vulnerable to various forms of DoS and eavesdropping attacks. These attacks can be broadly classified into three categories:

1. Traffic analysis and eavesdropping - the attacker passively analyses traffic on the network.
2. Service denial - the optical signal is disrupted by the attacker(s).
3. QoS degradation - the attacker overpowers legitimate optical signals with attack signals.

In particular, DoS and QoS degradation attacks must be detected and identified at all nodes in the network where attacks and signal degradations may occur. Moreover, the speed of attack detection must commensurate with the data transmission for the following main reasons:

1. The high data rates ensure that large amounts of data can be compromised in short time.
2. The large network latency causes large amounts of data to be ‘in flight’ at any instance.
3. An attack which is erroneously identified as failure can spread through the network.

4. Inappropriate action might be taken by the NMS, if attacks are not identified at any of the nodes.

Transparency in AONs may then introduce significant miscellaneous transmission impairments such as crosstalk, Amplified Spontaneous Emission (ASE) noise, and gain competition.

As a result, those impairments aggregate and can impact the signal quality as it progresses towards its destination, so that the received BER at the destination node might become unacceptable high.

2.3.1 Categories of Attack Recovery

Burst loss may still occur after using the different loss minimization mechanisms. Hence, loss recovery mechanisms are essential in addition to loss minimization mechanisms to support a reliable OBS transport network.

All loss recovery mechanisms are classified into one of two categories, namely, Reactive and Proactive. Reactive loss recovery mechanisms are generally optimistic about the successful reception of the transmitted burst at the destination. Hence, reactive mechanisms only attempt to recover when they receive an explicit failure message. On the other hand, proactive loss recovery mechanisms are generally pessimistic about the successful reception of the transmitted burst at the destination. Proactive mechanisms transmit additional information (overhead) along with the original burst so as to handle certain loss scenarios. Broadly speaking, reactive mechanisms are better suited when burst loss is rare and bandwidth utilization needs to be optimized. Proactive mechanisms are better suited when burst losses are high and delay needs to be optimized. The different loss recovery mechanisms are described below. Note that a combination of loss recovery mechanisms can be implemented to further reduce the loss in the OBS network.

2.3.1.1 Retransmission

The basic idea of burst retransmission is to allow contending bursts to be retransmitted in the OBS layer. In this scheme, BCHs are sent out prior to data burst transmission in order to reserve resources. After an offset time, the burst is transmitted. At the same time, the ingress node stores a copy of the transmitted burst for possible retransmissions. As the BCH traverses through the core nodes, if the channel reservation fails due to a burst contention, the core node will send an *Automatic Retransmission Request* (ARQ) to the ingress node in order to report the reservation failure. Upon receiving an ARQ, the ingress node retransmits the corresponding duplicate preceded by its duplicate BCH.

2.3.1.2 Burst Cloning

In burst cloning, the idea is to replicate a burst and send duplicated copies of the burst through the network simultaneously. If any one of the burst copies is lost, the destination egress nodes can recover from the core loss using the other duplicate burst. Additional information needs to be stored in the BCHs to identify duplicates.

So that, in the case both original and duplicate burst reach the destination, the destination will select one of the bursts, disassemble the burst, and forward the constituent packets on to the corresponding destination hosts. Based on the load on different links in the network, the original and the clone could be sent on different paths. Primary design issues in burst cloning are to select the optimal node at which to clone and to prevent cloned bursts from contending for resources with their original bursts.

2.3.1.3 Quality of Service

QoS support is an important issue in OBS networks. QoS differentiation schemes generally focus on providing loss differentiation, delay differentiation, or bandwidth guarantees. In OBS networks, bursts follow an all-optical path from source to destination. Thus, the delay incurred from source to destination is primarily due to propagation delay, and bandwidth guarantee is implicitly provided by supporting loss guarantee. Hence, the focus of QoS support in OBS networks is to provide loss differentiation.

In IP networks, many queuing disciplines have been developed in order to provide QoS differentiation. Priority queuing (PQ) is a relative differentiation scheme that stores the packets in prioritized queues at each hop, and the packets are scheduled onto an output port only if all packet queues of higher priority are empty. Weighted fair queuing computes virtual finishing time for each packet at the head of each session queue, and transmits the packet with the smallest virtual finishing time. Weighted fair queuing can provide absolute QoS differentiation in the sense that it is able to guarantee a predictable amount of bandwidth and a maximum delay bound for a specific session.

On the other hand, a proportional QoS differentiation model has been proposed in order to provide relative QoS differentiation. Using this model, the relative QoS differentiation is refined and quantified in terms of queuing delay and packet loss probability.

Further, a dynamic class selection framework introduced to provide absolute QoS in which the proportional QoS differentiation approach controls the QoS spacing of each class at every hop, and the users dynamically search for an appropriate class to meet their absolute requirements.

In OBS networks, QoS differentiation can be provided by using extra offset time, intentional burst dropping, burst preemption, burst segmentation, deflection, and BCH scheduling.

2.3.1.4 Adaptive Routing

The adaptive Routing mechanism is based on link loss and route loss probability estimations. The proposed scheme uses a technique similar to the one used to compute the link loss probabilities of the individual links. For every destination R-candidate routes are pre-computed and are available at all source nodes. The estimate of the loss probability could be made on a per-link basis and route loss probabilities can then be estimated using the link loss probabilities.

In the proposed policy, link loss probability is computed at all nodes in the network, one for every outgoing link from the node. The loss probability estimate is initially set to zero. When a burst is successfully transmitted over a link, a positive feedback is generated, and when a burst is dropped on the link, a negative feedback is generated. Based on the updating scheme, the loss probability estimate for the link is computed. The source of every flow periodically sends a probe packet along the shortest routes to collect the loss probability estimates on all links along the route. The loss probability for the entire route is calculated from the loss probability estimates on each link.

2.3.1.5 Link loss Probability Estimation

Initially the loss probability of all links is set to zero. For each link, two parameters are recorded, the number of bursts arrived into that link and the number of bursts dropped on that link. Initially for each link the number of bursts arrived and the numbers of bursts dropped are set to zero. Based on the feedback received, these two parameters of the link are updated.

For a positive feedback (i.e. successful burst transmission), the number of bursts arrived on the link is incremented and for a negative feedback (i.e. for burst drop on the link), both the parameters (i.e. number of bursts arrived and number of bursts dropped) are incremented by one.Loss probability of the link is the ratio of the total number of bursts dropped on that link to the total number of bursts arrived on that link. The source of every flow periodically sends a probe packet along the pre-computed shortest route to collect the loss probability estimates on all links along the route. The loss probability for the entire route is defined as the maximum of the loss probabilities of all the individual links in that route.

For routing a burst, the source node dynamically selects a least congested route to minimize the burst contentions. Source node will compare the loss probabilities of R candidate routes between that source-destination pair and select a route with minimum loss probability for scheduling a burst. Instead of single fixed route for each source-destination pair, Adaptive Routing uses pre-computed R-candidate routes for each source-destination pair and it will dynamically chooses a best route among the existing R-candidate routes which is having least congestion. Due to dynamically selecting the least congested route proposed mechanism reduces the overall burst drop in the network.

2.3.1.6 Scheduling Schemes

Several scheduling schemes have been proposed in order to improve the performance of the LAUC-VF algorithm .This has been proposed as an improved void-filling scheduling algorithm that selects a data channel in which the ending void, newly generated after the transmission of an arriving burst, becomes minimum. Since the generated voids are used in minimum data channels will be efficiently utilized, thereby reducing burst loss probability compared to the LAUC-VF algorithm.

The paper [57] has generalized variations of void filling algorithms and has proposed minimum starting void (Min-SV) that achieves the same objective as LAUC-VF, minimum ending void (Min-EV) that achieves the same objective as in [58], maximum starting void (Max-SV) and maximum ending void (Max-EV) that are opposite to Min-SV and Min-EV respectively, and a best-fit algorithm that aims to minimize the overall voids generated (including ending voids and starting voids).

2.3.1.7 Reliable OBS

This section focuses on the goal of implementing a reliable optical burst switched network using loss minimization and loss recovery mechanisms.

2.3.2 Contention Resolution and Contention Avoidance schemes

All loss minimization mechanisms are classified into two broad categories, namely, *Contention Resolution* and *Contention Avoidance*. Contention resolution mechanisms attempt to minimize data loss when a contention has already occurred. On the other hand, contention avoidance mechanisms attempt to minimize the occurrence of contentions.

2.3.2.1 Contention Resolution Mechanisms

Since OBS core networks use one-way based signaling scheme, OBS networks suffer from random burst losses due to burst contentions, even at low traffic loads. There are many contention resolution schemes that can reduce random burst loss in OBS networks. The primary contention resolution mechanisms are optical buffering [59], wavelength conversion [60], deflection routing [61] and [62] burst segmentation [63] and [64]. These mechanisms minimize data loss when a contention has already occurred.

2.3.2.1.1 Fiber Delay Line

The fiber delay line scheme uses FDLs to delay the transmission of arriving contending bursts for some amount of time. A FDL is simply a length of fiber. Once a burst enters into a FDL, the burst must be delayed a fixed amount of time, i.e. the light transmission time in the fiber. This property of FDLs is referred to as the deterministic property. If the maximum delay provided by FDLs is not sufficient to avoid contention, the burst will be dropped if no other contention resolution schemes are used, which is referred to as the balking property of FDLs. The performance modeling of OBS networks with FDLs has been investigated in [65].FDLs can combine with optical switches to construct switched delay lines (SDL), where one or multiple blocks of FDLs are cascaded by optical switches and each block of FDLs consists of parallel FDLs with different delays.

SDLs can be classified into single-stage with a single block of FDLs and multi-stage with multiple blocks of FDLs. Single-stage SDLs are generally easier to control, while multiple-stage SDLs are able to practically accommodate large buffer depths. SDLs can also be classified into feed-forward and feedback. In feed-forward SDLs, each FDL feeds forward to the next stage of the switch. In feedback SDLs, the tail of a delay line is connected to the input of the same switch stage. Based on the positions of FDLs in a switch, there are four common configurations: output buffering, shared buffering, recirculation buffering, and input buffering. An output-buffering switch consists of a switch with FDLs on each output port, and an input-buffering switch consists of a switch with FDLs on each input port. In a shared-buffering switch, FDLs are shared among multiple output ports. In a recirculation-buffering switch, a number of recirculation loops from the output port of a switch feed back into the input ports. Note that the delay accommodated by FDLs is very limited, e.g. to delay a single burst for 1 ms requires a fiber with length of 200 km. The sustainable load of fiber delay line buffers is defined as the load at which a system with infinite buffering capacity becomes unstable. It has been shown that, due to finite delay granularity of FDLs, the sustainable load is generally less than 100%, and that the burst-size distribution also impacts the sustainable load.

2.3.2.1.2 Wavelength Conversion

Wavelength conversion uses wavelength converters to convert arriving contending bursts to different wavelengths in order to avoid contentions. Wavelength converter can be classified into three categories based on the forms of control signals and the mapping functions between input wavelength and output wavelength: optoelectronic, optical gating, and wave-mixing. Optoelectronic converters detect optical signals and retransmit the signals using O-E-O conversions. Optical gating converters employ an optical device which changes its characteristics depending on the intensity of the input signal. The change of its characteristics is monitored by a probe signal which contains the information in the input signal. Wave- mixing converters can offer the highest transparency using wave-mixing technology.

This type of converter is able to simultaneously convert multiple input wavelengths to multiple output wavelengths. More converter classification and comparisons have been described in [66].The performance of an optical packet switched network with wavelength converters has been evaluated in [67]. The results have shown that wavelength conversion greatly improves the performance of WDM packet networks. These improvements are manifested by the reduced switch complexity, switch size, number of optical gates, and number of wavelength channels. This paper has also claimed that the flexibility of scheduling packets on any available wavelength rendered by wavelength conversion even allows optical packet switches without using any FDL for buffering. Among current wavelength conversion technologies, the maximum conversion range is limited to retain a large signal to noise ratio for converted signals. Consequently, an input wavelength can only be converted into a limited range of wavelengths, referred to as limited wavelength conversion.

2.3.2.1.3 Deflection

In the deflection scheme, bursts are initially routed through their primary (shortest) paths. An arriving burst is redirected to an alternative path at a core node when the burst encounters a contention. The deflection scheme will increase traffic load in the network since the deflected bursts may traverse additional hops, which results in higher burst contention probability. The deflection scheme performs very well when the load in a network is low since bursts can be deflected to unused network resources. When the load in a network is high or medium, the amount of unused network resources in the network is small, hence the deflection scheme may exacerbate network performance. Several works have analyzed the burst loss probability in an OBS network with deflection [68]. The deflection scheme suffers from potential looping and insufficient offset time [69].

The potential looping problem is caused by rerouting a deflected burst back to nodes that have already been visited. The potential looping problem can be solved by setting a delay constraint for the deflected burst or by implementing a loop less deflection scheme.

The insufficient offset time is caused when a deflected burst traverses more hops along the alternative path than the primary path. Since the offset time between the deflected burst and its BCH is initially determined for the primary path, the deflected burst may lack sufficient offset time, thereby leading to the burst being dropped. One approach to solve the insufficient offset time problem is to introduce additional offset time at ingress nodes. However, it may be difficult to predetermine the additional offset time at ingress nodes. Another solution is to have optical buffers at core nodes in order to delay the deflected burst for the additional offset time introduced by deflection.

2.3.2.1.4 Segmentation

In the previous contention resolution schemes, if contentions cannot be resolved for arrival of bursts, those bursts will be entirely dropped. They have proposed segmentation to resolve burst contention, where only packets in a contending burst which are overlapped with another burst, are dropped. In [70], two segmentation approaches have been proposed in order to determine which packets to drop when using segmentation, namely tail dropping and head dropping. In tail dropping, the overlapping packets that are in the tail of a contending burst are dropped, and in head dropping, the overlapping packets that are in the head of a contending burst are dropped. One advantage of the tail dropping approach rather than the head dropping approach is that there is a better chance of in-sequence delivery of packets at the destination.

2.3.2.2 Contention Avoidance Mechanisms

The contention resolution mechanisms minimize packet losses based on the local information at the nodes where contentions occur, but do not address the more fundamental problem of congestion in the OBS core. In [72], two dynamic load-balanced routing techniques are proposed to avoid burst contentions. The simulation results show that the proposed contention avoidance techniques improve the network utilization and reduce data loss. In [73] and [74], the authors investigated similar load-balancing routing (or path switching) approaches using adaptive alternate path routing and concluded with similar observations as [75]. In [76], a proactive scheduling algorithm referred to as burst overlap reduction algorithm (BORA) is proposed.

The basic idea is to serialize the bursts on outgoing links to reduce the burst overlapping degree (and thus burst contentions and burst loss at down- stream nodes). The biggest side-effect of BORA is that it introduces significant delay at the edge during serialization of bursts. In addition, several other edge-based admission control techniques can be incorporated to minimize the number of contentions in the core.

2.4 OBS Security Vulnerabilities

This section presents the literature review on the security vulnerabilities for OBS networks. The two major security concerns in OBS include:

- Orphan Bursts
- Malicious Burst Headers

During transmission, the scheduling request for a BCH may be rejected due to overflowing demands and the corresponding data burst is disconnected thus becoming an Orphan Burst. These burst may flow along an unintended path wasting the bandwidth and tapped by an attacker to compromise the security [77][78].

Sometimes the BCH may be modified and compromised by fraudulent parties forming Malicious Burst Headers. It includes:

- Orphan burst
- Fake burst header attack
- Reply attack
- Burst tapping attack
- Burstification attack

2.4.1 Orphan Burst

TCP/OBS network, there is one to one correspondence between the data burst and the burst header, which is sent ahead of the data burst on a separate control channel. The burst header contains the control information and takes care of making the WDM channel reservation for matching burst. In case if the scheduling request is rejected at one of the

OBS core routers, there will be absence of valid optical path established for the upcoming burst. Since the burst has been launched already, anyway it is going to reach the input of the core router. Now the burst is no longer connected with its burst header and becomes an orphan burst. Depending on the configuration of the switching fabric at the time of the burst arrival, the orphan burst can choose some path unknown in advance.

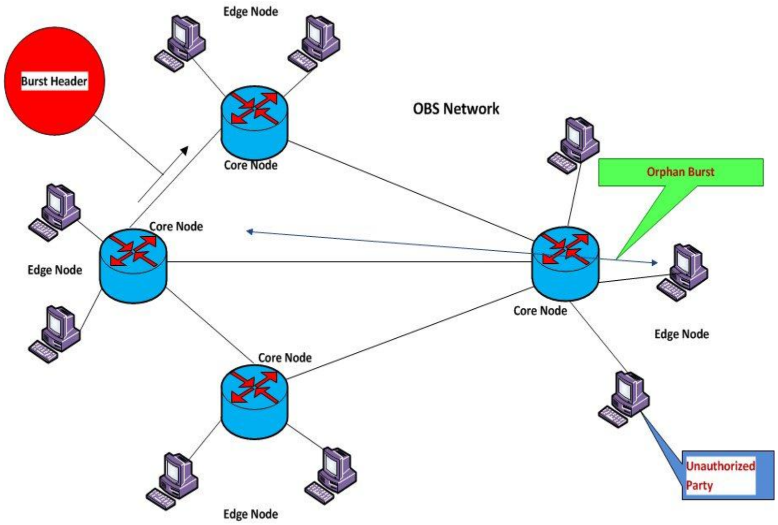


Figure 2.4.1 Example of an Orphan Burst

2.4.2 Fake Burst Header Attack

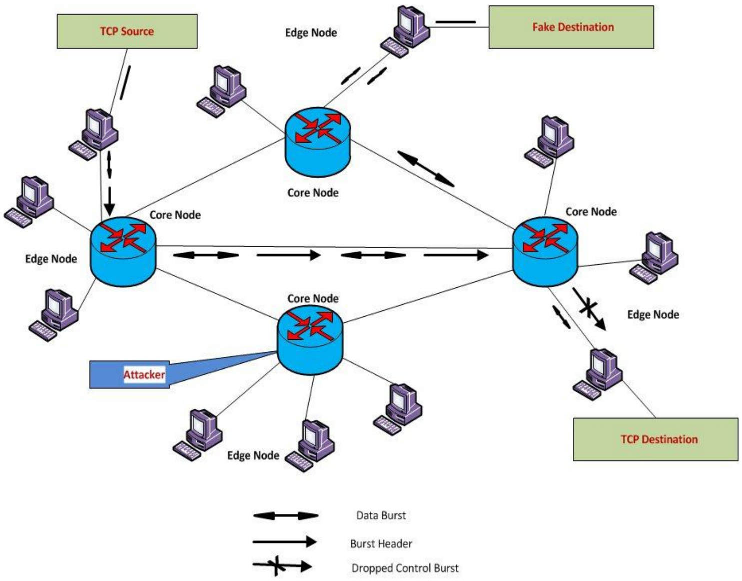


Figure 2.4.2 Example of Fake Burst Header Attack

In this attack, the attacker injects a malicious burst control header in the compromising intermediate node to direct the forthcoming data burst to the fake destination. After an offset time the data burst reaches the compromised node and from there it forwarded to the fake destination. There it is tapped off, compromising security. In order for the real destination to believe everything is fine in the network. The fake destination forwards the same data burst to the real destination through the compromised node impersonating the source by generating a new BCH as shown in Figure 2.4.2.

2.4.3 Reply Attack

Replay attack can be set up in motion by seizing a legal but expired burst and carrying at a later time, or by introducing an expired burst and carrying at a later time, or by injecting expired burst header to make the optical burst to circulate in the OBS network, holding up its delivery to the final destination as shown in Figure 2.4.3.

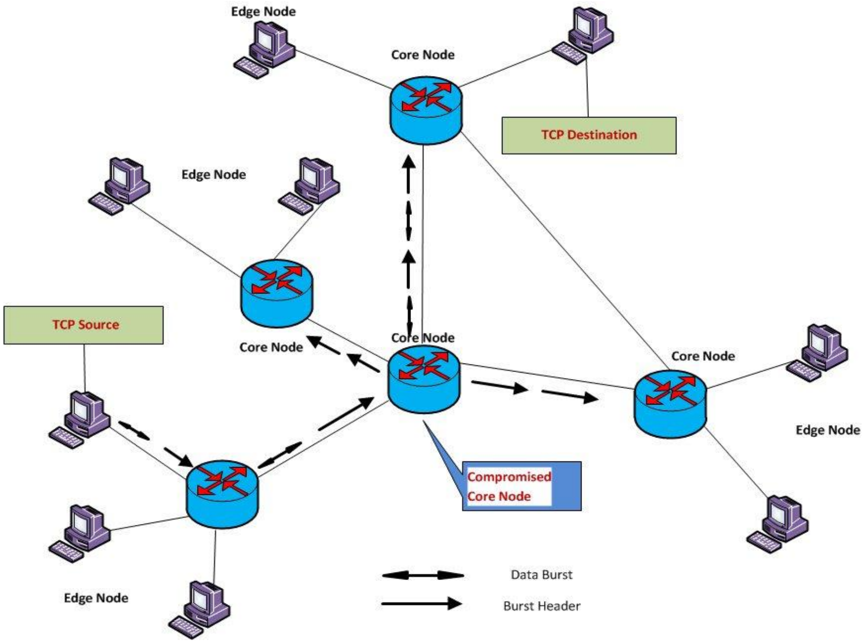


Figure 2.4.3 Example of Reply Attack

2.4.4 Burst Tapping Attack

To support multicast routing in WDM optical networks, virtual source nodes are unavoidable. A Virtual Source (VS) node is an optical node which has both light splitting capabilities as well as wavelength conversion capability. VS node can transmit an incoming burst to multiple destinations on any wavelength.

The task of the intermediate core node is to receive the burst header and establish the path for the respective data burst and forward the burst header to the next intermediate node until it reaches the egress node. If it is compromised and creates a copy of the original burst header and alter its value to establish a path to the malicious destination, then the respective data burst will reach the original destination as well as the malicious destination. The malicious destination will not acknowledge for this stolen burst just to escape from being caught. Thus it compromises the authentication of the data burst and it is named as burst tapping attack as shown in Figure 2.4.4.

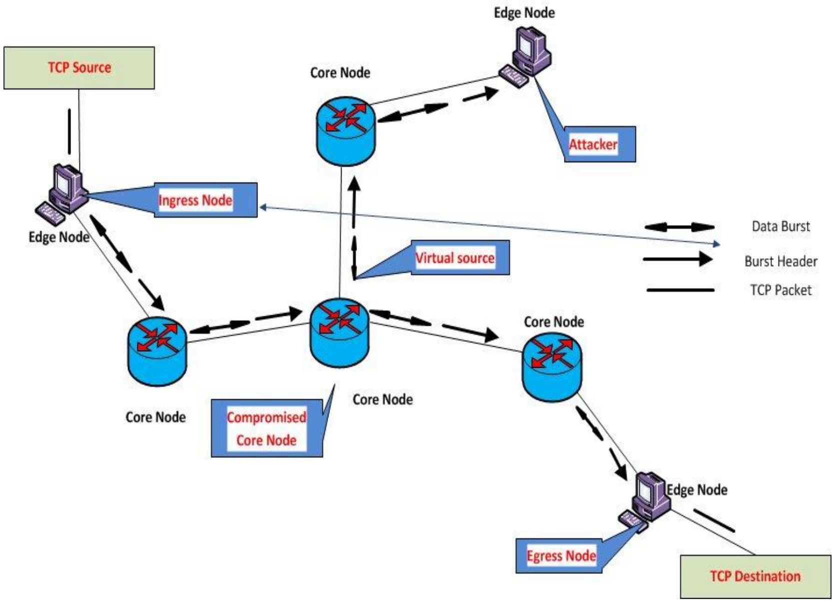


Figure 2.4.4 Example of Burst Tapping Attack

2.4.5 Burstification Attack

This section presents the security threat like the burstification attack that occurs either in edge node or core node as shown in Figure 2.4.5.1 and Figure 2.4.5.2 Bursts are assembled and created at the edge node. Bursts thus created are to be scheduled for particular channels at specified voids based upon one of the various void–filling scheduling approaches as mentioned. The voids present in the channels fit the burst scheduled. At some instant, a particular node present at the intermediary could be compromised by changing the value of assembled burst’s size at the BCH.

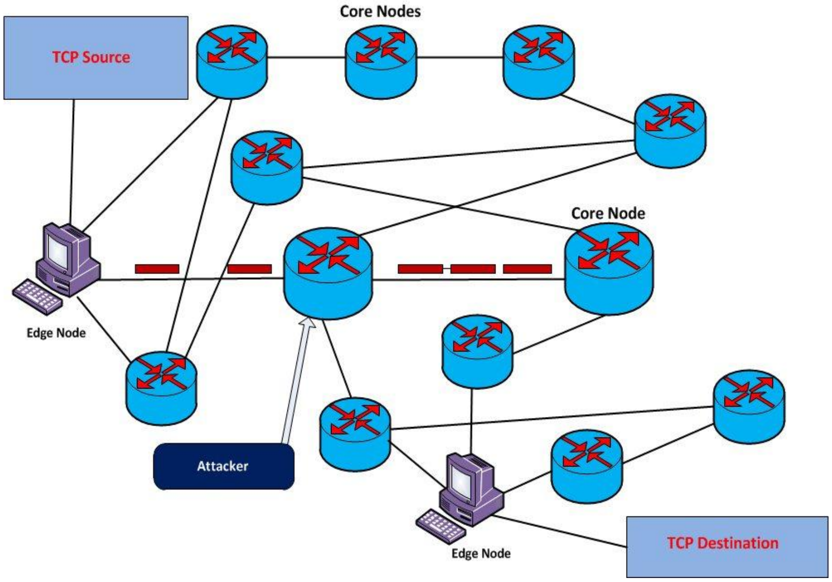


Figure 2.4.5.1 Burstification Attack at the core node

The increased burst size value induces the egress node to check the value of the burst size value during disassembly. In this attack, if the value is not comparable equal, then the burst is identified as another burst. For example, in Figure 2.4.5.1 the attacker compromises node 3 and alters the value of the burst size at the BCH. The burst still gets scheduled and forwarded to node 11 (i.e.., egress node), where it is checked for burst size. The burst size will not be the same and thus node 11 is forced to request for a retransmission. The same scenario could also happen at the ingress. The attacker can compromise ingress node and create bursts of bigger sizes.

This further increases the burst reservation duration considerably. Increase in the burst reservation time increases the propagation delay which in turn affects burst latency. The burst throughput, which is inversely proportional to the burst latency, gets lower as a result of the Burstification threat. Figure 2.4.5.2 shows the Burstification attack happening at ingress optical node 0. In which, an attacker compromises ingress node 0 and increases the size of all the incoming bursts of the outgoing link of the node 0 to node 3 and 9. This particular phenomenon will render burst reservation problem as the burst scheduler does scheduling at the ingress node itself.

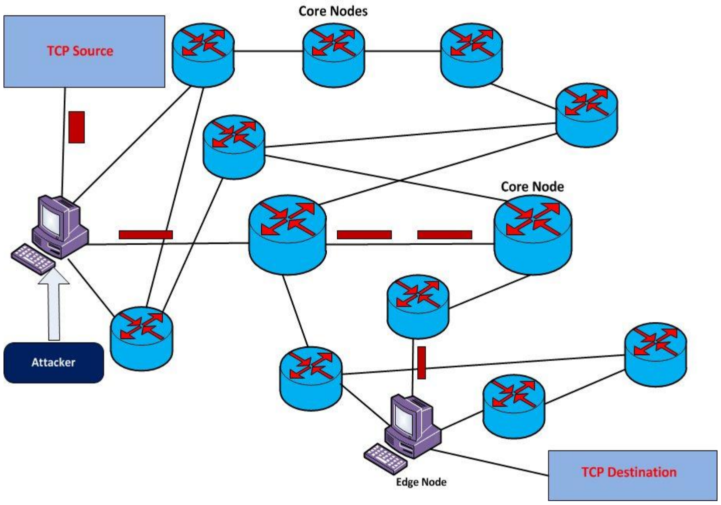


Figure 2.4.5.2 Burstification Attack at the Edge Node

2.5 Research Motivation

To meet the ever growing demand of bandwidth, copper cables are replaced by optical fibers in both the access networks as well as in the backbone networks. Optical fibers not only support huge bandwidth and also have other advantages, it has lower bit-error rate, no interference problem and security advantages without physical damages. Wavelength Division Multiplexing (WDM) technology, is deployed in optical networks, which divides the available bandwidth of the fiber into number of non-overlapping wavelength channels each operating at electronic speed and allows tens or hundreds of wavelength channels to be transmitted over a single optical fiber at a rate of 10 Gb/s/channel and beyond. This means that the rate can reach 10 Tb/s in each individual fiber.

To carry IP traffic over WDM networks three switching technologies exist; they are Optical Circuit Switching (OCS), Optical Packet Switching (OPS) and Optical Burst Switching (OBS). OCS and OPS have their own limitations when applied to WDM networks. OCS is not suitable for carrying bursty IP traffic with time-varying bandwidth demand. In addition, delays during connection establishment and release increase the latency especially for services with small holding times.OPS, which can adapt to changing traffic demands and requires no reservation, but the optical buffering and signal processing technologies, have not matured enough for possible deployment of OPS in core networks in future.

In this context OBS is the emerging and alternative switching technique, which combines the strengths and avoids the shortcomings of OCS and OPS. With rapid deployment of OBS, security has become one of the major problems that optical networks face today. To secure OBS networks, the possible attacks can be either prevention or detection mechanism, or combination of both strategies have been dealt. This thesis concentrates on attack mitigation for secure optical burst switched networks.

2.6 Problem Description

Optical networks are viable network for future communications, which is used to transport the data by the help of dedicated optical routers. Optical Burst Switching (OBS) is a technology for optical networks to cater the huge bandwidth demands. There is good amount of research in the area of security in optical networks. Also, the issue related to physical network security has been dealt. However, from our literature survey found that there is only limited amount of work has been done related to security issues in optical switched networks. This thesis concentrates on security issues in optical burst switched networks and identifies and categories critical attacks into Espionage attacks, Denial of Service attacks and Service Disruption attacks their possible countermeasures.

2.7 Research Objectives

The objectives of this research are.

- To detect and categorize Espionage, Denial of Service and Service Disruption attacks that may occur in optical burst switched networks.
- To provide mathematical solutions for mitigating attacks like Burst Hijacking Attack, Fake Spectral Attack, Burst Flooding Attack, Timeout Attack, Burst Circulating Attack and Land Attack.
- To study and analyze the performance of the secure optical burst switched networks under the impact of the attacks.
- To measure the performance of the optical burst switched networks in terms of network related metrics like Burst Throughput, Propagation Delay, Average Goodput and Burst delivery ratio.

2.8 Summary

In this chapter, we have discussed the detailed survey about security issues in regular network and optical networks and also have dealt with the physical security issues followed by a brief summary of vulnerabilities in OBS networks like orphan burst, fake burst header attack, reply attack, burst tapping attack and burstification attack and also dealing with critical issues in OBS networks.

CHAPTER 3

OBS: SIMULATION SETUP

3.1 Introduction

A research work is never complete without experimentation. In order to experiment an algorithm, an appropriate network simulator or actual networking hardware is required. In this chapter, the simulation method is mentioned first, followed by the architecture of a network simulator. OBS patches for ns2 are surveyed in order to choose the best simulating patch for ns2 and the modifications upon the same patch is dealt in the final phase.

Experimentation on an OBS Network can be achieved either upon real test-bed experimentation or by network simulators. Test bed implementation is costly that requires real optical nodes and optical links. Certain works has been done in OBS and experimented in real test fields called as OBS Test bed. Few multi-QoS traffic transmission experiments are carried out through test bed implementation as discussed in [79]. In OBS, network performance is tested in field trials on the Japan Gigabit Network II (JGN-II) test bed. Fast congestion control passes through the use of pre-calculated detour routes which were already demonstrated. Apart this, demonstration of service aware bandwidth reservation in a multi-granular OBS test-bed has been done. And also, demonstration of Novel Multi-Granular Switch Architecture on an Application-Aware End-to-End Multi- Bit Rate OBS has been done.

The second option to implement an OBS network is through simulation. This is achieved using network simulator. Network simulator (ns) is an important tool for researchers and people from academia to simulate and model the actual network at a reduced cost compared to the real test-bed experimentation. It is discrete and event-driven simulator developed at UC Berkeley that simulates variety of IP networks. It implements network protocols such as TCP and UDP. The variants of TCP can also be implemented in this simulator.

TCP and UDP are aliased as agents which are attached with the traffic sources. Here, traffic sources behavior's such as FTP, Telnet, Web, CBR and VBR are aliased as applications. So each source node under implementation is attached with an agent and further with a particular traffic source. Further router queue management mechanism such as Drop Tail queues, Random Early Detection (RED) and Class Based Queuing (CBQ) and source routing algorithms can also be modeled. NS also implements multicast traffic routing algorithms and some of the MAC layer protocols for LAN simulations.

3.2 Network Simulator Architecture

The NS project is a part of the VINT (Virtual Inter Network Test bed) project that develop a network simulator that will allow the study of scale and protocol interaction in the context of current and future network protocols. Currently, NS (ver. 2), abbreviated usually as ns2, written in C++ and OTcl (Tcl script language with Object-oriented extensions developed at MIT) is available [80] [81].

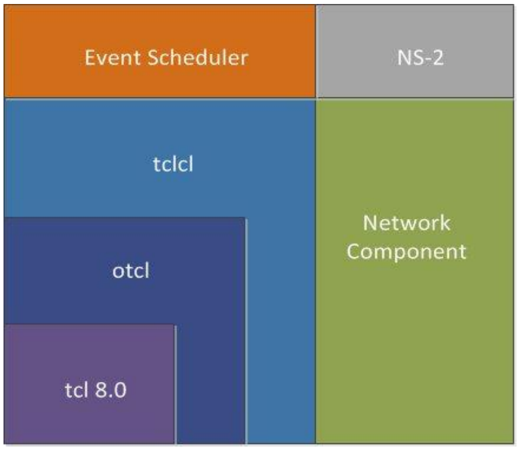


Figure 3.1 Architectural view of NS

The architectural view of network simulator is shown in Figure 3.1. The programmer or the general user works on tcl 8.0 and runs simulations with the assistance of otcl library. The general user is not/need not be the simulator developer. All the network components including the event scheduler is implemented in C++. These are used as otcl through an OTcl linkage that is implemented using tcl. This forms the architecture of ns2 which is Object Oriented concepts extended with the Tcl interpreter and network simulator libraries.

3.3 OBS Frameworks

The ns2 (by default) cannot aid to simulate optical nodes or optical links. Suitable OBS frameworks must be patched to the simulator to patch the same. There are number of simulators available to simulate OBS network. They are discussed below.

3.3.1 NCTUns

NCTU network simulator (National Chiao Tung University network simulator) is a module based network simulator developed at the Network and System Laboratory at National Chiao Tung University in 2002 [82]. Since it has module based structure, it supports real network applications. Sim -Real Incorporative is a virtual company that promotes the use of the NCTUns and emulator used for Network planning, research, application program, performance evaluation and educational needs, since its inception. This simulator deals mainly the OSI Data link and network layer protocols such as routing and switching. In the OSI physical layer, it simulates the Bit Error Rate (BER), down time periods, bandwidth and propagation delay. For All-Optical Networks, it caters two simulation environments: the first one is a traditional OCS network, while the other one is the OBS network. It can support both wired and wireless communication links. An environment implemented in C++, it is an event driven simulator. NCTUns 4.0 (latest version) was tested on Red-Hat Fedora 7 Linux OS with kernel version 2.6.21.

3.3.2 OBS Simulator

At North Caroline State University, USA, this network simulator was developed by Tend and Rouskas to model and simulate OBS networks, using C++. Based on the Burst Loss Probability (BLP), simulation of different resource reservation protocols can be implemented to study the performance of OBS, including network topologies. It defines the topology, scheduling algorithm, the number of edge nodes per core, the percentage of edge nodes that can generate bursts per core node and the propagation delay between core nodes. On command line, the program is called to give option parameters about the reservation protocols such as JIT, JET, Horizon, JIT+ and Jumpstart, the inter-arrival process (time between events), the burst generation ratio per node, the number of available data channels per link, the number of wavelength converters in each node, the time to process the setup message, the time to configure the optical cross-connect, and the propagation delay between edge and core nodes. Here, simulation results are only presented on console and plotting graph is not available for advanced data output analysis. The same happens with animation or real-time viewing, which are not supported.

The disadvantage includes the unavailability of GUIs, usage of a fixed file name *"n1.net"* (created by its developers) with a particular semantics; syntax to run any simulation and the absence of Network animation output view [83 – 85].

3.3.3 OPNET

OPtimized Network Engineering Tools (OPNET) was founded by OPNET technologies inc. (NASDAQ:OPNT). It could model and simulate simple, single core node model and multi core nodes. It is implemented using Object oriented programming and it is a discrete event simulator. The most vital feature of this simulator is that it uses GUI based debugging and analysis and does cater 32 and 64-bit fully parallel simulation. Also, OPNET accelerates the R&D process for analyzing and designing network; catering optical buffering (implementation of FDLs); wavelength conversion and different assembly algorithms. Figure 4.2 shows the simulation output of a campus network using OPNET.

3.3.4 JAVOBS

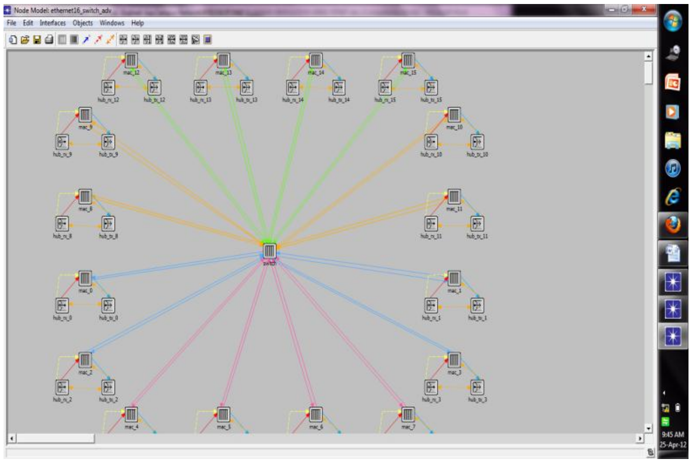


Figure 3.3.4.1 Campus Network Simulation on OPNET

JAVOBS is developed by Oscar Pedrola from Technical University of Catalonia, USA. It is a Java based Simulator exclusive to OBS network simulation that uses JAVANCO framework [56] shown in figure 3.3.4.1. Every OBS signaling protocols (such as JET, JIT, Horizon, E-JIT, JIT+) can be modeled here using hybrid discrete event modeling. Since this simulator is based on JAVA, which is flexible and have the complexity to build the simulator is reduced. Both C-OBS and E-OBS architectures can be implemented. This simulator is used for comparison of reservation protocols performance in C-OBS and E-OBS.

The main disadvantage is that runtime performance of JAVOBS is worse than those simulators implemented by C++ because of time consumed for the temporary compilation of user code to byte code. This code does not become executable code until the program is actually run.

3.3.5 OBS-ns Simulator

The OBS-ns simulator was the product of DAWN Networking Research Lab from University of Maryland. A redesigned version of DAWN Lab's simulator is the Optical Internet Research Center (OIRC) OBS-ns Simulator, a simulation tool developed by Optical Internet Research Center and Samsung Advanced Institute of Technology [86]. It is a redesigned version of OWns, which uses an older version of ns-2. It aims to solve the problems of version 0.9 and improve the software, thus introducing new features. As its predecessor, OIRC OBS-ns is an event-driven simulator built on ns-2. Here it is necessary to build a routing table using shortest path routing and specify traffic streams as done in ns-2. There is also an extended list of OTcl parameters with default values that is to be changed to specify details about OBS, such as the delay used of in FDL at end of link, the size of BHP and DB overhead, the burst timeout, the maximum burst size, the offset time for class and the edge node electronic buffer size.

