**İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY**

# PERFORMANCE COMPARISON AND TRAFFIC ANALYSIS IN OPTICAL BURST-SWITCHED NETWORKS

**M.Sc. Thesis by**

**Burak KANTARCI, B.Sc.**

**DEPARTMENT : COMPUTER ENGINEERING**

**PROGRAMME: COMPUTER ENGINEERING**

**JANUARY 2005**

**İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY**

# PERFORMANCE COMPARISON AND TRAFFIC ANALYSIS IN OPTICAL BURST-SWITCHED NETWORKS

**M.Sc. Thesis by**

**Burak KANTARCI, B.Sc.**

**(504021523)**

**Date of submission : 27 December 2004**

**Date of defence examination: 18 January 2005**

**Supervisor (Chairman): Assoc. Prof. Dr. Sema OKTUĞ**

**Members of the Examining Committee Prof.Dr. Ercan TOPUZ (İ.T.Ü )**

**Assist.Prof.Dr. Ayşegül GENÇATA (İ.T.Ü.)**

**JANUARY 2005**

**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

# OPTİK ÇOĞUŞMA ANAHTARLAMALI AĞLARDA BAŞARIM KARŞILAŞTIRMASI VE TRAFİK ANALİZİ

**YÜKSEK LİSANS TEZİ**

**Müh. Burak KANTARCI**

**Tezin Enstitüye Verildiği Tarih : 27 Aralık 2004**

**Tez Danışmanı :** **Doç.Dr. Sema OKTUĞ**

**Diğer Jüri Üyeleri** **Prof.Dr. Ercan TOPUZ (İ.T.Ü)**

**Yrd.Doç.Dr. Ayşegül GENÇATA (İ.T.Ü)**

**OCAK 2005**

**ACKNOWLEDGEMENTS**

First of all, I would like to thank to my advisor Assoc.Prof.Dr. Sema OKTUĞ for her endless and encouraging support at every phase of my work, also for guiding me with her valuable suggestions, and taking into care all of the components making up the thesis for the contribution of the study. It is obvious that working together is really a great chance and pleasure.

The content of this thesis study is also improved during TUBITAK-CNRS joint project "Performance Study of Optical Networks and its Traffic Modelling". Therefore, I have to thank to all of the project members in INT (Institute National des Telecommunications) especially Dr.Tülin ATMACA, and Daniel POPA for their valuable suggestions.

I also have to thank my family for their great support, encouraging treatment towards me during my studies since at some states I really needed comfort and a little bit more tolerance which I gained from them.

Besides these, Pınar SARISARAY also should gain my special thanks because of her valuable and motivative attitude especially when I required a lot during this work.

Finally, I would like to thank and wish great success in his studies to my roommate A.Çağatay TALAY who was the one to bear me everyday at the preparation state of this thesis…

Burak KANTARCI

December, 2004

## INDEX

**Page Number:**

**ABBREVIATIONS**

| | |
|---|---|
| **OBS** | **:** Optical Burst Switching |
| **OPS** | **:** Optical Packet Switching |
| **JET** | **:** Just-Enough-Time |
| **JIT** | **:** Just-In-Time |
| **WR** | : Wavelength Routing |
| **BHC** | **:** Burst Header Cells |
| **DB** | **:** Data Burst |
| **LAUC-VF** | **:** Latest Available Unused Channel with Void Filling |
| **G-LAUC-VF** | **:** Generalized LAUC-VF |
| **First-Fit VF** | **:** First FitVoid-Filling |
| **VF** | **:** Void-Filling |
| **DFMOC-VF** | **:** Delay First Minimum Overlap Channel with Void Filling |
| **MAX_DELAY** | : Maximum delay value that a burst can be delayed |
| **SIM_END** | **:** End of the simulation |
| **WSIZE** | **:** Size of a window in Group Scheduling |
| **SCB** | **:** Single Class Burst |
| **CCB** | **:** Composite Class Burst |
| **TIIP** | **:** Traditional Input IP Port |
| **EMOP** | **:** Electrical Matrix Output Port |
| **BA** | **:** Burst Assembler |
| **DP** | **:** Drop Policy |
| **DDP** | **:** Deflect and Drop Policy |
| **SDP** | **:** Segment and Drop Policy |
| **SDDP** | **:** Segment, Deflect and Drop Policy |
| **LAUT** | **:** Latest Available Unscheduled Time |
| **SS** | **:** Small Size Bursts (0-25μs) |
| **MS1** | **:** Medium Size Bursts-1(25μs-50μs) |
| **MS2** | **:** Medium Size Bursts-2(50μs-75μs) |
| **LS** | **:** Large Size Bursts-1(75μs-100μs) |
| **CCIP** | **:** Control Channel Input Port |
| **LS** | **:** Large Size Bursts-1(75μs-100μs) |
| **EMIP** | **:** Electrical Matrix Input Port |
| **OR** | **:** Optical Receiver |

**TABLE LIST**

**Page Number:**

**SYMBOLS**

| | |
|---|---|
| **Is** | : Starting time of a void |
| **Ie** | : End time of a void |
| **a** | : Arrival time of a burst |
| **Δ** | : Offset time for a data burst |
| **l** | : Length of a burst in time domain. |
| $\mathbf{W_{start}}$ | : Start time of a window |
| $\mathbf{W_{stop}}$ | : End time of a window |
| **Tp** | : Deadline for collecting bursts in Group Scheduling |
| **λ** | : Burst arrival rate |
| **TTh** | : Time Threshold |
| **STh** | : Size Threshold |

## FIGURE LIST

**Page Number:**

**PERFORMANCE COMPARISON AND TRAFFIC ANALYSIS IN OPTICAL BURST-SWITCHED NETWORKS**

## SUMMARY

As a result of a huge bandwidth demand in the Internet, Wavelegth Division Multiplexing (WDM) has been being employed in order to offer a bandwidth of terabits per second (Tbps) which is partitioned into wavelength channels of bandwidth of gigabits per second (Gbps). The initial switching paradigms were Wavelength Routing and Optical Packet Switching. Optical Burst Switching (OBS) is a novel switching paradigm in all-optical networks.

In Optical Burst Switching, at the ingress nodes of the network, the incoming IP packets that are routed to the same destination and that are of the same service class are assembled together to form an optical burst. An optical burst consists of a header and a payload. The header carries the control data such as routing, wavelength assignment, arrival time, destination address and service class. The burst header cell (BHC) is sent an offset time before the data burst (DB) and attempts to reserve available bandwidth through the path before the arrival of the payload. While scheduling the bursts, contention may occur in the optical links and an efficient switching and scheduling policy is required.

In this work, we use a 14-node NSFNET topology to simulate an OBS network. We focus on the OBS techniques that have been currently in the literature. Initially we make a general categorization of the switching and scheduling techniques in OBS based on their channel utilization policy, and analyze their performance in terms of loss rate and delay. Then we concentrate on the most practical OBS techniques by simulating them and comparing their performance. These techniques are Horizon, Latest Available Unused Channel with Void Filling (LAUC-VF), First-Fit with Void

Filling (First-Fit VF), Group Scheduling, and Delay First Minimum Overlap Channel with Void Filling (DFMOC-VF) Segmentation technique.

The simulation results under Poisson traffic show that the horizon scheduling, which is a non-void filling technique works as the fastest technique so the scheduling delay in the horizon algorithm seems as the least among all. However, even in a network load of 4 Erlang, the horizon algorithm causes a significantly higher burst loss rate among all the techniques. This result is caused by not utilizing the idle intervals which we call the voids. Besides this we observe that the performance of the void filling techniques LAUC-VF and First-Fit VF lead to a significantly lower loss rate in average network load while they bring an increase in scheduling delay as a result of taking voids into account when attempting to schedule the bursts. We also observe that Group Scheduling gives better burst loss rate since it collects the BHCs for a period and attempts to schedule the maximum number of collected bursts rather than scheduling the arriving bursts immediately. DFMOC-VF performs slightly better than Group Scheduling, and decreases the burst loss at a rate of 1-2% since if a burst contends with a previously scheduled burst it still attempts to schedule the non-contending segment of it. However both, Group Scheduling and DFMOC-VF lead to higher scheduling delays when compared to the other techniques.

The performance of the OBS techniques is studied under Poisson and self-similar traffic. Self-similar traffic with Hurst parameter 0.9 is generated at each ingress node. Two burst assembly schemes which are based on a time threshold mechanism and a time-and-queue threshold mechanism are employed. We observe that the relative performance of the techniques in terms of loss rate under self-similar traffic is the same as in Poisson traffic. It is observed that the incoming self-similarity with H = 0.9 is reduced to 0.50-0.53 by the employment of each technique. It is also shown that the time threshold based burst assembly leads to a slightly better self-similarity by increasing the time threshold.

The scheduling techniques are studied on the 14-node NSFNET topology, and it is observed that the techniques that have better utilization increase the self-similarity of the burst traffic.

**OPTİK ÇOĞUŞMA ANAHTARLAMALI AĞLARDA BAŞARIM KARŞILAŞTIRMASI VE TRAFİK ANALİZİ**

## ÖZET

İnternet üzerinde yüksek bant genişliği gereksiniminin artmasıyla birlikte, dalgaboyu bölümlemeli çoğullama yöntemine dayalı yönlendirme yapan tam optik ağlar (WDM) tarafından saniyede terabitler (Tbp/s) düzeyinde iletim hızı sağlanarak bu gereksinim karşılanmış oldu. Saniyede terabitler düzeyinde iletim hızına olanak sağlayan optik lifler, her biri saniyede gigabitler düzeyinde (Gbp/s) bant genişliğine sahip dalgaboyu kanallarına ayrıştırılmışlardır. Optik ağlarda ilk önerilen anahtarlama yöntemleri dalgaboyu yönlendirmeli anahtarlama (WR) ve optik paket anahtarlamadır (OPS). Optik Çoğuşma Anahtarlama (OBS), tam optik ağlarda yeni bir anahtarlama yöntemi olarak sunulmuştur.

Optik çoğuşma anahtarlamada, optik ağın girişindeki yönlendirici, kendisine gelen IP paketlerini gidecekleri hedef düğümler ve ait oldukları hizmet sınıflarına göre gruplayarak birarada toplar ve bir optik çoğuşma oluşturur. Bir optik çoğuşma, bir çoğuşma başlığı (BHC) ve çoğuşma verisinden (DB) oluşur. Çoğuşma başlığı, yönlendirme, dalgaboyu atama, varış zamanı, varış adresi ve servis sınıfı bilgilerini taşır. Çoğuşma başlığı, çoğuşma verisinden belirli bir göreli konum süresi kadar önce gönderilir ve verinin varış zamanından önce, yol üzerindeki uygun bant genişliklerini rezerv etmeye çalışır. Çoğuşma verilerinin sıralı gelmemesinden ötürü, optik hatlarda kimi kaynaklar için çekişme sözkonusu olduğu durumlar için optik çoğuşma anahtarlamalı ağlarda etkin bir anahtarlama ve sıralama algoritmasına gereksinim vardır.

Yaptığımız çalışmada, optik çoğuşma anahtarlamalı bir ağ benzetimi için 14 düğümlü NSFNET topolojisi kullanarak, şimdiye kadar önerilmiş anahtarlama ve sıralama teknikleri üzerine odaklanmaktayız. İlk olarak, bu teknikler için bir

kategorizasyon oluşturup her birinin başarımını farklı başarım kriterleri (çoğuşma kaybı ve sıralama gecikmesi) altında analiz ediyoruz. Bunun ardından, pratikte en uygulanabilir görünen sıralama ve anahtarlama teknikleri üzerinde yoğunlaşarak, bu teknikleri kendi oluşturduğumuz simulasyon ortamında çalıştırarak başarımlarını karşılaştırıyoruz. Horizon, Boşluk Doldurmalı En Son Uygun Kanal (LAUC-VF), Boşluk Doldurmalı İlk Uygun (First-Fit VF), Grup Tabanlı Sıralama ve Geciktirerek En Düşük Örtüşen Kanalda Boşluk Doldurma (DFMOC-VF) segmentasyon tekniklerini kullanmaktayız.

Poisson trafik altında alınan simulasyon sonuçları, boşluk doldurmayan Horizon sıralama tekniğinin en hızlı çalıştığını, bu durumda da sıralamadan kaynaklanan gecikmeler karşılaştırıldığında en düşük değere sahip olduğunu görmekteyiz. Bunun yanında aynı teknik, 4 Erlang gibi ortalama bir toplam yük altında bile büyük çoğuşma kayıplarına neden olmaktadır. Bunun nedeni ise, Horizon algoritmasının, kullanılmayan zaman aralıklarını yeni gelen çoğuşmalar için rezerv etmeye çalışmamasıdır. Buna ek olarak, boşluk doldurmalı LAUC-VF ve First-Fit VF tekniklerinin başarımlarını incelediğimizde, ortalama toplam yük altında belirgin bir şekilde daha düşük çoğuşma kaybıyla sonuçlanabildiğini, ancak sözkonusu tekniklerin, tüm rezerv edilmemiş zaman aralıklarını dikkate almalarından ötürü sıralama gecikmesini de ciddi anlamda arttırdığını görüyoruz. Grup Tabanlı Sıralama ile segmentasyon tabanlı DFMOC-VF sıralama teknikleri ise çoğuşma kaybı açısından daha iyi sonuç vermektedir. Bunun nedeni ise, grup tabanlı sıralamada, çoğuşma başlıkları, geldikleri anda işlenmeyerek, belirli bir periyod süresince toplanması ve toplanan başlıkların hepsi birden değerlendirilerek en yüksek sayıda çoğuşma için kaynak rezerv edilmesine çalışılmasıdır. Benzer şekilde, segmentasyon tabanlı sıralamada, iki çoğuşma aynı zaman aralığı için çekişiyorsa yönlendiriciye sonradan varan çoğuşmanın çekimeye neden olmayan segmenti dalgaboyu kanalı üzerinde sıralanmaktadır.

OBS anahtarlama ve sıralama tekniklerinin başarımları Poisson ve özbenzeşimli trafikler altında sınanmaktadır. Özbenzeşimli trafik, kenar düğümlerde 0.9 Hurst parametresi ile yaratılmaktadır. Bununla birlikte, her kenar düğümünde, zaman eşiği tabanlı ve melez (zaman eşiği ve uzunluk eşiği) çoğuşma oluşturma teknikleri uygulamaktayız. Sıralama ve anahtarlama tekniklerinin birbirlerine göre başarım sıralarının değişmediğini, ancak çoğuşma kayıp oranlarının, çoğuşma oluşturma

süresine bağlı olarak azaldığını göstermekteyiz. Kenar düğümlere 0.9 Hurst parametresi ile gelen trafiğin özbenzeşimliliği, girişlerde her iki çoğuşma oluşturma tekniğinde de Hurst parametresi 0.50-0.53 arasında değişen özbenzeşimliliğe düşürülmektedir. Ayrıca, zaman eşiği tabanlı çoğuşma oluşturma tekniği, zaman eşiği arttıkça daha düşük özbenzeşimliliğe neden olmaktadır. Bunun yanında, zaman eşiği tabanlı çoğuşma oluşturmada çıkışlarda çoğuşma kayıp oranları ve özbenzeşimlilik, melez çoğuşma oluşturma tekniği uygulandığı zamana göre daha düşük olmasına karşın, çoğuşma oluşturma için girişteki düğümlerde harcanan ortalama süre melez tekniğe göre çok daha fazladır.

OBS sıralama ve anahtarlama teknikleri 14 düğümlü NSFNET topoloji altında sınanmış ve, dalgaboyu kanalı kullanımını arttıran tekniklerin çoğuşma trafiğinin özbenzeşimliliğini arttırdığı gözlenmiştir.

# 1. INTRODUCTION

As a result of the increase in the bandwidth demand of the next generation Internet applications, optical WDM networks appeared as the best solution in order to offer bandwidth values in Tb/s by partitioning this bandwidth into a number of gigabits per second wavelength channels[1].

The common paradigms in optical switching are wavelength routing, optical packet switching (OPS), and optical burst switching (OBS) [2]. In a wavelength routed network, first an all optical path is established, and data transmission is done through the dedicated lightpath. In wavelength routed networks, wavelength converters are located at intermediate routers and permit switching from one wavelength to another within the lightpath [3]. The disadvantage of wavelength routing is to dedicate the whole wavelength for the flows from a specific source to destination for a period of time without knowing the utilization of the wavelength capacity.

OPS and OBS are suggested as an alternative to wavelength routing. In OPS, a packet is seperated into two partitions as header and data. Header is converted to electronic form at the nodes, and processed in the electronical domain in order to obtain the routing information of the packet. Data is carried in all-optical form based on the information obtained from its header. However, in OPS, in order to achieve optical transparency, optical buffers and/or optical logic are required. Unless the optical logic and/or the optical buffers are employed, there will be an overhead of optical-to-electronical and electronical-to-optical conversion [2].

OBS has similar properties both as OPS and wavelength routing. Similar to wavelength routing, the data is neither processed nor buffered at the intermediate nodes. However, the data and the header are processed seperately as in OPS [2]. In order to prevent optical buffering, OBS uses wavelength converters [4].

The major concern in OBS is the burst loss rate since the data bursts neither arrive one after another nor they can be buffered by using an optical RAM [28]. In order to prevent high burst loss rate, an effective burst scheduling algorithm should be

implemented so that a lower burst loss rate is obtained as fast as possible related to the time complexity of the algorithm.

Another challenging concern in OBS is the policy of aggregating IP packets into bursts at the edge router of the network. The reason of this is the burst assembly policy is said to affect the incoming traffic characteristics of the network [30] since it affects the aggregation time and the size of the bursts.

In this work, we analyze the performance of the optical burst swithcing techiques in terms of loss rate and performance under various traffic types that are Poisson and self-similar. Upon comparing the performance of OBS techniques we get in through the effect of burst assembly mechanisms on the incoming traffic to the OBS network. Besides these, we also analyze the effect of OBS techniques on the traffic characteristics. We show that the void-filling techniques perform better in each type of traffic in terms of burst loss rate while non-void filling techniques bring an advantage of scheduling the bursts on the wavelength channels faster. Beyond these, we show that grouping the bursts and using segmentation based policies to switch the bursts decreases the loss rate while they cause an overhead of scheduling time complexity. Another concern in this work shows that, implementing a burst assembly technique at the ingress of the network shapes the traffic by limiting the burst size and waiting time. Shaping traffic decreases the degree of self-similarity in short term behaviour and leads to a slightly lower burst loss rate of 5%.

The rest of the thesis is organized as follows: In Section 2, we present the architectural issues in OBS networks including the signaling and node architectures. The OBS channel scheduling and contention resolution problem are defined in Section 3. We examine the recently proposed OBS scheduling and contention resolution techniques by comparing their performances in terms of loss rate and scheduling time complexity. In Section 4, the burst assembly problem in OBS and the currently proposed assembly techniques are presented. In Section 5, we present our simulation environment and OBS network simulator in detail. Besides these, we define the scheduling techniques and the burst assembly techniques employed in our simulations. In Section 6, the results obtained in our simulations are discussed in a comparative way. Finally, conclusions and future work is given in Section 7.

## 2. OPTICAL BURST SWITCHING

The evolution of WDM networks is given as shown in Figure 2.1 As it is seen from Figure 2.1 [12]. The figure shows that, as the optical WDM networks evolve to OPS and OBS, the all-optical transfer and also the functionality of the optical network has increased. Besides this optical packet/burst switched networks brought the advantage of eliminating the optical-to-electronical and electronical-to-optical conversion of the user data. As it is given in Section 1, OBS combines the advantages of wavelength routing and OPS while eliminating the disadvantages.

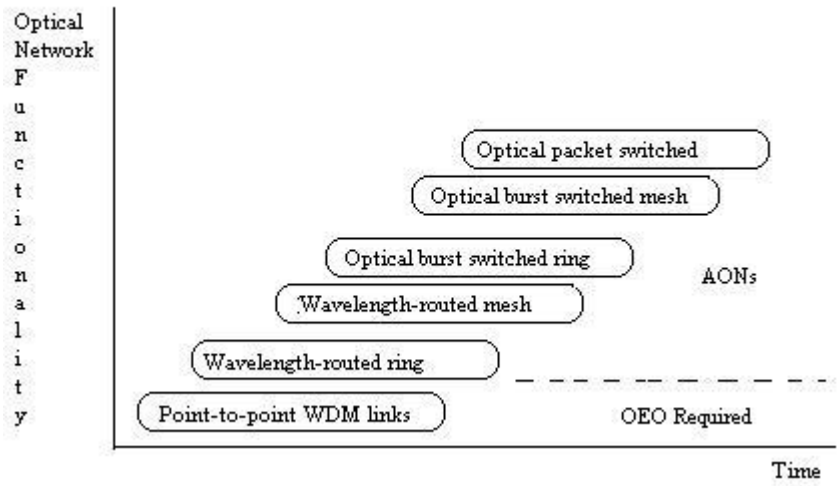



Figure 2.1. Evolution of optical WDM networks [12].

In OBS, incoming IP packets that are destined to the same destination and of the same service class are assembled into optical bursts at the ingress routers. Upon assembling the burst the optical burst is seperated into two partitions, the burst header cell (BHC) and the data burst (DB). BHC carries the control information which is, the routing information, the length of the burst in time domain, and the offset time and is transmitted from a different transmission channel an offset time before its corresponding DB [31]. The DBs follow their BHCs without waiting for an acknowledgement just an offset time later which is carried within the BHC [32].

Therefore, OBS seperates the data plane and the control plane in the network architecture completely [29].

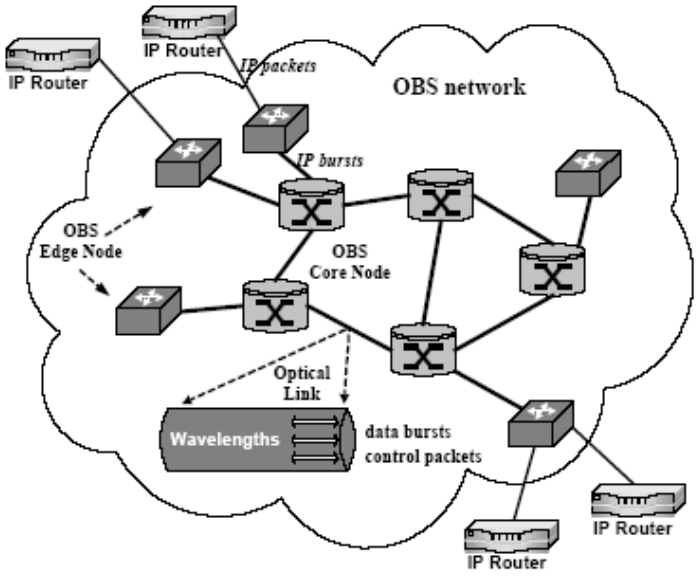



Figure 2.2. An OBS network architecture [46].

As it is seen from Figure 2.2, header and data are transmitted through different channels. Data Burst (DB) has to be carried over optical channels while the control burst (BHC) can be carried on any form [27] such as optical, electronical, wireless etc. Each intermediate node converts the BHCs into electronical form in order to draw information about the data burst. Each BHC includes: routing address information together with the wavelength channel assignment, and reservation of the available time intervals for the DBs which are carried in the optical form, the offset time and the size of the burst in terms of time duration [5].

## 2.1 Routing in OBS

In most cases it is possible to use a hop-by-hop basis routing consisting of just a fast table lookup in order to determine the next-hop [12]. We implement source routing at each OBS node by employing Dijkstra's shortest path algorithm, and provide the BHCs to carry the whole path with them when they are generated.

An alternative routing scheme is proposed which is derived from MPLS and called Labeled Optical Burst Switching (LOBS) in [33]. In LOBS, the bursts that belong to two or more Label Switching Paths (LSP) can be aggregated, and they are assigned a label. The label information is also carried by the BHCs. By using LOBS, the

forwarding of the bursts is speeded up. The bursts that belong to the same wavelength are aggregated if and only if they do not contend so in LOBS wavelength conversion is not used.

An explicit routing scheme which can also be used in OBS is implemented for optical networks in [34]. In explicit routing scheme an MPLS-like structure is used where the wavelengths stand for the labels. In MPLS, the incoming packets are mapped into a Forward Equivalence Class (FEC) at the ingress routers of the network. By using the contents of the labels the switches decides how to forward the packets. At each Label Switching router (LSR), the label is stripped off and a new label is assigned in order to inform the next hop how to forward the packet [35]. Lee et al. employs this technique on optical networks such that the wavelengths stand for the labels in MPLS. By using a GMPLS label mapping, a connection is established between an input port-input wavelength couple and an output port-output wavelength couple. Traffic Engineering is used together with GMPLS scheme in order to satisfy the QoS requirements of the connections when establishing routes such as bandwidth and delay requirement [34].

## 2.2 Wavelength Assignment in OBS

Wavelength assignment in OBS networks can be categorized as “with” and “without” wavelength conversion. In case of without wavelength conversion, from source to destination, a wavelength is selected and assigned to the connection [3]. During the transmission the bursts are always switched on that wavelength. If two bursts contend for the same wavelength, one of them has to be dropped.

In case of wavelength assignment with wavelength converters, each OBS node has wavelength converters, and when two bursts contend for the same wavelength in the same time interval the converter should switch the burst from the wavelength at the input to another wavelength In practice, all of the nodes do not have wavelength converters in the optical domain. Besides these, the number of wavelength converters are less than the number of wavelengths [12].

## 2.3 Signaling in OBS

There are two signaling schemes in OBS: Just-in-Time (JIT) [36], and Just-Enough-Time (JET) [18, 37]. In JIT, the control packet (BHC) is sent on a seperate wavelength channel and is followed by its corresponding data burst. Upon the arrival of control packet, an available wavelength is attempted to be reserved for the incoming data burst. When the data burst arrives, it is delayed by employing fiber delay lines (FDLs) until the end of the processing of the BHC. If the header succeeds in reserving a wavelength, the data burst is scheduled on that wavelength, otherwise it is dropped.

In JET, a control packet is sent initially through a control channel. It reserves a wavelength channel for the time that is equal to the length of its payload and that starts with the arrival of BHC plus the offset time of its data. After the offset time, the payload of the BHC is sent through a data channel [18]. JET is more preferrable signaling protocol in OBS networks since it does not bring the necessity of optical buffering or using fiber delay lines at each hop. As it is seen in Figure 2.3, the reservation starts at the time when the payload arrives so the delay scheme is employed by exluding the arrival time of the BHC. Besides these, since the burst length is carried within the BHC, the reservation is close-ended. In [30], it is stated that keeping the reservation close-ended allows to determine an effective utilization of the available bandwidth.

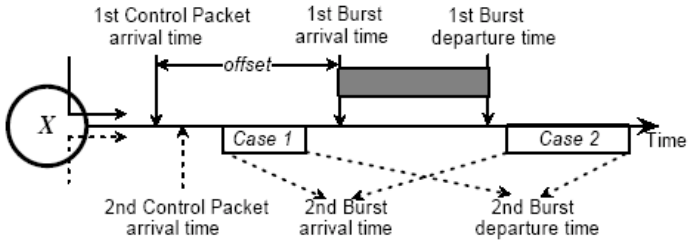



Figure 2.3. Just-Enough-Time (JET) protocol [30]

As it is seen from Figure 2.4, the residual offset ($R_j$) time at the intermediate node *j* of the burst decreases at each hop due to the processing delay of the header. Residual offset time is the time remaining before the data burst arrives [45]. Data is sent upon the completion of the processing of the BHC. The initial offset time is represented by *o* which shows the duration the source node will wait after sending the BHC. The average processing time per hop for a BHC is represented by *d*.

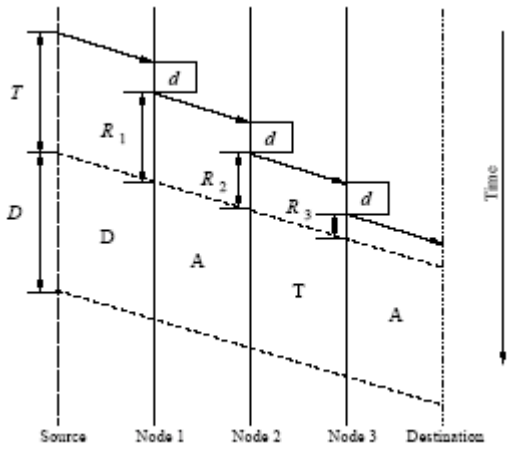



Figure 2.4. Just-Enough-Time (JET) protocol [45].

Then, the residual offset time is shown as follows:

$$R_j = o - (j - 1)*d. \tag{2.1}$$

## 2.4 OBS node structure

OBS node structure can be examined in two groups, as the OBS edge node structure and OBS core node structure since the functionalities of these two types of routers are different from each other. Initially, we focus on OBS edge nodes, then we go through OBS core nodes.

In order to define the node structure, it is useful to focus on the functions performed by an OBS edge router. These functions can be listed as follows [46]:

- Receiving IP packets from outside the OBS domain and forward IP packets to the routers following the egress node of the domain.
- Burst assembly by the aggregation of IP packets that are routed to the same destination and are of the same QoS class.
- Construction of a BHC for each burst and determination of an offset time for its corresponding DB.
- Conversion of electrical signal on which the IP packets are carried into optical form.
- Conversion of the bursts in optical format into electrical format if the node is an egress node and demultiplexing the IP packets from the burst.

- Run a source routing algorithm to forward the assembled burst.

Based on these, the hardware implementation of an OBS edge node structure is shown in Figure 2.5. If the node is assumed to behave as an ingress node, the electrical signal which carries the IP packets reaches the router through the traditional IP input port (TIIP) and switched to the electrical matrix output port (EMOP) in order to assemble burst. A burst header is generated in order to carry the control information. Such control information is constructed in the Forwarding Engine. After the route of the burst is determined, the control packet is converted into optical format by the electronic-to-optical converters (E/O). EMOPs switch the IP packets to the corresponding outlets that are connected to burst assemblers (BA). Burst assembly is performed in these units. Upon burst assembly, the electrical signal carrying the burst is converted into optical format by the electronic-to-optical converters (E/O) that are named as concentrators in [5]. Upon E/O conversion, the burst header and data burst are multiplexed and forwarded into the OBS domain through the optical transmitter [46].

