uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Sonia Parveen
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.A.Sc. (Electrical Engineering)
GRADE / DEGREE

School of Information Technology and Engineering
FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Simulation study of an Optical Burst Switched Agile All-Photonic Network

TITRE DE LA THÈSE / TITLE OF THESIS

T. Hall
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

G. Bochmann

C. Huang

Gary W. Slater
LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES /
DEAN OF THE FACULTY OF GRADUATE AND POSTDOCORAL STUDIES

# Simulation study of an Optical Burst Switched Agile All-Photonic Network

By

Sonia Parveen

A Thesis submitted to the Faculty of Graduate Studies and Postdoctoral Studies in Partial fulfillment of the requirements for the degree of

Masters of Applied Science
In
Electrical Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Information Technology and Engineering
University of Ottawa

*Your file Votre référence*
*ISBN: 0-494-11376-6*
*Our file Notre référence*
*ISBN: 0-494-11376-6*

Dedicated to my parents

## Acknowledgement

I would like to take this opportunity to express my heartiest thanks to my supervisor, Professor Trevor J. Hall, for providing guidelines and suggestions to me to pursue this work. Without his invaluable help and assistance, it would not have been possible to continue and complete this thesis.

I would also like to extend my sincere thanks to Dr. Sofia Paredes, not only for her constant help, thousands of suggestions and guidelines, but also for her understanding and encouragement to finish this thesis. Also sincere thanks to all of my friends and colleagues for their help and advices, especially Mazen for helping me with OpNet, and most importantly to my parents for their support and foresting me up to this point without realizing the hardship of this world.

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) and industrial and government partners, through the Agile All-Photonic Networks (AAPN) Research Network. I would also like to extend my thanks to the people involved in these two initiatives.

# Abstract

Optical Burst Switching (OBS) is considered to enable bandwidth in the context of an Agile All-Photonic Network (AAPN). However, before deploying the OBS technique in the AAPN backbone, one must consider the best way to aggregate the variable length packets into burst of suitable size. Burst assembly mechanism can be *timer-based*, *threshold-based* and *hybrid*. In this work, we developed an analytical model, which models the probability distribution function of aggregation delays for different types of burst aggregation methods. The presented model is numerically evaluated using MATLAB code. The accuracy of the analytical model is verified by OpNet simulation. The impact of different types of burst aggregation methods and associated parameters (e.g. *threshold-value* and *time-out* value) on blocking probability and mean delays is also investigated. Another major concern in optical burst switched network is contention, which occurs when more than two bursts are destined to the same output port at the same time. In this thesis, a simple reservation approach, which copes completely with the packet losses associated with conventional OBS network and retains the simplicity of the core node, has been implemented in the context of AAPN architecture.

# Table of Contents

# List of Figures

# List of Symbols

| | |
|---|---|
| $T_{out}$ | Aggregation time out |
| $B_{\max}$ | Threshold value |
| $q_{sd}$ | Aggregation queue at ingress node $s$ for destination $d$ |
| $d$ | Destination address |
| $\lambda$ | Mean arrival rate |
| $t_n$ | Time packet at location $n$ arrives |
| $\tau_n$ | Time packet at location $n$ waits before the burst departs |
| $q_n$ | The probability that location $n$ in burst $m$ is occupied |
| $p(\tau)$ | Probability distribution of the delay $\tau$ |
| $\mu$ | Mean inter-arrival time |
| $L$ | Packet length |
| $R$ | Transmission speed |
| $B$ | Bi-stochastic matrix |
| $\alpha$ | Tail index of a Pareto distribution |
| $H$ | Hurst parameter |

## List of Acronyms

| | |
|---|---|
| AAPN | Agile All-Photonic Network |
| TDM | Time Division Multiplexing |
| OBS | Optical Burst Switching |
| QoS | Quality of Service |
| VOQ | Virtual Output Queue |
| WDM | Wavelength Division Multiplexing |
| SCU | Switch Control Unit |
| FDL | Fibre Delay Line |
| JET | Just Enough Time |
| JIT | Just In Time |
| TAW | Tell and Wait |
| WR-OBS | Wavelength routed Optical burst switched network |
| BCP | Burst Control Packet |
| AWTA | Adaptive Wavelength-Timeslot Allocation |

# 1 Introduction

## 1.1 Motivation and Historical background

Broadband communications over the Internet will soon deliver, virtually instantly and transparently, 'anytime, anywhere' multimedia communications. This will require the development of highly flexible, intelligent networks that offer large amounts of bandwidth on demand. Current communication networks already exploit the power of light for transmitting the signal over fibre. However, the signal of light, originating in electrical format by the traffic sources, still need to be converted back into electrical signals at various stages of the communication procedure, especially during the switching process and also during the regeneration of the transmitted signals at certain points of the long-haul transmission, which is essential to preserve the integrity of the data format. These opto-electronic conversions are expensive, require huge amounts of power, and also limit scalability of the network – which is a bottleneck in today's optical communications networks.

This bottleneck can be removed by developing an all-photonic network core that could potentially stretch very close to the end-user by dramatically reducing optical/electrical conversions. Any practical switching paradigm to realize this objective should contain the following key features: agility, that is, the ability to perform time domain multiplexing to allocate bandwidth to traffic flows dynamically as the demand varies; control and routing functionality concentrated at the edge switches that surround the photonic core, and; rapidly reconfigurable all-photonic space-switching in the core. To address this vision, a Research Network entitled "Agile All-Photonic Networks" (AAPN) was launched in 2003, which aims to embrace all of the essential elements that are required to realize all-photonic networks of the future.

AAPN is based on the Petaweb architecture, a topology comprised of a set of overlaid star networks, with an optical space switch at each of the star centers and hybrid photonic/electronic switches at the edges, developed by R. Vickers, together with researchers at Nortel Networks under DARPARA funding as part of the Next Generation Internet Initiative and described in [1]. The capacity of such 'Petaweb' architecture reaches several hundred Pb/s as more overlays are introduced. The overlaid star topology has several advantages: it is extremely efficient from the switching viewpoint, robust, and potentially very scalable if the central switch is scalable. However, the proposed 'Petaweb' architecture includes very limited amount of agility. At any given instant, each wavelength within the photonic core provides a path from the source to a destination node. In the Petaweb architecture [1], such a path exists for a substantial time (on the order of 5 sec.) between reconfiguration of the core node switches. To enable bandwidth sharing at sub-wavelength channel capacity granularity in the AAPN, it must be possible to create and destroy wavelength paths within milli/micro seconds. Two schemes are under consideration for this purpose. The first scheme is based on slotted *Time Division Multiplexing* (TDM) and the second scheme is based on Optical Burst Switching (OBS).

Optical burst switching (OBS) was first proposed in [3][4][5] to provide optical switching for next generation Internet traffic in a flexible yet feasible way. OBS is based on the concept of electronic burst switching network. Although the idea of electronic burst switching [6] [7] has been introduced in 1980s, it has never been successful due to its complexity and realisation requirements are comparable to that of more flexible electronic packet switching technique. However, with the introduction of very high-speed optical transmission mechanism, the situation has changed. Now, there is an even increasing discrepancy between the optical transmission speed and electronic switching capability [8]. Moreover, it is advantageous to keep the data in the optical domain and to avoid O/E/O conversion due to cost and complexity aspects. Nevertheless, optical packet switching is still too immature to perform all the processing in the optical domain. Hence, a hybrid approach like burst switching seems promising for optical networks [8].

Optical Burst Switching (OBS) is a technique, where several IP packets with the same destination and other common attributes such as Quality of Service (QoS) parameters are aggregated into a burst and forwarded through the network as one entity. The idea is to use no buffering at the intermediate nodes and to send the control packet (header) and the associated payload on two different wavelength channels with the control packet being sent slightly ahead in time; thereby permitting the sophisticated electronic processing of the control packet. A further detail description of Optical Burst Switched Agile All Photonic Network (OBS-AAPN) architecture and its associated design issues is given in chapter 2.

Before deploying the OBS technique in the AAPN backbone one must consider the best way to aggregate the variable length packets into suitable sized bursts. Data burst assembly involves decisions on two key parameters: the maximum waiting time in the aggregation queue (*time- out,* $T_{out}$) and the maximum size of the burst (*threshold value*). Based on these two parameters burst assembly mechanism can be classified as: *timer-based* approach and *threshold-based* (also termed as *length-out)* approach. In a *timer-based* burst assembly approach, a timer is started upon the arrival of the first packet in the empty queue, and a burst is sent into the optical network as soon as the *time-out* occurs. At low input traffic volumes, this scheme offers the minimum queuing delay in the aggregation buffers. However, under high input traffic volumes, this scheme might generate bursts that are quite long, which can make the transmission and propagation delay intolerable. In a *threshold-based* burst assembly approach a burst is created and sent out when the size of the aggregation queue reaches or exceeds a certain threshold value of $B_{max}$ in bits. The major drawback of this approach is: this scheme does not give any guarantee on the assembly delay that the packets will experience during the burst aggregation process. To address the deficiency associated with each type of burst assembly mechanism mentioned above both the *time- based* and *threshold-based* approaches can be combined into a single *hybrid approach.* Recent OBS literatures on burst assembly algorithms have mainly focused on its effect on the self- similar characteristics of the aggregated traffic. However, as far as we are aware no prior work has analyzed the probability distribution of packet delays in the aggregation queues during the burst assembly mechanism.

Another major concern in optical burst switched networks is contention, which occurs when multiple bursts contend for the same link. Existing approaches to avoid packet losses in an OBS network are: optical buffering [9], wavelength conversion [10] or deflection routing [11]. However, deploying the OBS technique in AAPN architecture does not allow any buffering or wavelength conversion mechanism in the core node. Hence, the idea of using optical buffering or wavelength conversion to avoid packet losses can be discarded. Contention resolution through deflection routing is also possibly ruled out. Hence, it is required to introduce some methods to avoid packet losses in the context of AAPN.

## 1.2 Research Objective

The objectives of this research are:

- To find the best way of aggregating the variable length packets into suitable sized bursts in an Optical Burst Switched Agile All Photonic Network (OBS-AAPN).
- To analyse the probability distribution of delays in the aggregation queue.
- To investigate the impact of different types of burst aggregation methods and associated parameters (e.g., maximum burst size and time-out value) on mean delays and blocking probability in an OBS-AAPN.
- To introduce a method that avoids the shortcomings of the existing approaches and retains the simplicity of the core node to avoid packet losses in the network.

## 1.3 Thesis Contribution

The main contributions of the work presented in this thesis can be listed as:

- The development of an analytical model to evaluate the probability distribution of packet delays during the aggregation process for Poisson arrivals. The presented model is numerically evaluated
- The implementation of a scalable and modular framework for simulating an OBS-AAPN in OpNet Modeler, whose main purpose was to verify the accuracy of the analytical model and to observe the impact of different types of burst aggregation approaches and its related parameters ($B_{\max}$ and $T_{out}$) on the performance of an OBS-AAPN under different traffic scenario, however this is open to other kind of experiments of an OBS-AAPN.
- The implementation of a simple, enhanced reservation method with a single acknowledgement, which eliminates the packet losses observed in an OBS-

AAPN.

## 1.4 Structure of the thesis

This thesis consists of five chapters. This chapter outlined a brief introduction to OBS-AAPN as well as the research objectives and thesis contribution. The recent literature on OBS presents many variations and designs of OBS. In **Chapter 2**, those variations of OBS, which are appropriate in the context of AAPN, are discussed and they are categorized based on their effect on network operation.

**Chapter 3** presents an analytical model to evaluate the probability distribution of delays for different types of burst aggregation methods and the presented model is numerically evaluated by Matlab code .The numerical analysis shows that the probability distribution of delays manifest a spike superimposed on a pedestal. In **Chapter 3,** a scalable and modular framework for simulating an OBS-AAPN has also been implemented using OpNet Modeler to check the accuracy of the analytical model and to observe the blocking probability and mean delays of different approaches to burst aggregation.

**Chapter 4** describes the proposed OBS technique with reservation approach and also presents the simulation results. Performance measures of interest were delay statistics and data was obtained using Poisson, non-uniform and bursty types of traffic.

**Chapter 5** identifies major accomplishments of this research and gives future directions.

**Appendix A** explains the top-level network architecture and the network model components including the edge nodes, core nodes, and different process models built in OpNet Modeler to simulate an OBS-AAPN and finally, **Appendix B** presents the OpNet code written for developing the simulation model.

# 2 Optical Burst Switching as a bandwidth sharing method in an Agile All Photonic Network

## 2.1 Introduction

Optical Burst Switching (OBS) is considered for subdividing the bandwidth of each wavelength in the context of an Agile All-Photonic Network (AAPN), which is an overlaid star architecture that can be deployed to provide services to IP and others. Several architectural and design issues need to be addressed, however before deploying the OBS technique in an AAPN backbone. In this chapter, the architecture of an AAPN is discussed, focusing mainly on topological aspects and resource allocation schemes. Different OBS design issues, which are appropriate for AAPN, are also discussed.

This chapter is organized as follows: section 2.2 presents a topological overview of an Agile All Photonic Network. Section 2.3 discusses different types of bandwidth allocation schemes considered for AAPN. Section 2.4 gives a detailed description about optical burst switching (OBS) technique and finally, section 2.5 discusses different types of OBS signaling schemes; wavelength allocation techniques; reservation protocols etc. in the context of AAPN.

## 2.2 AAPN Architecture

The majority of the existing photonic networks have a mesh topology, which distributes the load over many switches. However, in the AAPN, the photonic core switches have enormous capacity, which together with timing considerations leads to an overlaid star topology [1]. The overlaid star architecture consists of a set of hybrid photonic/electronic edge nodes connected together via a wavelength stack of bufferless transparent switches placed at the core nodes. An example of a 2 layer overlaid star architecture is shown in Figure 2-1. In this diagram the single lines represent the bi-directional traffic flow from the edge nodes to the core node and the core node is responsible for performing the routing of the traffic between the edge nodes.

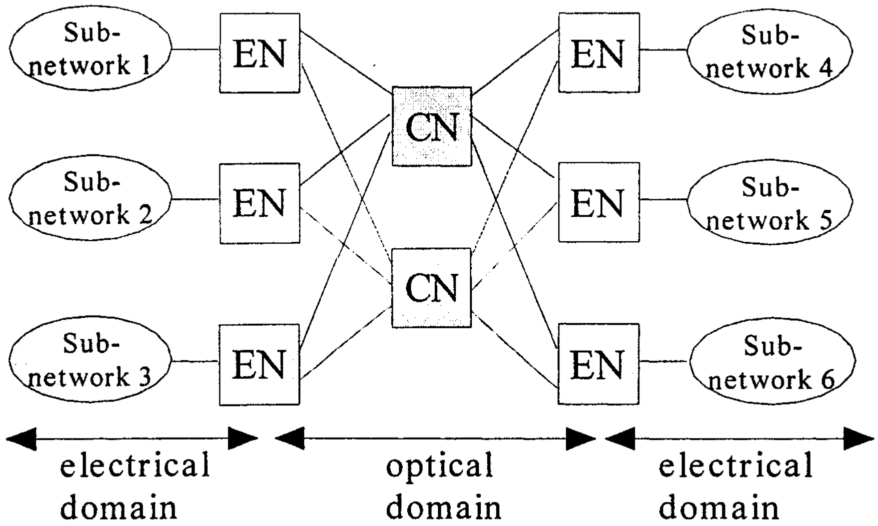


Figure 2-1. Overlaid Star topology

In an AAPN, data is transferred in fixed time intervals. During this interval, a transparent light path is established at a particular wavelength between the edge nodes. The edge node acts as a boundary between the electronic world outside the AAPN and the photonic domain of AAPN. Hence, it is a hybrid electrical/photonic component and it provides optoelectronic conversion of the incoming signals. Each edge node contains virtual output queues (VOQ) for the traffic destined to each of the other edge nodes. If the network consists of N edge nodes then every edge node will have (N-1) VOQs. The incoming packets are directed to appropriate VOQ based on the destination address. The traffic collected in the VOQs is aggregated and sent across the network as a single entity.

In the AAPN architecture, the number of edge nodes is determined by the number of core switch ports. Current switching technology is limited to 64 ports; hence to support more than 64 edge nodes a method of sharing ports is required in a core node. The number of wavelengths supported by each edge node can vary. However, each edge node is connected to multiple photonic core switches in the overlaid architecture to provide robustness in case of link or core node failure. The division of edge node traffic between different star layers of AAPN need not be into equal parts. However, each edge node must be capable of transmitting the traffic to every other edge node in the network via more than one star [2].

The core nodes are responsible for transporting the packets all-optically. The adaptive AAPN core consists of several bufferless core nodes, which can be distributed over a large geographic area. Each core node consists of a layered switch with each layer allowing the time slots to switch at a given wavelength. Hence, the number of layers in a core switch is equal to the number of wavelengths on the network.

The function of the control plane is to provide an effective time synchronization method due to propagation delays between source nodes and the core nodes. To ensure proper synchronization, the network should be equipped with clock counters, which are tied to the core nodes and the edge nodes. If a selector is used for port sharing, it has to be also synchronized with the core node. The core controller also selects the light paths through the associated core nodes and also reconfigures the paths in response to the requests sent by the edge nodes based on dynamic changes in traffic volumes.

The photonic path between an edge node and the core node in an AAPN is shown in Figure 2-2. The path starts with a laser at the edge module. Each edge node uses multiple lasers; hence a wavelength multiplexer is used in the edge node to support multiple wavelengths on a single fibre. The next component is the selector. The function of the selector is to combine the traffic of multiple edge nodes onto a single fibre to allow port sharing in the core node. In the selector, the incoming signal from different edge nodes are demultiplexed into individual wavelengths and send to one of the ports of the core node through a multiplexer. The inbound connection, from the ingress edge node to the core node will be mirrored in the outbound connection from the core node to the egress edge node. Amplifiers, regenerators and other devices are also introduced in the photonic path that is not shown in Figure 2-2. The number and spacing of these devices depend on the transmission design.

## 2.3 Bandwidth allocation schemes in AAPN

In the last few years there has been an explosive growth in Internet Protocol (IP) traffic that has motivated research to develop new high-speed transmission and switching technologies capable of supporting this growth. Wavelength-division multiplexing (WDM) is one of the most promising technologies for the next-generation IP backbone network due to its ability to support a large number of high-speed channels on a single fibre. This provides enormous bandwidth in the physical layer. The challenge then is how to transmit IP traffic over WDM most effectively.

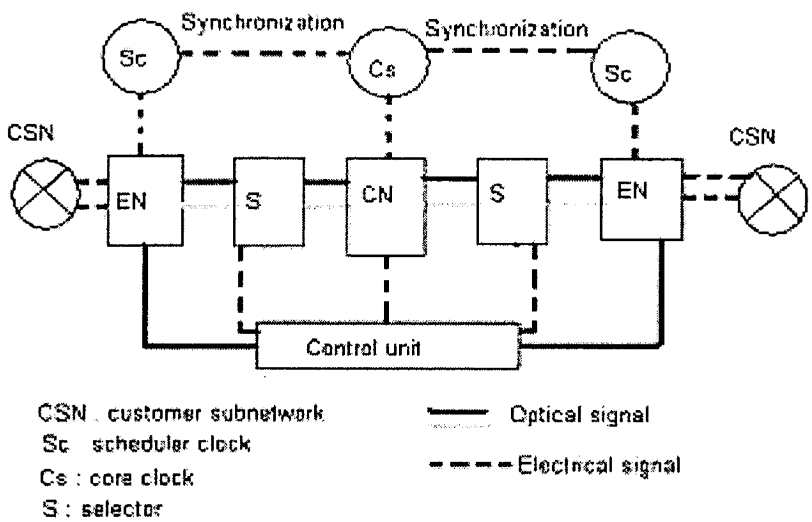


Figure 2-2. Photonic path of an AAPN

In order to carry data among different edge nodes, wavelength assignment was recommended as an approach, where a direct light path is set up from the source to a destination node to carry the information. However, in the overlaid star architecture [1], such a path exists for a substantial time (on the order of 5 sec.) between reconfiguration of the core node switches. To enable bandwidth sharing at sub-wavelength channel capacity granularity in the AAPN, it must be possible to create and destroy wavelength paths within milli/micro seconds. Two schemes can be considered for this purpose. The first scheme is based on a *Time Division Multiplexing* (TDM) technique and the second scheme is based on Optical Burst Switching. A brief description of different bandwidth allocation schemes relevant to AAPN is given in the following.

### 2.3.1 Time Division Multiplexing Technique

Time Division Multiplexing is a technique where the transmission facilities are shared via rotating time slots. In this scheme, a frame (a given repeating period) is divided into $F$ distinct time slots and each slot is associated with a particular configuration of the network switches. It is then possible for a flow to make use of only $1/F$ of the wavelength channel's capacity during one time slot. The number of allocated slots in the frame can be adjusted as the traffic demand varies. The traffic is constantly monitored both in the core node and the edge nodes. If the volume of the traffic flowing between a source-destination pair exceeds or approaches the allocated resources then additional time slots are provided to that pair of edge nodes. On the other hand, if the bandwidth allocated between two edge nodes left underutilized for long periods of time then the number of assigned time slots is reduced.

Time Division Multiplexing is more suitable for predictable traffic where the edge node can easily adapt to the varying traffic by requesting more capacity or canceling the existing connection in advance. It is also well suited for providing bandwidth sharing for long-term connections. However, for bursty traffic with stringent timing requirement in the bandwidth reservation, it may not be a good choice.

### 2.3.2 Optical Burst Switching Technique

The third scheme is based on Optical Burst Switching (OBS); a technique where several IP packets with the same destination and other common attributes such as Quality of Service (QoS) parameters are aggregated into a much larger burst and transmitted through a bufferless network as a single entity. Our research is concentrated on Optical Burst Switching technique. Hence, a more detailed description of this technology will be given in this section.

In the evaluation of optical switching paradigms, the most important switching techniques are: wavelength routing, optical packet switching and optical burst switching. The wavelength routing mechanism uses the concepts of traditional circuit switched networks. In this mechanism, a communication channel is established by choosing a route between the source-destination pair with appropriate wavelengths allocated on the links along the route. If, the signaling protocol works in a distributed mode, then a two way reservation process is needed to establish a light path between a source-destination pair, whereby a source sends a request to make a reservation, then waits for an acknowledgement to come back before transmitting the data. Wavelength routing has the advantage that no optical buffering is required at the intermediate nodes.