The main disadvantage is the lack of documentation about resource reservation protocols supported in this simulator. Furthermore, OBS-ns simulator implements only shortest path routing.

3.3.6 ADOBS

AD-hoc OBS (ADOBS) event-driven simulator implemented in C++ language is an event driven simulator for OBS networks. The main motive behind creation of such a simulator is to study the performance of OBS Network layer. Protocols such as JET, Horizon can be simulated using this tool [87].

The disadvantages include complexity cost that is used to achieve speed and efficiency. All the concepts and operations are entirely dependent on the system's hardware. The input file is strictly predefined and not scripted unlike other network simulators.

3.3.7 DESMO-J

Developed in 1989 at the University of Hamburg, in the context of student's projects, DESMO-J (Discrete-Event Simulation and MOdeling in Java) is an object-oriented framework can be used for modeling OBS. This Java implementation supports the discrete-event simulation i.e.., all system state changes are supposed to happen at discrete points in time like queuing networks. The DESMO was done in Modula-2, was an inspiration of DEMOS, a package for discrete-event simulation in Simula that was earlier developed by Birtwistle in 1979. In the year 1999, the core DESMO-J framework was completed. It was then extended in various aspects, like providing special components for the simulation of production systems or harbor logistics.

3.3.8 OWNS

OWns (Optical Wavelength division multiplexing network simulator) was created as an extension to network simulator. It is honored as the first ns2 based OBS network simulator. The OIRC and OBS-NS are redesigned versions of OWNs. Optical switching node, multi-wavelength links, routing can be simulated using this one. It was developed by studying other simulators such as REAL, OPNET, NS and BONES. The NAM Output for this simulator is called as OWNAM [88]. The vital demerit here is the possibility of implementation of very limited number of scheduling, assembly and routing algorithm for OBS networks. To counter these demerits later was OIRC OBS-ns and nOBS were created.

3.3.9 OBSIM

OBSim (Optical Burst Switch Simulator) is a event driven network simulator and the events are messages [89]. These messages are generated by either by user or by network node. OBSim make use of Java and is designed to implement OBS networks in form of objects [62]. It further allows to assess and compare the performance of signaling protocols and load profiles to a given network topology. This simulator is used to measure the performance of an OBS networks and a tool to predict the behavior of OBS network.

3.3.10 nOBS

In OWns and OIRC OBS-ns, only shortest path routing can be simulated. But in nOBS, any routing can be implemented using source routing. Here, the Wavelength Converters and FDLs are combined into pools and shared among all ports which called as Share-per-Node architecture.

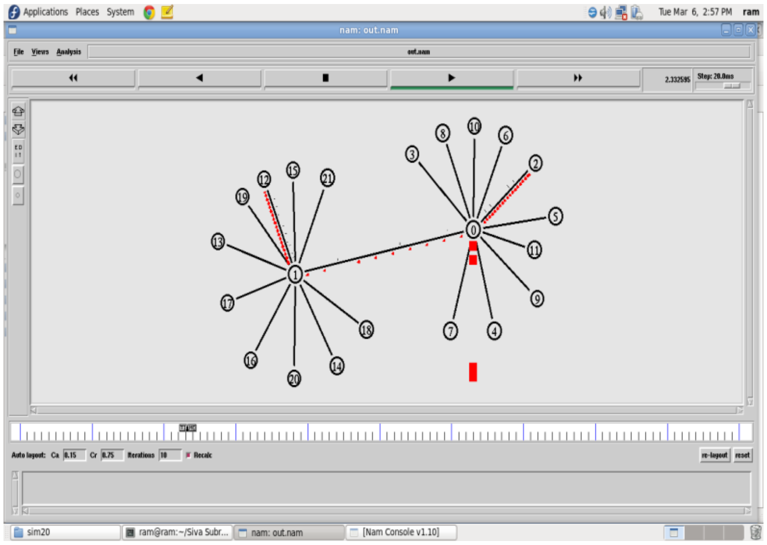


Figure 3.3.10.1 BS Network with dumbbell topology represented on nOBS

The nodes and links are modified into optical nodes and optical links. The address classifier is replaced by another one that differentiates TCP/IP packets from Bursts. The simulation output for nOBS is given in Figure 3.3.10.1 and Figure 3.3.10.2.

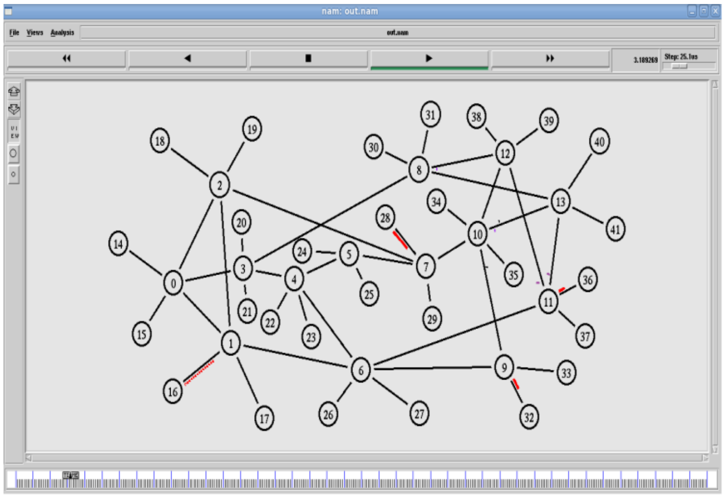


Figure 3.3.10.2 TCP/OBS Network in NSF topology represented on nOBS

3.3.11 OMNeT++

The primary function of OMNeT++ is not network simulation, but a generic discrete event simulator framework to create scenarios, from the hard disk operation to the behavior of an Ethernet network [90 - 91]. The important advantage of OMNeT++ is that it is open source and has an Academic Public License which makes it free for non-commercial use. Availability on all common platforms, like Windows, Mac OSX and Linux adds another advantage. All source code is in C++ and can be compiled with Microsoft Visual C++.Apart from the advantages discussed above, this simulator suffers from few disadvantages such as all OBS routing protocols cannot be modeled here.

3.3.12 IKRSimLib

The IKRSimLib is an object-oriented class library for event-driven simulation. It is implemented in C++ language. IKR Simulation Library V 2.5 is documented. First and foremost, it was designed towards powerful support of performance evaluation of communication networks. However, as it was flexible so been applied to assess software system performance of statistically evaluate traffic measurement traces. This simulator is employed, to simulate the Fiber Delay Lines.

IKRSimLib contains components like queue to servers. Added packages containing a TCP implementation like Reno, Vegas etc.., and packet scheduler models were developed on top of the library.

The merits of this simulator include simulation of asynchronous, non-periodic data transmission. Other benefits are GUI, Object-oriented design of simulation entities and easy extensibility. Again all OBS protocols cannot be modeled in this simulator which is a vital demerit of this simulator.

There are twelve known network simulators of TCP over Optical Burst Switched Networks. Among these six are compared with one another. The comparison of OBS Simulators is done in Table 3.1.

Table 3.1 Comparison of OBS Simulators

Simulators	OBS Protocols	language	License type	Input type	Output type	GUI
NCTUns	JET	C++	Commercial	Graphical Model Construction	Plot Graph, Animation	Powerful GUI
OBS-ns	JET	C++	Free Use	OTCL Script	Plot Graph	network animator
OBS Simulator	JET,JIT, JIT+ Horizon, Jump start	C++	Private Use	Script	Console output	NO GUI
ADOBS	JET, Horizon	C++	Private Use	Predefined input	Plot Graph	GUI
JAVOBS	JET,JIT, EJIT,JIT+ Horizon	JAVA	Private Use	Graphical, script, XML	Plot Graph	Powerful GUI
nOBS	TAW, TAG, JIT, JET	C++	Free Use	Script	X Graph, nam	network animator

The table does the comparison on OBS signaling protocols, Programming language used, the license type, Input and the output nature and the User Interfaces. This comparative survey could be helpful to spot the best network simulator for a particular scenario. In addition to the simulators discussed above; an adaptive threshold burst assembly is simulated using Visual C++.

3.4 nOBS Patch architecture

As discussed in the previous sections, the network simulator could simulate the OBS environment only with appropriate OBS framework.

The nOBS patch is preferred over other OBS patches because of the following reasons:

- The software is open source
- All routing algorithms in OBS can be implemented.
- Most of the scheduling, assembly and routing algorithms in OBS can be implemented.
- All TCP implementations are possible

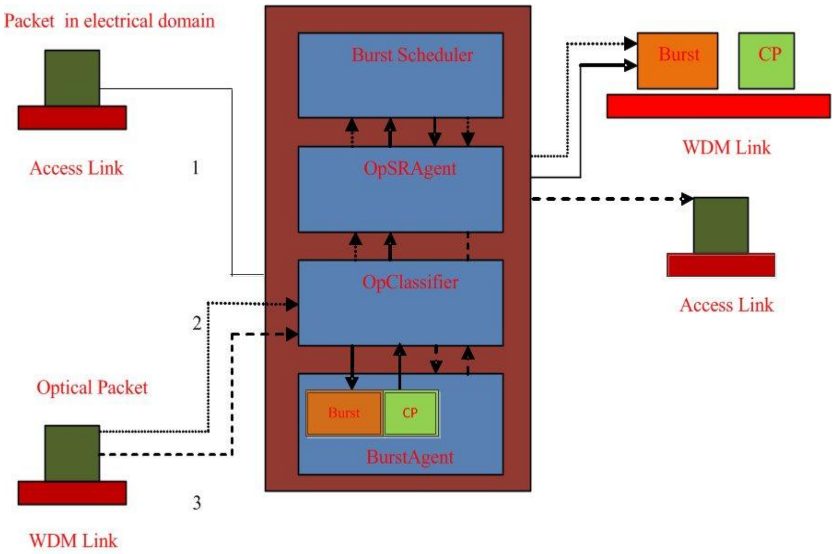


Figure 3.4.1 Architecture of Optical Node

A typical Optical Node architecture i.e. modelled by the nOBS patch is shown in Figure 3.4.1. There are three paths shown in dark, dotted and dashed lines. They are:

1. Burstification path starts with a packet in electrical domain arriving at the optical node through an access link *(shown in dark line).*
2. An optical forwarding path, i.e.., an optical packet is received by the OpClassifier through an incoming WDM link *(shown in dotted line).*
3. OpClassifier sends the optical packet to the Burst Agent for deburstification *(shown in dashed line).*

The packet is initially processed by Optical Classifier (OpClassifier). If the next hop for this packet is in the optical domain, OpClassifier forwards the packet to the Burst Agent. Burst Agent puts the burst data in assembly buffer that corresponds to a BCH.

When a burst is ready for transmission, its associated BHP is sent to OpClassifier and forward to Optical Source Routing Agent (OpSRAgent). OpSRAgent puts the optical domain routing information into the control packet and its corresponding burst. Then, OpSRAgent checks for a suitable time interval through the Burst Scheduler block. This block includes OpSchedule, OpConverterSchedule and OpticalFDLSchedule, which aids to keep records of the reservations on outgoing channels, wavelength converters and FDLs, respectively. If a suitable time interval is found, OpSRAgent forwards the BHP and schedule the corresponding burst to be transmitted after an offset time. Otherwise, the burst will be dropped.If an application running on ingress router, it contains data to be sent into the OBS network, the burstification path starts with OpSRAgent, where the route information for the packet is written, followed by the OpClassifier which will forward the packet to the BurstAgent. Required functionalities of optical nodes are divided into four separate modules. They are:

- Burst Scheduler
- OpSRAgent
- OpClassifier
- BurstAgent

All edge nodes and intermediate nodes of the OBS network require the same functionalities of the above mentioned modules. Therefore they share the same architecture.

In case of optical forwarding (path 2), an optical packet is received by the OpClassifier through an incoming WDM link. Since the next hop is in the optical domain, OpClassifier forwards the packet to the OpSRAgent, which produce a request to the Burst Scheduler block for a valid reservation. If the optical packet has the property to be a control packet and a reservation for the associated burst, then the control packet is forwarded to the corresponding WDM link. If the optical packet is a burst and a reservation has been already made, the burst is forwarded to the WDM link. Otherwise, the optical packet is dropped.When the next hop for an optical packet is not in the optical domain,

OpClassifier sends this optical packet to the BurstAgent for deburstification (path 3). If the optical packet is a control packet, it is dropped.

If it is a burst, then the packets inside the burst are sent to the OpClassifier, which forwards them to OpSRAgent. OpSRAgent sends these packets through outgoing electrical links towards their destination nodes.

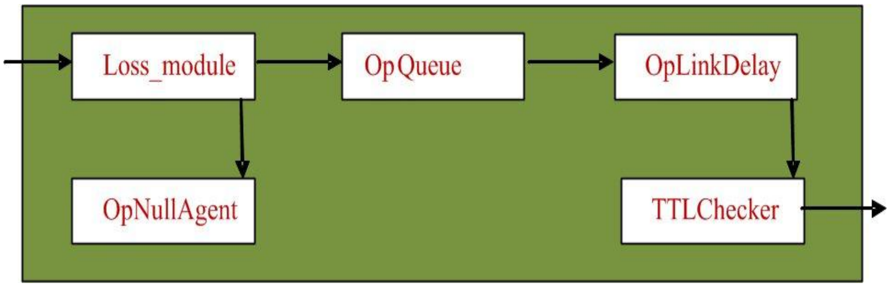


Figure 3.4.2 WDM Link Architecture in nOBS

The architecture of an optical link is shown in Figure 3.4.2. The OpQueue module immediately forwards all incoming packets to OpLinkDelay without any blocking, packet dropping or queueing since wavelength reservation, contention resolution and FDL buffering operations are already performed by Burst Scheduler and OpSRAgent in the node architecture. It was possible to remove OpQueue and connect loss module and OpDelayLink directly, but OpQueue is kept for easier implementation of future OBS architectures, which may need a queue component on the links. When the Loss module associated with the link determines that an optical packet must be dropped, the packet is sent to OpNullAgent component, which frees individual packets inside the burst. The operations on the link are memory-less and independent of the wavelength. Therefore multiple packets arriving at the same time on different wavelengths can be served without affecting each other.

OpClassifier would classify and forward the data packets within optical nodes. When an optical packet arrives, OpClassifier checks the type and destination of the incoming packet and handles it as follows:

a) If the incoming packet is not an optical burst and the packet's destination address is not the address of the present node, OpClassifier checks the source routing table of the packet. Looking up in the routing table of the packet, OpClassifier checks whether the packet's next node is an optic node.

If it is, the packet needs to enter the OBS domain. Furthermore, the node that owns this OpClassifier should act as an ingress node and apply burstification. Therefore, OpClassifier forwards this packet to the corresponding burstifier agent which called as BurstAgent. Moreover this, if OpClassifier realizes that this packet is coming from the BurstAgent after the deburstification process, it leaves the OBS domain. So, OpClassifier forwards this packet to the source routing agent that will forward the packet to the next hop over an electronic link.

b) If the packet is an optical burst and the packet's destination address is the address of the present node, it means that a burst has reached its destination. OpClassifier forwards the packet to the BurstAgent for the deburstification process.

c) If the packet is an optical burst and the packet's destination address is not the address of the present node, it means that this is a burst in transit. Therefore, OpClassifier forwards this packet to the source routing agent that will forward it to the next hop which is specified in the source routing table of the packet.

d) If the packet is not an optical burst and the packet's destination address is the address of the present node, it means that the packet is coming from the BurstAgent after deburstification process and the receiver of this packet is in the address of the present node. OpClassifier forwards this packet to the port classifier, which will forward the packet to its destination agent.

BurstAgent is responsible for the burstification of electronic packets and deburstification of optical bursts. A single BurstAgent is attached to OpClassifier in each optical node. When a new packet arrives from OpClassifier, BurstAgent checks whether this packet is an electronic packet or an optical burst. If the packet received from OpClassifier is an optical burst, BurstAgent disassembles the IP packets inside the payload of the burst and sends these IP packets back to the OpClassifier to be delivered to their destination agents. If the packet is an electronic packet, BurstAgent compares the source routing table of the packet with the list of nodes contained in the table opticnodes and finds the corresponding egress node from where this packet will leave the OBS domain. Next, BurstAgent inserts the incoming packet to one of the assembly queues responsible for burstifying packets destined for this destination egress node.

The assembly algorithm implemented in the BurstAgent is a hybrid size/timer-based algorithm that keeps track of the size of the burst and the delay experienced by the first packet in the burst. BurstAgent creates a burst when the delay of the first packet reaches a given timeout, or the number of IP packets in the burst reaches a threshold. The number of assembly buffers per egress router, M, can be between 1 and the number of flows, N, as shown in Figure 3.4.3. An incoming packet is forwarded to a per egress burstifier queue group based on the routing information, and it is classified further into an assembly buffer based on the flow ID depending on N and M. If an incoming optical packet is the first packet in the assembly queue, BurstAgent starts the burstification delay timer. When the burst is ready for transmission, BurstAgent creates a control packet carrying all the necessary information for this burst. Before sending the burst, BurstAgent copies the packets in the assembly queue to the burst's payload. Then, BurstAgent sends the control packet to OpClassifier.

On other side, OpSRAgent is responsible for adding the source routing information to packets, forwarding the packets to links according to the routing information, and controlling when and how to send optical packets using FDLs and wavelength converters. When OpSRAgent receives a packet, OpSRAgent first checks whether source routing information is available in the packet header and whether this packet is an optical burst or a control packet.

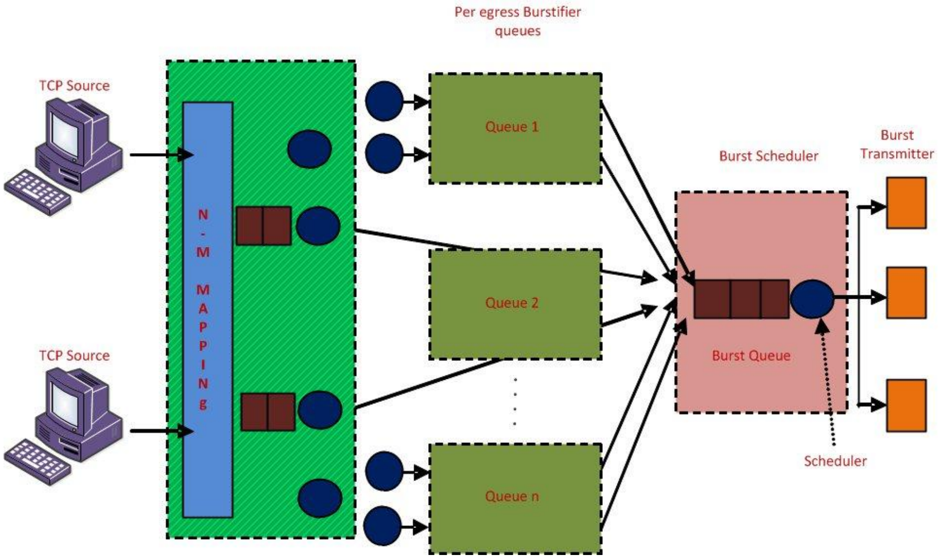


Figure 3.4.3 Ingress Node Model

If there is no source routing information in the packet header, OpSRAgent considers two scenarios:

a) If this packet is an electronic packet, OpSRAgent writes the routing information to the header of the packet. Then, OpSRAgent checks whether the next hop is an optical node in the same OBS domain. If this case arises, OpSRAgent sends the packet to OpClassifier, which forwards the packet to the BurstAgent for burstification. On other side, if the optical node is the egress node for this packet, OpSRAgent forwards the packet to the next node on an electronic link.
b) If this packet is an optical burst, it means that OpSRAgent has received a newly created burst and control packet pair, so OpSRAgent writes the routing information to the header of both the control packet and the burst.

After ensuring that the source routing information is available in the packet, OpSRAgent checks whether the current node is the destination of this packet. If this is the case, OpSRAgent send this packet to the OpClassifier. On other side, if it is an electronic packet, OpSRAgent sends the packet to the next hop via an electronic link. If this is an optical packet, OpSRAgent tries to send it to an optical link after checking the schedulers. First, OpSRAgent checks the scheduling on this wavelength and link by sending the packet to OpSchedule. OpSchedule returns a result depending on the type of the packet and availability of the channel.

If the packet is a control packet, OpSRAgent takes the following actions based on the result received from the OpSchedule:

a) If there is no contention, OpSRAgent sends the control packet to the optical link for transmission immediately. If this is the first hop of the control packet, OpSRAgent sends the burst corresponding to this control packet to the optical link after delaying the burst for H∆, where H is the number of hops to be traversed by the burst and ∆ is the processing delay per hop.
b) If there is a contention, OpSRAgent checks whether there are unused FDLs or wavelength converters available at the node. If there is, OpSRAgent retries the

reservation request, by applying different combinations of available FDLs and converters and chooses the best schedule, if any, according to the scheduling algorithm. OpSchedule learns the availability of FDLs and converters from OpConverterSchedule and OpFDLSchedule, respectively, which are described below. If available, FDLs or converters cannot resolve the contention, due to this OpSRAgent drops the control packet.

If the packet is a burst, OpSRAgent takes the following actions based on the result received from the OpSchedule:

a) If there is a reservation for the burst without any contention, OpSRAgent sends the burst to the optical link. If there is a required FDL delay specified in the reservation, OpSRAgent delays the burst before sending to the optical link.
b) If there is no existing reservation for the burst, i.e., the control packet could not succeed in making a reservation for the burst, OpSRAgent drops the burst.

Every optical node in the framework holds a record of the outgoing channel reservations, shared FDLs and wavelength converters that are available at the node. OpSchedule holds reservations on outgoing channels while OpConverterSchedule and OpFDLSchedule maintain schedule for wavelength converters and FDLs, respectively. The wavelength converters and FDLs at each node are combined into pools that are shared among all ports at the optical switch. This is called as a share-per-node architecture.

At the ingress, bursts may be kept in the electrical buffers until they are scheduled and then sent into the optical network. If OpSRAgent cannot find a suitable interval for the burst, it checks possible combinations of wavelength converters and FDLs depending on the node type. If a burst cannot be scheduled, it is dropped. OpSchedule class is responsible for keeping, checking and making reservations on all wavelengths of all links. OpSchedule is connected to the OpSRAgent. When OpSchedule receives an optical packet from the OpSRAgent, it first checks the type of the packet.

If the packet is a control packet, OpSchedule tries to do a reservation for the burst specified in the control packet and returns whether reservation is successful or not. If the packet is a burst, OpSchedule searches for a reservation in its reservation table, which is made earlier by the control packet, and returns whether there is a valid reservation or not. OpSchedule uses Latest Available Unscheduled Channel with Void Filling (LAUC-VF) or Minimum Starting Void (Min-SV) scheduling algorithms in combination with Just Enough Time (JET) signaling.

Here, OpSchedule uses a linked-list for storing the reservation list and also responsible for calculating and updating the delay between the control and burst packets. OpConverterSchedule and OpFDLSchedule are very similar to OpSchedule. These two schedulers are connected to the OpSRAgent, and they are responsible for keeping, checking and making reservations of converters and FDLs at the corresponding nodal pools. They inform the OpSRAgent when OpSRAgent asks for availability in the specified timeline.

It is possible to choose whether multiple bursts on a wavelength can use the same FDL subsequently, but the second burst may enter the FDL before the first burst leaves the FDL. An important difference between these two schedulers and OpSchedule is that when OpSRAgent sends a control packet to the OpSchedule and if reservation is possible, OpSchedule does the reservation directly.

However, OpConverterSchedule and OpFDLSchedule require a parameter called action. When a control packet is send to corresponding schedulers and if action variable is set to be zero, then these schedulers return only whether reservation of converter or FDL is possible or not. They do not do the reservation, unless action variable is set one.

This is because the scheduling algorithm may use a combination of FDL and wavelength conversion for resolving the contention and the OpSRAgent must make sure that both the queried FDL and converter are available. If both schedulers return an affirmative reservation signal, then OpSRAgent informs the schedulers to perform the actual reservations.

3.5 Modifications on nOBS patch

This section deals with the modifications done on nOBS framework. The nOBS patch architecture, as described in the past section, shows four functional components/modules namely Op Scheduler, Op SRAgent, Op Classifier and a Burst Agent. All these components are loaded as source files in C++ with included header files. The installation procedure of the patch is given in detail in appendix 1. After installation there are modifications to be done on this patch to cater the node capability based multicasting. The scripts are written in tcl to model multicast network for the standard NSF network configuration as given in appendix 2 (a).

The first change that is to be made upon the nOBS patch is on the OpSRAgent module. The optical nodes defined in the existing source code were not configured for multicast traffic routing and they can only unicast a data burst. The Virtual source capability (i.e., splitting and Wavelength Conversion capability) is added to those random nodes (selected by a random number variable in tcl script) that are considered as VS nodes. The number of VS nodes in the OBS network is governed by the script that is run on ns2. Two variables for splitting and Wavelength conversion for every optical node are to be declared in the corresponding headers and must be defined in the corresponding source code. If particular node carries corresponding capability, then the value is set to ‘1’, otherwise it is ‘0’. The wavelength conversion variable must be passed to the Op ConverterScheduler module every time when a wavelength conversion is initiated for a particular data. After making these changes, the terminal commands ‘make clean’ and ‘make’ are to be executed to apply changes to the framework.

To simulate fake spectral threat in nOBS framework, it is required to change code module of the agent.cc file. The agent.cc file models the Burst agent component for nOBS framework with an associated header file. Six wavelengths are modelled initially for burst data transmission in OBS network (with values such as 1,2,3,4,5 and 6). To invoke this particular threat, the outgoing channel is set to ‘1’ for all bursts for the affected node. This scenario will start burst dropping subroutine due to wavelength unavailability. The affected node is identified using a newly declared variable ‘compromisedVSNode’.

The burstfication threat can also be simulated in a similar way by altering code module of the agent.cc file. There is another variable that must be newly declared as ‘MaxBurstlen’ that will have a huge value for the size of the burst. For a compromised node of the Burstification threat, the Burstification size of outgoing bursts is assigned to ‘MaxBurstlen’. This scenario will change the burst threshold value to a huge value thus forming massive sized data bursts. After making these changes, the terminal commands ‘make clean’ and ‘make’ are to be executed to apply changes to the framework. The modifications done on source code in nOBS patch for the node capability based routing and the security threats are given in appendix 2(b).

3.6 OBS Network Simulation Assumptions

The simulations consisted of creating sample demands and then running various routing schemes on them. NS2 and modified nOBS patch were at the heart of our simulation models. First of all, we created an NS2 code that reads a NSFNet work topology as a list of links and then finds all the possible paths for all the OD (Origin-Destination) pairs in the network.

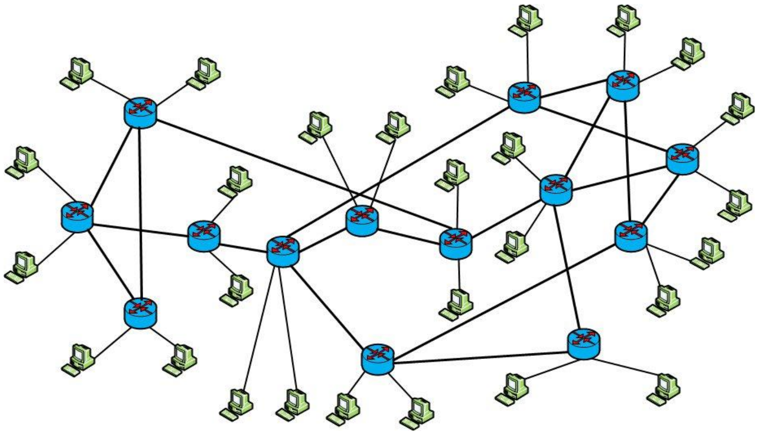


Figure 3.6 NSFNet Topology Network Model

For simulation purposes, we only used the 14 optical nodes and 28 electronic nodes. The optical network is modelled with 1Gbps bandwidth and 10ms propagation delay. The TCP/IP links have 155Mbps bandwidth each with 1 ms link propagation delay. The Figure 3.6 shows the NSFNet topology and the below Table 3.2 represents the assumptions for the proposed simulation.

Table 3.2 Simulation Assumptions

Content	Assumptions
Topology	NSFNet
Number of Optical Nodes	14
Number of Electronic Nodes	28
Number of TCP/IP Connection	10
Maximum number of attacker nodes	03
Maximum number of packets	200
Maximum Lambda	20
Link Speed	1GB
Switch Time	0.000005

3.7 Summary

Over the years, various methodologies were employed for network implementation, namely real test-bed implementation and simulation. Due to high cost involved to import optical node and link components there is a need for researchers to shift their attention to potential OBS simulators. The architecture of a network simulator was presented in this chapter, which required appropriate patches to simulate OBS networks. A dozen network simulator patches are also discussed in this chapter. The OBS frameworks were compared with these patches. Among that, nOBS patch was chosen as optimal one to simulate multicast traffic routing in OBS networks. In this chapter, nOBS architectural components are also discussed in detail. The modifications done on the nOBS patch is also mentioned in the last part of this chapter. The simulation results done for this work are simulated using the modifications done on nOBS patch in ns2. Besides these, next chapter deals with the proposed work.

CHAPTER 4

ESPIONAGE ATTACKS FOR OBS NETWORKS

4.1 Introduction

Optical network communication is definitely the future communication technology due to its massive bandwidth. But, when the vulnerability of the Optical Burst Switched network is exploited through attacks, the performance and reputation of the network becomes critical. Especially, the class of attack called Espionage attack drastically reduces the network reliability. Hence, the need for proposing mathematical strategy based mitigation solution becomes essential. In this chapter, two types of Espionage attacks viz., Burst hijacking attack and Fake spectral attack are elaborated with their associated mathematical model for mitigation. Further, the mitigation algorithms proposed for each of the Espionage attack with a brief explanation is also presented. The lethality of the attacks are depicted and studied through simulation experiments conducted based on network related parameters like Burst blocking probability, Average Goodput, Burst loss probability and Burst throughput are also demonstrated.

4.2 Burst Hijacking Attack

Generally, in optical networks during data transmission, Burst Control Header (BCH) is converted from optical form to electronic form and is processed at every intermediate core node. The core node receives the BCH and establishes the path for each and every corresponding Data Burst (DB) and forwards to next intermediate optical node until it reaches the egress node. To support multicast routing in WDM optical networks, Virtual Source (VS) nodes are essential. This Virtual node is an optical node which has light splitting capabilities as well as wavelength conversion capability which can transmit an incoming burst to multiple destinations on any wavelength.

In case of the Burst Hijacking Attack, if a compromised optical Virtual node receives the BCH, it maliciously creates a copy of original BCH and modifies the value to establish a malicious path to the destination. Then, the corresponding DB instead of the travel to the original destination as shown in Figure 4.2, it travels to the malicious destination. The malicious destination does not sent the acknowledgement for this hijacked burst and it escapes from being caught. Thus, it compromises the authentication of DB and introduces denial of services.

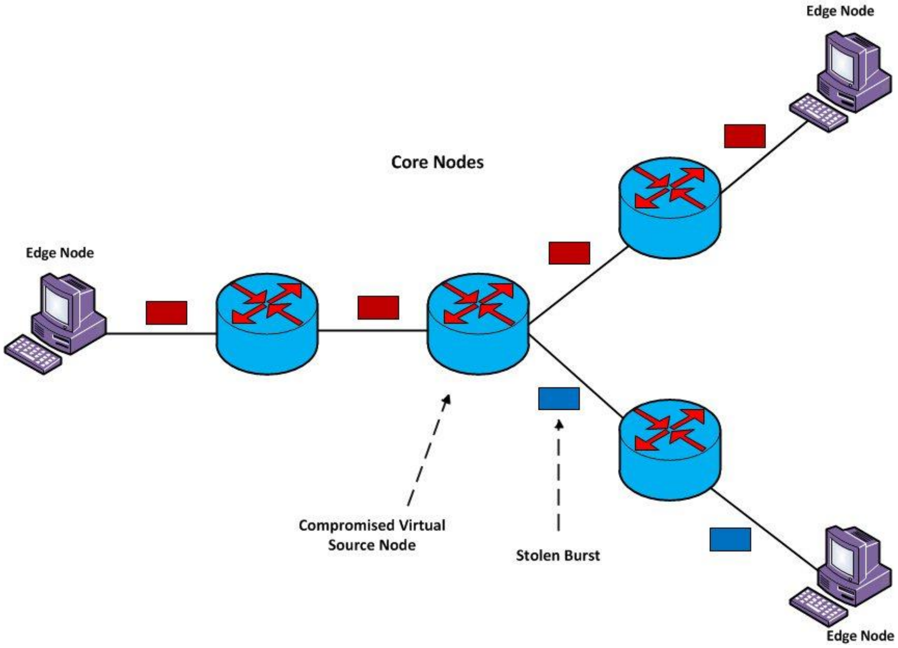


Figure 4.2 Burst Hijacking Attack

4.2.1 Attack Detection and Counter Measures

To detect the burst hijacking attack, the intermediate core nodes in the Optical Burst Switched network are monitored. The monitoring work is achieved through the trusted monitor node. The trusted monitor node remains static throughout the whole session. When a session is started, every intermediate node sends the network traffic statistics to the monitor. The monitor node analyzes these statistical values associated with each and every core node and checks whether any node in the routing path acts maliciously based the Cornbach Alpha factor based mathematical model as presented in the section 4.2.1.1.

Further, based on the statistics report, determine the number of burst packets received by a node and the number of packets forwarded that node for ‘k’ sessions. Then, estimate the number of burst packets dropped by a node $(P_{drop(i)})$ and the average packet drop P_d in the monitored node for ‘k’ sessions. Furthermore, calculate the probability of a node in delivering packets is denoted by P_f.Finally, Manipulate the Cornbach Alpha Factor ‘α’ based on the values of P_d and P_f, in each burst transmission and if the node’s trust value reaches below threshold, inform other nodes about maliciousness. In addition, The trusted monitor node analyses network traffic and verifies whether a new connection is established between the virtual source node with a designated burst_id and also confirms with the other statistics required for performing the one to one matching with the source and destination.

4.2.1.1 Cornbach Alpha Factor Based Model

Cornbach Alpha Reliability Coefficient Based Mitigation Model mitigates the Hijacking attack is based upon a factor called Cornbach Alpha factor. This Cornbach Alpha factor aids in estimating the trust worthy factor of the monitored node and enables effective and efficient mitigation of Hijacking attack from the routing path established for multicasting. The details of the detection mechanism are as follows:

Let us consider the number of burst packets received by a node shall be $R_{P1,}R_{P2,}R_{P3,\dots}R_{Pn}$ and the number of packets forwarded by that node as $F_{P1,}F_{P3,\dots,}F_{Pn}$ for ‘k’ sessions.

The number of burst packets dropped by a node in any particular session says in session $'i$, can be given in (1),

$$P_{drop(i)} = R_{Pn(i)} - F_{Pn(i)} \qquad (1)$$

Then, the average packet drop in ‘k’ sessions is computed by (2),

$$P_d = \sum_{i=1}^{k} \frac{P_{drop(i)}}{k} \qquad (2)$$

Further, the probability of a node in delivering packets is denoted by P_f and given by (3)

$$P_f = \frac{F_P}{R_P} \qquad (3)$$

Based on the values of P_d and P_f, the variance and the total variance in packet delivery of the core node in each burst transmission is given in (4) and (5)

$$V_k = P_f\left(1 - P_f\right) \qquad (4)$$

$$V_{tl} = \sum(P_d - \bar{P}_d)^2 \qquad (5)$$

where, $\bar{P}_d$ represents the packet drops in each session.

Then, the Cornbach Alpha Factor is computed as given in (6),

$$\alpha = \frac{k}{k-1}\left(1 - \frac{\sum V_k}{V_{tl}}\right) \qquad (6)$$

When the manipulated value of ‘α’ is comparatively lower than the threshold, the Burst hijacking attack can be identified and mitigated.

If the node’s trust value reaches below threshold, inform other nodes, then the ingress node is affected by Burst hijacking attack and it is given in Algorithm 4.1.

Algorithm 4.1 Burst hijacking attack detection Algorithm

```
recv (struct *PACKET packet)
{
    A) Determine node Type from packet
        if ((node Type=’intermediate core node’)
           OR
        (node Type=’egress node’))
{
    B) Extract burst id, source, destination, num_of_packets, burst_size from packet.
    C) Create new packet and store the extracted information inside the new packet.
    D) Send the new packet to the trusted node.
}
else if(node Type==’trusted_node’)
{
    A) Extract statistics from packet.
    B) Insert the statics into the linked list based on burst id.
    C) Collect statistics report from trusted node
    D) Based on the statistics report, determine the number of burst packets received by a node ( $R_{P1,}R_{P2,}R_{P3,...}R_{Pn}$) and the number of packets forwarded the node $(F_{P1,}F_{P3,...,}F_{Pn})$ for ‘k’ sessions.
    E) Estimate the number of burst packets dropped by a node $(P_{drop(i)})$
    F) Further, estimate the average packet drop $P_d$ in the monitored node for ‘k’ sessions. Furthermore, calculate the probability of a node in delivering packets is denoted by $P_f$.
    G) Manipulate Cornbach Alpha Factor ‘α’ based on the values of $P_d$ and $P_f$, in each burst transmission
    H) If the node’s trust value reaches below threshold, inform other nodes about maliciousness. In addition, extract the source, destination and burst id from the linked list head.
    I) Finally, verify whether a new connection is established between the virtual source node and burst_id through other required statistics elucidated for matching with source and destination.
    J) Call, mitigation procedure for the Burst hijacking attack
    }
    }
```

4.2.2 Simulation Results and Analysis

The espionage security attacks like burst hijacking attack is simulated on a 14 node NSF network configuration with the simulation parameters given in Table 4.1 and with the assumption of a single random compromised node and it is explained in normal scenario, attack scenario and solution scenario.

Table 4.1 Simulation Parameters for burst hijacking attack

Number of Electronic Nodes	28
Number of Optical Nodes	14
Arrival Rate	0.01ms
Total Simulation Time	50 ms
Number of TCP/IP Connections	18
Number of OBS Connections	17
Number of Packets	200
Number of Channels	3
Link Speed	1 GB

The performance metrics used for studying the Cornbach Alpha Reliability Coefficient Based Mitigation Model proposed for Burst Hijacking Attack are burst blocking probability, Average Goodput, Burst loss probability and Burst throughput. The definition for the above mentioned performance metrics are as follows:

Burst blocking probability:

It is defined as the ratio of the number of burst data received by the destination during transmission to the number of burst data actually expected to be delivered at the destination.

Average Goodput:

It is defined as application level throughput, i.e. the number of useful information bits delivered by the network to a certain destination per unit of time.

Burst loss probability:

It is defined as the probability of number of burst data lost during transmission to the number of burst data expected to be delivered at the destination

Burst throughput:

It is defined as the gross bit rate that is transferred physically in to the reliable channel between the source and destination.

4.2.2.1 Performance Evaluation - Experiment 1

In experiment 1, the impact of Burst Hijacking Attack is studied with respect to time based on Burst blocking probability, Average Goodput, Burst loss probability and Burst throughput with respect to three scenarios namely Normal scenario, Attack scenario and Solution scenario are illustrated through Figures 4.2.2.1.1, 4.2.2.1.2, 4.2.2.1.3 and 4.2.2.1.4.

The following Figure 4.2.2.1.1 portrays that the burst block probability increases with respect to time for all the three scenarios viz., Normal scenario, Attack scenario and Solution scenario.