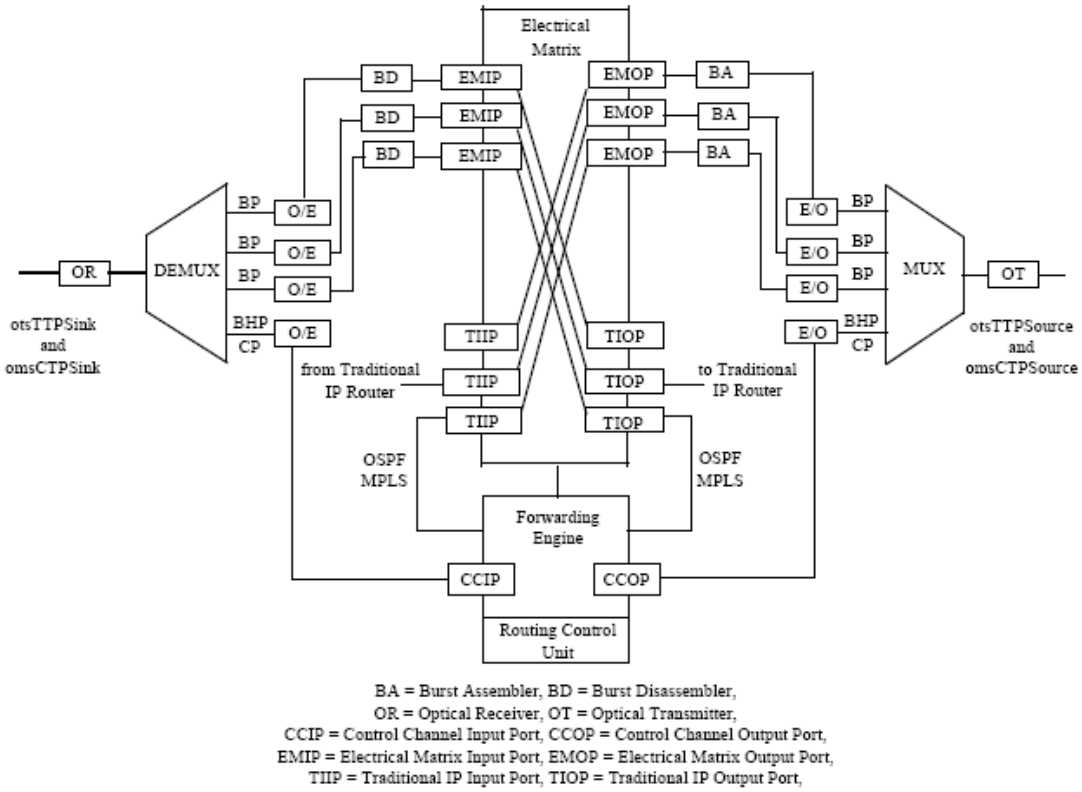



Figure 2.5. A general framework for an OBS edge router [46]

If the node is supposed to behave as an egress node, the optical signal carrying the bursts enters the node through optical receivers (OR). Bursts are seperated into data

bursts and corresponding burst headers. Headers are converted into electrical form and processed in Forwarding Engine and transmitted through control channel input port (CCIP) in order to determine the next hop. Upon determining the routing information, the header is switched to traditional IP output port (TIOP). The data packets are converted into electrical form and switched to the corresponding outlets through electrical matrix input port (EMIP). Finally all of the packets are in electrical format and forwarded to the out of the OBS domain through TIOP [46].

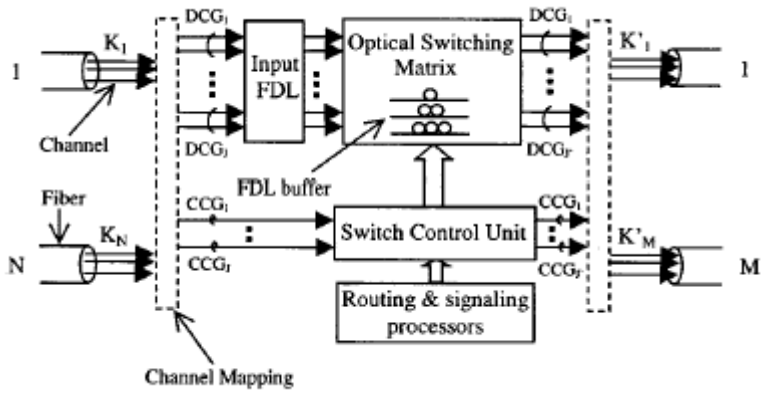



Figure 2.6. A general framework for an OBS core router [38].

An ( *N* * *M*) OBS core node structure is illustrated in Figure 2.6. The main components of the node is input fiber delay lines (FDLs), an optical switching matrix, a switch control unit, routing and signalling processors. Data bursts (DBs) are carried through data channels (DCGs) and connected to the FDLs in order to delay the bursts if necessary. Upon passing through the FDLs, the DBs are connected to optical switching matrix. The BHCs are carried in control channels (CCGs) and they are carried into the switch control unit (SCU). In the SCU, by the help of routing and signalling processors, the optical switching matrix is configured in order to determine which of the DBs from an input wavelength channel is to be switched to which of the output wavelength channel. Upon being processed in the SCU, the BHCs are transmitted by the CCGs to the next hop and the DBs are transmitted through DCGs [38].

# 3. BURST ASSEMBLY TECHNIQUES IN OBS

At the edge router of the OBS domain, multiple IP packets are assembled into a burst. When assembling the bursts, it is aimed to form the bursts whose sizes do not vary much and lead to higher utilization of bandwidth [47]. Several techniques are proposed to form bursts from incoming IP packet traffic. The burst assembly techniques can be categorized as timer based, threshold based, timer and threshold based, adaptive, and QoS based burst assembly algorithms.

## 3.1 Timer Based Burst Assembly

In [20, 48], Timer Based Burst Asembly is proposed. The aim of timer based burst assembly is limiting the waiting time of the edge router to generate the bursts. In this technique, a minimum burst size, $\mu_{min}$, is determined in order to form the bursts a number times greater than their correponding BHC. Besides this, in order to limit the time to wait for assembling burst, a time threshold value *TTh* is also determined. For each destination and service class, a virtual queue is constructed. A timer is assigned and started for each virtual queue. The following steps are executed for each queue:

- *Step 0:* While the timer value is lower than the *TTh*
    - Accept new arriving packets
    - Check timer value
    - Go to *Step 0.*
- *Step 1:* Check the burst size
    - If the burst size is less than $\mu_{min}$ then
        - pad it to $\mu_{min}$
    - Generate the burst and send it to the corresponding output port.
    - Empty the queue
    - Reset the timer

- Go to *Step 0*.

Although this scheme limits the time to generate the bursts, it causes a variation in burst sizes which may lead to lower utilization at low loads [47]. However, this scheme may lead to too large burst sizes at some loads or it may lead to much more burst formation delay for short bursts.

## 3.2 Size Threshold Based Burst Assembly

In [48], as an alternative to time threshold based burst assembly, a size threshold based burst assembly is proposed. Here, a threshold value for the burst size, *STh* is determined. The aim here is avoiding the variation in burst sizes and having fixed size of bursts. Since the bursts are of fixed size, the utilization is supposed to be more regular than the scheme which leads to variance in burst sizes.

For each destination and QoS class, a virtual queue is constructed. The algorithm is so simple such that for each virtual queue the queue size is checked at each timeslot. If the queue size is equal to the *STh* then the IP packets in the queue are assembled in a burst and the ingress node starts waiting for new packets.

The problem with this scheme is that the burst assembler can wait for a very long time unless the queue size becomes equal to the *STh* by the arrival of the packets. Although this burst assembly technique avoids variances in burst sizes and leads to fixed size of bursts, it may not outperform the time threshold based technique in terms of burst formation time. In [24], another algorithm is used in order to overcome this problem (See section 3.3).

## 3.3 Hybrid Burst Assembly

In [24], a burst assembly technique is used which uses time-based and size threshold based schemes together. The ingress router goes on pushing IP packets into the corresponding virtual queue unless the size of the queue exceeds *STh*. However, the ingress router generates a burst if the timer exceeds *TTh* while the size of the queue does not exceed *STh*.

In our work we modify this technique by not allowing the bursts to exceed the size threshold and also the time threshold. Therefore, the bursts are strictly limited by the time and size constraints. Moreover, we do not let the bursts to be generated unless

the size of their corresponding queue do not reach a minimum value since the DB must have a minimum value in order to have a feasible greater size than its BHC [50]. The algorithm works as follows:

*Step 0:* At the beginning all virtual queues are empty and all timers are reset.

*For each virtual queue*

*Step 1:*While *timer* < *TTh*

Begin

- Wait for new arriving packet and put it into the virtual queue

- If the virtual queue size >= *STh*

Then

Begin

- Generate the burst

- Reset the timer for that virtual queue

- Generate the header

- Start *header-and-burst-transfer-based-on-JET*

- Go to *Step1*

Endthen

Else

- Go to *Step1*

Endwhile

/* means *timer* = *TTh* */

- Check virtual queue size

- If virtual queue size < minimum burst size

Then

- Pad it to minimum burst size

- Generate the burst

- Reset the timer for that virtual queue

- Generate the header

- Start *header-and-burst-transfer* based on JET

- Go to *Step1*

### 3.4 Adaptive Period Based Burst Assembly

This technique is also called Adaptive Assembly Period (AAP) algorithm [49]. In fact AAP, is a time threshold based burst assembly technique with dynamic time threshold. The time threshold for each queue at each ingress node is determined by the duration of the burst which has just been sent from that queue. Before starting to define AAP, it is useful to define the term used in the algorithm. $L_{sd}$ is the average burst length in bytes from $s$ to $d$. $TTh_{sd}$ is the assembly period for the virtual queue at the ingress node $s$ for the destination $s$. The average burst length is calculated by using the most recently generated bursts and the smoothed value of the average burst length where $SL$ is the most recently sampled average burst length, $\eta$ and $\epsilon$ are some weights such that,

$$L_{sd} = \varepsilon * L_{sd} + \eta * SL \quad \text{and} \quad \eta + \epsilon = 1. \tag{3.1}$$

In [49], $\eta$ is set to ¼, while $\epsilon$ is set to ¾. The reason of selecting the weight of the most recently sampled burst length greater is that a TCP connection is assumed to send more packets if a long burst is generated and sent.

Besides these, in determining the burst assembly period (time threshold) for an ingress node $s$ and one of the destinations $d$, the following constraint is used,

$$\frac{L_{sd}}{Bandwidth * Channel} \leq TTh \tag{3.2}$$

Based on the constraint in (3.2), by multiplying the left side of the inequality with a factor $\alpha$ ($\alpha \geq 1$), an equation can be derived for $TTh$ such that,

$$TTh = \alpha * \frac{L_{sd}}{Bandwidth * Channel} \tag{3.3}$$

The $\alpha$ coefficient is called the *assembly factor* [49]. If we draw $\alpha$ from (3.3), we obtain the formula in (3.4). Based on the formula, the value of $\alpha$ is almost equal to the number of virtual queues constructed by the ingress node *s*.

$$\alpha = \frac{Bandwidth * Channel}{\frac{L_{sd}}{TTh}} \tag{3.4}$$

## 3.5 Differentiated Burst Assembly

In [41], composite burst assembly techniques[22] are proposed in order to satisfy QoS requirements. In the differentiated burst assembly schemes, the burst that arrives to a node first is called the *original burst*, the burst which contends with the original burst and arrives at the node later is called the *contending burst*. These techniques are based on burst segmentation for QoS (see Section 4.5) such that, when a high priority contending burst contends with a low priority burst, the packets that form the tail of the original burst are dropped and the contending burst is scheduled (Figure 3.1.a). If a low priority burst contends with a high priority burst, the whole contending burst is dropped (Figure 3.1.b). In case of a contention of two bursts of equal priority, the tail of the original burst is segmented and dropped, the contending burst is scheduled (Figure 3.1.c).

Before defining the differentiated burst assembly techniques, we have to give the parameters used in order to make it easier to understand. The number of input packet classes is represented by *N*, and the number of burst priorities is represented by *M*. Based on this, *K* represents the number of burst types, and $N \leq K \leq 2^{(N-1)}$. The other parameters are as follows;

- $L_k^{\min}$ : Minimum burst length for the bursts of type *k*.

- $L_k^{\max}$ :Maximum burst length for the bursts of type *k*.

- $R_{kj}^{\min}$ :Minimum number of packets of class *j* in a burst of type *k*.

- $R_{kj}^{\max}$ : Maximum number of packets of class *j* in a burst of type *k*.

- $S_k = \left\{ j \mid R_{jk}^{\max} > 0 \right\}$:The set of packet classes that may be included in a burst of type *k*.

- $P_k$ :Priority of burst of type *k*.

- $TTh_k$ is the time threshold value for bursts of type *k*, and $STh_k$ is the size threshold value for bursts of type *k*.

The burst is generated only and only if the sum of the number of packets of class *j* exceeds the size threshold for that class such that $\sum_{j \in C_k} X_j \geq STh_k$ where $C_k \subset S_k$ [22].

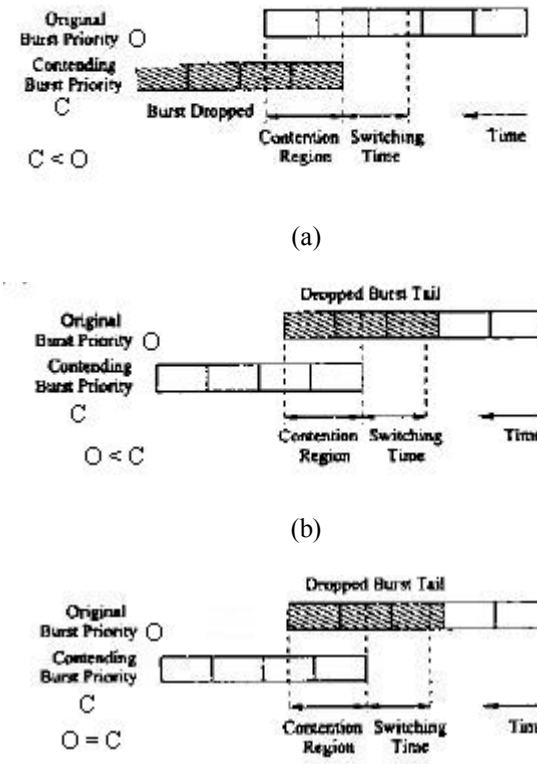



Figure 3.1. Contention of bursts
a. A low priority burst is contending with a high priority burst. b. A high priority burst is contending with a low priority burst. c. Two equal priority bursts contend [22].

The differentiated burst assembly techniques are categorized in 4 groups such as *Single Class Burst Assembly (SCB) with N = M*, *Composite Class Burst (CCB) with N = M*, *SCB with N > M*, and *CCB with N > M*. These techniques are explained in the following subsections.

### 3.5.1 Single Class Burst (SCB) with N = M

In this scheme, the number of classes is is equal to the number of priorities. The assembly process is illustrated in Figure 3.2. The number of burst types is set to be equal to M since $M \leq K \leq (2^N\text{-}1)$. Since the burst is supposed to be composed of just one class, for $0 \leq k \leq K$;

- $S_k$ is set to be $S_k = \{k\}$ so $C_k$ is also supposed to be equal to $S_k$.
- Each burst priority is assigned with the class' index such that $P_k = k$.
- $L_k^{\min} = L_k^{\max} = STh_k$
- $R_{kj}^{\max} = R_{kj}^{\min} = STh_k$ if $j = k$.
- $R_{kj}^{\max} = R_{kj}^{\min} = 0$ if $j \neq k$.

Each packet is collected on a seperate queue for its class. When the size of the queue exceeds $STh_k$ or the timer exceeds $TTh_k$ the burst is generated and sent into the OBS domain [22].

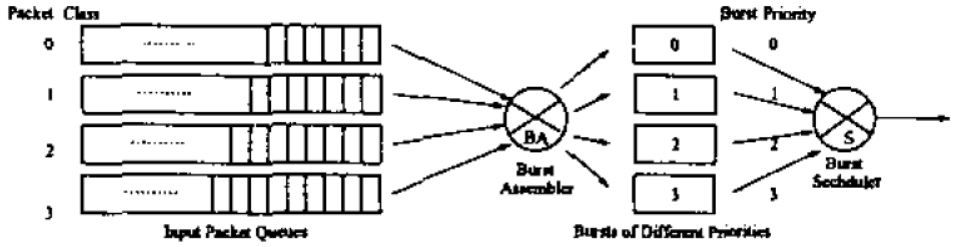



Figure 3.2. Illustration of a SCB with N = M = 4 [22].

### 3.5.2 Composite Class Burst (CCB) with N = M

Here, each burst may consist of packets of multiple classes. In Figure 3.3, illustration of assembling a composite class burst with $N = M = 4$ is shown. The number of burst types is set to be equal to M since $M \leq K \leq (2^N\text{-}1)$. Since the burst is supposed to be composed of different classes, for $0 \leq k \leq K$;

- $S_k$ is set to be $S_k = \{k, k+1\}$
- $C_k$ is also supposed to be equal to $\{k\}$.
- Each burst priority is assigned with the index of the class such that $P_k = k$.

- $L_k^{\min} = STh_k$ and $L_k^{\max} = \infty$
- $R_{kj}^{\min} = 0$ for $0 < j < N$.
- $R_{kj}^{\max} = \infty, j \in S_k$ and $R_{kj}^{\max} = 0, j \notin S_k$

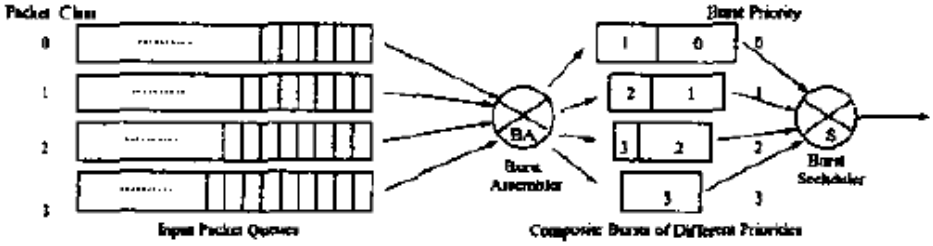



Figure 3.3. Illustration of a CCB with N = M = 4 [22].

Each packet is collected on a seperate queue for its class. When the size of the queue exceeds $STh_k$ or the timer exceeds $TTh_k$ the burst is generated and sent into the OBS domain. On the other hand, since a burst of type $k$ can consist of different packet classes, the packets can be of the calss $k$ or $(k+1)$ as defined in $S_k$. When a burst of type $k$ is generated, the packets of the class $(k + 1)$ form the head (not the header) of the burst while the packets of class $k$ form the tail of the burst [22].

### 3.5.3 Single Class Burst (SCB) with N > M

In this burst assembly scheme, a burst consists of just one class of packets. However, number of burst priorities is less than the number of the packet classes. This time, bursts that consist of packets of different classes may have equal priority as shown in Figure 3.4. This time, a bursts consisting of packets of class $k$ is generated when the size of its corresponding queue exceeds the $STh$ value for that queue. The number of burst types is set to be equal to the number of classes so $K$ is set to be equal to $N$. Since the burst is supposed to be composed of just one class, for $0 \leq k \leq K$ [22];

- $S_k$ is set to be $S_k = \{k\}$
- $C_k$ is also supposed to be equal to $\{k\}$.
- Burst priorities is organized as $P_k = \{k*M/N\}$.
- $L_k^{\min} = L_k^{\max} = STh_k$
- $R_{kj}^{\min} = R_{kj}^{\max} = STh_k$ if $j = k$.

- $R_{kj}^{\min} = R_{kj}^{\max} = STh_k$ if $j \neq k$.

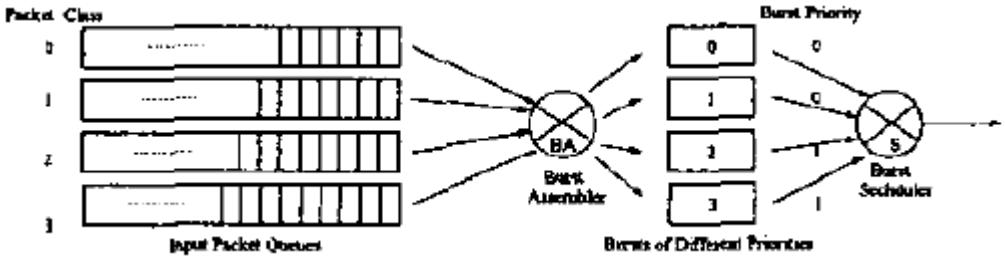



Figure 3.4. Illustration of a SCB with N = 4 and M = 2 [22].

### 3.5.4 Composite Class Burst (CCB) with N > M

In CCB assembly technique, a burst may consist of packets of different classes. Here, apart from SCB with $N > M$, the bursts that consist of different class packets have different priorities as seen from Figure 3.5. The number of burst types is set to be equal to the number of priorities so $K$ is set to be equal to $M$. Since the burst is supposed to be composed of packets of different classes, for $0 \leq k \leq K$ [22];

- $S_k$ is set to be $S_k = \left\{ \frac{k*N}{M}, \ldots, \frac{(k+1)*N}{M} - 1 \right\}$
- $C_k$ is also supposed to be equal to $S_k$
- Burst priorities, $P_k$ is set to be equal to $k$ where $k$ is the burst type.
- $L_k^{\min} = L_k^{\max} = STh_k$
- $R_{kj}^{\min} = 0 \quad 0 \leq \mathrm{j} < \mathrm{N}.$
- $R_{kj}^{\max} = STh_k, j \in S_k.$
- $R_{kj}^{\max} = 0, j \notin S_k$

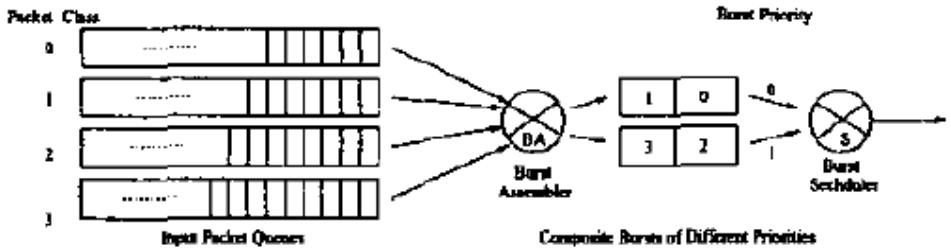


Figure 3.5. Illustration of a CCB with N = 4 and M = 2 [22].

A burst from type $k$ is generated when the sum of the packets in that queue exceeds $STh$ or the timer exceeds $TTh$ for that queue. Here, similar to CCB with $N = M$, when a burst of type $k$ is generated, then all the packets in the corresponding queue is assembled in that burst such that the packets of class $(k + 1)$ form the head part while the packets of the class $k$ form the tail of the burst [22].

## 3.6 Round-Robin Burst Assembly

In [51], another burst assembly scheme is proposed which is called round-robin burst assembly. In this scheme, the IP packets are buffered in the virtual queues depending on their destination (Figure 3.6). For each queue, the $TTh$ value is constant, and a cycle time of round-robin is defined for the total processing time at all queues. The IP packets are collected during the cycle time. The aim here is transmitting the bursts into the OBS domain at fixed time intervals. The algorithm skips the queue and processes the next queue unless there exists at least one IP packet in the queue during the cycle time.

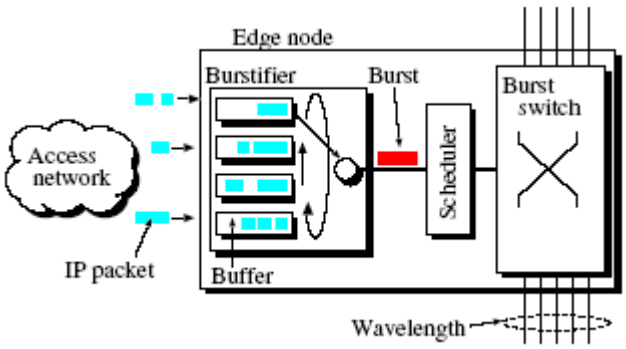



Figure 3.6. Illustration of round robin burst assembly scheme [51].

The assumption in this scheme is that the IP packets arrive the ingress node by following a Poisson distribution with arrival rate $\lambda$. If the number of virtual queues (destinations) is $L$, the arrival rate of the packets to each queue is $\lambda / L$. If the

bandwidth of a wavelength is $B$, and the size of a packet is $M$ bits, then the duration of a packet is $\mu = M / B$.

If there is at least one IP packet in the burstifier unit, the bursts are transmitted into the scheduler unit. Based on this, the behaviour of the scheduler is modeled as an ON/OFF source. The fixed intervals in which the bursts are being transmitted are the ON states, while the intervals in which no burst is transimtted are the OFF states. The departure period of the bursts from the scheduler (the processing time for burst assembly at a queue) is $T$ so the cycle time for buffering the bursts is equal to $LT$. Therefore the average transmission time of a burst is $\lambda T\mu = \lambda T M / B$ so the roun-robin burst assembly scheme can be modeled as a $M/M/W/W$ queueing model [51] where there are $W$ wavelengths at the output of the scheduler.

# 4. CHANNEL SCHEDULING AND CONTENTION RESOLUTION TECHNIQUES IN OBS

"Channel Scheduling and Contention Resolution Techniques" is one of the main concerns of this thesis. Since the BHCs of the bursts do not arrive one after another, more than one burst may attempt to reserve the same time interval on the same channel concurrently. This is a common problem in OBS Networks. In Figure 4.1, a simple contention scenario is illustrated. The dashed object represents the incoming burst to be scheduled and the others are the DBs that have been scheduled. As it is seen from the figure, there's no available interval which fits with the arriving DB.

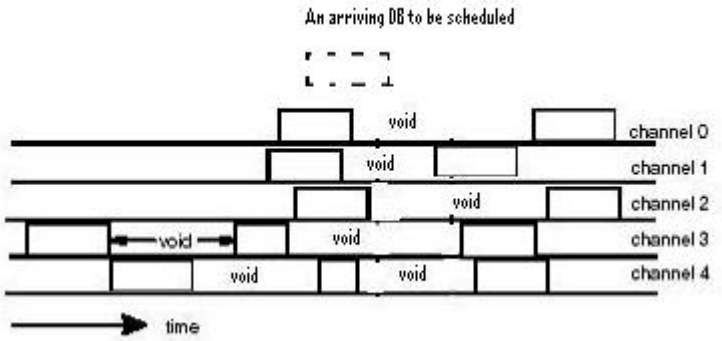



Figure 4.1. A simple contention scenario

We analyze the channel scheduling techniques in OBS in 4 groups: Non-void filling, void-filling, groupping based and segmentation based techniques. In OBS terminology, the time intervals that are unscheduled (idle) are called voids. In Figure 4.2, an optical link consisting of 4 wavelength channels is illustrated and the time intervals which are not scheduled by the bursts can be seen. Based on this, we can define what is meant by the names of the techniques.

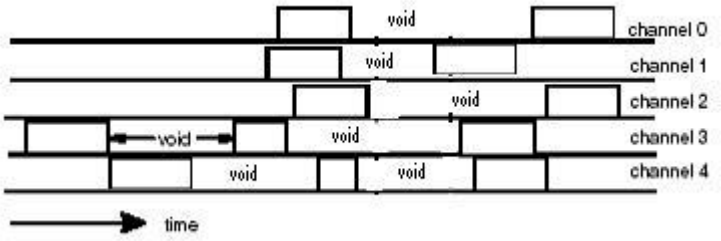



Figure 4.2. Voids on wavelength channels

- *Non-void filling techniques*: Non-void filling techniques do not attempt to utilize the idle intervals on the wavelength channels. For a non-void filling technique, a channel is said to be feasible if the last scheduled time interval is less than the arrival time of the DB.

- *Void-filling techniques:* The aim of void-filling techniques is to utilize the voids in the wavelength channels shown in Figure 4.2. If the arrival time of a DB is *a* and its length is *l,* then an interval I(Is, Ie) is feasible for an arriving burst b(*a, l*) if Is < *a* and *Ie* > *l*. The voids which satisfy these two inequalities are atempted to be filled by the void-filling techniques.

- *Group Scheduling technique(s):* Void-filling and non-void filling techniques try to reserve available time interval on a wavelength channel for an arrival burst immediately upon the arrival of a BHC. Group Scheduling is based on collecting the BHCs for a period of time instead of scheduling them immediately and attempting to minimize the number of contending DBs.

- *Segmentation based techniques:* Segmentation is scheduling the non-contending segment of a DB and dropping the contending segment of it. Segmentation techniques are based on reducing the loss rate by attempting to schedule the non-contending segment of a contending burst.

## 4.1 Non-Void Filling Scheduling Techniques

In [4], switching the incoming data bursts to unscheduled wavelength channels is proposed as a practical and fast approach for OBS. Since the switching operation here deals with only unscheduled channels, the idle intervals are not considered as a resource candidate so these techniques are called Non-Void Filling Scheduling Techniques.

### 4.1.1 Non-Void Filling Scheduling Techniques- The Horizon Algorithm

In the horizon scheduling algorithm, the scheduler keeps track of a horizon for each channel which refers to the time that no reservation is done on that channel after that time [38].

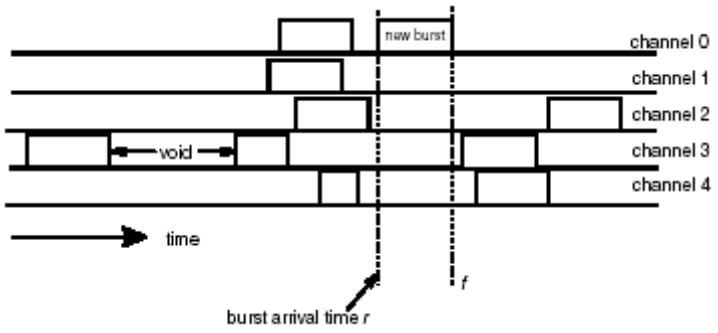



Figure 4.3. Illustration of the horizon algorithm [13].

In Figure 4.3, the horizon algorithm is illustrated. The arriving burst arrives at the router at time *r*, and the end time of the burst is given as *f*. The scheduler searches for the channel, with the latest horizon value. Upon finding the latest horizon value, it checks out the horizon value with the arrival time of the burst (*r)*. In the example above, channel 0 is found as the appropriate channel and the burst is scheduled on channel 0 for time *r*. Upon scheduling the burst on the channel 0, the new horizon value for the channel becomes the end time of the burst (*f)* [13].