However, wavelength routing suffers from several shortcomings. One of the limitations of wavelength routing mechanism is that no statistical bandwidth sharing is allowed between two distinct light paths simultaneously even though one of them is idle, resulting in low bandwidth utilization. Another disadvantage of the wavelength routing technique is that it requires a certain amount of time for setting up and tearing down a light path. If the path holding times are too short then this overhead may cause poor channel usage. From a signaling overhead point of view, this technique will be efficient for long holding times; however, this case leads to a reduced ability to adapt to traffic dynamics, especially if the behavior of the IP traffic is bursty.

To overcome the problems associated with the wavelength routing mechanism another switching technique called *optical packet switching* has been introduced; where the processing unit is a fixed length packet that consists of a header and a payload. In Packet switching, as opposed to wavelength routing, resources are allocated in an on-demand fashion and hence bandwidth utilization can be greatly improved. Optical Packet switching uses a *store and forward* method to transmit the data; hence a packet needs to be buffered completely at each intermediate node before it can be forwarded. However, since there is no optical equivalent of the random access memory, using fibre delay lines is the only way to implement optical buffers. Because of the variations of the header processing at the intermediate nodes, this technique also needs stringent synchronization. Another problem associated with the optical packet switching technique is the high control overhead resulting from small payloads.

In order to carry the IP traffic over WDM in the most efficient way, an all-optical switching method which avoids optical buffering at each intermediate node and which supports fast resource provisioning and asynchronous transmission of variable sized packets must be introduced. This consideration leads to the concept of "burst switching" where the advantages of the above two approaches are retained while their shortcomings are eliminated as possible.

The first step is to change the basic transmission unit from a fixed size packet to a variable length "super packet" also called *burst*. In an optical burst switching technique, several IP packets with the same destination and other common attributes such as Quality of Service (QoS) parameters are aggregated into a much larger burst and transmitted through the network as a single entity. In this way, the controlling overhead is alleviated.

Headers must be processed (which is best done electronically and thus requires O/E/O conversion); therefore they are put on a separate control channel and are sent slightly ahead in time to permit data bursts (which do not require processing) to remain in the optical domain without buffering at the intermediate nodes. OBS uses a one-way resource reservation protocol, which means that a data burst follows its corresponding control packet without waiting for acknowledgement. In this way the significant signaling delay can be reduced. In addition, the complete separation between the control packet and its corresponding burst in both time and wavelength domain can eliminate the problem of synchronization associated with optical packet switching.

## 2.4 OBS Network architecture

The basic concept of an optical burst switched network, which consists of optical core routers and electronic edge routers interconnected by WDM links, is shown in Figure 2.3. Here, the burst assembly/disassembly functions and legacy interfaces (e.g., gigabit Ethernet) are located at the edge routers; whereas the core router, which consists of an optical switching matrix and a switch control unit, performs the routing function entirely in the optical domain.

The function of the ingress routers is to collect IP Packets having the same network egress address and other common attributes such as Quality of Service (QoS) requirements, and to assemble them into bursts. In addition to this, the ingress nodes also create a control packet (burst header) for each data burst. The control packet contains all the necessary routing information including the offset time (see below), the burst length and the destination address. This information is used by the switch control unit (SCU) at each intermediate node to configure the optical switching matrix in order to switch the data burst to the correct destination port in the optical domain

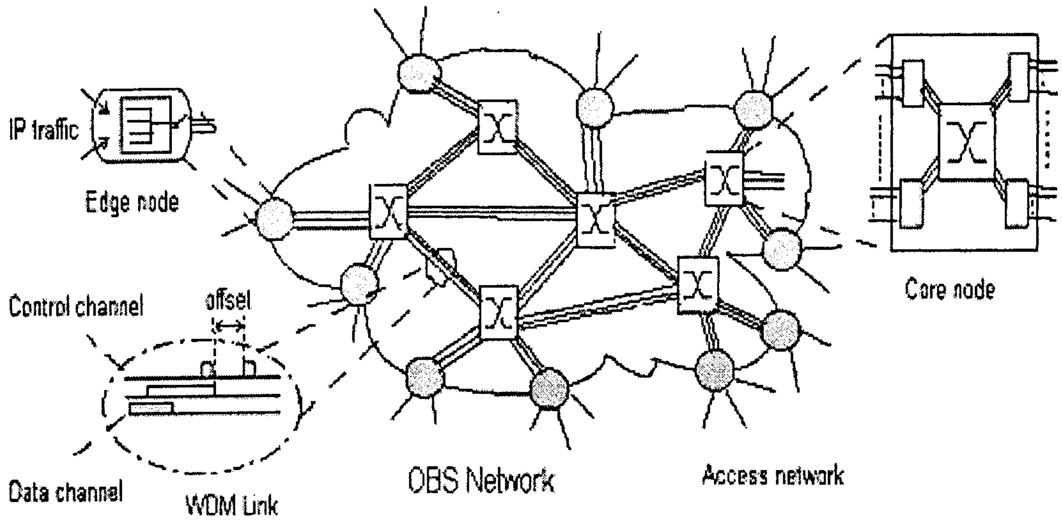


Figure 2-3. An overview of an optical burst switched network

In an OBS network an offset time is included between the data burst and its corresponding control packet. The function of this offset time depends on the design of core routers. For the core routers, which use Fiber Delay Line (FDL) buffers to delay the data bursts, the main function of the offset time is to resolve the contentions between the control packets. On the other hand, for the core routers which do not use FDL buffers, the function of the offset time is to provide sufficient time to the switch control unit (SCU) to process the control packet before the burst arrives. The offset time can also be adjusted to provide different levels of Quality of Service (QoS) [17]. Another important feature of an OBS network is that it uses a one-way resource reservation scheme. This one-way reservation protocol is advantageous because it eliminates the two-way propagation delay, even though the burst loss in a one-way reservation technique is high. The very fast OBS transmission speeds make it impossible to retransmit the dropped data bursts; instead, in an OBS this functionality is left to the upper network layers.

## 2.5 OBS Design issues

In this section, the key architectural and design issues that must be addressed before deploying the OBS technique in an Agile All-Photonic Network are discussed. Some examples from literature are discussed as well.

### 2.5.1 Determination of offset times at the edge

An important OBS network design issue is to determine an appropriate offset time between the burst header and its associated data burst. The value of the offset time should be large enough to allow the optical cross-connects to process the control packet completely before its associated data burst arrives. Three different schemes that can be considered for estimating the required offset time in an OBS-AAPN are reviewed below.

#### 2.5.1.1 No Offset

In this technique, both the control packet and the data burst are sent at the same time on two different wavelength channels. No additional complexity is introduced at the edge nodes to delay the burst and no calculation of the header processing time is required to send the control packet in advance. Of course, the header processing time must be compensated, which is done by delaying the data burst at the intermediate nodes using fibre delay lines (FDLs). However, using FDLs may limit the flexibility of OBS since their length (and therefore the delay they can provide) is limited.

#### 2.5.1.2 Fixed Offset time

A fixed offset time scheme called *Just Enough Time* (JET) is proposed in [4] where the offset time is estimated to be the sum of the total processing time and switch fabric configuration time at all the intermediate nodes. This scheme, however, has some major drawbacks:

- *First,* this scheme requires knowledge of the precise number of hops from source to destination.
- *Second,* in this scheme the processing and switch configuration time at each OBS node are assumed to be approximately the same, but in practice these times may vary from node to node because of possible queuing delays in the control channel.
- *Third,* this scheme is not robust in a distributed environment due to contention at intermediate nodes among bursts arriving from geographically dispersed sources.

#### 2.5.1.3 Statistical Offset time

Another offset setting technique that can be considered for optical burst switched Agile All-Photonic Network is the statistical offset time scheme [14]. In this scheme a leaky bucket regulator is used to generate transmission tokens based on a Poisson process with a predefined arrival rate. No buffering is provided for the tokens. As soon as a burst reaches the head of the burst queue, its corresponding control packet is sent toward the destination and the burst itself is delayed until it is able to grab the next available transmission token in a first come first serve basis. However, if there is a burst in the queue waiting for transmission but the token arrives before the completion of transmission of the previous data burst, then the token cannot be grabbed and the burst will have to wait for the generation of the next token. Also, if no new data burst is available in the queue then the arriving token will be lost. This scheme has the following desirable properties:

- It regulates the average data burst rate;
- It prevents synchronization among the token streams since the tokens are generated randomly, which consequently reduces the burst blocking probability.

### 2.5.2 Resource reservation schemes

Two significant components of OBS networks are the signaling and reservation schemes.

The design objectives of OBS signaling can be summarized as follows:

- The protocol should support unidirectional point-to-point data transmission.
- The protocol should support different classes of data-burst in the optical domain. Retransmission due to blocking is allowed but must be as rare as possible for critical data-burst traffic.
- The protocol should support packet-switched type services (or bursty transmission), as well as conventional circuit-switched services. The exact mode of operation (the type and class of service) should be explicitly defined by the sender.
- The reservation scheme should be able to assign different priority levels to data-burst with different QoS requirements.

Several reservation mechanisms have been proposed in OBS literatures. The proposed mechanisms can be differentiated based on the time when the allocation of the WDM channel starts and their way of indicating the end of transmission of a burst. In the following, four different types of reservation techniques that can be considered for AAPN-OBS are discussed.

#### 2.5.2.1 Tell and Go (TAG)

Tell and Go (TAG) is a simple reservation approach where the end of a burst is indicated by sending an additional control packet or using an in band terminator (IBT). In both cases, the control packet containing the reservation request does not carry any burst length information. The Just-in-Time (JIT) signaling protocol proposed by Wei et al. [15] is an example of a TAG based scheme. In JIT, a wavelength channel is allocated immediately after the arrival of the reservation request. The burst is sent after the offset time and the wavelength remains allocated until another control packet for releasing the connection arrives. If no wavelength is available then the request is rejected and consequently the associated burst is dropped. The scheme is described in Figure 2-4.

In JIT approach, the only information needed in network nodes is whether a wavelength is currently in use or not, which makes it a simple approach with lower complexity in both edge and core nodes. However, the simplicity of JIT comes at the cost of high packet loss rate due to the poor resource occupation schemes, which can be unacceptable for certain applications. A burst can be dropped even without any transmission conflict between different bursts on the same wavelength. This is illustrated in Figure 2-5, where Burst2 is discarded since the end of the previously accepted burst (Burst1) is unknown at the instant when Burst2's control packet arrives, even though the start time of Burst2 occurs later than the stop time of Burst1.

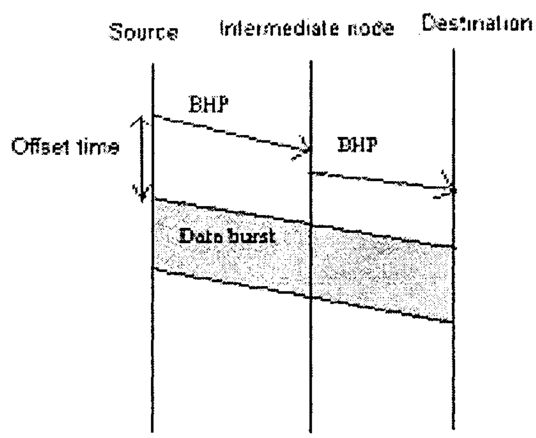


Figure 2-4. JIT reservation scheme

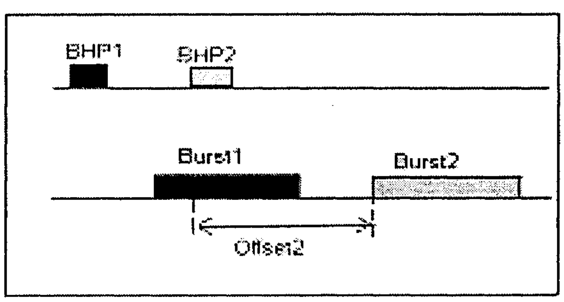


Figure 2-5. A JIT reservation scenario that results in unnecessary packet loss

#### 2.5.2.2 Reserve a limited duration (RLD)

An improvement to JIT can be made by reserving the resources for a limited duration (RLD). Mechanisms based on such a policy need to send the information about burst length in the control packet carrying the reservation request. Upon the arrival of the reservation request, a wavelength channel is reserved only for a limited duration to allow subsequent transmission requests having a start time larger than the stop time of an already scheduled burst. The *Horizon* mechanism developed by Turner [5] is an example of RLD-based scheme. In *Horizon*, the network performance is enhanced by maintaining a single scheduling horizon for each channel on the link. The scheduling horizon for a channel is defined by the time until which the wavelength is allocated. When a new request arrives then the wavelength channel that precedes the burst arrival with the latest scheduling horizon is allocated for the upcoming burst. Once the channel is reserved, the scheduling horizon is recomputed to be equal to the time when the transmission of the burst is expected to be finished (based on the information about the burst length). While the horizon based scheduling algorithm aims to minimize the gap between successive bursts, it can waste resources due to its failure to schedule short bursts between those gaps.

#### 2.5.2.3 Reserve a fixed duration (RFD)

A higher efficiency can be achieved by considering both the start and stop time of accepted bursts during reservation. Mechanisms based on such phenomenon are called RFD (reserve a fixed duration). In RFD schemes, channels are reserved only for a fixed period of time corresponding to the burst transmission time. One example of an RFD based technique is Just Enough Time (JET), proposed by Qiao and Yoo [4]. JET relies on setup and release time estimation for each data-burst; that is, an exact time period for reserving wavelength channels along a working path is calculated at the source node, and then assigned to each intermediate node by a control packet. A functional overview of the JET scheme is shown in Figure 2-6. Here the source node sends a control packet, which is followed by a burst after an offset time T. To find out what will be the suitable value of the offset time T let us assume that δ is the total delay encountered by the control packet. Hence, $\delta = H * D$ where, $D$ is the time to process the control packet at each intermediate node (the reception and transmission time of the control packet is ignored) and H is the total number of hops between source and destination. The burst is buffered at the source for the control packet to be processed at each intermediate node. Accordingly; the minimum latency of the burst is $\delta$, where $\delta = H * D$.

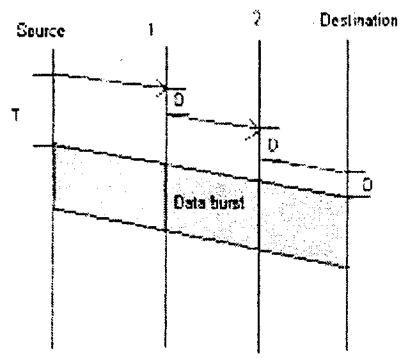


Figure 2-6. JET reservation scheme

The choice of offset time $T = \delta$ ensures that there is enough headroom for each node to complete the processing of the control packet and, simultaneously, minimizes the latency. Unlike JIT, with this scheme no control packet is sent for releasing the network resources. On the other hand, and in contrast to Horizon, the JET-signaling protocol is capable of scheduling a burst in between two already reserved bursts. Hence, the burst acceptance probability is higher compared to JIT and Horizon especially when the offset time is large. However, in JET it is required to keep information about both the start and stop time of all the accepted bursts, which makes it a rather complex system.

#### 2.5.2.4 Tell and Wait (TAW)

The packet loss probability of an OBS-AAPN can be further reduced by adopting the Tell and Wait reservation approach. In this approach, a control packet containing the reservation request is sent along the burst's route to gather all the wavelength availability information at every node in the path. As a response, a channel assignment algorithm is executed at the destination node to determine the reservation period on each link based on the earliest available channel periods of all the intermediate nodes and a reply is sent in backward direction, which then tries to reserve the wavelength at each intermediate node for the appropriate transmission period. If at any node, the required wavelength is already occupied then a packet containing the fail message is sent to the destination to release the previously reserved resources along the path. The burst is sent to the destination at the scheduled time only if the reply message can reach the source node successfully. The schematic diagram of this approach is shown in Figure 2-7.

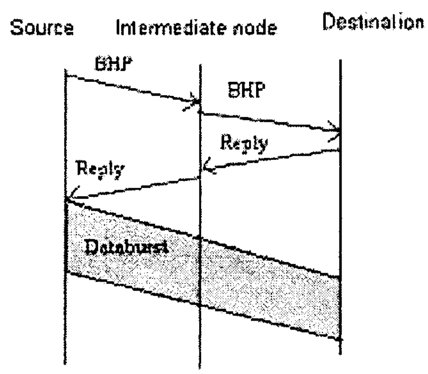


Figure 2-7. Tell and Wait reservation approach

Tell and Wait performs well in terms of packet loss probability due to the use of a channel reservation acknowledgement. However, it incurs another kind of delay: the round trip set up time. Hence, TAW is suitable for delay insensitive traffic. On the other hand, JET suffers from high packet loss compared to TAW but the end-to-end delay is less, since it follows a one-way reservation process. Therefore JET is instead more suitable for traffic that is less sensitive to packet loss.

### 2.5.3 Wavelength Allocation: with or without conversion

There are two possible ways to allocate wavelengths in an optical burst switched network. The first approach is called a wavelength continuity allocation scheme, where a single wavelength is used for the transmission of the data burst along the whole path. The probability of getting a specific wavelength free on all the fiber links is very low, so this scheme results in a high burst loss. The second approach assumes that every OBS node has the capability of wavelength conversion. With this scheme, if two bursts contend for the same wavelength channel then the OBS node will convert one wavelength into the other. However, the photonic core of an AAPN does not allow any wavelength conversion and hence, this technique can be ruled out in the context of AAPN.

An alternative scheme, Wavelength Routed Optical Burst Switched network (WR-OBS), proposed by M. Duser and P. Bayvel [18], employs a centralized *first –fit wavelength-continuity allocation* mechanism where the wavelengths are used as routing labels. The WR-OBS centralized scheduler has the knowledge about the whole network and it can search for a wavelength that is free over the entire path. At a certain point of the aggregation process, a request is sent from the edge node for transmission of the burst. After receiving an acknowledgement, the burst is transmitted over a specific wavelength channel assigned by the centralized scheduler. This scheme requires a two-way reservation process, which makes it more suitable for relatively small networks (metropolitan or local network). If the size of the network is large then the round trip propagation delay might increase the End-To-End packet delay significantly.

### 2.5.4 QoS support

The first proposals on Optical Burst Switching considered only a single class of burst [5] [15]. However, in recent literatures the OBS concept has been extended to support multiple classes of traffic. An offset time based QoS scheme proposed by Yoo, Qiao [17] can be used to provide different classes of services (in terms of loss rate) in an OBS-AAPN. In an optical burst switched network, the delay of each data burst consists of a fixed component and a variable component. The fixed component is equal to the propagation delay, which is inevitable in a network. On the other hand, the variable delay is the sum of aggregation delays at the OBS edge nodes and the pre-transmission offset time. Qiao and Yoo took advantage of this pre-transmission offset time to support multiple service classes. In this case, the offset time of a data burst is comprised of a *basic offset* and a *QoS offset*. The *basic offset* represents the sum of processing times of the control packet at all intermediate nodes and the *QoS offset* is specific to each service class. For higher priority traffic a longer *QoS offset* time is assigned. In [17] the authors gave examples to explain how the longer offset can increase the chances for the control packet to reserve the required resources at the OBS core nodes. However, in this scheme the longer offset increases the end-to-end latency of the packets, which can affect the performance of multimedia applications because of their stringent delay requirement.

# 3 Burst Aggregation

## 3.1 Introduction

The burst aggregation algorithm impacts network performance by shaping the traffic at the edge nodes; hence its design is important in an OBS network. Usually, the *edge routers* perform the burst aggregation/deaggregation function. In an assembly unit (shown in Figure 3-1), IP packets are queued and concatenated in an *aggregation buffer* to form a burst and each burst is labeled by a burst control packet (BCP), which contains information about, amongst other things, the source, destination and quality of service.

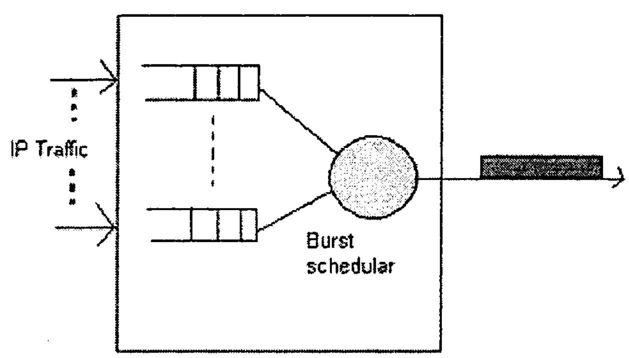


Figure 3-1. Block diagram of a burst assembly unit

Data burst assembly involves decisions on two key parameters: the maximum waiting time in the aggregation buffer (*time- out*) and the maximum length of the burst (*threshold value*). Based on these two parameters, the burst assembly mechanism can be *timer-based* or *threshold-based* or *hybrid*, a combination of two. Recent research on burst aggregation algorithms has mainly focused on its effect on the self-similar characteristics of the traffic [20] [21] [22] [23]. Self-similarity indicates the long-range dependence of the network traffic that is manifest in particular by IP traffic. Initially it was shown that the long-range dependence of Internet traffic can be reduced by adopting burst aggregation in the ingress node [20]. However, based on simulation in [21] it was demonstrated later that the long-range dependence in the input traffic will not change after aggregation using *hybrid* based burst assembly algorithm. The performance of different types of burst aggregation algorithms was also evaluated under the TCP congestion control mechanism. It was claimed that TCP performance is more sensitive to the assembly period than the burst length, and hence ideally, the assembly period should adapt to the TCP's sending window size [22]. However, as far as we are concerned no prior work has analysed the probability distribution of delays encountered by each packet in the aggregation buffer. In this chapter, an analytical model is developed for Poisson arrivals to address this open issue for different burst aggregation algorithms. The accuracy of the analytical model is verified by OpNet simulation. The impact of different burst aggregation methods and associated parameters (e.g. *threshold-value* and *time-out* value) on the blocking probability of an OBS-AAPN is also observed.