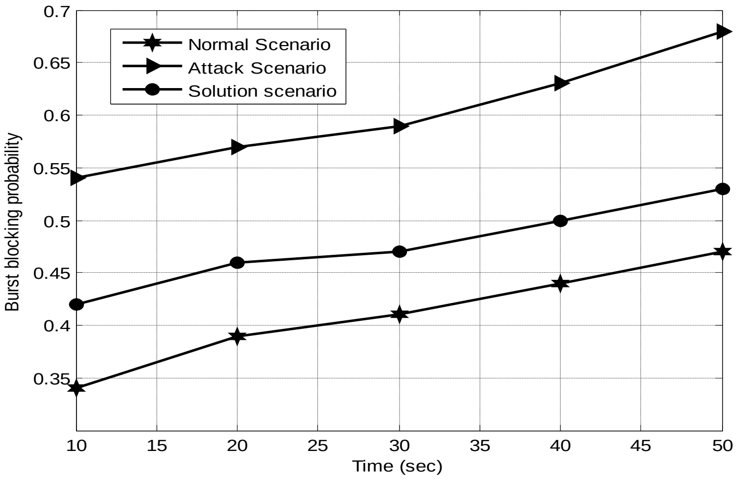


Figure 4.2.2.1.1 Burst Hijacking Attack – Burst blocking probability (Based on varying time)

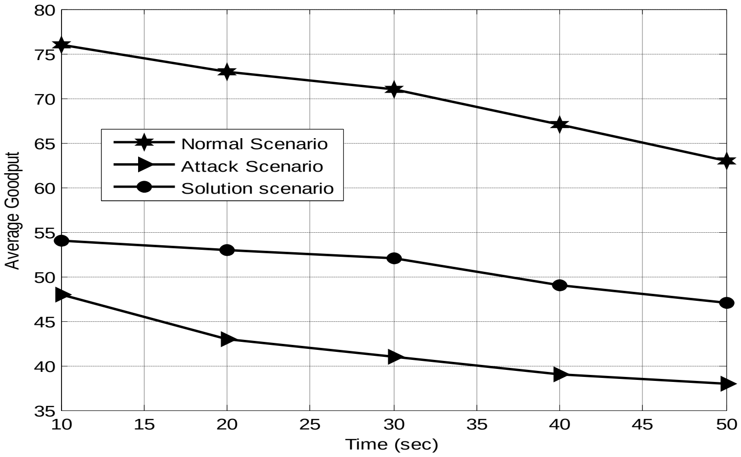


Figure 4.2.2.1.2 Burst Hijacking Attack – Average Goodput (Based on varying time)

It is evident that the burst block probability decreases drastically in the Attack scenario when compared to the Normal scenario to a maximum level of 35%. But, when the Cornbach Alpha Reliability Coefficient Based Mitigation mechanism is implemented, it decreases the Burst block probability by 23%.

Further, Figure 4.2.2.1.2 portrays that the Average Goodput delay increases with varying amount of time for all the three scenarios viz., Normal scenario, Attack scenario and Solution scenario.

It is also evident that the Average Goodput decreases drastically in the Attack scenario when compared to the Normal scenario to a maximum level of 23%. But, when the Cornbach Alpha Reliability Coefficient Based Mitigation mechanism is implemented, it increases Average Goodput by 18% with respect to varying time.

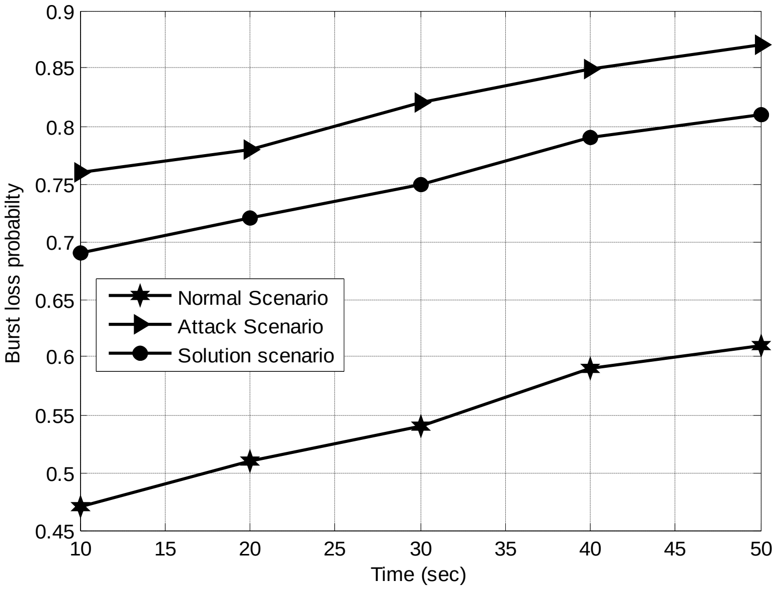


Figure 4.2.2.1.3 Burst Hijacking Attack – Burst loss probability (Based on varying time)

Furthermore, Figure 4.2.2.1.3 depicts that the Burst loss probability with respect to time increases for all the three scenarios viz., Normal scenario, Attack scenario and Solution scenario. It is also evident that the Burst loss probability drastically increases in the Attack scenario when compared to the Normal scenario to a maximum level of 21%.But, when the Cornbach Alpha Reliability Coefficient Based Mitigation mechanism is implemented, it decreases Burst loss probability by 23% with respect to varying time.

In addition, Figure 4.2.2.1.4 depicts that the Burst throughput with respect to time decreases for all the three scenarios viz., Normal scenario, Attack scenario and Solution scenario.

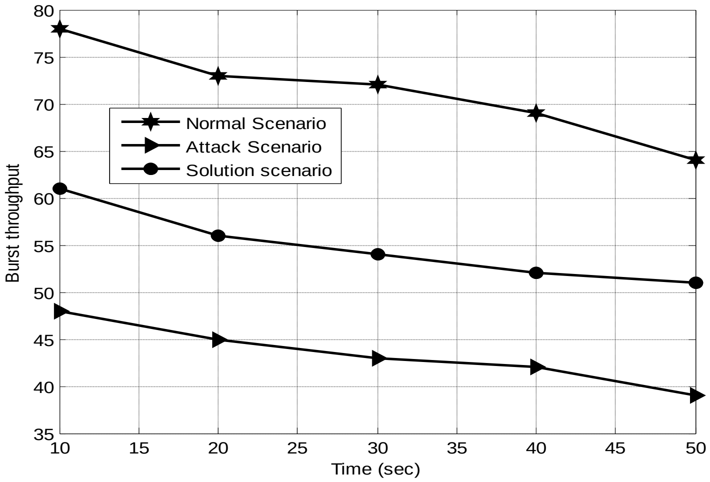


Figure 4.2.2.1.4 Burst Hijacking Attack – Burst throughput (Based on varying time)

It is also evident that the Burst throughput drastically decreases in the Attack scenario when compared to the Normal scenario to a maximum level of 28%.But, when the Cornbach Alpha Reliability Coefficient Based Mitigation mechanism is implemented and it increases the Burst throughput by 21%.

4.2.2.2 Performance Evaluation - Experiment 2

Further, In experiment 2, the impact of Burst Hijacking Attack is studied with respect to time based on Burst blocking probability, Average Goodput, Burst loss probability and Burst throughput with respect to three scenarios namely Normal scenario, Attack scenario and Solution scenario are illustrated through Figures 4.2.2.2.1, 4.2.2.2.2, 4.2.2.2.3 and 4.2.2.2.4.

The Figure 4.2.2.2.1 portrays that the burst block probability increases with respect to load for all the three scenarios viz., Normal scenario, Attack scenario and Solution scenario. It is evident that the burst block probability decreases drastically in the Attack scenario when compared to the Normal scenario to a maximum level of 31%.

But, when the Cornbach Alpha Reliability Coefficient Based Mitigation mechanism is implemented, it decreases the Burst block probability by 20%.

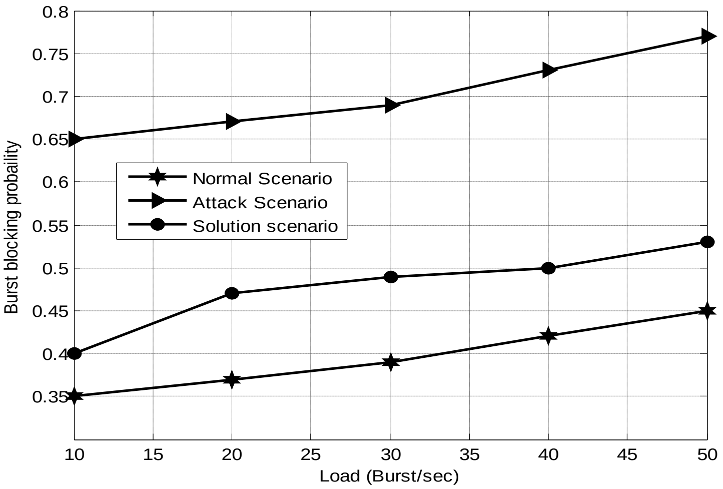


Figure 4.2.2.2.1 Burst Hijacking Attack – Burst blocking probability (Based on varying load)

Further, Figure 4.2.2.2.2 portrays that the Average Goodput delay increases with varying amount of load for all the three scenarios viz., Normal scenario, Attack scenario and Solution scenario.

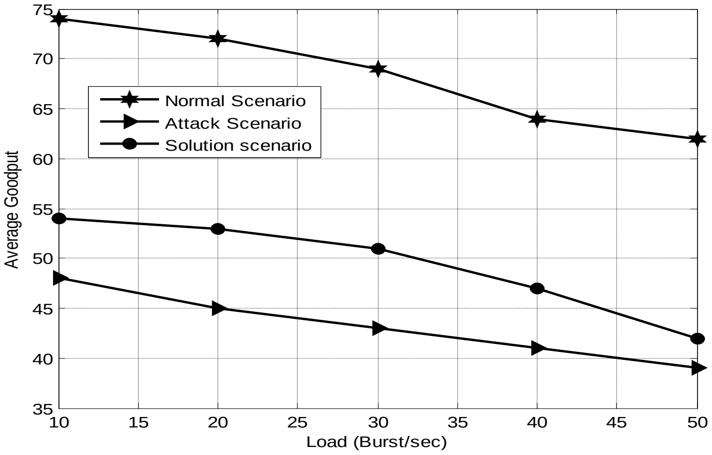


Figure 4.2.2.2.2 Burst Hijacking Attack – Average Goodput (Based on varying load)

It is also evident that the Average Goodput decreases drastically in the Attack scenario when compared to the Normal scenario to a maximum level of 19%. But, when the Cornbach Alpha Reliability Coefficient Based Mitigation mechanism is implemented, it increases Average Goodput by 16% with respect to varying load.

Furthermore, Figure 4.2.2.2.3 depicts that the Burst loss probability with respect to load increases for all the three scenarios viz., Normal scenario, Attack scenario and Solution scenario. It is also evident that the Burst loss probability drastically increases in the Attack scenario when compared to the Normal scenario to a maximum level of 18%.But, when the Cornbach Alpha Reliability Coefficient Based Mitigation mechanism is implemented, it decreases Burst loss probability by 21% with respect to varying load.

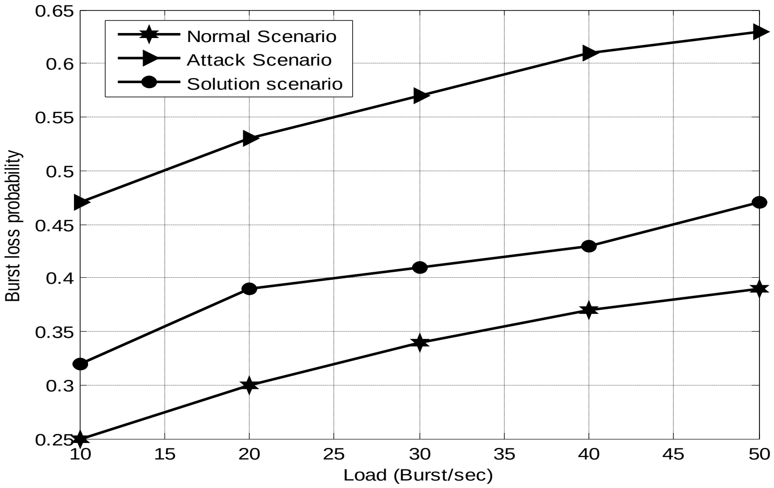


Figure 4.2.2.2.3 Burst Hijacking Attack – Burst loss probability (Based on varying load)

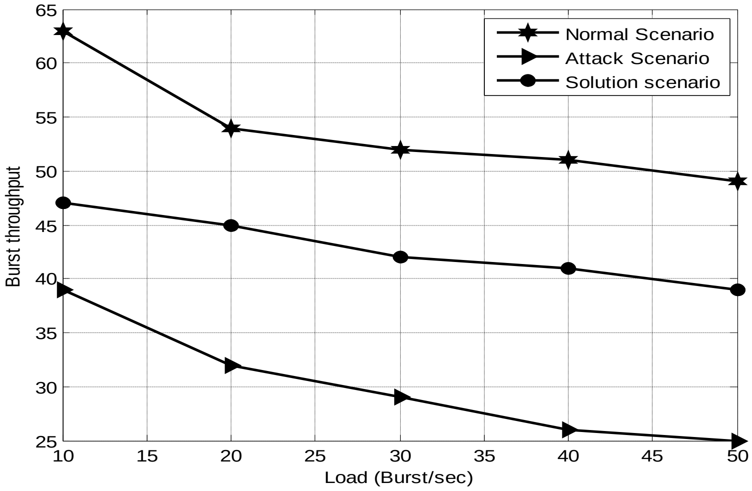


Figure 4.2.2.2.4 Burst Hijacking Attack – Burst throughput (Based on varying load)

In addition, Figure 4.2.2.2.4 depicts that the Burst throughput with respect to load decreases for all the three scenarios viz., Normal scenario, Attack scenario and Solution scenario. It is also evident that the Burst throughput drastically decreases in the Attack scenario when compared to the Normal scenario to a maximum level of 25%.But, when the Cornbach Alpha Reliability Coefficient Based mitigation mechanism is implemented and it increases the Burst throughput by 19%.

4.3 Fake Spectral Attack

In fake spectral attack, the attacker may compromise a core node and then change the wavelength of an incoming burst. Thus, this attack may send the data burst to a compromised channel. It may also change path of all the incoming BCH to a particular malicious channel and hence, there is a chance for stealing the corresponding DB. This particular attack mainly takes place in spectral/wavelength domain as shown in Figure 4.3.

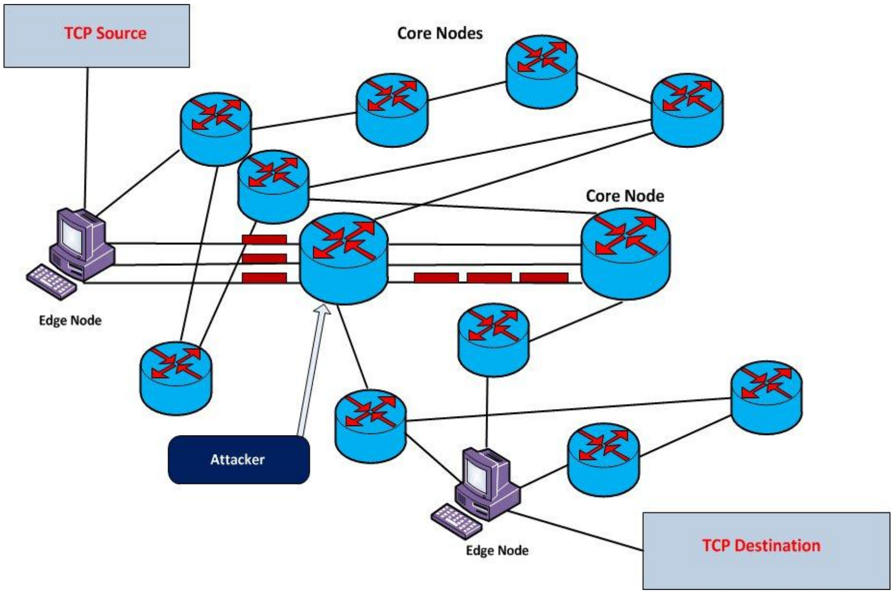


Figure 4.3 Fake Spectral Attack

4.3.1 Attack Detection and Countermeasures

To detect the fake spectral attack, the intermediate core nodes in the Optical Burst Switched network are monitored through the trusted monitor node. When a session is started, every intermediate node sends the network traffic statistics to the monitor.

The monitor node will analyze these values associated with each core node and look into if any of the nodes in the session acts maliciously through variations alpha coefficient approach as depicted in 4.3.1.1. From, the statistics data collected, the node N_i is monitored for k sessions based on the burst packetp_i.Compute the constant mean 'μ 'for all monitored sessions and the observed burst delivery rate$(\mu_p - \mu)$. Further, compute average derivation of node behavior $(\mu_i - \mu)$ and the deviation in the observed and the expected behavior $\left(X_{p_i} - \mu_p - \mu_i + u\right)$.Furthermore, Manipulate Expected mean square (Msp) based on the parameters derived earlier. If the value of 'Msp' is less than the threshold value of fake spectral detection, then the fake spectral attack is detected and mitigated. Finally, the monitor node analyses network traffic loads in all the outgoing channels of all the nodes and if a node drops higher number of bursts then it is tagged as malicious. At this malicious node, the monitor checks the network load. If the difference in network loads between adjacent channels is high then the monitor checks for the wavelength field of each outgoing burst on the present node. If they are found same predominantly, then the fake spectral attack is detected and it is given in algorithm 4.2.

4.3.1.1 Variations Alpha Co-efficient Based Model

In this variations alpha coefficient approach of Fake Spectral Attack mitigation, the trust factor of the monitored nodes is manipulated as follows:

When the node N_i is monitored for k sessions based on the burst packet p_i, then the alpha coefficient can be formulated as,

$$X_{p_i} = \mu + \left(\mu_p - \mu\right) + \left(\mu_i - \mu\right) + \left(X_{p_i} - \mu_p - \mu_i + u\right) \qquad (1)$$

where the first term μ is the constant mean for all monitored sessions, the second term is N_i observed burst delivery rate represented in derivation as$(\mu_p - \mu)$.

The third term is an average derivation of node behavior $(\mu_i - \mu)$ and last term $\left(X_{p_i} - \mu_p - \mu_i + u\right)$ reflects the derivation in observed to the expected behavior. Further, Expected mean square (Msp) is calculated based on the standard deviation $\sigma_{res}{}^2\ and\ \sigma_p{}^2$ as given by equations (2), (3) and (4).

$$E(Msp) = {\sigma_{res}}^2 + n_i {\sigma_p}^2 \qquad (2)$$

where,

$${\sigma_{res}}^2 = \frac{Ms_i - {\sigma_{res}}^2}{n_i} \qquad (3)$$

$${\sigma_p}^2 = \frac{Ms_p - {\sigma_{res}}^2}{n_i} \qquad (4)$$

With, $1 \leq i \leq n$

Hence, if the value of ‘Msp’ is less than the threshold value of fake spectral detection as specified in [96], the fake spectral attack is detected and mitigated. Further, Different burst assembly policies uses burst data packets into optical bursts and to send the bursts into the network. The routing module selects the appropriate output port for each packet and sends each packet to the corresponding burst assembler module. Each burst assembler module assembles bursts consisting of packets which are headed for a specific egress router. In the burst assembler module, there is a separate packet queue for each class of traffic. The scheduler creates a burst based on the burst assembly technique and transmits the burst through the intended output port. At the egress router, a burst disassembly module disassembles the bursts into packets and sends the packets to the upper network layers.

Furthermore, during the implementation and design of the optical burst switching (OBS) network the critical issue occurred at the edge nodes of the OBS network is the burst assembly mechanism. At the edge of an optical burst switching network the burst assembly will group the incoming packets from various sources into the bursts. The arriving packets are forwarded from the switching unit to bursts assembly unit. In one burst assembly unit the packets with same output lines are processed. The burst algorithm is characterized into traffic burstiness and traffic self-similarity. Thus, the end to end performance is done by the burst assembly algorithm (BAA). In burst assembly strategy, the overall delay can be reduced to degrade the packet burstification delay and to enlarge the burst size, meaning that the amount of burst produced will be reduced while increasing the burst size at the core nodes.

Algorithm 4.2 Fake spectral attack detection Algorithm

```
recv (struct *PACKET packet)
{
    A) Determine node Type from packet
     if ((node Type=’intermediate core node’) OR
     (node Type=’egress node’)
{
    B) Extract burst id, S, D, num_of_packets, burst_size from packet.
    C) Create new packet and store the extracted information inside the new packet.
    D) Send the new packet to the trusted node.
} else if(node Type==’trusted_node’)
{
    A) Extract statistics from packet.
    B) Insert the statics into the linked list based on burst id.
    C) Collect statistics report from trusted node
    D) Based on the statistics data collected, the node $N_i$ is monitored for $k$ sessions based on the burst packet$p_i$. And Compute the constant mean ‘$\mu$ ‘for all monitored sessions and the observed burst delivery rate$(\mu_p - \mu)$.
    E) Further, compute average derivation of node behavior $(\mu_i - \mu)$ and the deviation in the observed and the expected behavior $(X_{p_i} - \mu_p - \mu_i + u)$
    F) Furthermore, Manipulate Expected mean square $(Msp)$ based on the parameters derived from step (E) and (F)
    G) If the value of ‘Msp’ is less than the threshold value of fake spectral detection then, the fake spectral attack is detected and mitigated.
    H) Finally, the trusted node monitor verifies a present node is ingress and malicious and check the burst length and other required statistics matches with original source and destination.
    I) If the node’s trust value greater than the threshold, inform others.
}
}
```

4.3.2 Simulation Results and Analysis

The espionage security attacks like fake spectral attack is simulated on a 14 node NSF network configuration with the simulation parameters given in Table 4.2 and with the assumption of a single random compromised node and it is explained in normal scenario, attack scenario and solution scenario.

Table 4.2 Simulation Parameters for fake spectral attack

Number of Electronic Nodes	28
Number of Optical Nodes	14
Arrival Rate	0.01ms
Total Simulation Time	50 ms
Number of TCP/IP Connections	18
Number of OBS Connections	17
Number of Packets	200
Number of Channels	3
Link Speed	1GB

The performance metrics used for studying the variations alpha coefficient approach of Fake Spectral Attack mitigation are burst blocking probability, Average Goodput, Burst loss probability and Burst throughput. The definition for the above mentioned performance metrics are as follows:

Burst blocking probability: It is defined as the ratio of the number of burst data received by the destination during transmission to the number of burst data actually expected to be delivered at the destination.

Average Goodput: It is defined as application level throughput, i.e. the number of useful information bits delivered by the network to a certain destination per unit of time.

Burst loss probability: It is defined as the probability of number of burst data lost during transmission to the number of burst data expected to be delivered at the destination

Burst throughput: It is defined as the gross bit rate that is transferred physically in to the reliable channel between the source and destination.

4.3.2.1 Performance Evaluation - Experiment 1

In experiment 1, the impact of Fake spectral attack is studied with respect to time based on Burst blocking probability, Average Goodput, Burst loss probability and Burst throughput with respect to three scenarios namely Normal scenario, Attack scenario and Solution scenario are illustrated through Figures 4.3.2.1.1, 4.3.2.1.2, 4.3.2.1.3 and 4.3.2.1.4.

The following Figure 4.3.2.1.1 portrays that the burst block probability increases with respect to time for all the three scenarios viz., Normal scenario, Attack scenario and Solution scenario.

It is evident that the burst block probability decreases drastically in the Attack scenario when compared to the Normal scenario to a maximum level of 27%. But, when the variations alpha coefficient approach of Fake Spectral Attack mitigation mechanism is implemented, it decreases the Burst block probability by 26%.

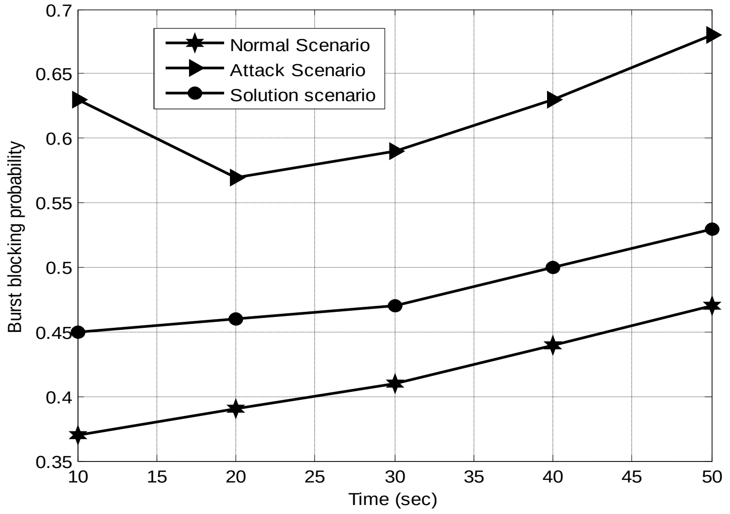


Figure 4.3.2.1.1 Fake Spectral Attack – Burst blocking probability (Based on time)

Further, Figure 4.3.2.1.2 portrays that the Average Goodput delay increases with varying amount of time for all the three scenarios viz., Normal scenario, Attack scenario and Solution scenario. It is also evident that the Average Goodput decreases drastically in the Attack scenario when compared to the Normal scenario to a maximum level of 26%.

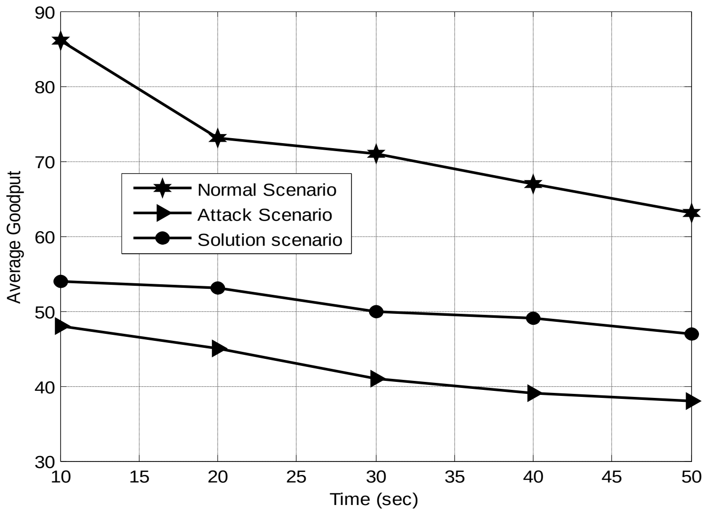


Figure 4.3.2.1.2 Fake Spectral Attack – Average Goodput (Based on time)

But, when the variations alpha coefficient approach of Fake Spectral Attack mitigation mechanism is implemented, it increases Average Goodput by 23% with respect to varying time. Furthermore, Figure 4.3.2.1.3 depicts that the Burst loss probability with respect to time increases for all the three scenarios viz., Normal scenario, Attack scenario and Solution scenario.

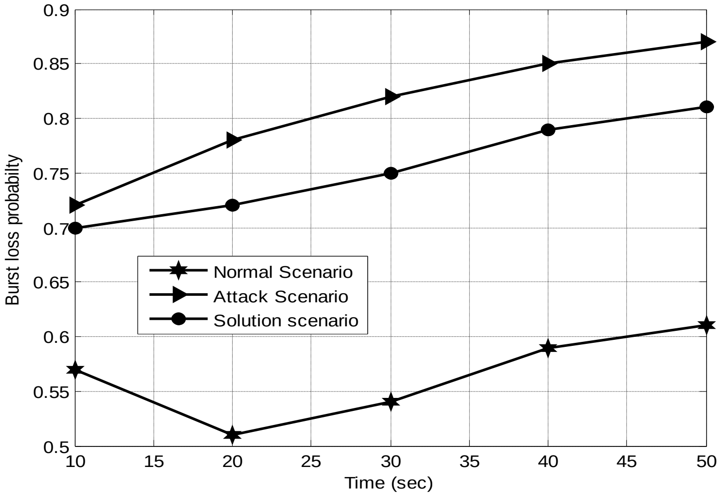


Figure 4.3.2.1.3 Fake Spectral Attack – Burst loss probability (Based on time)

It is also evident that the Burst loss probability drastically increases in the Attack scenario when compared to the Normal scenario to a maximum level of 16%. But, when the variations alpha coefficient approach of Fake Spectral Attack mitigation is implemented, it decreases Burst loss probability by 21% with respect to varying time.

In addition, Figure 4.3.2.1.4 depicts that the Burst throughput with respect to time decreases for all the three scenarios viz., Normal scenario, Attack scenario and Solution scenario. It is also evident that the Burst throughput drastically decreases in the Attack scenario when compared to the Normal scenario to a maximum level of 17%.But, when variations alpha coefficient approach of Fake Spectral Attack mitigation is implemented; it increases the Burst throughput by 19%.

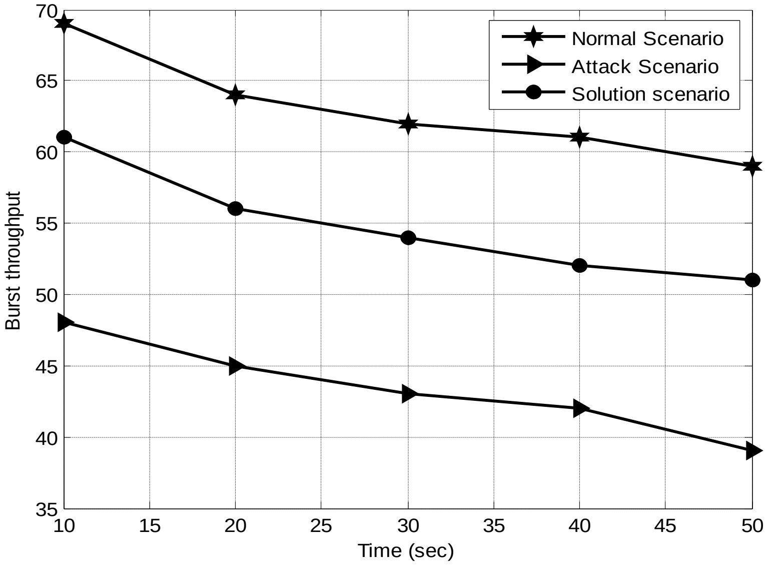


Figure 4.3.2.1.4 Fake Spectral Attack – Burst throughput (Based on time)

4.3.2.2 Performance Evaluation - Experiment 2

In experiment 2, the impact of Fake spectral attack is studied with respect to load based on Burst blocking probability, Average Goodput, Burst loss probability and Burst throughput with respect to three scenarios namely Normal scenario, Attack scenario and Solution scenario are illustrated through Figures 4.3.2.2.1, 4.3.2.2.2, 4.3.2.2.3 and 4.3.2.2.4.

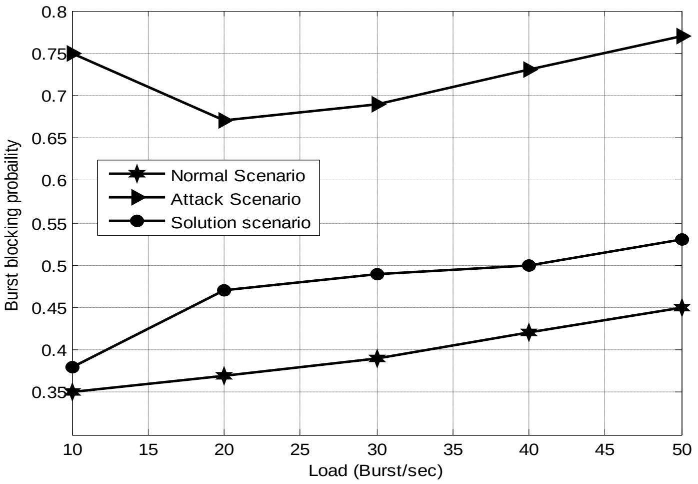


Figure 4.3.2.2.1Fake Spectral Attack – Burst blocking probability (Based on load)

The Figure 4.3.2.2.1 portrays that the burst block probability increases with respect to load for all the three scenarios viz., Normal scenario, Attack scenario and Solution scenario. It is evident that the burst block probability decreases drastically in the Attack scenario when compared to the Normal scenario to a maximum level of 22%. But, when the variations alpha coefficient approach of Fake Spectral Attack mitigation mechanism is implemented, it decreases the Burst block probability by 21%.

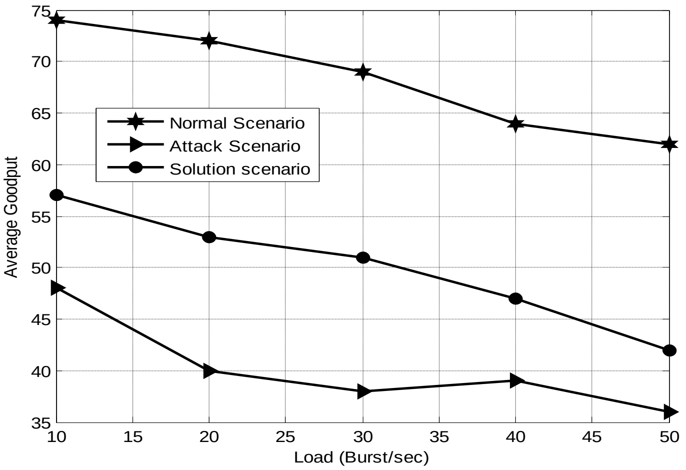


Figure 4.3.2.2.2 Fake Spectral Attack – Average Goodput (Based on load)

In Figure 4.3.2.2.2 portrays that the Average Goodput delay increases with varying amount of load for all the three scenarios viz., Normal scenario, Attack scenario and Solution scenario.

It is also evident that the Average Goodput decreases drastically in the Attack scenario when compared to the Normal scenario to a maximum level of 21%. But, when the variations alpha coefficient approach of Fake Spectral Attack mitigation mechanism is implemented, it increases Average Goodput by 18% with respect to varying load. Furthermore, Figure 4.3.2.2.3 depicts that the Burst loss probability with respect to load increases for all the three scenarios viz., Normal scenario, Attack scenario and Solution scenario.

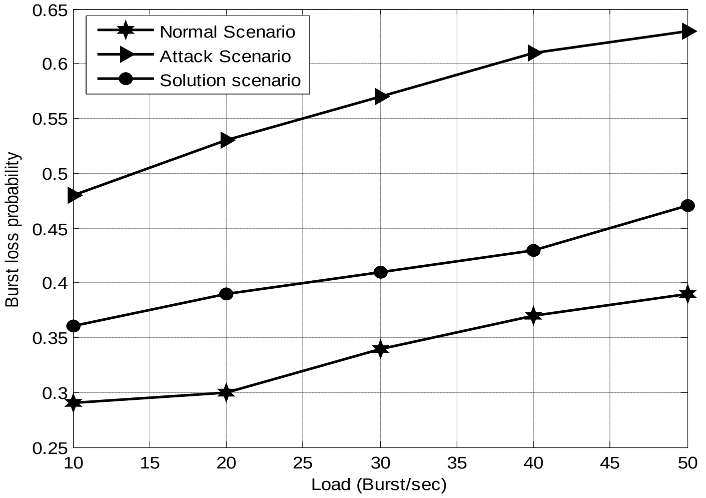


Figure 4.3.2.2.3 Fake Spectral Attack – Burst loss probability (Based on load)

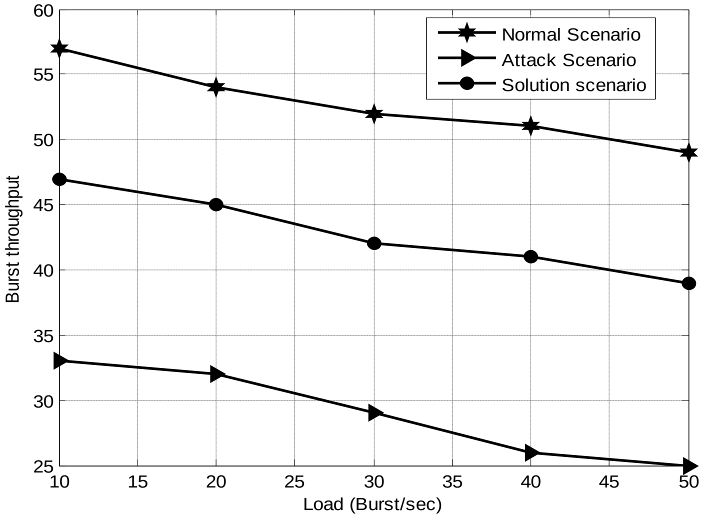


Figure 4.3.2.2.4 Fake Spectral Attack – Burst throughput (Based on load)

It is also evident that the Burst loss probability drastically increases in the Attack scenario when compared to the Normal scenario to a maximum level of 13%. But, when the variations alpha coefficient approach of Fake Spectral Attack mitigation is implemented, it decreases Burst loss probability by 16% with respect to varying load.

In addition, Figure 4.3.2.2.4 depicts that the Burst throughput with respect to load decreases for all the three scenarios viz., Normal scenario, Attack scenario and Solution scenario. It is also evident that the Burst throughput drastically decreases in the Attack scenario when compared to the Normal scenario to a maximum level of 13%.But, when variations alpha coefficient approach of Fake Spectral Attack mitigation is implemented; it increases the Burst throughput by 15%.

4.4 Summary

In this Chapter, two espionage attacks namely burst hijacking attack and the fake spectral attack is thoroughly analyzed and the possible countermeasures with the mathematical model are also discussed. The effects of these attacks on data traffic routing are shown using simulation graphs. The Inference elucidated from above discussed two espionage attacks is too dangerous.

CHAPTER 5

DENIAL OF SERVICE ATTACKS FOR OBS NETWORKS

5.1 Introduction

Denial of service attack in the Optical network communication is critical since exploitation of vulnerability in the switched architecture may not utilize massive bandwidth. Hence, the optical Burst Switched architecture needs to be viable and must ensure integrity of communication. Specifically, Denial of service attack further drastically disrupts the service offered by optical switched networks in terms of availability and reliability. In this chapter, two potential denials of service attacks viz.., Burst flooding attack and Timeout attack are introduced and the formulated mathematical model for mitigation are also demonstrated. Further, the lethality of the attacks is shown through simulation experiments and studied under various burst load and time.

5.2 Burst Flooding Attack

In the Burst flooding attack, any optical node when compromised by the intruders may generate multiple BCH as shown in Figure 5.2 and this may lead to situation called denial of service in which deadlock may occur in the allocation of all the network resources. In other words, the compromised intermediate node creates multiple copies of same BCH and forwards it to the successor node and induces flooding to each and every intermediate node with the duplicate copies of the original BCH. Further, the intermediate node tries to do reservation for these bogus control headers. Hence, the overflow of buffers may take place at the intermediate core node and if the wavelength conversion is implemented then this bogus control header reserves different wavelength for its corresponding data burst. Finally, the uncompromised node cannot reserve the resources after receiving a valid BCH.

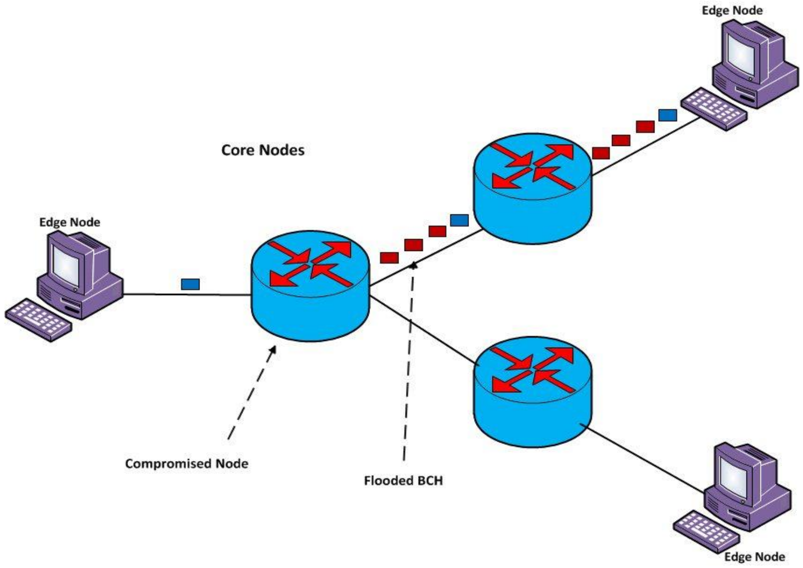


Figure 5.2 Burst Flooding Attack

5.2.1 Attack Detection and Countermeasures

In the burst flooding attack, the intermediate core nodes in the Optical Burst Switched network are monitored through a trusted monitor node using the Kappa Coefficient based mitigation model as depicted in 5.2.1.1. When a session is started, every intermediate node sends the network traffic statistics to the monitor. From the statistical data, the trusted monitor estimates $and\ TUP_t$ and TUP_r are categorized as the number of burst packets forwarded and burst packets received by the core node as monitored from the trusted node. Similarly, the trusted node calculates the trustworthiness of the core node (T_o) in terms of incoming bursts as observed by the trust monitor using equation (2).