In a network system of k wavelength channels, the total number of search operations do not exceed O(log k) which points out the speed of the horizon algorithm. In Figure 4.3, there are unscheduled void intervals behind the horizon and/or in the other channels. These unscheduled void intervals decrease the channel bandwidth utilization and as a result, packet losses may occur. Although the horizon algorithm has an advantage in hardware implementation of this model, it brings out the disadvantage of increasing the number of voids between the scheduled bursts [11]. In the following sections a comparative study of the horizon technique with the other techniques are given.

## 4.2 Void Filling Scheduling Techniques

In [13], 4 void filling algorithms are proposed in order to avoid contention in OBS. Besides this, there are some other techniques that use fiber delay lines (FDL) in order to utilize voids which will also be covered in this section. Initially, in order to understand the problem in void-filling well, it has to be implemented geometrically.

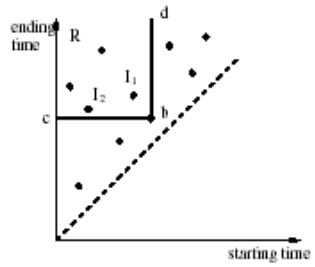



Figure 4.4. Geometrical modeling of the problem [13].

In Figure 4.4, all the points are located above the line y = x. The burst *b* is also somewhere above the line. In the figure the burst is represented by *(c, d)* as the arrival and the departure times. The points whose x-coordinates are less than *c* and y-coordinates are greater than *d* form the feasible region for the burst *b(c, d)* which is drawn by thick line.

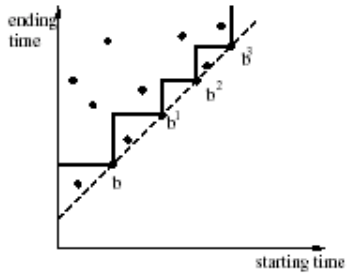



Figure 4.5. Geometrical modeling of the problem when FDLs deployed [13].

If FDLs are deployed, the arrival time of the burst will be delayed as long as the length of the delay lines. In Figure 4.5, *B* different delay lines are considered, and the arrival and the departure times of the burst *b* are collected in the sets such that, starting times = {c, c, $+ d_1, c + d_2, \ldots, c + d_B$} and the end times = {d, d $+ d_1, d + d_2, \ldots, d + d_B$} where $d_i$'s are the delay values provided by the FDLs. In the Figure, each step on the boundary of the feasible region represents a delayed arrival and departure.

Now, the problem of channel scheduling is reduced to selecting a feasible point from the feasible region in or der to utilize the channel in the most efficient way.

### 4.2.1 Latest Available Unused Channel with Void Filling (LAUC-VF)

LAUC-VF keeps track of all unused intervals (voids), and performs a binary search in order to find the feasible intervals whose starting times are less than the burst's arrival and whose end time is greater than the burst's departure time [38]. Here it has to be noticed that, even if one of the resources is scheduled, it may still be a feasible void for the burst. In Figure 4.6, LAUC-VF algorithm is illustrated for 4 wavelength channels and N voids. The resource allocation of the channels are the same as in Figure 4.3.

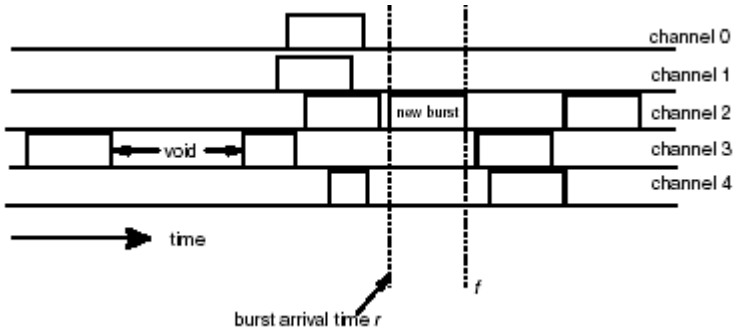



Figure 4.6. Illustration of the LAUC-VF algorithm [13].

Upon arrival of a burst b(*r, f)*, the scheduler in the OBS router performs a binary search in the list which it keeps track of all voids. When it obtains a set of feasible voids, then it attempts to select the void with the latest starting time in order to schedule the burst [38].

The number of voids (*N*) is mostly greater than the number of channels (*k*) such that $k << N$ so the time complexity of LAUC-VF is expected to be O(log *k*). As it can be seen, the time complexity of LAUC-VF is much greater than the complexity of the horizon technique. However, since the technique improves the channel utilization by utilizing the unscheduled intervals, burst loss rate tends to decrease.

### 4.2.2 Minimum Start Void Fit (Min-SV Fit) and Minimum End Void Fit (Min-EV Fit)

The LAUC-VF algorithm can be modified as minimizing the void that is being generated just before the burst's arrival which is called Minimum Start-Void Fit. LAUC-VF can also be modified in order to minimize the void being generated just after the burst's departure which is called Minimum End-Void Fit (Figure 4.7) [13].

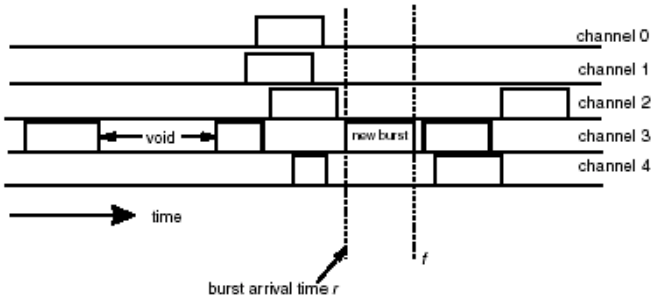



Figure 4.7. Illustration of the Min-EV Fit algorithm [13].

Geometrically, Min-SV Fit stands for the point in the feasible region, that is closest to the line x = *r* (Figure 4.4) where *r* is the burst arrival time. In [13], the data structure to realize Min-SV Fit algorithm is a balanced binary search tree, $T_{start}$ whose leaves are the elements of *I* where *I* is the set of all voids. The structure of $T_{start}$ is shown in Figure 4.8.

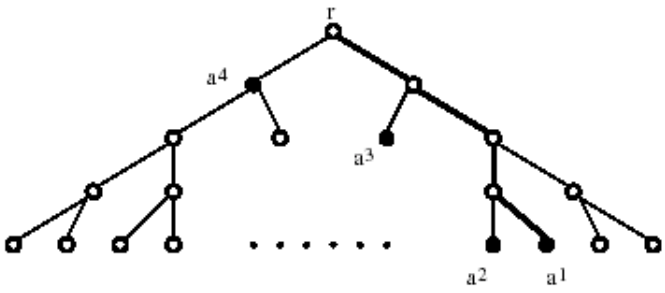



Figure 4.8. Balanced binary search tree structure used in Min-SV, Min-EV and Best Fit [13].

The algorithm works as follows[13]:

Construction of the tree:

- The intervals that are the elements of *I* are ordered with respect to their starting times in ascending order.
- The median starting time in *I* is assigned as the root of the $T_{start.}$

- The tree is constructed recursively such that for each node (in the tree) which is the root of a subtree, all the starting times of the nodes in the left subtree are less than the root's starting time while the starting times of the voids in the right subtree are greater than the starting time of the root.

Search Procedure:

- The allocation nodes are determined such that from the root to the leaves, if the starting time of a node is less than the incoming burst's arrival time then
    - Proceed to the left
- Else
    - Left child is marked and proceeded to the right.
- At the end, the marked nodes are the allocation nodes with starting times less than the arrival time of the DB.
- A sequence is obtained based on the starting times of the allocation nodes such that $A=\{a^1, a^2, a^3,...,a^h\}$.
- Starting from $a^1$, an allocation node is searched in A that is in the feasible region of the DB.

If the number of elements of the set *I* is *m,* there can be at most *m* nodes in A. Checking the feasibility of $a^i$ takes O(1) time so the total execution time for the algorithm is expected to be O(log m).

In [13], a similar method is proposed for Min-EV Fit. The difference is that the binary search in Min-EV Fit is performed top-down so the voids in *I* are ordered in descending order according to their end times, and the search tree is constructed from *I*. The allocation nodes are ordered according to their heights in descending order.

In Figure 4.9, a comparison of performances of the horizon, LAUC-VF and Min-SV fit techniques is given in terms of scheduling time. Two results can be drawn from the figure. The first one is that the change in offered load does not affect the scheduling time of the horizon and Min-SV but the time complexity of LAUC-VF increases. The horizon and Min-SV have almost the same scheduling time since Min-SV keeps a more complex and greater data structure than the horizon while the

horizon just keeps track of just the last scheduled timeslots. This causes the scheduling time of a void-filling algorithm not to be affected by the increase of the offered load as the non-void filling horizon algorithm.

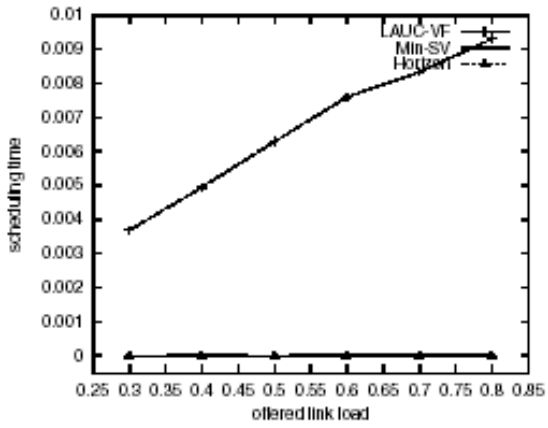



Figure 4.9. Performance comparison of the horizon, Min-SV and LAUC-VF in terms of scheduling time [13].

As we have just mentioned, the data structure kept by LAUC-VF is track of all void intervals, and a classical binary search is obtained while Min-SV keeps a red-black tree like balanced binary tree and performs a binary search on that tree. Therefore LAUC-VF leads to larger time in comparison to Min-SV and the horizon. Besides these, the greater size of the data structure kept in Min-SV also causes another result that the scheduling time of a void filling algorithm to be almost equal to the horizon.

In Figure 4.10, a comparative study of the horizon, LAUC-VF and Min-SV Fit is given in terms of loss rate related to the offset time. As it is seen from the figure, the loss rate of LAUC-VF and Min-SV Fit almost remains the same as the offset time increases while the horizon leads to higher burst loss rate as the offset time increases. Xu et al. explain the reason of this by analyzing the channel utilization in the horizon. Horizon keeps track of just the latest scheduled time interval so if data transmission takes long time, the unused bandwidth increases; as expected, wasted bandwidth results with high cell loss rate.

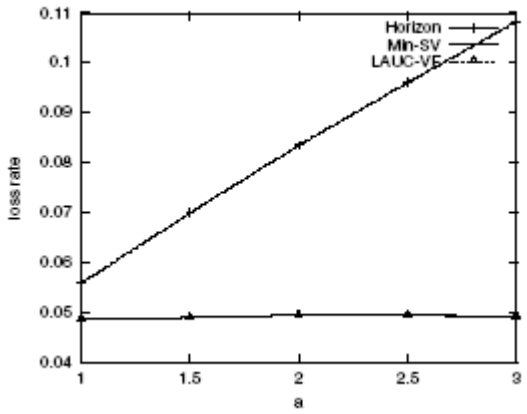



Figure 4.10. Performance comparison of the horizon, Min-SV and LAUC-VF in terms of loss rate [13].

Xu et al. also compare Min-SV and Min-EV, first in terms of loss rate with resect to offered load, and then in terms of scheduling time with respect to offered load. As it can be observed from Figure 4.11, Min-SV has a slightly better performance in terms of loss rate as the offered link load increases. The reason of this can be explained such that Min-SV attempts to reserve the left side of the void while Min-EV attempts to schedule the the right side of the void. As it is known, the time expiration goes from left to the right so Min-EV Fit is a little bit late to catch the left side expiration and schedule the bursts so this brings out higher loss rate.

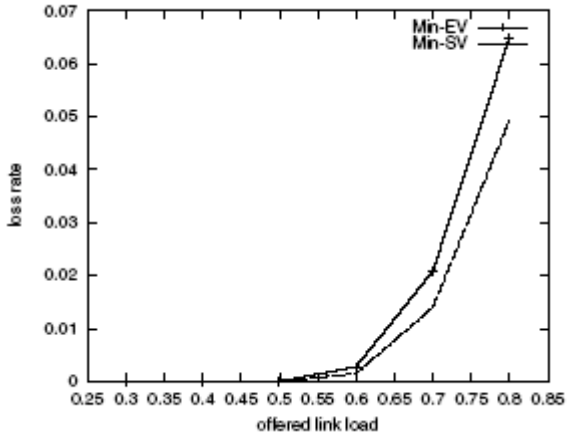



Figure 4.11. Performance comparison of the horizon, Min-SV and Min-EV in terms of loss rate [13].

Another performance comparison is done by Xu et al. by using the scheduling time latencies of Min-SV and Min-EV Fit as shown in Figure 4.12. Min-EV Fit has a significantly less scheduling latency when compared with Min-SV Fit. Here, we have to remember the data structures in both techniques. Min-SV Fit keeps the balanced binary search tree, $T_{start}$ in which the nodes are placed in increasing order according

to the start times of the voids. Min-EV Fit keeps a very similar balanced binary search tree structure $T_{end}$ but the main difference is that in $T_{end}$, the end times of the voids are placed in the tree in decreasing order. When a BHC for an incoming burst arrives; the search process is performed according to its end time, and there are a significantly a few number of voids whose end time is greater than the incoming burst's departure time. Therefore, the search process is performed in the most bottom subtree of $T_{end}$, and this decreases the time for searching a feasible interval and scheduling the bust in comparison to Min-SV Fit.

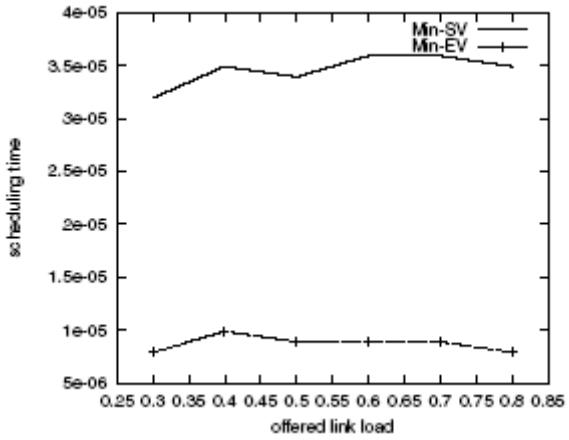



Figure 4.12. Performance comparison of the horizon, Min-SV and Min-EV in terms of scheduling time [13].

### 4.2.3 Best Fit

Another approach here, is minimizing the two voids which are generated just before the arrival time of and just after the departure time of the burst which is called the Best Fit. The geometric representation of the model can also be alnalyzed as in Figure 4.13.

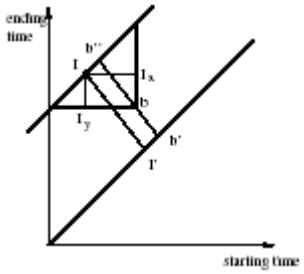



Figure 4.13. Geometric representation of Best-Fit [13].

The burst is represented by the point where x = *r* and y = *f*, such that *b(r, f)*. Let us assume that, *I* is a candidate feasible void for *b*, and *I* is represented as *I(s, e)*. The following sum has to be minimized:

$$L = (r - s) + (f - e) \tag{4.1}$$

In the figure, *L* is equal to the sum of the following segments; $\|(b \rightarrow I_y)\| + \|b \rightarrow I_x\|$ where $I_y$ is the vertical projection of *I* on the line y = *f* and $I_x$ is the horizontal projection of *I* on the line x = *r*. In fact, *L* is also equal to $\sqrt{2}( \|I \rightarrow I'\| - \| b \rightarrow b'\| )$ where *I'* and *b*' are the orthogonal projections of *I* and *b* on the line y = x respectively. Here $b \rightarrow b'$ is fixed so the value of $\|I \rightarrow I'\|$ has to be minimized.

In order to realize a best-fit scheduling for an incoming data burst, a balanced binary search tree, $T_{end}$ is constructed based on the end times of the intervals. Also, for each internal node of $T_{end}$, another tree is constructed by using the nodes that make up the subtree $T_{end}$ whose root is that node (say *v*) based on the starting times of the nodes, such that $T_{start}^{v}$.

In each node of $T_{end}$ the following parameters are kept:

- $I_{ym}^{v}$ : The median interval based on the end time in the subtree $S^v$ where *v* is the root of the subtree $S^v$.
- $I_{xmin}^{v}$ : The minimum starting time of the intervals in subtree $S^v$ whose root is *v*.

In each node *u* of $T_{start}^{v}$, the following parameters are kept:

- $I_{xm}^{u}$ : The median interval based on the starting time in the subtree of $T_{start}^{v}$ where *u* is the root of the subtree.
- $p_{pmin}^{u}$: A pointer to the minimum projection distance in the subtree of $T_{start}^{v}$ where *u* is the root of the subtree.

Based on these parameters, the following steps are employede in order to implement the best-fit technique:

- Similar to Min-SV fit, construct all of the allocation nodes of the burst *b* in the sequence $A_{end}$.
- Search the best-fit for each allocation node "*a*" such that

- $S^a$ is a horizontal strip in the coordinate plane where $y \geq f$. Compute the allocation nodes $A_{start}{}^a$ in $T_{start}{}^a$.
- For each allocation node u of $A_{start}{}^a$, search for the best-fit by using the parameter $p_{pmin}{}^u$ to the minimum projection distance in the subtree of $T_{start}{}^a$ rooted at $u$.

Note that, the time complexity to search for the allocation nodes in $A_{end}$ takes maximum O(log $m$) time. However, it also takes O(log $m$) time to search for the allocation nodes in $A_{start}{}^a$ so the total time complexity for the best-fit is $O(\log^2 m)$ [13].

### 4.2.4 Void Filling Scheduling Algorithms used with FDLs

In [14], a scheduling algorithm is proposed in order to minimize the voids that are generated by arriving bursts (Figure 4.14). The algorithm aims to minimize the void between the end time of the incoming burst and the start time of the other burst which is scheduled following the incoming burst that is so close to Min-EV Fit. The data burst is delayed $D$ units each time no eligible channel is found up to time ($t + B * D$) where $B$ is the number of the fiber delay lines and $D$ is the unit delay of one FDL. Here, the bandwidth efficiency is improved while end-to-end delay is increased but the time complexity of the algorithm rises up to $O(B * log\ m)$.

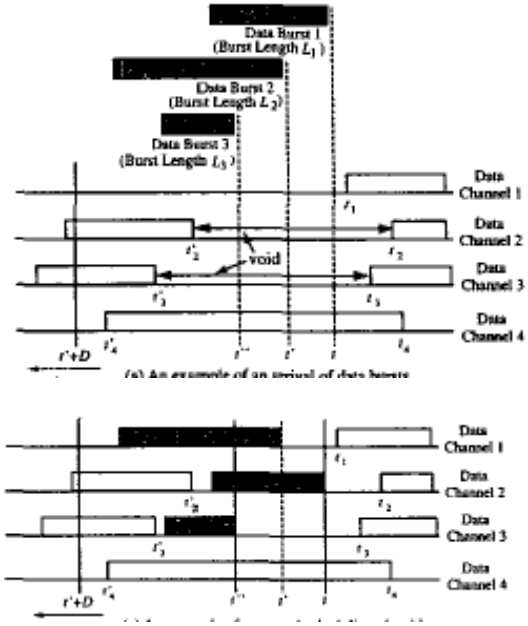



Figure 4.14. An algorithm proposed to minimize voids generated by the arriving bursts [14].

In [13], an alternative aproach to scheduling with FDLs is proposed which is called *Batching FDL Algorithm*. Batching FDL Algorithm first performs either Min-SV or Min-EV Fit for the burst. If no eligible interval is found then the folowing steps work and a first-fit aproach is performed by the following steps:

- A set of delays is constructed such that D = $\{d_1, d_2, ..., d_p\}$ where $1 < p \leq B$ where $B$ is the number of delay lines. The feasible region $R_D$ is enlarged to $R_D$' as illustrated in Figure 4.15. A binary search tree $T_{end}$ is constructed based on the end times of the intervals in the united region $R_D$'. The allocation nodes of $T_{end}$ are found and the united region is decomposed into subregions. Each subregion stands for an allocation node. In Figure 4.15, *P1*, *P2*, *P3* and *P4* are the decomposed subregions of $R_D$'.

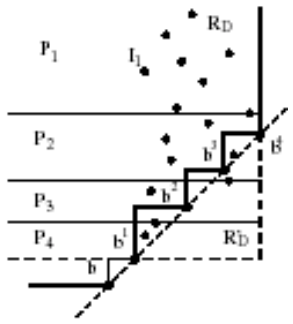



Figure 4.15. Geometrical view of Batching FDL scheduling [13].

- An interval, $I_j$ with the smallest starting time is found beginning from the top most subregion, $P_j$ (*P1* here). $I_j$ is checked whether it is inside $R_D$. Two end times $f^q$ and $f^{q+1}$ that bound $I_j$ are selected such that $1 \leq q < p$. If starting time of $I_j$ is greater than $r_q$, then $I_j$ is out of $R_D$, otherwise it is inside $R_D$.

- If $I_j$ is outside $R_D$, the region $P_j$ is decomposed into subsubregions by using the key $f^{q'+1}$ such that $f^{q'+1} \geq f^{q+1}$. All the steps are repeated in the subsubregion. If an eligible interval is found, the algorithm stops, otherwise the algorithm goes on with the subregion $P_{j-1}$.

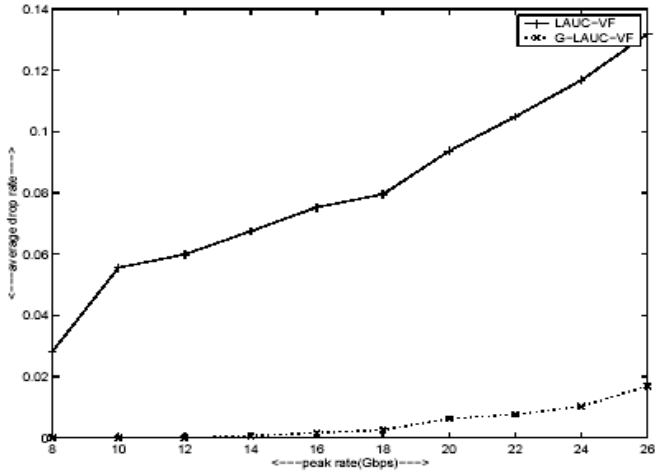
The maximum number of decompositions is $p$ so the maximum time that the algorithm stops is expected to be less than or equal to O*(p * log m)*.

### 4.2.5 Generalized LAUC-VF (G-LAUC-VF) to support QoS

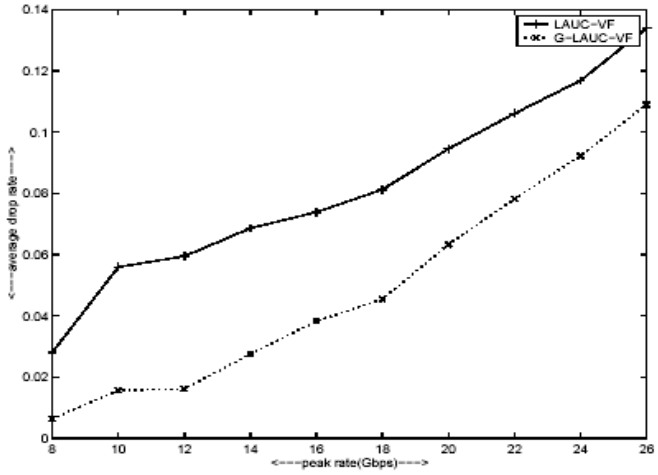
G-LAUC-VF is proposed to supply QoS requirements in OBS networks [43]. As known, in IP networks Integrated Services (IntServ) and Differentiated Services (DiffServ) models are used in order to satisfy QoS requirements. In IntServ models, for a packet from source to destination all the resources along the path are reserved by RSVP on each hop. Moreover, on each intermediate router, the information is kept per-flow basis and this situation leads to limit the scalability in IP networks. In order to come over this phenomenon, DiffServ is proposed as an alternative and by means of DiffServ the IP packets of the same QoS class are aggregated together so the number of flows from source to destination are decreased in order to maintain scalability [44].

G-LAUC-VF is a technique which aims to decrease packet loss rate by means of DiffServ. The technique works as follows: The IP packets arriving an ingress router are aggregated in a burst based on their QoS class and destination. There are $n$ different service QoS classes.At each optical link, the scheduler constructs a queue for each of the $n$ service class such that $\{ Q_1, Q_2,\ldots,Q_n\}$. $Q_i$, includes the BHCs of the $i^{th}$ service class. The following steps are performed in each timeslot[44]:

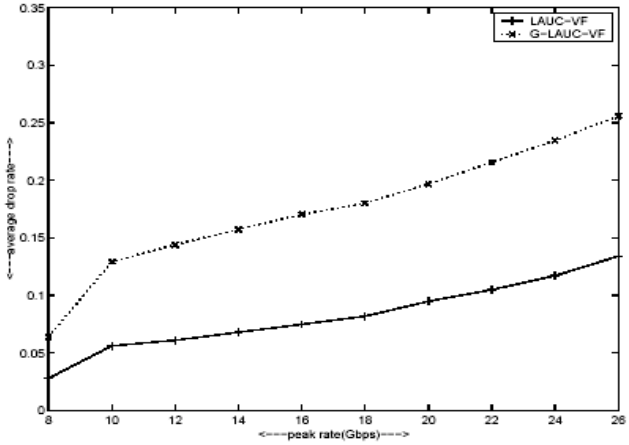
```
Begin
    For i = 1 to n
    Begin
        While (A BHC is found in the queue Q_i that belongs to this
        timeslot.)
        Begin
            Run LAUC-VF algorithm and schedule the DB of the
            i^th service class.
        EndWhile
    EndFor
End
```




(a) Burst loss rate for service class 1.




(b) Burst loss rate for service class 2.




(c) Burst loss rate for service class 3

Figure 4.16. Burst loss rates in G-LAUC-VF in comparison to LAUC-VF for 3 service classes [44].

As explained in 3.2.1, LAUC-VF algorithm keeps the unreserved timeslots and the last reserved timeslots on each wavelength channel in two distinct sets *Sg* and *Sm* respectively. The search process is initially performed on the set *Sg*. If a feasible interval cannot be found on *Sg*, then the same search process is performed on *Sm*. If no feasible interval is found by these processes and if FDLs are available, the burst is delayed for the duration offered by the delay lines and the search processes are repeated. If no feasible interval is found, the burst is dropped [38].

In Figure 4.16, the average burst loss rates under varying load are shown for the three QoS classes.

### 4.2.6 Merit-based Scheduling

Merit-based scheduling is a technique to reduce the loss rate of the bursts with longer routes by giving them a priority over the bursts with shorter routes [45]. In order to achieve this, a burst is represented by *B* and assigned a metric *V(B)*.

*V(B)* is a function of the following characteristics:

- *L(B):* Length of the route of *B*.
- *H(B):* Current hop count of *B*. This also gives the distance between *B* and its destination.
- *R(B):* Residual offset time of *B*.
- *T(B):* Initial offset time of *B*.
- *D(B):* Duration of *B* in time.
- *V(B)* = f{ *L(B), H(B), R(B), T(B), D(B)* }

Based on these parameters, *V(B)* is higher if the burst *B* is close to its destination or spent a lot of time to be processed. There are a number of ways to define *V(B)*. When a new burst arrives, the scheduler runs the LAUC-VF algorithm in order to reserve available bandwidth. If LAUC-VF fails to reserve a bandwidth for the burst $B_n$, merit based scheduling algorithm is employed for *k* wavelengths. For each wavelength *i*, the contending bursts are collected in a set, $S_i$. For each wavelength *i*, a metric called the "figure of metric" ($f_i$) is computed and is defined as:

$$f_i = \max_{B \varepsilon S_i} \{V(B)\} \tag{4.2}$$

As a result, the burst with the highest *V(B)* value represents the group of the bursts. For a burst, $B_n$, that cannot be scheduled by LAUC-VF, the $V(B_n)$ value is computed and is compared with the elements of the set $\{f_i\}$. If $V(B_n)$ is greater than an element of the set ($V(B_n) > f_j$ ), then $V(B_n)$ becomes the new merit value for group *j*. $B_n$ is scheduled on the channel *j*, and the other bursts in $S_j$ are discarded [45].

In [45], 5 different ways are proposed to compute the *V(B)* value such that:

- *H*L*: The bursts that have traveled along longer routes have higher priority (Metric 1).
- $\frac{H * L}{(L - H)}$: The bursts that have traveled along longer routes and closer to their destination have higher priority (Metric 2).
- *(H + L)*: Means the same as (H*L) (Metric 3).
- $\frac{H + L}{(L - H)}$: Means the same as $\frac{H * L}{(L - H)}$ (Metric 4).
- *T / R*: The bursts, the rate of whose residual offset times to their initial offset times are greaer have higher priority (Metric 5).

In Figure 4.17, a comparison on the performance of the 5 metrics and classical LAUC-VF is given. It is obvious that all the metrics perform better than standard LAUC-VF. As it is seen from the figure, Metric 1 leads to the lowest burst loss rate since the bursts who have travelled longer also decrease their residual offset time so by applying such a policy those bursts survive. The other metrics have closer performance but all of them perform better than LAUC-VF.

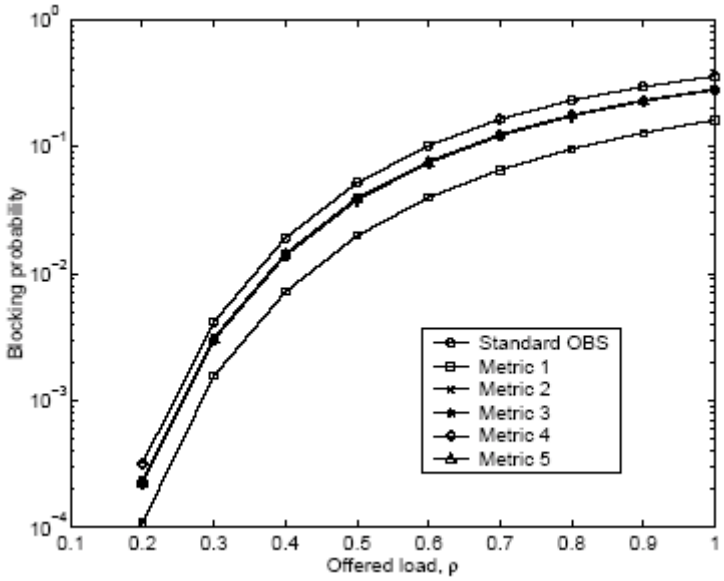



Figure 4.17. Burst loss rates in merit based scheduling for different metrics and in LAUC-VF (Standard OBS) [45].