This chapter is organized as follows: section 3.2 and section 3.3 describe the *timer-based* approach and *threshold-based* approach respectively. Section 3.4 discusses the *hybrid-based* approach. Section 3.5 presents an analytical model developed to evaluate different burst assembly algorithms for Poisson statistics. Section 3.6 shows the analytical results. Section 3.7 presents the simulation results and verifies the accuracy of the analytical model by OpNet simulation and finally, section 3.8 summarises the chapter and draws some interim conclusions

## 3.2 Timer-Based Approach

A simple and intuitive burst aggregation method is the timer-based approach; where a time out interval $T_{out}$ is scheduled upon the arrival of the first packet to an empty queue. A burst consisting of all packets in the aggregation queue is sent out as soon as the time-out occurs. Several variations of the timer based aggregation algorithm are proposed in the OBS literature. One extreme is to use the same fixed assembly time out value for all the edge nodes and another extreme is to use different but fixed time out values for different aggregation buffers at each ingress node [22]. In our simulation study, it is assumed that all the edge nodes use the same value of the assembly time out parameter.

The pseudo code developed for the timer-based algorithm is given below, where $q_{sd}$ represents the aggregation queue at ingress node $s$ for destination $d$ and $T_{out}$ is assembly time out period.

**Event: Packet arrives at node s**

$d$ = destination address of the incoming packet;

**If** ( $q_{sd}$ is empty) **then**

Schedule a time out interrupt for $q_{sd}$ ;

**End;**

append the packet to the corresponding queue $q_{sd}$ ;

**Event: Assembly time out (** $T_{out}$ **)**

Generate a control packet for this burst;

Insert all the information into the control packet and send it on a control channel;

Schedule the data burst to be sent out on a data channel after an offset time;

This scheme suffers from several drawbacks. First, in this scheme, the value of *Tout* should be set carefully. If the value is too small then too many small bursts will be created at the edge nodes and consequently the control overhead will be too high. On the other hand, if the value is too large then the end-to-end packet delay might be intolerable. Second, this scheme might also produce undesirable burst lengths at different loads.

## 3.3 Threshold-based Approach

In a *threshold-based* (also termed as *length-out*) burst aggregation approach, a burst is created and sent out when the size of the aggregation queue reaches or exceeds a certain threshold value of $B_{\max}$ in bits.

The following pseudo code provides a more detailed description of the threshold-based algorithm, where $q_{sd}$ represents the aggregation queue at ingress node $s$ for destination $d$ and $B_{\max}$ is the threshold value in bits.

**Event: Packet arrives at node s**

$d$ = destination address of the incoming packet;

Insert the packet into the corresponding queue $q_{sd}$ ;

**If** (length of $q_{sd} \geq B_{\max}$ ) **then**

Generate a control packet for this burst;

Insert all the information into the control packet and send it to the destination $d$ ;

Schedule the data burst to be sent out after an offset time;

**End**;

The major drawback of this approach is this scheme does not give any guarantee on the assembly delay that the packets will experience during the burst aggregation process.

## 3.4 Hybrid Approach

In order to address the deficiency associated with each type of burst assembly mechanism mentioned above both the timer-based and threshold-based approach can be combined into a single burst assembly mechanism named the *hybrid approach.* To perform all the functions properly a hybrid burst assembly algorithm can work as follows:

- A logical queue is associated with each destination at the edges of the burst switch network.
- A timer is started whenever a packet arrives in an empty queue.
- A burst is created and queued for transmission when either the elapsed time $T_{out}$ equals burst assembly time-out (which could be adaptive to the traffic conditions) or the length of the queue reaches or exceeds the maximum data burst length $B_{\max}$ .
- After queuing the burst for transmission, the timer is reset to 0 until it is restarted by the arrival of the next packet.

The pseudo code developed for the hybrid approach is given below:

**Event: Packet arrives at node s**

$d$ = destination address of the incoming packet;

**If** ( $q_{sd}$ is empty) **then**

Schedule a time out interrupt for $q_{sd}$ after $T_{out}$ ;

**End;**

Insert the packet to corresponding queue $q_{sd}$ ;

Update the new burst length information;

**If** (length of $q_{sd} \geq B_{\max}$ ) **then**

Generate a control packet for this burst;

Insert all the information into the control packet and send it to the proper destination $d$ ;

Schedule the data burst to be sent out after an offset time;

Cancel the time out interrupt scheduled for $q_{sd}$ ;

**End**;

**Event: Aggregation time out**

Generate a control packet for this burst;

Insert all the information into the control packet and send it to the destination address $d$ ;

Schedule the data burst to be sent out after an offset time;

Note that, in a hybrid approach if the value of $T_{out}$ is set too small, i.e. $\lambda T_{out} << B_{\max}$ (where, $\lambda$ is the mean arrival rate) then the length of the assembled burst is unlikely to reach $B_{\max}$ . In such a case this approach will work in the same way as the *timer based* approach works. On the other hand, the *threshold based* approach can be treated as a special case of *hybrid based* approach when $\lambda T_{out} >> B_{\max}$ . Usually, the increase in $B_{\max}$ and $T_{out}$ will increase the end-to-end latency for the packets. How to choose the maximum burst size and the maximum aggregation time out is still an open issue in an OBS network

## 3.5 Analytical model for evaluating the burst aggregation method for Poisson arrivals

Let:

| | |
|---|---|
| $\lambda$ | Mean arrival rate |
| $B_{\max} = N+1$ | Maximum burst size in packets (assumed fixed size) |
| $T > 0$ | Time-out |
| $n = 0,\ldots N$ | Location of packet within a burst |
| $t_n$ | Time packet at location $n$ arrives |
| $\tau_n$ | Time packet at location $n$ waits before the burst departs |

Under Poisson statistics a single event can occur in a sufficiently small time interval $\delta t$ with probability $\lambda\delta t$. Suppose an event occurs at time $t = 0$ then the probability that the next event occurs at time $t = t_1$ is:

$$p(t_1 = t)\delta t = (1-\lambda\delta t)^{(M-1)}\lambda\delta t \quad , \quad \delta t = \frac{t}{M}$$

$$\Rightarrow$$

$$p(t_1 = t) = \lambda\exp(-\lambda t) \quad , \quad M \to \infty$$

The probability that a further event occurs at time $t = t_2$ is:

$$\begin{aligned} p(t_2 = t) &= \int_0^t p(t_1 = t-s)p(t_1 = s)ds \\ &= \int_0^t \lambda\exp[-\lambda(t-s)]\lambda\exp[-\lambda s]ds \\ &= \lambda\lambda\exp[-\lambda t]\int_0^t ds \\ &= \lambda(\lambda t)\exp[-\lambda t] \end{aligned}$$

By induction, the *n-event inter-arrival* time probability distribution is:

$$p(t_n = t) = \lambda\frac{(\lambda t)^{(n-1)}}{(n-1)!}\exp[-\lambda t] \quad , \quad n = 1,2\ldots,$$
$$t \in [0,\infty]$$

Define:

$$p(t_0 = t) = \delta(t)$$

Every burst aggregation is initiated by a packet arrival at a time that may be taken as $t = 0$ without loss of generality. The aggregation process is completed either by a time-out at time $T$ if $t_N > T$ or otherwise by the arrival of a packet at location $N$ that creates a burst of maximal size $B = B_{\max} = N + 1$, a process referred to here as a length-out. A time-out will result in a burst of length $B < B_{\max}$ if $t_{B-1} \leq T < t_B$ which implies $t_N > T$ through the time ordering $0 < t_1 < t_2 < ... < t_N$.

Suppose packets are labeled by $\alpha = (i, j)$ where $i$ is the location the packet occupied within the burst $j$ that transported that packet. The probability $q_n = P(i = n \mid j = m)$ that location $n$ in burst $m$ is occupied before time-out is:

$$q_n = \int_0^T p(t_n = t)dt \quad , \quad n = 1,...N$$

Note:

$$\begin{aligned}
q_n &= \int_0^T p(t_n = t)dt \\
&= 1 \quad , \quad n = 0 \\
&= \lambda \int_0^T \frac{(\lambda t)^{(n-1)}}{(n-1)!} \exp(-\lambda t)dt \quad , \quad n = 1,2,... \\
&= 1 - \exp(-\lambda T) \quad , \quad n = 1 \\
&= \left[ -\frac{(\lambda t)^{(n-1)}}{(n-1)!} \exp(-\lambda t) \right]_0^T + \lambda \int_0^T \frac{(\lambda t)^{(n-2)}}{(n-2)!} \exp(-\lambda t)dt \\
&= -\frac{(\lambda T)^{(n-1)}}{(n-1)!} \exp(-\lambda T) + I_{n-1} \quad , \quad n > 1 \\
&\Rightarrow \\
q_n &= 1 - \sum_{k=0}^{n-1} \frac{(\lambda T)^k}{k!} \exp(-\lambda T) \quad , \quad n = 1,2,...
\end{aligned}$$

and hence:

$$q_0 = 1$$
$$q_n = 1 - \sum_{k=0}^{n-1} \frac{(\lambda T)^k}{k!} \exp(-\lambda T) \quad , \quad n = 1,\ldots N$$

which can be recognized as the probability that there are at least $n$ events in the interval $[0, T]$ for the Poisson counting process.

The probability distribution $p(t_n = t | i = n)$ of the arrival time $t_n$ of a packet transported in location $n$ of the burst is:

$$p(t_0 = t | i = 0) = \delta(t)/q_0$$
$$p(t_n = t | i = n) = p(t_n = t)/q_n \quad , \quad n = 1,\ldots N$$
$$t \in [0, T]$$

Note that this distribution satisfies the requirement that transported packets have arrival times restricted to the interval $[0, T]$ and that a transported packet carries some arrival time within this interval with unity probability.

The expected number of packets transported per burst is:

$$\sum_{k=0}^{N} q_k$$

of which $q_n$ are transported in location $n$. Therefore the probability $p_n$ that a packet chosen at random was transported in location $n$ is:

$$p_n = q_n \Big/ \sum_{k=0}^{N} q_k$$
$$\rightarrow 1/(N+1) \quad , \quad \lambda T >> B\max$$

The delay $\tau_n = T - t_n$ experienced by a packet transported in location $n$ under the *time-out* mechanism is therefore distributed as:

$$\hat{p}(\tau_n = t) = \frac{1}{q_n} p(t_n = T - t) \quad , \quad n = 0,\ldots N-1$$
$$t \in [0, T]$$

The delay $\tau_n = t_N - t_n$ experienced by a packet transported in location $n$ under the *threshold-based* mechanism is distributed as the $N - n$ event inter-arrival time conditioned on $\tau_n \in [0, T]$ and is given by:

$$\tilde{p}(\tau_n = t) = \frac{1}{q_{N-n}} p(t_{N-n} = t) \quad , \quad n = 0,...N-1$$
$$\tilde{p}(\tau_N = t) = \delta(t)$$
$$t \in [0,T]$$

Combining both the *time- out* and *threshold based* (or *length-out*) mechanisms yields the unconditional probability distribution $p_n(\tau_n = t)$ of the delay experienced by a packet-transported at location $n$ given by:

$$p_n(\tau_n = t) = q_N \tilde{p}(\tau_n = t) + [1 - q_N]\hat{p}(\tau_n = t)$$
$$t \in [0,T]$$

The first term corresponds to a *length-out* that occurs with the same probability that the last location in the burst is occupied. The second term corresponds to a *time-out* that occurs with the same probability that the last location in the burst is unoccupied.

Finally, the probability distribution $pb(\tau)$ of the delay $\tau$ experienced by a packet averaged over the locations in a burst is:

$$pb(\tau) = \frac{1}{\sum_{k=0}^{N} q_k}\left[\sum_{k=0}^{N} q_k p(\tau_k = \tau)\right]$$
$$= \frac{1}{\sum_{k=0}^{N} q_k}\left[q_N \sum_{k=0}^{N} q_k \tilde{p}(\tau_k = \tau) + (1 - q_N)\sum_{k=0}^{N} q_k \hat{p}(\tau_k = \tau)\right]$$
$$= \frac{1}{\left[\sum_{k=0}^{N} q_k\right]}\left[q_N \sum_{k=0}^{N} \frac{q_k}{q_{(N-k)}} p(t_{(N-k)} = \tau) + (1 - q_N)\sum_{k=0}^{N} p(t_k = T - \tau)\right]$$
$$= \frac{1}{\left[\sum_{k=0}^{N} q_k\right]}\left[q_N \sum_{k=0}^{N} \frac{q_{(N-k)}}{q_k} p(t_k = \tau) + (1 - q_N)\sum_{k=0}^{N} p(t_k = T - \tau)\right]$$
$$\tau \in [0,T]$$

Collecting results:

$$pb(\tau) = \frac{1}{\left[\sum_{k=0}^{N} q_k\right]}\left[q_N \sum_{k=0}^{N} \frac{q_{(N-k)}}{q_k} p(t_k = \tau) + (1 - q_N)\sum_{k=0}^{N} p(t_k = T - \tau)\right]$$
$$\tau \in [0,T]$$

Where:

$$p(t_0 = t) = \delta(t)$$

$$p(t_n = t) = \lambda \frac{(\lambda t)^{(n-1)}}{(n-1)!} \exp[-\lambda t] \quad , \quad n = 1,2...,$$

And:

$$q_0 = 1$$

$$q_n = 1 - \sum_{k=0}^{n-1} \frac{(\lambda T)^k}{k!} \exp(-\lambda T) \quad , \quad n = 1,...N$$

If $\lambda T >> N$ then $q_n \to 1$, $n = 0,...,N$ and the delay distribution simplifies:

$$pb(\tau) = \frac{1}{(N+1)} \sum_{k=0}^{N} p(t_k = \tau)$$

$$= \frac{1}{(N+1)} \left[ \delta(\tau) + \lambda \sum_{k=0}^{N-1} \frac{(\lambda \tau)^k}{k!} \exp(-\lambda \tau) \right]$$

$$\tau \in [0,T]$$

But

$$\sum_{k=0}^{N-1} \frac{(\lambda \tau)^k}{k!} \approx \exp(\lambda \tau) \quad , \quad \lambda \tau < N$$

And hence the *length- out* distribution is qualitatively described by:

$$pb(\tau) \approx \frac{1}{(N+1)} [\delta(\tau) + \lambda] \quad , \quad \lambda \tau < N$$

$$\approx 0 \quad , \quad \lambda \tau > N$$

Conversely, if $\lambda T < N$ then $q_n \approx 1, n \leq \lambda T$ $q_n \approx 0, \lambda T < n \leq N$ and the delay distribution simplifies:

$$pb(\tau) = \frac{1}{\left[ \sum_{k=0}^{N} q_k \right]} \left\{ \delta(\tau - T) + \lambda \sum_{k=0}^{N-1} \frac{[\lambda (T-\tau)]^k}{k!} \exp[-\lambda (T - \tau)] \right\}$$

$$\tau \in [0,T]$$

And hence the *time- out* distribution is qualitatively described by:

$$pb(\tau) \approx \frac{1}{[1 + \lambda T]} [\delta(\tau - T) + \lambda] \quad , \quad 0 < \lambda \tau < \lambda T$$

$$\tau \in [0,T]$$

For equiprobable locations in burst, which is the case when *time-outs* are infrequent, the *time- out* distribution is the translation by T of the reflection of the *length-out* distribution about $t = 0$. This gives the bimodal distribution of the observations. The delta at zero delay due to the packets that complete a burst is mirrored by a delta at the time-out delay due to the packets that initiate burst aggregation. To compare with frequency data it is necessary to integrate the probability distributions over the sample interval $\Delta\tau$. This has the effect of transforming the numerators into unit spikes superimposed on a pedestal of height $\lambda\Delta\tau$ with the result normalized by the same denominators.

## 3.6 Numerical Evaluation

The analytical model developed in section 3.5 was numerically evaluated by Matlab code. The mean arrival rate was set to 5.0e4 packets/s. The maximum burst length was 32 (in packets) and the time out was 0.7 ms. Figure 3-2 illustrates the probability distribution of the delay experienced by each packet (averaged over burst locations) considering only the *threshold based* (or, *length-out*) approach. From Figure 3-2 it is clearly observed that for *length-out* approach, the probability distribution of the delay experienced by each packet manifest a spike at zero delay superimposed on a pedestal.

Figure 3-3 shows the probability distribution of the delay experienced by the packets in the aggregation queue with time out mechanism only. The delta, which was observed in Figure 3-2 at zero delay due to the packets that complete a burst, is reflected in Figure 3-3 by a delta at the *time-out* delay due to the packets that initiate burst aggregation. This is consistent with the theoretical expression.

Figure 3-4 shows the probability distribution of the delay experienced by each packet (averaged over burst locations) combining both the *length-out* and *time-out* mechanisms into a single *hybrid* burst assembly mechanism. For the given parameter setting, length-out mechanism was the dominant factor. Hence, only one spike at zero delay is observable in Figure 3-4. Note that for hybrid approach; usually one can see only one spike or the other depending on which mechanism is dominant in the aggregation process except for a very narrow range of parameters, where both spikes can be seen

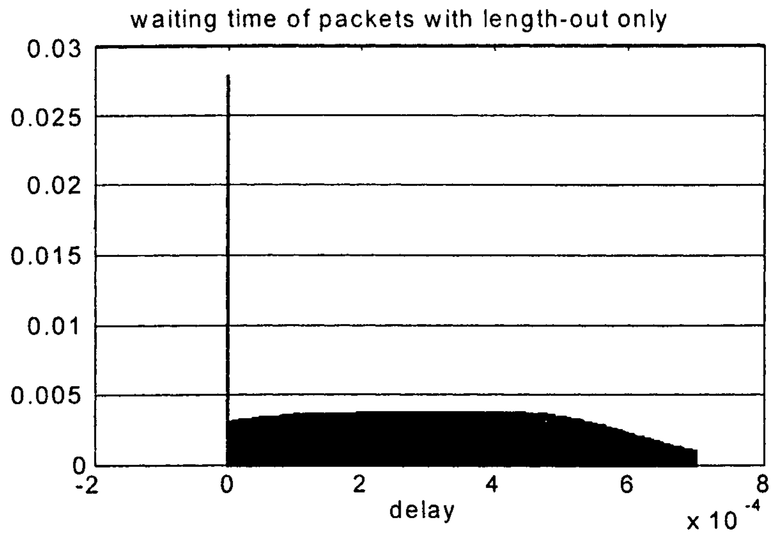


Figure 3-2. Probability distribution of the delay experienced by a packet considering the threshold based burst aggregation mechanism with a sampling interval 1.33e-5s.

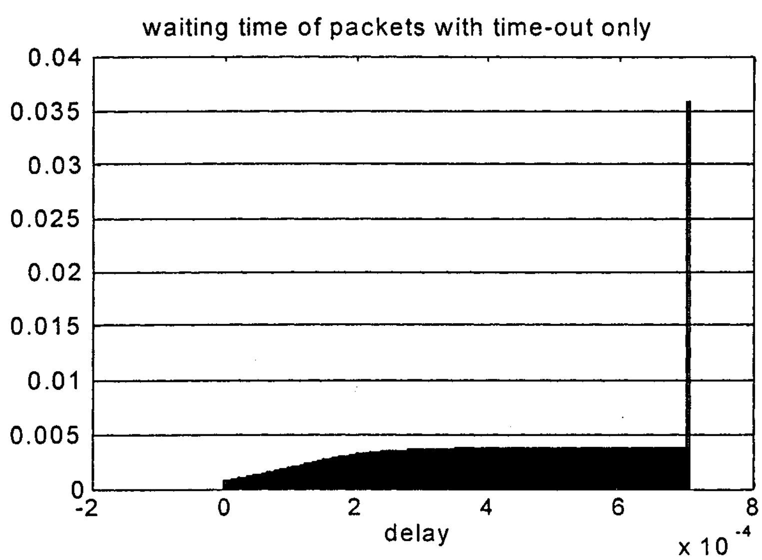


Figure 3-3. Probability distribution of the delay experienced by a packet considering only the time-out mechanism with a sampling interval 1.33e-5 s.

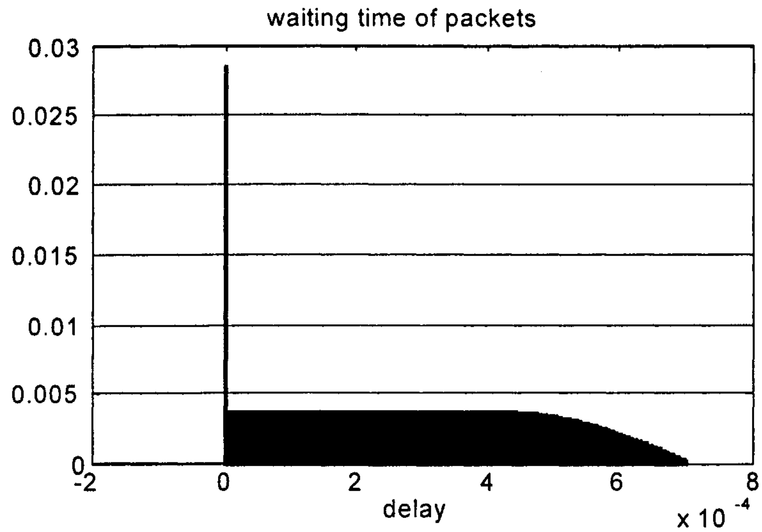


Figure 3-4. Probability distribution of the delay experienced by a packet with the hybrid burst assembly mechanisms with a sampling interval 1.33e-5 s.

## 3.7 Simulation results

In order to verify the accuracy of our analytical model and to observe the blocking probability and mean delays of different approaches to burst aggregation an optical burst switched network model was developed in OpNet Modeler. The example star network topology is shown in Figure 3-5. It consists of 8 users, 8 edge nodes and 1 core node. The network core represents the optical transport network. Switches with aggregation capabilities are used as the network edge devices. The links connecting the users to the edge nodes and those connecting the edge nodes to the core node are bi-directional and carry only one data channel with a capacity of 10 Gb/s. In this simulation, all the channels have zero bit error rates.