Further, the monitor manipulates the expected trustworthiness $'T_c'$ of the monitored core node by multiplying the managerial total (m_1) corresponding to the sum of TUP_t and TUP_r for all the sessions. Finally, the Kappa coefficient $'KC_{(i)}$' for estimating the node's trustworthiness is calculated using equation(4) .If the value lies between -1 to 0, then the node is compromised. Then, call the core node maliciousness mitigation procedure. Else, perform normal routing activity as presented by algorithm 5.1.

Furthermore, the monitor node analyzes these values associated with each core node and looks into if any of the nodes in the session acts maliciously. This approach is based on trusted node concept in network security for IP networks. In addition, the monitor node analyses network traffic verify a new connection is established in virtual source node and burst_id and other required statistics matches with source and destination. Finally, if the node's trust value reaches below threshold, inform other nodes, then the ingress node is affected by Burst flooding attack.

5.2.1.1 Kappa Coefficient Based Model

Kappa Coefficient based mitigation model for detecting Burst Flooding Attack is based upon a coefficient called Kappa Coefficient (KC), which helps in identifying the trust level of the nodes participating in the multicast environment. The trusted node monitors and stores the related information about the nodes communicating in the optical burst environment. The monitoring process is carried out by the trusted node.

If TUP_t and TUP_rbe the number of burst packets forwarded and burst packets received by the core node as monitored from the trusted node. Further, if TDP_t and TDP_r be the packets forwarded and receive by the trusted node from the core node respectively. Then, the trustworthiness of the group leader (T_p) as observed is given by (1)

$$T_p = \frac{TUP_t + TDP_r}{n} \qquad (1)$$

where 'n' is number of observations monitored for detecting Burst Flooding Attack. Similarly, trustworthiness can also be calculated through (2)

$$T_o = \frac{TDP_t + TUP_r}{n} \qquad (2)$$

Then, the expected trustworthiness of the monitored core node is calculated by multiplying the managerial total (m_1) corresponding to the sum of TUP_t and TUP_r for all four scenarios, i.e., $(m_1, m_2, f_1 \, and \, f_2)$, then expected trustworthiness is given by equation (3) as,

$$T_c = \frac{\frac{m_1 f_1}{n} + \frac{m_2 f_2}{n}}{n} \qquad (3)$$

where $f_1 and\ f_2$ corresponds to third and fourth managerial total corresponding to TUP_r with TDP_r andTDP_t withTDP_r.

Therefore, the Kappa coefficient to estimate the node’s trustworthiness is given by equation (4),

$$KC_{(i)} = \frac{T_o - T_c}{1 - T_c} \qquad (4)$$

The possible values of $KC_{(i)}$ is form -1 to 1. The value between -1 to 0 indicates that the core node is compromised. Hence the core node maliciousness has to be mitigated from the network or else normal routing is induced.

Algorithm 5.1 Burst flooding attack detection Algorithm

```
recv (struct *PACKET packet)
{
Determine node Type from packet
        if ((node Type=’intermediate core node’) OR (node Type=’egress node’)
{
        A) Extract burst id, source, destination, num_of_packets, burst_size from
           packet.
        B) Create new packet and store the extracted information inside the new
           packet.
        C) Send the new packet to the trusted node.
}
else if( node Type==’trusted_node’)
{
        A) Extract statistics from packet.
        B) Insert the statics into the linked list based on burst id.
        C) Collect statistics report from trusted node
        D) From the statistical data, $TUP_t$ and $TUP_r$ are categorized as the number of burst packets forwarded and burst packets received by the core node as monitored from the trusted node.
        E) The trusted node calculates the trustworthiness of the core node $(T_p)$ in terms of incoming bursts as observed by the trust monitor using equation (1)
        F) Similarly, the trusted node calculates the trustworthiness of the core node $(T_o)$ in terms of incoming bursts as observed by the trust monitor using equation (2)
        G) Further, the monitor manipulates the expected trustworthiness $'T_c'$ of the monitored core node by multiplying the managerial total $(m_1)$ corresponding to the sum of $TUP_t$ and $TUP_r$ for all the sessions.
        H) Finally, the Kappa coefficient $'KC_{(i)}$' for estimating the node's trustworthiness is calculated using equation(4)
        I) If the value lies between -1 to 0, then the node is compromised.
        J) Call the core node maliciousness mitigation procedure
        K) Else, perform normal routing activity.
        L) In Burst Flooding Attack, verify the statistics based in burst_id, burst size, number of packets inside the burst and reduce the node's trust value if the statistics mismatches.
        M) If the node's trust value reaches below threshold, inform other nodes.
}
}
```

5.2.2 Simulation Results and Analysis

The denial of service security attacks like burst flooding attack is simulated on a 14 node NSF network configuration with the simulation parameters given in Table 5.1 and with the assumption of a single random compromised node and it is explained in normal scenario, attack scenario and solution scenario.

Table 5.1 Simulation Parameters for burst flooding attack

Number of Electronic Nodes	28
Number of Optical Nodes	14
Arrival Rate	0.01ms
Total Simulation Time	50 ms
Number of TCP/IP Connections	18
Number of OBS Connections	17
Number of Packets	200
Number of Channels	3
Link Speed	1 GB

The performance metrics used for studying the Kappa Coefficient based mitigation model for detecting Burst Flooding Attack are burst blocking probability, Average Goodput, Burst loss probability and Burst throughput. The definition for the above mentioned performance metrics are as follows:

Burst blocking probability: It is defined as the ratio of the number of burst data received by the destination during transmission to the number of burst data actually expected to be delivered at the destination.

Average Goodput: It is defined as application level throughput, i.e. the number of useful information bits delivered by the network to a certain destination per unit of time.

Burst loss probability: It is defined as the probability of number of burst data lost during transmission to the number of burst data expected to be delivered at the destination

Burst throughput: It is defined as the gross bit rate that is transferred physically in to the reliable channel between the source and destination.

5.2.2.1 Performance Evaluation - Experiment 1

In experiment 1, the impact of Burst Flooding Attack is studied based on burst blocking probability and Burst Throughput by varying the load under different burst blocking probability of 0.3, 0.4 and 0.5 respectively with respect to three scenarios namely Normal scenario, Attack scenario and Solution scenario.

The following Figure 5.2.2.1.1 illustrates that the proposed solution yields the maximum value of burst throughput as 75.56% which is 26.72% higher than that of the attack scenario.

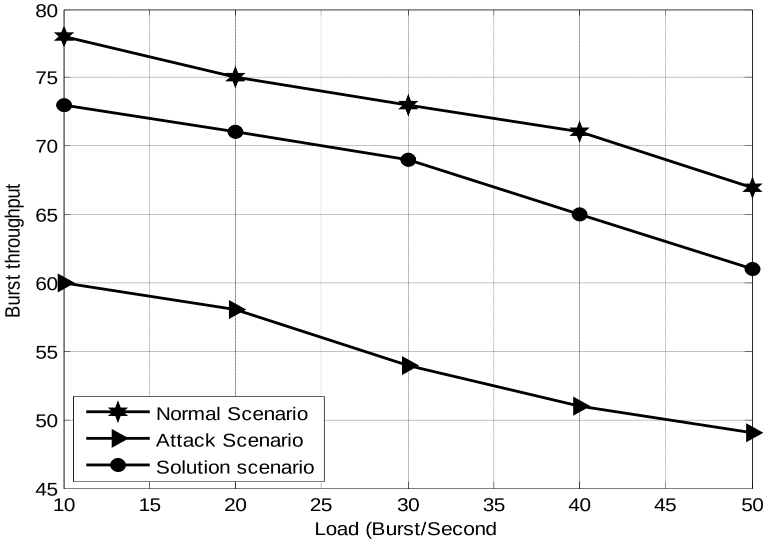


Figure 5.2.2.1.1 Burst Flooding Attack – Burst Throughput (Burst blocking probability of 0.3)

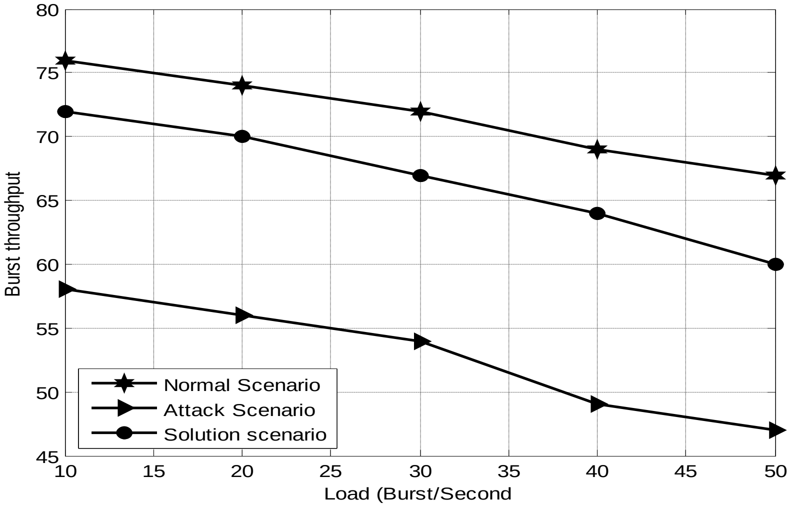


Figure 5.2.2.1.2 Burst Flooding Attack – Burst Throughput (Burst blocking probability of 0.4)

The obtained burst probability of the solution scenario is only 12.37% lesser than that of the ideal case which shows the propose method efficiently mitigates burst flooding attack present in the NSF network.

The results also infer that the solution mechanism known as Kappa Coefficient Based Mitigation Model (KCBM) improves the network performance by increasing the burst probability to an average value of 69.75% with the burst probability value as 0.3. Further, the Figure 5.2.2.1.2 shows the plot for burst probability obtained by varying the load values corresponding to the fixed value of burst blocking probability as 0.4. Since the burst blocking probability value is increased, the performance of the overall network is decreased to some extent when compared with previous scenario (burst blocking probability - 0.3). The obtained results in this scenario indicate that the solution mechanism yields maximum burst throughput rate of 72.67%, which is 23.45% higher than that of the attack scenario.

Furthermore, the Figure 5.2.2.1.3 shows the plot of burst throughput obtained by varying the load values with the corresponding burst block probability rate is 0.5. This increase in burst block probability rate also increases the influence of the attack in the network which in turn decreases the performance of the network in terms of burst throughput. But, the presence of KCBM model in the NSF network increases the burst throughput to the maximum value of 67.56% which is 21.34% higher than that of the attack scenario.

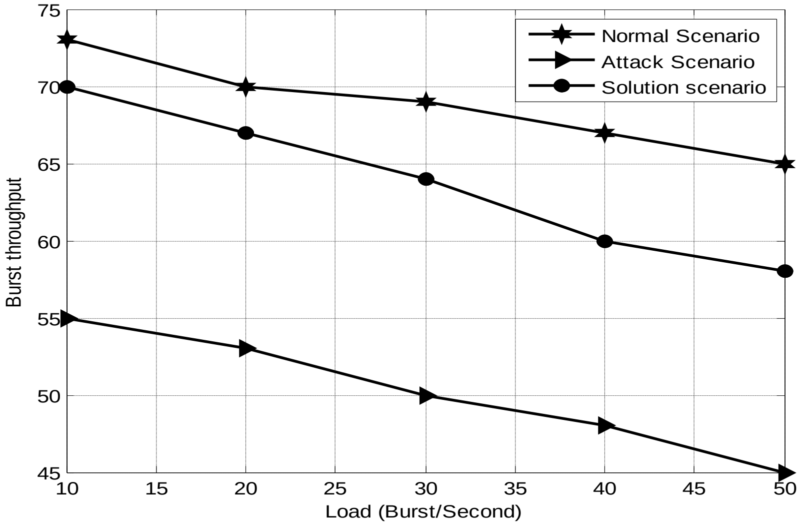


Figure 5.2.2.1.3 Burst Flooding Attack – Burst Throughput (Burst blocking probability of 0.5)

In addition to this, the Figure 5.2.2.1.4 portrays the plots of burst block probabilities derived in the various scenarios by varying the load of the network. The values of the burst block probability indicates that the impact of burst flooding attack in this network environment. The increase in the burst block probability rate degrades the performance of the network in terms of burst throughput.

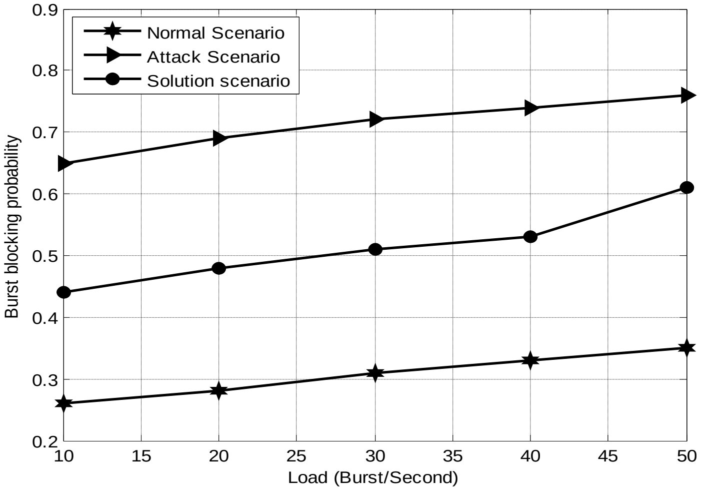


Figure 5.2.2.1.4 Burst Flooding Attack – Burst blocking probability

The graphical figure illustrates that, the presence of KCBM in the network environment mitigates the burst flooding attack and decreases the burst block probability to the minimum extent of 0.367. Whereas, the average burst block probability value obtained in attack scenario is 0.698 which is 53% more than that of the solution scenario value. The obtained results conclude that, the deployment of KCBM in the network scenario mitigates the burst flooding attack in a robust manner and hence increases the performance of the network by increasing the burst through rate.

5.2.2.2 Performance Evaluation - Experiment 2

Yet, the performance of the network is evaluated by analyzing the burst throughput values obtained by the time required for propagation. The following figure 5.2.2.2.1 shows the burst throughput plots obtained by varying the time of propagation in normal, attack and solution scenarios. The obtained results indicate that, the solution scenario yields the maximum burst probability rate of 74% which is 19% greater than that of the attack scenario.

This increase in burst throughput is due to the deployment of KCBM in the network environment which mitigates the burst flooding attack in a rapid and precise manner. Further, The Figure 5.2.2.2.2 illustrates the plots of burst lost probability obtained in various scenarios by varying the time of propagation. The increase in burst lost probability shows degradation in network performance in terms of burst throughput.

But, the implementation of KCPM in the network environment mitigates the burst flooding attack at the rate of 38%, which in turn decreases the burst lost probability in an average value of 0.345. The obtained burst lost probability rate in the solution scenario is 59% lesser than that of the attack scenario.

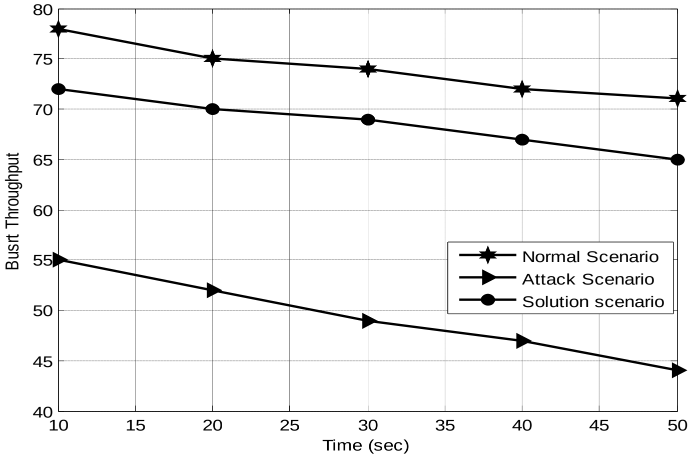


Figure 5.2.2.2.1 Burst Flooding Attack – Burst Throughput

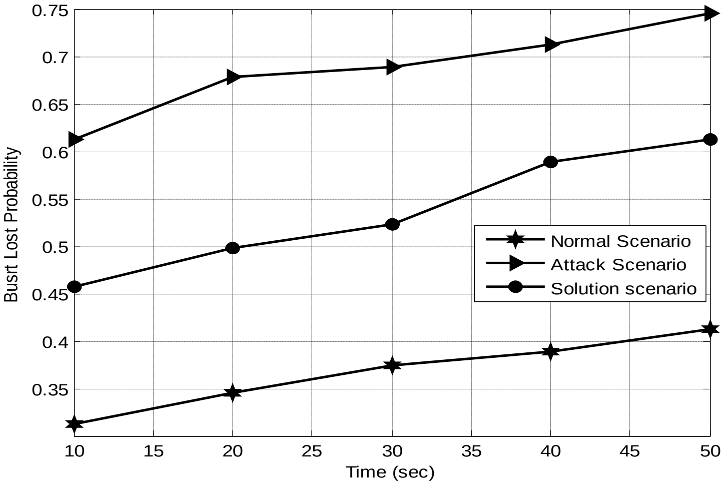


Figure 5.2.2.2.2 Burst Flooding Attack – Burst Lost Probability

Yet another method of evaluating the performance of the network is through the analysis of average good put. The following figure 5.2.2.2.3 and 5.2.2.2.4 show the plots for average goodput obtained by varying time and propagation delay respectively. The results are derived in terms three scenarios viz., normal, attack and solution scenarios.

In Figure 5.2.2.2.3 shows the maximum average goodput value yielded in the solution scenario as 75.12% which is 34% greater than that of the attack scenario. This considerable variation in the performance is due to the deployment of KCBM in the network environment which mitigates burst flooding attack at the rate of 38%. Further figure 5.4.1.2.8 show the maximum average goodput value as 73.89% for solution scenario which is only 6.74% lesser than that of the normal scenario. The implementation of KCBM in the network environment increases the average goodput value by 24% when compared to the attack scenario.

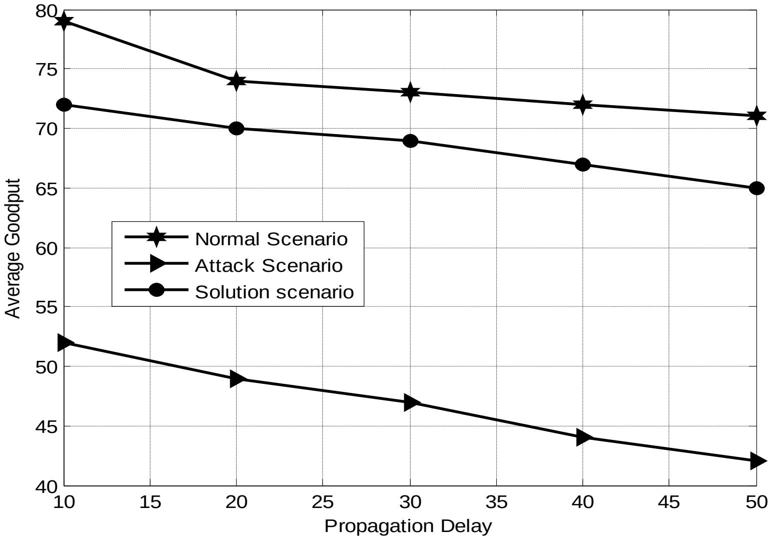


Figure 5.2.2.2.3 Burst Flooding Attack – Average Goodput

On the whole, the implementation of KCBM in the network scenario efficiently mitigates the burst flooding attack which in turn increases Burst throughput, Average goodput and at the same time decreases burst block probability and burst lost probability.

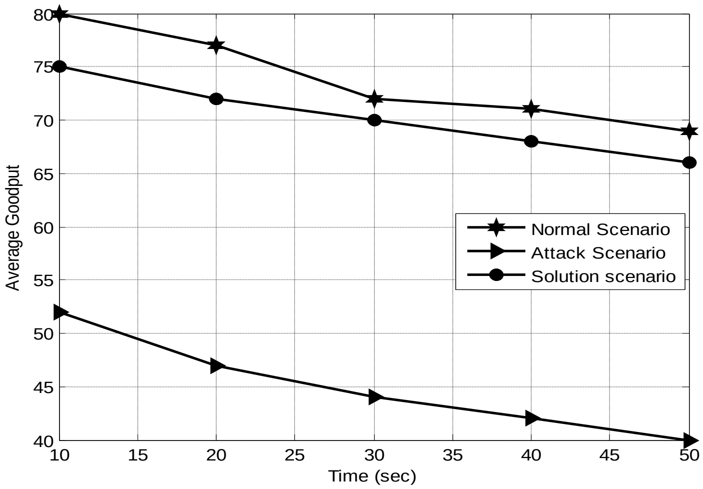


Figure 5.2.2.2.4 Burst Flooding Attack – Burst Throughput

5.3 Timeout attack

In OBS networks, the ingress router the packets are assembled to form a burst. There are mainly two assembling schemes. First is based on the threshold and the second is based on the time. In the timer based scheme, a timer is initialized during burst assembly based on maximum number of packet. A data burst is generated when the timer exceeds the burst assembly period or when the maximum number of packet is reached. In case of Timeout attack, the timeout value of the optical node plays a critical role when it goes below a threshold. As the result, ingress node starts to produce many small bursts and send to the destination causing unwanted traffic as shown in Figure 5.3.

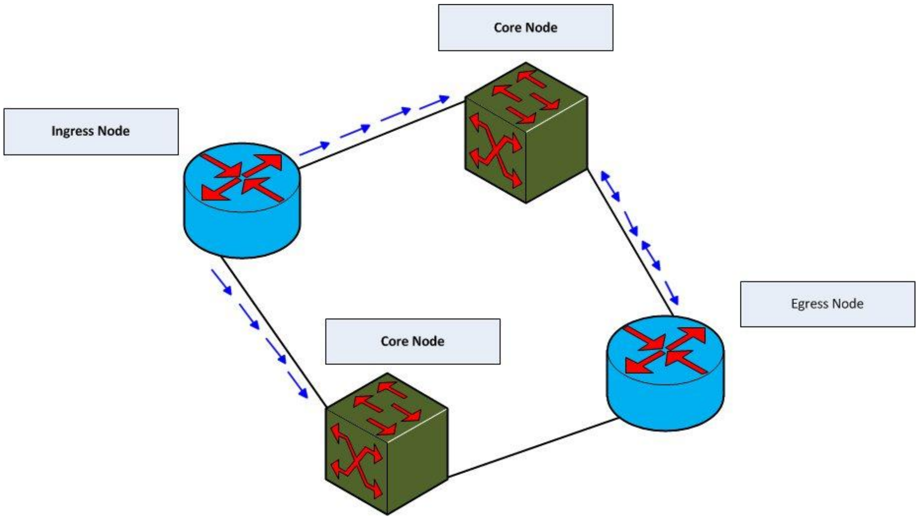


Figure 5.3 Timeout Attack

5.3.1 Attack Detection and Countermeasures

For detecting Timeout attack, the value of the TIMEOUT and MAX_PACKET_NUM is predefined. If an attacker compromises the ingress edge router, the attacker modifies the value of the TIMEOUT, resulting in the formation of the burst which are smaller in size. Since the bursts are smaller in size number of burst generation is high and it is leading to disrupting traffic. The attack may just convert the standard value of the burst formation. The solution for this attack is based on the Optimal Time Threshold based mitigation approach in which the burst formation utilizes a standard value for burst assembly that is stored in the ingress node. The node creates the burst as per the value stored in it as depicted in algorithm 5.2. The trust monitor, estimates'N_i 'that represents the number of burst packets in any buffer i. Then, manipulates the cumulative sum of bursts packets transmitted by the node as $N_1 + N_2 + \cdots + N_n$. Further, it calculates time bound through various burst assembling time and computes the optimal average time required for burst assembly$\left(t_f(i)\right)$.when the average time required $\left(n_f(i)\right)$ for burst assembly is less than the threshold time required for burst assembly$\left(n_r(i)\right)$, then, the time attack is detected and the mitigation mechanism comes into act.

5.3.1.1 Optimal Time Threshold Based

In this Optimal Time Threshold based mitigation approach for Timeout Attack, the trust factor of the monitored nodes is manipulated as follows:

In OBS networks, the ingress router the packets are assembled to form a burst. There are mainly two assembling schemes. First is based on the threshold based and the second is based on the time based. In timer based scheme, a timer is started at the initialization of burst assembly. The latter is based on maximum number of packet. A data burst is generated when the timer exceeds the burst assembly period or when the maximum number of packet is reached. Then the detection model is enumerated as follows:

Let N_i represents the number of burst packets in the buffer i. Then the state $s(t)$ is defined by equation (1) as,

$$S(t) = (N_1, N_2, \ldots, N_n) \qquad (1)$$

where, $1 \le i \le n$,

Then, the cumulative sum of bursts packets in the system is$N_1 + N_2 + \cdots + N_n$. Since the buffer capability of each buffer is N, the number of states is $(N + 1)^N$. In this model, the wavelengths are not considered, since the burst packets being relayed are independent those of the transmission time. Thus, the state space of the model is given equation (2) as follows,

$$E = [1, 1, 2, \dots, i, \dots L_N] \qquad (2)$$

Further, based on the statistical data obtained the number of ways of classifying burst assembling time [97] is given by

$$P(n, m, k) = \sum_{j=0}^{m} (-1)^j \binom{m}{j} \binom{n+m-j(k+1)-1}{m-1} \qquad (3)$$

where, n – number of burst packets
m – Number of burst packets assembled
k – Time expected for burst assembly

Then, the optimal average time required for burst assembly$(t_f(i))$ is derived from (4) as

$$t_f(i) = \sum_{j=1}^{N} \frac{P(n, m, k/b_L = j)}{P(n, m, k)} \qquad (4)$$

where, b_L - buffer length and $j \leq k$.

When the average time required $\left(n_f(i)\right)$ for burst assembly is less than the threshold time required for burst assembly$\left(n_r(i)\right)$, the timeout attack is detected and the mitigation mechanism comes into play.

Algorithm 5.2 Timeout attack detection Algorithm

```
recv (struct *PACKET packet) {
Determine node Type from packet
if ((node Type=’intermediate core node’) OR (node Type=’egress node’)
{
    A) Extract burst id, source, destination, num_of_packets, burst_size,
       TIME_OUT VALUE from packet.
    B) Create new packet and store the extracted information inside the new
       packet.
    C) Send the new packet to the trusted node.
} else if(node Type==ingress node’)
{
    A) Extract statistics from packet and invoke the proc.
    B) Insert the TIMEOUT VALUE into the linked list based on burst id.
    C) Collect statistics report from trusted node
    D) Estimate’$N_i$ ‘that represents the number of burst packets in any buffer$i$.
    E) Manipulate the cumulative sum of bursts packets transmitted by the node
       as $N_1 + N_2 + \cdots + N_n$
    F) From the statistical data obtained, calculate time bound through various
       burst assembling time given by
           $P(n,m,k) = \sum_{j=0}^{m} (-1)^j \binom{m}{j} \binom{n+m-j(k+1)-1}{m-1}$
    G) Compute the optimal average time required for burst assembly$(t_f(i))$ as
    $t_f(i) = \sum_{j=1}^{N} \frac{P(n,m,k/b_L = j)}{P(n,m,k)}$
    H) When the average time required $\left(n_f(i)\right)$ for burst assembly is less than
       the threshold time required for burst assembly$\left(n_r(i)\right)$,
    I) Then, the time attack is detected and the mitigation mechanism comes
       into play.
    J)  Further, extract the STANDARD_TIME_OUT VALUE and
        MAX_NUM_PACKETS from the linked list head.
    K)  In Timeout Attack, verify the statistics based in burst id, if
        STANDARD_TIME_OUT VALUE, MAX_NUM_ PACKETS values are
        mismatches the attack is identified and invoke procedure.
    L)  If standard time value is assign to the current timeout value and do the same.
}
}
```

5.3.2 Simulation Results and Analysis

The denial of service security attack like Timeout attack is simulated on a 14 node NSF network configuration with the simulation parameters given in Table 5.2 and with the assumption of a single random compromised node and it is explained in normal scenario, attack scenario and solution scenario.

Table 5.2 Simulation Parameters for Timeout attack

Number of Electronic Nodes	28
Number of Optical Nodes	14
Arrival Rate	0.01ms
Total Simulation Time	50 ms
Number of TCP/IP Connections	18
Number of OBS Connections	17
Number of Packets	200
Number of Channels	3
Link Speed	1GB

The performance metrics used for studying the Optimal Time Threshold based mitigation approach for Timeout Attack are burst blocking probability, Average Goodput, Burst loss probability and Burst throughput. The definition for the above mentioned performance metrics are as follows:

Burst blocking probability: It is defined as the ratio of the number of burst data received by the destination during transmission to the number of burst data actually expected to be delivered at the destination.

Average Goodput: It is defined as application level throughput, i.e. the number of useful information bits delivered by the network to a certain destination per unit of time.

Burst loss probability: It is defined as the probability of number of burst data lost during transmission to the number of burst data expected to be delivered at the destination

Burst throughput: It is defined as the gross bit rate that is transferred physically in to the reliable channel between the source and destination.

5.3.2.1 Performance Evaluation - Experiment 1

The performance evaluation of the proposed Optimal Time Threshold Based Mitigation (OTTBM) is carried out network environment with above stated characteristics. The evaluation is done by means of estimating parameters like burst through, average goodput burst block probability and burst lost probability by considering three scenarios viz, normal, attack and solution scenario. The normal scenario illustrates the performance of the network that is exhibited in the condition. Whereas, attack scenario describes the performance exhibited during the presence of timeout attack and the solution scenario describes the performance of the network when OTTBM model is deployed in the transmission protocol.

The following Figure 5.3.2.1.1 illustrates plots for burst through obtained by varying the load in various scenarios. The results indicate that the proposed solution yields the maximum value of burst throughput as 75.96% which is 22.82% higher than that of the attack scenario. The obtained burst probability of the solution scenario is only 7.37% lesser than that of the ideal case which shows the propose method efficiently mitigates time out attack present in the NSF network. The results also infer that the solution mechanism OTTBM improves the network performance by increasing the burst probability to an average value of 67.75% with the burst probability value as 0.3.

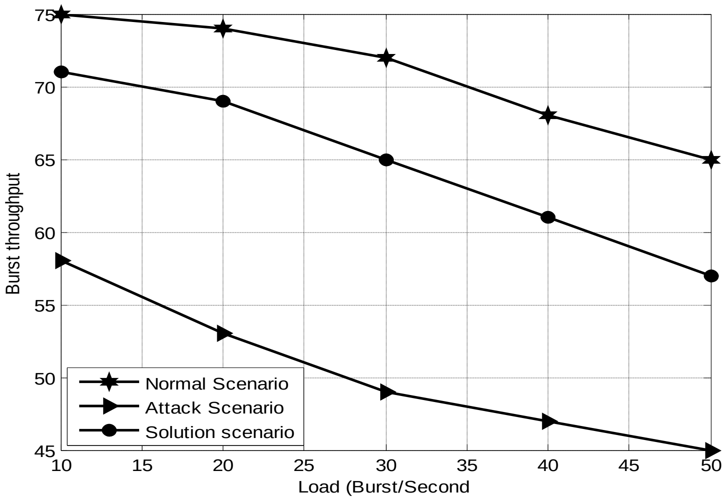


Figure 5.3.2.1.1 Timeout Attack – Burst Throughput (Burst blocking probability of 0.3)

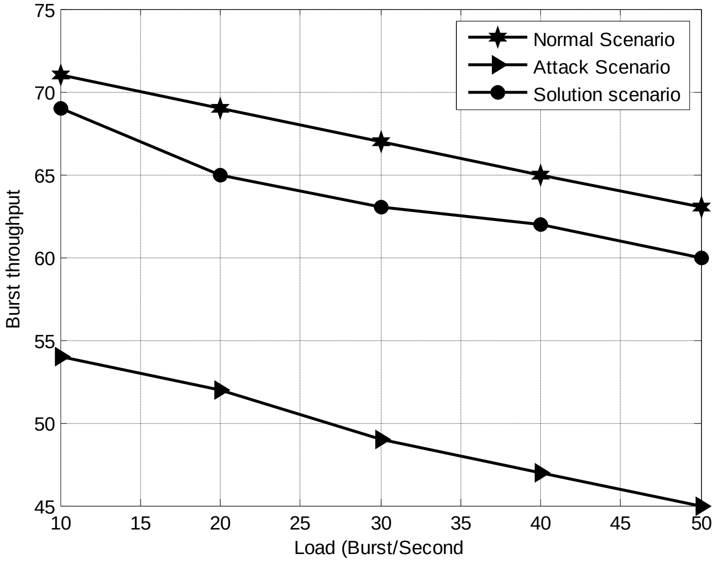


Figure 5.3.2.1.2 Timeout Attack – Burst Throughput (Burst blocking probability of 0.4)

Further, the Figure 5.3.2.1.2 shows the plot for burst probability obtained by varying the load values corresponding to the fixed value of burst blocking probability as 0.4. Since the burst blocking probability value is increased, the performance of the overall network is decreased to some extent when compared with previous scenario (burst blocking probability - 0.3).

The obtained results in this scenario indicate that the solution mechanism OTTBM yields maximum burst throughput rate of 69.67%, which is 15.45% higher than that of the attack scenario

Furthermore, the Figure 5.2.2.1.3 shows the plot of burst throughput obtained by varying the load values with the corresponding burst block probability rate is 0.5. This increase in burst block probability rate also increases the influence of the attack in the network which in turn decreases the performance of the network in terms of burst throughput. But, the presence of OTTBM model in the NSF network increases the burst throughput to the maximum value of 66.46% which is 12.34% higher than that of the attack scenario.

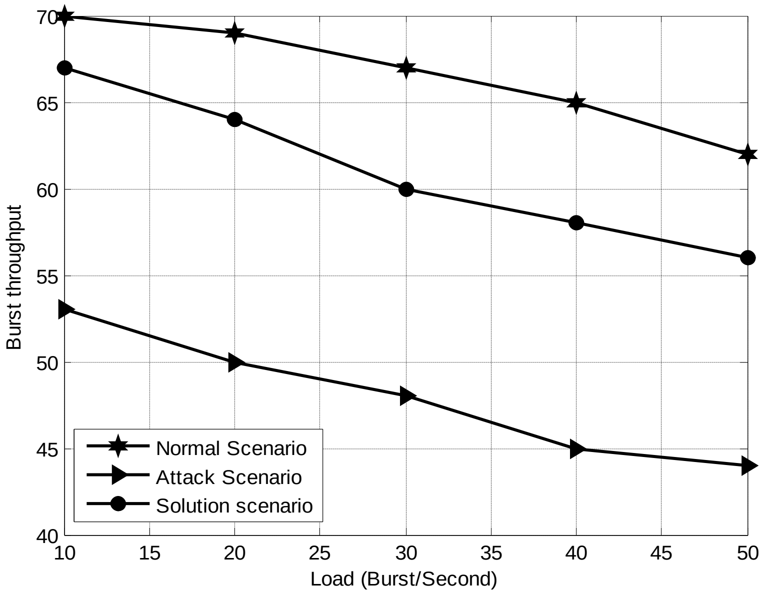


Figure 5.3.2.1.3 Timeout Attack – Burst Throughput (Burst blocking probability of 0.5)

In addition to this, the Figure 5.3.2.1.4 portrays the plots of burst block probabilities derived in the various scenarios by varying the load of the network. The values of the burst block probability indicates that the impact of time out attack in this network environment. The increase in the burst block probability rate degrades the performance of the network in terms of burst throughput.

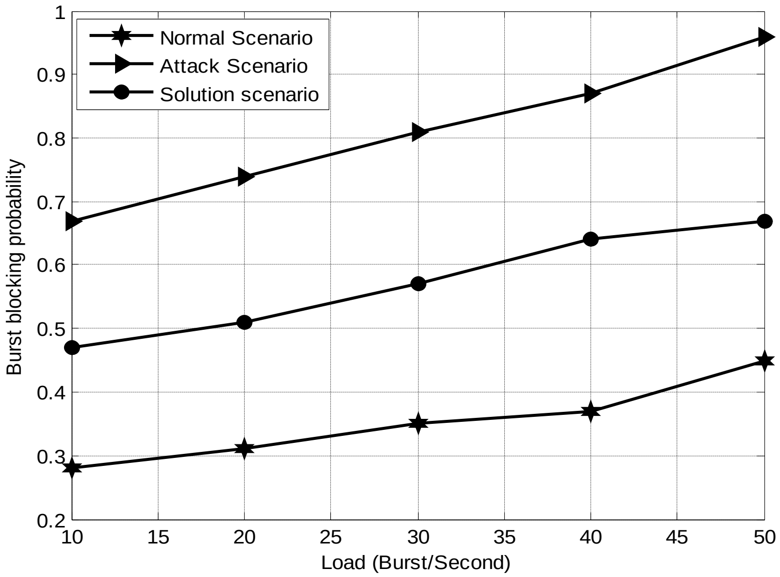


Figure 5.3.2.1.4 Timeout Attack – Burst blocking probability

The graphical figure illustrates that, the presence of KCBM in the network environment mitigates the time out attack and decreases the burst block probability the average value of 0.567. Whereas, the average burst block probability value obtained in attack scenario is 0.998 which is 43% more than that of the solution scenario value. The obtained results conclude that, the deployment of KCBM in the network scenario mitigates the time out attack in a robust manner and hence increases the performance of the network by increasing the burst through rate.

5.3.2.2 Performance Evaluation - Experiment 2

Yet, the performance of the network is evaluated by analyzing the burst throughput values obtained by the time required for propagation. The following Figure 5.3.2.2.1 shows the burst throughput plots obtained by varying the time of propagation in normal, attack and solution scenarios. The obtained results indicate that, the solution scenario yields the maximum burst probability rate of 70% which is 17% greater than that of the attack scenario. This increase in burst throughput is due to the deployment of OTTBM in the network environment which mitigates the time out attack in a rapid and precise manner.

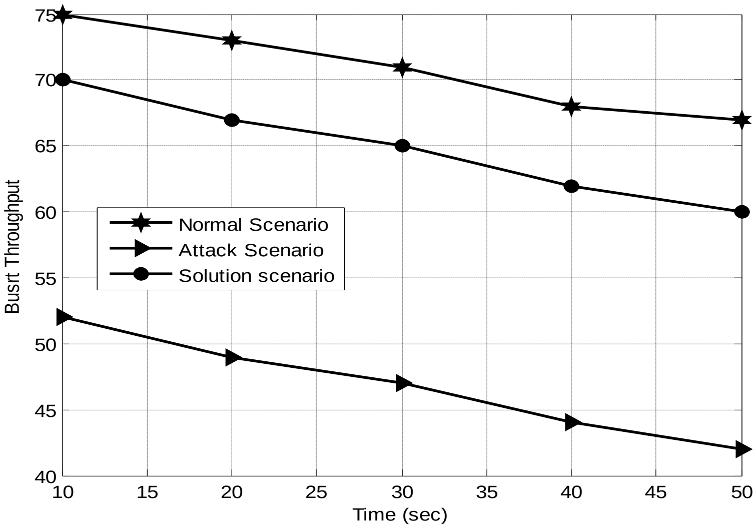


Figure 5.3.2.2.1 Timeout Attack – Burst Throughput

Further, Figure 5.3.2.2.2, illustrates the plots of burst lost probability obtained in various scenarios by varying the time of propagation. The increase in burst lost probability shows degradation in network performance in terms of burst throughput.

But, the implementation of OTTBM in the network environment mitigates the time out attack at the rate of 29%, which in turn decreases the burst lost probability in an average value of 0.3545. The obtained burst lost probability rate in the solution scenario is 24.4% lesser than that of the attack scenario.