## 4.3 Group Scheduling Technique

Although we group the OBS scheduling techniques in 4 main categories, there can be another approach to categorize the currently proposed techniques, as immediate scheduling and group scheduling. Non-Void Filling, Void-Filling, and segmentation based techniques try to reserve an available time interval for an arriving burst upon the arrival of its BHC. However, that kind of an immediate reservation scheme may cause decrease in utilization, and as a result, an increase in the loss rate [16].

An alternative approach is to consider the utilization of the wavelength channels in a longer timescale. In such a model, it is required to delay the BHCs for a pre-determined time period, and come up with a nearly optimum reservation scheme by dealing with the BHCs in that group instead of scheduling them individually as they arrive. This is the main philosophy of Group Scheduling [16].

In Group Scheduling, a wavelength channel is represented as a time line and that time line is partitioned into small time windows so reservation for the data bursts is performed window-based. A window is represented by 2 parameters: $W_{start}$, $W_{end}$ that are the start time of the window and the end time of the window respectively. The collection period for each window is constant and the end of the collection period for window $i$ is represented as *Tclose(i)*. In order to achieve the job, the system is

seperated into three modules; BHC Grouper Module, Classification and Channel Assignment Module and Channel Scheduler Module. In Figure 4.18, the system model is illustrated.

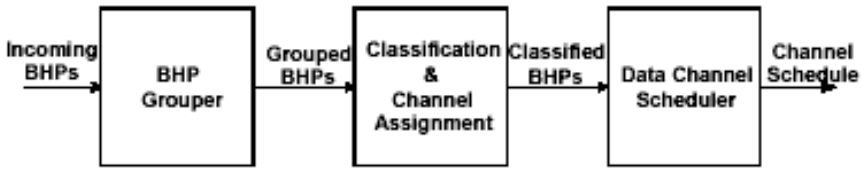



Figure 4.18. System model for Group Scheduling [16].

### 4.3.1 BHP Grouper

Each window has a closing time which points the latest arrival time of a BHC in order to reserve that window for its DB. When a BHC arrives, the BHC grouper module looks up the BHC's DB's arrival time, determines the window to which the DB belongs and puts the BHC into the corresponding basket. The DBs requesting the same window are collected in a basket so there are $N$ baskets where the channel consists of $N$ time windows.

Basket 1 collects the BHCs whose DBs request the time window of [$W_{start}$, $W_{end}$] until the cloesing time of the window, $T_{close}(1)$. Basket 2 collects the BHCs whose DBs request the time window of [$W_{start}$ + One window period, $W_{end}$ + One window period] until the closing time of the window, $T_{close}(2)$. BHC Grouper module is illustrated in Figure 4.19.

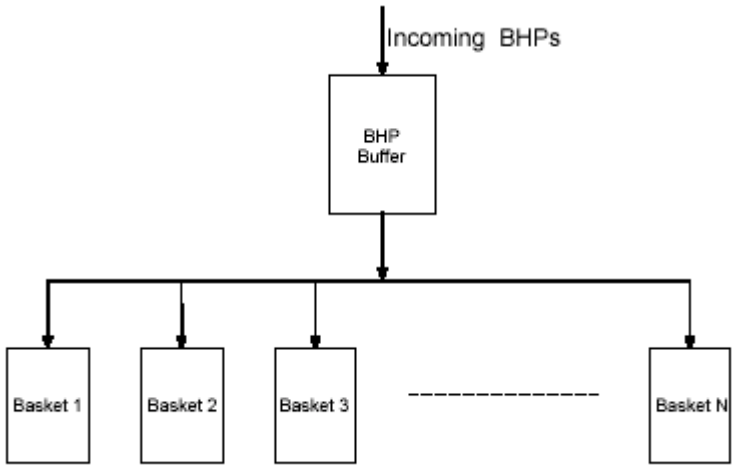



Figure 4.19. BHP Grouper [16].

### 4.3.2 Classification and Channel Assignment

Upon groupping, the BHCs have to be classified into different service classes and pushed into corresponding queues. This module, first classifies and pushes the groupped BHCs into the corresponding service queues. In our work, we assume the number of service classes as 1. After queueing, the BHCs in the service class queues are placed into corresponding wavelength channel schedulers. The process in this module is illustrated in Figure 4.20.

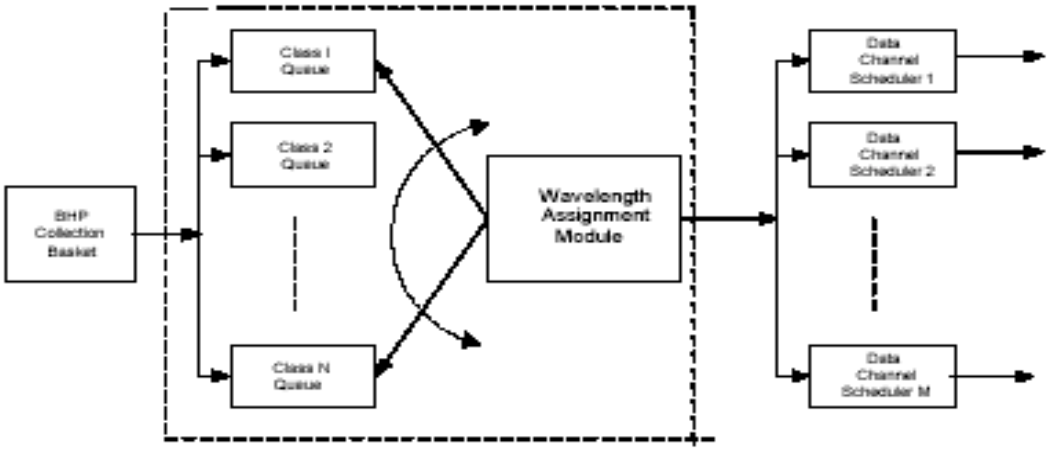



Figure 4.20. Classification and Channel Assignment [16].

In assigning the incoming bursts to wavelength channels, different policies may be employed. However, there is not a clear reference for this issue in the literature. In our work, we use a round-robin scheme in order to perform this job. Consider that there are $N$ wavelength channels and assume that the incoming bursts are numbered based on the arrival order of the BHCs so $i^{th}$ BHC is assigned to $(i \bmod N)^{th}$ channel.

### 4.3.3 Channel Scheduler

This module performs the scheduling process of the BHCs according to their corresponding DBs' arrival times and lengths. To achieve this goal, the scheduler, first, constructs an interval representation graph. In order to understand the construction of the interval representation graph, think about the burst arrival pattern in Figure 4.21. As it can be seen from Figure 4.21, there are conflicts between DBs of the BHCs in that window. In order to schedule maximum number of bursts in this window, each DB is assumed as a node of the interval graph. The nodes whose corresponding DBs conflict are connected by an edge in the graph so the following interval graph in Figure 4.22 is obtained.

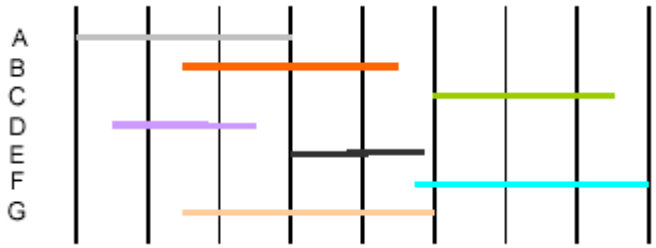



Figure 4.21. Channel Scheduler interval profile [16].

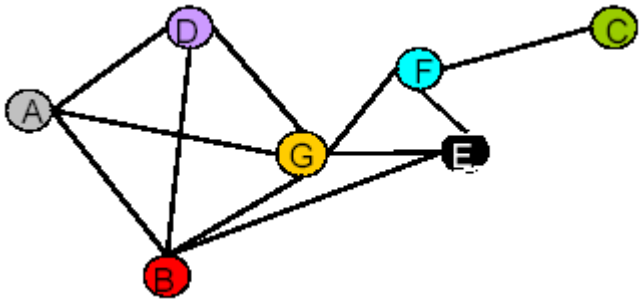



Figure 4.22. Interval graph derived from Figure 4.21 [16].

Here, the problem becomes equivalent to the maximum independent set problem [39]. The adjacent nodes in the graph in Figure 4.22 represents the conflicting bursts. The aim is minimizing the number of contending bursts so that optimizing channel utilization. In other words, the problem here is, selecting the maximum number of nodes who are non-adjacent with any other node from the interval graph. As stated in [39], this is an NP-Complete problem. A heuristic is developed in order to deal with this problem which is called perfect vertex elimination [16].

### 4.3.4 The scheduling algorithm

Input: Interval graph in Figure 4.22.

Output: Maximum stable set of the interval graph. The maximum stable set, includes the maximum number of nodes that are non-adjacent.

Lexicographic Breadth First Search (LexBFS) is performed on the nodes of the graph. Initially, each node of the graph is numbered from 1 to $n$ where $n$ is the number of nodes of the graph. Then, each number is labeled by the numbers of its adjacent nodes in decreasing order such that $L1 = [p_1, p_2, \ldots, p_k]$ where the node 1 is adjacent to k nodes whose numbers are $p_1, p_2, \ldots, p_k$ in decreasing order. Say, $L2 = [q_1, q_2, \ldots, q_l]$ and label $L1$ is greater than the label $L2$ when either for $i < j$, $p_i = q_i$

and $p_j > q_j$ or $p_i = q_i$ for $i \leq l$ and $k>l$. For instance [9, 7, 6, 1] < [9, 8, 5] and [6, 4, 3] < [6, 4, 3, 2]. The LexBFS is performed as folows:

- All the vertices of the interval graph are kept in a list
- The vertice with the greatest lexicographic label is selected and signed as visited.
- The visited vertice and all of its neighbors are popped from the list.
- The previous 2 steps are performed until the list becomes empty.

In [16], based on the performance analysis of Group Scheduling, it is shown that this technique leads to a decrease of 5% in burst loss rate in comparison to immediate scheduling technique that causes the least loss rate as it is seen in Figure 4.23.

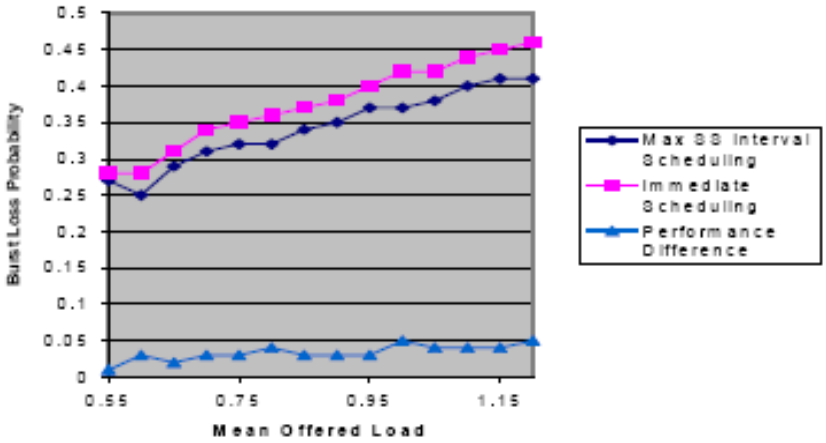



Figure 4.23. Group Scheduling and Immediate Scheduling in terms of loss rate vs offered load [16].

Here, finding the maximum lexicographic label takes $O(n)$ time and it is repeated $O(n)$ times so the complexity of the group scheduling technique is exptected to be $O(n^2)$. This shows that the scheduling time delay of this technique is greater than the void filling and non void filling techniques.

## 4.5 Segmentation Based Scheduling Techniques

In most of the OBS scheduling techniques, in case of a contention, the whole contending burst is dropped. Segmentation based techniques bring a new approach by dropping just the contending packets of a contending burst and scheduling the remaining part. Here, the burst is represented in terms of segments that may consist

of one or multiple packets. Each segment is a possible partitioning unit in case of a contention in the OBS domain [40]. In Figure 4.24, a simple segmentation scheme is illustrated in order to make the term clear.

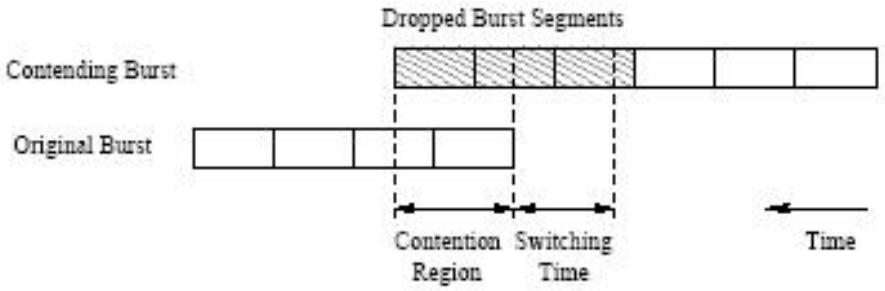



Figure 4.24. A simple segmentation scheme.

When two bursts contend, the problem comes from determining which segment of which burst to drop. Two policies are proposed for the solution of this problem. The first one is dropping the tail of the first burst (who arrived earlier) and the second one (who arrived later) is dropping the head of the second burst. In [40], it is stated that dropping the tail of the first burst is advantageous since in-sequence delivery of bursts may be provided later by retransmitting the lost ones.

Besides these, if there are packets that belong to more than one QoS classes in a burst, it is better to assemble the burst by using such a policy that the tail of the burst having a lower priority [22, 40, 41]. This kind of burst assembly approaches are discussed in Section 3. In the following sub-sections some proposed segmentation techniques are defined.

### 4.5.1 Segmentation with deflection

In order to perform deflection routing in OBS networks, for each source-destination (*s, d)* pair a deflection path is constructed. Consider a scenario such as the one illustrated in Figure 4.25. S sends a BHC to reserve available timeslots in the wavelengths along the path (S-A-B-D) that is constructed at the beginning of the connection. BHC attempts to reserve the resources along the path but at the link B-D it fails to reserve a time interval in the wavelengths. Here, B looks-up in the deflection route table for B-D and routes the burst by using the path B-C-D. [42]

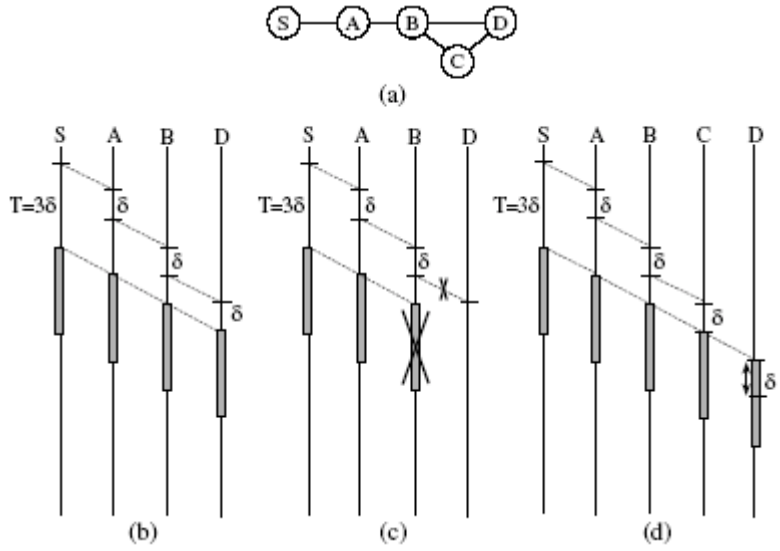



Figure 4.25. Deflection Routing
(a) A simple OBS network topology illustrating a connection between S-D (b) Succesful transmission from S to D. (c) Scheduling failes at link B-D (d) Deflection routing employed [42].

Segmentation with deflection technique uses the same principle in order to deflect the contending bursts. However, when contention occurs, the entire contending burst is either deflected or its contending part is segmented and deflected [40]. In Figure 4.26, segmentation with deflection scenario is shown for two contending bursts.

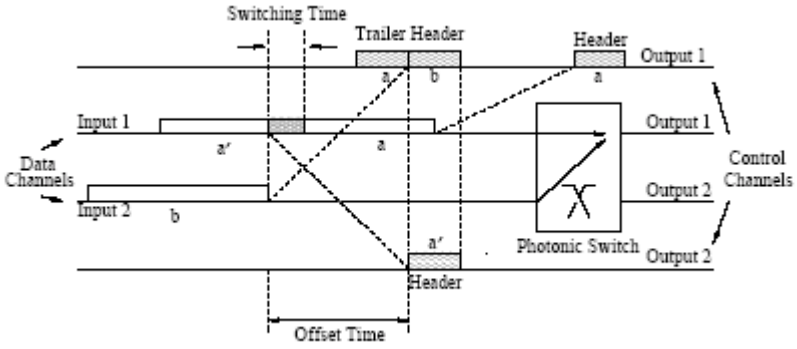


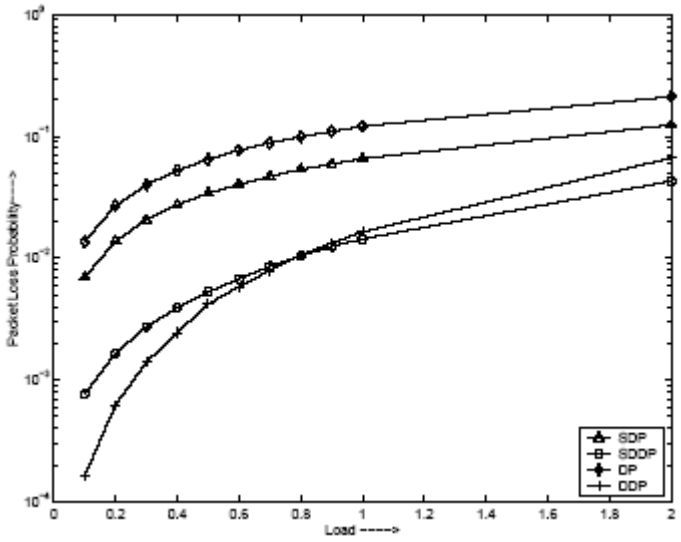

Figure 4.26. Segmentation with deflection scenario for two contending bursts [40].

Upon deflection, because of increasing the length of the path that the burst travels, the total processing time for the BHC may increase so the reservation process may not be completed before the arrival of DB. In order to prevent this, at the ingress node, a specific value is determined in order to specify the maximum number of hops that a burst can go through. When the number of hops that the BHC passed exceeds that value, the burst is dropped [40].

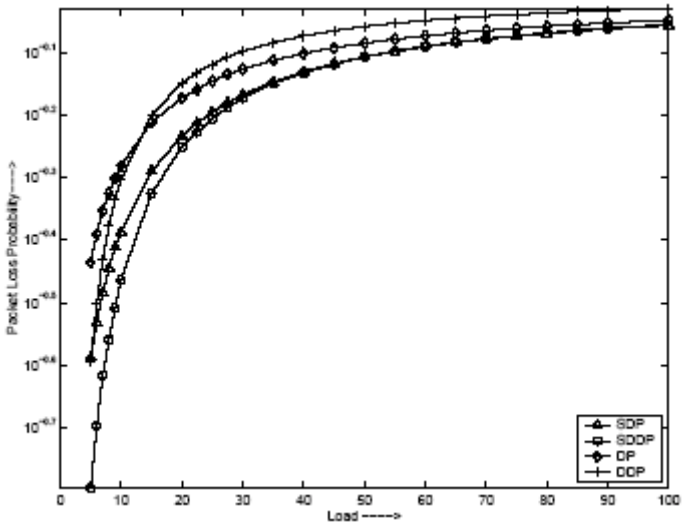
In OBS networks, in case of a contention one of 4 main policies may be employed. These policies are as follows [40]:

- *Drop Policy(DP) :* The whole contending burst is dropped.
- *Deflect and Drop Policy (DDP):* When two bursts contend, the contending burst is first deflected to an alternate output port. If the output port is not available, the burst is dropped.
- *Segment and Drop Policy (SDP):* In case of a contention of two bursts, the contending burst is segmented and attempted to be scheduled on a feasible void. If no feasible void is found, the entire segment is dropped.
- *Segment, Deflect, and Drop Policy (SDDP):* The contending part of the burst is segmented and deflected to an alternate output port. If the deflected segment cannot be scheduled on an interval, it is dropped.

The VF, Non-VF and Group Scheduling based techniques use either DP or DDP when contention occurs. The performance of these policies and segmentation with deflection is compared in [40]. In Figure 4.27 it is seen that SDP leads to lower packet loss rate when compared with DP since the contending packets have chance to be scheduled by segmentation. Moreover, DDP and SDDP results with lower loss rates and their performance is almost equal at low loads. However, SDDP leads to the lowest loss rate when the load is over 0.50 Erlang.




(a)




(b)

Figure 4.27. Loss rates of the segmentation based and non-segmentation based policies (a)under low load (b) under high load [40].

### 4.5.2 Delay First Minimum Overlap Channel (DFMOC)

In [15], two different non-void filling scheduling algorithms with FDLs are proposed. DFMOC is one of those algorithms. Here, it is useful to define some terminology related to the technique. The first term is *Overlap* that stands for the difference between the latest available unscheduled time of chanel $i$ ($LAUT_i$) and burst arrival time ($t_{ub}$), such that ;

$Overlap_i = LAUT_i - t_{ub.}$

The algorithm computes the minimum overlap values in each channel. If an available channel ($Overlap_i = 0$) is found, the data burst is scheduled on that channel. If all channels are busy ($Overlap_i \neq 0$), the channel with the minimum overlap is selected. While $L$ stands for the length of the burst and $MAX\_DELAY$ stands for the maximum duration that the burst can be delayed, the following steps are performed [15]:

- If $Overlap_i = LAUTi - t_{ub} > L + MAX\_DELAY$, then the entire burst is dropped.
- If $Overlap_i > MAX\_DELAY$, then the burst is delayed for a duration of $MAX\_DELAY$, non-overlapping burst segment is scheduled, the overlapping burst segment is dropped.
- Otherwise, the burst is delayed for the duration of $Overlap_i$, and scheduled on that channel.

In Figure 4.28, there is no available channel to schedule the arriving burst by the empolyment of the horizon technique. All the overlap values are computed, and $Overlap_2$ is found to be the minimum of all. The burst is delayed to the starting time of the first void plus a switching time and scheduled on that channel.

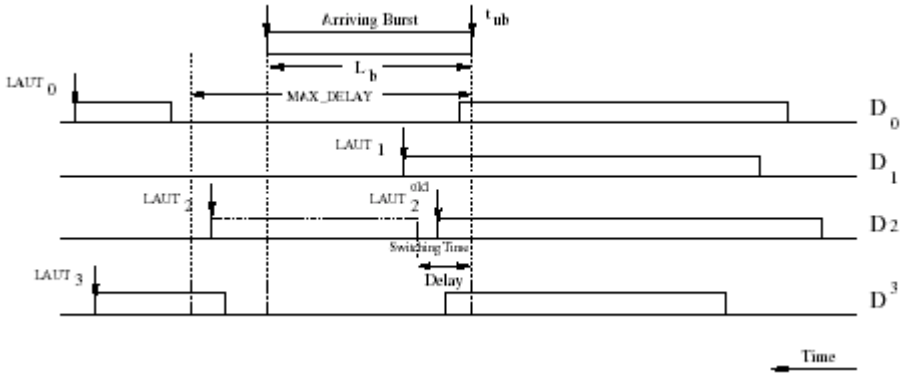



Figure 4.28. Illustration of DFMOC [15].

### 4.5.3 Segment First Minimum Overlap Channel (SFMOC)

SFMOC computes the overlap values on each channel. If an available channel is found ($LAUT_i = 0$), the burst is scheduled on that channel with minimum gap where:

$gap_i = t_{ub} - LAUT_i$.

Otherwise the following steps are employed [15]:

– If the overlap = $LAUT_i$ - $t_{ub}$ $\geq L + MAX\_DELAY$, then the entire burst is dropped.

– Else, the non-overlapping burst segment is scheduled on that channel. The overlapping burst segment is re-scheduled by the employment of DFMOC.

In Figure 4.29, there is no available channel to schedule the burst. $D_2$ is selected as the channel with the minimum overlap. The non-overlapping segment is scheduled on that channel. The overlapping is re-scheduled by the employment of DFMOC so $D_1$ is selected as the channel that requires minimum delay for the burst and the burst is scheduled on that channel by being delayed for the required duration.

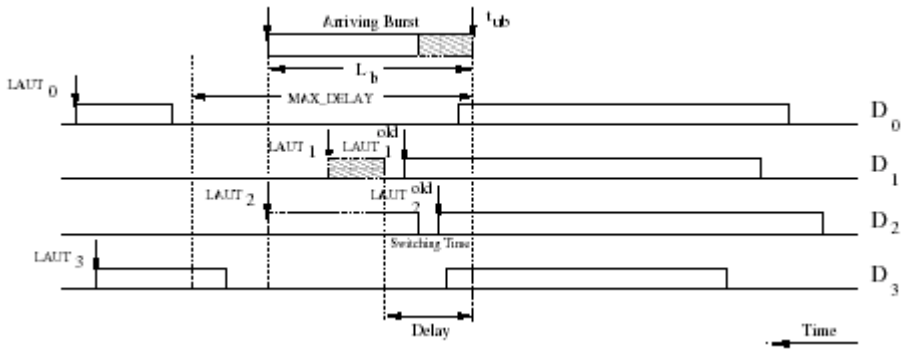



Figure 4.29. Illustration of SFMOC [15].

### 4.5.4 Delay First Minimum Overlap Channel with Void Filling (DFMOC-VF)

In [15], there are also 2 different void filling segmentation based techniques are proposed. These techniques are derived from the non-void filling ones (DFMOC, SFMOC) that are previously discussed in this section. DFMOC-VF is derived from DFMOC. A burst is represented by the parameters ($t_{ub}$ , $L$) where $t_{ub}$ is the burst arrival time and $L$ is the burst length so ($t_{ub} + L$) corresponds to burst's end time. This technique works as follows :

- The delay until the first void is computed for each channel. The channel with the minimum delay is selected.
- If all channels are busy ($LAUT_i = 0$ for all $i$) and starting time of the first void is greater than $t_{ub} + L$ + MAX_DELAY where ($t_{ub} + L$) corresponds to the end time of the burst, the entire burst is dropped.
- Else
    - If "starting time of the first void" > $t_{ub}$ + MAX_DELAY

- The burst is delayed for MAX_DELAY
- The overlapping segment is dropped, non-overlapping segment is scheduled.

o Else the burst is delayed until the starting time of the first void

- The overlapping segment is dropped, non-overlapping segment is scheduled.

In the scenario in Figure 4.30, all the channels are busy so $D_0$ is selected as the channel wih the minimum delay. The arriving burst overlaps with the scheduled burst which is represented by the start point $S_{0,0}$ and the end point $E_{0,0}$. Therefore, it is delayed until the starting time of the first void and scheduled there by the requirement of no segmentation.

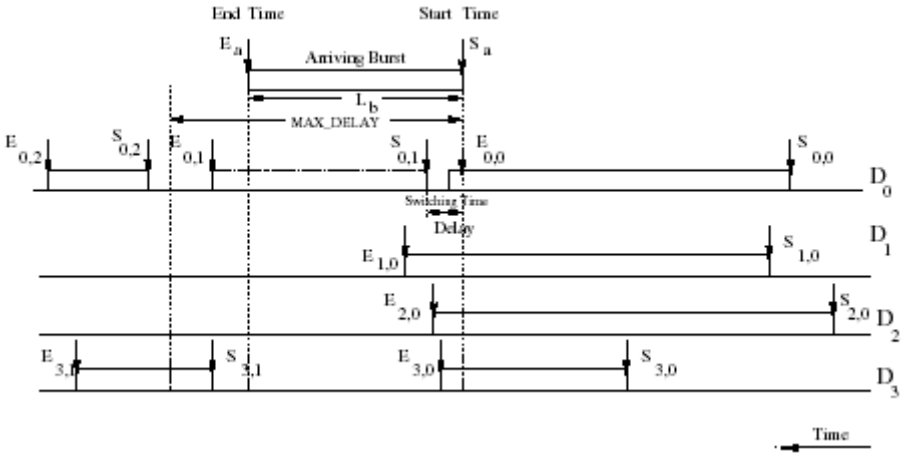



Figure 4.30. Illustration of DFMOC-VF [15].

### 4.5.5 Segment First Minimum Overlap Channel with Void Filling (SFMOC-VF)

SFMOC-VF is another void filling algorithm which is proposed in [15] and derived from SFMOC. Here, a burst is represented by the parameters ($t_{ub}$ , $L$) where $t_{ub}$ is the burst arrival time and $L$ is the burst length so ($t_{ub}$ + $L$) corresponds to burst end time. The technique works by the following algorithm :

- The loss on each wavelength channel is computed. The channel with the minimum loss is selected.
- If the "starting time of the first void" > ($t_{ub}$ + $L$ + MAX_DELAY), the entire burst is dropped.

- Else if the "starting time of the first void" > ($t_{ub} + L$), then DFMOC-VF is employed for the burst.
- Else the burst is segmented such that
    - The non-overlapping segment is scheduled.
    - The overlapping segment is re-scheduled by the employment of DFMOC-VF.

In the scenario in Figure 4.31, all channels are busy. $D_0$ is selected as the channel with the minimum loss. The non-overlapping part of the burst is segmented and scheduled on the channel in the interval with the starting time $S_{0,1}$ and the end time $E_{0,1}$. The overlapping part is scheduled by the employment of DFMOC-VF so $D_3$ is selected as the channel with the minimum delay and at the end that part is scheduled on the channel.