In OpNet, a network is made of individual nodes and each node is made of modules. The process model defines the behavior of a module. A detail discussion of the edge nodes and the core node of our network model is given in the Appendix

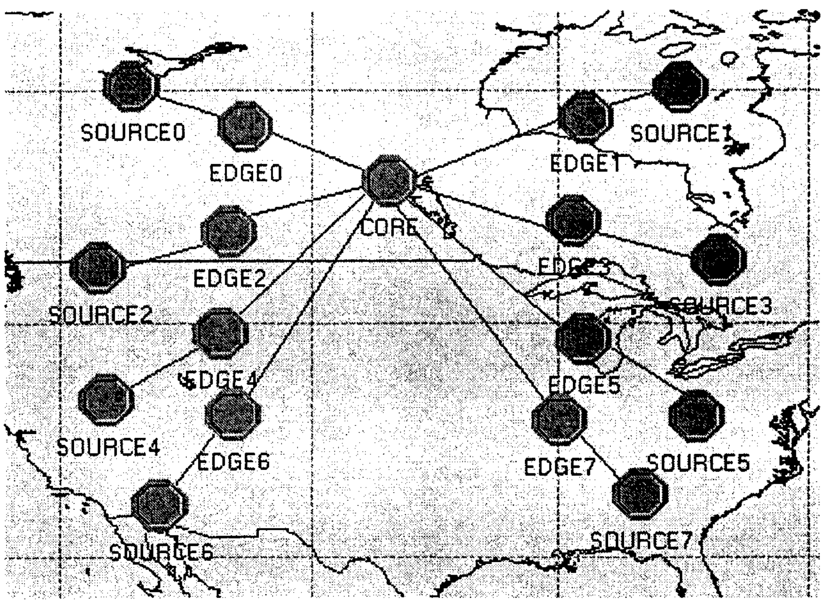

Figure 3-5. Optical Burst Switched Network model

### 3.7.1 Accuracy of the analytical model

In the first part of our simulation, the objective was to verify the accuracy of the analytical model developed in section 3.5 for Poisson arrivals. For this purpose, the mean arrival rate was set to 5.0e4 packets/s. The maximum burst length was fixed to 32 (in packets) and the time out was set to 0.7 ms. Figure 3-6 shows the probability distribution of packet delays obtained from OpNet simulation considering the threshold based burst aggregation mechanism. Figure 3-7 shows the comparison between OpNet simulation and Matlab simulation when both are overlaid on the same plot. From figure 3-7 it is observed that OpNet simulation overlaps well with the analytical result and OpNet simulation.

Figure 3-8 models the probability distribution of packet delays obtained from OpNet simulation for *time-out* mechanism. Figure 3-9 shows the comparison between OpNet simulation and numerical evaluation when both are overlaid on the same plot. It is observed that there is a discrepancy between the two results at the lower delay region. Both the theory / Matlab & OpNet were thoroughly checked but the cause of the discrepancy has not been resolved. Our best guess is that it reflects a feature of the complex data structure OpNet employs to describe packets and the way they are queued which is not transparent to the user.

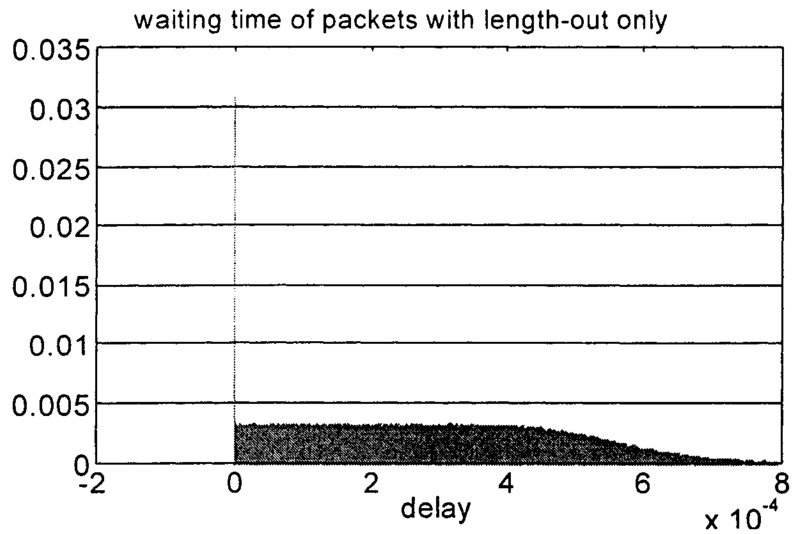


Figure 3-6. Probability distribution of the delay experienced by a packet with the threshold burst assembly mechanisms with a sampling interval 1.33e-5 s.

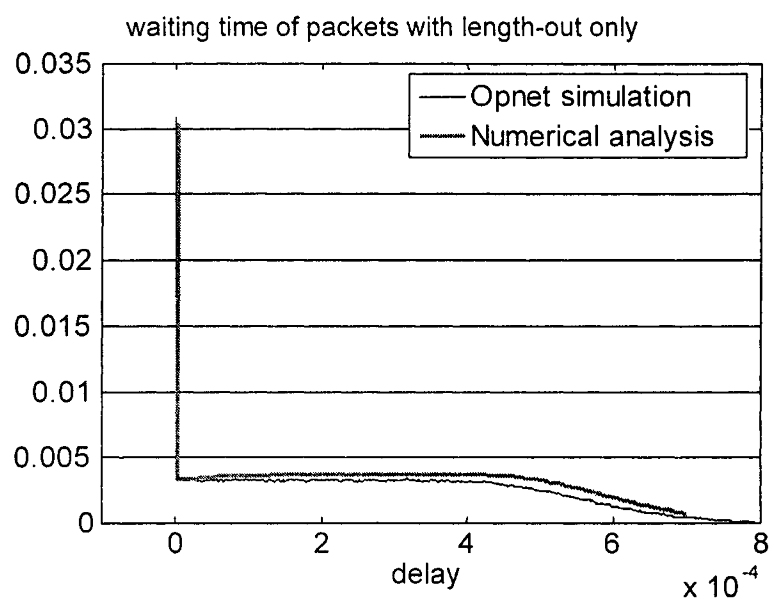


Figure 3-7. Comparison between the OpNet simulation and numerical evaluation for threshold approach

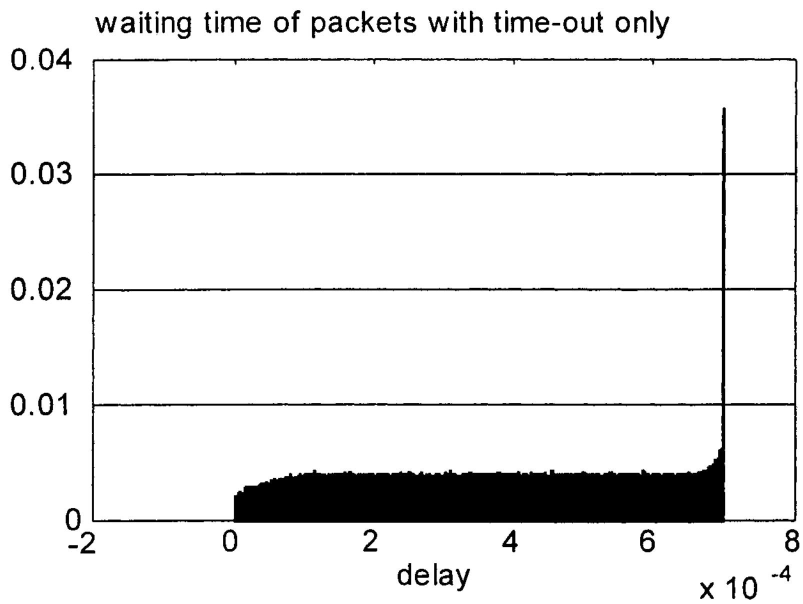


Figure 3-8. Probability distribution of the delay experienced by a packet with the timer based burst assembly mechanisms with a sampling interval 1.33e-5 s.

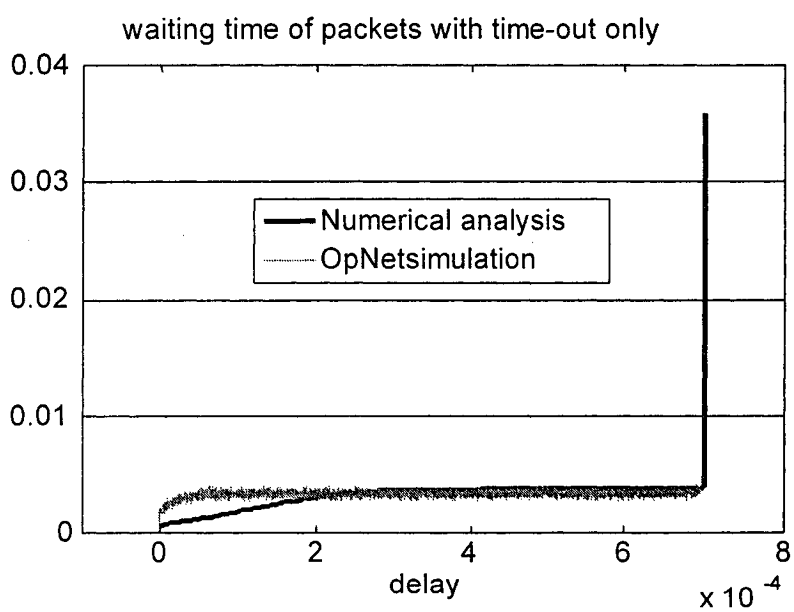


Figure 3-9. Comparison between the OpNet simulation and numerical evaluation for time-out approach

Figure 3-10 shows the probability distribution of the waiting of packets in the aggregation queue obtained from OpNet simulation for *hybrid approach* with a sampling interval 1.33e-5 s. Figure 3-11 shows a good agreement between the analytical results and OpNet simulation.

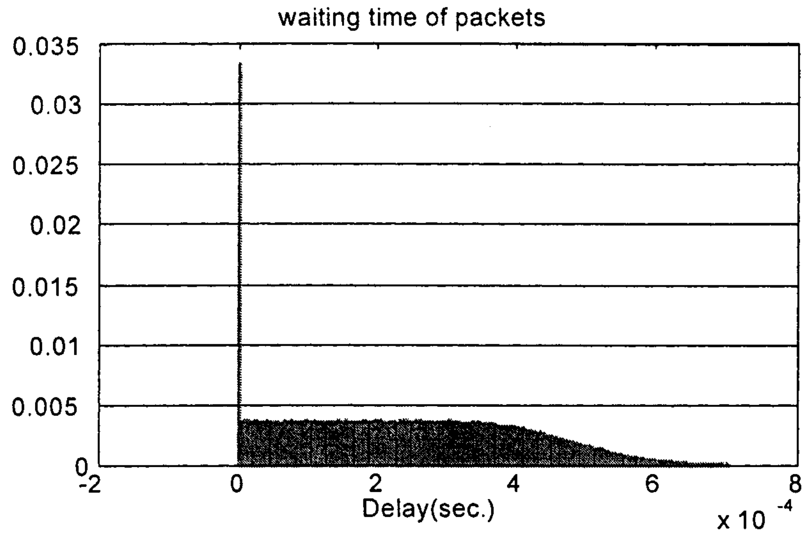


Figure 3-10. Probability distribution of the delay experienced by a packet with the hybrid burst assembly mechanisms with a sampling interval 1.33e-5 s.

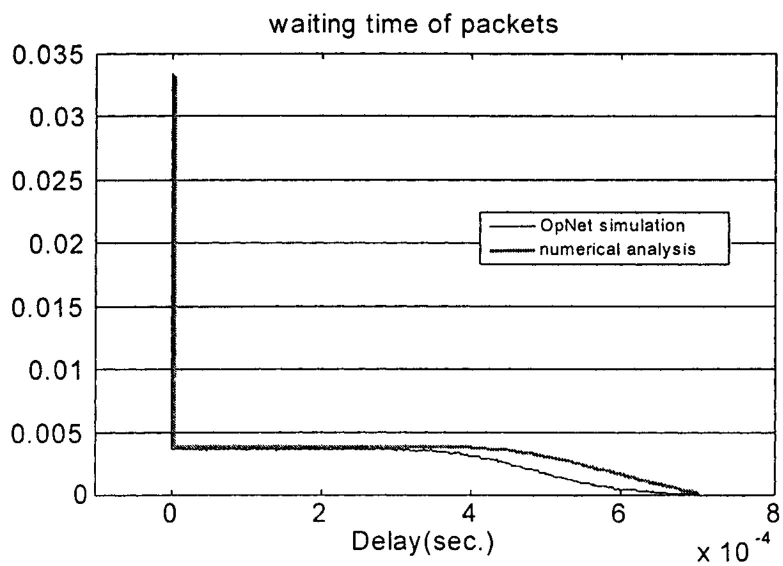


Figure 3-11. Comparison between the OpNet simulation and numerical evaluation for hybrid approach

### 3.7.2 Performance impact of different types of burst aggregation method on ETE packet delay and blocking probability

In the second part of our simulation, the objective was to study the impact of different types of burst aggregation techniques on delay characteristics and blocking probabilities. Note that, the average end to end (ETE) packet delay experienced by each packet is the sum of four terms, namely: mean aggregation delay; transmission and propagation delay; offset time; and de-aggregation delay. In our simulation, it was assumed that the de-aggregation delay is negligible compare to other delays. The offset time was equal to 1 ms. Note that, the propagation delays can't have any effect in the current version of OBS-AAPN beyond adding a constant delay to each source destination flow. Hence, the propagation delay was set to zero in this work.

The following simulations were carried out using Poisson traffic with the following parameters:

- The packet inter-arrival time (in seconds) is exponentially distributed with a mean $\mu$, where $\mu = \dfrac{L}{\rho \times R}$

  $\rho$ = Network load $(0 \le \rho \le 1)$;

  $L$ = Packet length = 12000 bits (constant);

  $R$ = Transmission speed = $10Gb/s$;
- The destination of each packet is chosen randomly with uniform probabilities, i.e., the same probability for all destinations (and the probability is $1/N = 1/8$)
- The maximum burst size, $B_{max}$ = 64 Mbit and the assembly time out, $T_{out}$ = 10 ms.

#### 3.7.2.1 Threshold based approach

Figure 3-12 illustrates different packet delay characteristics of *threshold-based* approach versus network load. The mean aggregation delay is seen to decrease with an increase in load. This can be explained by the fact that as the load increases the aggregation buffer fills more rapidly reducing the time to reach the threshold value $B_{max}$ and hence the waiting time of the packets. In the *threshold-based* approach, the burst size is fixed for fixed length packets. Hence, the transmission delay is constant (as shown in Figure 3-12). This constant transmission delay and the fixed offset time will be added to the aggregation delay to yield the total end-to-end (ETE) packet delay.

Figure 3-13 shows the impact of $B_{max}$ on ETE delay. It can be seen that a longer $B_{max}$ yields longer ETE delays which is intuitive since the packets queued in the aggregation buffer must wait for the queue length to reach $B_{max}$ and the larger the $B_{max}$ the longer the wait. Figure 3-14 demonstrates that changing the threshold value ($B_{max}$) has no effect on the packet loss probability, which is consistent with the result obtained in [24]. This can be explained by the fact that in *threshold-based* approach the only parameter of interest is the mean burst length, which is equal to the threshold value $B_{max}$. The standard deviation and other higher order moments are zero in this case. Hence, changing the value of $B_{max}$ will change the statistics linearly and thus the blocking probability will not be affected at a constant load.

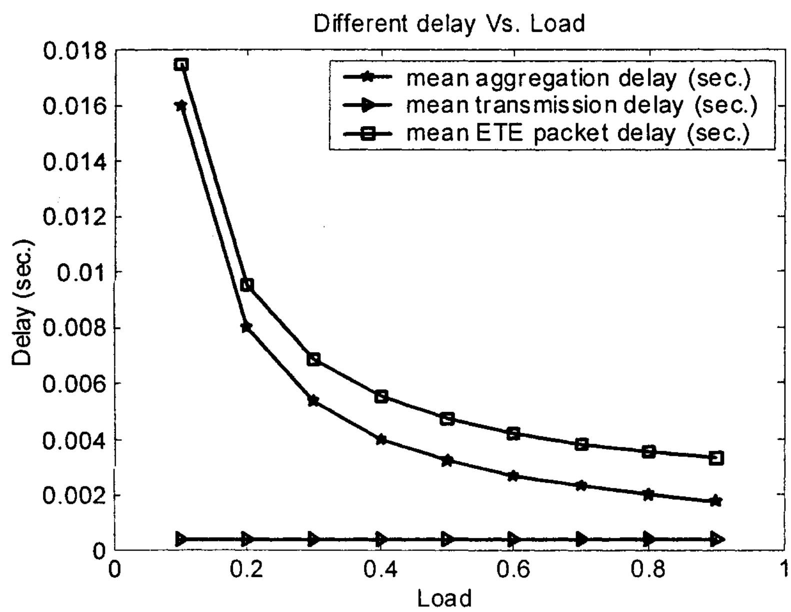


Figure 3-12. Delay characteristics: Threshold based approach under Poisson traffic scenario

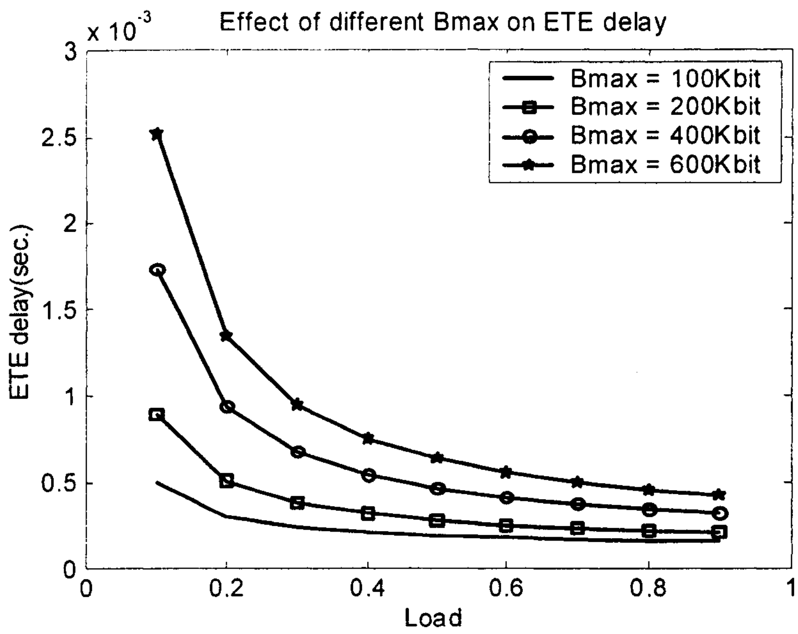


Figure 3-13. Effect of threshold value on ETE delay under Poisson traffic scenario

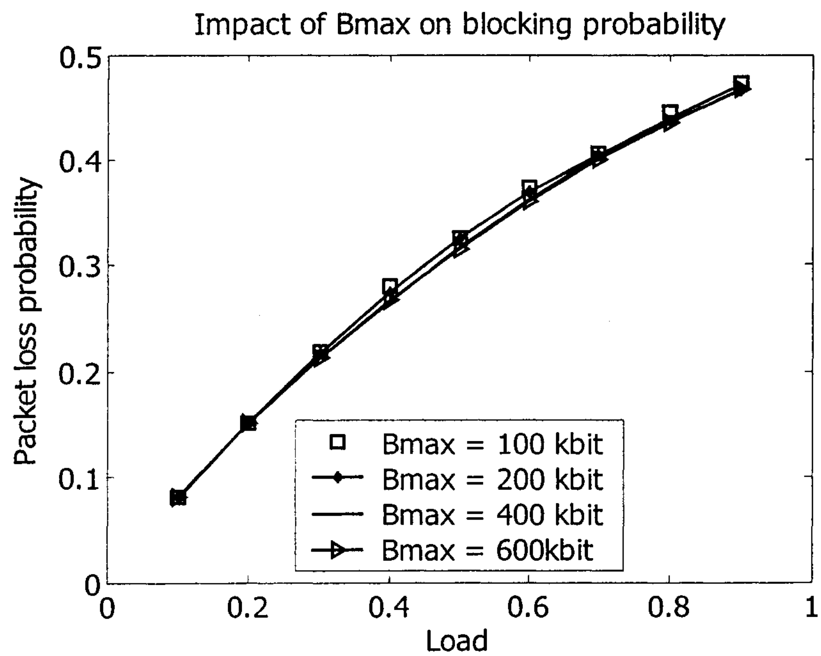


Figure 3-14. Packet loss probability: Threshold-based approach under Poisson traffic scenario

### 3.7.2.2 Timer based approach

Figure 3-15 illustrates the packet delay characteristics for the timer-based approach. It can be seen that the mean aggregation delay is constant at different loads, as expected, but the total ETE delay nevertheless increases with the increase in load. This can be explained by the fact that under high input offered load, the aggregation method generates bursts that are large, increasing the transmission delay (as shown in Figure 3-15). The impact of varying $T_{out}$ on ETE delay is trivial. Hence, this is not shown in our simulation results. However, changing the value of the assembly time out has a great impact on blocking probability. Figure 3-16 shows the blocking probability versus an increase in load for different values of $T_{out}$. In contrast to threshold-based approach, the burst length is a random variable in this case. Hence, changing $T_{out}$ will change the statistics of the output process. From Figure 3-16 it is observed that a longer time-out gives a higher packet loss probability for the same load. This can be explained by the fact that, at the same load longer $T_{out}$ will create larger bursts and thereby more packets are lost in each burst drop, which will in turn increase the packet loss probability of the network.