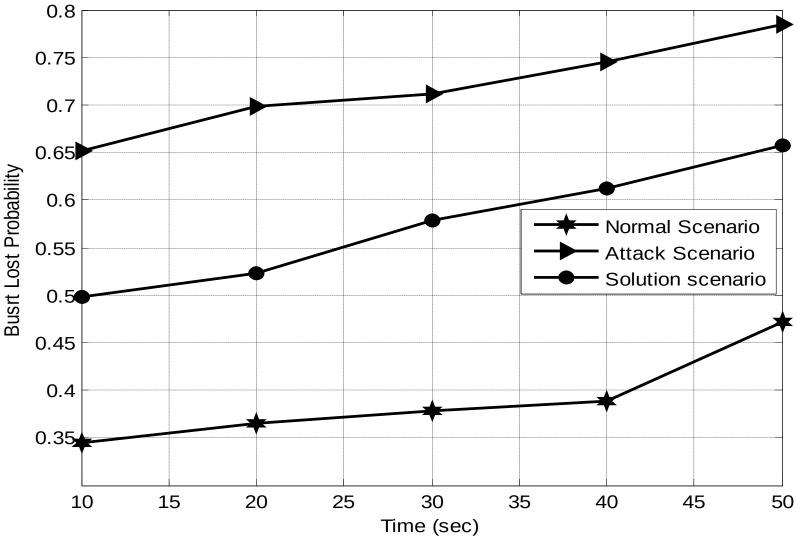


Figure 5.3.2.2.2 Timeout Attack – Burst Lost Probability

Yet another method of evaluating the performance of the network is through the analysis of average good put. The following Figures 5.3.2.2.3 and 5.3.2.2.4 show the plots for average goodput obtained by varying time and propagation delay respectively. The results are derived in terms three scenarios viz., normal, attack and solution scenarios.

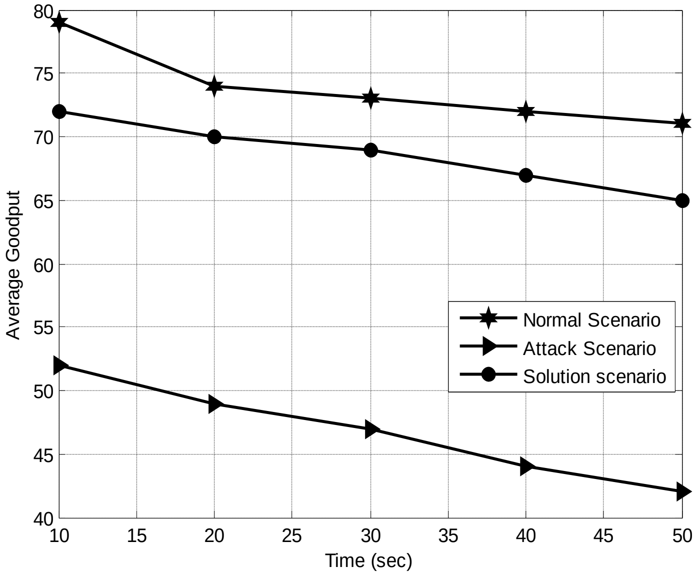


Figure 5.3.2.2.3 Timeout Attack – Average Goodput (Varying Time)

Figure 5.3.2.2.3 shows the maximum average goodput value yielded in the solution scenario as 72.12% which is 20% greater than that of the attack scenario. This considerable variation in the performance is due to the deployment of OTTBM in the network environment which mitigates time out attack at the rate of 29%.

Further Figure 5.3.2.2.4 show the maximum average goodput value as 73.89% for solution scenario which is only 6.74% lesser than that of the normal scenario. The implementation of OTTBM in the network environment increases the average goodput value by 14% when compared to the attack scenario.

On the whole, the implementation of OTTBM in the network scenario efficiently mitigates the burst flooding attack which in turn increases Burst throughput, Average goodput and at the same time decreases burst block probability and burst lost probability.

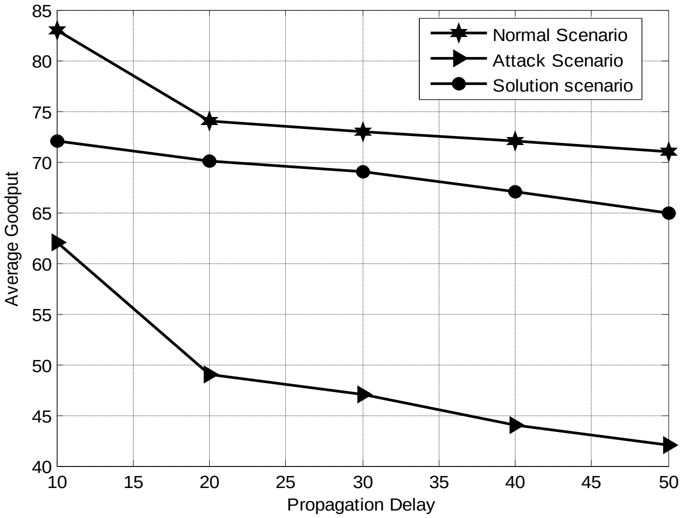


Figure 5.2.2.8 Timeout Attack – Average Goodput (Varying Propagation delay)

5.6 Summary

In this Chapter, two identified denial of service attacks namely burst flooding attack and the Timeout attack is dealt with the possible countermeasures are also dealt in the next part of the same chapter. The effects of these attacks on data traffic routing are shown using simulation graphs and inferred conclusions are discussed with the impacts of the two denials of service attacks which are considered to be vulnerable. This Chapter concludes with the performance analysis and significance of mitigating these denials of service attacks.

CHAPTER 6

SERVICE DISRUPTION ATTACKS FOR OBS NETWORKS

6.1 Introduction

Service disruption attack in optical network communication is highly crucial since the compromised node produces a great impact on the performance of the network. This attack not only decreases the reliability of the network but also crumbles the access of the significant services. In this chapter, two service disruption attacks viz., Burst circulating attack and Land attack are introduced with their associated mathematical model. Further, the impacts of these service disruption attacks are shown through simulation experiments.

6.2 Burst Circulating Attack

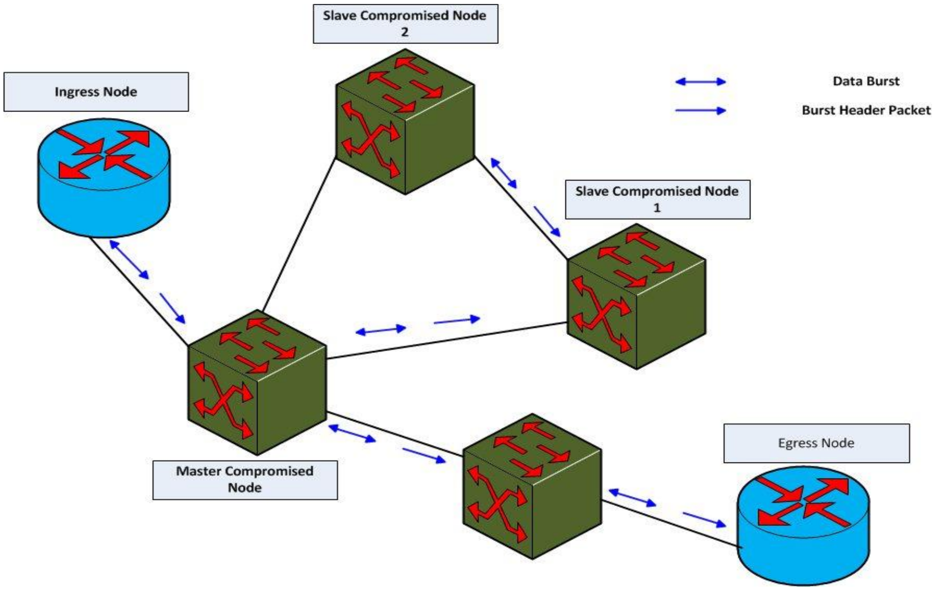


Figure 6.2 Burst Circulating Attack

Burst Circulating attack is a special case of service disruption attack which delays the delivery of the data to the destination. Further, the compromised nodes forms a circuit in which one of one of the nodes acts as a master node and remaining node acts as a slave nodes. The BCH reaching the master node circulates among the nodes of the circuit formed by the compromised nodes for some a stipulated of time and data burst is also relayed with the BCH in the channel. After some time, the BCH gets from the circuit making and gets forwarded to the correct destination. This attack hence delays the delivery time of DB which may lead to wastage of network resources as shown in Figure 6.2.

6.2.1 Attack Detection and Countermeasures

In case of burst circulating attack, the intermediate core nodes in the Optical Burst Switched network are to be monitored. The monitoring of these nodes can be done by a trusted monitor node. When a session is started, every intermediate node sends the network traffic statistics to the monitor. and the monitor node analyses network traffic verify a new connection is established in virtual source node and burst_id and other required statistics matches with source and destination. If the node's trust value reaches below threshold, inform other nodes, then the ingress node is affected by Burst circulating attack as presented in algorithm 6.1.

The trust monitor estimates (b_i+1) as the first burst lost in the 'i'th duplicate time period. Then, the TCP sender retransmits the missing segments in the next round the successive round and W_{xi} – L segments of bursts is transmitted. Further, the total number of bursts successfully transmitted in the routing is manipulated as given by equation (1) and manipulate the expected contention free factor for successful burst delivery as given by equation (2) under the expectation of h_i, which is approximately equal to the product of β and expectation of W_x. If, the expected contention free factor for successful burst delivery is below the threshold, call Burst circulation attack mitigation. Finally, verify the certificate authorities based on digital signature with burst id, burst size, number of packets inside the burst and reduce the node's trust value if the statistics mismatches. Else, if the node's trust value reaches below threshold, inform other nodes.

Then, the monitor node analyses network traffic verify a new connection is established in virtual source node and burst_id and other required statistics matches with source and destination In contrast to the centralized network, there is an absence of central controller in a distributed network.

6.2.1.1 Contention Aware Model

The burst circulation attack mainly occurs due to the contention of the network. Hence, a contention detection based mathematical model is formulated for encountering them as follows,

Suppose, if (b_i+1) is the first burst lost in the ‘i’th duplicate time period. Then, the additional segment of bursts is sent only when the (b_i+1) burst is sent and lost. Now, The TCP sender retransmits the missing segments in the next round and hence, in the successive round W_{xi} – L segments of bursts will be transmitted. where Wxi is the current window size and ‘L’ is the number of burst segments lost in the previous rounds. Further, the total number of bursts successfully transmitted is given by (1).

$$Y = ai + hi + \mathrm{Wxi} - \mathrm{L} \qquad (1)$$

Since, expectation of h_i is approximately equal to the product of β and expectation of W_x. The expected contention free factor for successful burst delivery is given by (2)

$$E[Y] =E[u] + [\beta+1][E[Wi] - L] \qquad (2)$$

Then, the monitor node analyses network traffic verify a new connection is established in virtual source node and burst_id and other required statistics matches with source and destination In contrast to the centralized network, there is an absence of central controller in a distributed network. The distributed network can perform independently among the nodes, where their wavelength is accessible along the corresponding path that is not known to the source and destination nodes. The currently available wavelength information along the forward path is collected by means of PROBE packet in the backward reservation.

The destination node is aware of the wavelength available along the path with the PROBE packet. The information at each node is gathered by PROBE packet are outdated due to the link propagation or processing delay, since the possibility remains of request is blocked by the utility of PROBE based inspection.

Further blockings are broadly categorized into two forms namely, Forward blocking (due to the insufficient network capacity there is no wavelength available between the source and destination nodes and the blocking occurs in forward direction) and Backward blocking: due to vulnerable period between the RESV packet and PROBE packet passing an intermediate node in backward direction. The RESV packet arrive at an intermediate node, the blocking occurs and the node found the wavelength in the packet has been previously reserved by the light path request which is arrived earlier.

According to the observation, we can estimate the blocking probability P_b as:

$$P_b=f_b+ (1-f_b)*bB \qquad (3)$$

where, f_b is caused by forward blocking and bB is caused by backward blocking.

The backward blocking (bB) makes blocking probability (P_b) relatively high since the wavelength resource are insufficient, the forward blocking is expected to upgrade the situation by using a wavelength converter by selecting another route. In backward blocking, we prevent the PROBE information obsolete without this expensive technology. When the traffic load is low, then the backward blocking has a large influence. As a result for high speed light path communications it is important to reduce the backward blocking.

Algorithm 6.1 Burst circulating attack detection Algorithm

```
recv (struct * PACKET packet)
 {
     A)  Determine node Type from packet.
```

```
        if ((node Type = ‘intermediate core node’) OR (node Type = ‘egress node’))
{
    B) Extract burst id, source, destination, num_of_packets, burst_size from packet.
    C) Create a new packet and store the extracted information inside the new packet.
    D) Send the new packet to the trusted node.
    E) Issue the certificate using digital signature
}
else if (node Type == ‘trusted_node’)
{
    A) Extract statistics from packet.
    B) Insert the statistics into the linked list based on burst id.
    C) Collect some more statistics.
    D) Estimate ($b_i$+1) as the first burst lost in the ‘i’$^{th}$ duplicate time period.
    E) The TCP sender retransmits the missing segments in the next round the
       successive round and $W_{xi}$ – L segments of bursts is transmitted.
    F) Determine the total number of bursts successfully transmitted as given by
       equation (1)
    G) Manipulate the expected contention free factor for successful burst delivery as
       given by equation (2) under the expectation of $h_i$ ,which is approximately
       equal to the product of $\beta$ and expectation of $W_x$.
    H) If, the expected contention free factor for successful burst delivery is below
       the threshold, call Burst circulation attack mitigation.
    I) Finally, verify the certificate authorities based on digital signature with burst
       id, burst size, number of packets inside the burst and reduce the node’s trust
       value if the statistics mismatches.
    J) If the node’s trust value reaches below threshold, inform other nodes.
}
}
```

6.2.2 Simulation Results and Analysis

The disruption of service security attack like burst circulating attack is simulated on a 14 node NSF network configuration with the simulation parameters given in Table 6.1 and with the assumption of a single random compromised node and it is explained in normal scenario, attack scenario and solution scenario.

Table 6.1 Simulation Parameters for burst circulating attack

Number of Electronic Nodes	28
Number of Optical Nodes	14
Arrival Rate	0.01ms
Total Simulation Time	50 ms
Number of TCP/IP Connections	18
Number of OBS Connections	17
Number of Packets	200
Number of Channels	3
Link Speed	1GB

The performance metrics used for studying the CAMM approach proposed for Burst circulating Attack are burst blocking probability, Average Goodput, Burst loss probability and Burst throughput. The definition for the above mentioned performance metrics are as follows:

Burst blocking probability: It is defined as the ratio of the number of burst data received by the destination during transmission to the number of burst data actually expected to be delivered at the destination.

Average Goodput: It is defined as application level throughput, i.e. the number of useful information bits delivered by the network to a certain destination per unit of time.

Burst loss probability: It is defined as the probability of number of burst data lost during transmission to the number of burst data expected to be delivered at the destination

Burst throughput: It is defined as the gross bit rate that is transferred physically in to the reliable channel between the source and destination.

Further, the performance evaluation of proposed CAMM approach is carried out through two experiments. In experiment 1, the superiority of proposed CAMM approach is studied based on the evaluation parameters like Burst blocking probability, Average Goodput, Burst loss probability and Burst throughput with respect to three scenarios viz., Normal scenario, Attack scenario and Solution scenario by varying the load parameters. Here normal scenario refers to the performance of the network with ideal network characteristics.

Whereas, the attack scenario depicts the performance of the network in case of Burst Hijacking Attack and solution scenario illustrates the performance of the network when CAMM approach is implemented with the transmission protocol. In case of experiment 2, the performance of proposed CAMM is studied based on varying the time parameters.

6.2.2.1 Performance Evaluation - Experiment 1

The experiment 1 is carried out to illustrate the impact of Burst Hijacking Attack during the data transmission of the network and at the same time, the analysis about the improvement in the performance when CAMM approach is implemented on the transmission protocol.

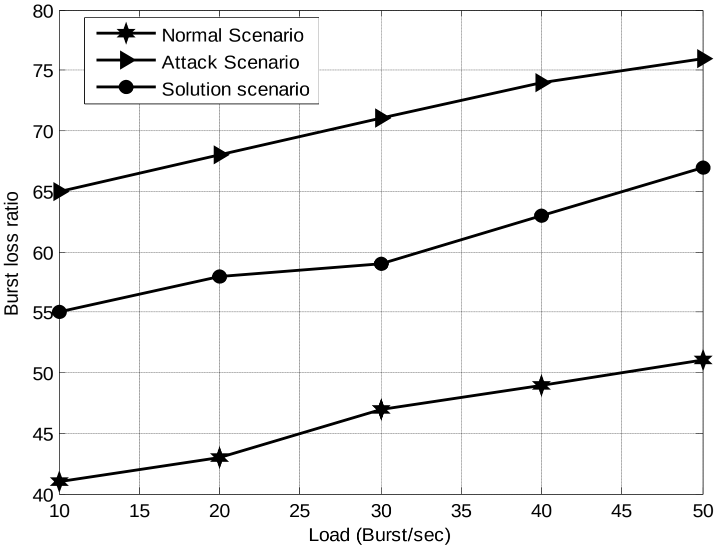


Figure 6.2.2.1.1 Burst Circulating Attack – Burst Loss Ratio (Varying Load)

The Figure 6.2.2.1.1 illustrates the plots for obtained burst loss ratio by varying the load factor of a network in three different scenarios. The obtained results indicates that the solution scenario exhibits the minimum value of burst loss ratio as 63.45% which is 12.56% lesser than that of the attack scenario. Also the value obtained for solution scenario is very near to the value of normal scenario.

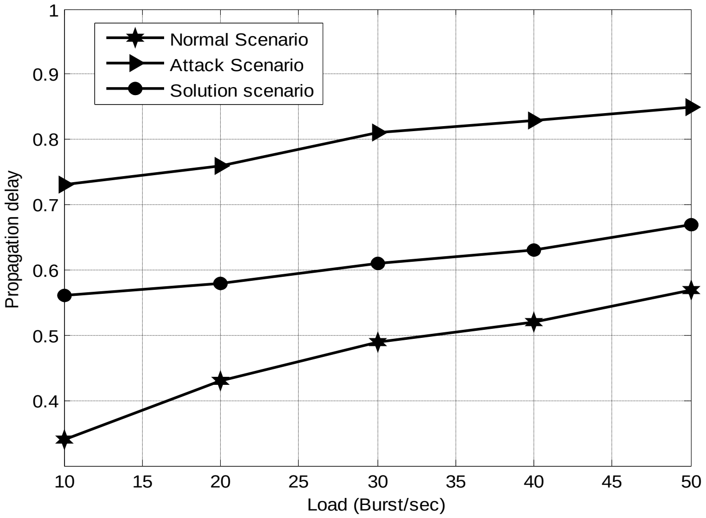


Figure 6.2.2.1.2 Burst Circulating Attack – Propagation delay (Varying Load)

The Figure 6.2.2.1.2 shows the propagation delay plots obtained by varying the load of the network in three different scenarios. From the obtained results it is obvious that, the solution scenario exhibits only minimum propagation delay as 0.647 which is 20.45% lesser than that of the propagation delay exhibited during the attack scenario. This significant improvement in performance is due the deployment of CAMM in the transmission protocol.

Further, the Figure 6.2.2.1.3 shows the plot of burst throughput obtained by varying the load value. This increase in burst block probability rate also increases the influence of the attack in the network which in turn decreases the performance of the network in terms of burst throughput. But, the presence of CAMM model in the NSF network increases the burst throughput to the maximum value of 54.56% which is 12.34% higher than that of the attack scenario.

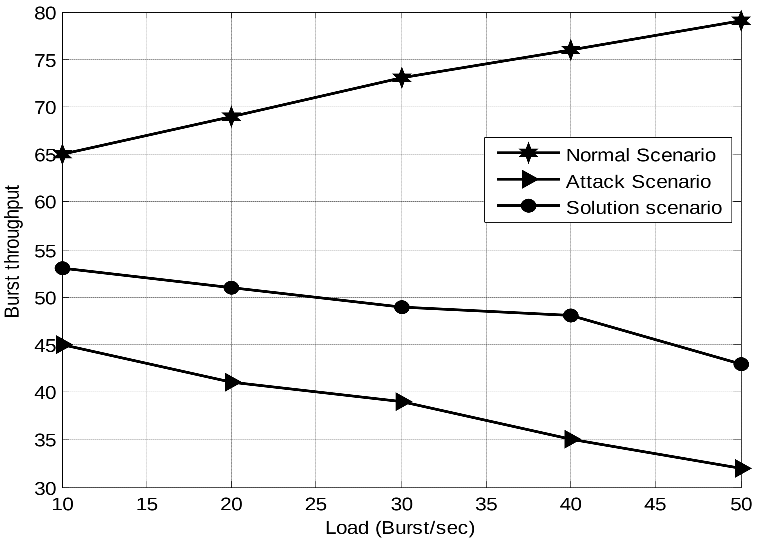


Figure 6.2.2.1.3 Burst Circulating Attack – Burst throughput (Varying Load)

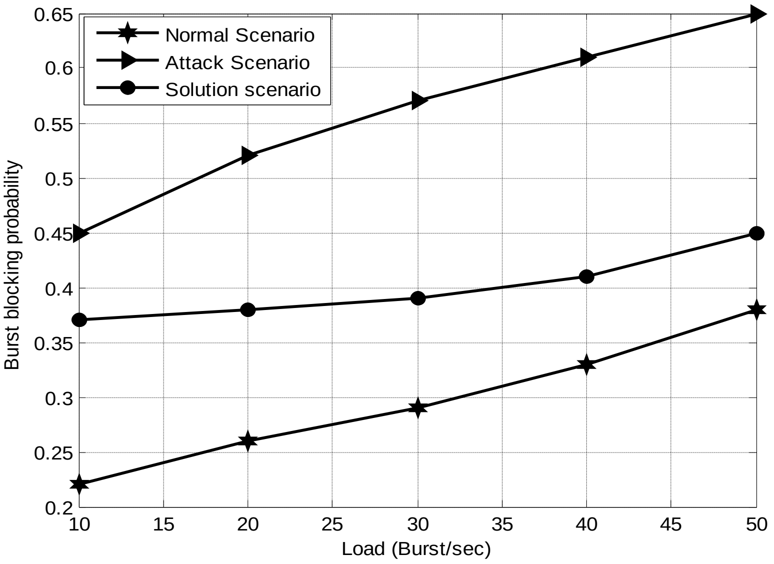


Figure 6.2.2.1.4 Burst Circulating Attack – burst blocking probability (Varying Load)

In addition to this, the Figure 6.2.2.1.4 portrays the plots of burst block probabilities derived in the various scenarios by varying the load of the network. The values of the burst block probability indicates that the impact of burst hijacking attack in this network environment. The increase in the burst block probability rate degrades the performance of the network in terms of burst throughput.

The graphical figure illustrates that, the presence of CAMM in the network environment mitigates the burst flooding attack and decreases the burst block probability to the minimum extent of 0.467. Whereas, the average burst block probability value obtained in attack scenario is 0.698 which is 38% more than that of the solution scenario value. The obtained results conclude that, the deployment of CAMM in the network scenario mitigates the burst flooding attack in a robust manner and hence increases the performance of the network by increasing the burst through rate.

6.2.2.2 Performance Evaluation - Experiment 2

The experiment 2 is carried out to illustrate the impact of Burst Hijacking Attack during the data transmission of the network and at the same time, the analysis about the improvement in the performance when CAMM approach is implemented on the transmission protocol.

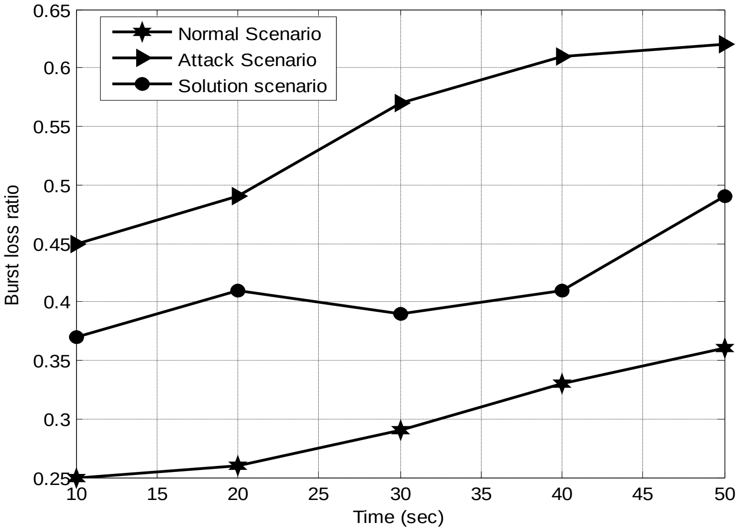


Figure 6.2.2.2.1 Burst Circulating Attack – Burst Loss Ratio (Varying Time)

The Figure 6.2.2.2.1 illustrates the plots for obtained burst loss ratio by varying the time factor of a network in three different scenarios. The obtained results indicates that the solution scenario exhibits the minimum value of burst loss ratio as 0.45 which in turn improves the network performance to greater extent of 23.56% than that of the performance exhibited in the attack scenario.

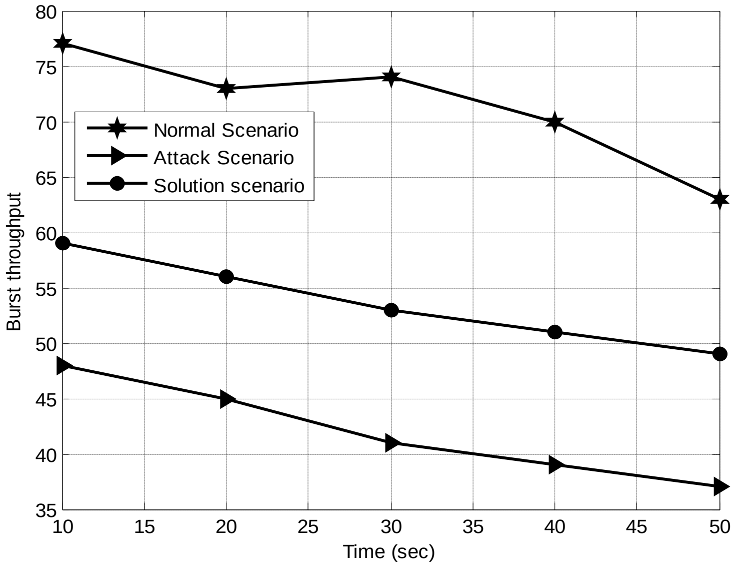


Figure 6.2.2.2.2 Burst Circulating Attack – Burst through put (Varying Time)

The Figure 6.2.2.2.2 shows the burst through put plots obtained by varying the time factor in three different scenarios. The results indicate that the solution scenario shows the high burst throughput value of 60.23%, which is 23% greater than that of the attack scenario.

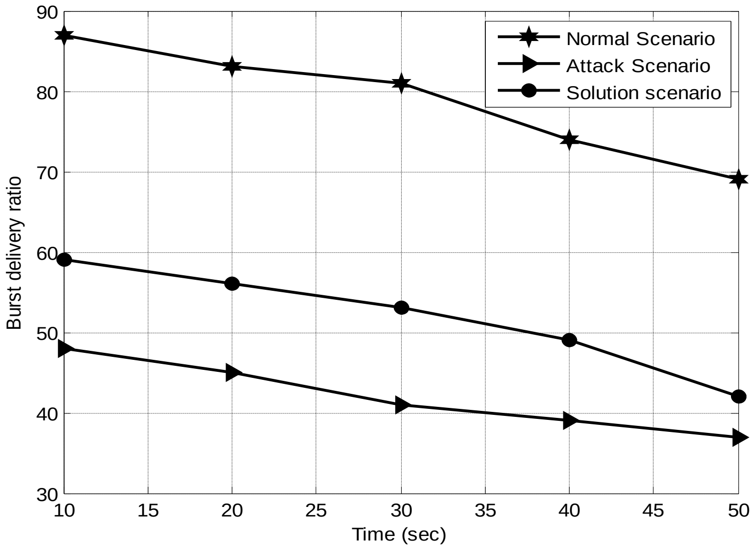


Figure 6.2.2.2.3 Burst Circulating Attack – Burst delivery ratio

The Figure 6.2.2.2.3 shows the burst delivery ratio plots obtained by varying the time factor in three different scenarios. The results indicate that the solution scenario shows the high burst delivery ratio of 60.23%, which is 32% greater than that of the attack scenario. This increase in packet delivery rate is due to the deployment of CAMM approach in the transmission protocol.

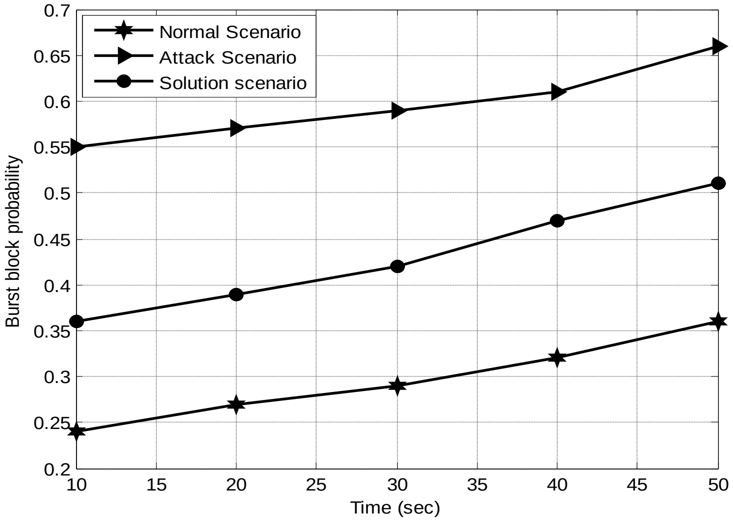


Figure 6.2.2.2.4 Burst Circulating Attack – Burst block probability

In addition to this, the Figure 6.2.2.2.4 portrays the plots of burst block probabilities derived in the various scenarios by varying the time of the network. The values of the burst block probability indicates that the impact of circulating attack in this network environment. The increase in the burst block probability rate degrades the performance of the network in terms of burst throughput. The graphical figure illustrates that, the presence of CAMM in the network environment mitigates the circulating attack and decreases the burst block probability the average value of 0.467. Whereas, the average burst block probability value obtained in attack scenario is 0.878 which is 39% more than that of the solution scenario value. The obtained results conclude that, the deployment of CAMM in the network scenario mitigates the burst circulating attack in a robust manner and hence increases the performance of the network by increasing the burst through rate. On the whole, the proposed CAMM approach mitigated the burst circulating attack at the rate of 54% which is due improve the burst through put and burst delivery ratio ad decreases the burst block probability.

6.3 Land Attack

In Land attack, if any one of the optical node is compromised node in the OBS network, it makes a copy of the burst header and changes the destination address to the source address by the attacker. That means the compromised node redirects the burst header to source address itself. Thus, the burst header never reaches the destination. In the other way, it just escapes of being caught; the compromised node just makes a copy then leaves the bust header back into the normal channel. But, the copied burst header address will never be changed in the source address. Thus, the illegal burst header reaches the source leading to congestion traffic in the channels due to contention as shown in Figure 6.3.

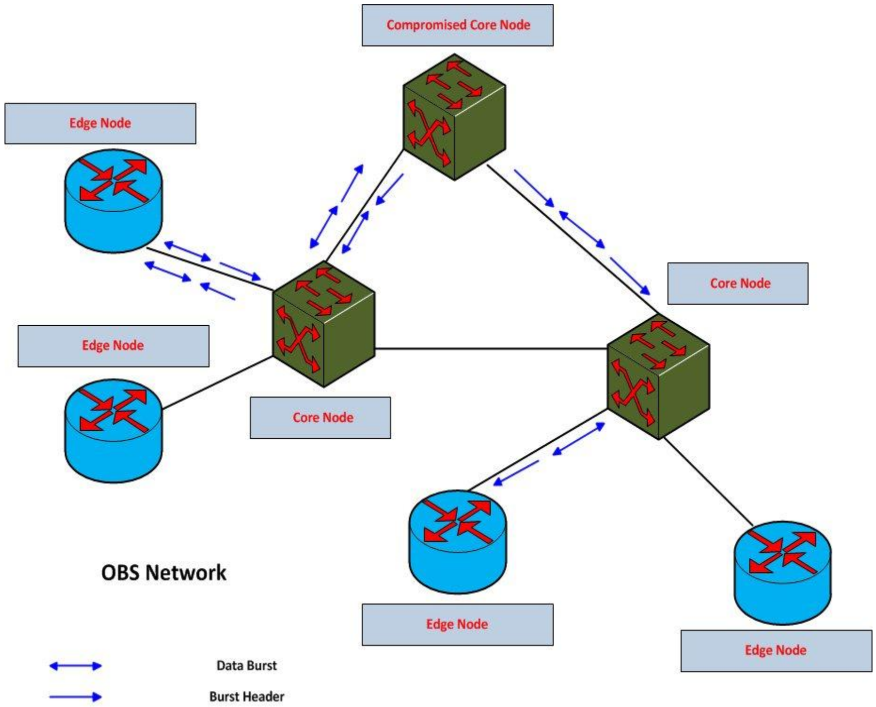


Figure 6.3 Land Attack

6.3.1 Attack Detection and Countermeasures

In case of Land attack, the intermediate core nodes in the Optical Burst Switched network are monitored by a trusted monitor node and confidentiality is achieved through RSA algorithm with private key and public key. Extract source id, destination id and burst id form BCH, then as the first step, decrypt the destination id using the private key of intermediate node. Next, in the second step, encrypt the Destination id from BCH using the same process considering this node and next node as one pair.

Then, based on the life time of the path' P_i' manipulate the overall path stability of the route PS_{route} . Further, compute Stratified $'\alpha'$ and if the Stratified $'\alpha'$ is less than the reliability vale of threshold, normal routing is enabled. In addition, the monitor node will analyze these values associated with encryption and decryption mechanism for each core node and look into the nodes whether they are malicious in the session. If the node's digital signature will be compared and find the malicious node and the attack is named as land attack as presented by algorithm 6.2.

6.3.1.1 Stratified Alpha Coefficient Model

The Land Attack is migrated using a stratified alpha coefficient as enumerated as follows,

Let, the life time of the path P_i can be defined as

$$P_i = \sum_{i=1}^{n} C_{j,j+1(t)} \qquad (1)$$

where $C_{j,j+1}$ the cost of transmission and 'n' is the number of burst packets forwarded in the path then, life time of the path L_{path} is given by

$$L_{path} = \frac{MIN_{LP}}{P_i} \qquad (2)$$

where MIN_{LP} indicates the minimum life time of the path.

Therefore, the overall path stability of the route PS_{route} is manipulated as the weighted sum of $E_u(i)$, F_m, L_{path} given by (1), (2), (5).

$$PS_{route} = \alpha E_u(i) + \beta F_m + \gamma L_{path} \qquad (3)$$

If the value of PS_{route} is less than the stability threshold of 0.4 as specified in [ ], then alternate burst generation takes place. Consider a scenario with k sessions, $i = 1, \ldots, k$ with session level total scores of X_{ts} and trustd_i.

Let x represent the total score obtained as the weighted sum of X_{ts}. Then the trustworthy of the node is manipulated using stratified coefficient alpha from (24) as

$$\text{Stratified } \alpha = 1 - \sum_{i=1}^{k} \frac{{\sigma_{xts}}^{2}(1-\alpha_{L})}{{\sigma_{x}}^{2}} \qquad (4)$$

The stratified α is calculated through the process of wavelength prediction table which is updated by checking the wavelength topology when the selected-λ is modified. The wavelength prediction table is modified by reordering the wavelength topology, by which the PROBE packet is relayed towards the downstream node. The behavior of the destination node can be determined by receiving the PROBE packet through the intermediate node. Now checking the wavelength availability is performed on the wavelength prediction table. If the wavelength is available then discard the frequency range otherwise proceed to the next step.

Finally the wavelength topology is modified and the reply packet is returned to the source node. Basically, the optical burst switching techniques is how the resources in the network are reserved. A rehash of the Asynchronous Transfer Mode (ATM) which is also known as ATM Block Transfer (ABT) is known as optical burst switching techniques. It has two different versions, first is with immediate transmission and the second one is with delayed transmission.

The distinction between immediate transmission and the delayed transmission is, there is no time gap between the control burst and data burst in immediate transmission but in a delayed transmission there is an special offset value in which there exist a time separation between control burst and data burst and it can be consider to reach the node and also the processing time.

Further, in case of an optical signal progresses along its path, it may get altered by different physical processes in the optical path and nodes through which it propagates. When those kind of alterations result in the signal degradation, then these processes are called as impairments These characteristics might be an significant constraint for consideration since a GMPLS control plane could support path establishment and maintenance in wavelength that are utilized in the switched optical networks.

Algorithm 6.2 Land attack detection Algorithm

```
If (node == ingress node)
{
    Step: 1 Extract source id, destination id and burst id from BCH.
    Step: 2 Encrypt the destination id using RSA algorithm using the public key of the ingress node.
    Step: 3 The encrypted signed code get stored the signature field
    Step: 4 Find the Hash code for the data using SHA algorithm and encrypt the data
    Step: 5 store the hash value in the data field
}
Else if (node == Intermediate node)
{
    Step: 1 Extract source id, destination id and burst id form BCH
    Step: 2 Decrypt the destination id using the private key of intermediate node.
    Step: 3 Encrypt the Destination id from BCH using the same process considering this node and next node as one pair.
    Step 4: Based on the life time of the path' $P_i$' using equation(1)
    Step5: Manipulate the overall path stability of the route $PS_{route}$ using equation (3)
    Step6: Compute Stratified $'\alpha'$ using equations (1), (2)and (3)
    Step 7: If the Stratified $'\alpha'$ is less than the reliability vale of threshold,
        Call normal routing
    Step: 8 Else, Send the BCH and DB to next node.
}
Else (node==Egress node)
{
    Step: 1 Extract source id, destination id and burst id form BCH
    Step: 2 Decrypt the destination id using RSA algorithm using the private of the egress node.
Step: 3 Decrypt the Data field and find the hash value using SHA
Step: 4 Compare the both signature values
}
```

6.3.2 Simulation Results

The disruption of service security attack land attack is simulated on a 14 node NSF network configuration with the simulation parameters given in Table 6.2 and with the assumption of a single random compromised node and it is explained in normal scenario, attack scenario and solution scenario.

Table 6.2 Simulation Parameters for land attack

Number of Electronic Nodes	28
Number of Optical Nodes	14
Arrival Rate	0.01ms
Total Simulation Time	50 ms
Number of TCP/IP Connections	18
Number of OBS Connections	17
Number of Packets	200
Number of Channels	3
Link Speed	1GB

The performance metrics used for studying the Stratified Alpha Reliability Coefficient Based Mitigation Model proposed for Land attack are burst blocking probability, Average Goodput, Burst loss probability and Burst throughput. The definition for the above mentioned performance metrics are as follows:

Burst blocking probability: It is defined as the ratio of the number of burst data received by the destination during transmission to the number of burst data actually expected to be delivered at the destination.

Average Goodput: It is defined as application level throughput, i.e. the number of useful information bits delivered by the network to a certain destination per unit of time.

Burst loss probability: It is defined as the probability of number of burst data lost during transmission to the number of burst data expected to be delivered at the destination

Burst throughput: It is defined as the gross bit rate that is transferred physically in to the reliable channel between the source and destination.

6.3.2.1 Performance Evaluation - Experiment 1

In experiment 1, the impact of Land Attack is studied with respect to load based on Burst loss probability, Propagation delay , Burst loss probability and Burst throughput with respect to three scenarios namely Normal scenario, Attack scenario and Solution scenario are illustrated through Figures 6.3.2.1.1, 6.3.2.1.2, 6.3.2.1.3 and 6.3.2.1.4.