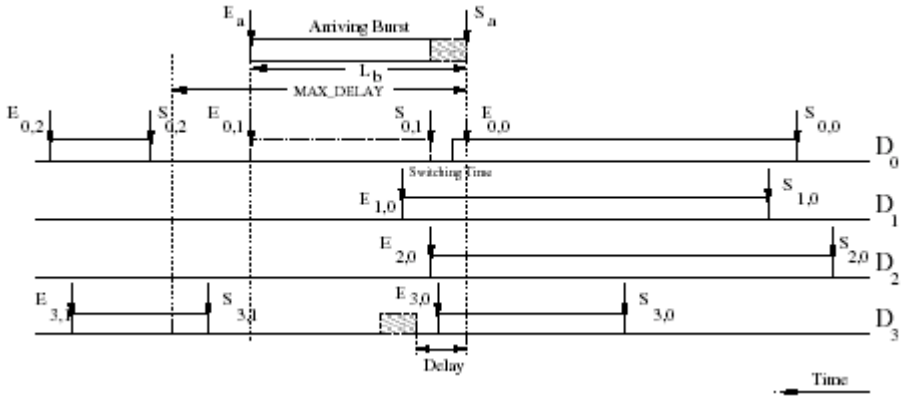



Figure 4.31. Illustration of SFMOC-VF [15].

In Figure 4.32, a performance comparison of the segmentation techniques with other proposed techniques is given. The first result drawn from the figure can be that the segmentation techniques lead to less loss rates in comparison to no-segmentation techniques at each link load. Another interesting result is that the "delay-first" techniques lead to lower loss rates. The reason is that the burst segments which are marked to be re-scheduled by the "segment-first" techniques may be discarded because of a previous reservation on the channel. Besides this, VF techniques always lead to lower loss rates as expected by theoretical analysis.

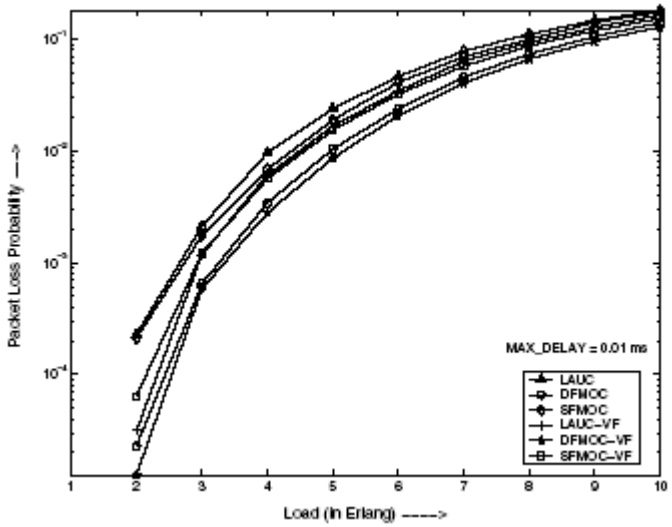



Figure 4.32. Segmentation and non-segmentation based techniques in terms of packet loss rate [15].

Another performance criteria in the study of segmentation based techniques is the average end-to-end delay. In Figure 4.33, it is clearly seen that "delay-first" algorithms cause a longer end-to-end delay at high loads since in case of a contention, DFMOC and DFMOC-VF utilize the channel by delaying the contending burst to a non-overlapping state.

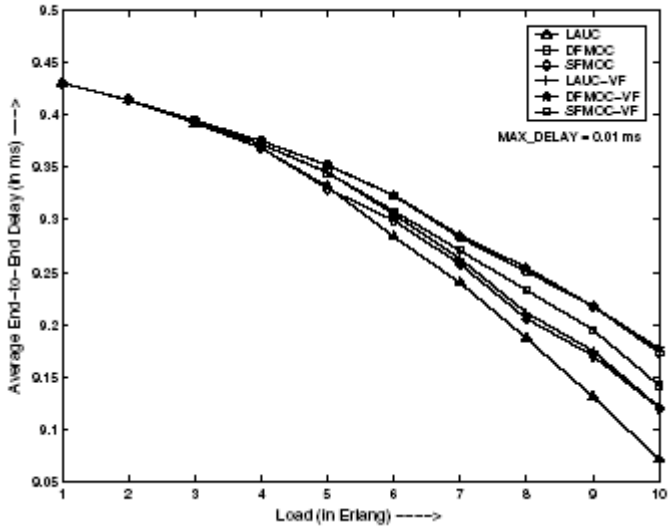



Figure 4.33. Segmentation and non-segmentation based techniques in terms of average end-to-end delay [15].

In general, it is also observed that, the segmentation based techniques lead to higher end-to-end delay. In LAUC-VF or other non-segmentation based techniques, an incoming burst is discarded when it contends with a previously scheduled burst but in segmentation based techniques, a re-scheduling scheme is employed for the

contending segment of the incoming burst. This increases the scheduling time for the incoming bursts and results with higher end-to-end delay[15].

## 5. SIMULATION ENVIRONMENT

The simulations run on a Pentium 4, 2.59 Ghz with 1GB available memory space. The simulation software is developed by using Visual C++ 6.0 tool.

The 14-node NSFNET topology given in Figure 5.1 is employed. It is assumed that there exist 8 wavelengths at each optical link consisting of 2 fibers, one for each direction, and at a line rate of 10Gbps.

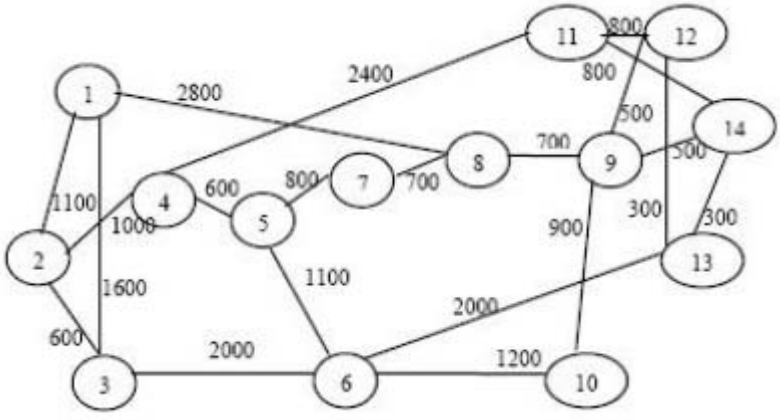



Figure 5.1. NSFNET topology used in our simulations

All the nodes are configured such that their routing tables provide source routing by using the Dijkstra's shortest path algorithm[23]. Each node can behave both as an edge node and a core node. Each node is equipped with a traffic generator. In our simulations we generate two types of packet traffic: Poisson and Self-Similar.

The average BHC processing time is taken as 100μs while the switching time for any cell is taken as 5μs.

The signaling protocol is configured to be JET as described in [18]. For an incoming DB, with its pre-determined offset time $\Delta$ and its length $\ell$, the corresponding control packet that arrives at time $t$ attempts to reserve the interval $(t + \Delta, t + \Delta + \ell)$ on an available wavelength.

From the beginning of the simulation until a pre-determined time (SIM_END) burst arrivals are observed. In our simulations, simulation duration covers a time period of 5 minutes. In each of the traffic type, each node has an equal probability to be a destination of an incoming burst.

## 5.1 Traffic Types and Their Modeling

### 5.1.1 Poisson Traffic Modeling

In Poisson traffic, the bursts are configured to follow an arrival rate of λ bursts per second. In order to satisfy this condition we configure the interarrival time between bursts to follow an exponential distribution with 1/λ as stated in [17].

In order to satisfy this, in the simulation program we compute the interarrival time between two consecutive bursts such that we obtain a random number in the interval (0, 1) and call it $X$. If current time is called $t$, it is assumed that at $(t+\Delta t)$ a burst will arrive so the following formula runs to compute the next arrival

$$\Delta t = -\lambda * \ln(X) \tag{5.1}$$

In Poisson modeled traffic, the bursts are assumed to be formed and to arrive following a Poisson distribution so a burst assembly scheme is not required to be implemented here. Thus, the burst sizes are assumed to follow an exponential distribution with 40μs which corresponds to 50KB on 10Gbps link rate.

The traffic load is configured by changing the interarrival time (λ) of the bursts.

### 5.1.2 Self-Similar Traffic Modeling

In self-similar traffic, the traffic is bursty at some timescales and shows structural similarity in long timescale [21]. We use the same parameters in [20] in order to configure self-similar traffic. An input traffic is set up by assigning 20 ON/OFF sources to the nodes whose ON/OFF periods show a Pareto distribution. The probability distribution function of Pareto distribution is as follows:

$$F(x) = 1 - \frac{1}{x^a} \tag{5.2}$$

Here, α gives the characteristics of the distribution tail. As the number of sources increases the Pareto distributed ON/OFF sources model converges to self-similar traffic model with Hurst parameter H = (3-α)/2. As known, Hurst parameter shows the level of self-similarity so the traffic is said to be self similar if $0.5 \leq H < 1$.

In measuring the Hurst parameter, we use the logarithmic plotting approach that is described in [26]. As an example, in Figure 5.2 a self-similar traffic pattern is seen. The time granularity of the pattern in Figure 5.2.a is less than the time granularity of the pattern in Figure 5.2.b.

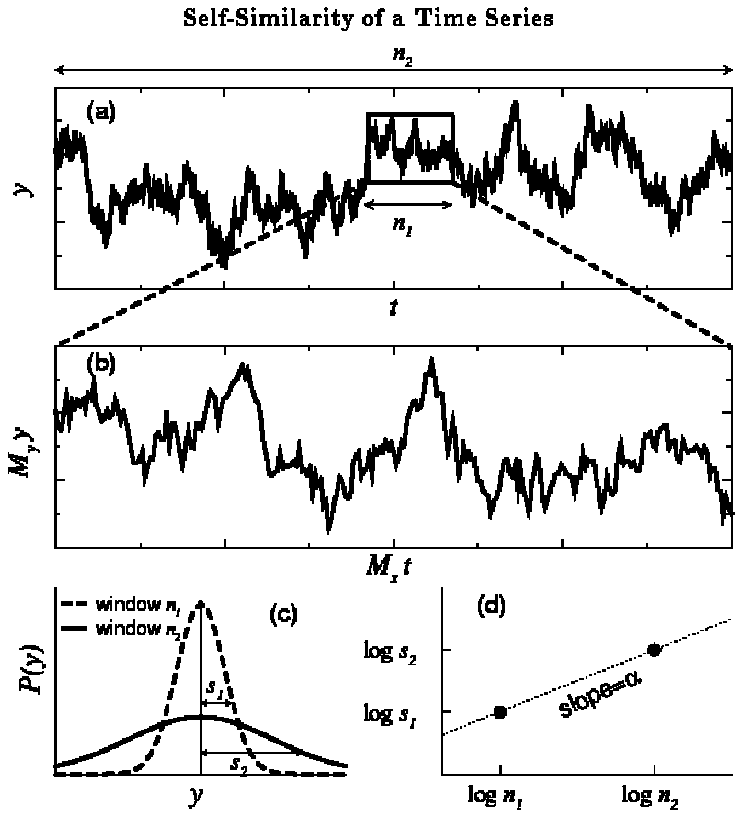



Figure 5.2. An example of a self-similar time traffic and obtaining the self-similarity [26].

The time granularities are given as *n1 and n2*. The standard deviation of the distribution of the traffic is computed from the figures and named as *s1* and *s2*. Then in Figure 5.2.d the logarithms of the granularities are placed in the horizontal axis while the standard deviations are placed on the vertical axis of the logarithmic plot

plane. Each of the couples are represented by the points in the plane and the slope of the line that connects the points gives the degree of self-similarity of the pattern.

## 5.2 The Burst Assembly Schemes and Their Implementation

In our simulations we implement 2 different burst assembly schemes in order to form optical bursts from incoming IP packets when we generate self-similar traffic at the ingress nodes. As explained in Section 3, there are various types of proposed burst assembly schemes. In our simulations we use a timer threshold based burst assembly which is used in [20] and a hybrid burst assembly scheme that uses both a time threshold and a queue size threshold.

### 5.2.1 Time Threshold Based Burst Assembly

In time threshold based burst assembly we construct a virtual queue and a timer for all destination nodes at each ingress node. At the beginning of each burst assembly period we determine a time threshold value for each destination. In our simulations we run our simulation for 2 different time threshold values of 25 times an IP packet duration and 40 times an IP packet duration which corresponds to 10μs and 16μs on 10Gbps link rate respectively. Besides these, we also determine a minimum burst size which is 1μs (1250B on 10Gbps).

The algorithm works as follows:

*Step 0:* At the beginning all queues are empty and timers are reset.

*Step1*: Check the timer value

-If timer is equal to the time threshold than check the queue size

- If queue size is less than the minimum burst size, then pad it to minimum burst size.
- Generate the burst and route it to the corresponding destination.
- Reset the timer and queue size and wait for the new packets' arrival

- Go to Step 1.

-Else, wait for the new packets' arrival and push the incoming packets to the corresponding queue.

-Go to Step 1.

### 5.2.2 Hybrid BurstAssembly

In hybrid burst assembly, we construct a virtual queue and a timer for all destinations. At the beginning we determine a time threshold *TTh*, minimum burst size (lower bound) and a size threshold (upper bound) *STh* for the bursts to be generated. Our simulation runs hybrid burst assembly algorithm also for upper bounds that are 25 and 40 times of an IP packet size (10μs and 16μs). Lower bound we use is also 1μs here while we guarantee the burst size not to exceed the queue size threshold which is 125KB (100μs on 10Gbps) as we obtained as the result of analysis of [24] and [25]. The algorithm works as we define in Section 3.3.

## 5.3 Modeling the Events

In our simulations, we use 5 different OBS techniques that are the horizon scheduling algorithm, LAUC-VF, First-Fit Void-Filling, Group Scheduling and DFMOC-VF. In this section the data structures that we use when modeling the switching techniques are described.

### 5.3.1 The Horizon Algorithm

In the horizon scheduling algorithm, the scheduler module of an optical link keeps track of the intervals that are the ones with the latest starting times which are called the horizons. The data structure here, is so simple such that a single array whose elements are of type timeslots and whose size is *W*, where *W* is the number of wavelengths on that optical link.

When a new burst header arrives, the scheduler computes the arrival time of the data burst and compares this arrival time with each element of the horizon array. If any elements of the array is less than the arrival time, the scheduler selects the one with the latest starting time and schedules the burst. Upon scheduling the burst, the current

element of the horizon array is updated and is set to the value of the currently scheduled burst's departure time.

### 5.3.2 LAUC-VF and First-Fit VF

The data structure for the LAUC-VF and the First-Fit VF techniques is more complicated in comparison to the one for the horizon technique. The scheduler of an optical link has to keep track of all idle intervals (voids). The scheduler searches for a feasible interval in order to schedule the incoming burst. Here, it is better to remember what is meant by a feasible interval. The following module is used as a key to decide whether the void is a feasible one for an incoming burst.

BHC:

arrival_time = $a$;

corrseponding_DB_offset_time = $\Delta$;

corresponding_DB_length = $\ell$;

Interval:

starting_time = $Is$;

end_time = $Ie$;

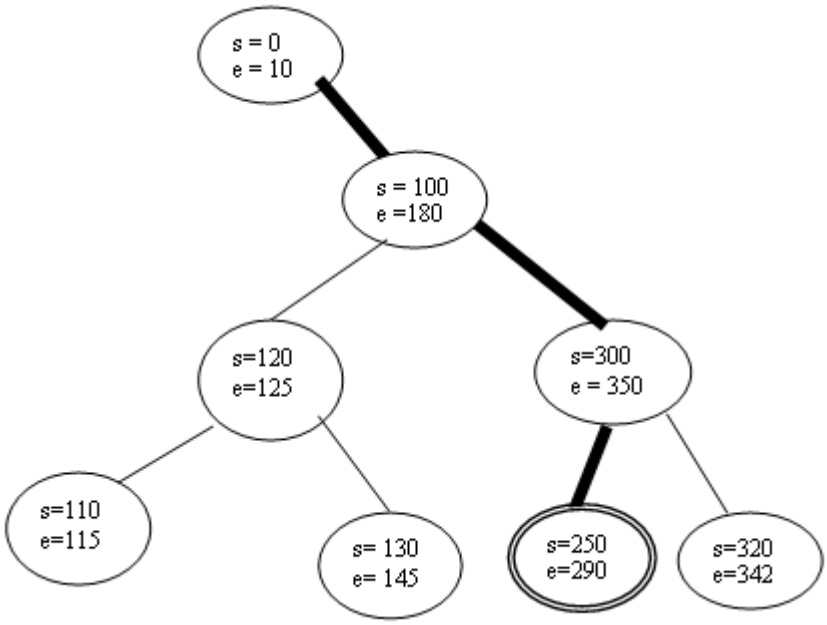
An interval I($Is$, $Ie$)is feasible for the corresponding DB of a BHC b($a$, $\Delta$, $\ell$) if and only if the following equation holds:

$$Is < (a + \Delta) \text{ and } Ie > (a + \Delta + \ell) \tag{5.3}$$

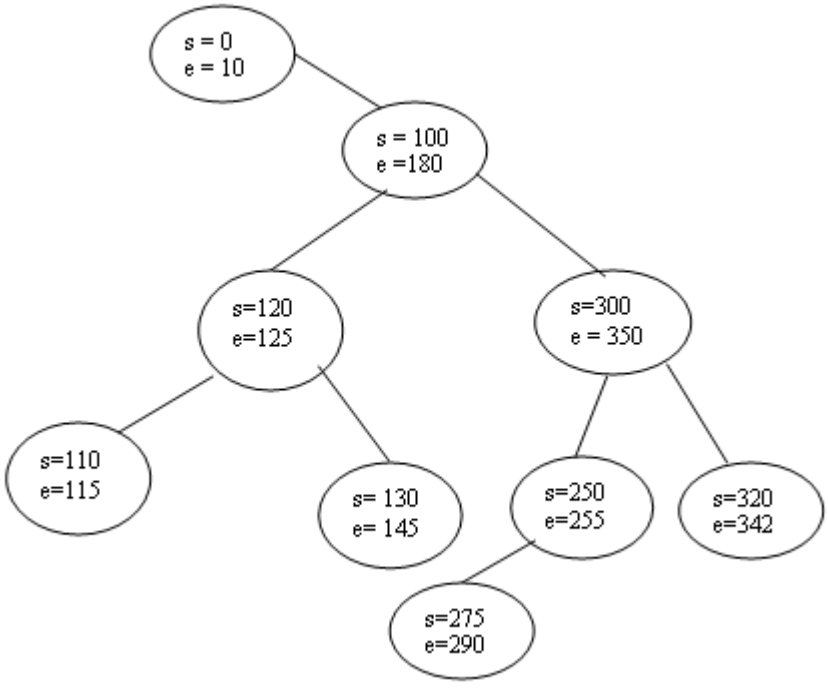
The time complexity for the search process has to converge to a minimum value so for each wavelength, a binary tree is constructed as shown in Figure 4.3. Each node of the tree represents an interval with a specific starting time and an end time. At the beginning, the tree consists of just one node whose starting time is 0 and whose end time is infinity. When the first burst arrives, it is scheduled on that channel.

Upon scheduling the burst, the recently found feasible interval has to be split up into 2 new intervals such that, if the burst $b(a, \Delta, \ell)$ is scheduled on the interval $I(Is, Ie)$ then the interval is split up into $I_1(Is, a+\Delta)$ and $I_2(a+\Delta+\ell, Ie)$. The scheduler modifies the node that represents the found interval as setting the end time to be $a+\Delta$, and creates a new node for the second interval $I_2$. The new node is pushed into the tree by

satisfying the rule that the left child of a node has to be always less than its parent and the right child has to be greater.




(a)




(b)

Figure 5.3. A Sample binary tree to search the feasible intervals.
(a) A burst with start and end times (255,275) arrives; the interval (250,290) is found as the feasible void by the bold path. (b) The structure of the tree upon scheduling the arriving burst on the feasible interval

The binary tree in Figure 5.3.a represents the void intervals. Consider a burst whose arrival time corresponds to the 255$^{th}$ timeslot and departure corresponds to the 275$^{th}$ timeslot. The search procedure includes the path that is dashed thick in Figure 5.3.a. Upon the modification and the new interval generation procedure, the tree structure in Figure 5.3.b is obtained.

In the simulation, if LAUC-VF is employed, the feasible intervals are collected from the binary tree of each wavelength and the one with the latest starting time is selected to schedule the burst. Otherwise, if First-Fit-VF is employed, the scheduler stops searching when a feasible interval is found.

### 5.3.3 Group Scheduling

In order to achieve group scheduling, we form a similar structure as we explain in Section 4.3. We add a new property to the BHCs pointing the window they are requesting for. In our simulation, a window has a size of 20 bursts which corresponds to 800μs. The BHC collecting deadline for each window is taken as *wstop – Tp*. Each node is equipped an array of baskets for each wavelength and window so since our simulation covers a time duration of *SIM_END*, the number of baskets for a destination is *SIM_END / WSIZE*. A basket just consists of bursts.

The wavelength assignment is performed by using a round-robin scheme. Each node also keeps a pointer to the last used wavelength and assigns the incoming BHC to the corresponding basket of the wavelength which follows the last used wavelength.

In the simulation program, when a BHC arrives at a node, the node just checks which window the corresponding DB belongs and then pushes the BHC to the related basket. When the global time is equal to the collecting deadline for current window, the node pops all of the bursts from the baskets for all wavelengths on each link, and constructs an interval graph as we mentioned in Section 4.3 by finding the contending bursts. Upon running the perfect vertex elimination algorithm [16], it obtains a set of nodes that correspond the non-contending bursts so it schedules the corresponding bursts while dropping the remaining ones.

### 5.3.4 DFMOC-VF

Since DFMOC-VF is a segmentation based technique, we add a property to the bursts that shows the percentage of the segmented part of the burst. Besides this, the

simulation uses the same data structure that is used for LAUC-VF and First-Fit-VF scheduling. When DFMOC-VF is called by the simulation, initially First-Fit-VF is run in order to find a feasible void to schedule the incoming burst. If a feasible interval is found, the burst is just scheduled on that interval.

The overhead of DFMOC-VF algorithm occurs when the incoming burst is marked to be dropped since a feasible void cannot be found. Here, for all wavelengths, DFMOC-VF scheme uses the same binary tree structure and the end time of the corresponding DB of the currently arrived BHC as the key $( a + \Delta + \ell)$ in order to find the voids whose starting time ($Is$) is the nearest to the end time of the burst. Then the scheduler calculates the overlap for all of the voids such as the following;

$$Overlap_i = Is - ( a + \Delta + \ell) \tag{5.4}$$

Upon calculating the overlap values on each channel the scheduler follows the algorithm below:

- If $(MAX\_DELAY + (a + \Delta + \ell)) < Is$
  - Drop the whole burst
- Else if $(a + \Delta) < Is$ then
- Delay the burst for MAX_DELAY, segment the contending part and drop then schedule on the interval $(Is, a+\Delta+ \ell)$.
- Else delay the burst up to $Is$ and schedule it on the interval.

In implementation of DFMOC-VF, FDLs are required so we also implement FDLs in simulation with the longest delay line size equal to the maximum burst size which is 100μs.

# 6. SIMULATION RESULTS

In fact, in the literature it is hard to find a work in which the performance metrics are compared by taking into account the overall network load since most of the work is done by just focusing on one link between two optical nodes. Since we evaluate the overall network performance, it is not possible to compare the numerical results with the other works in literature. For this reason, we generate our test environment by simulation. This section gives the results obtained for various optical burst switching techniques under Poisson and self-similar traffic.

## 6.1 Results Under Poisson Traffic Flow

In the first part of our simulations we generate a Poisson traffic flow and change the load by varying the λ parameter. Initially, we compare the performance of Void-Filling and Non-Void-Filling techniques in terms of burst loss rate. In Figure 6.1 the overall burst loss rates are given when the horizon, LAUC-VF and First-Fit-VF are employed at various overall network loads when NSFNET topology is employed.

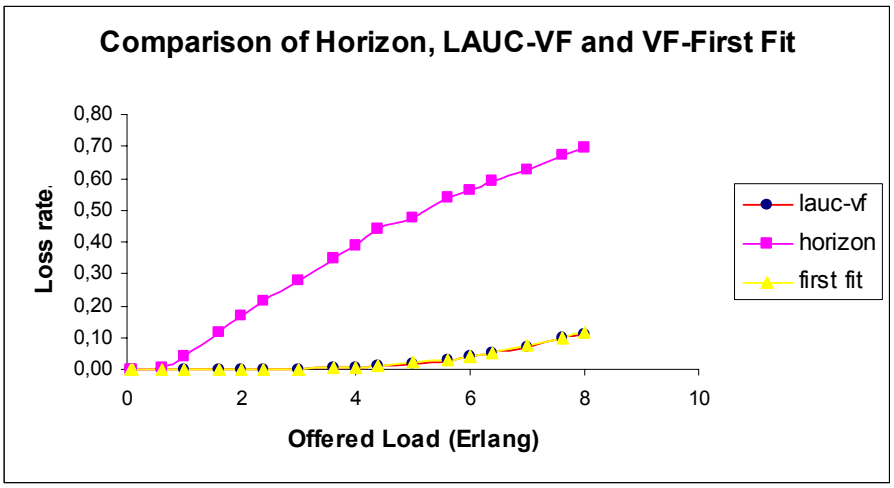



Figure 6.1. The comparison of performances of Horizon, LAUC-VF and First-Fit-VF in terms of burst loss rate for various network loads.

It is observed that the loss rate in the horizon algorithm is higher and increases faster than the LAUC-VF and First-Fit-VF. The reason is that the horizon does not utilize the voids as the VF techniques do so it leads to lower utilization which causes high loss rate.

Besides this, the performance of LAUC-VF and First-Fit-VF seems to be equal in terms of loss rate with a slight increase in the loss rate of First-Fit-VF until an aggregate network load of 7 Erlang. After the aggregate load of 7 Erlang LAUC-VF seems to have a slightly less loss rate than the First-Fit-VF. Most of the time the performance of the both VF techniques have almost the same performance behaviour although First-Fit-VF tends to cause a slight more loss rate at most of the loads. The reason for this is that LAUC-VF schedules the burst on the void which has the latest starting time among all the feasible intervals. By doing so, LAUC-VF accepts the DBs of the BHCs that will arrive later but whose offset time is less than the currently scheduled burst. Therefore, LAUC-VF leads to a slight decrease in loss rate in comparison to First-Fit-VF. The results obtained comply with the work done in [13].

Upon comparing the performances of VF and Non-VF techniques we compare the performance of the immediate OBS scheduling techniques and Group Scheduling in terms of burst loss rate. In Figure 6.2, it can be observed that starting from a load of 5.6 Erlang up to full utilization load Group Scheduling leads to an obviously less loss rate in comparison to both LAUC-VF and First-Fit-VF techniques. The reason is so simple; as it is explained in the previous sections, the immediate scheduling techniques attempt to schedule the DB of the incoming BHC as fast as possible while Group Scheduling collects the BHCs that request a period of time and it attempts to obtain a scheduling scheme which uses the available wavelength channels in the most effective way. The results in Figure 6.2 comply with the results in [16].

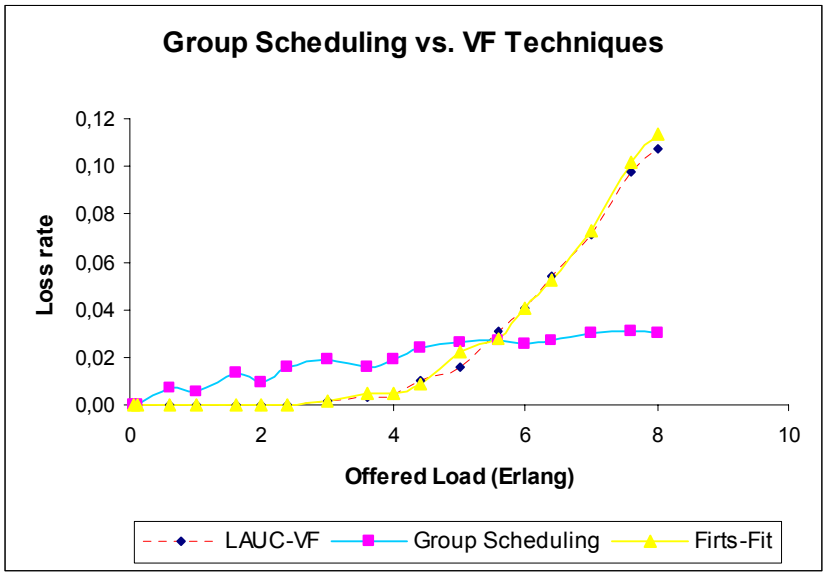



Figure 6.2. The comparison of performances of OBS immediate scheduling techniques and Group Scheduling.

Although we use the window size as the size of 20 bursts in our simulations for Group Scheduling, we also show the effect of window size in the performance of Group Scheduling in Figure 6.3. We run our simulation with window sizes of 10, 20 and 50 bursts for various network loads. We observe that the increase in the window size leads to lower loss rates. However, as the window size increases, the offset time assigned to the bursts should also increase in order to wait for the deadline of the BHC collecting period so increasing the window size will also cause an increase in end-to-end delay. Therefore, selecting the optimum window size also seems as a challenging problem here.

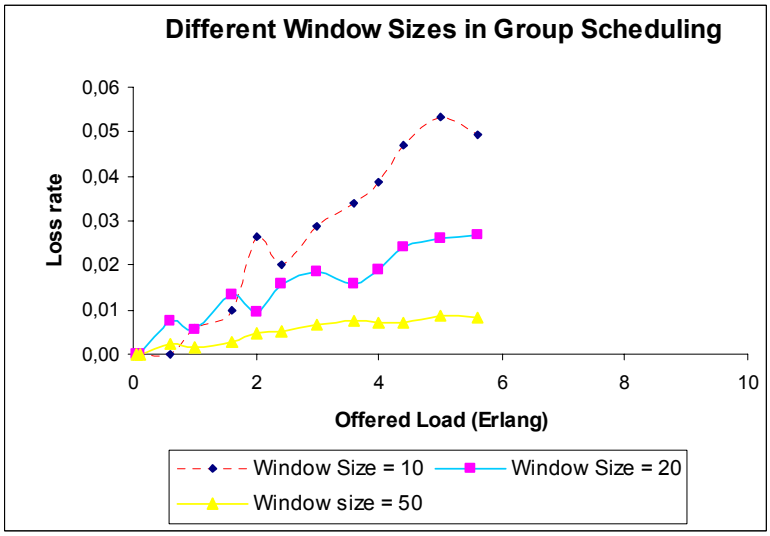



Figure 6.3. The comparison of performances of segmentation based techniques and non-segmentation-based techniques.