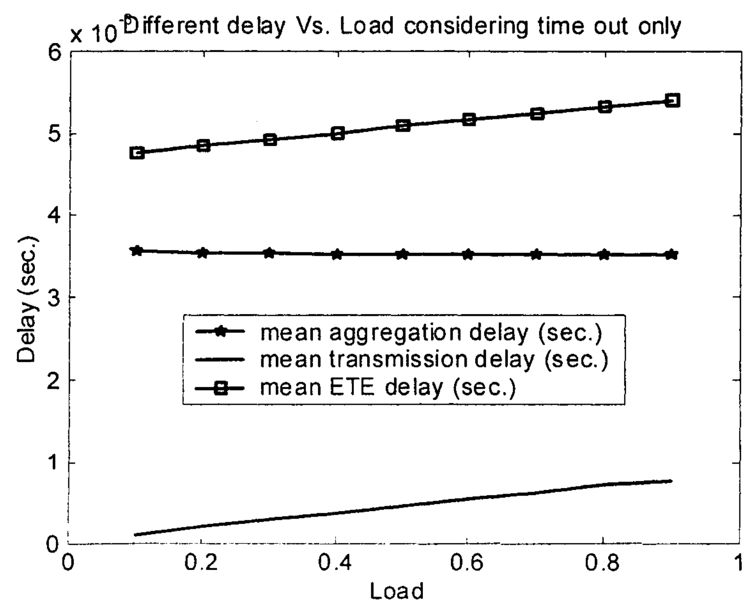


Figure 3-15. Delay characteristics: Timer based approach under Poisson traffic scenario with Tout =10 ms

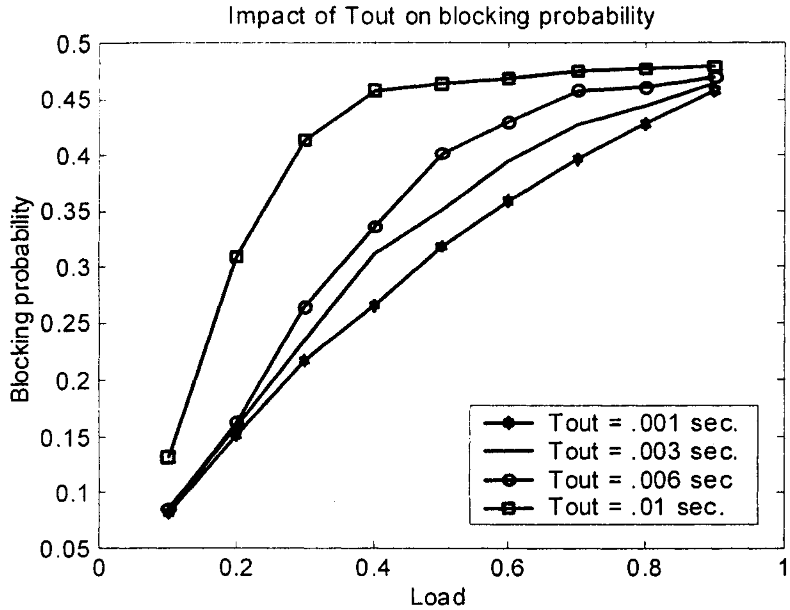


Figure 3-16. blocking probability: Timer-based approach under Poisson traffic scenario

### 3.7.2.3 Hybrid approach

Figure 3-17 demonstrates the packet delay characteristics for *hybrid approach.* It is observed that for the given parameters, the ETE delay increases with increasing load in the region of 10% to 59% load. This is because at low load, it becomes rare that the aggregation buffers will fill before the timer triggers. Hence the time out mechanism becomes the controlling factor in the aggregation process and when time out is dominant then ETE delay increases with increasing load, because the burst size increases which increases the transmission time. However, from 60% load, the buffers will fill rapidly to $B_{\max}$ in a mean time of $T_{fill} = B_{\max}/\rho * R$. Consequently, the length out mechanism becomes the dominant factor at high load and the ETE delay starts to decrease as the load increases further since when *length-out* is dominant then ETE decreases because the assembly time decreases.

Figure 3-18 shows a performance comparison for different types of burst aggregation method in terms of blocking probability. From Figure 3-18 it is observed that the blocking probability for the length-out mechanism is systematically lower than the blocking probability for the time-out only mechanism. This is because the time-out only mechanism results in variable length bursts and therefore there are more bursts generated for the same throughput compared to the case of fixed maximum length bursts. The greater number of bursts for the time-out only mechanism is directly responsible for the increased blocking probability. It is observed that in the hybrid case and for large burst maximum lengths there is a sharp transition between essentially time-out mechanism only and length-out mechanism. For the parameters given, the transition between the higher time-out only mechanism limit and the lower length-out only mechanism limit will be started at 40% load and it will be completed at 60% load i.e. when $\rho * R * T_{OUT} \approx B_{\max}$

In practice for the hybrid approach, the parameters would be set so that the transition from the time out mechanism to threshold based mechanism occurs at much lower loads and hence, its effect would not be observable on ETE delay and blocking probabilities. Examples are given later on this section

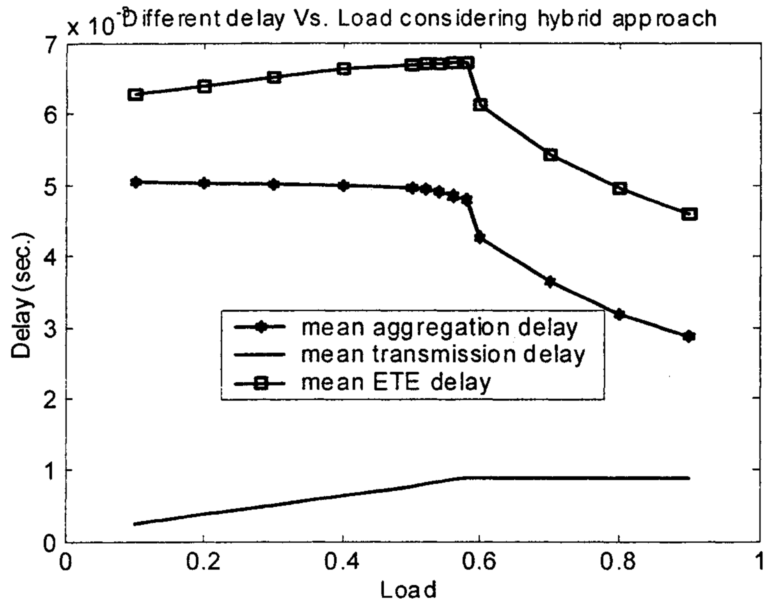


Figure 3-17. Delay characteristics: Hybrid approach under Poisson traffic scenario with Bmax =64 Mbit and Tout =10 ms

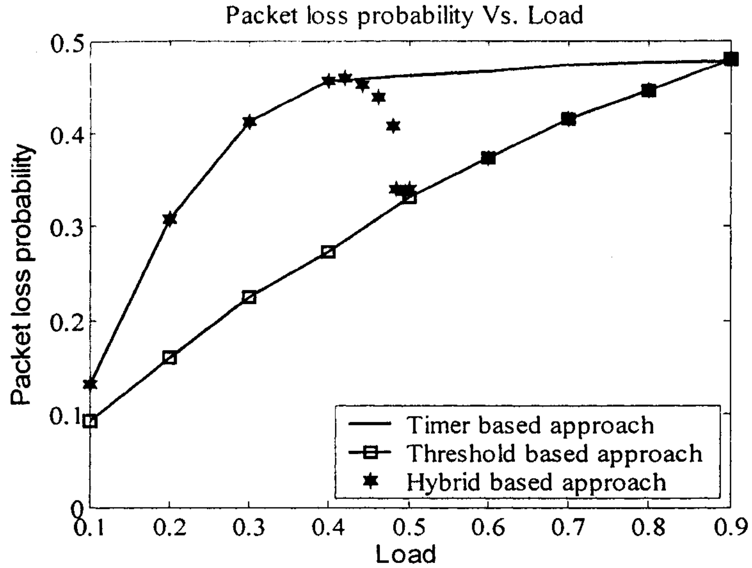


Figure 3-18. Blocking probabilities for different types of aggregation mechanisms under Poisson traffic scenario with Bmax = 64 Mbit and Tout =10 ms

The simulation results discussed so far were carried out under uniform Poisson traffic, i.e. the destinations of the packets were uniformly distributed. However, the traffic statistics of the incoming packets is likely to have a great impact on network performance especially on the queuing delay in the aggregation buffers and the packet loss ratio. To analyse this, the performance of an OBS-AAPN was evaluated using two other different traffic scenarios, non-uniform Poisson traffic and bursty traffic.

To generate a possible non-uniformly distributed traffic a bi-stochastic matrix $B = [b_{ij}]$ was used to determine the destinations probability of the packets. For each matrix, the element $b_{ij}$ is the probability that a packet generated at edge node $i$ will be destined for edge node $j$, where $i, j = 0,1,2,3............7$.

$$B = \begin{matrix} 0.4816 & 0.0072 & 0.0031 & 0.2134 & 0.0936 & 0.0000 & 0.0008 & 0.2004 \\ 0.0374 & 0.0002 & 0.1368 & 0.0349 & 0.2181 & 0.0415 & 0.4918 & 0.0393 \\ 0.0012 & 0.1370 & 0.1031 & 0.0383 & 0.0020 & 0.4980 & 0.2132 & 0.0072 \\ 0.0235 & 0.0433 & 0.2176 & 0.1361 & 0.0047 & 0.0394 & 0.0397 & 0.4956 \\ 0.0470 & 0.2504 & 0.0027 & 0.5283 & 0.1568 & 0.0076 & 0.0059 & 0.0015 \\ 0.2528 & 0.4988 & 0.0383 & 0.0000 & 0.0431 & 0.1398 & 0.0269 & 0.0002 \\ 0.1373 & 0.0205 & 0.4984 & 0.0078 & 0.0004 & 0.0404 & 0.0795 & 0.2158 \\ 0.0192 & 0.0426 & 0.0000 & 0.0412 & 0.4813 & 0.2333 & 0.1424 & 0.0400 \end{matrix}$$

The packet inter-arrival (in seconds) was exponentially distributed with a mean $\mu$, equal to $\frac{L}{\rho \times R}$, where $\rho$ = network load $(0 \le \rho \le 1)$, $L$ = packet length = 12000 bit (constant) and $R$ = transmission speed $= 10Gb/s$. The sources were independent from each other but they use all the same parameters for the random generation of packets. After generating packets they were queued in the aggregation buffers associated with their destinations. Bursts were then created according to the previously explained *hybrid approach*. In this case we set $B_{\max} = 100\text{Kbit}$ and $T_{out} = 1ms$

We expect that under non-uniform traffic, packets destined to higher probability destinations will experience smaller queuing delay and vice versa. Intuitively this can be understood by noting that aggregation buffers associated with higher probability destinations will reach its threshold value $B_{\max}$ rapidly. Figure 3-19 shows the mean queuing delay characteristics of the packets destined to different destinations from edge2. The curves behave as expected.

Figure 3-20 and Figure 3-21 demonstrates that the average queuing delay and the blocking probability for non-uniform traffic are systematically lower than that obtained from uniform traffic. This can be explained by the fact that in non-uniform traffic, aggregation buffers associated with higher probability destinations always reach $B_{\max}$ rapidly before $T_{out}$ triggers, which in turn reduces the overall probability of assembly time out in the network compared to uniformly distributed traffic. This lower probability of assembly time out is directly responsible for lowering the mean queuing delay and the blocking probability. In order to produce the same throughput the network will have to generate larger number of bursts using uniform traffic and thereby increasing the blocking probability of the network

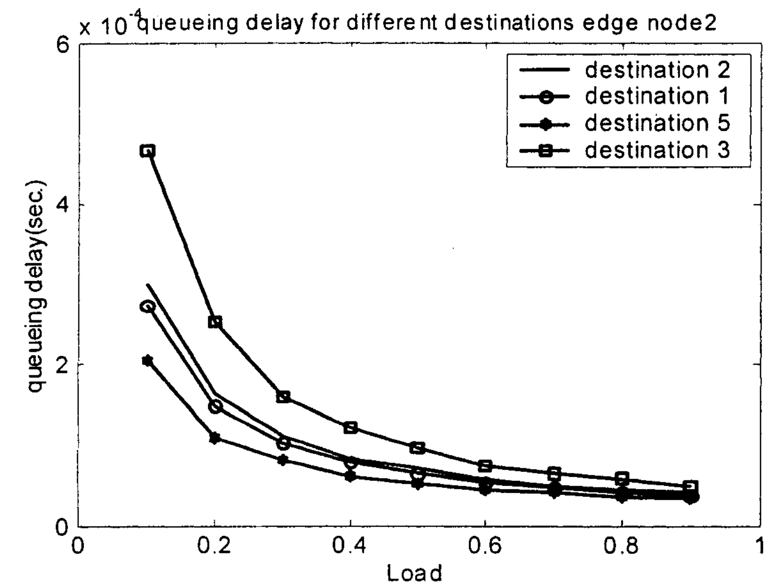


Figure 3-19. Queuing delays of the packets destined to different edge nodes under Non-uniform Poisson traffic scenario with Bmax = 100 Kbit and Tout =1ms

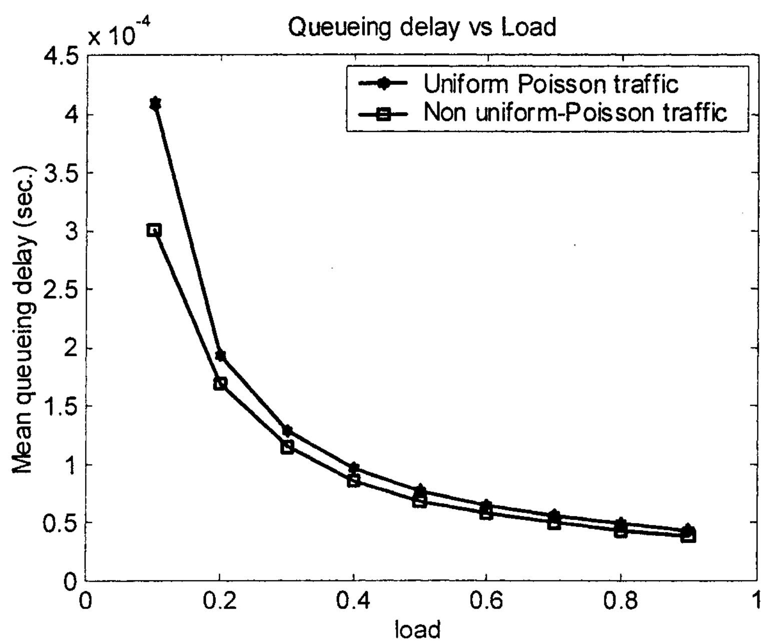


Figure 3-20. Queuing delay: Uniform and non-uniform traffic with Bmax=100 Kbit and Tout =1ms

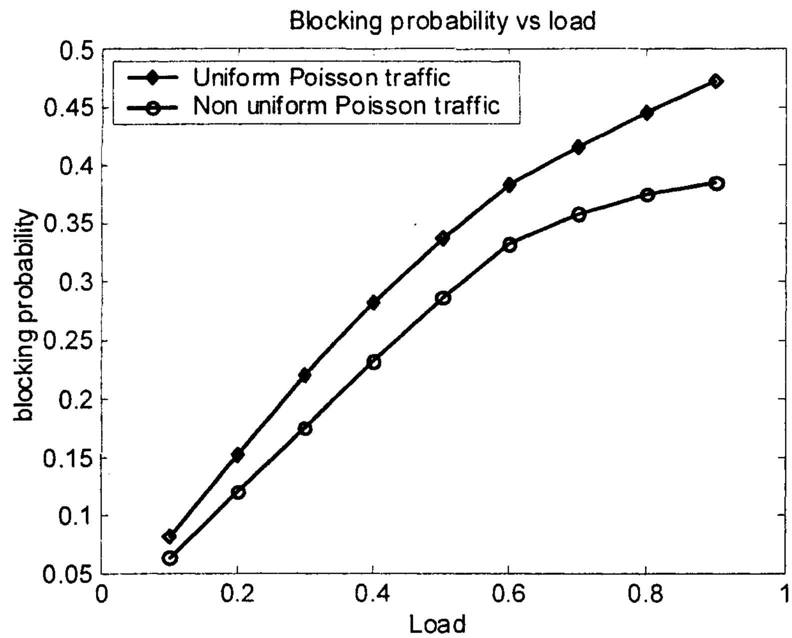


Figure 3-21. Blocking probability: uniform and non-uniform traffic with Bmax = 100 Kbit and Tout =1ms

Existing Internet traffic has fractal characteristics and can be described by self-similar stochastic processes. However, this feature is a major drawback in terms of queuing performance. Usually, the queuing performance is much worse when loaded with self-similar traffic than that obtained from short-range dependent traffic. Hence, traffic self-similarity is a major concern for the network engineers [20]

In our simulation, the performance of an OBS-AAPN was evaluated when fed by self-similar traffic. A bursty traffic model was generated by using an ON-OFF source at each edge node with independent probability density functions for the ON state, P (ON), and the OFF state, P (OFF). Both ON and OFF periods were realised from the heavy tailed Pareto distribution, given by $P(t)=\frac{(\alpha\cdot A^{\alpha})}{t^{\alpha+1}}$, where $1<\alpha<2, A>0$, $t\geq A$ and $\alpha$ indicate the heaviness of the distribution tail. If the number of sources is infinite, then this kind of traffic results in a self-similar process with the Hurst parameter $H=\frac{(3-\alpha)}{2}$ [20]. When a high but not infinite number of sources are used in simulations, the resulting traffic is an approximation of a purely self-similar traffic [20].

During the ON period, the packet inter-arrival was taken from an exponential distribution with a mean $\mu=1.2\mu s$. The packet length $L$ was 1500 bytes (constant). The load of the network was calculated from:

$$\rho = \frac{E_{mean\_ON}}{E_{mean\_OFF} + E_{mean\_OFF}} \times \frac{L}{\mu \times R}$$

where,

$$E_{mean\_ON} = \frac{\alpha_{ON} \cdot A_{ON}}{(\alpha_{ON} - 1)}$$

$$E_{mean\_OFF} = \frac{\alpha_{OFF} \cdot A_{OFF}}{(\alpha_{OFF} - 1)}$$

The final destination of each packet was randomly chosen from a uniform distribution and it was queued in the buffer associated with that destination. Bursts were then created according to the previously explained hybrid approach. For this simulation, we set $B_{\max} = 100$ Kbit and $T_{out} = 1ms$.

Figure 3-22 and Figure 3-23 illustrate the queuing delay and blocking probability for two different cases:

**Case1.** $\alpha = \alpha_{ON} = \alpha_{OFF} = 1.2$

**Case2.** $\alpha = \alpha_{ON} = \alpha_{OFF} = 1.5$

It is observed that for the parameters given, the degree of self-similarity of the two input traffic types only affect the network performance below a 30% load. Beyond that point the queuing delay and the blocking probability are almost the same for both cases. This is because; perhaps the burst aggregation process does affect the long-range dependency of the input traffic.

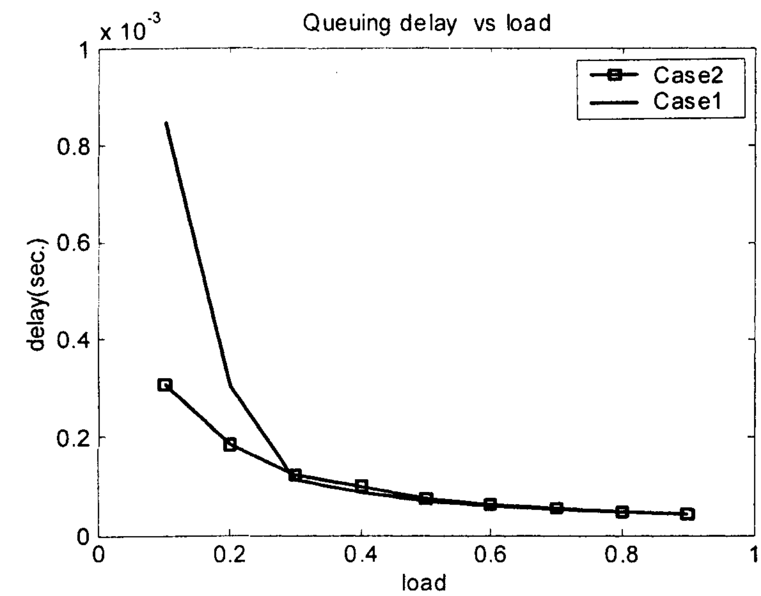


Figure 3-22. Queuing delay: self-similar traffic with Bmax = 64 Mbit and Tout =1ms

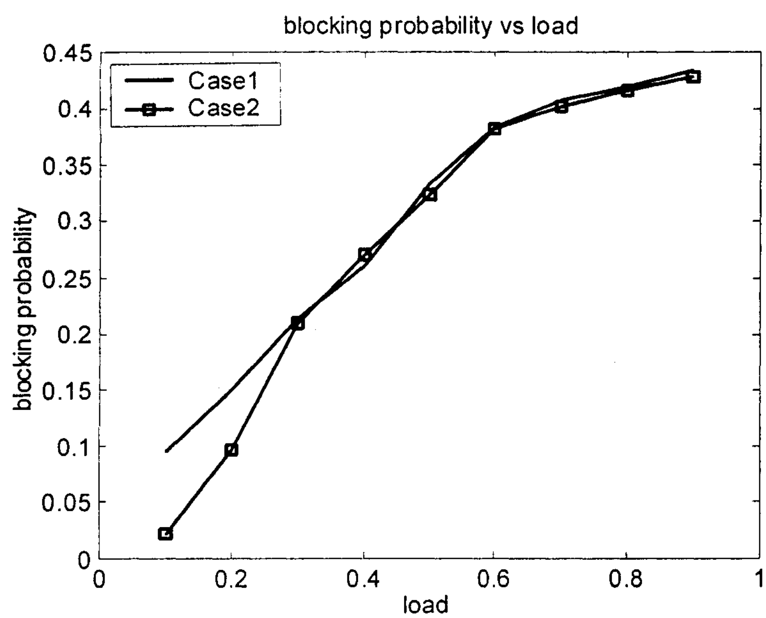


Figure 3-23. Blocking probability: Self-similar traffic with Bmax = 64 Mbit and Tout =1ms

## 3.8 Conclusion

In this chapter, the performance of different types of burst assembly algorithms has been discussed. The *hybrid* mechanism is very simple; nevertheless, it fulfills the maximum burst size requirements on bursts and keeps an upper bound on the delay experienced by the packets. An analytical model has also been developed to model the probability distribution of delays experienced by each packet in the aggregation queues for Poisson statistics. The presented model is numerically evaluated and the numerical analysis shows that the probability distribution of delays manifest a spike superimposed on a pedestal. The accuracy of the analytical model is verified by OpNet simulation. The impact of different burst aggregation methods and associated parameters (e.g. *threshold-value* and *time-out* value) on the blocking probability has also been investigated. The blocking probability is threshold value invariant. On the other hand, changing the value of the assembly time-out has a great impact on blocking probability. It is observed that the blocking probability increases with the increase in *time-out* value.

# 4 Proposed method of OBS with reservation for AAPN

## 4.1 Introduction

A major concern in optical burst switched networks is contention, which occurs when multiple bursts contend for the same link. Contention in an optical burst switched network is aggravated by the variable burst sizes and long burst durations. The performance study of Optical Burst Switching technique as a bandwidth sharing method in AAPN implies the simplification of a single intermediate node (the optical core switch), no time slot conversion (no FDL buffering) [9] and no wavelength conversion [10]. Contention resolution through deflection in wavelength and time are therefore ruled out. Deflection in space [11] is also probably ruled out (although tandem switch may be exploited for some low bandwidth connections).