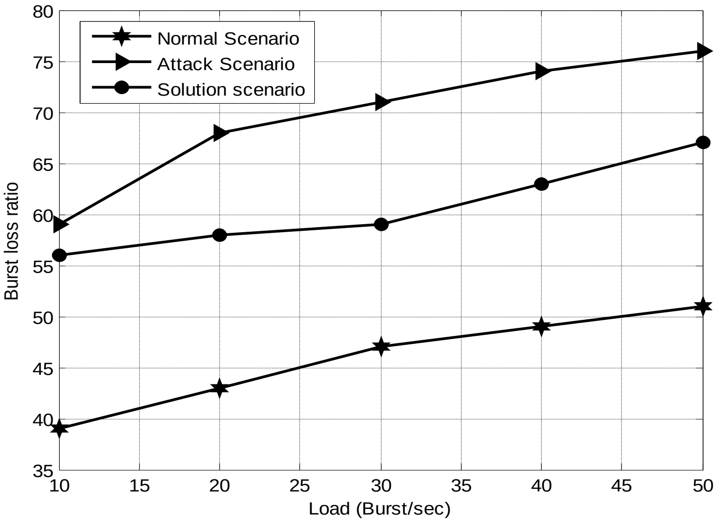


Figure 6.3.2.1.1 Land Attack – Burst loss ratio (Based on varying load)

The Figure 6.3.2.1.1 portrays that the burst loss probability increases with respect to varying load for all the three scenarios viz., Normal scenario, Attack scenario and Solution scenario. It is evident that the Burst loss probability increases drastically in the Attack scenario when compared to the Normal scenario to a maximum level of 24%. But, when the stratified alpha coefficient based Mitigation mechanism for Land attack is implemented, it decreases the Burst loss probability by 21%.

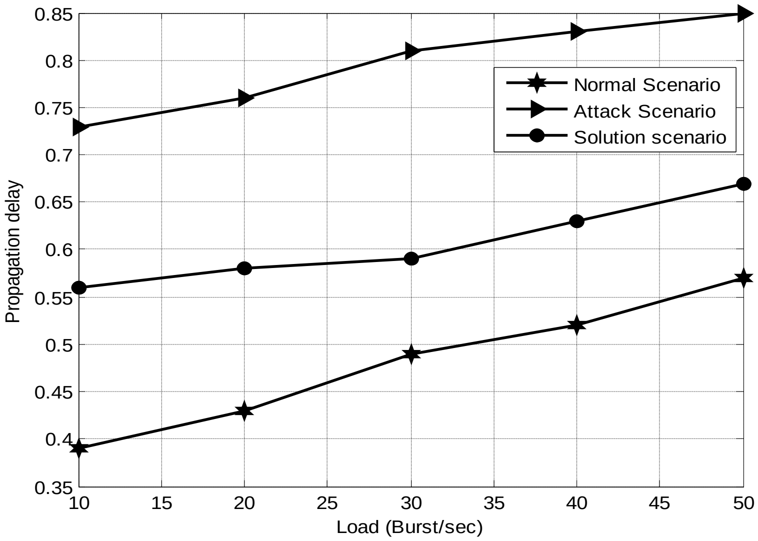


Figure 6.3.2.1.2 Land Attack – Propagation delay (Based on varying load)

Further, Figure 6.3.2.1.2 portrays that the Propagation delay increases with varying amount of load for all the three scenarios viz., Normal scenario, Attack scenario and Solution scenario. It is also evident that the Propagation delay increases drastically in the Attack scenario when compared to the Normal scenario to a maximum level of 18%. But, when the stratified alpha coefficient based Mitigation mechanism for Land attack is implemented, it increases Propagation delay by 15% with respect to varying time.

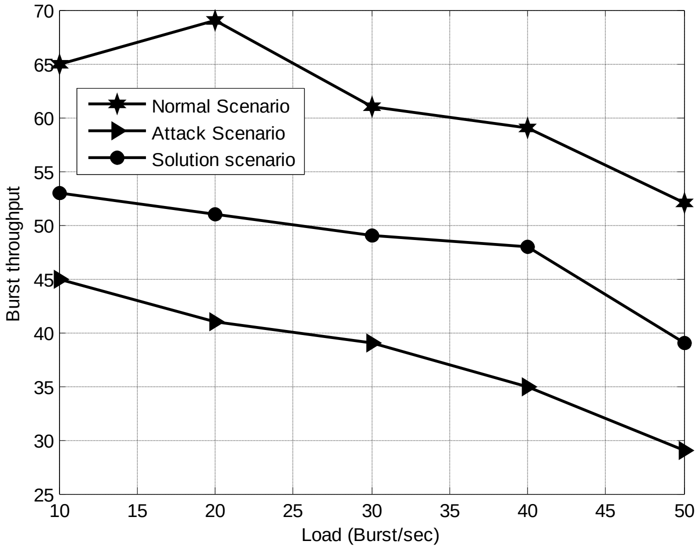


Figure 6.3.2.1.3 Land Attack – Burst throughput (Based on varying load)

Furthermore, Figure 6.3.2.1.3 depicts that the Burst throughput with respect to varying load increases for all the three scenarios viz., Normal scenario, Attack scenario and Solution scenario.

It is also evident that the Burst throughput drastically decreases in the Attack scenario when compared to the Normal scenario to a maximum level of 29 %. But, when the stratified alpha coefficient based Mitigation mechanism for Land attack is implemented, it decreases Burst throughput by 20% with respect to varying time.

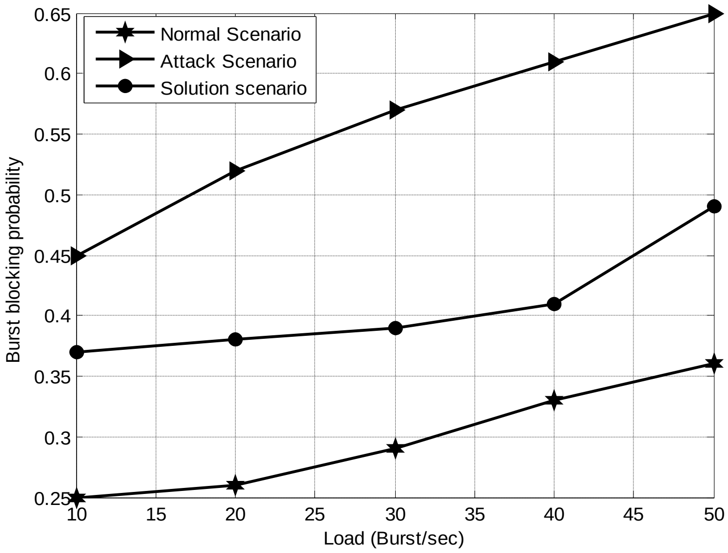


Figure 6.3.2.1.4 Land Attack – Burst blocking probability
(Based on varying load)

In addition, Figure 6.3.2.1.4 depicts that the Burst blocking probability with respect to varying load increases for all the three scenarios viz., Normal scenario, Attack scenario and Solution scenario. It is also evident that the Burst blocking probability drastically increases in the Attack scenario when compared to the Normal scenario to a maximum level of 28%.But, when the alpha coefficient based Mitigation mechanism for Land attack is implemented and it decreases the Burst blocking probability by 16%.

6.3.2.2 Performance Evaluation - Experiment 2

In experiment 2, the impact of Land Attack is further studied with respect to time based on Burst loss ratio, Burst throughput, Burst delivery ratio, Burst block probability with respect to three scenarios namely Normal scenario, Attack scenario and Solution scenario are illustrated through Figures 6.3.2.2.1, 6.3.2.2.2, 6.3.2.2.3 and 6.3.2.2.4.

The following Figure 6.3.2.2.1 portrays that the burst loss probability increases with respect to time for all the three scenarios viz., Normal scenario, Attack scenario and Solution scenario.

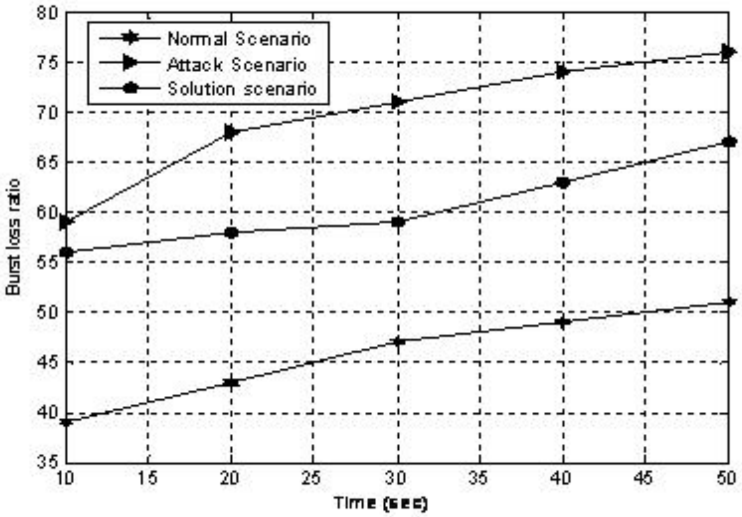


Figure 6.3.2.2.1 Land Attack – Burst loss ratio (Based on varying time)

It is evident that the Burst loss probability increases drastically in the Attack scenario when compared to the Normal scenario to a maximum level of 21%. But, when the stratified alpha coefficient based Mitigation mechanism for Land attack is implemented, it decreases the Burst loss probability by 16%.

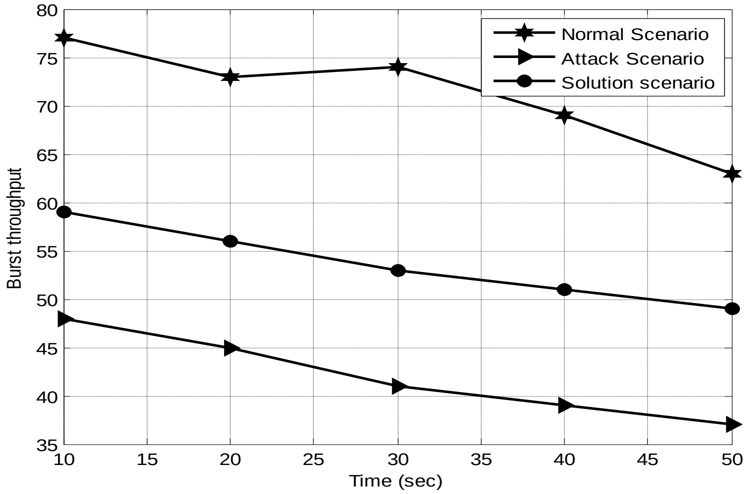


Figure 6.3.2.2.2 Land Attack – Burst throughput (Based on varying time)

Further, Figure 6.3.2.2.2 portrays that the Burst throughput decreases with varying amount of time for all the three scenarios viz., Normal scenario, Attack scenario and Solution scenario. It is also evident that the Burst throughput decreases drastically in the Attack scenario when compared to the Normal scenario to a maximum level of 18%. But, when the stratified alpha coefficient based Mitigation mechanism for Land attack is implemented, it increases Burst throughput by 10% with respect to varying time.

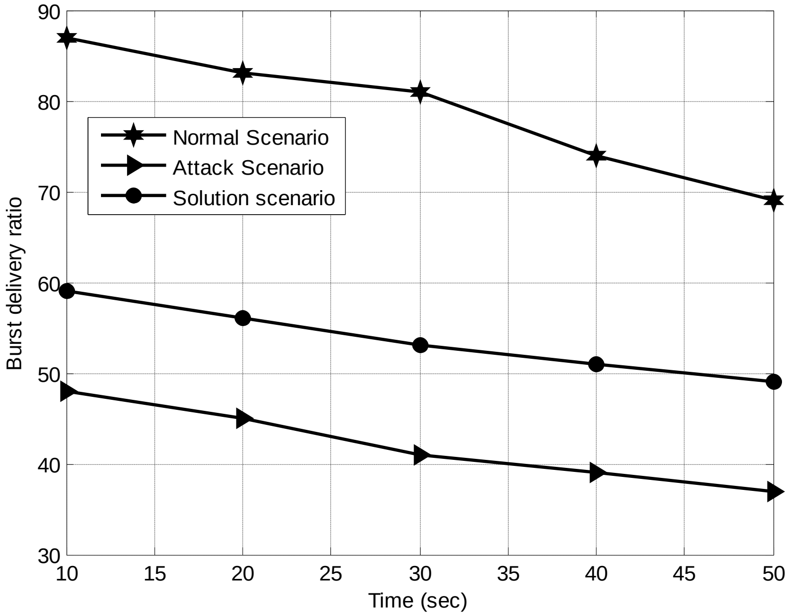


Figure 6.3.2.2.3 Land Attack – Burst delivery ratio (Based on varying time)

Furthermore, Figure 6.3.2.2.3 depicts that the Burst delivery ratio with respect to time decreases for all the three scenarios viz., Normal scenario, Attack scenario and Solution scenario. It is also evident that the Burst throughput drastically decreases in the Attack scenario when compared to the Normal scenario to a maximum level of 26 %.But, when the stratified alpha coefficient based Mitigation mechanism for Land attack is implemented, it increases Burst delivery ratio by 16% with respect to varying time.

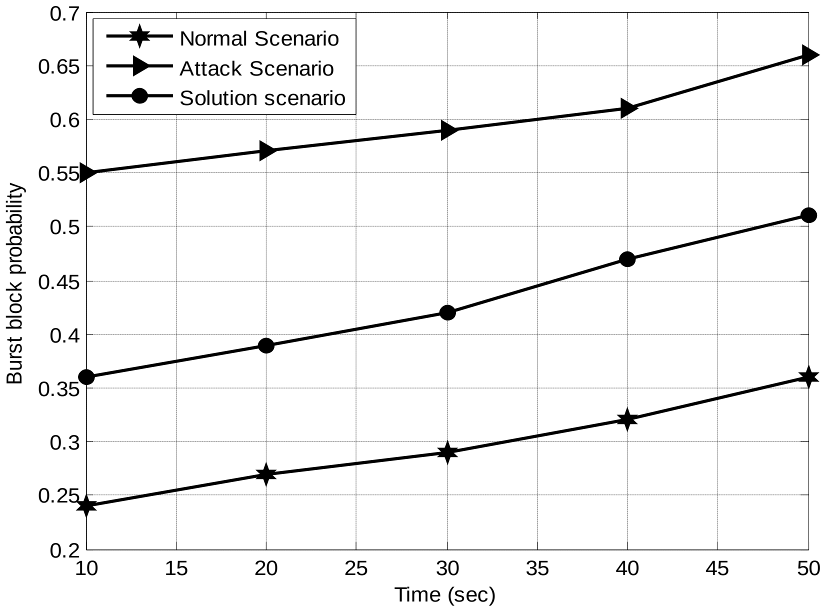


Figure 6.3.2.2.4 Land Attack – Burst blocking probability (Based on varying time)

In addition, Figure 6.3.2.2.4 depicts that the Burst blocking probability with respect to time increases for all the three scenarios viz., Normal scenario, Attack scenario and Solution scenario. It is also evident that the Burst blocking probability drastically increases in the Attack scenario when compared to the Normal scenario to a maximum level of 24%.But, when the alpha coefficient based Mitigation mechanism for Land attack is implemented and it decreases the Burst blocking probability by 14%.

6.6 Summary

In this Chapter, the impact of two identified disruption of service attacks namely burst circulating attack and the land attack are analyzed with various performance metrics and significance of the solution associated to each of those attack are elaborated. The possible detection and countermeasures for this attack are also dealt with the mathematical model for reliable identification and isolation. The effects of these attacks on data traffic routing are shown using simulation graphs.

CHAPTER 7

CONCLUSION AND FUTURE RESEARCH DIRECTIONS

Exponential increase in the internet traffic demands induces the researchers to explore optical domain for routing for sheer velocity. Optical Burst Switching caters optical switching techniques with minimal data loss compared to the other optical architectures like Optical Circuit Switching and Optical Packet Switching. This chapter highlights the summary of research contributions and further elaborates on the open security issues for future research in Optical Burst Switched networks.

7.1 Conclusion

Optical networks are the future network that uses the technology to provide an end to end optical path among the communicating parties. But, an optical burst switched network possesses many limitations or vulnerabilities that are quite natural to sustain in case of the security attacks. The present work addresses the security problems in OBS networks and the main goal of the proposed system is to secure the OBS network. In this section, we have presented the outcomes and impacts of security attacks viz., burst hijacking attack, burst flooding attack, burst circulating attack, timeout attack and land attack on OBS networks. Further, the statistical approach for detecting and preventing attacks are discussed for the normal scenario, attack scenario and attack solution scenario through ns2 simulator with the modified nOBS patch.

Furthermore, the mitigation mechanisms proposed for each of the potential attacks infers the following striking points with respect to the network performance in terms of Burst blocking probability, Average Goodput, Burst loss probability and Burst throughput obtained by varying the load and time. They are:

a) The Cornbach Alpha Reliability Coefficient Based Mitigation mechanism for Burst Hijacking attack in an average decreases the Burst block probability, Burst loss probability by 23%, 21% respectively and increases the Average Goodput and Burst Throughput by 18% and 22%.

b) The variations alpha coefficient approach of Fake Spectral Attack mitigation in an average decreases the Burst block probability, Burst loss probability by 18%, and 16% respectively and increases Average Goodput and Burst Throughput by 26 % and 20% respectively.

c) The Kappa Coefficient based mitigation model for detecting Burst Flooding Attack based mitigation mechanism for Burst Flooding attack in an average decreases the Burst block probability, Burst loss probability by 34% and 27% respectively and increases the Average Goodput and Burst Throughput by 22% and 22% respectively.

d) The Optimal Time Threshold based mitigation approach for Timeout Attack in an average decreases the Burst block probability, Burst loss probability by 21% and 26% respectively and increases the Average Goodput and Burst Throughput by 17% and 29% respectively.

e) The contention detection based mathematical model for burst circulating attack in an average decreases the Burst block probability, Burst loss probability by 16%, 26% respectively and increases the Average Goodput and Burst Throughput by 15% and 29% respectively.

f) The Stratified Alpha Reliability Coefficient based Mitigation mechanism for Land attack in an average decreases the Burst block probability, Burst loss probability by 24%, 20% respectively and increases the Average Goodput and Burst Throughput by 28% and 25% respectively.

7.2 Future Research Directions

The future research direction of this research can be further extended into two major issues namely QoS issues and security issues. This work can be enhanced in the future with the following ideas in the mind:

- A cost effective and practical countermeasure for the identified security attacks over OBS routing can be implemented.
- The work on node capability based routing can be extended to manycasting with constraints like resources, time and vulnerabilities to select the quorum group.
- The results can be enhanced through Real Test bed simulation, which is not done for the work due to unavailability of optical nodes and links.
- More statistical mathematical mitigation mechanism may be formulated and analyzed for the identified potential attacks.

REFERENCES

[01]. Mukherjee .B, “WDM Optical Communication Networks: Progress and Challenges,” *IEEE Journal on Selected Areas in Communications*, pp. 1810-1823, October 2000.

[02]. Verma .S, Chaskar .H, and Ravikanth .R, “Optical Burst Switching: A Viable Solution for Terabit IP Backbone,” *IEEE Network*, pp. 48-53, November/December 2000.

[03]. Battestilli .T and Perros .H, “An Introduction to Optical Burst Switching”, *IEEE Journal of Optical Communications*, vol. 41, no.8, pp. 510–515, August 2003.

[04]. Chen .Y, Qiao .C and Yu .X, “An Optical Burst Switching: A New Area in Optical Networking Research,” *IEEE Journal of Networks*, vol. 18, no. 5, pp. 16–23, 2004.

[05]. Qiao .C and Yoo. M, “Optical burst switching OBS-a new paradigm for an optical internet”, *Journal of High Speed Networks*, vol. 8, no. 1, pp. 69–84, 1999.

[06]. Yu .X, Qiao .C, Liu .Y and Towsley .D “Performance Evaluations of TCP Traffic Transmitted over OBS Networks”, *Tech. Report* 2003-13, *CSE Dept, SUNY Buffalo*, 2003.

[07]. Sunil Gowda, Ramakrishna K Shenal, Krishna M Sivalingam and HakkiCandanCankaya,” Performance Evaluation of TCP over Optical Burst – Switched (OBS) WDM Networks”, *Proceeding of IEEE ICC,* May 2003.

[08]. Arnold Bragg†, Ilia Baldine and Dan Stevenson,” A transport layer architectural framework for OBS networks”, IEEE *comunications magazine*, Dec. 2005.

[09]. Koduru .K, “New Contention Resolution Techniques for Optical Burst Switching,” *Master's thesis, Louisiana State University*, May 2005.

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

[11]. GurayGurel and EzhanKarasan,” Effect of Number of Burst Assemblies on TCP Performance in Optical Burst Switching Networks”, *in Proceedings of the IEEE BROADNETS 2006,* October 2006.

[12]. Turner .J, “Terabit Burst Switching,” *Journal of High Speed Networks,* vol.8, pp. 3-16, January 1999.

[13]. Yoo .M and Qiao .C, “A Novel Switching Paradigm for Buffer-Less WDM Networks,” *In Optical Fiber Communication Conference (OFC),* pp. 177-179, February 1999.

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

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

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

[17]. Ding .A and Poo G.S, “A survey of Optical Multicast over WDM Networks”, *Elsevier Journal of Computer Comm.,,* vol.26, no.2, pp 193-200, Feb. 2003.

[18]. Dozer. K, Gauger .C, Spath .J and Bodamer .S, “Evaluation of Reservation Mechanisms for Optical Burst Switching,” *AEU International Journal of Electronics and Communications,* vol. 55, no.1, pp. 2017- 2022, January 2001.

[19]. Farahmand .F, Jue .J.P, Vokkarane .V et.al., ”A Layered Architecture for Supporting Optical Burst Switching”, *in proceeding of the Advanced Industrial Conference on Telecommunications,* pp. 213–218, 17 October 2005, Dallas, USA.

[20]. Kantarci .B and Oktug .S, "Adaptive Threshold Based Burst Assembly In OBS Networks", *in proceeding of IEEE Canadian Conference on Electrical and Computer Engineering,* (CCECE '06), pp. 1419 – 1422, May 2006, Ottawa, Canada.

[21]. Kaheel .A and Alnuweiri .H, “Quantitative QoS guarantees in labeled optical burst switching networks”, *in proceeding of IEEE Global Telecommunications Conference,* (GLOBECOM), vol. 3, pp. 1747–1753, (29 November-3 December) 2004, Dallas, Texas, USA

[22]. Kocyigit .A, Gokisik .D, Bilgen .S, "All-Optical Networking", *Turkey Journal of Electrical Engineering,* vol.9, no.2, pp. 69-121, January 2001.

[23]. Gipser .T and Kao M.S, “An all-optical network architecture”, *IEEE Journal of Lightwave Technology*, vol. 14 , no. 5, pp. 693-702, May 1996.

[24]. Guo .H, Lan .Z, Wu .J and Gao .Z, "A Testbed for Optical Burst Switching Network", *in proceeding of Optical Fiber Comm. Conf.*, vol.5, pp. 12-17,March 2005, California, USA.

[25]. Liu .D.Q and Liu .M.T, “Differentiated Services and Scheduling Schemes in Optical Burst Switched WDM Networks”, *in proceeding of International conference on Networks,* pp.23-27, 27-30 August 2002, Singapore.

[26]. Ljolje .M, Inkret .R and Mikac .B, “ A Comparative Analysis of Data Scheduling Algorithms in Optical Burst Switching Networks”, *in proceeding of Optical Network Design and Modeling*, pp 493-500, February 2005, Milan, Italy.

[27]. Sreenath .N, Krishna Mohan Reddy .K, Mohan .G and Siva Ram Murthy .C, “Virtual source based Multicast routing in WDM Networks with Sparse Light Splitting”, *IEEE Workshop on High Performance Switching and Routing*, pp. 141-145, May 2001, Dallas, Texas, USA.

[28]. Sreenath .N, Satheesh .K, Mohan .G and Siva Ram Murthy .C, “Virtual Source Based Multicast Routing in WDM Optical Networks”, *in proceeding of International Conference on Networks*, pp. 385-389, 5-8 Sep.,2000, India.

[29]. Zervas .G, Qin .Y, Nejabati .R, Simeonidou .D, "Service Oriented Optical Burst Switched Edge and Core Routers for Future Internet", *in proceeding of 5th International Conference on Broadband Communications, Networks and Systems,* (BROADNETS), pp. 97-104, 8-11 September 2008, London, England.

[30]. Vokkarane .V.M, Jue .J.P, and Sitaraman .S, “Burst Segmentation: An Approach for Reducing Packet Loss in Optical Burst Switched Networks”, *in proceeding of IEEE conference on comm..,*, vol.5, pp. 2673-2677, May 2002, New York, USA.

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

[32]. Tan. G. M. S. K and Chua .K, “Feedback-based offset time selection for end-to-end proportional QoS provisioning in WDM optical burst switching networks”, *Journal of Computer Communications*, vol. 30, no. 4, pp. 904–921, February 2007.

[33]. Chandra P.K, Turuk A.K, Sahoo .B, "Survey on Optical Burst Switching in WDM Networks",*in proceeding of International Conference on industrial and Information systems*, pp.83-88, December 2009, Sri Lanka.

[34]. Shi .K, Fan .G and Xie .H, “Signaling protocol to reduce blocking probability in Optical Burst Switching mesh networks with limited wavelength conversion capabilities”,*in proceeding of 3rd International Conference in communications and Networking*, pp 448-453, 25-27 August 2008, Hangzhou China.

[35]. Yoo .M and Qiao .C, “Supporting multiple classes of services in IP over WDM networks”, *in proceeding of IEEE Global Comm. Conference*, pp 1023–1027, April 1999, Ghana.

[36]. Shihada .B and Ho .P.H, "TCP in OBS: Issues, solutions and challenges", *IEEE Communication Surveys*, vol.10, no.2, pp- 70-86, 2nd quarter 2008.

[37]. Xu .L, Perros H.G, and Rouskas G.N, "Techniques for Optical Packet Switching and Optical Burst Switching", *IEEE Comm. Magazine*, Vol.39, no.1, pp. 136-142, January 2001.

[38]. Zervas .G, Sadeghioon .L, Klonidis .D et.al., "Demonstration of Novel Multi-Granular Switch Architecture on an Application-Aware End-to-End Multi- Bit Rate OBS Network Testbed", *in proceeding of 33rd European Conference on Optical Communication*, (ECOC), pp. 1-2, 16-20 September 2007, Berlin, Germany.

[39]. Yu. X, Qiao .C and Liu .Y, "TCP Implementations and False Time out Detection in OBS Networks", *in proceeding of IEEE Informatics and Comm.*, pp. 774-784, China, 2004.

[40]. Yoo .M, Qiao .C and Dixit .S, "QoS performance of optical burst switching in IP-over-WDM networks", *IEEE Journal of Selected Areas in Comm.*, vol. 18, pp. 2062–2071, October 2000.

[41]. Zhang .H, Jue .J.P and Mukherjee .B, "A review of routing and wavelength assignment approaches for optical WDM Networks", *Optical Network Magazine*, vol.1, no.2, Jan. 2000.

[42]. Zhang .Q, Vokkarane .V, Jue J.P and Chen .B, "Absolute QoS differentiation in Optical Burst Switched networks", *IEEE Journal of Selected Areas Communications*, vol. 22, no. 9, pp. 1781–1795, November 2004.

[43]. Cao .X, Li .J, Chen .Y, and Qiao .C, "Assembling TCP/IP Packets in Optical Burst Switched Networks., *in Proceeding of IEEE Globecom*, December 2002.

[44]. Chlamtac .I, Ganz .A and Karmi .G, "Lightpath communications: An approach to high-bandwidth optical WANs", *IEEE Trans., on Comm.*, vol. 40, no. 7, July 1992.

[45]. Zhang .W, Wu .J, Lin .J, Minxue .W and Jindan .S, "TCP performance experiment on OBS network testbed", *in proceeding of 11th International IFIP TC6 Conf., ONDM and Springer Optical Network Design and Modelling LNCS*, vol. 4534, pp. 186-193, May 2007, Greece.

[46]. Akar .N, Karasan .E, Vlachos .K.G, Varvarigos .E.A, et.al., "A survey of quality of service differentiation mechanisms for optical burst switching networks", *Journal of Optical Switching and Networking*, vol. 7, no. 1, pp. 1–11, January 2010.

[47]. Wei J.Y and McFarland R.I, "Just-In-Time Signaling for WDM Optical Burst Switching Networks," *Journal of Light wave Technology*, vol. 18, Dec. 2000.

[48]. Carena .A, Vaughn .M.D, Gaudino .R, Shell .M, and Blumenthal .D.J, "Opera: An optical packet experimental router architecture with label swapping capability", *IEEE Journal of Light wave Technology*, vol.16, no.12, pp. 2135-2145, Dec., 1998.

[49]. Chen .Y, Hamdi .M, and Tsang .D, “Proportional QoS over OBS networks”, *in Proceeding of IEEE Global Telecommunications Conference,* (GLOBECOM), vol. 3, pp. 1510–1514 , 25-29 November 2001, San Antonio, Texas, USA.

[50]. Guo .H, Wu .J, Xin .L and Lin .J, "Multi-QoS Traffic Transmission Experiments on OBS Network Testbed", *in proceeding of 31st European Conference on Optical Communication,* (ECOC), vol.3, pp. 601-602, 25-29 Sept.,2005, Glasgow, Scotland.

[51]. Yang S. Z. M and Verchere .D, “A QoS supporting scheduling algorithm for optical burst switching DWDM networks”, *in proceeding of IEEE Global Telecommunications Conference,* vol. 1, pp. 86–91, 25-29 Nov. 2001, Texas, USA.

[52]. Yang .L and Rouskas G.N, “Optimal wavelength sharing policies in OBS networks subject to QoS constraints” ,*IEEE Journal of Selected Areas in Comm.,* vol. 25,Dec 2007.

[53]. Yates .J.M, Rumsewicz .M.P and Lacey J.P.R, “Wavelength Converters in dynamically reconfigurable WDM Networks”, *IEEE Comm., surveys and Tutorials,* vol.2, Dec.1999.

[54]. Yang .L and Rouskas .G.N, “Generalized wavelength sharing policies for absolute QoS guarantees in OBS networks”, *IEEE Journal Selected Areas of Communication*, vol. 25, no. 3, pp. 93–104, April 2007.

[55]. Kaheel .A and Alnuweiri .H, “A strict priority scheme for quality-of-service provisioning in optical burst switching networks”, in Proceeding of 8th IEEE International Symposium on Computers and Communication (ISCC), vol. 1, pp. 16–21, 30 June-3 July 2003, Kemer-Antalya, Turkey.

[56]. Kamal A.E and Al-Yatama A.K, "Blocking Probabilities in Circuit-Switched WDM Networks under multicast service", *Elsevier Journal of Performance Evaluation*, vol.47, no.1, pp.43-71, January 2002.

[57]. González-O, Miguel .A, Gonzalez A.S, Jose C.., “Loss Differentiation in Full-Wavelength Conversion Capable networks by Burst cloning”, *IEEE Comm., Letters,* vol. 15, Jan 2011.

[58]. Sahara .A, Ono .T, Yamawaku .J, Takada .A et.al.., "Congestion-Controlled Optical Burst Switching Network with Connection Guarantee: Design and Demonstration",*Journal of Light wave Technology*, Vol. 26, No. 14, pp. 2075-2086, July 15, 2008.

[59]. Yoo .M and Qiao .C, "Just-Enough-Time (JET): A High Speed Protocol for Bursty Traffic in Optical Networks", *in proceeding of IEEE/LEOS Vertical-Cavity Lasers*, pp. 26- 27, 11-13 August 1997,Montreal, Canada.

[60]. Wang .S.Y, "Using TCP Congestion Control to Improve the Performances of Optical Burst Switched Networks", *in proceeding of IEEE Conference on communications*, vol.2, pp.1438-1442, 11-15 May 2003, Hsinchu, Taiwan.

[61]. Al-Shargabi .M.A and Abid .A, "The Impact of OBS Burst Aggregation on VBR Performance", in *proceeding of IEEE International Conference on Telecommunications and Comm.,* pp.303-306, 14-17 May 2007, Malaysia.

[62]. Rouskas .G and M.H, "Multi-destination communication over single-hop lightwave WDM Networks", *in proceeding of the IEEE Informatics and Communications Conference,* (INFOCOM), pp.1520-1527, June 1994, Toronto, Canada.

[63]. González-Ortega, Miguel .A, Merayo .N et.al, "The impact of delayed ACK in TCP flows in OBS networks", in *proceeding of European conference on Networks*, (NOC), pp. 367-374, January 2005, Innovation Strategy, Telefonica, Spain.

[64]. Liao .W and Loi .C.H, "Providing service differentiation for Optical Burst Switched networks", Journal of Lightwave Technology, vol. 22, pp. 1651–1660, July 2004.

[65]. Hernandez J.A, Aracil .J, Lopez .V and Palacios .J.F, "A resilience based comparative study between OBS and OCS technologies", *in proceedings of IEEE Conference on Transparent Optical Networks*, vol. 3, pp. 231-234, June 2006, Nottingham, England.

[66]. Orawiwattanakul,Ji .Y, Zhang .Y and Li .Y, "Fair Bandwidth Allocation in Optical Burst Switching Networks", *IEEE Journal of Light wave Technology,* vol. 27 , no. 16, pp. 3370 - 3380, 15th Aug 2009.

[67]. Phuritatkul .J, Ji .Y, and Yamada .S, “Proactive wavelength pre-emption for supporting absolute QoS in optical-burst-switched networks”, *Journal of Light wave Technology*, vol. 25, no. 5, pp. 1130–1137, May 2007.

[68]. Qiao .C, Jeong .M, Guha .A, Zhang .X and Wei .J, “WDM Multicasting In IP Over WDM Networks”, *in proceeding Of 7th International Conference On Networks Protocols*, pp. 89-96, November 1999, Toronto, Ontario, Canada.

[69]. González-Ortega, Miguel .A, Lopez-Ardao J.C, Lopez-Garcıa .C, Argibay-Losada .P, Rodrıguez-Rubio .R.F, and Pi˜neiro-Valladar .M, “Loss Differentiation in OBS without Wavelength Conversion”, *IEEE Comm. Letters*, Vol. 12, No. 12, pp.903-905, Dec.2008.

[70]. Tan .W, Mohan .G and Lui .J.C, “Achieving multi-class service differentiation in WDM Optical Burst Switching networks: A probabilistic preemptive burst segmentation scheme”, *IEEE Journal of Selected Areas in Communication*, vol. 24, no. 12, pp. 106–119, Dec.2006.

[71]. Gayathri .T, Venkadajothi .S, Kalaivani .S, C.Divya and C.Suresh .G.D., “Security Problems in the TCP/IP Protocol Suite”, *MASAUM Journal of Computing*, Vol 1, No. 2, Sep. 2009.

[72]. Yuhua Chen and Pramode K. Verma,” Secure Optical Burst Switching: Framework and Research Directions”, *IEEE Communication Magazine*, pp 40-45, August 2008.

[73]. Harrisa .B, Huntb .R, “TCP/IP security threats and attack methods”, *Elsevier Science Computer Communications vol.22*, pp 885–897. June 1999.

[74]. Stamatios V. Kartalopoulos, ”Optical Network Security: Counter measures in view of Channel attack”, *milcomp.p 1-5, MILCOM,* October – November 2006.

[75]. Medard .M, Marquis .D, Barry .R. A. and Finn .S. G, “Security Issues in All-Optical Networks”, *IEEE Network*, vol. 11, no. 4, pp. 109-130, May/June 2009.

[76]. Yuhua Cen, Pramode K. Varma and SubhashKak, “Embedded security framework for integrated classical and quantum cryptography services in optical burst switching network”, *Security Communications Networks,* 2009.

[77]. Rejeb .R, Pavlosoglou .I, Leeson .M.S, and Green .R.J., “Securing All-Optical Networks”, *ICTON 2003*, vol. 1, pp. 87-90, Warsaw, July 2003.

[78]. Médard .M, Marquis .D, and Chinn .S. R, “Attack Detection Methods for All-Optical Networks”, *Network and Distributed System Security Symposium, session 3, paper 2, San Diego,* March 11-13, 1998.

[79]. Al-Shargabi M.A, Ismail A.S and Idrus S.M, “A Feature Comparative study of Real-Time Traffic Simulation over OBS Network”, *International Journal of Computer Applications*, vol.21, no.4, pp. 13-16, May 2011.

[80]. Bodamer .S, Dolzer .K, Gauger .C and Barisch .M, “The IKR Simulation Library”, Institute of Communication Networks and Computer engineering, University of Stuttgart, www.ind.unistuttgart.de/IKRSimLib., June 2004.

[81]. Espina .F, Armendariz .J, García N et.al, “OBS network model for OMNeT++: A performance evaluation”, *in proceeding of International Conference on Science and Technology*, pp.18-25, March 2010, Torremolinos, Malaga, Spain.

[82]. Gurel .G, Alparslan .O and Karasan .E, “ nOBS: an ns2 based simulation tool for performance evaluation of TCP traffic in OBS networks, ” *Annals of Telecommunications,* vol. 62, no. 5-6, pp. 618-632, May 2007.

[83]. Pedrola .O, Rumley .S and Klinkowski .M, “Flexible Simulators for OBS Network Architectures”, *in proceeding of International Conference on Transparent Optical Networks,* (ICTON), pp. 117-122, June 2008, Athens, Greece.

[84]. Rodrigues J.P.C, Garcia N.M and Lorenz .P, “Object-oriented modelling and simulation of Optical Burst Switching Networks”, *in proceeding of IEEE Communication Society’s Global Communication Conference,* (GLOBECOM), pp 288-292, April 2004, Portugal.

[85]. Soares V.N.G.J, Veiga I.D.C and Rodrigues J.P.C, "OBS simulation tools: A comparative study", *in proceeding of International Conference on Communications Workshops*, (ICC Workshops' 08), pp. 256-260, 19-23 May 2008, Beijing, China.

[86]. Toledo .M.C.F and Zucchi .W.L, “Simulation of an Optical Burst Switch using Fiber Delay Lines”, *in proceeding of International Conference on Microwave and Optoelectronics,* pp. 334-340, 25-28 July 2005, Brasilia, Brazil.

[87]. GurayGurel, OnurAlparslan and EzhanKarasan,”nOBS: an ns2 based simulation tool for performance evaluation of TCP traffic in OBS networks”, *European Symposium on Simulation Tools for Research and education in Optical networks, Brest, France*, Oct.2005.

[88]. Wang .S.Y, Chou .C. L., Huang .C. H, Hwang .C. C, Yang .Z. M, Chiou .C.C, and Lin .C. C, “The design and Imple. Of the NCTUns 1.0 Network Simulator”.

[89]. Mohamed A. Alshargabi, Abdul Samed Ismail, Sevia M. Idrus,”A Feature Comparative Study of Real Time Traffic Simulation Over OBS Network,” *International .Journal of Computer Applications*, 2011.

[90]. Oscar .P, Miroslaw Klinkowski, Davide Careglio, and Josep .S.pareta,”JAVOBS: A Flexible Simulator for OBS Network Architectures”, *Journal of Networks,* vol. 5, no. 2, 2010.

[91]. Younglak Kim, Eunhyk Lim, Chul Kim, Kwangil Lee, Douglas Montgomery, Oliver Borchert, Richard Rouil, David Su,” GLASS – A Scalable Discreate Event Network Simulator for GMPLS based Optical Internet,”

[92] J. Xu, Chunming Qiao, Jikai Li, and Guang Xu, “Efficient Channel Scheduling Algorithms in Optical Burst Switching Networks,” In Proceeding of IEEE INFOCOM, vol. 3, pp. 2268-2278, 2003.

[93] Abdeliouab Belbekkouche, Abdelhakim Hafid, Michel Gendreau and Mariam Tagmouti, “Path –Based QoS Provisioning for Optical Burst Switching Networks”, Journal of Light wave technology, Vol. 29, No. 13, 2011.