In Figure 6.4, we give the results obtain by comparing the loss rates of DFMOC-VF with the other techniques that we give in the previous figures. The aim here is just to compare the performance of segmentation and non-segmentation based optical burst switching techniques in terms of loss rate. As it is seen, DFMOC-VF leads to the least loss rate even less than Group Scheduling. DFMOC-VF leads to the least loss rate since in non-segmentation based techniques, if a DB of a BHC contends with another burst in all wavelength channels the whole burst is dropped while in segmentation based techniques the non-contending part of the burst survives and is segmented. The result that we obtain support the work in [15].

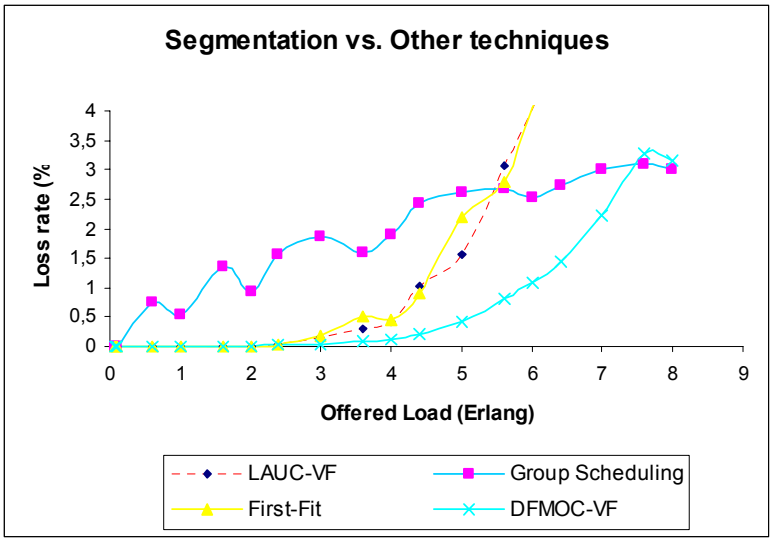



Figure 6.4. The comparison of performances of segmentation based techniques and non-segmentation-based techniques.

Upon comparing the performance of the OBS techniques in terms of loss rate, we run our simulation in order to evaluate the performance of the techniques in terms of the average scheduling time per burst. In Figure 6.5, as expected, the least scheduling time is spent by the horizon algorithm since the data structure it uses requires a binary search among the latest available unused times of the wavelength channels on the link. The First-Fit-VF and LAUC-VF lead to higher burst scheduling duration in comparison to the horizon algorithm since they perform a binary search by using all of the idle intervals. However, First-Fit-VF has a less scheduling delay than LAUC-VF since it schedules the burst on the first found feasible interval. LAUC-VF has an overhead of obtaining a set of feasible intervals and performing a new search on that set in order to obtain the void with the latest starting time. Group Scheduling and

DFMOC-VF have the highest scheduling duration. DFMOC-VF leads to a high burst scheduling delay since it first performs a LAUC-VF scheme and then tries to delay and segment the non-contending part of the burst as explained in Section 4.5.4. So DFMOC-VF always has a burst scheduling delay of at least as much as the LAUC-VF scheme. Besides these, Group Scheduling also leads to the highest burst scheduling duration. Here, the reason is in order to schedule the DBs of the BHCs it delays the BHCs until the end of the collecting period for the window they request. Moreover, as we analyze in Section 4.3.4, the complexity of the perfect vertex elimination algorithm proposed in [16] is $O(N^2)$ where N is the number of bursts requesting for the same window so the results also comply with the analysis.

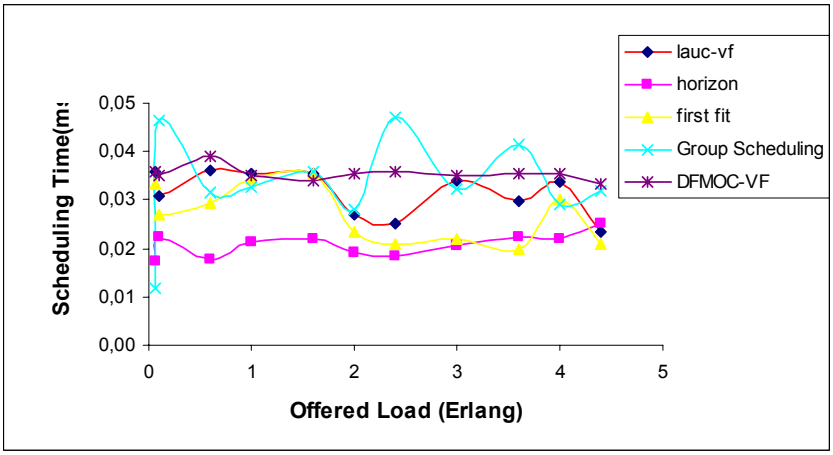



Figure 6.5. The comparison of average scheduling time per burst for all of the simulated techniques.

In the next section we run our simulations under self-similar traffic and compare the performance metrics with the ones obtained in Poisson traffic for each OBS scheduling technique, and under different burst assembly schemes.

## 6.2 Results Under Self-Similar Traffic

In the second part of our simulations, we generate the self-similar traffic which we define in Section 5.1.2 with Hurst parameter 0.9. The results we obtain under self-similar traffic can be examined under different subtitles such that the traffic characteristics obtained at the ingress nodes, performance of OBS scheduling techniques inside the domain, and the traffic characteristics obtained at the egress nodes. At the igress nodes, we measure the Hurst parameter just after the burstification process so that we can observe the effect of burst assembly on the

burstiness of the traffic. Besides this, we examine the distribution of the burst sizes under different assembly techniques and *TTh, STh* parameters by partitioning the time interval covering maximum burst size duration into 4 partitions as shown in Figure 6.6. Bursts are classified according to their sizes such as minimum size bursts (1µs), maximum size bursts (equal to or greater than 100µs), small size bursts (1-25µs), medium size-1 bursts (25-50µs), medium size-2 bursts (50-75µs), and large size bursts (75-100µs). We implement the burstification schemes as defined in Section 5.2.1 and Section 5.2.2. Actually, in order to understand the relation between the drop rate and burst size, we study the burst size distribution at the egress nodes too. Moreover, we also compare the performance of the burst assembly techniques in terms of burstification delay.

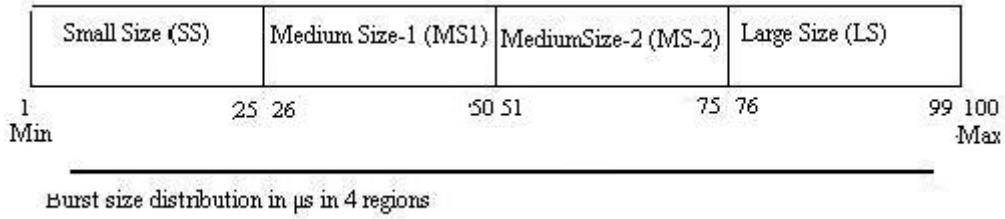



Figure 6.6. A Scheme to classify the bursts according to their duration in µs.

At the egress nodes we first study the burst loss rates using five different OBS scheduling schemes. Moreover, we measure the Hurst parameter of the outgoing traffic for each of the OBS scheduling technique employed in the network. Another study taken is on the comparison of packet loss rate and burst loss rate in order to observe the effect of burst duration on the loss rate.

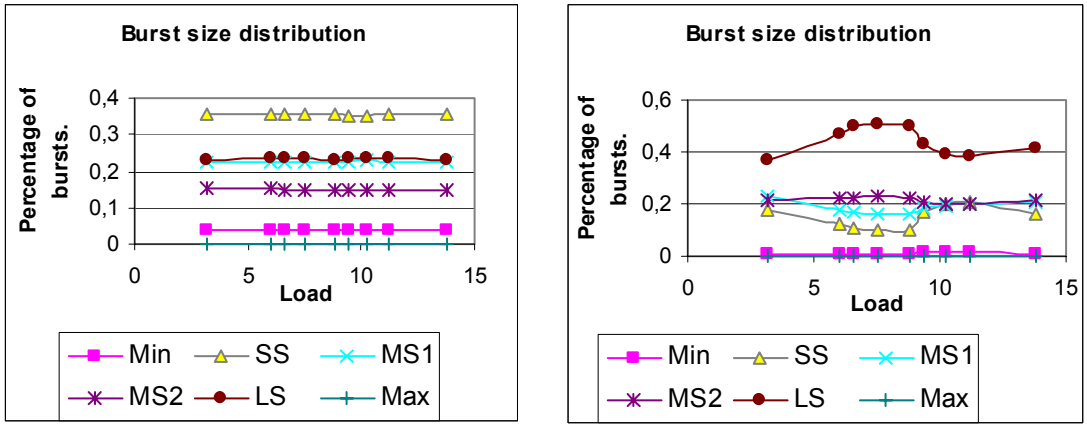
### 6.2.1 Traffic Characteristics at the Ingress Nodes

#### 6.2.1.1 Under Time Threshold Based Burst Assembly

Upon generating the self-similar traffic, at each ingress node we first employ the time threshold based burst assembly scheme. We run our simulations for two different *TTh* values, that are 25 and 40 times of an IP packet duration on 10 Gbps. First, we set the time threshold to be 25 times an IP packet duration, which corresponds to 10µs. And measure the self-similarity of the traffic after the burst assembly. We measure the output Hurst parameter at the ingress nodes under an average network load which corresponds to 7 Erlang and a high network load which

corresponds to 13 Erlang. As shown in Figure A1, and Figure A2. As it is observed, the time threshold based burst assembly at the ingress nodes decreases the self-similarity significantly in short term. However, although the Hurst parameter is decreased to 0.52 from 0.9, the self-similar nature of the traffic still goes on. Besides this, we also observe that, the output self-similarity of the traffic does not depend on the network load at the ingress nodes of the OBS domain.

The next step is analyzing how the distribution of the size of the generated bursts is affected by the employment of this scheme. As it is seen from Figure 6.7.(a), about 35% the generated bursts are from the 1st region which lead to small size bursts. These are followed by the large size bursts with a rate of about 25%. In general, the rate of the large bursts (Max, LS, and MS2) seem to be about 40% of the generated bursts.




(a) At the ingress (b) At the egress

Figure 6.7. Distribution of burst sizes. *TTh* = 10µs
(a) At the ingress nodes after assembly (b) At the egress nodes after employing the Horizon algorithm inside.

In Figure 6.7.(b), we show the distribution of burst sizes at the egress routers of the OBS domain when the Horizon algorithm is employed inside the network. The reason why we employ the horizon for this analysis is the horizon's high loss rate to obtain significant difference between generated and loss bursts in order to analyze these kind of experiments as we discussed in the previous sections. If we look at Figure 6.7.(b), it seems that most of the bursts (about 70%) that can be switched and scheduled along their path in the intermediate routers are large bursts (Max, LS, and MS2) because of the traffic shape due to the burst assembly scheme. As it is seen, the smaller size bursts have higher drop probability in case of a contention inside the

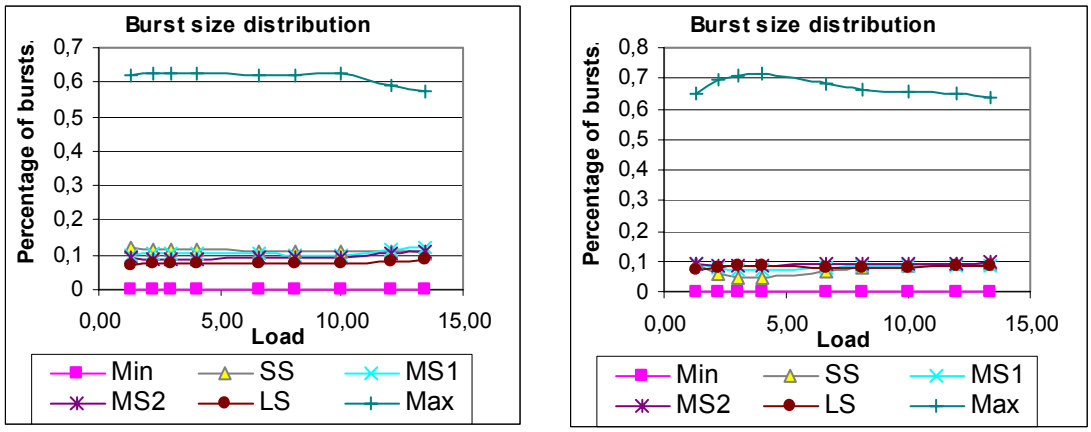
OBS domain since a long burst reserves. Another result that can be drawn from the figures is that the distribution of burst sizes are independent of the network load.

The burstification delay for all bursts is 10μs here.

In order to observe the effect of the time threshold in the performance and the traffic characteristics of OBS networks at the ingress routers, we set the *TTh* to 40 times an IP packet duration which is 16μs on 10Gbps. Initially, we measure the Hurst parameter just after the burst asembly. As it is seen from Figure A3 and Figure A4, at all loads the short term characteristics of the incoming traffic is changed. Here, as a result of buffering the IP packets for a longer period the burstiness of the traffic is decreased a little bit more than the previous experiment. As shown in the figures, the output Hurst parameter is decreased to 0.51 here.

The next experiments are about the analysis of the distribution of the burst sizes at the ingress nodes just after the burst assembly, and at the egress nodes after employing the horizon scheduling algorithm inside the OBS domain. As it is seen from Figure 6.8.(a), increasing the time to wait for generating the bursts (*TTh*), most of the bursts' (60% of the generated bursts) size exceed the 100μs duration. There is almost no burst whose size is equal to the minimum burst size. The percentage of all the bursts of the other sizes (SS, MS1, MS2, and LS) have the same ratio (~10%) after the assembly.

In Figure 6.8.(b), the distribution of the burst sizes at the egress nodes after running the horizon algorithm at the intermediate nodes is illustrated. Similar to the previous experiment, the highest amount of the transmitted bursts (70%) consists of the bursts of the maximum size. Here, again the longer bursts reserve more resources and the shorter bursts are dropped when they contend with the long bursts.




(a) At the ingress (b) At the egress

Figure 6.8. Distribution of burst sizes. *TTh* = 16µs

(a) At the ingress nodes after assembly (b) At the egress nodes after employing the horizon.

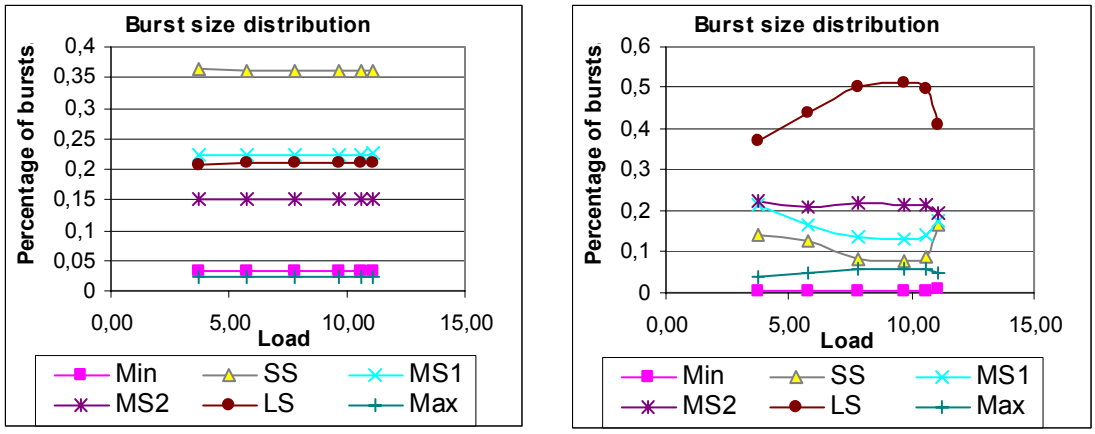
Based on the analysis above, it can be said that, the Time Threshold Based Burst Assembly Technique reduces the burstiness of the incoming packet traffic. However, it does not change the long-term characteristics of the traffic since the Hurst parameter measured keeps the self-similar nature of the traffic ($0.5 \leq H \leq 1$). It is also observed that increasing the *TTh* value decreases the burstiness of the traffic as a result of spending more time for buffering the incoming IP packets. Besides these, it is also clear that, the longer bursts are obtained as the *TTh* is increased, and those longer bursts have higher probability to survive in case of a contention in the wavelength channels. It is useful to note that the burstification delay is equal to the *TTh* as expected.

#### 6.2.1.2 Under Hybrid Burst Assembly

In order to perform the same experiments and compare the performance and the traffic characteristics at the egress nodes upon the employment of the two different assembly schemes we run the Hybrid Burst Assembly technique at the ingress nodes. At the first step, we set the *TTh* to 10µs which corresponds to 25 times an IP packet duration as we do in 6.2.1.1. In all the experiments in which Hybrid Burst Assembly runs, *STh* is taken as 100µs (125KB on 10Gbps). The difference of our scheme from the other time-and-threshold based burst assembly schemes is that, we do not let the queue sizes to exceed the *STh* in order to bound the burst sizes from above and below. Besides these, we also do not allow the bursts to wait at the ingress nodes more than the *TTh* value.

In Figures A.5-6, the traffic characteristics at an ingress router is shown for average (7 E) and high (14 E) network loads respectively. As seen from the figures, the network load does not affect the burstiness of the incoming traffic. Besides this, the Hurst parameter is decreased to the level of 0.515. This shows that the traffic still keeps its self-similar nature but its short-term behaviour is changed. The results here are similar to the results obtained from Time Threshold Based Burst Assembly. It can be said that by using these parameters, the effects of the two assembly scheme on the incoming traffic characteristics is almost the same.

Another characteristic that the burst assembly scheme may affect is the distribution of burst sizes. Intuitively, the bursts that are generated within a threshold time and bound from above and below by their size should be of short burst sizes and distributed on an average value by a standard deviation. Figure 6.9.a supports this thesis where a significantly higher number of bursts belong to the *SS* and *MS1* region. Most of the longer bursts belong to the *LS* region. However, the total rate of the longer bursts (maximum size, *LS*, and *MS2* regions) is about 40% where the shorter bursts (minimum size, *SS*, and *MS1* regions) make up the remaining 60% of the generated bursts. There is also another point to note that the distribution of the burst sizes in terms of long and short bursts show similarity to the results obtained from Time Threshold Based Burst Assembly technique.




(a) At the ingress (b) At the egress

Figure 6.9. Distribution of burst sizes. *TTh* = 10µs and *STh* = 100µs
(a) At the ingress nodes after assembly (b) At the egress nodes after employing the Horizon algorithm inside.

When we observe Figure 6.9.b, it seems that the longer bursts (*LS* and *MS2*) are close to be transmitted without dropping when the bursts are switched by the employment of the horizon technique inside the OBS domain. 40-50% of the switched bursts are

from the *LS* region so the characteristics of the burst size distribution is also so similar to the results obtained from Time Threshold Based Burst Assembly scheme.

Another performance metric that we consider is the burstification delay at the ingress nodes. In Figure 6.10, the change in the rate of the generated bursts versus waiting time in the queue is given. The results are taken under high (13 E) network load. It seems that about 85% of the bursts wait for the *TTh* (10μs) but the 15% of the bursts reach the *STh* value before the timer overflows so the average burst waiting time duration is reduced to 8.86μs. The reduction in the waiting time of the bursts also causes the Hybrid Burst Assembly scheme to lead to a lower end-end delay in the OBS domain.

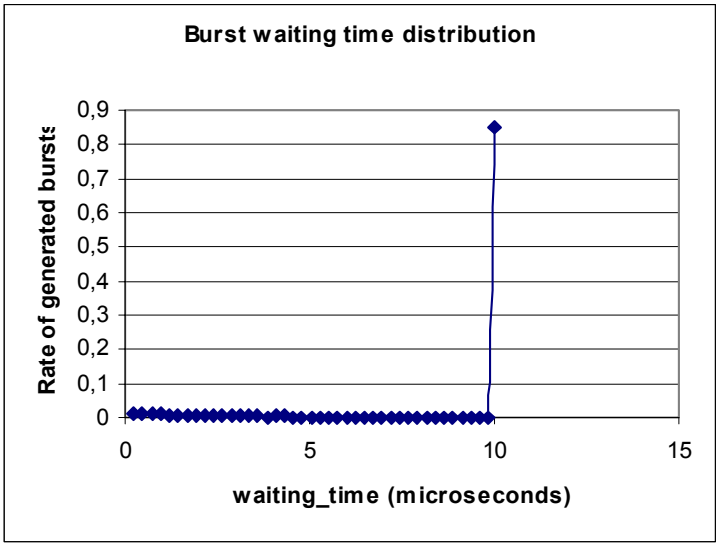
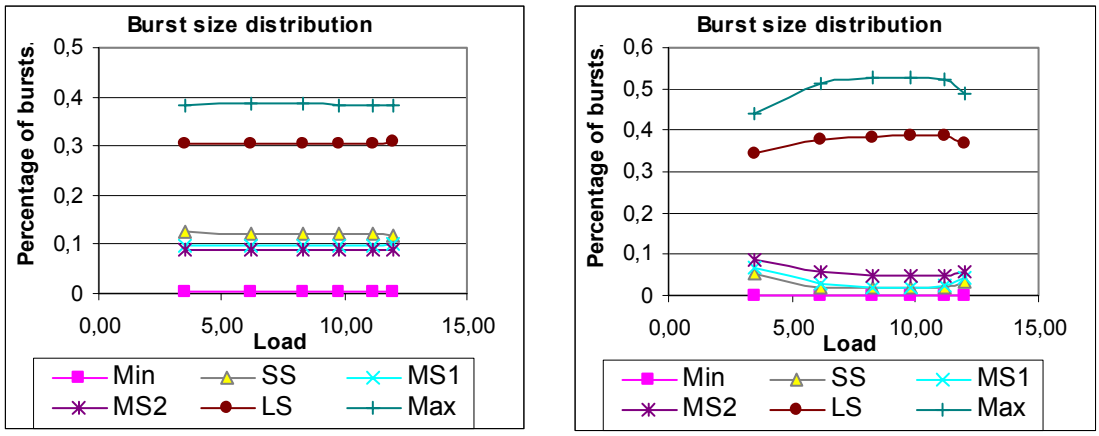



Figure 6.10. Burst waiting time distribution with *TTh* = 10μs and *STh* = 100μs.

Upon getting these results we run the same simulations by setting *TTh* to 16μs. The traffic characteristics at an ingress router is shown in Figures A.7-A.8 under average and high network loads respectively. The traffic characteristics are not affected from the load change. Here, we see that the Hurst parameter is reduced to a level of 0.53 which shows that increasing the *TTh* do not lead to a lower burstiness as it is in Time Threshold Based Burst Assembly. The reason is that, here, as the bursts are forced to have sizes in the interval [1, *STh*], increasing the buffer size does not cause the bursts to be delayed for a longer time at the ingress nodes. However, this decreases the waiting time (also the interarrival time) of the bursts. Therefore, by increasing the *TTh*, we obtain a closer but slightly worse trafic characteristics at the ingress nodes in comparison to time Threshold Burst Assembly in terms of self-similarity .




(a) At the ingress (b) At the egress

Figure 6.11. Distribution of burst sizes. *TTh* = 16µs and *STh* = 100µs. (a) At the ingress nodes after assembly (b) At the egress nodes after employing the Horizon algorithm inside.

In Figure 6.11, the distribution of the burst sizes are given at the ingress nodes and at the egress nodes when the horizon scheduling algorithm is performed inside the OBS network. About 38% of the bursts are of maximum size, and the others seem to be distributed to the other regions (Figure 6.11.a). When compared to the results obtained from Time Threshold Based Burst Assembly, it seems that the average burst size here is significantly less since it is not allowed to exceed the *STh* although there is still some time to wait. At the egress nodes upon employing the horizon algorithm, we again see that the bursts that are succesfully delivered to their destinations are the longer size bursts which are of maximum size and from the region *LS*. The bursts from remaining regions are just 10% of the outgoing bursts from the OBS domain since long bursts consume more bandwidth in the wavelength channels.

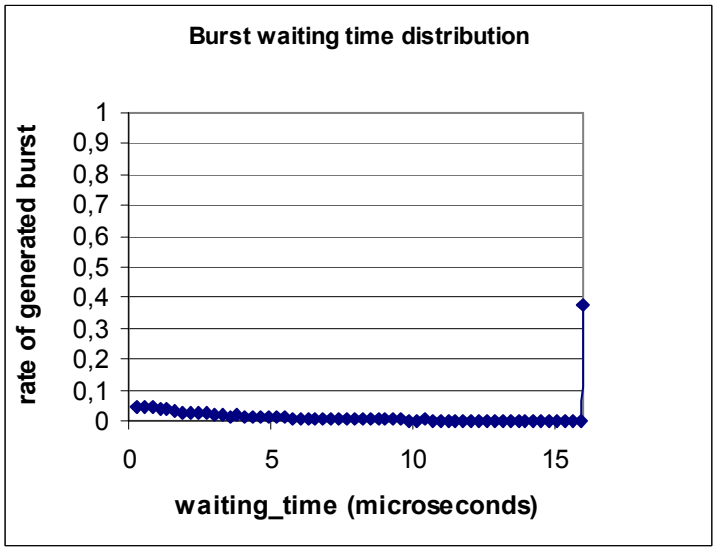



Figure 6.12. Burst waiting time distribution with *TTh* = 16µs and *STh* = 100µs.

In Figure 6.12 the distribution of the burstification delay is shown when *TTh* is 16μs. Here, the percentage of the bursts that wait for the timer overflow decreases to 40%. However, this reduction leads to a lower average waiting time of 8.2μs at the ingress node. Since the IP packets in the virtual queues are buffered for a longer time, the size of the queues reach the *STh* quicker than the state when the time threshold is 25 times a packet size so the average delay for burst formation is also decreased as seen in the figure.

In Table I, generalized traffic results from random ingress nodes in the OBS domain under different network loads by the employment of the two burst assembly techniques is given.

Table 1. Output Hurst parameters at the ingress nodes for different burst assembly schemes

| | H. at *TTh* = 10μs<br>Input H. = 0.9 | | H. at *TTh* = 16μs<br>Input H. = 0.9 | |
|---|---|---|---|---|
| Assembly Scheme | Average Load (7 E) | High Load (14 E) | Average Load (7 E) | High Load (14 E) |
| Time Threshold Based | 0.5245 | 0.5225 | 0.5105 | 0.5020 |
| Hybrid | 0.5110 | 0.5118 | 0.5276 | 0.5335 |

### 6.2.2 Performance of OBS Scheduling Techniques

In this section, we focus on the performance of OBS scheduling techniques by measuring the overall loss rate when each of the five techniques employed inside the OBS domain under different *TTh* values and burstification schemes. The five techniques are the ones we test in 6.1.

### 6.2.2.1 Under Time Threshold Burst Assemby

Initially, we set the *TTh* to 25 times an IP packet duration (10μs) and run the Time Threshold Based Burst Assembly algorithm. At first, we employ the horizon algorithm inside the OBS domain and observe the relation between the packet loss rate and burst loss rate. In Figure 6.13, it seems that the burst loss rate is about 2 times the packet loss rate since longer bursts reserve more bandwidth on the wavelength channel so the number of dropped bursts that are of the short burst size is higher than the number of longer bursts as we know from the results in 6.2.1.

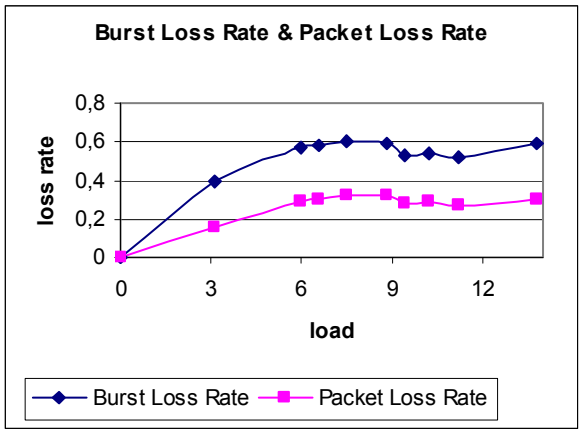



Figure 6.13. Burst Loss and Packet Loss Comparison. *TTh* = 10μs.
Scheduling scheme: The horizon algorithm. Burst Assembly Scheme = *TTh* Based

Upon running the horizon technique inside the OBS domain, we run First-Fit VF, LAUC-VF, Group Scheduling and DFMOC-VF techniques in order. The first metric we measure is the loss rate. In Figure 6.14, the overall loss rate for each of the technique can be seen. It is observed that, the horizon technique leads to the highest loss rate. Since the void filling techniques utilize the voids, it seems that they lead to lower loss rates in comparison to the horizon technique. Besides these, since First-Fit VF does not attempt to select the void with the latest starting time, it leads to a slightly higher loss rate in comparison to LAUC-VF. The lowest loss rate is obtained by segmentation based and Group Scheduling techniques. The reason of segmentation technique leading to lower loss rate is that DFMOC-VF attempts to schedule the non-contending part of a contending burst in order to drop the burst completely. As it is seen from Figure 6.14, Group Scheduling leads to the lowest loss rate since instead of scheduling the incoming bursts immediately, it collects the BHCs for a pre-determined collecting period and attempts to minimize the number of contending bursts in that time period. The relative performance of the scheduling techniques seem to be parallel to the results obtained from the Poisson traffic.