In a realistic setting lost packets can be re-transmitted possibly exacerbating congestion and leading to collapse of the network throughput. One very crucial parameter for the feasibility of this approach is the size of the buffers. Especially, if the size of the network is very large then the propagation delay will be significant enough and hence it might be needed to keep copies of many of the transmitted bursts. The authors in [31] introduced a congestion control mechanism which keeps the average number of retransmission below 1.5 by reducing the offered load as a function of the collision rate. Nonetheless, with this approach the utilization of the network remains below its potential. Hence, the authors in [33] proposed two other modifications of the basic retransmission scheme. However, the proposed approaches described in [33] still suffer from several drawbacks. For high load, the average burst transmission delay and the buffer capacity are highly dependent on network diameter. The implementation of this scheme also results in out of sequence delivery of packets. In this chapter, a simple and enhanced reservation approach, which aims to eliminate packet loss completely and alleviate the shortcomings associated with the retransmission scheme, is proposed in the context of AAPN.

This chapter is organized as follows: section 4.2 explains the proposed reservation technique. Section 4.3 presents the simulation results and finally, section 4.4 summarises the chapter and draws some interim conclusions

## 4.2 Proposed Reservation approach

In pure OBS-AAPN architecture, the network control is exercised at the edge nodes; there is no control at the core nodes. The function of the core node is to provide only a transparent data path and the burst is simply lost in case of contention. However, in OBS with reservation, both the core node and edge nodes are involved in the process. Indeed, the core node will schedule the transmission window of an incoming burst based on the knowledge of round trip propagation delay and other routing information e.g. burst length, destination address, etc.; and send an acknowledgement back to the edge node with the necessary information about how long it should wait before transmitting the burst.

Following the proposed reservation approach, an edge node first sends a *setup request* message to the core node in a separate reservation channel (which may be physically in band but logically out of band). The *request* message contains the source, destination addresses, and other necessary routing information (e.g. burst length). In the core node all the *setup requests* arrive over time and in our approach, we would like to schedule each as soon as possible. In order to facilitate this task, we freeze the current schedule of each destination by maintaining an ordered list of transition times ('start' and 'stop' times) and try to schedule the next burst without changing the slots of any of the previously scheduled bursts. In other words, it is assumed that we have scheduled $N$ bursts for destination $d$ then we will not change any of them to schedule the $(N+1)th$ burst. Nevertheless, one could reschedule the bursts up to the critical time the acknowledgement must be sent. The current schedule creates empty slots or holes where we first try to fit the next burst based on a *scheduling algorithm*. However, if it is not possible to fit the burst in any of the empty slots, then its transmission window will be scheduled after the transition of the $Nth$ burst based on the knowledge of the round trip delays and a *setup ack* packet is sent back to the concerned edge node. The *setup ack* packet contains the offset field that informs the edge node how long it should wait before transmitting the burst. Figure 4-1 provides a flowchart of the *scheduling algorithm* executed in the proposed reservation scheme. For the purpose of this work, the switch reconfiguration time will be ignored. It was assumed that switches reconfigure instantly and the packets containing *setup request* and *setup ack* messages are sent on a separate channel and assumed to be much shorter than the actual data bursts. Note that, in this work we investigated single wavelength layer- equivalently separately scheduled layers. The impact of multiple wavelengths is a topic for further study.

In our model, it was assumed that the *setup request* and *setup ack* message would reach the edge node successfully. However, in a realistic setting it is possible that the *setup request* and *setup ack* message is lost in the way. In that case, a timer can be set to a round trip propagation delay from the source to destination and if the *ack* message is not received before the time out occurs, the edge node will retransmit the *setup request* message

The proposed approach completely eliminates the packet losses. Nevertheless, it introduces additional burst waiting time at the edge routers. For the remainder of this chapter, we will term this additional waiting time as the *scheduling delay* (*scheduling delay* = the time when an edge sends a *setup request* message to the core – the time when the edge starts transmitting the corresponding burst). Another very crucial parameter for the feasibility of such a scheme is the size of the network. If the size of the network is very large then the round trip delays could increase the scheduling delay very significantly. The proposed reservation approach is more suitable for metropolitan or local area networks and for those applications, which have stringent packet loss requirements but relaxed delay requirements.

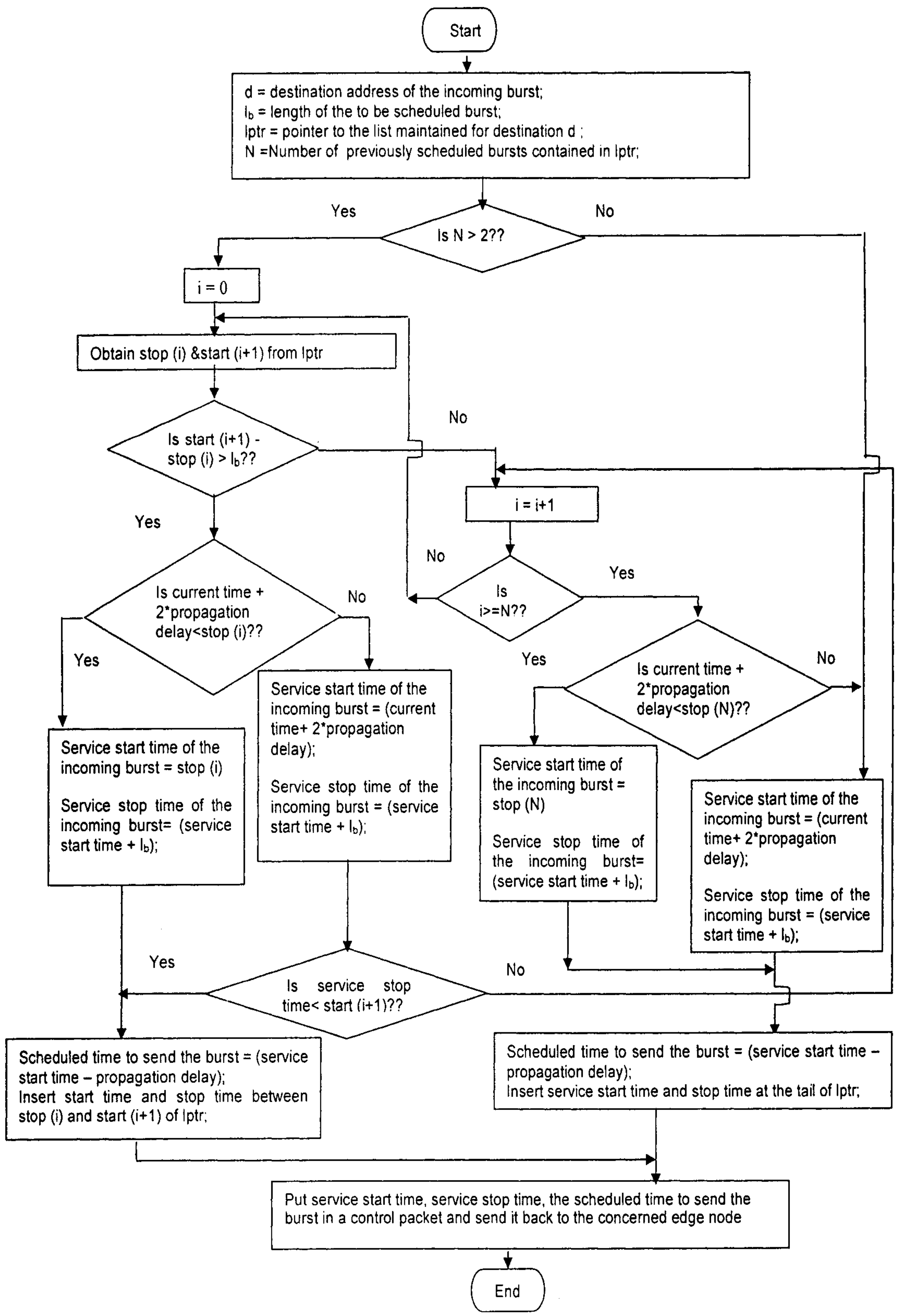


Figure 4-1. Flow chart of the proposed reservation approach

## 4.3 Simulation Results

In order to evaluate the performance of the proposed reservation approach, a scalable and modular framework for simulating an OBS-AAPN consisting of 8 sources, 8 edge nodes and 1 core node has been implemented using OpNet Modeler™ for a single wavelength channel. The links connecting the users to the edge nodes and those connecting the edge nodes to the core node are bi-directional and carry only one data channel with a capacity of 10 Gb/s. In this simulation all the channels have zero bit error rates. A detailed description of the network model is given in the appendices. Note that, in this case, the average ETE delay experienced by each packet is the sum of four terms, namely (i) aggregation delay; (ii) scheduling delay; (iii) burst transmission and propagation delay; (iv) de-aggregation delay. As mentioned in chapter 3, in our simulation, it was assumed that the de-aggregation delay is negligible compared to other delays.

The following simulations were carried out using Poisson traffic with the following parameters:

- The packet inter-arrival time (in seconds) is exponentially distributed with a mean $\mu$, where $\mu = \dfrac{L}{\rho \times R}$

  $\rho$ = Network load $(0 \le \rho \le 1)$;

  $L$ = Packet length = 12000 bits (constant);

  $R$ = Transmission speed = $10Gb/s$;

- The destination of each packet is chosen randomly with uniform probabilities, i.e., the same probability for all destinations (and the probability is $1/N = 1/8$)
- The sources are independent from each other but they use all the same parameters for the random generation of packets.
- The maximum burst size, $B_{\max}$ = 64 Mbit and the assembly time out, $T_{out}$ = 10 ms.
- The propagation delay was set to zero (however, in section 4.3.3 we will set different propagation delays to observe its impact on the proposed reservation scheme.)

### 4.3.1 Threshold based

Figure 4-2 illustrates different delay characteristics of the proposed reservation approach adopting the *threshold based* burst aggregation mechanism. As observed in chapter 3, the mean aggregation delay will decrease with an increase in load for the *threshold-based* approach. However, in this case the scheduling delays will increase with the increase in network load. This is because as the load increases the number of bursts generated in the network also increases, which will in turn increase the scheduling delay. This increasing scheduling delay and the constant transmission delay will be added to the aggregation delay to yield the total end-to-end (ETE) packet delay.

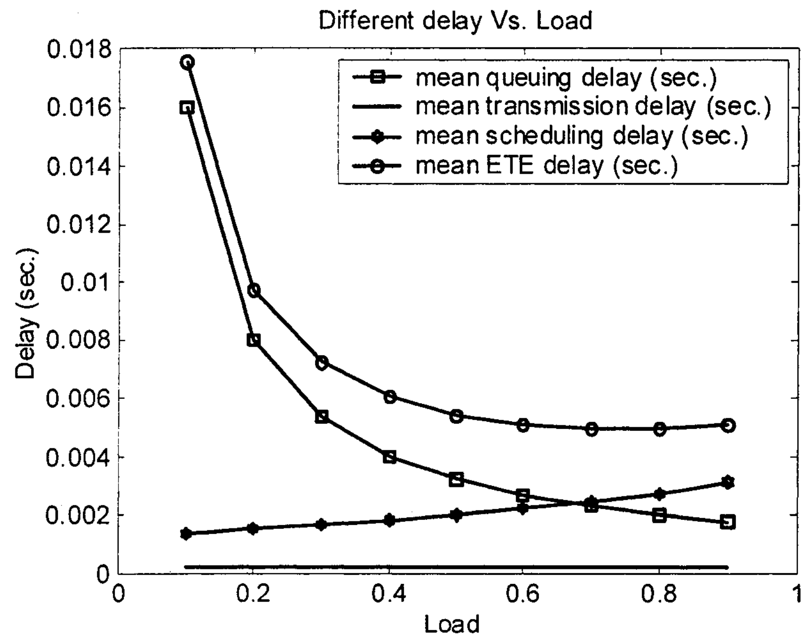


Figure 4-2. Delay characteristics: Reservation scheme-adopting Threshold based burst aggregation mechanism with Bmax = 64 Mbit

### 4.3.2 Timer based

Figure 4-3 demonstrates the delay properties of the proposed reservation approach when bursts were created according to the timer-based burst assembly mechanism. As mentioned in chapter 3, the mean aggregation delay is constant for timer-based approach. However, in this case, the ETE delay increases almost linearly with the increase in load. This can be explained by the fact that under high input load, it generates bursts that are quite large, increasing the scheduling delay, which in turn increases the average ETE packet delay (as shown in Figure 4-3).

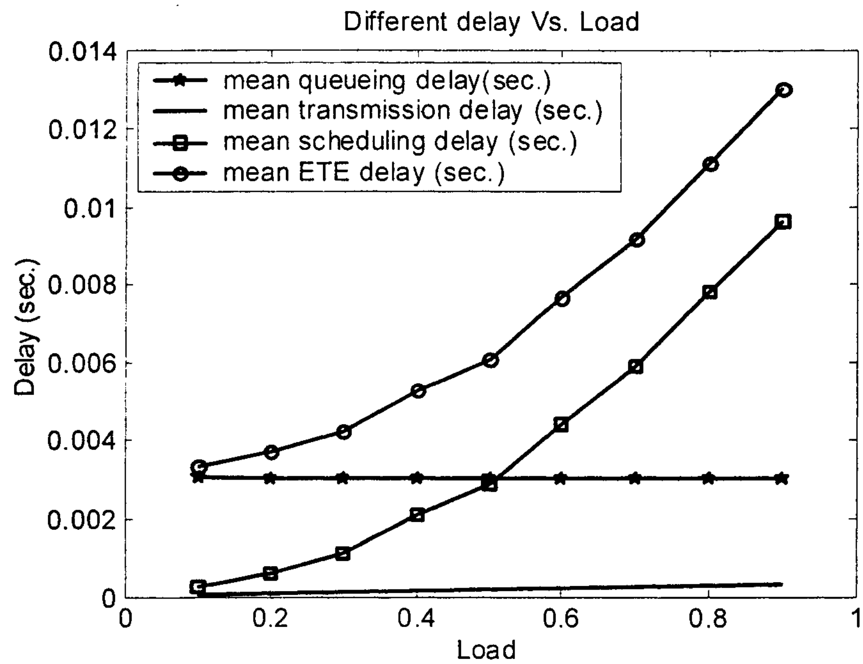


Figure 4-3. Delay characteristics: Reservation scheme adopting the Timer-based burst aggregation mechanism with Tout =10 ms

### 4.3.3 Hybrid Based

Figure 4-4 illustrates the delay characteristics of the proposed reservation approach adopting the hybrid burst aggregation mechanism. It is observed that for the given parameters, the ETE delay increases almost linearly until the load reaches 40% and beyond that limit it starts decreasing as the load increases. This is because at low load (from 0 to 40% load), it becomes rare that the aggregation buffers will fill before the timer triggers. Hence, the time out mechanism becomes the controlling factor in the aggregation process and consequently the mean delay will be increased with the increase in load as observed in Figure 4-3 for timer based approach. However, when the load is high (above 40% load) then the buffers will fill rapidly to $B_{max}$. Consequently the length out mechanism becomes the dominant factor at high load and the ETE delay will decrease with an increase in network load as observed in Figure 4-2 for the threshold-based mechanism.

In practice for the hybrid approach, the parameters would be set so that the transition from the time out mechanism to threshold based mechanism occurs at much lower loads and hence, its effect would not be observable on mean delays. An example is shown in Figure 4-5, where $B_{max}$ =100 Kbit and $T_{out}$ = 1 ms. In this case, $\rho * R * T_{OUT} >> B_{max}$ and hence the transition from time out mechanism to threshold-based mechanism occurs at much lower loads (below 10% load). Therefore, its effect is not noticeable on the delay characteristics.

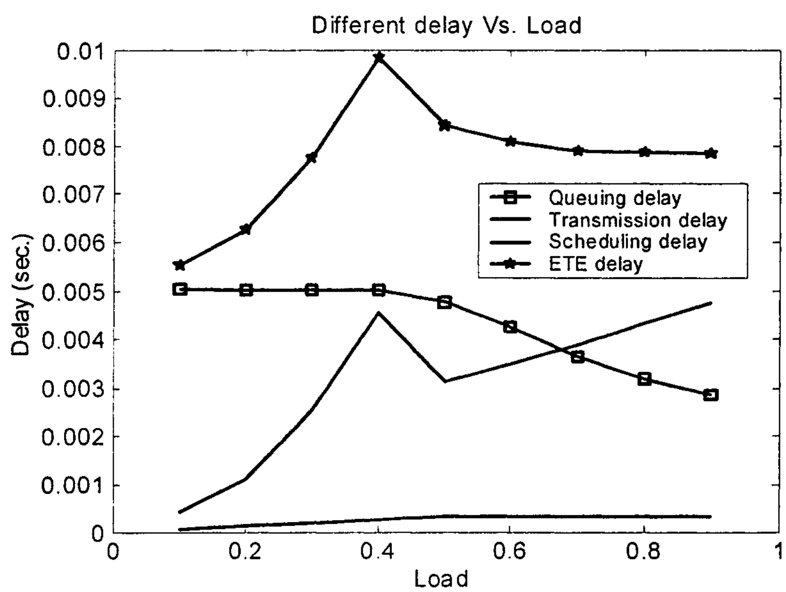


Figure 4-4. Delay characteristics: Reservation schemes adopting Hybrid based burst aggregation mechanism with Bmax = 64 Mbit and Tout =10 ms

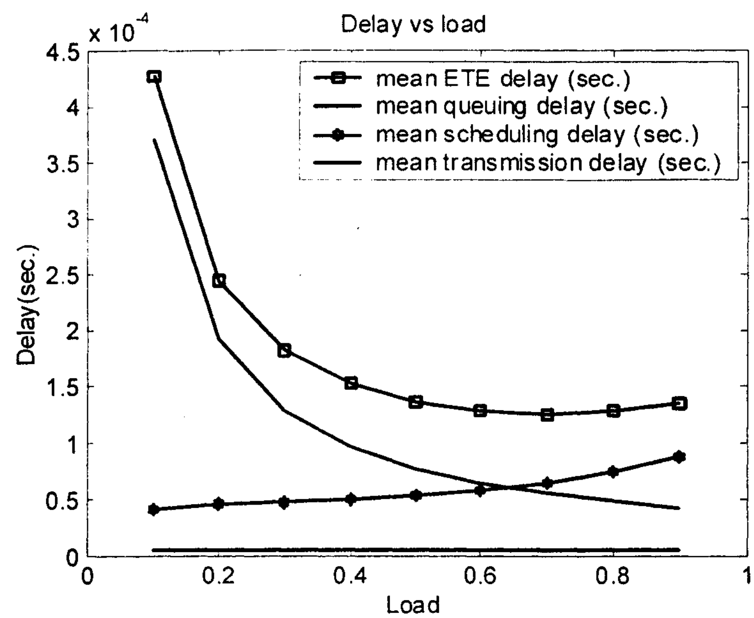


Figure 4-5. Delay characteristics: Reservation schemes adopting Hybrid approach with Bmax = 100 Kbit and Tout =1 ms

The performance of the proposed reservation approach has also been verified using two other traffic models, the non-uniform Poisson traffic and the self-similar traffic model. The non-uniform traffic aims to capture the performance of the network when the destination address distribution of the generated traffic is not uniformly distributed. On the other hand, self-similar traffic deals with the actual Internet traffic model and captures the performance of the network under bursty traffic. It is generated by multiplexing several heavy-tailed Pareto distributed ON-OFF sources with $\alpha = 1.5$. Usually, the performance of the network is much worse under self-similar traffic. Further details of the generation of these two-traffic models are given in chapter 3.

Figure 4-6 shows the scheduling delay versus an increase in network load under non-uniform and bursty traffic scenarios. Bursts were created according to the hybrid approach, where $B_{max}$ = 100 Kbit and $T_{out}$ = 1 ms. It is observed that under self-similar traffic, the scheduling delay is much larger than that obtained under non-uniform traffic for the same load. Consequently, the mean ETE delay is also higher under bursty traffic, which is shown in Figure 4-7.

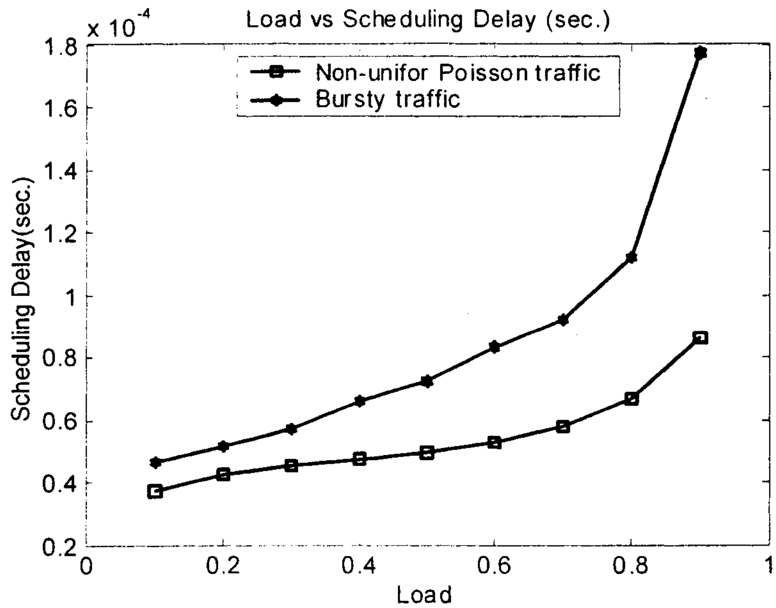


Figure 4-6. Scheduling delay: Reservation scheme under non-uniform and bursty traffic with Bmax = 100 Kbit and Tout =1 ms

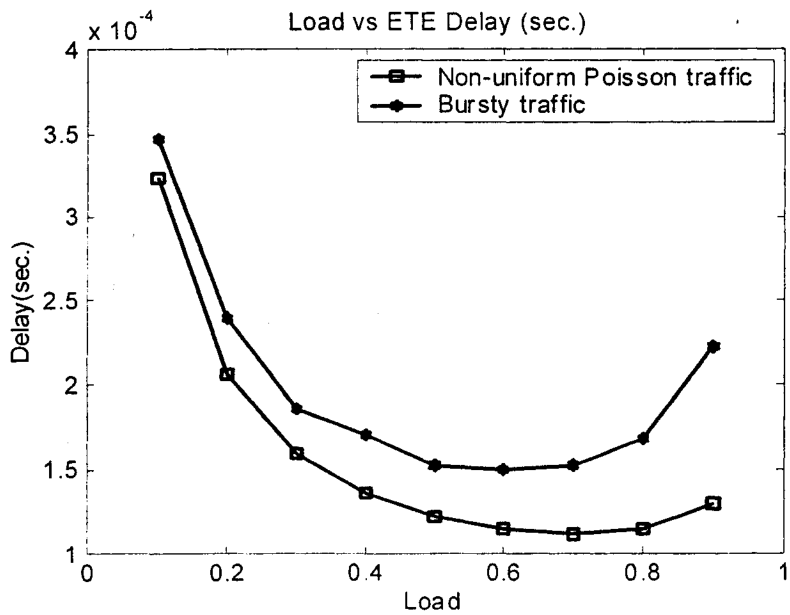


Figure 4-7. ETE delay: Reservation scheme under non-uniform and bursty traffic with Bmax = 100 Kbit and Tout =1 ms

The simulation results discussed so far were carried out with the propagation delay set to zero. However, one very crucial parameter for the feasibility of the reservation scheme is the size of the network. Hence, the proposed approach has been evaluated for the following different metro scenarios, with equal or varying distances from edge nodes to core node.