[94] Yufeng Xin, Jing Teng, Gigi Karmous –Edwards, George N.Rouskas, Daniel Stevenson, “A Novel Fast Restoration Mechanism for Optical Burst Switched Networks”, IEEE 2011.

PUBLICATIONS

[1] K.Muthuraj and N.Sreenath, ”Optical Internet Security: A new Time based threat identification and its prevention,” International Journal of Research in Computer and Communication Technology, vol 2, no. 2, pp. 59 – 64, 2013.

[2] K.Muthuraj and N.Sreenath,”Optical Internet: A Comparative Assessment on TCP over OBS networks Simulation Tools,” International Journal of Advanced Computer Technology, vol 2, no. 1, pp. 01 – 09, 2013.

[3] K.Muthuraj and N.Sreenath,” Secure Optical Internet: An Attack on OBS node in a TCP over OBS network,” International Journal of Emerging Trends & Technology in Computer Science, vol 1, no. 4, pp. 75 – 80, 2012.

[4] K.Muthuraj and N.Sreenath, ”Optical Internet: Possible Attacks on TCP/OBS Networks,” International Journal of Computer Science and Information Security, vol 10, no. 12, pp. 20 – 25, 2012.

[5] K.Muthuraj and N.Sreenath,”Secure Optical Internet: A Novel Attack Prevention Mechanism for an OBS node in TCP/OBS Networks,” International Journal of Advanced Computer Science and Applications, vol 3, no. 12, pp. 76 – 80, 2012.

[6] N.Sreenath, K.Muthuraj and N.Ramkumar, ”Optical Internet: Securing TCP over OBS Networks from Security Vulnerabilities,” in Proc. International Conf. on Information, Computing and Telecommunications, India, Dec. 2012, pp. 25 – 30.

[7] N.Sreenath, K.Muthuraj and N.Ramkumar,” Secure Optical Internet: A Novel Threat Detection and its Countermeasures,” in Proc. International Conf. on Electrical, Electronics & Computer Science, India, Dec. 2012, pp. 57 – 62.

[8] N.Sreenath, K.Muthuraj and N.Ramkumar, ”Optical Internet: Analysis of Network Simulation Tools,” in Proc. International Conf. on Advanced Engineering and Technology, India, Dec. 2012, pp. 01 – 05.

[9] N.Sreenath, K.Muthuraj and G.Vinoth@Kuzhandaivelu, ”Threats and Vulnerabilities on TCP/OBS Networks,” in Proc. International Conf. on Computer Communication and Informatics, India, Jan. 2012, pp. 289 – 293.

[10] N.Sreenath, K.Muthuraj and G. Vinoth@Kuzhandaivelu, ”Optical Internet: A Survey on Security Issues in TCP/OBS Networks,” in Proc. International Conf. on Advanced Computing, Networking & Security, India, Dec. 2011, pp. 247 – 257.

[11] N.Sreenath, K.Muthuraj and P.Sivasubramanian, ”Secure Optical Internet: Attack Detection and Prevention Mechanism,” in Proc. IEEE International Conf. on Computing, Electronics and Electrical Technologies, India, Mar. 2012, pp. 135 – 140.

[12] N.Sreenath, K.Muthuraj and P.Sivasubramanian, ”Optical Internet: A Novel Attack Detection Mechanism for TCP/OBS Networks,” in Proc. IEEE International Conf. on Computational Intelligence and Computing Research, India, Dec. 2011, pp. 956 – 959.

[13] P.Sivasubramanian and K.Muthuraj, ”Threats in Optical Burst Switched Network,” International Journal Computer Technology Applications, vol 2, no. 3, pp. 510 – 514, 2011.

[14] N.Sreenath, K.Muthuraj and K.Brabagaran,” Security threats and countermeasures for optical burst switched networks”, International *Conf. on Recent trends in Engineering and Technology*, India, March 2014.

[15] N.Sreenath, K.Muthuraj and K.Brabagaran," Optical Internet: TCP/OBS security threats and attack methods with countermeasures", International *Conf. on Electronics & Communication Engineering*, India, April 2014.

Communicated Papers:

[16] K.Muthuraj and N.Sreenath, " Attack mitigation technique for optical burst switching in optical networks," *Journal of Optical Switching and Networking*.

[17] K.Muthuraj and N.Sreenath," Securing optical burst switched networks against threat s and vulnerabilities" *Journal of Photonic Network Communications*.

[18] K. Muthuraj and N.Sreenath ," Mitigating the security services for secure optical burst switched networks", *Journal of Security and Communication Networks*.

APPENDIX 1(A)

INSTALLATION PROCEDURE OF nOBS PATCH

Step 1 : Install alternate GCC on fedora 14

- Create directory named ***gcc*** under ***/home/user/*** directory.
- Download ***gcc-3.4.0.tar.bz2*** from http://ftp.gnu.org/gnu/gcc/gcc-3.4.0/gcc-3.4.0.tar.bz2
- Extract ***gcc-3.4.0.tar.bz2*** into ***/home/user/gcc*** directory**.**
- Create a directory named ***build*** at the location***/home/user/gcc/***
- Go to terminal and type ***cd /home/user/gcc/build/***
- Copy and paste the below commands in terminal:

```
/home/user/gcc/gcc-3.4.0/configure \
    --prefix=/opt/gcc34 \
    --program-suffix=34 \
    --enable-languages=c,c++ \
    --enable-shared --enable-threads=posix --disable-checking \
    --with-system-zlib --enable-__cxa_atexit --disable-libunwind-exceptions
```

- Now, type ***make*** to compile.
- To install this, login as root (Super User). So, type ***su***and enter the password.
- Type ***make install***
- Create a file named ***gcc34.sh*** in ***/home/user/*** and add the following lines

```
#!/bin/sh
GCC34_BIN=/opt/gcc34/bin
PATH=$GCC34_BIN:$PATH
export PATH
```

- Save and close the ***gcc34.sh*** file.

- Type ***cp /home/user/gcc34.sh /etc/profile.d/gcc34.sh.***

- Type ***ls -la /etc/profile.d/gcc34.sh*** and make sure to get an output in the terminal as ***-rw-r--r-- 1 root root 66 Jun 15 21:38 /etc/profile.d/gcc34.sh***

- Type ***chmod 755 /etc/profile.d/gcc34.sh***.

- Type ***ls -la /etc/profile.d/gcc34.sh*** and make sure to get an output in the terminal as shown below

```
-rwxr-xr-x  1 root root 66 Jun 15 21:38 /etc/profile.d/gcc34.sh
```

- Type ***exit***and thus you logout super-user.

- Again, type***exit*** to logout terminal.

- To verify successful installation of gcc, open a new terminal and type ***which gcc34*** and make sure to get as output ***/opt/gcc34/bin/gcc34***

- Type ***which g++34*** and make sure to get ***/opt/gcc34/bin/g++34***

- Type ***gcc34 –v*** and make sure to get

```
Reading specs from /opt/gcc34/lib/gcc/i686-pc-linux-gnu/3.4.4/specs
Configured with: /home/user/gcc/gcc-3.4.4/configure --prefix=/opt/gcc34
--program-suffix=34 --enable-languages=c,c++ --enable-shared --enable-threads=posix
--disable-checking --with-system-zlib --enable-__cxa_atexit
--disable-libunwind-exceptions
Thread model: posix
gcc version 3.4.4
```

- Go to the ***bashrc*** file located at ***etc/bashrc*** and add these lines (as super-user) *(Refer fig 7 for bashrc file)*

```
          export CC=gcc34
          export CXX=g++34
  ./configure
```

- NOTE :For more information on installing gcc on fedora visit the link ***http://www.mjmwired.net/resources/mjm-fedora-gcc.html***

Step 2: Install ns2.27 after changing *GNU Compiler Collection* (GCC)

- Get a copy of ***ns-allinone-2.27.tar.gz***from ftp://ftp.isi.edu/nsnam/ns-allinone-2.27.tar.gz
- Extract ***ns-allinone-2.27.tar.gz*** *into* ***/home/user*** directory
- Go to the terminal and type ***cd ns-allinone-2.27.***
- Type ***./install***to install the ns-software
- Some errors are shown during installation *(Refer Fig 1 from Appendix 1(B)).*
- To solve the errors , download the attached files ***tclcl.h, agent.h*** and ***tkBind.c*** from***ns2_27_gcc34_compilation_errors.rar****(given below)* into ***/home/user/***
- Open terminal and type ***cp /home/user/agent.h /home/user/ns-allinone-2.27/nam-1.10/agent.h*** (This will solve Nam error. Refer fig 10 from *Appendix 1(B)*for error msg)
- Type ***cp /home/user/tclcl.h /home/user/ns-allinone-2.27/tclcl-1.15/tclcl.h***
- Type ***cp /home/user/tkBind.c /home/user/ns-allinone-2.27/tk8.4.5/generic/tkBind.c***
- Download patch ***ns-2.27-gcc34.patch*** (*ns-2.27-gcc34.rar* given below) into ***/home/user*** location.
- Type ***cd /home/user***
- Type ***patch -p0 < ns-2.27-gcc34.patch****. (Refer Fig 2 from Appendix 1(B))..*
- Type ***cd /home/user/ns-allinone-2.27.***
- Again, type ***./install***to install the ns-software *(Refer Fig 3 from Appendix 1(B))..*
- Close terminal and other windows. Logout and login fedora.
- Once installation is successful the environment variables need to be set by editing ***bashrc*** file as super-user (su).

Step 3 : Edit read-only *bashrc* file

- Go to terminal and type ***su***and enter the password
- Type ***cd /etc***in the terminal.
- Now type ***vi bashrc*** and the file is open within terminal
- Edit the file (within terminal) using ***INSERT***key and type the below lines in the bashrc file(After the comment lines and before program start. *(Refer Fig 4 from Appendix 1(B))..*

```
export PATH=$PATH:"/home/user/ns-allinone-2.27/bin:/home/user/ns-allinone-2.27/tcl8.4.5/unix:/home/user/ns-allinone-2.27/tk8.4.5/unix"
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:"/home/user/ns-allinone-2.27/otcl-1.8:/home/user/ns-allinone-2.27/lib"
export TCL_LIBRARY=$TCL_LIBRARY:"/home/user/ns-allinone-2.27/tcl8.4.5/library"
```

- After editing, hit ***ESC***key.
- Type ***:wq*** in the terminal and thus you save the made changes.
- Type ***exit***and thus you logout super-user.Again, type ***exit***to logout terminal.

Step 4 : Install OBS patch for ns 2.27

- Download ***obs patch*** from the source-link ***http://www.anarg.jp/personal/a-onur/optical-burst-switching(obs)-simulator.zip***
- Extract the patch *to* ***/home/user/*** directory.
- Copy the ***optical*** directory (found in the above patch) to ***home/user/ns-allinone-2.27/ns-2.27 location.***
- Copy the files under ***routing*** directory (found in the above patch) and replace or overwrite the files in the location /***home/user/ns-allinone-2.27/ns-2.27/routing/***.
- Copy the files under ***tcl*** directory (found in the above patch) and replace or overwrite the files in the location ***/home/user/ns-allinone-2.27/ns-2.27/tcl/***.
- Copy the files under ***queue*** directory (found in the above patch) and replace or overwrite the files in the location ***/home/user/ns-allinone-2.27/ns-2.27/queue/.***

- Copy the files under ***tcp*** directory (found in the above patch) and replace or overwrite the files in the location ***/home/user/ns-allinone-2.27/ns-2.27/tcp/.***

- Copy the files under ***common*** directory (found in the above patch) and replace or overwrite the files in the location ***/home/user/ns-allinone-2.27/ns-2.27/common/.***

- Add the following code to end of ***OBJ_CC*** in ***Makefile*** file from the location ***/home/user/ns-allinone-2.27/ns-2.27/*** *(Refer Fig 8 from Appendix 1(B)).*

```
optical/op-delay.o \
optical/op-queue.o \
        optical/op-burst_agent.o \
        optical/op-classifier.o \
        optical/op-classifier-hash.o \
        optical/op-classifier-sr.o \
        optical/op-sragent.o \
        optical/op-queue2.o \
        optical/op-schedule.o \
        optical/op-converterschedule.o\
        optical/op-fdlschedule.o\
```

- Find ***autoconf.h*** file from the location ***/home/user/ns-allinone-2.27/ns-2.27/***and openin a text editor and add this line *(Refer Fig 9 fromAppendix 1(B)).*

```
        #define CPP_NAMESPACE std
```

- At terminal, type ***cd /home/user/ns-allinone-2.27/ns-2.27***

- Now compile by typing ***make clean.*** Then type ***make***.

- Now type ***ns***and confirm **%** symbol for a successful installation.

Step 5 : Run a test file in nOBS patch

- Copy the file ***sample_obs.tcl*** to ***/home/user/*** directory

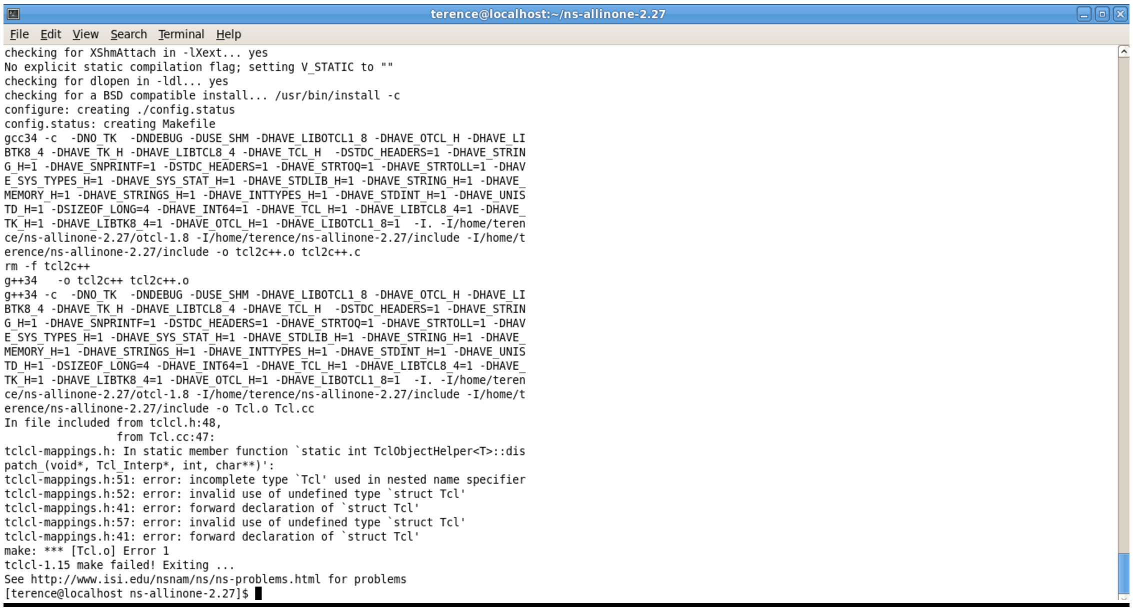
- Open terminal and type ***cd /home/user/***

- Type ***ns sample_obs.tcl*** to compile the file

- Now type ***nam out.nam*** to get the NAM output. Refer fig 5,6,7 in *Appendix 1(B)* for sample NAM output.

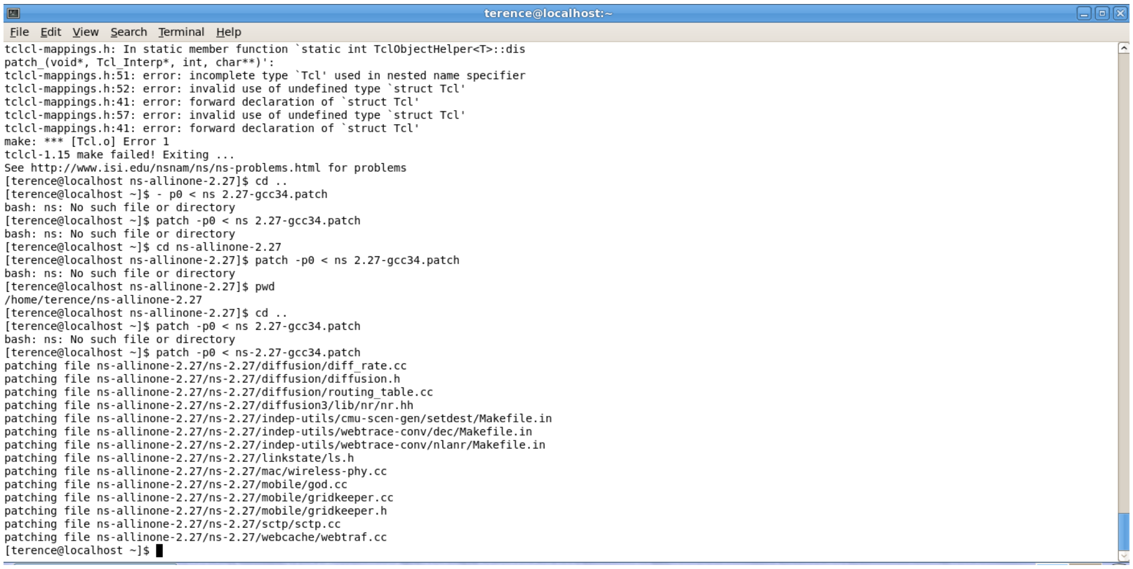
APPENDIX 1(B)

SCREENSHOTS FOR INSTALLATION PROCEDURE OF NOBS PATCH


terence@localhost:~/ns-allinone-2.27
File Edit View Search Terminal Help

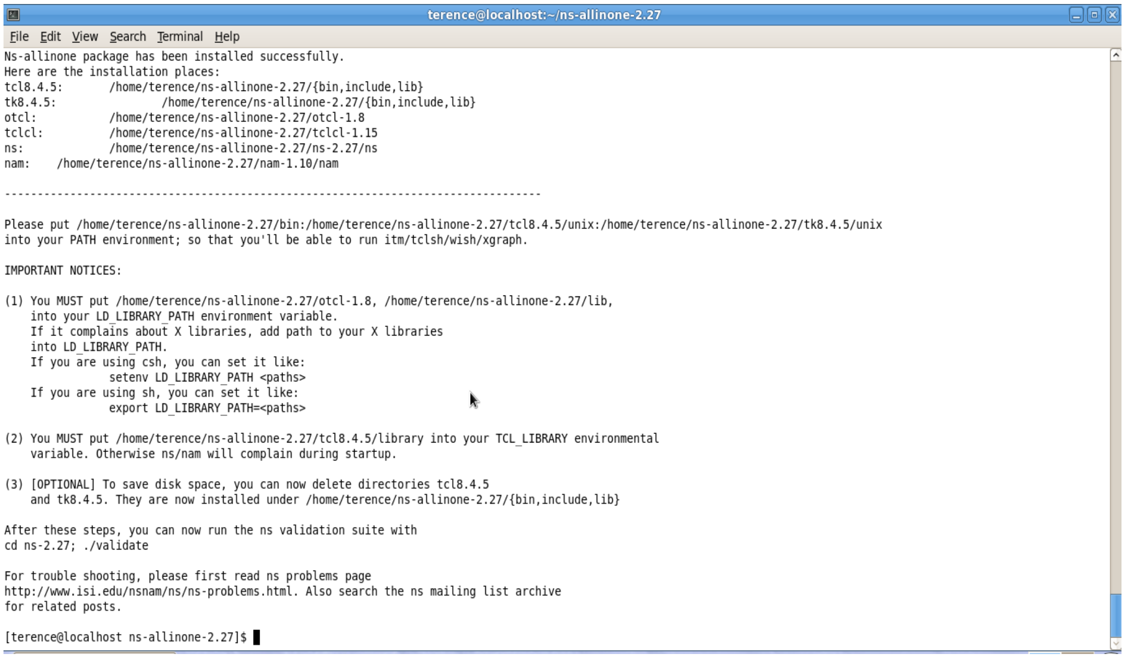
```
checking for XShmAttach in -lXext... yes
No explicit static compilation flag; setting V_STATIC to ""
checking for dlopen in -ldl... yes
checking for a BSD compatible install... /usr/bin/install -c
configure: creating ./config.status
config.status: creating Makefile
gcc34 -c  -DNO_TK  -DNDEBUG -DUSE_SHM -DHAVE_LIBOTCL1_8 -DHAVE_OTCL_H -DHAVE_LI
BTK8_4 -DHAVE_TK_H -DHAVE_LIBTCL8_4 -DHAVE_TCL_H  -DSTDC_HEADERS=1 -DHAVE_STRIN
G_H=1 -DHAVE_SNPRINTF=1 -DSTDC_HEADERS=1 -DHAVE_STRTOQ=1 -DHAVE_STRTOLL=1 -DHAV
E_SYS_TYPES_H=1 -DHAVE_SYS_STAT_H=1 -DHAVE_STDLIB_H=1 -DHAVE_STRING_H=1 -DHAVE_
MEMORY_H=1 -DHAVE_STRINGS_H=1 -DHAVE_INTTYPES_H=1 -DHAVE_STDINT_H=1 -DHAVE_UNIS
TD_H=1 -DSIZEOF_LONG=4 -DHAVE_INT64=1 -DHAVE_TCL_H=1 -DHAVE_LIBTCL8_4=1 -DHAVE_
TK_H=1 -DHAVE_LIBTK8_4=1 -DHAVE_OTCL_H=1 -DHAVE_LIBOTCL1_8=1  -I. -I/home/teren
ce/ns-allinone-2.27/otcl-1.8 -I/home/terence/ns-allinone-2.27/include -I/home/t
erence/ns-allinone-2.27/include -o tcl2c++.o tcl2c++.c
rm -f tcl2c++
g++34   -o tcl2c++ tcl2c++.o
g++34 -c  -DNO_TK  -DNDEBUG -DUSE_SHM -DHAVE_LIBOTCL1_8 -DHAVE_OTCL_H -DHAVE_LI
BTK8_4 -DHAVE_TK_H -DHAVE_LIBTCL8_4 -DHAVE_TCL_H  -DSTDC_HEADERS=1 -DHAVE_STRIN
G_H=1 -DHAVE_SNPRINTF=1 -DSTDC_HEADERS=1 -DHAVE_STRTOQ=1 -DHAVE_STRTOLL=1 -DHAV
E_SYS_TYPES_H=1 -DHAVE_SYS_STAT_H=1 -DHAVE_STDLIB_H=1 -DHAVE_STRING_H=1 -DHAVE_
MEMORY_H=1 -DHAVE_STRINGS_H=1 -DHAVE_INTTYPES_H=1 -DHAVE_STDINT_H=1 -DHAVE_UNIS
TD_H=1 -DSIZEOF_LONG=4 -DHAVE_INT64=1 -DHAVE_TCL_H=1 -DHAVE_LIBTCL8_4=1 -DHAVE_
TK_H=1 -DHAVE_LIBTK8_4=1 -DHAVE_OTCL_H=1 -DHAVE_LIBOTCL1_8=1  -I. -I/home/teren
ce/ns-allinone-2.27/otcl-1.8 -I/home/terence/ns-allinone-2.27/include -I/home/t
erence/ns-allinone-2.27/include -o Tcl.o Tcl.cc
In file included from tclcl.h:48,
                 from Tcl.cc:47:
tclcl-mappings.h: In static member function `static int TclObjectHelper<T>::dis
patch_(void*, Tcl_Interp*, int, char**)':
tclcl-mappings.h:51: error: incomplete type `Tcl' used in nested name specifier
tclcl-mappings.h:52: error: invalid use of undefined type `struct Tcl'
tclcl-mappings.h:41: error: forward declaration of `struct Tcl'
tclcl-mappings.h:57: error: invalid use of undefined type `struct Tcl'
tclcl-mappings.h:41: error: forward declaration of `struct Tcl'
make: *** [Tcl.o] Error 1
tclcl-1.15 make failed! Exiting ...
See http://www.isi.edu/nsnam/ns/ns-problems.html for problems
[terence@localhost ns-allinone-2.27]$ █
```

Figure 1 List of errors in ns2 installation



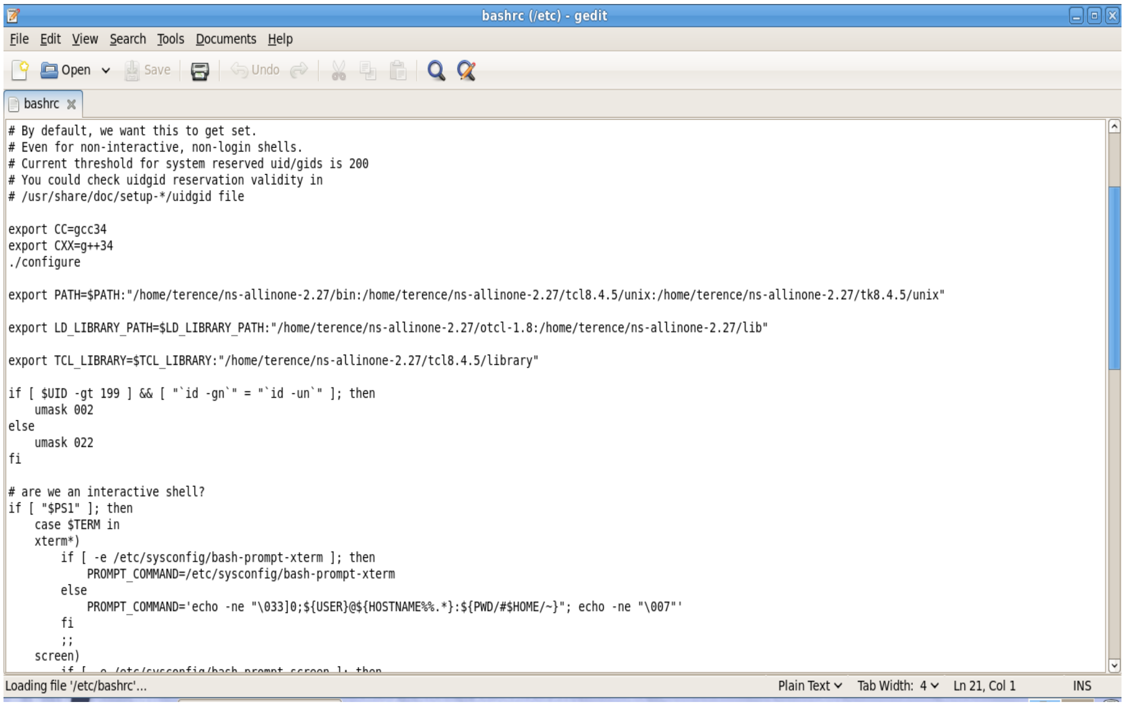
```
tclcl-mappings.h: In static member function `static int TclObjectHelper<T>::dis
patch_(void*, Tcl_Interp*, int, char**)':
tclcl-mappings.h:51: error: incomplete type `Tcl' used in nested name specifier
tclcl-mappings.h:52: error: invalid use of undefined type `struct Tcl'
tclcl-mappings.h:41: error: forward declaration of `struct Tcl'
tclcl-mappings.h:57: error: invalid use of undefined type `struct Tcl'
tclcl-mappings.h:41: error: forward declaration of `struct Tcl'
make: *** [Tcl.o] Error 1
tclcl-1.15 make failed! Exiting ...
See http://www.isi.edu/nsnam/ns/ns-problems.html for problems
[terence@localhost ns-allinone-2.27]$ cd ..
[terence@localhost ~]$ - p0 < ns 2.27-gcc34.patch
bash: ns: No such file or directory
[terence@localhost ~]$ patch -p0 < ns 2.27-gcc34.patch
bash: ns: No such file or directory
[terence@localhost ~]$ cd ns-allinone-2.27
[terence@localhost ns-allinone-2.27]$ patch -p0 < ns 2.27-gcc34.patch
bash: ns: No such file or directory
[terence@localhost ns-allinone-2.27]$ pwd
/home/terence/ns-allinone-2.27
[terence@localhost ns-allinone-2.27]$ cd ..
[terence@localhost ~]$ patch -p0 < ns 2.27-gcc34.patch
bash: ns: No such file or directory
[terence@localhost ~]$ patch -p0 < ns-2.27-gcc34.patch
patching file ns-allinone-2.27/ns-2.27/diffusion/diff_rate.cc
patching file ns-allinone-2.27/ns-2.27/diffusion/diffusion.h
patching file ns-allinone-2.27/ns-2.27/diffusion/routing_table.cc
patching file ns-allinone-2.27/ns-2.27/diffusion3/lib/nr/nr.hh
patching file ns-allinone-2.27/ns-2.27/indep-utils/cmu-scen-gen/setdest/Makefile.in
patching file ns-allinone-2.27/ns-2.27/indep-utils/webtrace-conv/dec/Makefile.in
patching file ns-allinone-2.27/ns-2.27/indep-utils/webtrace-conv/nlanr/Makefile.in
patching file ns-allinone-2.27/ns-2.27/linkstate/ls.h
patching file ns-allinone-2.27/ns-2.27/mac/wireless-phy.cc
patching file ns-allinone-2.27/ns-2.27/mobile/god.cc
patching file ns-allinone-2.27/ns-2.27/mobile/gridkeeper.cc
patching file ns-allinone-2.27/ns-2.27/mobile/gridkeeper.h
patching file ns-allinone-2.27/ns-2.27/sctp/sctp.cc
patching file ns-allinone-2.27/ns-2.27/webcache/webtraf.cc
[terence@localhost ~]$ █
```

Figure 2 Messages shown after patching gcc34 with ns-2.27


terence@localhost:~/ns-allinone-2.27
File Edit View Search Terminal Help

```
Ns-allinone package has been installed successfully.
Here are the installation places:
tcl8.4.5:       /home/terence/ns-allinone-2.27/{bin,include,lib}
tk8.4.5:                /home/terence/ns-allinone-2.27/{bin,include,lib}
otcl:           /home/terence/ns-allinone-2.27/otcl-1.8
tclcl:          /home/terence/ns-allinone-2.27/tclcl-1.15
ns:             /home/terence/ns-allinone-2.27/ns-2.27/ns
nam:    /home/terence/ns-allinone-2.27/nam-1.10/nam

----------------------------------------------------------------------------------

Please put /home/terence/ns-allinone-2.27/bin:/home/terence/ns-allinone-2.27/tcl8.4.5/unix:/home/terence/ns-allinone-2.27/tk8.4.5/unix
into your PATH environment; so that you'll be able to run itm/tclsh/wish/xgraph.

IMPORTANT NOTICES:

(1) You MUST put /home/terence/ns-allinone-2.27/otcl-1.8, /home/terence/ns-allinone-2.27/lib,
    into your LD_LIBRARY_PATH environment variable.
    If it complains about X libraries, add path to your X libraries
    into LD_LIBRARY_PATH.
    If you are using csh, you can set it like:
                setenv LD_LIBRARY_PATH <paths>
    If you are using sh, you can set it like:
                export LD_LIBRARY_PATH=<paths>

(2) You MUST put /home/terence/ns-allinone-2.27/tcl8.4.5/library into your TCL_LIBRARY environmental
    variable. Otherwise ns/nam will complain during startup.

(3) [OPTIONAL] To save disk space, you can now delete directories tcl8.4.5
    and tk8.4.5. They are now installed under /home/terence/ns-allinone-2.27/{bin,include,lib}

After these steps, you can now run the ns validation suite with
cd ns-2.27; ./validate

For trouble shooting, please first read ns problems page
http://www.isi.edu/nsnam/ns/ns-problems.html. Also search the ns mailing list archive
for related posts.

[terence@localhost ns-allinone-2.27]$ █
```

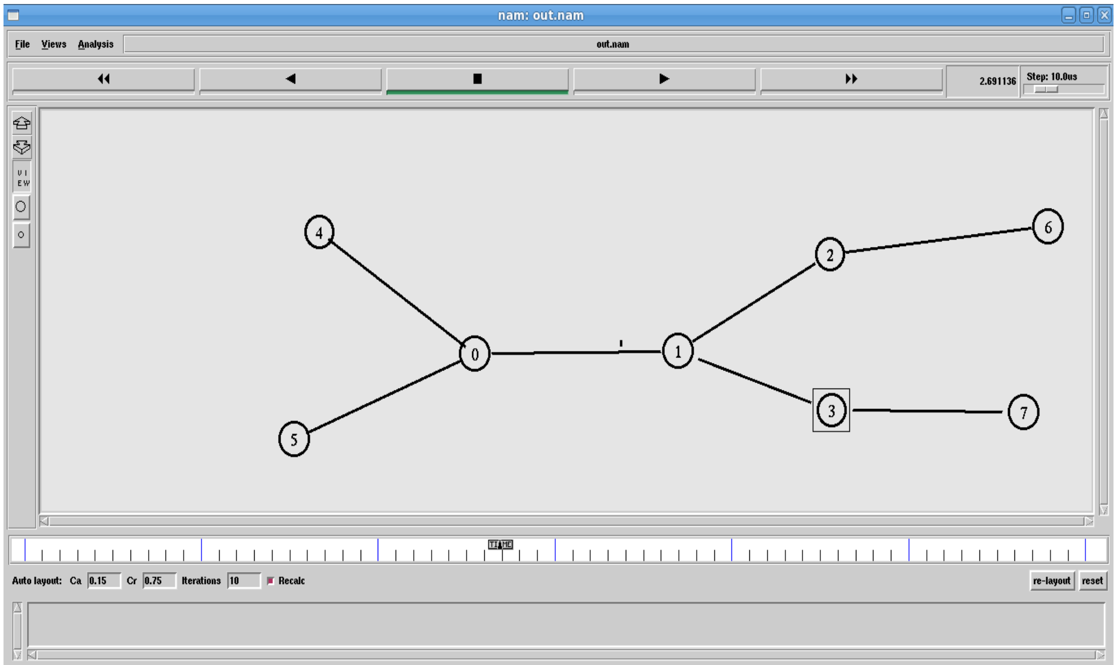
Figure 3 Successful ns2 installation



```
# By default, we want this to get set.
# Even for non-interactive, non-login shells.
# Current threshold for system reserved uid/gids is 200
# You could check uidgid reservation validity in
# /usr/share/doc/setup-*/uidgid file

export CC=gcc34
export CXX=g++34
./configure

export PATH=$PATH:"/home/terence/ns-allinone-2.27/bin:/home/terence/ns-allinone-2.27/tcl8.4.5/unix:/home/terence/ns-allinone-2.27/tk8.4.5/unix"

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:"/home/terence/ns-allinone-2.27/otcl-1.8:/home/terence/ns-allinone-2.27/lib"

export TCL_LIBRARY=$TCL_LIBRARY:"/home/terence/ns-allinone-2.27/tcl8.4.5/library"

if [ $UID -gt 199 ] && [ "`id -gn`" = "`id -un`" ]; then
    umask 002
else
    umask 022
fi

# are we an interactive shell?
if [ "$PS1" ]; then
    case $TERM in
    xterm*)
        if [ -e /etc/sysconfig/bash-prompt-xterm ]; then
            PROMPT_COMMAND=/etc/sysconfig/bash-prompt-xterm
        else
            PROMPT_COMMAND='echo -ne "\033]0;${USER}@${HOSTNAME%%.*}:${PWD/#$HOME/~}"; echo -ne "\007"'
        fi
        ;;
    screen)
        if [ -e /etc/sysconfig/bash-prompt-screen ]; then
```

Loading file '/etc/bashrc'...