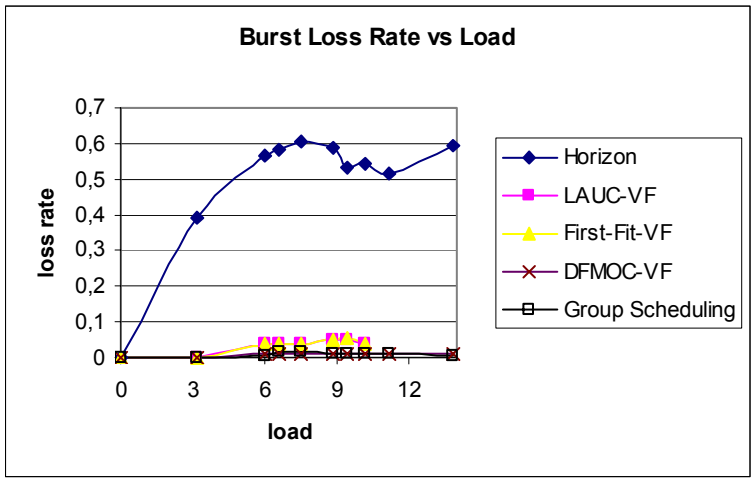



Figure 6.14. Burst Loss Rate comparison for the horizon, First-Fit VF, LAUC-VF, Group Scheduling, and DFMOC-VF. *TTh* = 10μs. Burst Assembly Scheme = *TTh* Based

We again run the horizon algorithm inside the OBS domain by setting the *TTh* value to 16μs which corresponds to 40 times of an IP packet duration. As we see from Figure 6.15, burst loss rate is significantly decreased with a rate of 1/3 and packet loss rate is decreased with a rate of ¼. Since the waiting time for generating the bursts is increased, the interarrival time between the bursts increase. This gives the chance to the bursts to be scheduled in the wavelength channel with a lower contention probability. Besides this, since the *TTh* value increases the burstification delay, it also increases the average burst size. As we mention in the previous sections, longer bursts reserve more bandwidth and a high amount of bursts that increase the loss rate consist of short bursts that cannot be scheduled since a long time interval is reserved by other long bursts.

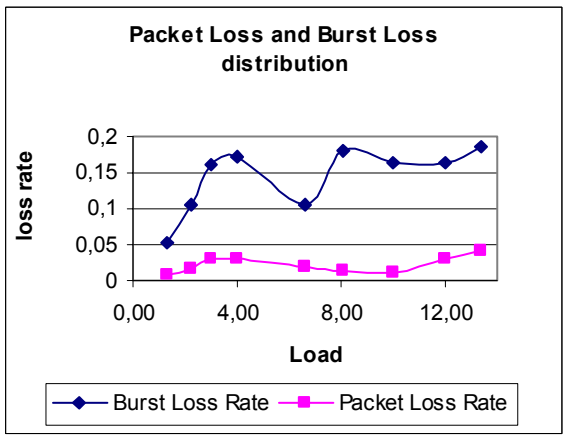



Figure 6.15. Burst Loss Rate and Packet Loss Rate Comparison. *TTh* = 16μs. Scheduling scheme: The horizon algorithm. Burst Assembly Scheme = *TTh* Based

In Figure 6.16, a comparison of the performance of the 5 OBS scheduling techniques is given when *TTh* value is 40 times an IP packet duration at the egress of the

network. All of the techniques lead to a significant lower rate in comparison to the previous results. Notice that, the loss rates for both of the *TTh* values with this burstification scheme is lower than the results obtained from the Poisson distributed incoming burst traffic since the queueing model at the ingress nodes shapes the traffic inside the OBS domain. Actually, the increase in the *TTh* also causes loss rate to decrease when Time Threshold Based Burst Assembly runs at the ingress nodes.

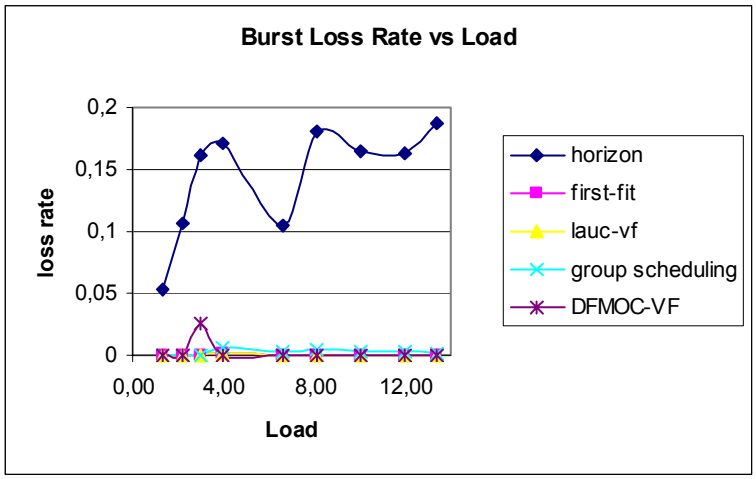



Figure 6.16. Burst Loss Rate Comparison for the horizon, First-Fit VF, LAUC-VF, Group Scheduling, and DFMOC-VF. *TTh* = 16μs. Burst Assembly Scheme = *TTh* Based

In the next section, we evaluate the performance of the same techniques with Hybrid Burst Assembly, for different *TTh* values, and compare them with the results obtained here.

#### 6.2.2.2 Under Hybrid Burst Assemby

The first part of this subsection consist of the results taken with *TTh* value that is equal to 25 times of an IP packet duration (10μs). In Figure 6.17, we compare the burst loss rate with packet loss rate after the employment of the horizon algorithm. It seems that the burst loss rate is about 1.5 times the packet loss rate. The reason is that the longer bursts reserve more bandwidth on the wavelength channel so the number of dropped bursts that are of the short burst size are higher than the number of longer bursts as we know from the results we obtained when we employed Time Threshold Based Burst Assembly algorithm at the ingress nodes. This implies that, packet loss rate is greater than the one obtained when *TTh* based burst assembly algorithm employed at the ingress nodes with the same parameter. The reason is that, in Hybrid burst assembly scheme, the bursts are forced not to exceed a pre-determined size

threshold so the average burst size is shorter and shorter bursts to be generated are allowed which decreases the survival possibility in case of a contention.

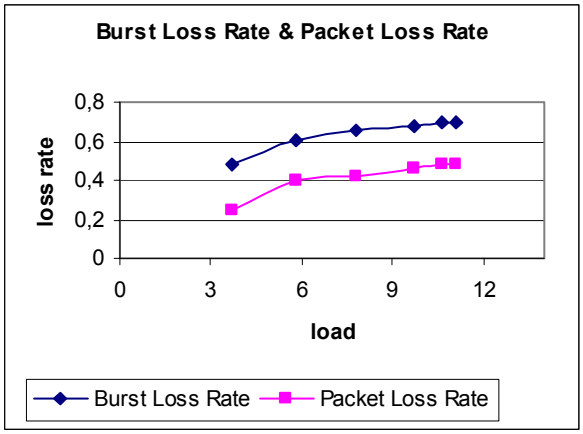



Figure 6.17. Burst Loss and Packet Loss Comparison. *TTh* = 10µs and *STh* = 100µs. Scheduling scheme: The horizon algorithm. Burst Assembly Scheme = Hybrid.

The comparison of the performance of the OBS scheduling techniques when Hybrid burst assembly scheme is employed at the ingress nodes with these parameters are given in Figure 6.18. The relative loss rates of the techniques remain the same as they are in the previous experiments. The effect of not forcing the bursts to have a size under a threshold value on the performance in terms of loss rate by the *TTh* based burst assembly is also seen here as we discuss for the results in Figure 6.17. The burst loss rates of the techniques lead to about 15% increase when Hybrid burst assembly is employed instead of *TTh* based burst assembly with *TTh* = 10µs.

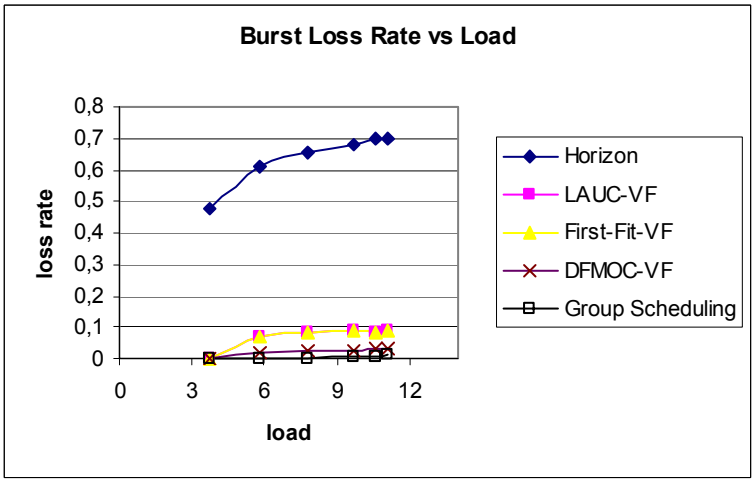



Figure 6.18. Burst Loss Rate Comparison for the horizon, First-Fit VF, LAUC-VF, Group Scheduling, and DFMOC-VF. *TTh* = 10µs. Burst Assembly Scheme = Hybrid

Similar to the experiments in 6.2.2.1, we again set the *TTh* to 16µs and run the same simulations. In the first experiment seen in Figure 6.19, the relation between burst loss rate and packet loss rate is observed by running the horizon technique inside the

network. As it is seen from the figure, burst loss rate is seen to be almost 1.5 times the packet loss rate. The reason is the same again here, as explained for the other experiments referring to the relation between burst loss rate and packet loss rate. We also see that there is a decrease in the loss for packets and bursts with a rate of 20% as the load gets higher when compared to the results obtained with *TTh* = 10μs. The reason of the reduction in the loss rate is because of having longer bursts and obtaining longer interarrival times between the generated bursts.

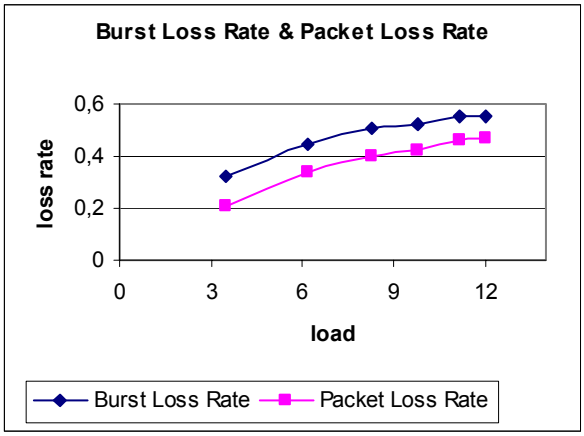



Figure 6.19. Burst Loss and Packet Loss Comparison. *TTh* = 16μs and *STh* = 100μs. Scheduling scheme: The horizon algorithm. Burst Assembly Scheme = Hybrid.

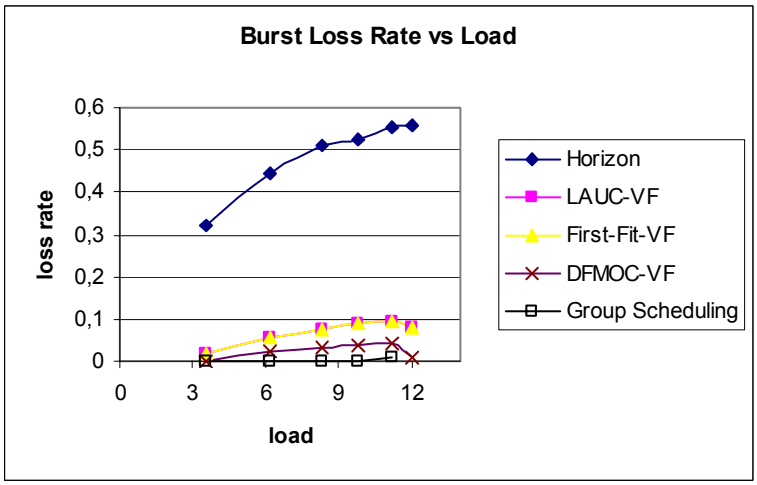



Figure 6.20. Burst Loss Rate Comparison for the horizon, First-Fit VF, LAUC-VF, Group Scheduling, and DFMOC-VF. *TTh* = 16μs. Burst Assembly Scheme = Hybrid

We also, compare the relative loss rates of the five OBS scheduling techniques. In Figure 6.20 it is observed that, the increase in the time threshold does not cause a significant change for the techniques other than the horizon. Besides these, if we compare the loss rates of these techniques with the ones employed after *TTh* Based Burst Assembly technique with these parameters, it can be said that the loss rates

here are higher because of the difference of the generated bursts in shape in these techniques as previously explained.

As we see in this section, since the traffic shaping effect of the burstification schemes, the burst loss rate is lower than it is in Poisson traffic. Moreover, Time Threshold Based Burst Assembly technique is more prone to the change in *TTh* value, since in Hybrid Burst Assembly the burst size is limited from below and above, the the burst size and the interarrival time cannot reach to larger values as *TTh* increases.

### 6.2.3 Traffic Characteristics at the Egress Nodes

At the egress nodes, we also measure the traffic characteristics for the two burst assembly algorithms with two different *TTh* values. In Section 6.2.1, we show that the burst assembly algorithms reduce the Hurst parameter at the ingress nodes and lead to different self-similarity characteristics under two different *TTh* values. Here, again we employ the horizon, First-Fit VF, LAUC-VF, Group Scheduling, and DFMOC-VF inside the OBS domain. At the output, we analyze the traffic characteristics by measuring the Hurst parameter for each state that one of those five techniques is employed.

#### 6.2.3.1 Under Time Threshold Based Burst Assembly

Next experiments at the edge nodes include the traffic characteristics at the output upon the employment of each OBS scheduling technique. In Figure B.1, the traffic characteristics at a randomly selected egress node can be seen when the horizon technique is employed inside the network with a traffic load of 13 E. As it can be observed, the output Hurst parameter is 0.54. This shows that, the incoming packet traffic with Hurst parameter 0.9 is reduced to a degree of 0.51 by the *TTh* Based Burst Assembly, and inside the OBS domain the bursts are switched by the horizon technique so the burstiness is increased at the output. However, although the traffic is still self-similar at the output, its long term behaviour changes significantly by the reduction in the Hurst parameter from 0.9 to its value at the output.

The traffic characteristics at the output after the employment of First-Fit VF are given in Figure B.2. As known, First-Fit VF utilizes the voids between the scheduled bursts. This property leads to increase the Hurst parameter ($H$ = 0.5522) at the output

slightly more than the horizon technique. Beyond these, the short term characteristics of the traffic is changed and at the output First-Fit VF does not break the significant reduction in the self-similarity that is obtained by the burst assembly technique.

In Figure B.3, the results from the same experiment by running LAUC-VF technique inside the network is given. As it is seen from the graphics, LAUC-VF caused a slightly more bursty traffic in comparison to the traffic obtained by the horizon. Here the Hurst parameter is so close to First-Fit VF but as a matter of higher utilization, the Hurst parameter is increased slightly to 0.5610. However, the traffic still keeps its self-similar nature but its burstiness is significantly lower than it is at the ingress nodes. Based on these three graphics (Figures B.1-3), it seems that in the OBS network, as the idle intervals in the wavelength channels of the optical links utilized more, the output Hurst parameter tends to increase. The results from the other techniques also supports this thesis.

Employing Group Scheduling technique with a high network (13 E) load inside the OBS domain increases the burstiness of the output traffic at a randomly selected egress node as shown in Figure B.4. The Hurst parameter is increased to 0.6111 which shows that this technique leads to a more self-similar characteristics at the output in comparison to the other techniques. This is related to the policy of not scheduling the bursts immediately in order to utilize the intervals in the wavelength channels as much as it can. However, when compared to the input Hurst parameter, it is obvious that, even a groupping based OBS technique does not cause the traffic to approach to its original characteristics whose self-similarity is reduced upon burst assembly process at the ingress nodes.

In Figure B.5, the traffic characteristics at a random node after the employment of DFMOC-VF under high network load is given. As it is seen from the figure, the self-similarity of the traffic at the ingress nodes just after the burst assembly is increased more than the other VF techniques and the horizon techniques. However, employing DFMOC-VF leads to a lower self-similar character in comparison to Group Scheduling. Segmentation brings out a dense utilization and burstiness. However, in long term, the concept of the immediate scheduling works here, and that densely utilized and bursty characteristics of the wavelength channels do not seem to be as much as the characteristics in partitioning the wavelength channels and intervals in time windows.

In addition, we see that none of these OBS scheduling techniques change the self-similar nature of the traffic. Moreover, they do not cause the traffic to converge to its original burstiness. We also see that, the techniques which are good at utilizing the idle intervals in the wavelength channels increase the self-similarity up to 0.61 from 0.52 whose original value was 0.9 before the burst assembly algorithm.

Upon these measurements, we set the *TTh* value to 16μs which is 40 times an IP packet duration and run the Time Threshold Based Burst Assembly algorithm at the ingress nodes of the network by employing the defined 5 OBS scheduling techniques inside the OBS domain and measure the same results at the egress nodes of the network.

In order to estimate the output Hurst parameter at the egress nodes of the OBS network we collect the data which is plotted in Figures B.6-10 with overall network load 13 E. In Figure B.6, the output traffic characteristics is shown when the horizon technique is employed inside the OBS domain. The output Hurst parameter is 0.4987 which still seems to have a long range dependence but approaching to a Poisson-like characteristic since the buffering time at the ingress nodes increases. Apart from the previous experiments with *TTh* = 10μs, the smooth traffic shape obtained at the ingress nodes is conserved here so there exists no increase in the self-similarity of the incoming traffic.

In Figure B.7, the results obtained upon the employment of First-Fit VF are given. As we can observe, the shape of the traffic is so similar to the one in Figure B.6. The measured Hurst parameter also supports this thesis since it is 0.4933. Here, the change of the OBS technique inside the network does not affect the outgoing traffic characteristics. This is because of the increase in the burstification time which causes a less bursty traffic since larger interarrival intervals occur in the wavelength channels. In the following Figures B.8-10 the results obtained from the other techniques also support this opinion.

As we see in Figure B.8, LAUC-VF technique also results with similar traffic characteristics with the previous scheduling techniques. The input Hurst parameter can be said to be kept here too. Another point is that, the Hurst parameters that are obtained with this greater *TTh* are also less then the output Hurst parameters at the egress nodes when *TTh* is 10μs. This result fits with what is concluded in [20].

Employment of Group Scheduling seems to cause a significant increase in the self-similarity of the traffic as we know from the results with *TTh* = 10μs. In Figure B.9, the output Hurst parameter seems as 0.5650. However, since the interarrival time between the bursts increases by the increase in the burstification delay, the Hurst parameter obtained here is less than the one obtained when *TTh* = 10μs so although the time window based approach of Group Scheduling increases the self-similarity, as a result of selecting longer time threshold, we obtain a traffic characteristics at the egress nodes which is less self-similar.

The results upon the employment of DFMOC-VF are shown in Figure B.10, and they are similar to the results plotted in Figures B.6-8. Here, the output Hurst parameter is measured as 0.5087. This implies that, the self-similarity of the traffic at an egress node of the OBS network is not affected by the change of the scheduling algorithm inside the OBS domain as long as the algorithms schedule the bursts immediately. The reason is that, the channel utilization policy of these techniques cannot change the characteristics of the traffic since the total processing time of the BHCs and the arrival scheme of the DBs of these algorithms are too long in comparison to the buffering time at the ingress nodes.

#### 6.2.3.2 Hybrid Burst Assembly

In this section, we repeat the same experiments we perform in Section 6.2.2.1 by employing Hybrid Burst Assembly scheme here. We set the *TTh* value to be 25 times an IP packet duration which corresponds to 10μs. In order to limit the bursts by an upper limit, the *STh* value is selected to be 100μs. The bursts are neither allowed to have a size greater than *STh* nor to be generated within a time greater than *TTh*.

We also plot and measure the outgoing traffic at randomly selected egress nodes by the employment of these 5 OBS scheduling techniques. The first experiment results are shown in Figure B.11, and are obtained from the outgoing bursts that are scheduled by the horizon algorithm. The output Hurst parameter is measured as 0.5327. This implies that the traffic shape at the egress nodes is almost the same as at the ingress nodes since the Hurst parameter at the input and output is almost equal.

In Figure B.12, the traffic characteristics observed at an egress node when First-Fit VF is employed. The output Hurst parameter is measured as 0.5862. Similar to this one, LAUC-VF leads to a Hurst parameter of 0.5817 under the same conditions as

seen in Figure B.13. The increase in the output Hurst parameter by the employment of these techniques comes from utilizing the idle intervals on the channels more than the horizon in short and long timescale. These parameters are so close which are obtained in *TTh* Based Burst Assembly with *TTh* = 10μs. Based on these, it can be said that, the traffic shape at the egress nodes is almost the same in both of the burst assembly schemes when the maximum burstification delay is 25 times an IP packet duration.

In Figure B.14, it seems that Group Scheduling leads to a Hurst parameter of 0.7102 which shows that self-similarity at the ingress nodes of the OBS network is increased by grouping the bursts and scheduling them by the utilization of time windows. This time, more self-similar traffic is obtained at the output of the OBS domain in comparison to the employment of the same scheduling technique after *TTh* Based Burst Assembly with the same parameters. This is because of the effect of groupping the bursts plus the difference in the shape of the bursts generated by the two techniques as explained in Section 6.2.1.1 and Section 6.2.1.2.

Moreover, the employment of DFMOC-VF, leads to an output Hurst parameter of 0.6012 at an egress node of the network. In Figure B.15, we plot the traffic characteristics. Segmented bursts fill the time intervals which causes a more bursty traffic in comparison to the horizon, LAUC-VF, and First-Fit VF techniques. Besides these, DFMOC-VF after Hybrid Burst Assembly scheme also leads to a more self-similar traffic in comparison to the traffic shape obtained when this technique is employes after *TTh* Based Burst Assembly because of the same reason as we have just explained for the results on Group Scheduling.

In order to observe the effect of the *TTh* size on the traffic characteristics of the OBS network when Hybrid Burst Assembly technique is employed, we set the *TTh* value to 16μs which is equal to 40 times an IP packet duration on 10Gbps line rate.

The last experiments are on the characteristics of the outgoing traffic. The first technique employed inside the OBS domain is the horizon algorithm. As seen from Figure B.16, increasing the burst formation delay does not affect the self-similarity of the traffic upon employing the horizon algorithm since the output Hurst parameter is measured as 0.5314 at an egress node. We explain the reason of this results upon getting the results by the employment of DFMOC-VF within this section.

We measure the output Hurst parameter as 0.56417 when we employ First-Fit VF after the Hybrid Burst Assembly with *TTh* = 16μs. In Figure B.17, we plot our measurements. The results obtained by the employment of LAUC-VF are so similar to the results we have upon running the First-Fit VF inside the network, and they are plotted in Figure B.18. The Hurst parameter is measured as 0.56631.

In comparison to the results obtained by employing *TTh* Based Burst Assembly at the ingress nodes of the network, the self-similarity is higher here. However, we observe that the degree of self-similarity of the outgoing traffic is not affected by the change in *TTh* value as we explain the reason in this section.

Upon employing the Group Scheduling algorithm inside the network we obtain a very slight increase in the outgoing Hurst parameter at an egress node by increasing the *TTh* value. As seen from Figure B.19, the output Hurst parameter is measured as 0.7341. This implies that, the outgoing traffic shape is not affected significantly by the change in the time threshold in burstification when Group Scheduling runs inside the network too.

DFMOC-VF is the technique by the employment of which the outgoing traffic shape is affected most as the time threshold in burstification changes. As it is seen from Figure B.20, the output Hurst parameter is measured as 0.68427. The increase in the *TTh* value also increases the self-similarity. The reason in increasing the Hurst paremeter is that DFMOC-VF switches the bursts by segmenting them and increasing the burstiness of the outgoing of the traffic.

In Table II, we give a generalized traffic results scheme from random egress nodes in the OBS domains for 5 different OBS scheduling techniques together with the two burst assembly schemes.

Table 2. Output Hurst parameters at the egress nodes for different burst assembly schemes and scheduling techniques.

| | Time Threshold Based Burst Assembly | | Hybrid Burst Assembly | |
|---|---|---|---|---|
| OBS Scheduling Technique | H. at *TTh* = 10μs<br>H.after assembly = 0.5225 | H. at *TTh* = 16μs<br>H.after assembly = 0.5020 | H. at *TTh* = 10μs<br>H.after assembly = 0.5118 | H. at *TTh* = 16μs<br>H.after assembly = 0.5335 |
| Horizon | 0.5405 | 0.4987 | 0.5327 | 0.5314 |
| First-Fit VF | 0.5510 | 0.4933 | 0.5817 | 0.5642 |
| LAUC-VF | 0.5600 | 0.5001 | 0.5862 | 0.5663 |
| Group Scheduling | 0.6110 | 0.5635 | 0.7102 | 0.7341 |
| DFMOC-VF | 0.5665 | 0.5087 | 0.6012 | 0.6843 |

# 7. CONCLUSION AND FUTURE WORK:

In this thesis, we study the performance of optical burst switching (OBS) techniques, and the traffic characteristics of the network traffic. OBS which is a novel switching paradigm in optical WDM networks uses the advantages of optical packet switching and optical circuit switching. Burst generation is an important issue in OBS. We examine currently proposed burst assembly schemes and make a categorization of six groups: Time Threshold Based, Size Threshold Based, Hybrid, Adaptive Period Based, Differentiated, and Round Robin burst assembly schemes.

We put the OBS switching and scheduling techniques into 4 main groups: Non-Void Filling, Void Filling, Group Scheduling, and Segmentation Based Scheduling techniques. In order to analyze the performance of the proposed techniques we generate a Poisson distributed burst traffic at the edges of the NSFNET topology and implement five selected OBS scheduling techniques in the network. In terms of loss rate, we conclude that Group Scheduling and segmentation based techniques perform better as the network load increases, while the Non-Void Filling techniques perform the worst. Void Filling techniques do not perform as good as the Group Scheduling and segmentation based techniques but lead to a significantly higher performance in terms of burst loss rate when compared to the Non-Void Fillling techniques. When we compare the scheduling delay of these techniques we obtain the best performance in Non-Void Filling techniques since the complexity of the scheduling algorihms are just bounded by the number of wavelength channels. The worst performance in scheduling delay occurs in the Group Scheduling and segmentation based techniques since the complexities of the algorithms are higher due to attempting to increase the utilization as much as possible. Void Filling techniques lead to lower scheduling delay but significantly higher delay than the Non-Void Filling techniques. As a result, it can be said that there is a trade-off between scheduling time and burst loss rate in OBS techniques.

We also generate a self-similar IP packet traffic with Hurst parameter 0.9 at the ingress nodes of the OBS network. We run Time Threshold Based Burst Assembly, and Hybrid Burst Assembly schemes for two different *TTh* values. For each technique, the Hurst parameter at the ingress nodes upon running the assembly algorithms is measured in (0.50-0.52). However, we observe that Time Threshold Based Burst Assembly reduces the self-similarity more since it does not limit the size of the bursts and allows to have a fixed and longer interarrival time. We also measure the average burst formation delay, at the ingress nodes of the OBS network and see that Hybrid technique significantly reduces the delay for burst generation especially for high time threshold values if appropriate upper bound is selected for the burst size.

Time Threshold Based Burst Assembly is more prone to reduce the self-similarity due to an increase in *TTh* value such that when we increase the *TTh*, we obtain a less self-simlar traffic upon the employment of Time Threshold Based Burst Assembly scheme ($H \in (0.50,0.52)$ ), while Hybrid Burst Assembly scheme leads to a higher self-similar incoming traffic such that $H$ varying between 0.52 and 0.54.

We also analyze the performance of five different OBS scheduling algorithms in the OBS domain. These scheduling algorithms are the horizon, First-Fit VF, LAUC-VF, Group Scheduling, and DFMOC-VF. We observe that, under Time Threshold Based Burst Assembly, each of these techniques lead to a lower burst loss rate as the *TTh* gets higher. Beyond this we observe that, all of the techniques except DFMOC-VF and Group Scheduling lead to lower burst loss rate as the *TTh* gets higher under Hybrid Burst Assembly scheme. However, Time Threshold Based Burst Assembly leads to more reduction in the loss rate. Based on the analysis at the ingress nodes, we observe that, the Time Threshold Based Burst Assembly leads to longer bursts and most of that exceed the *STh* value of 100μs. This property allows to have larger interarrival time between the bursts and longer bursts to be created which have more chance to survive in case of a contention since they reserve more bandwidth. We conclude that, the Hurst parameter at the output depends on both the burst assembly technique and the OBS scheduling technique inside the OBS domain. The scheduling techniques which attempt to increase the channel utilization lead to a higher self-similarity at the output. Another result is that, if no upper bound for burst sizes is specified when generating the bursts, the increase in time threshold value decreases

the self-similarity of the outgoing traffic at the egress nodes for each of OBS switching technique employed inside the OBS domain.

As a future work, we are going to concentrate on new burst assembly schemes, especially adaptive burst assembly algorithms. Moreover, these, we are planning to run our simulations under various QoS classes. We will add alternate routes to the routing tables of the nodes in order to attempt to reschedule the burst on an alternative path, and prevent contention.

## 8. REFERENCES:


[1] **M. Mishra**, **E. L. Johnson, K. M. Sivalingam**, "Scheduling in Optical WDM Networks using Hidden Markov Cahin-based Traffic Predictors", *in 2000 Proc. IEEE International Conference on Networks*, pp.380-384, September 2000.

[2] **A. Kaheel**, **T. Khattab**, **H. Alnuweiri**, "Quality of Service Mechanisms in IP-over-WDM Networks"*, IEEE Communications Magazine*, pp.38-43, December 2002.

[3] **J. Kennington**, **E. Olinick**, "Wavelength Translation in WDM Networks: Optimization Models and Solution Procedures"*, INFORMS Journal on Computing*, vol. 16, issue 2, pp.174-187, Spring 2004.

[4] **J.S.Turner**, "Terabit Switching", *Journal of High Speed Networks*, vol 8., no. 1, pp.3-16, March 1999.

[5] **J. Ramamirtham, J. Turner** , "Time Sliced Optical Burst Switching", *in 2003 Proc. IEEE INFOCOM. Twenty-second Annual Joint conference of the IEEE Computer and Communications Societies*, vol. 3, pp.2030-2038, March 2003.

[6] **R. Ramaswami R, K. Sivarajan**, "Optical Networks"*,* Morgan Kaufmann, 2002.

[7] **R. Srinivasan, A. K. Somani**, "A generalized framework for analyzing time-space switched optical networks", *in 2001 Proc. IEEE INFOCOM,* pp. 179-188, April 2001.

[8] **D. K. Hunter, D. G. Smith**, "NewArchitectures for optical TDM switching", IEEE/OSA Journal of Lightwave Technology, vol 11, no-3, pp.495-511, March '93.