**Case1**. Zero propagation delay between core and edge nodes

**Case2**. 0 slots $= 0\mu\sec. \approx 0km$ between edge1 and core node.

1 slots $= 10\mu\sec \approx 2km$ between edge2 and core node

2 slots $= 20\mu\sec. \approx 4km$ between edge3 and core node

3 slots $= 30\mu\sec. \approx 6km$ between edge4 and core node

7 slots $= 70u\sec. \approx 14km$ between edge5 and core node

8 slots $= 80\mu\sec \approx 16km$ between edge6 and core node

9 slots $= 90\mu\sec. \approx 18km$ between edge7 and core node

10 slots $= 100\mu\sec \approx 20km$ between edge8 and core node

**Case3**. 10 slots delay $= 100\mu s \approx 20km$ between core and edge nodes

Bursts were created according to hybrid approach. We set $B_{\max} = 100Kbit$ and $T_{out} = 1ms$. Figure 4-8 and Figure 4-9 shows the simulation results when the network was fed with uniform Poisson traffic. It was expected that if the size of the network is increased then the mean scheduling delay will be also increased due to higher round trip propagation delays, which will in turn increase the mean ETE delay. The simulation results confirm what we expected. It is observed that for case 3; i.e., when the average distances between the core and edge node is largest, then the packets will experience the highest mean ETE delay.

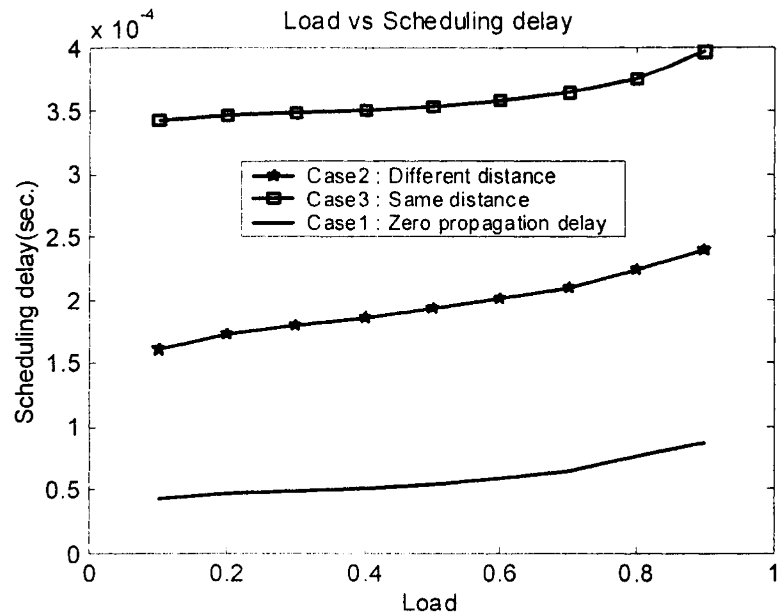


Figure 4-8. Scheduling delay: Reservation scheme under different propagation delay with Bmax = 100 Kbit and Tout =1 ms

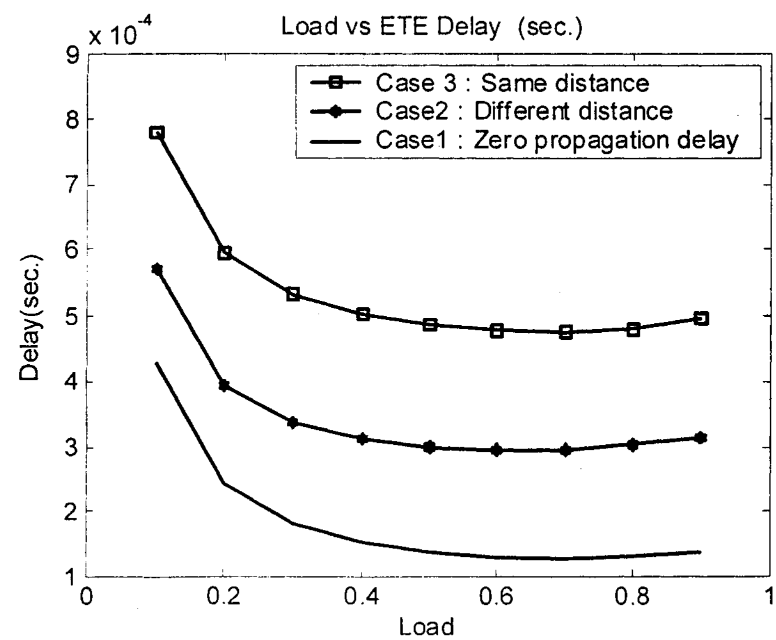


Figure 4-9. ETE delay: Reservation scheme under different propagation delay with Bmax = 100 Kbit and Tout = 1 ms

## 4.4 Conclusion

In this chapter, a simple and enhanced reservation method with a single acknowledgement, which eliminates completely the packet loss associated with conventional OBS-AAPN, has been implemented. This method does not introduce any additional cost or hardware complexity in the network. Nonetheless, it incurs another kind of delay, which is the additional waiting time in the edge nodes. The proposed reservation scheme is evaluated under different metropolitan scenarios and different traffic conditions adopting various burst aggregation techniques. Simulation results show that the reservation scheme is sensitive to the network size. This technique is more suitable for small networks.

# 5 Conclusion and Future work

Optical Burst Switching, OBS, is a promising switching technology for transferring data in the IP backbone as it avoids the electronic speed bottlenecks inherent in SONET/SDH and overcomes the lack of fine bandwidth granularity in traditional wavelength division multiplexing (WDM) systems. In this thesis, OBS is considered for subdividing the bandwidth of each wavelength in the context of an Agile All-Photonic Network (AAPN). Several architectural and design issues need to be addressed, however, before deploying the OBS technique in the AAPN backbone.

One important design issue in an OBS network is the burst assembly algorithm. In this thesis, various approaches to the burst assembly mechanism have been implemented and evaluated to find the best way of aggregating the variable length packets into suitably sized bursts, and to observe the impact of the different types of burst aggregation approaches and its related parameters on the performance of an OBS-AAPN. Performance measures of interest were mean delays and blocking probabilities.

A simulation platform for OBS over AAPN was implemented in OPNET Modeler to perform the evaluation of this bandwidth sharing method in AAPN. In the edge nodes, packets, which have the same destination address, are assembled into bursts. Data burst assembly depends on two important parameters: the maximum burst size and the waiting time before a burst is sent even if the maximum burst size has not been reached (also called *time out interval*). In the core node, a dynamically varying list of transmission times (start and stop times) of reserved transmission windows is maintained for each output port in the core node. If the transmission time of the incoming burst overlaps with any of the bursts already scheduled for transmission, then the incoming burst is dropped; otherwise, it is scheduled for transmission and its time window is inserted in the list. When a burst is transmitted, its corresponding entry is removed from the list.

In our simulation we did not have the provision to simulate a scheduled circuit switch in OPNET Modeler; therefore in this work the burst aggregation process at the edge node was modeled only to derive the header departure process and, after transmitting the header, the bursts were destroyed. Such a simplication does not have a deteriorating impact on the results because the header contains all the relevant information and the burst itself is not needed anymore to obtain the desired performance measures.

In our simulation model, bursts were created and sent into the network according to three different types of techniques, namely: *timer-based*, *threshold-based* and *hybrid* mechanisms. In a *timer-based* burst assembly approach, a burst was created and sent into the optical network at pre-defined time intervals. The major drawback of this approach is that this scheme might produce undesirable burst lengths at different loads. In a *threshold-based* approach, a burst was created and sent out when the size of the aggregation queue reaches or exceeds a certain threshold value of $B_{max}$ in bits. The problem of this approach is that it does not give any guarantee on the assembly delay that the packets will experience during the burst aggregation process. In contrast to *timer-based* and *threshold-based* approach, in the *hybrid* mechanism bursts were created when either the size of the buffers reaches the threshold value or the time-out occurs. The hybrid mechanism performs better than other approaches since at low traffic condition it does not need to wait for long to gather enough packets to create a predefined sized burst or at high volume load it does not produce undesirable long bursts. The results obtained by simulation show that the *hybrid mechanism* is very simple, yet:

- Fulfills the maximum burst size requirements on bursts.
- Keeps an upper bound on the delay experienced by the packets by choosing the *time-out* comparable to the *maximum tolerable delay*, but only the packets in the low rate flow suffer this delay.
- Reduces the blocking probability and the reservation delays since resources are reserved by bursts for a smaller period.

An analytical model, which incorporates the different types of burst aggregation methods, has also been developed for Poisson traffic with fixed length packets. We derived close formulas to model the probability distribution function of the aggregation delays. The presented model was evaluated numerically using MATLAB code and the analysis shows that the approximate equations provide a good qualitative description of the exact distributions. A good agreement between the numerical evaluation and the OPNET simulation shows the correctness of our model.

Given that the no-reservation approach leads to very high packet loss probability, a simple OBS technique which reserves the resources in advance has been proposed. The proposed approach eliminates completely the packet losses observed in the network with the previous approach and retains the simplicity of the core node. This approach, however, adds additional burst waiting time at the edge routers and it may also cause the optical links to idle unnecessarily, which would likely require compensation in the form of an increase in speed-up of the network (ratio of internal capacity to external capacity). Simulation results also show that the performance of the reservation technique is highly dependent on round-trip propagation delays and hence it is suitable for small size networks.

Following the performance analysis of the two OBS approaches studied, one can say that the reservation approach is appropriate for traffic that has stringent packet loss requirements but relaxed delay requirements. On the other hand, with the conventional OBS technique the loss probability is high but the end-to-end delay is less since there is no need to wait for an acknowledgement. Therefore, conventional OBS is suitable for loss insensitive traffic. None of the two techniques offers flexibility with respect to both delay and packet loss, which suggests that a further combination of the two approaches may be required to handle successfully these two types of traffic that are ever present in all networks.

There are a number of potential avenues where further work could be valuable:

- Extend the presented analytical model to observe the impact of variable length packets and other different traffic statistics on aggregation delays.
- Develop an analytical approach to evaluate the blocking probabilities for different types of burst aggregation methods, which is still an open issue in Optical Burst Switched networks. A popular simplified traffic model, Poisson model, can be applied to estimate the loss probabilities.
- Investigate the impact of multiple wavelength channels on the performance of the reservation scheme so that the core node could let the edge node know which wavelength it should use to transmit the data.

- Address the fairness issue, which is inherent in the reservation technique due to different propagation delays in the network.
- Compare the performance of the OBS technique to the Adaptive Wavelength-Timeslot Allocation (AWTA) scheme in context of AAPN and to assess their relative merits. Full implementation of a simulation test bed for the AAPN architecture with the AWTA method is already developed in Opnet Modeler. Different traffic and deployment scenarios (e.g. varying distances from edge nodes to core node) are of interest to make the comparison.

# References


1. R.Vickers and M.Beshai,"Petaweb architecture", Networks 2000 Symposium, Toronto, Canada, 2000
2. G.V.Bochmann, Trevor J. Hall, Oliver Yang, M.J. Coates, L Mason and R. Vickers, "The Agile All-Photonic Network: An architectural Outline", 22nd Biennial Symposium on Communications, Kingston, Canada, May31-June3, 2004, pp217-218.
3. M.Yoo and C.Qiao, "Just-enough-time (JET): A high speed protocol for bursty traffic in optical network," In Proceeding of IEEE/LEOS conf. on Technologies for a Global Information, Infrastructure, August 1997, pp.26-27.
4. C.Qiao and M.Yoo, "Optical burst switching (OBS)-a new paradigm for an optical Internet", Journal of High speed Networks, vol.8, no.1, pp.69-84, 1999
5. J.Turner, "Terabit Burst Switching", Journal of High speed Networks, vol.8, no.1, pp.3-16, 1999
6. E. Haselton, "A PCM frame switching concept leading to burst switching network architecture," IEEE Communications Magazine, vol. 21, pp. 13-19, June 1983.
7. S. Amstutz, "Burst switching - an introduction," IEEE Communications Magazine, vol. 21, pp. 36-42, Nov. 1983.
8. Klaus Dolzer, Christoph Gauger, Jan Spath, and Stefan Bodamer, "Evaluation of reservation mechanism for optical burst switching", AEU Int. J. Electron. Communication, vol.55 (1), Jan.2001, pp. 1-8, Jan 2001
9. Chlamtac, A. Fumagalli, L. G. Kazovsky, et al., "CORD: Contention Resolution by Delay Lines," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp.1014-1029, June 1996.
10. Ramaswami and K.N. Sivarajan, "Routing and Wavelength Assignment in All-Optical Networks," IEEE/ACM Transactions on Networking, vol. 3, no. 5, pp.489-500, Oct.1995
11. Ching-Fang Hsu, Te-Lung Liu, and Nen-Fu Huang, "Performance Analysis of

Deflection Routing in Optical Burst-Switched Networks", Proceedings IEEE, INFOCOM 2002, New York, NY, pp. 786-790, June 2002

12. Yijun Xiong, Marc Vandenhoute, Hakki Cankaya, "Control Architecture in Optical Burst-Switched WDM Networks", IEEE Journal on Selected Areas in Communications, vol.18(10), Oct.2000, pp.1838-1851.
13. Vinod M. Vokkarane, Jason J. Pue, "Prioritized Burst Segmentation and Composite Burst-Assembly Techniques for QoS Support for Optical Burst Switched Networks", IEEE Journal on Selected Areas in Communications, vol.21 (7), Sept.2003, pp.1198-1209.
14. Sanjeev Verma, Hemant Chaskar, and Rayadurgam Ravikanth, "Optical Burst Switching: A Viable Solution for Terabit IP Backbone", IEEE Network, vol.14 (6), November 2000, pp.48-53.
15. John Y.Wei, Ray I. Mcfarland, " Just-In-Time Signaling for WDM Optical Burst Switching Networks", Journal of Lightwave Technology, vol.18(12), December 2000, pp.2019-2037.
16. Jonathan S. Turner, "Terabit Burst Switching", Journal of High Speed Networks, vol. 8 (1), 1999, pp.3-16.
17. M. Yoo, C. Qiao, Sudhir Dixit, "Optical Burst Switching for Service Differentiation in the Next Generation Optical Internet", IEEE Communication Magazine, Feb.2001, pp.98-104.
18. M. Duser, P. Bayvel, "Analysis of a dynamically wavelength-routed optical burst switched network", Journal of Lightwave Technology, vol.20 (4), April 2002, pp.574-582.
19. E. Kozlovski, M Duser, A.Zapta and P.Bayvel, "Service differentiation in WR-OBS networks", Tech. Dig.OFC2002, ThGG114, Anaheim, CA, March 2002, pp.774-776
20. An Ge, Franco Callegati, and Lakshman S. Tamil, "On Optical Burst Switching and Self-Similar Traffic", IEEE Communications Letters, vol.4 (3), March2000, pp.98-10
21. X. Yu, Y. Chen, and C. Qiao, "Study of traffic statistics of assembled burst traffic in optical burst switched networks", In Proceeding of Opticomm , Boston, MA,

July-Aug 2002, pp 149-159.

22. X. Cao, J. Li, Y. Chen, and C. Qiao, "Assembling TCP/IP Packets in Optical Burst Switched Networks", In proceeding of Globecom 2002, Taipei, Taiwan, November 2002, pp. 730.
23. K. Dolzer and C. Gauger, "On burst assembly in optical burst switching networks - a performance evaluation of Just-Enough-Time", In Proceedings of the 17th International Teletraffic Congress (ITC 17), September 24-28, 2001 Brazil, pages 149-160.
24. J.Choi, H. Le Vu, Craig W Cameron, M. Zukermau, M. Kang, "The effect of Burst Assembly on performance of Optical Burst Switched Network", The International Conference on Information Networking, Feb18-20, South Korea, 2004, LNCS 3090, pp. 729-739.
25. D. Morato, J. Aracil, L.A. Diez, M. Izal, and E. Magana., "On linear prediction of internet traffic for packet and burst switching networks", In proceedings of Tenth International Conference on Computer Communications and Networks, October 10-15, 2001, Arizona, USA, pp 138-143.
26. V. Vokkarane, K. Haridoss, and J.P. Jue., "Threshold-based burst assembly policies for QoS support in optical burst-switched networks", In Proceeding of Opticomm, Boston, MA, July-Aug 2002,pp 125-136.
27. Z. Haas, "The 'Staggering Switch': An Electronically Controlled Optical Packet Switch," *IEEE Journal of Lightwave Technology*, vol. 11, no. 5/6, May/June 1993, pp. 925-936.
28. D. K. Hunter, W. D. Cornwell, T. H. Gilfedder, et al., "SLOB: A Switch with Large Optical Buffers for Packet Switching," *IEEE Journal of Lightwave Technology*, vol.16, no. 10, Oct. 1998, pp. 1725-1736.
29. A.Detti, V. Eramo, and M. Listanti. "Performance evaluation of a new technique for IP support in a WDM optical network: optical composite burst switching (OCBS)" Journal of Lightwave Technology, vol.20 (2), 2002, pp.154-165.
30. Vinod M. vokkarane, Jason P.Jue, and S. Sitaraman, "Burst segmentation: An approach for reducing packet loss in OBS Networks", IEEE International Conference on Communications, vol.5, April 2002, pp. 2673-2677.

31. Abdelilah Maach, G. Bochmann and H.T. Mouftah, "Robust Optical Burst Switching." In Proc. of Networks'2004, Vienna, Austria, June 2004, pp. 447-452.
32. Sonia Parveen, Sofia Pardes, Trevor J.Hall, "Evaluation of burst aggregation method in an Optical Burst Switched Agile All-Photonic Network", Photonics North, September 12-14, 2005, Toronto, Canada (accepted).
33. Anna Agusti, Gregor v. Bochmann, Cristina Cervello-Pastor,' "Retransmission schemes for Optical Burst Switching over star networks", Wireless and Optical Communication Network (WOCN) 2005, Dubai, United Arab Emirates, March 6-8, 2005.

# Appendix A. Implementation of the OBS-AAPN in OpNet Modeler

The sample star network topology used for the simulations is shown in Figure A-1. It consists of 8 traffic sources, 8 edge nodes and 1 core node. The network core represents the optical transport network. Switches with aggregation capabilities are used as the network edge devices. In this model, bursts are queued for transmission when either the size of the bursts reaches its maximum size or the time out occurs. The links connecting the users to the edge nodes and those connecting the edge nodes to the core node are bi-directional and carry only one data channel with a capacity of 10 Gb/s. All the channels have zero bit error rates.

In an AAPN network there is only one optical switch in the path to configure (that of the core node); therefore no onward transmission of a header is required. Ideally headers should be time-stamped and queued for processing in the core node.

Contention occurs at the output ports of the core node; therefore headers must be sorted by destination address and their requested transmission window compared with the current output schedule for that port. This output schedule should be then used to control all cross-points connected to that output; i.e., if the input ports form the rows and the output ports form the column of a crossbar, the cross points occur at the intersections of the row and columns, and the output port schedule controls the setting of the cross points in the column corresponding to that output. When a burst is successfully scheduled, that information should be used to close the appropriate cross-point in the future during the transmission window reserved by the header, thus allowing the data burst to pass through the switch to its correct destination transparently. Cross-points should be otherwise left open – i.e. no connection. This ensures that bursts that are not successfully scheduled meet an open cross point on arrival at the switch and are therefore dropped. Unfortunately, in our simulation we did not have the provision to simulate a scheduled circuit switch in OPNET Modeler; therefore in this work the burst aggregation process at the edge node was modeled only to derive the header departure process and, after transmitting the header, the bursts were destroyed. The actual transmission and switching of bursts was not implemented; only the headers were transmitted from one edge node to another to calculate the delays.

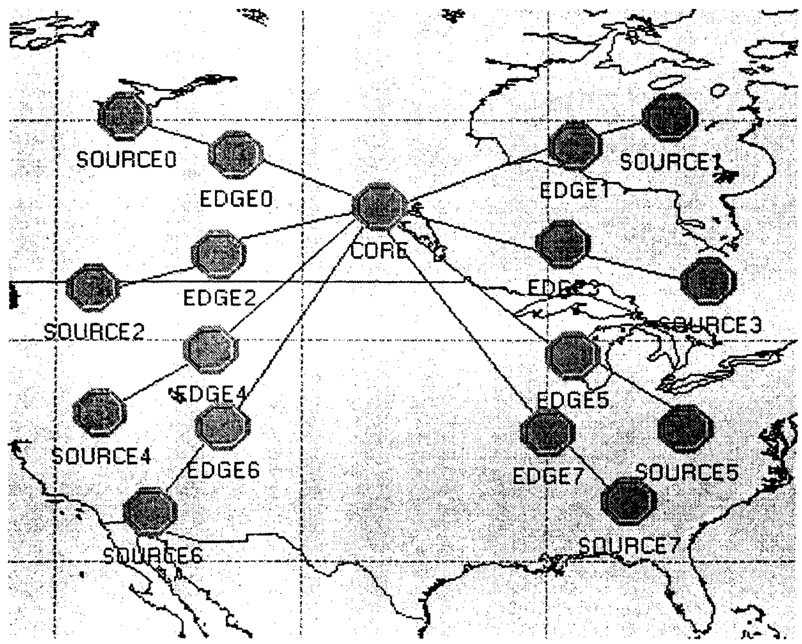


Figure A-1. Optical Burst Switched Network model

In OPNET, a network is made of individual nodes and each node is made of modules. The process model of a module defines its behavior and it is defined by a finite state machine. A detail discussion of the edge nodes and the core node of our network model is given in the following sections.