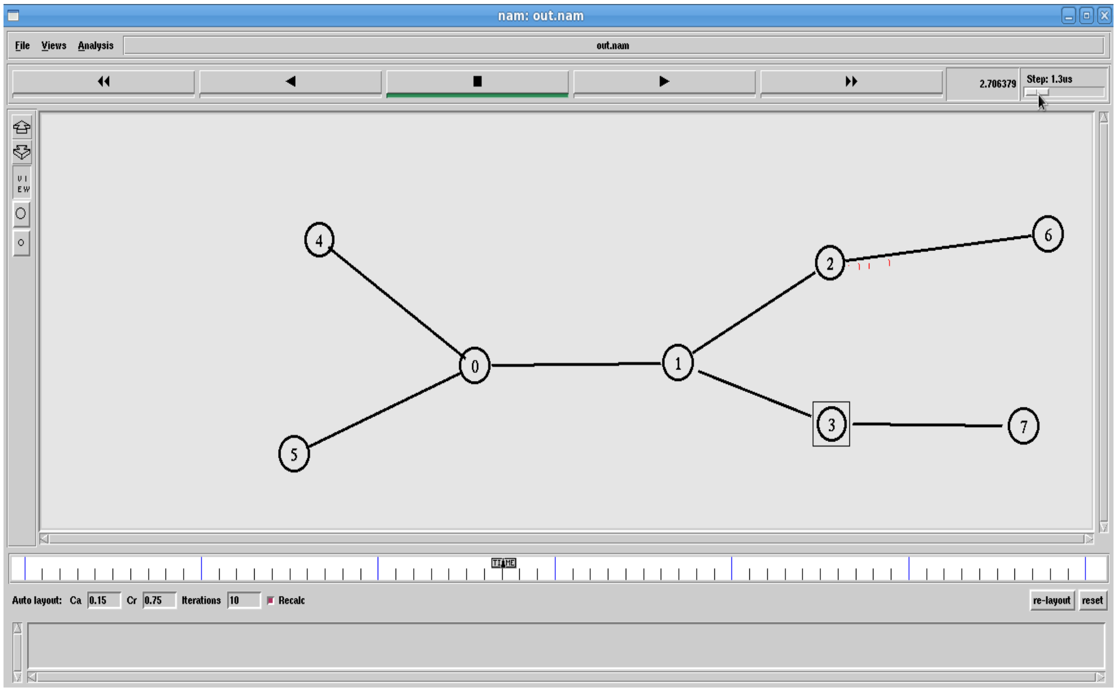
Figure 4 bashrc file after editing environment variables



2.691136 Step: 10.0us

Auto layout: Ca 0.15 Cr 0.75 Iterations 10 Recalc

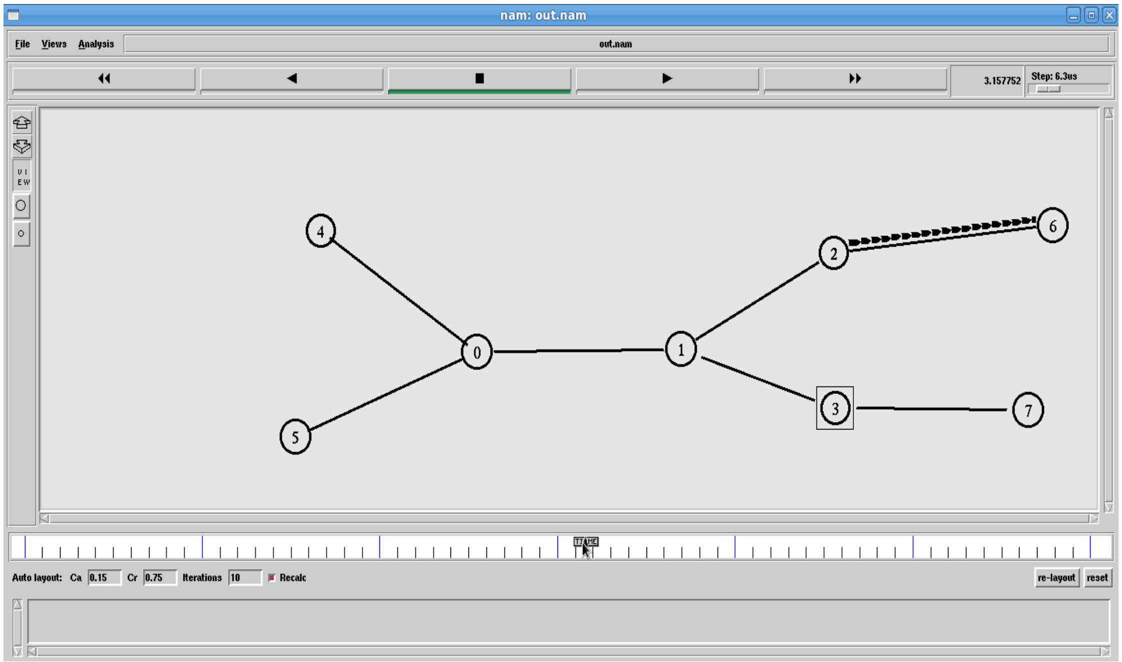
Figure 5 NAM Output 1



2.706379 Step: 1.3us

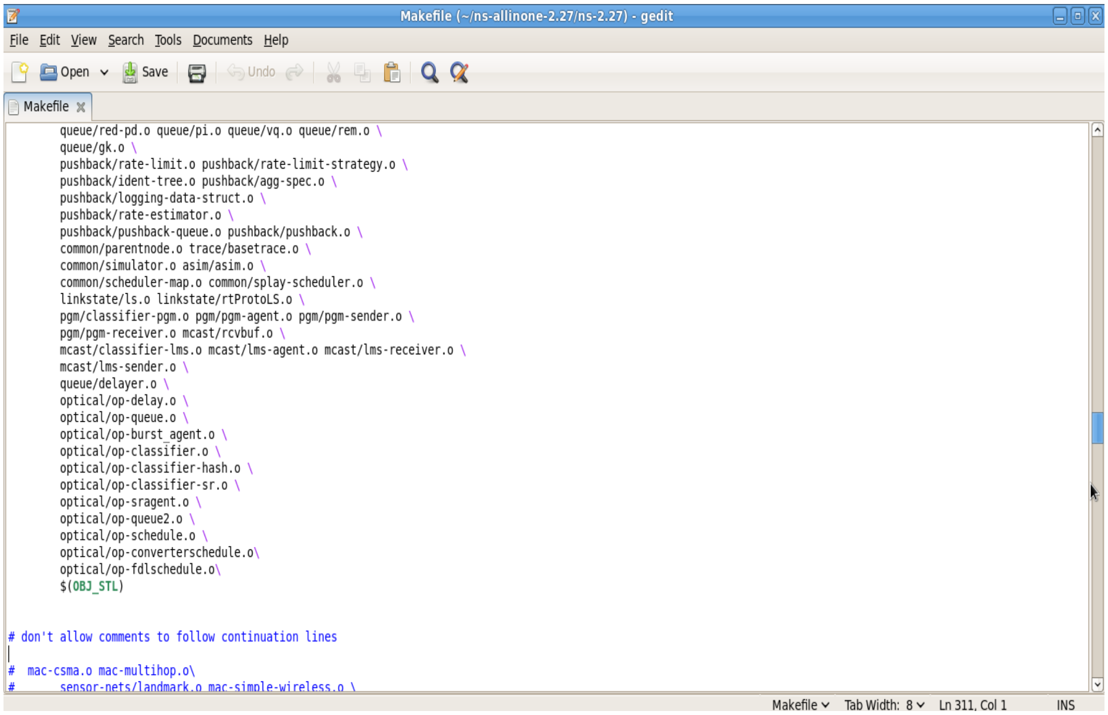
Auto layout: Ca 0.15 Cr 0.75 Iterations 10 Recalc

Figure 6 NAM Output 2



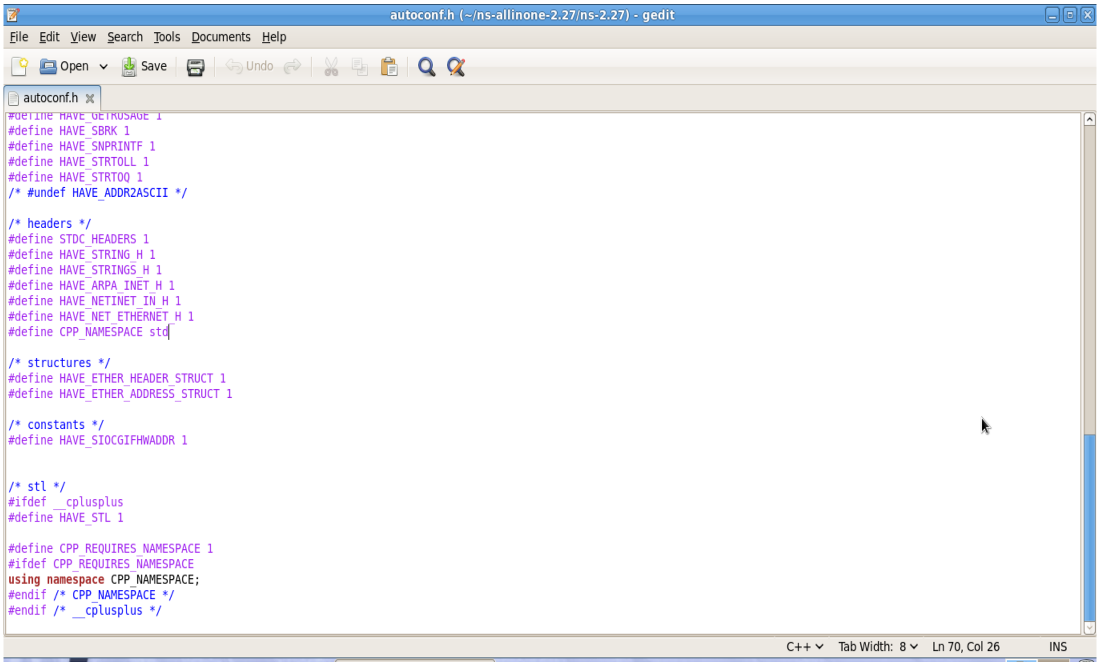
Auto layout: Ca 0.15 Cr 0.75 Iterations 10 Recalc

Figure 7 NAM Output 3



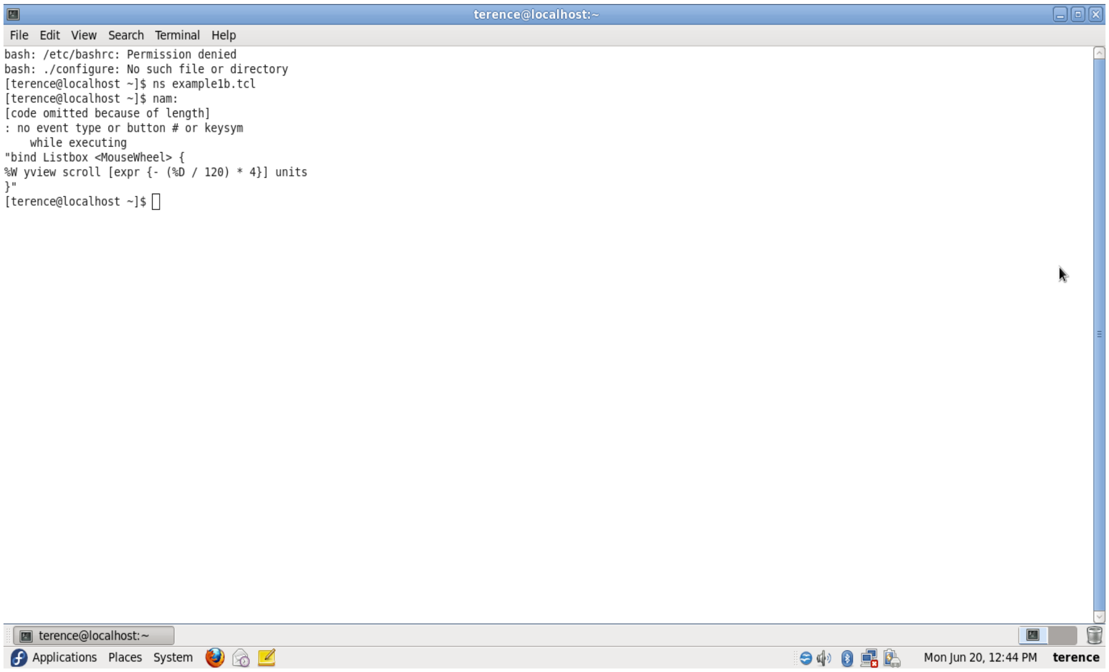
```
        queue/red-pd.o queue/pi.o queue/vq.o queue/rem.o \
        queue/gk.o \
        pushback/rate-limit.o pushback/rate-limit-strategy.o \
        pushback/ident-tree.o pushback/agg-spec.o \
        pushback/logging-data-struct.o \
        pushback/rate-estimator.o \
        pushback/pushback-queue.o pushback/pushback.o \
        common/parentnode.o trace/basetrace.o \
        common/simulator.o asim/asim.o \
        common/scheduler-map.o common/splay-scheduler.o \
        linkstate/ls.o linkstate/rtProtoLS.o \
        pgm/classifier-pgm.o pgm/pgm-agent.o pgm/pgm-sender.o \
        pgm/pgm-receiver.o mcast/rcvbuf.o \
        mcast/classifier-lms.o mcast/lms-agent.o mcast/lms-receiver.o \
        mcast/lms-sender.o \
        queue/delayer.o \
        optical/op-delay.o \
        optical/op-queue.o \
        optical/op-burst_agent.o \
        optical/op-classifier.o \
        optical/op-classifier-hash.o \
        optical/op-classifier-sr.o \
        optical/op-sragent.o \
        optical/op-queue2.o \
        optical/op-schedule.o \
        optical/op-converterschedule.o\
        optical/op-fdlschedule.o\
        $(OBJ_STL)


# don't allow comments to follow continuation lines

#  mac-csma.o mac-multihop.o\
#       sensor-nets/landmark.o mac-simple-wireless.o \
```

Figure 8 makefile


autoconf.h (~/ns-allinone-2.27/ns-2.27) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
autoconf.h
C++
Tab Width: 8
Ln 70, Col 26
INS

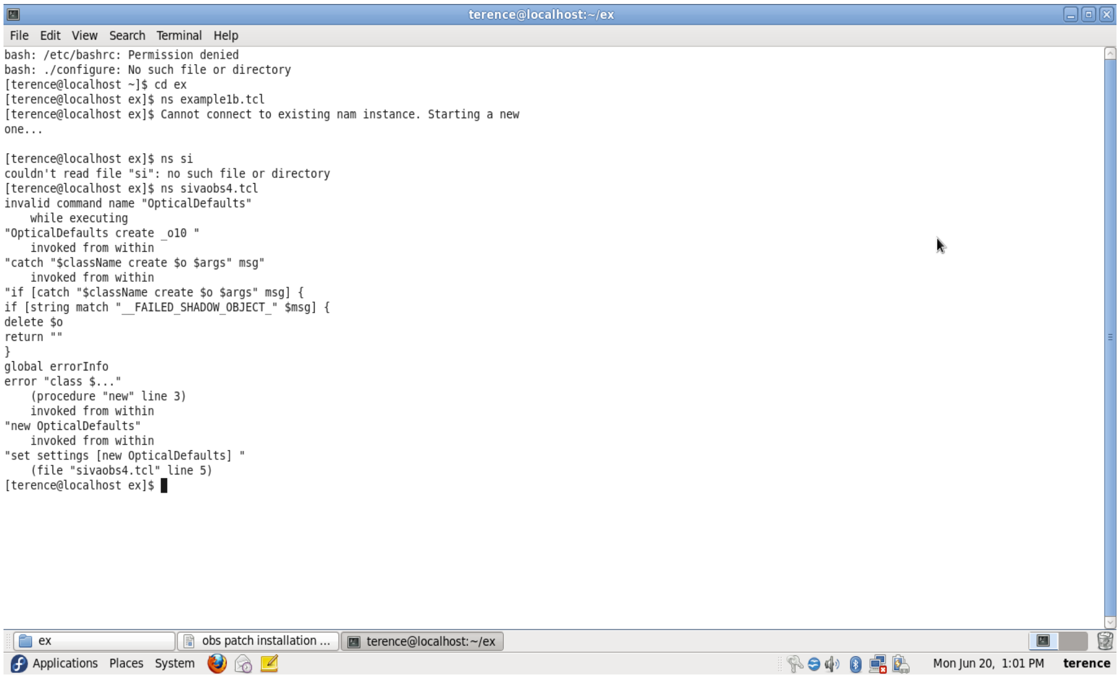
```
#define HAVE_GETRUSAGE 1
#define HAVE_SBRK 1
#define HAVE_SNPRINTF 1
#define HAVE_STRTOLL 1
#define HAVE_STRTOQ 1
/* #undef HAVE_ADDR2ASCII */

/* headers */
#define STDC_HEADERS 1
#define HAVE_STRING_H 1
#define HAVE_STRINGS_H 1
#define HAVE_ARPA_INET_H 1
#define HAVE_NETINET_IN_H 1
#define HAVE_NET_ETHERNET_H 1
#define CPP_NAMESPACE std

/* structures */
#define HAVE_ETHER_HEADER_STRUCT 1
#define HAVE_ETHER_ADDRESS_STRUCT 1

/* constants */
#define HAVE_SIOCGIFHWADDR 1


/* stl */
#ifdef __cplusplus
#define HAVE_STL 1

#define CPP_REQUIRES_NAMESPACE 1
#ifdef CPP_REQUIRES_NAMESPACE
using namespace CPP_NAMESPACE;
#endif /* CPP_NAMESPACE */
#endif /* __cplusplus */
```

Figure 9 autoconf.h



```
bash: /etc/bashrc: Permission denied
bash: ./configure: No such file or directory
[terence@localhost ~]$ ns example1b.tcl
[terence@localhost ~]$ nam:
[code omitted because of length]
: no event type or button # or keysym
    while executing
"bind Listbox <MouseWheel> {
%W yview scroll [expr {- (%D / 120) * 4}] units
}"
[terence@localhost ~]$ 
```

Figure 10 NAM error (without tkBind.c file)


terence@localhost:~/ex
File Edit View Search Terminal Help
ex
obs patch installation ...
terence@localhost:~/ex
Applications Places System
Mon Jun 20, 1:01 PM
terence

```
bash: /etc/bashrc: Permission denied
bash: ./configure: No such file or directory
[terence@localhost ~]$ cd ex
[terence@localhost ex]$ ns example1b.tcl
[terence@localhost ex]$ Cannot connect to existing nam instance. Starting a new
one...

[terence@localhost ex]$ ns si
couldn't read file "si": no such file or directory
[terence@localhost ex]$ ns sivaobs4.tcl
invalid command name "OpticalDefaults"
    while executing
"OpticalDefaults create _o10 "
    invoked from within
"catch "$className create $o $args" msg"
    invoked from within
"if [catch "$className create $o $args" msg] {
if [string match "__FAILED_SHADOW_OBJECT_" $msg] {
delete $o
return ""
}
global errorInfo
error "class $..."
    (procedure "new" line 3)
    invoked from within
"new OpticalDefaults"
    invoked from within
"set settings [new OpticalDefaults] "
    (file "sivaobs4.tcl" line 5)
[terence@localhost ex]$ █
```

Figure 11 Error if an OBs simulation compiled without nOBS patch

APPENDIX 2(A)

SCRIPT IN TCL TO MODEL THE OBS NETWORK

A TCL file to configure TCP over OBS network with electronic and optical nodes is given in this Appendix. The 14 node NSFNet (Network Science Foundation Net) topology configuration is used to demonstrate the effect of burst dropping in TCP over OBS network as shown in figure below. Nodes 0 to 13 represent the optical nodes and 14 to 41 represent the electronic nodes. The optical nodes are modeled with 1Gbps bandwidth and 10milliseconds propagation delay. The TCP/IP links have 155Mbps bandwidth each with 1millisecond link propagation delay.

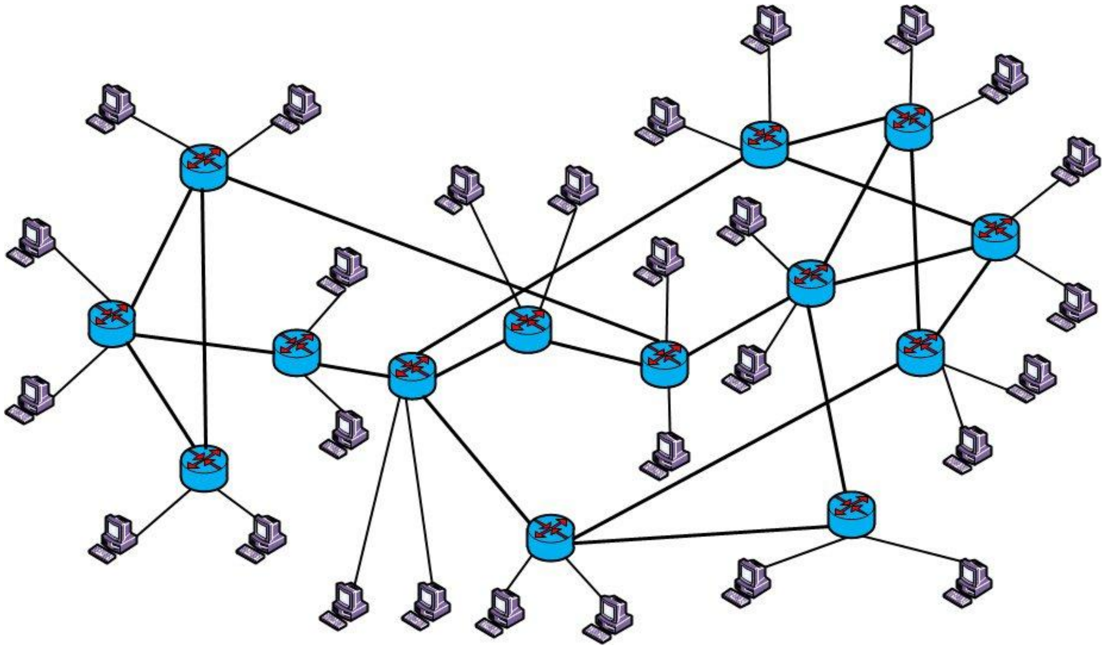


Figure 1 NSF net 14 Optical node and 28 Electronic nodes

```
proc my-duplex-link {ns n1 n2 bw delay queue_method queue_length}

{

        $ns optical-duplex-link $n1 $n2 $bw $delay $queue_method
        $ns queue-limit $n1 $n2 $queue_length
        $ns queue-limit $n2 $n1 $queue_length
}
proc my-duplex-link2 {ns n1 n2 bw delay queue_method queue_length}
{
        $ns optical-simplex-link $n1 $n2 $bw $delay $queue_method
        $ns simplex-link $n2 $n1 $bw $delay DropTail
        $ns queue-limit $n1 $n2 $queue_length
        $ns queue-limit $n2 $n1 $queue_length
}
#Create 14 optical nodes
for {set i 0} {$i < 14} {incr i}
{
        set n($i) [$ns OpNode]
        set temp [$n($i) set src_agent_]
        $temp optic_nodes 0 1 2 3 4 5 6 7 8 9 10 11 12 13
        $temp set nodetype_ 0
        $temp set conversiontype_ 1
        $temp create
#Whether acks are burstified
        $temp set ackdontburst 1
        set temp [$n($i) set burst_agent_]
        $temp optic_nodes 0 1 2 3 4 5 6 7 8 9 10 11 12 13
#Whether acks are burstified
```

```
        $temp set ackdontburst 1

        set temp [$n($i) set classifier_]

        $temp optic_nodes 0 1 2 3 4 5 6 7 8 9 10 11 12 13

}

#Create 20 electronic nodes

        for {set i 14} {$i < 34} {incr i} {

set n($i) [$ns node]

# Define optical nodes

        set temp [$n($i) set src_agent_]

        $temp optic_nodes 0 1 2 3 4 5 6 7 8 9 10 11 12 13

        set temp [$n($i) set classifier_]

        $temp optic_nodes 0 1 2 3 4 5 6 7 8 9 10 11 12 13

}

set queue_length 100000

#Create links between the nodes

my-duplex-link2 $ns $n(0) $n(1) 1000Mb 10ms OpQueue $queue_length

my-duplex-link2 $ns $n(0) $n(2) 1000Mb 10ms OpQueue $queue_length

my-duplex-link2 $ns $n(0) $n(3) 1000Mb 10ms OpQueue $queue_length

my-duplex-link2 $ns $n(1) $n(2) 1000Mb 10ms OpQueue $queue_length

my-duplex-link2 $ns $n(1) $n(7) 1000Mb 10ms OpQueue $queue_length

my-duplex-link2 $ns $n(2) $n(5) 1000Mb 10ms OpQueue $queue_length

my-duplex-link2 $ns $n(3) $n(10) 1000Mb 10ms OpQueue $queue_length

my-duplex-link2 $ns $n(3) $n(4) 1000Mb 10ms OpQueue $queue_length

my-duplex-link2 $ns $n(4) $n(5) 1000Mb 10ms OpQueue $queue_length
```

```
my-duplex-link2 $ns $n(4) $n(6) 1000Mb 10ms OpQueue $queue_length
my-duplex-link2 $ns $n(5) $n(12) 1000Mb 10ms OpQueue $queue_length
my-duplex-link2 $ns $n(5) $n(9) 1000Mb 10ms OpQueue $queue_length
my-duplex-link2 $ns $n(6) $n(7) 1000Mb 10ms OpQueue $queue_length
my-duplex-link2 $ns $n(7) $n(8) 1000Mb 10ms OpQueue $queue_length
my-duplex-link2 $ns $n(8) $n(11) 1000Mb 10ms OpQueue $queue_length
my-duplex-link2 $ns $n(8) $n(9) 1000Mb 10ms OpQueue $queue_length
my-duplex-link2 $ns $n(8) $n(13) 1000Mb 10ms OpQueue $queue_length
my-duplex-link2 $ns $n(10) $n(11) 1000Mb 10ms OpQueue $queue_length
my-duplex-link2 $ns $n(10) $n(13) 1000Mb 10ms OpQueue $queue_length
my-duplex-link2 $ns $n(11) $n(12) 1000Mb 10ms OpQueue $queue_length
my-duplex-link2 $ns $n(12) $n(13) 1000Mb 10ms OpQueue $queue_length
for {set i 14} {$i < 16} {incr i}
{
        $ns duplex-link $n($i) $n(0) 155Mb 1ms DropTail
        $ns queue-limit $n($i) $n(0) $queue_length
        $ns queue-limit $n(0) $n($i) $queue_length
}
for {set i 16} {$i < 18} {incr i}
{
        $ns duplex-link $n($i) $n(1) 155Mb 1ms DropTail
        $ns queue-limit $n($i) $n(1) $queue_length
        $ns queue-limit $n(1) $n($i) $queue_length
}
```

```
for {set i 18} {$i < 20} {incr i}
{
        $ns duplex-link $n($i) $n(2) 155Mb 1ms DropTail
        $ns queue-limit $n($i) $n(2) $queue_length
        $ns queue-limit $n(2) $n($i) $queue_length
}
for {set i 20} {$i < 22} {incr i}
{
        $ns duplex-link $n($i) $n(5) 155Mb 1ms DropTail
        $ns queue-limit $n($i) $n(5) $queue_length
        $ns queue-limit $n(5) $n($i) $queue_length
}
for {set i 22} {$i < 24} {incr i}
{
        $ns duplex-link $n($i) $n(7) 155Mb 1ms DropTail
        $ns queue-limit $n($i) $n(7) $queue_length
        $ns queue-limit $n(7) $n($i) $queue_length
}
for {set i 24} {$i < 26} {incr i}
{
        $ns duplex-link $n($i) $n(10) 155Mb 1ms DropTail
        $ns queue-limit $n($i) $n(10) $queue_length
        $ns queue-limit $n(10) $n($i) $queue_length
}
```

```
for {set i 26} {$i < 28} {incr i}
{
	$ns duplex-link $n($i) $n(11) 155Mb 1ms DropTail
	$ns queue-limit $n($i) $n(11) $queue_length
	$ns queue-limit $n(11) $n($i) $queue_length
}
for {set i 28} {$i < 30} {incr i}
{
	$ns duplex-link $n($i) $n(12) 155Mb 1ms DropTail
	$ns queue-limit $n($i) $n(12) $queue_length
	$ns queue-limit $n(12) $n($i) $queue_length
}
for {set i 30} {$i < 32} {incr i}
{
	$ns duplex-link $n($i) $n(13) 155Mb 1ms DropTail
	$ns queue-limit $n($i) $n(13) $queue_length
	$ns queue-limit $n(13) $n($i) $queue_length
}
for {set i 32} {$i < 34} {incr i}
{
	$ns duplex-link $n($i) $n(9) 155Mb 1ms DropTail
	$ns queue-limit $n($i) $n(9) $queue_length
	$ns queue-limit $n(9) $n($i) $queue_length
}
```

```
for {set i 14} {$i < 16} {incr i}
{
        set d [expr $i + 10]
#Create a TCP agent and attach it to node n0
        set cbr($i) [new Agent/TCP/Reno]
        $ns attach-agent $n($i) $cbr($i)
        $cbr($i) target [$n($i) set src_agent_]
        set ftp($i) [$cbr($i) attach-source FTP]
        set null($i) [new Agent/TCPSink]
        $ns attach-agent $n($d) $null($i)
        $null($i) target [$n($d) set src_agent_]
        $ns connect $cbr($i) $null($i)
        set temp [$n($i) set src_agent_]
        $temp install_connection $d      $i $d   $i 0 10 $d
        set temp [$n($d) set src_agent_]
        $temp install_connection $i      $d $i   $d 10 0 $i
 }
for {set i 16} {$i < 18} {incr i}
{
        set d [expr $i + 10]
#Create a TCP agent and attach it to node n0
        set cbr($i) [new Agent/TCP/Reno]
        $ns attach-agent $n($i) $cbr($i)
        $cbr($i) target [$n($i) set src_agent_]
```

```
          set ftp($i) [$cbr($i) attach-source FTP]

          set null($i) [new Agent/TCPSink]
          $ns attach-agent $n($d) $null($i)
          $null($i) target [$n($d) set src_agent_]
          $ns connect $cbr($i) $null($i)
          set temp [$n($i) set src_agent_]
          $temp install_connection $d       $i $d   $i 1 11 $d
          set temp [$n($d) set src_agent_]
          $temp install_connection $i       $d $i   $d 11 1 $i
 }
for {set i 18} {$i < 20} {incr i}
{
          set d [expr $i + 10]
#Create a TCP agent and attach it to node n0
          set cbr($i) [new Agent/TCP/Reno]
          $ns attach-agent $n($i) $cbr($i)
          $cbr($i) target [$n($i) set src_agent_]
          set ftp($i) [$cbr($i) attach-source FTP]
          set null($i) [new Agent/TCPSink]
          $ns attach-agent $n($d) $null($i)
          $null($i) target [$n($d) set src_agent_]
          $ns connect $cbr($i) $null($i)
          set temp [$n($i) set src_agent_]
```

```
        $temp install_connection $d     $i $d  $i 2 12 $d

        set temp [$n($d) set src_agent_]

        $temp install_connection $i     $d $i  $d 12 2 $i

 }

for {set i 20} {$i < 22} {incr i}

{

        set d [expr $i + 10]

# Create a TCP agent and attach it to node n0

        set cbr($i) [new Agent/TCP/Reno]

        $ns attach-agent $n($i) $cbr($i)

        $cbr($i) target [$n($i) set src_agent_]

        set ftp($i) [$cbr($i) attach-source FTP]

        set null($i) [new Agent/TCPSink]

        $ns attach-agent $n($d) $null($i)

        $null($i) target [$n($d) set src_agent_]

        $ns connect $cbr($i) $null($i)

        set temp [$n($i) set src_agent_]

        $temp install_connection $d     $i $d  $i 5 13 $d

        set temp [$n($d) set src_agent_]

        $temp install_connection $i     $d $i  $d 13 5 $i

 }

for {set i 22} {$i < 24} {incr i}

{

        set d [expr $i + 10]
```

```
#Create a TCP agent and attach it to node n0
        set cbr($i) [new Agent/TCP/Reno]
        $ns attach-agent $n($i) $cbr($i)
        $cbr($i) target [$n($i) set src_agent_]
        set ftp($i) [$cbr($i) attach-source FTP]
        set null($i) [new Agent/TCPSink]
        $ns attach-agent $n($d) $null($i)
        $null($i) target [$n($d) set src_agent_]
        $ns connect $cbr($i) $null($i)
        set temp [$n($i) set src_agent_]
        $temp install_connection $d      $i $d   $i 7 9 $d
        set temp [$n($d) set src_agent_]
        $temp install_connection $i      $d $i   $d 9 7 $i
 }
# Optical node color start
for {set i 0} {$i < 14} {incr i}
{
        $n($i) color red
# Optical node color end
# Electronic node color start
for {set i 14} {$i < 34} {incr i}
{
        $n($i) color blue
}
```

```
# Electronic node color end
	set rand2RNG [new RNG]
	set timeRNG [new RNG]
	set stimeRNG [new RNG]
	set arrivalrate 0.01
	set stoptime 1000
	set t [expr $arrivalrate*$stoptime]
	set rand2_ [new RandomVariable/Uniform]
	$rand2_ set min_ 14
	$rand2_ set max_ 23
	$rand2_ use-rng $rand2RNG
	set time_ [new RandomVariable/Uniform]
	$time_ set min_ 0
	$time_ set max_ $t
	$time_ use-rng $timeRNG
	set stime_ [new RandomVariable/Uniform]
	$stime_ set min_ 0
	$stime_ set max_ 20
	$stime_ use-rng $stimeRNG
	array set stftp {}
	set file [open nodes.txt w]
proc ftpstart {}
{
	set ns [Simulator instance]
```

```
    global traf t time_ rand2_ stime_ file stftp i ftp
    set now [$ns now]
    set time [expr round([$time_ value])]
    set stoptime [expr round([$stime_ value])]
    set stnode [expr round([$rand2_ value])]
    foreach {node n} [array get stftp]
{
    if {$n == $stnode}
    {
        $ns at $now "ftpstart"
        return 1
    }
}
set stftp(node($stnode)) $stnode
foreach {node n} [array get stftp]
{
    puts "$node $n"
}
    set starttime [expr $now+$time]
    $ns at $starttime "$ftp($stnode) start"
    $ns at [expr $starttime+$stoptime] "ftpstop $stnode"
    puts $file "$stnode starts at $starttime stops at [expr $starttime+$stoptime] [expr $starttime%10]"
    $ns at [expr $now+$t] "ftpstart"
```

```
}
proc ftpstop {snode}
{
        global stftp ns ftp
        set now [$ns now]
        unset stftp(node($snode))
        $ns at $now "$ftp($snode) stop"
}
        $ns at 0 "ftpstart"
        $ns at 2020 "finish"
        $ns run
de($stnode)) $stnode
foreach {node n} [array get stftp]
{
        puts "$node $n"
}
set starttime [expr $now+$time]
$ns at $starttime "$ftp($stnode) start"
$ns at [expr $starttime+$stoptime] "ftp
```

Appendix 2(B)

Modifications on nOBS Framework

To simulate spectral threat in nOBS framework in ns2, it is required to add the following code module into the agent.cc file found in the nOBS implementation. The agent.cc file models the Burst agent component for nOBS framework.

```
int lambdastart=0;

lambdaend=6;

if((conversion==1)|(burstch->lambda==-1))

{

        //full wavelength conversion is possible in the node or burst is still in electronic domain so we can choose a lambda

        lambdastart=0;

        lambdaend=MAX_LAMBDA;

}

else if(conversion==2)

{

        // Wavelength conversion not possible or there is a lambda selected before for this burst

        lambdastart=burstch->lambda;

        lambdaend=burstch->lambda+1;

}

if(DEBUG==1)

        printf("conversion %d burstch->lambda %d lambdastart %d lambdaend %d\n",conversion,burstch->lambda,lambdastart, lambdaend);
```

```
for(k=lambdastart;k<lambdaend;k++)
{
    if(DEBUG==1)
    printf("llllllll k %d\n",k);
    temp=Head[k];
    if(temp->end!=-1)
    {
        if((temp->start -switch_time>= leave)&&(arrival < min_gap))
        {
            if(DEBUG==1)
            {
                printf("k%d WILL BE THE HEAD temp->Next->start -switch_time>= leave \n",k);
                bestlambda=k;
                besttemp=temp;
                is_head=1;
            }
        }
        finishit=0;
    }
}
// End
```

The next step is to add the following codes into the op_classifier.cc which models the optical classifier component for nOBS patch. This is presented below.

```
Void Op-burst_Classifier::InitializeSpectralAttack(int option)
{
        int i, j, CompromiseVSNode, lamda;
        char ch;
        FILE *fp = NULL;
        CompromiseVSNode = OpticalDefaults::instance()
        ->COMPROMISEVS_NODE;
        if (CompromiseVSNode != 0)
        return;
        senderInfo = fopen("/home/terence/ns-allinone-2.27/ns-
        2.27/scripts/nsfnet/no_of_req_sent.txt", "w");
        fclose(senderInfo);
        receiverInfo = fopen("/home/terence/ns-allinone-2.27/ns-
        2.27/scripts/nsfnet/no_of_req_success.txt", "w");
        fclose(receiverInfo);
        virtualsourceinfo = fopen("/home/terence/ns-allinone-2.27/ns-
        2.27/scripts/nsfnet/no_of_req_fails.txt", "w");
        fclose(virtualsourceinfo);
        if (option == VIRTUAL_SOURCE_BASED_MULTICASTING)
        {
                printf("Initializing VIRTUAL SOURCE BASED
                MULTICASTING\n");
                printf("Initializing SPECTRAL ATTACK\n");
                for(k=0;k<=6;k++)
                {
                        If(conversion==2)
                        {
                                if(DEBUG==-1)
                                {
                                    bestlambda=k;
                                    besttemp=temp;
                                    temp=besttemp;
                                    temp=Head[k];
                                }
                                printf("k%d WILL BE THE HEAD temp->Next->start -
                                switch_time>= leave \n",k);
                                printf("found a lambda %d WITH no contention  %.15f
                                leave %.15f\n\n",bestlambda,arrival, leave);
                        }
                }
                if(bestlambda!=-1)
                {
                // Store this lambda inside the control packet
                        burstch->lambda=bestlambda;
                        printf("praba - src = %d, dest = %d, burstid = %d, lambda =
                        %d, bursttype = %d\n", burstch->source, burstch->destination,
                        burstch->burst_id, burstch->lambda, burstch->burst_type);
                }
        }
}
// End
```

To simulate burstification threat in nOBS framework in ns2, it is required to add the following code module into the agent.cc file. This is presented below.

```
void Op-burst_Classifier::InitializeBurstificationAttack(int option)
{
    int i, j, CompromiseVSNode, MinBurstlen,Maxburstlen,temp;
    int changelen;
    char ch;
    FILE *fp = NULL;
    CompromiseVSNode = OpticalDefaults::instance()-
>COMPROMISEVS_NODE;
    if (CompromiseVSNode != 0)
    return;
// Just opened the file in write mode, such that the file contents will be emptied.
    senderInfo = fopen("/home/terence/ns-allinone-2.27/ns-
    2.27/scripts/nsfnet/no_of_req_sent.txt", "w");
    fclose(senderInfo);
    receiverInfo = fopen("/home/terence/ns-allinone-2.27/ns-
    2.27/scripts/nsfnet/no_of_req_success.txt", "w");
    fclose(receiverInfo);
    virtualsourceinfo = fopen("/home/terence/ns-allinone-2.27/ns-
    2.27/scripts/nsfnet/no_of_req_fails.txt", "w");
    fclose(virtualsourceinfo);
    started=clock();
    if (option == VIRTUAL_SOURCE_BASED_MULTICASTING)
    {
        printf("Initializing VIRTUAL SOURCE BASED
    MULTICASTING\n");
        printf("Initializing BURSTIFICATION ATTACK\n");
        If(DEBUG==-1)
    // Update burst length
        {
            temp=burstlen;
            if(burstlen<=MaxBurstlen)
            {
                temp=burstlen;
                i=temp;
                temp=changelen;
                i=temp;
                temp=Head[i];
            }
            printf("k%d WILL BE THE HEAD temp->Next->start -
            switch_time>= leave \n",k);
            printf("found a burstlen %d WITH no contention  %.15f leave
            %.15f\n\n",burstlen,arrival, leave);
        }
    }
// Store this burstlen inside the control packet
    burstch->burstlen=changelen;
    printf("praba - src = %d, dest = %d, burstid = %d, burstlen = %d, bursttype =
    %d\n", burstch->source, burstch->destination, burstch->burst_id, burstch-
    >burstlen, burstch->burst_type)
    ended=clock();
}
// End
```

APPENDIX 3(A)

BASIC OPTICAL COMPONENTS

Development of optical networks cannot be fully understood without having a basic knowledge about its key building blocks. We briefly name the major optical components used in optical networks and describe their basic functionalities.

Couplers:

These devices combine light into fiber or split light out of a fiber. Three common types of couplers are splitters, combiners, and directional couplers.

Optical fiber:

Optical fibers are essential building blocks of any optical transmission system. Typical fiber characteristics include low insertion losses, low wavelength shift, and low cross talk from adjacent signals.

Optical amplifiers:

These devices play an important part in optical transmission systems. Optical signals are prone to losses in the fibers as they propagate through them. These signals have to be strengthened to enable propagation. Optical amplifiers are used in three different ways in a fiber transmission system: power amplifier, line amplifier, preamplifiers. Depending on the fiber type, and the distance between the transmitter and receiver, different types of optical amplifiers may be needed. Common types of amplifiers include Erbium-Doped Fiber Amplifier (EDFA), Semiconductor Optical Amplifier (SOA) and Raman Amplifier.

Transmitters and receivers:

The basic operation of these devices is converting digital signals into optical signals or converting optical signals to digital signals, respectively. Transmitters differ depending on the type of the lasers they use. Examples of laser types include Semiconductor Laser Diodes, Fabry - Perot Lasers, External Cavity Laser, and Mechanically Tuned Lasers. An important characteristic of an optical receiver is its sensitivity toward the received optical signal.

Switches:

Switches are the vital components in any optical networks. Switches allow optical signals to be switched without having to convert them to electronic signals. Different types of switches can be employed in optical networks such as Fiber cross-connects, wavelength-routing switches, and photonic packet switches.

Wavelength converters:

The function of wavelength converters is to convert data from the incoming wavelength to an outgoing wavelength. Classification of wavelength converters is done based on the wavelength range they operate. The basic types of wavelength converters are fixed-input/fixed-output, variable-input/fixed-output, fixed-input/variable-output, and variable-input/variable-output.

APPENDIX 3(B)

QOS METRICS FOR OBS NETWORKS

In OBS networks having the following important metrics for analyzing their QoS. The matrices are:

- Burst Throughput (η)
- Burst Latency (τ)
- Transmission Delay (τt)
- Propagation Delay (τp)
- Burst Loss Probability (E_b)

Burst Throughput (η)

Burst Throughput (η) refers to the amount of received burst data per unit time. To provide lower burst losses for multicast traffic routing this parameter must be higher. The throughput in an Optical Burst Switched network is given by:

$$\textbf{Burst Throughput } (\eta) = R_b / T$$

where, R_b → *Number of received bursts*

T → *Total Network time*

Burst Latency (τ)

Burst Latency (τ) represents the overall delay incurred by a Burst. The delay variation in Burst transfer is Jitter (J). These delays include propagation delay, transmission delay, and queuing delay. Due to lack of potential Optical buffers (Fiber Delay Lines), the queuing delay in OBS Networks is absent.

$$\textbf{Burst Latency } (\tau) = \tau_t + \tau_p$$

where,τ_t → Transmission Delay

τ_p → Propagation Delay

Transmission Delay (τ_t)

Transmission Delay (τ_t) is defined as the time taken to push a Burst to the transmission line.

Transmission Delay (τ_t) = Data Burst Size / B_o

where, B_o → Optical Bandwidth

Propagation Delay (τ_p)

Propagation Delay (τ_p) is the time taken by the Burst to travel from Source to destination.

Propagation Delay (τ_p) = $t_{cp} + t_b$

where, t_{cp} → Time taken to process CP at cores
t_b → Time taken to burstify/ deburstify at edges

Burst Loss Probability (ø)

Burst Loss Probability (ø) is the probability of the Burst Losses at the intermediate nodes due to contention or insufficient QoS. Burst losses due to contention are termed as Contention blocking. Burst losses due to insufficient QoS are termed as QoS blocking. It is the ratio between the Burst losses at the cores to the total Burst generated at the edges.

Burst Loss Probability (ø) = L_b/ G_b

where, L_b → Burst losses at the Core nodes
G_b →Burst generated at edge nodes