## A.1 Edge node

Usually, the function of the edge node is to perform the aggregation/de-aggregation of bursts at the edge of the network. The node model of the edge router used in the simulations is shown in Figure A-2. It consists of 2 inputs and 2 outputs each represented by a point-to-point receiver and transmitter, respectively. All the transmitters and receivers are connected to a central processor (called "aggregator") via packet streams.

The central processor is responsible of performing the aggregation function and calculating the average end-to-end packet delay.

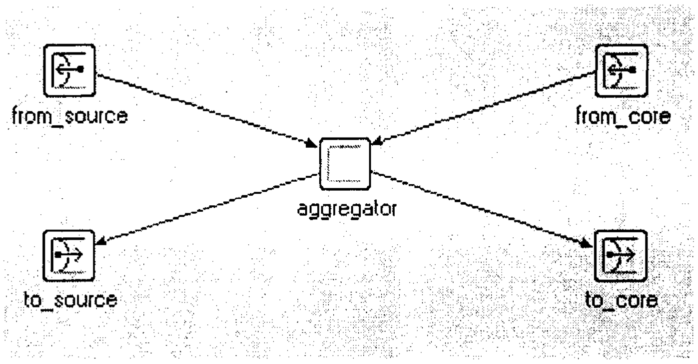


Figure A-2. Node model of the Edge router

The process model of the edge node is shown in Figure A-3. This process model has five states: *initial state, idle state*, *aggregate state*, *header transmission state* and *header arrival state*.

At the start of the simulation the *initial state* is activated. In it, a statistical file is declared to keep record of the end-to-end delay. An aggregation queue is associated with each destination edge node in the aggregation state.

When no actions are required and the edge node is only waiting for any packet to arrive, the simulation control stays in the *idle* state.

When a packet arrives in the edge node from the source, the *aggregation* state will be activated. In it, the destination address of the packet will be read and the packet will be inserted in the appropriate queue for this destination. During insertion, a check is also made to determine if the queue was empty. If so, then it will schedule a *time out interrupt* for this burst.

When the size of a queue reaches the maximum data burst size or the time out period elapses for any queue, the *header transmission* state will be activated. In this state, a header will be created containing all the information of its corresponding burst and sent to the core node. The newly created burst header consists of five fields as shown in Figure A-4. After transmitting the header, the burst is destroyed for the sake of simplifying (and hence speeding up) the simulation since the circuit switch of the core could not be properly implemented anyway, as it was explained above. Such a simplication does not have a deteriorating impact on the results because the header contains all the relevant information and the burst itself is not needed anymore to obtain the desired performance measures.

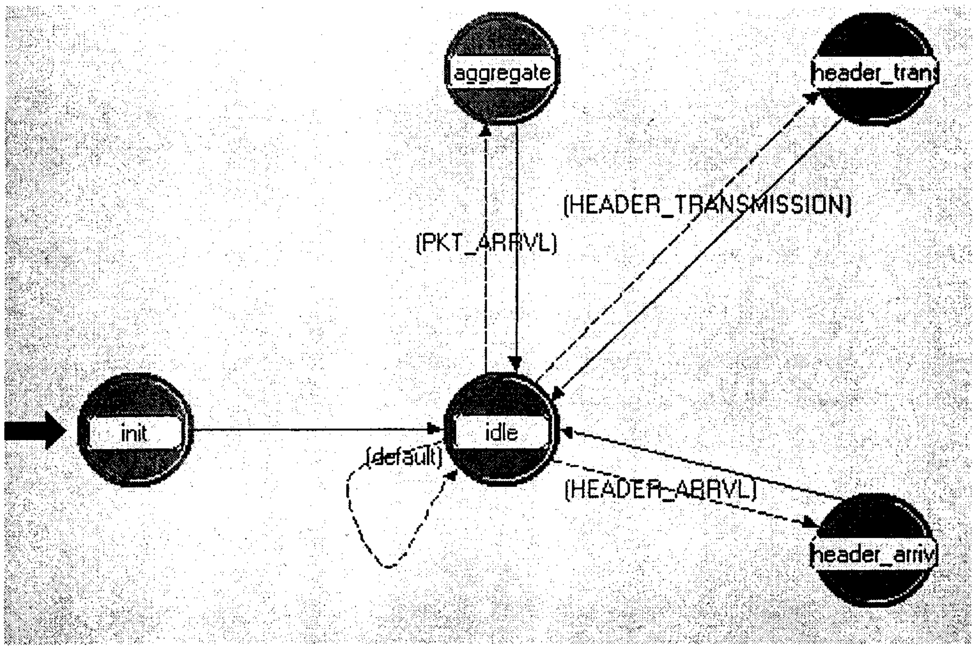


Figure A-3. Process model of the Edge node

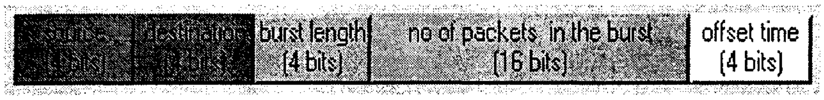


Figure A-4. Format of the control packet (Burst Header)

When the edge node receives a header from the core node, this process will enter the *header arrival* state. This state will calculate the end-to-end delay for each burst from the information contained in the header and will write it into a statistical file. After calculating the End-to-End delay the burst header is destroyed.

## A.2 Core node

A simplified architecture of the core node used in our simulation as is shown in Figure A-5. No optical buffering or wavelength conversion components were used in the core node. It consists of 8 inputs and 8 outputs, each represented by a point-to-point receiver and transmitter, respectively. All the transmitters and receivers are connected to a central scheduler via packet streams. The main role of the central scheduler is to process the headers to calculate the output schedules and to forward them based on their destination addresses. This module also calculates the blocking probability by keeping a count of the number of packets contained in those bursts that are not successfully scheduled.

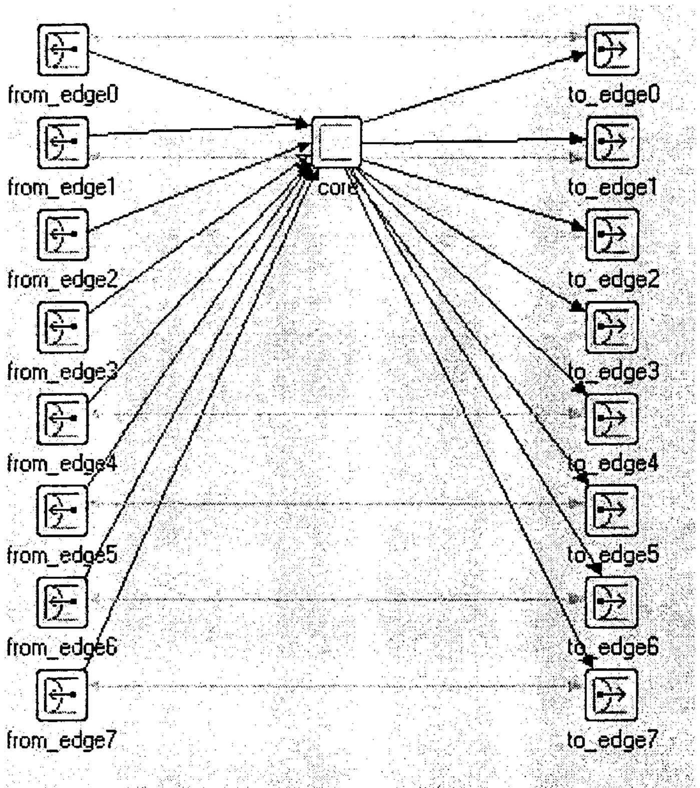


Figure A-5. Node model of the core scheduler

The process model of the scheduler is shown in Figure A-6. This process model has two states: *initial state* and *forwarding state*.

In the initial state, a statistical file to keep record of the blocking probability is declared. In the forwarding state, a dynamically varied list of burst transmission windows ("service start" and "service stop" times) of reserved/scheduled bursts is maintained for each output port, which corresponds to each destination edge node. When a header is received in the core node, the process enters the forwarding state and its destination address $d$ is obtained from the information contained in the header. The header is then processed against the output schedule (a list of bursts already scheduled) of the corresponding destination to determine whether the transmission time of the burst overlaps with that of any other previously scheduled time or not. If not, then the burst is successfully scheduled; otherwise it is dropped. If the burst is successfully scheduled for transmission then the following actions take place:

- The transmission window of the incoming burst is inserted in the list.
- A duplicate of the header is created and appended in tail of the list.
- The header is forwarded to its destination for further processing

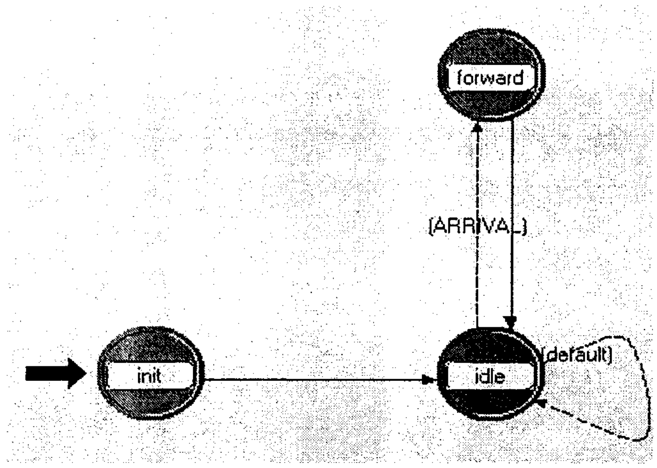


Figure A-6. Process model of the core node's central scheduler module

When the simulation ends the blocking probability is calculated from:

*blocking_probability = (number of packets dispatched from edge node i to edge node j that are lost) / (total number of packets dispatched from edge node i to edge node j)*

If the reservation approach is used, the header will be processed according to the algorithm described in Section 4.2 to obtain the time at which the source edge node will have to transmit the burst. The following actions take place after processing the header:

- The time-to-transmit is sent back to the source edge node inside a control packet
- The newly calculated transmission window for the burst is inserted in the list of the corresponding output schedule
- The header is forwarded to its destination for further processing.

# Appendix B. OpNet Code for Simulation

## B.1 Aggregation state

```
////////////////////////////////////////////////// For destination one ////////////////////////////////////////////////////////////

If(dest_address== 1 )

        {
        If(pk_seg_num1 == 0)
                {
                sbh1= op_sar_buf_create(OPC_SAR_BUF_TYPE_SEGMENT,OPC_SAR_BUF_OPT_DEFAULT);
                firstpacket_time1 = op_sim_time();
                aggregation_start_time1 = op_sim_time();
                time_out_interrupt1 = op_intrpt_schedule_self(firstpacket_time1+burst_time_out,1);
                }

        op_sar_segbuf_pk_insert(sbh1,pkptr,pk_seg_num1++);
        sum_arrivingpacket_time1 = sum_arrivingpacket_time1 + packet_receive_time;
        buffer_size = op_sar_buf_size(sbh1);

        If(buffer_size>=max_burst_size)
                {
                op_ev_cancel(time_out_interrupt1);
                max_burst_interrupt1 = op_intrpt_schedule_self (op_sim_time(),9);
                }
        }
```

## B.2 Core scheduler Without Reservation

```
////////////////////////////////////////////////////// Forwarding state ////////////////////////////////////////////////

if(pk_num == 0)
        {
        for(i=0;i<8;i++)
                lptr[i]= prg_list_create();
        pk_num++;
        }

/*get a packet from the stream*/

pkptr= op_pk_get(op_intrpt_strm());
total_burst_count++;
```

```
/* extract all its information from the header */

source=op_intrpt_strm();

op_pk_fd_get(pkptr,0,&destination_address);
op_pk_fd_get(pkptr,1,&burst_length);
op_pk_fd_get(pkptr,4,&offset_time);
op_pk_fd_get(pkptr,3,&packets_number);
total_packet_count = total_packet_count+ packets_number;

/*calculate the service start and service stop time of the received header*/

header_receive_time = op_sim_time();
current_header_service_start_time = header_receive_time+offset_time;
current_header_service_stop_time = current_header_service_start_time +burst_length;

/*set the required service start,service stop, source field of header*/

op_pk_fd_set(pkptr,5,OPC_FIELD_TYPE_INTEGER,source,50);
op_pk_fd_set(pkptr,6,OPC_FIELD_TYPE_DOUBLE,current_header_service_start_time ,50);
op_pk_fd_set(pkptr,7,OPC_FIELD_TYPE_DOUBLE,current_header_service_stop_time ,50);
size_list0= prg_list_size(lptr[destination_address]);
if(prg_list_size(lptr[destination_address]) == 0)
    {
        duplicate_pkptr = op_pk_copy(pkptr);
        op_pk_send(duplicate_pkptr,destination_address);
        prg_list_insert(lptr[destination_address],pkptr,0);

    }
else
    {

        for(iii=0;iii<prg_list_size(lptr[destination_address]);iii++)
                {
                    previous_pkptr = prg_list_access(lptr[destination_address],iii);
                    op_pk_fd_get(previous_pkptr,6,&previous_header_service_start_time);
                    op_pk_fd_get(previous_pkptr,7,&previous_header_service_stop_time);

                if(current_header_service_start_time>previous_header_service_stop_time||
                current_header_service_stop_time<previous_header_service_start_time)
                        header_lost_count0 = header_lost_count0;
                else
                        header_lost_count0++;
                }

        if(header_lost_count0 == 0)

              {
                duplicate_pkptr = op_pk_copy(pkptr);
                op_pk_send(duplicate_pkptr,destination_address);
                prg_list_insert(lptr[destination_address],pkptr,PRGC_LISTPOS_TAIL);

                op_pk_fd_get(pkptr,3,&lost_packets_number);

                }
```

```
else
        {

        loss_burst_count++;

        op_pk_fd_get(pkptr,3,&lost_packets_number);
        loss_packet_count = loss_packet_count+lost_packets_number;
        burst_loss_probability = (loss_burst_count/total_burst_count);
        packet_loss_probability = (loss_packet_count/total_packet_count);
        op_stat_write(loss_probability_burst_gsh,burst_loss_probability);
        op_stat_write(loss_probability_packet_gsh,packet_loss_probability);
        op_pk_destroy(pkptr);
        header_lost_count0 = 0;
        }
}

if(prg_list_size(lptr[destination_address])>0)
        {
        for(ii=0;ii<prg_list_size(lptr[destination_address]);ii++)
                {
                previous_pkptr = prg_list_access(lptr[destination_address],ii);
                op_pk_fd_get(previous_pkptr,7,&previous_header_service_stop_time);
                );
                if(previous_header_service_stop_time<op_sim_time())
                        {
                        destroy_pkptr = prg_list_remove(lptr[destination_address],ii);
                        op_pk_destroy(destroy_pkptr);

                        }
                }
        }
```

## B.3 Core scheduler with reservation approach

```
//////////////////////////////////////////////////////////// Forwarding state ////////////////////////////////////////////////////////////


if(pk_num == 0)
        {
        for(i=0;i<8;i++)
                lptr[i]= prg_list_create();
        pk_num++;
        }

/*get a packet from the stream*/

pkptr= op_pk_get(op_intrpt_strm());
propagation_delay = (op_sim_time()-op_pk_creation_time_get(pkptr));

/* extract all its information from the header */
```

```
source=op_intrpt_strm();
op_pk_fd_get(pkptr,0,&destination_address);
op_pk_fd_get(pkptr,1,&burst_length);

/*calculate the service start and service stop time of the received header*/

current_time = op_sim_time();

/*set the required service start,service stop, source field of header*/

op_pk_fd_set(pkptr,6,OPC_FIELD_TYPE_INTEGER,source,50);
if(prg_list_size(lptr[destination_address])<2)
  {
     current_header_service_start_time = (current_time + 2*propagation_delay);
     current_header_service_stop_time=(current_header_service_start_time +burst_length);
     time_to_send_burst = (current_time+propagation_delay);
     op_pk_fd_set(pkptr,7,OPC_FIELD_TYPE_DOUBLE,current_header_service_start_time ,50);
     op_pk_fd_set(pkptr,8,OPC_FIELD_TYPE_DOUBLE,current_header_service_stop_time ,50);
     op_pk_fd_set(pkptr,9,OPC_FIELD_TYPE_DOUBLE,time_to_send_burst ,50);
     duplicate_pkptr = op_pk_copy(pkptr);
     op_pk_send(duplicate_pkptr,source);
     prg_list_insert(lptr[destination_address],pkptr,PRGC_LISTPOS_TAIL);

        }
else
        {

             for(iii=1;iii<prg_list_size(lptr[destination_address]);iii++)
                 {
                  ii = (iii-1);
                 previous_pkptr_0 = prg_list_access(lptr[destination_address],ii);
                 previous_pkptr_1 = prg_list_access(lptr[destination_address],iii);
                 op_pk_fd_get(previous_pkptr_0,7,&previous_header_service_start_time_0);
                 op_pk_fd_get(previous_pkptr_0,8,&previous_header_service_stop_time_0);
                 op_pk_fd_get(previous_pkptr_1,7,&previous_header_service_start_time_1);
                 op_pk_fd_get(previous_pkptr_1,8,&previous_header_service_stop_time_1);
                 time_difference=(previous_header_service_start_time_1
                 previous_header_service_stop_time_0);
                 if(time_difference>burst_length)

                   {

                   if((current_time+2*propagation_delay)<=previous_header_service_start_time_1)

                     {
                       current_header_service_start_time = previous_header_service_stop_time_0;
                      current_header_service_stop_time=(current_header_service_start_time
                       +burst_length);
                     time_to_send_burst = (current_header_service_stop_time-propagation_delay);
                    op_pk_fd_set(pkptr,7,OPC_FIELD_TYPE DOUBLE,current_header_service_start_time
                    ,50);
                    op_pk_fd_set(pkptr,8,OPC_FIELD_TYPE_DOUBLE,current_header_service_stop_time
                    ,50);
                     op_pk_fd_set(pkptr,9,OPC_FIELD_TYPE_DOUBLE,time_to_send_burst ,50);
                     duplicate_pkptr = op_pk_copy(pkptr);
                    op_pk_send(duplicate_pkptr,source);
```

```
            prg_list_insert(lptr[destination_address],pkptr,iii);
            goto sos;
            }


    else
        {
        current_header_service_start_time = (current_time+2*propagation_delay);
        current_header_service_stop_time = (current_header_service_start_time +burst_length);
        if(current_header_service_stop_time<=previous_header_service_start_time_1)
          {
          time_to_send_burst = (current_time-propagation_delay);
          op_pk_fd_set(pkptr,7,OPC_FIELD_TYPE_DOUBLE,current_header_service_start_time
,50);
          op_pk_fd_set(pkptr,8,OPC_FIELD_TYPE_DOUBLE,current_header_service_stop_time ,50);
          op_pk_fd_set(pkptr,9,OPC_FIELD_TYPE_DOUBLE,time_to_send_burst ,50);
          duplicate_pkptr = op_pk_copy(pkptr);
          op_pk_send(duplicate_pkptr,source);
          prg_list_insert(lptr[destination_address],pkptr,iii);
          goto sos;
                          }
                  }
              }

          header_lost_count0++;

          }

if(header_lost_count0>0)
    {

    last_pkptr = prg_list_access(lptr[destination_address],PRGC_LISTPOS_TAIL);
    op_pk_fd_get(last_pkptr,7,&last_pkptr_service_start_time);
    op_pk_fd_get(last_pkptr,8,&last_pkptr_service_stop_time);
    if((current_time+2*propagation_delay)<=last_pkptr_service_stop_time)

      {
      current_header_service_start_time = last_pkptr_service_stop_time ;
      current_header_service_stop_time = (current_header_service_start_time +burst_length);
      time_to_send_burst = (current_header_service_start_time-propagation_delay);
      op_pk_fd_set(pkptr,7,OPC_FIELD_TYPE_DOUBLE,current_header_service_start_time ,50);
      op_pk_fd_set(pkptr,8,OPC_FIELD_TYPE_DOUBLE,current_header_service_stop_time ,50);
      op_pk_fd_set(pkptr,9,OPC_FIELD_TYPE_DOUBLE,time_to_send_burst ,50);
      duplicate_pkptr = op_pk_copy(pkptr);
      op_pk_send(duplicate_pkptr,source);
      prg_list_insert(lptr[destination_address],pkptr,PRGC_LISTPOS_TAIL);

      }

    if((current_time+2*propagation_delay)>last_pkptr_service_stop_time)
        {

        current_header_service_start_time = (current_time+2*propagation_delay);
        current_header_service_stop_time = (current_header_service_start_time +burst_length);
        time_to_send_burst = (current_header_service_start_time-propagation_delay);
        op_pk_fd_set(pkptr,7,OPC_FIELD_TYPE_DOUBLE,current_header_service_start_time ,50);
```

```
            op_pk_fd_set(pkptr,8,OPC_FIELD_TYPE_DOUBLE,current_header_service_stop_time ,50);
            op_pk_fd_set(pkptr,9,OPC_FIELD_TYPE_DOUBLE,time_to_send_burst ,50);
            duplicate_pkptr = op_pk_copy(pkptr);

        op_pk_send(duplicate_pkptr,source);
        prg_list_insert(lptr[destination_address],pkptr,PRGC_LISTPOS_TAIL);

        }

    }
}

sos:

header_lost_count0 = 0;
if(prg_list_size(lptr[destination_address])>0)
            {
            for(ii=0;ii<prg_list_size(lptr[destination_address]);ii++)
                {
                previous_pkptr = prg_list_access(lptr[destination_address],ii);
                op_pk_fd_get(previous_pkptr,8,&previous_header_service_stop_time);
                if(previous_header_service_stop_time<op_sim_time())
                    {
                    destroy_pkptr = prg_list_remove(lptr[destination_address],ii);
                    op_pk_destroy(destroy_pkptr);

                    }

                }
            }
```