[9] **H. F. Jordan, D. Lee, S. V. Ramanan**, "Serial array time slot interchangers and optical implementation", *IEEE Transactions on Computers*, vol. 43, no 11, pp.1309-1318, November 1994.

[10] **H. J. Chao, S. Y. Liew**, "A new optical cell switching paradigm", *in 2003 Proc. International Workshop on Optical Burst Switching,* October 2003. Available at http://www.cse.buffalo.edu/~qiao/ wobs/wobs2003/files/WOBS106OCS.pdf

[11] **J-B Chang, C-S Park,** "Efficient channel scheduling algorithm in optical burst switching architecture", *in 2002 Workshop on High Performance Switching and Routing,* pp. 194-198, May 2002.

[12] **T. Battestilli, H. Perros**, "An introduction to Optical Burst Switching", *IEEE Optical Communications*, August 2003, pp.10-15.

[13] **J. Xu J, C. Qiao, J. Li, G. Xu**, "Efficient Channel Scheduling Algorithms in Optical Burst Switched Networks"*, in 2003 Proc. IEEE INFOCOM. Twenty-second Annual Joint conference of the IEEE Computer and Communications Societies*, vol. 3, pp.2268-2278, March 2003.

[14] **M. Iizuka, M. Sakuta, Y. Nishino, I. Sasase**, "A Scheduling Algorithm Minimizing Voids Generated by Arriving Bursts in Optical Burst Switched

WDM Network", *in 2002 Proc. IEEE Global Telecommunicaitons Conference,* vol. 3, pp.2736-2740, November 2002.

[15] **V. M. Vokkarane, G. P. V. Thodime, V. U. B. Challagulla, J. P. Jue**, "Channel Scheduling Algorithms using Burst Segmentation and FDLs for Optical Burst-Switched Networks", *in 2003 Proc .IEEE International conference on Communications*, vol. 2, pp.1443-1447*,* May 2003.

[16] **S. Charcranoon, T. S. El-Bawab, H. Cankaya, J. Shin**, "Group Scheduling for Optical Burst Switched (OBS) Networks"*, in 2003 IEEE Global Telecommunications Conference,* vol 5, pp. 2745-2749, December 2003.

[17] **A. M. Law, W. D. Kelton**, 2000, Simulation Modelling And Analysis, Mc Graw Hill, 389-393

[18] **M. Yoo, C. Qiao,** "Just-Enough-Time (JET): a high speed protocol for bursty traffic in optical networks"*, in 1997 Digest of the IEEE/LEOS Summer Topicql Meetings*, pp. 26-27, Aug 1997.

[19] **M. D. Cano, J. Malgosa-Sanahuja, F. Cerdan**, **J. Garcia-Haro, "**Internet measurements and data study over the regional", *in 2001 Proc. IEEE Pacific Rim conference on Communications, Computers and Signal Processing,* Vol. 2, pp.393 – 396, August 2001.

[20] **A. Ge, F. Callegati, L. S. Tamil**, "On Optical Burst Switching and Self-similar Traffic"*, IEEE Communications Letters*, Vol 4, No. 3, pp.98-100, March2000.

[21] **Z. Sahinoglu, S. Tekinay**, "On multimedia networks:Self-similar traffic and Network Performance"*, IEEE Communications Magazine*, Vol 37, pp. 48-42, Jan. 1999.

[22] **V. M. Vokkarane, J. P. Jue**, "Prioritized Burst Segmentation and Composite Burst-Assembly Techniques for QoS Support in Optical Burst-Switched Networks"*, IEEE Journal on Selected Areas in Communications*, Vol 21, No 7, September2003, pp.1198-1209.

[23] **A. Tanenbaum,** 1996. Computer Networks, Prentice-Hall Inc, pp.348-352.

[24] **G.Hu**, **K. Dolzer, C.Gauger,** "Does burst assembly really reduce the self-similarity?", *in Optical Fiber Communications Conference, OFC2003,* vol.86 of OSA Trends in Optics and Photonics Series, Washington, D. C, 2003, pp.124-126.

[25] **J**. **Luo, Q. Zeng, H. Chi, Z. Zhang, H. Zhao,** "The impacts of burst assembly on the traffic properties in optical burst switching networks", *in 2003 IEEE Proceedings of International Conference on Communication Technology,* vol.1, pp.521-524, April 2003.

[26] **Fractal Objects and Self-Similar Processes***, in Tutorial PhysioNet research resource for complex physiologic signals*, Available: http://www.physionet.org/tutorials.

[27] **C. Qiao, Y. Chen**, "The potentials of optical burst switching (OBS)", *in 2003 IEEE Optical Fiber Communications Conference,* vol 1, pp.219-220, March 2003.

[28] **J. Li, C. Qiao,** "Proactive contention avoidance in OBS networks", *in 2003 IEEE Optical Fiber Communications Conference,* vol 1, pp.129.130, March 2003.

[29] **A. Banerjee, N. Singhal, J. Zhang, D. Ghosal, C. Chen-Nee, B. Mukherjee,** "A time-path scheduling problem (TPSP) for aggregating large data files from distributed databases using an optical burst-switched network", *in 2004 IEEE International Conference on Communications,* vol. 3, pp.1569-1573, June 2004.

[30] **Y. Chen, C. Qiao, X. Yu,** "Optical burst switching: A new area in optical networking research", *in 2004 IEEE-Network,* vol. 18, pp.16-23, May-June, 2004.

[31] **S. Verma, H. Chaskar, R. Ravikanth,** "Optical burst switching: A viable solution for terabit IP backbone", *IEEE Network Magazine,* pp.48-53, December 2000.

[32] **Z. Rosberg, H. L. Vu,** "Performance analyses of optical burst-switched networks", *IEEE Journal on Selected Areas in Communications,* vol. 21, pp. 1187-1197, September 2003.

[33] **C. Qiao,** "Labeled optical burst switching for IP-over-WDM integration", *IEEE Communications Magazine,* vol. 38, pp. 104-114, September 2000.

[34] **S. K. Lee, D. Girffith, V. Coussot, D. Su,** "Explicit Routing with QoS constraints in IP over WDM", *in 2002 Proc. IEE Communications,* vol 149, pp. 83-91, April 2002.

[35] **J. Lawrence,** "Designing multiprotocol label switching networks", *IEEE Communications Magazine,* vol. 39, pp. 134-142, July 2001.

[36] **J. Y. Wei, R. I. McFarlan,** "Just-In-Time signaling for WDM Optical Burst Switching Networks", *Journal of Lighwave Technology,* vol. 18, no.12, pp. 2019-2037, December 2000.

[37] **C. Qiao, M. Yoo,** "Optical Burst Switching (OBS)-a new paradigm for an Optical Internet", *Journal of HighSpeed Networks,* vol. 8, pp. 69-84, August 1999.

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

[39] **T. Back, S. Khuri,** "An evolutionary heuristic for the maximum independent set problem", *in 1994 Proc. IEEE World Congress on Computational Intelligence,* vol. 2, pp.531-535, June 1994.

[40] **V. M. Vokkarane, J. P. Jue,** "Burst segmentation : an approach for reducing packet loss in optical bust switched networks", *in 2002 Proc. IEEE International Conference on Communications,* vol. 5, pp. 2673-2677, April-May 2002.

[41] **V. M. Vokkarane, Z. Qiong, J. P. Jue, C. Biao,** "Generalized burst assembly and scheduling techniques for QoS support in optical burst-switched networks", *in 2002 Proc. IEEE Global TelecommunicationsConference*, vol. 3, pp. 2747-2751, Nov 2002.

[42] **C-F. Hsu, T-L Liu, N-F Huang,** "Performance analysis of deflection routing in optical burst-switched networks", *in 2002 Proc. Twenty-first Annual Joint Conference of the IEEE Computer and Communications Societies,* vol. 1, pp. 66-73, June 2002.

[43] **M. Yang, S. Q. Zheng, D. Verchere,** "A QoS supporting scheduling algorithm for optical burst switching DWDM networks", *in 2001 Proc. IEEE Global Telecommunications Conference,* vol. 1, pp.25-29, Nov 2001.

[44] **M. Welzl, M. Muhlhauser,** "Scalability and quality of service: a trade-off?", *in IEEE Communications Magazine,* vol. 41, issue. 6, pp.32-36, June 2003.

[45] **J. A. White, R. S. Tucker, and K. Long** , "Merit-Based Scheduling Algorithm for Optical Burst Switching"**,** *in 2002 Proc. COIN- International Conference on Optical Internet*, vol.1, pp.75-77, July 2002.

[46] **C. Kan, H. Balt, S. Michel, D. Verchere,** "Information model of an optical burst edge switch", *in 2002 Proc. IEEE International Conference on Communications,* vol. 5, pp. 2717-2721, April-May 2002.

[47] **S. Oh, M. Kang,** "A burst assembly algorithm in optical burst switching networks", *inn 2002 Optical Fiber Communications Conference and Exhibit,* pp. 771-773, March 2002.

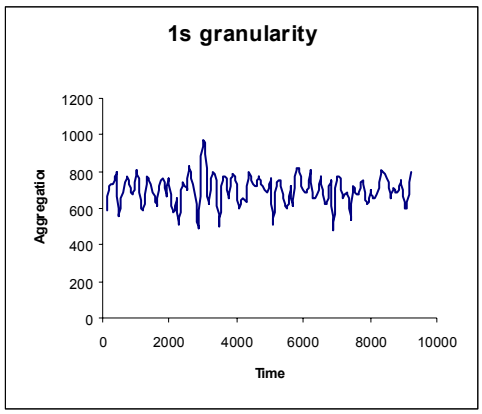
[48] **J. Luo, Q. Zeng, H. Chi, Z. Zhang, H. Zhao,** "The impacts of burst assembly on the traffic properties in optical burst switching networks", *in 2003 Proc. International Conference on Communication Technology,* vol. 1, pp. 521-524, April 2003.

[49] **X. Cao, J. Li, Y. Chen, C. Qiao,** "Assembling TCP/IP packets in optical burst switched networks", *in 2002 Proc.IEEE Global Telecommunicatons Conference,* vol. 3, pp.2808-2812, November 2002.

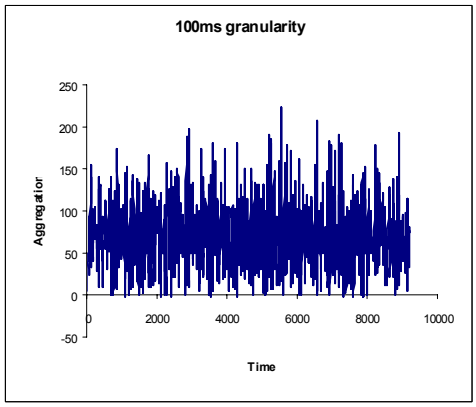
[50] **Y. Arakawa, N. Yamanaka, I. Sasase,** "Performance of Optical Burst Switched WDM Ring network with TTFR System", *in 2004 Proc. IFIP Optical Networks and Technologies Conference,* vol. 1, pp. 95-102, October 2004.

[51] **T. Tachibana, T. Ajima, S. Kasahara,** "Round-robin burst assembly and constant transmission scheduling for optical burst switching networks", *in 2003 Proc. IEEE Global Telecommunications Conference,* vol. 5, pp.2772, 2776, December 2003.

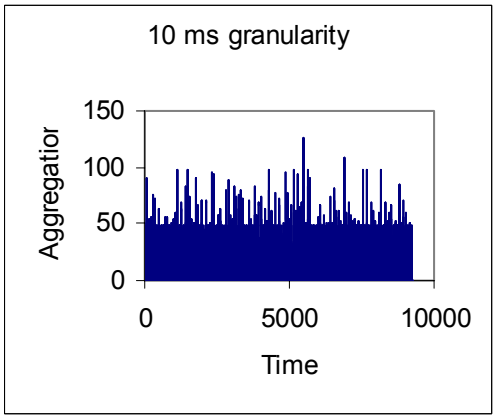
## APPENDIX A




(a)1s granularity

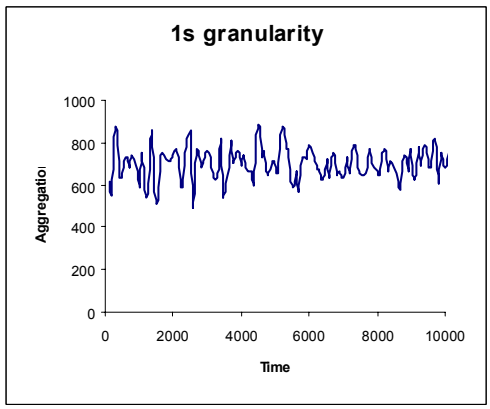



(b) 100ms granularity

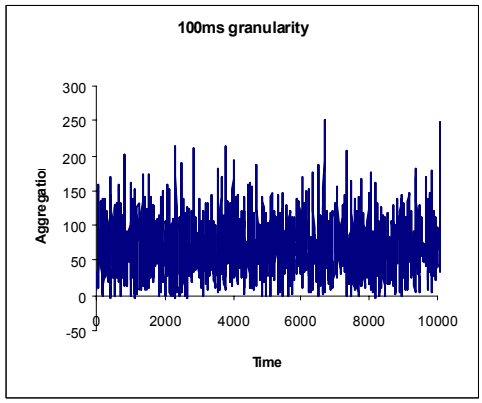



(c) 10ms granularity

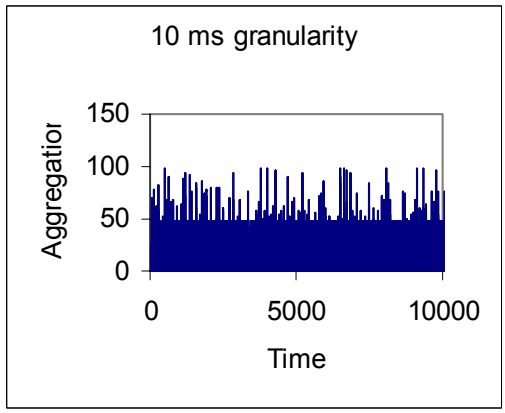
Figure A.1. Traffic characetristics at an ingress node. *TTh* = 10µs. Input H=0.9. Employed scheme: Time Threshold Based Burst Assembly. Network Load: 7 E. Output H = 0.5204




(a)1s granularity

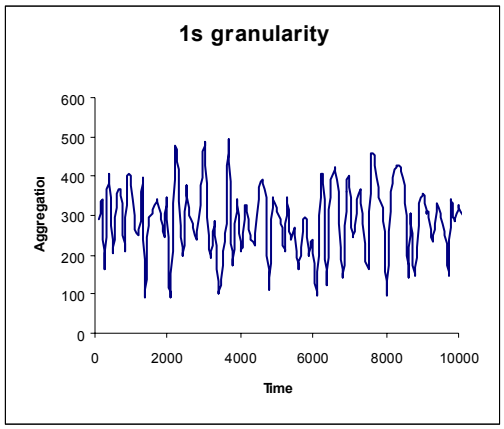



(b) 100ms granularity

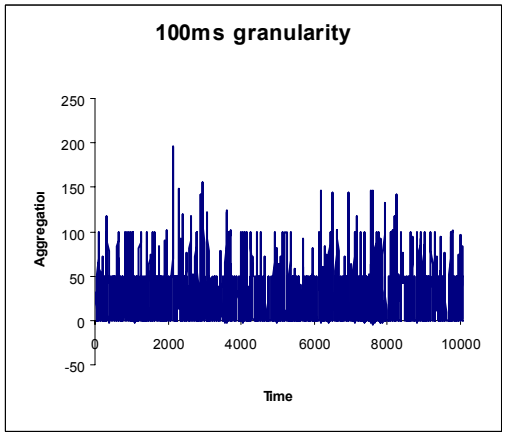



(c) 10ms granularity

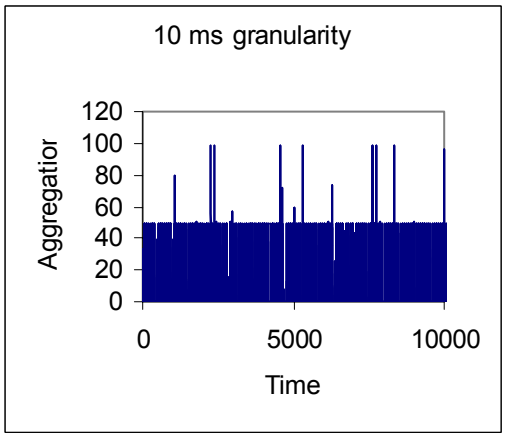
Figure A.2. Traffic characetristics at an ingress node. *TTh* = 10µs. Input H=0.9. Employed scheme: Time Threshold Based Burst Assembly. Network Load: 14 E. Output H = 0.5250.




(a)1s granularity

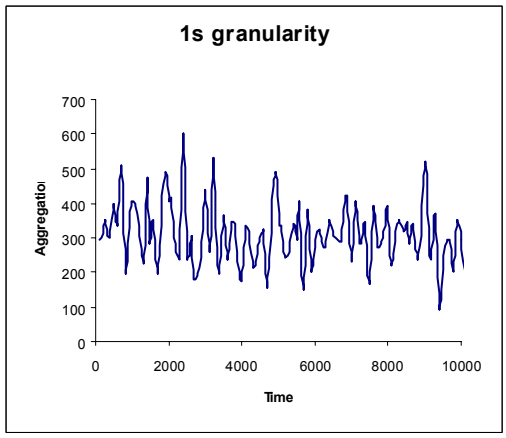



(b) 100ms granularity

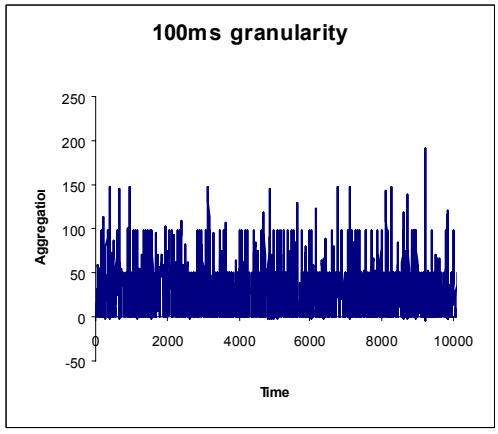



(c) 10ms granularity

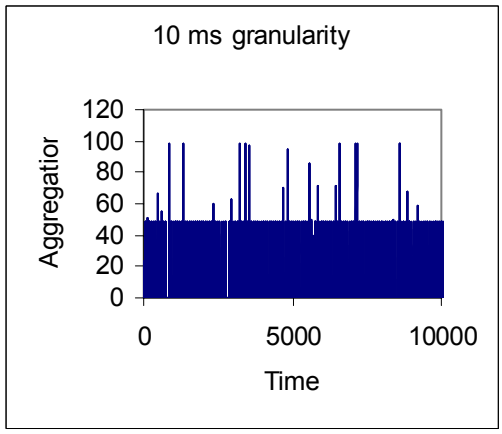
Figure A.3. Traffic characetristics at an ingress node. TTh = 16µs. Input H=0.9. Employed scheme: Time Threshold Based Burst Assembly. Network Load: 7 E. Output H = 0.5115




(a)1s granularity

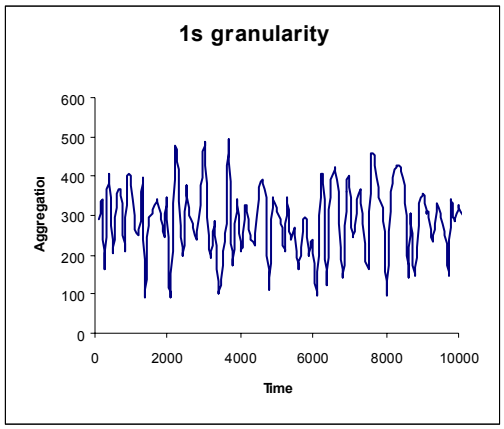



(b) 100ms granularity

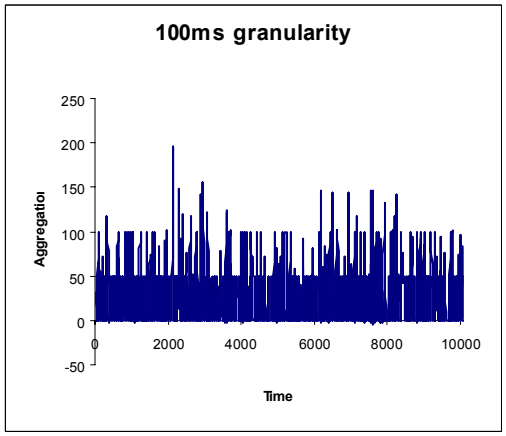



(c) 10ms granularity

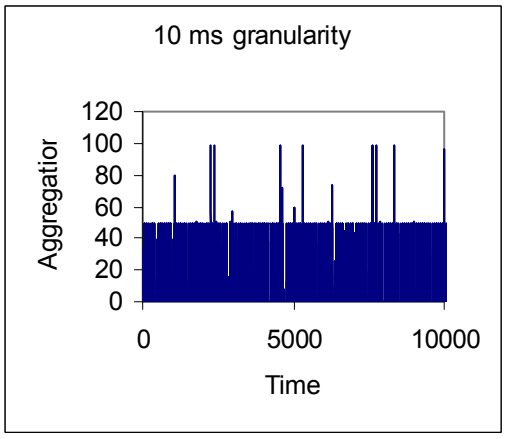
Figure A.4. Traffic characetristics at an ingress node. *TTh* = 16µs. Input H=0.9. Employed scheme: Time Threshold Based Burst Assembly. Network Load: 14 E. Output H = 0.5065




(a)1s granularity

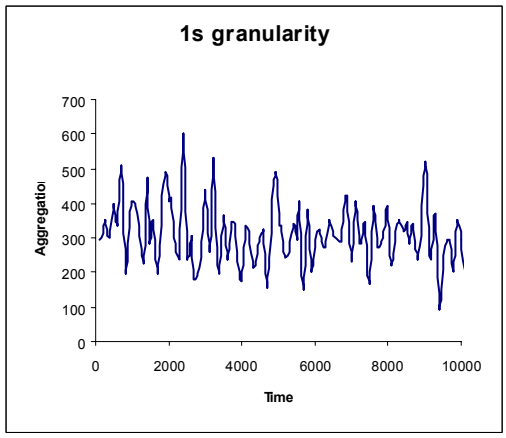



(b) 100ms granularity

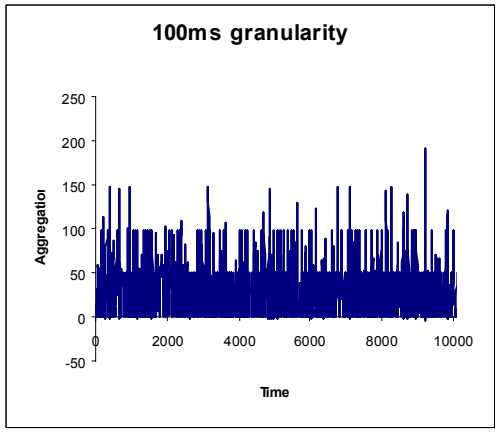



(c) 10ms granularity

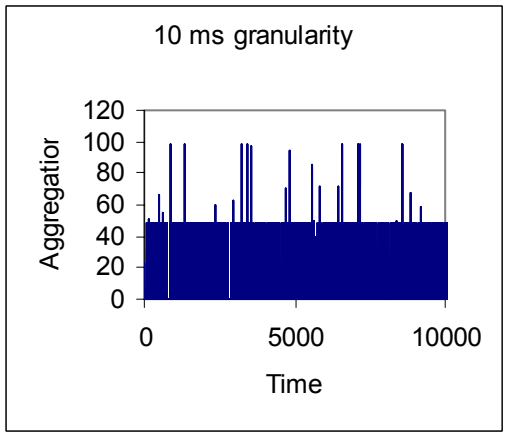
Figure A.5. Traffic characetristics at an ingress node. TTh = 10µs. Input H=0.9. Employed scheme: Hybrid Burst Assembly. Network Load: 7 E. Output H = 0.5146




(a)1s granularity

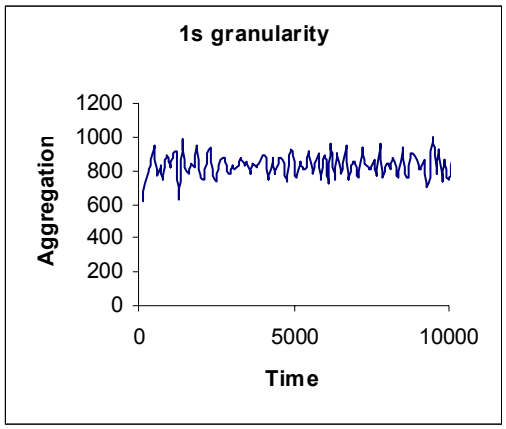



(b) 100ms granularity

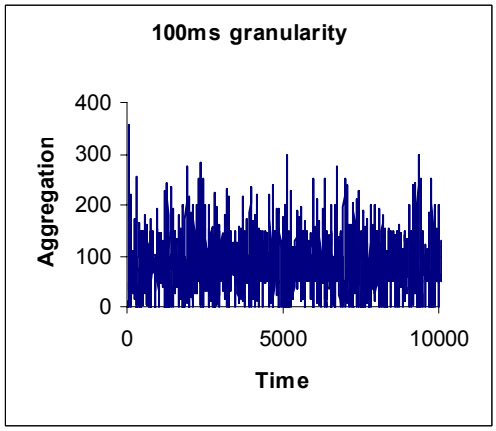



(c) 10ms granularity

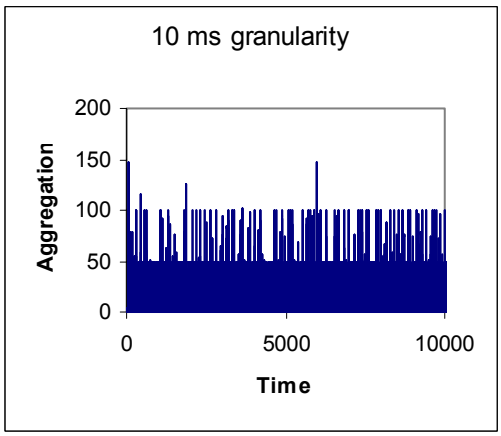
Figure A.6. Traffic characetristics at an ingress node. *TTh* = 10µs. Input H=0.9. Employed scheme: Hybrid Burst Assembly. Network Load: 14 E. Output H = 0.5142




(a)1s granularity

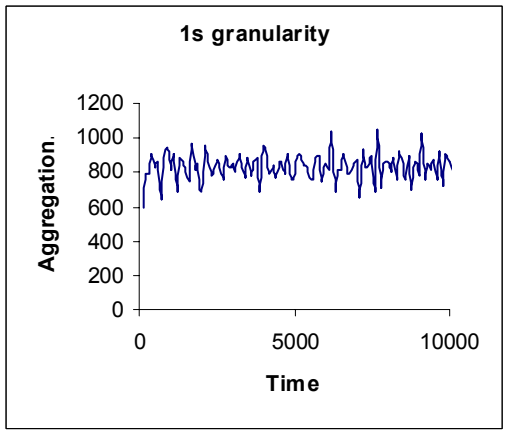



(b) 100ms granularity

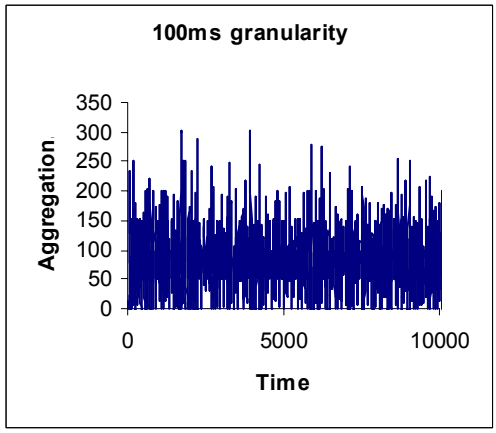



(c) 10ms granularity

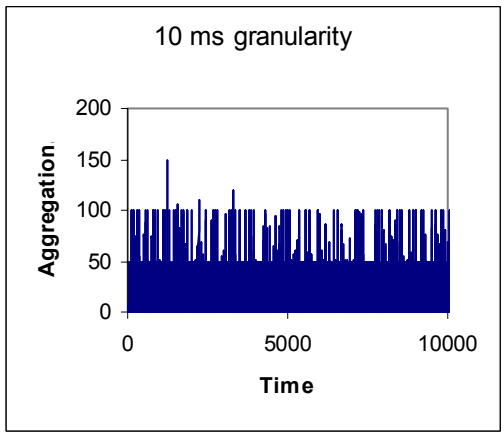
Figure A.7. Traffic characetristics at an ingress node. *TTh* = 16μs. Employed scheme: Hybrid Burst Assembly. Network Load: 7 E. Output Hurst Parameter = 0.52636




(a)1s granularity



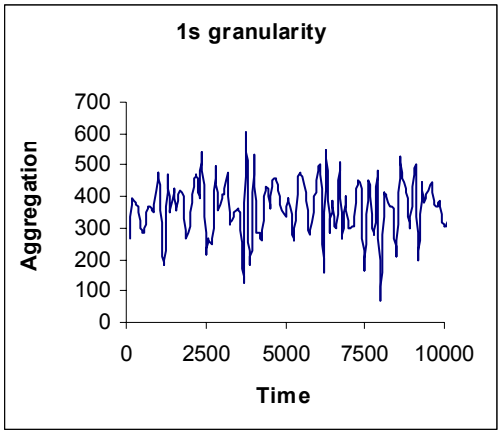

(b) 100ms granularity



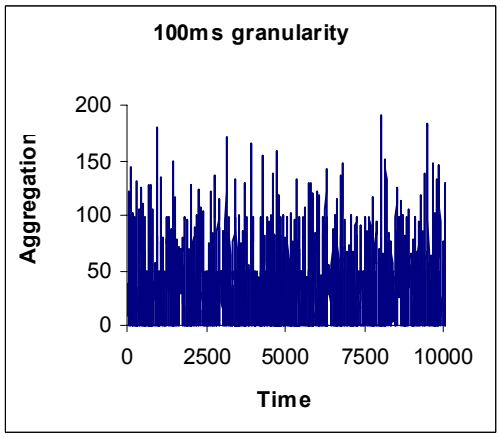

(c) 10ms granularity

Figure A.8. Traffic characetristics at an ingress node. *TTh* = 16μs. Employed scheme: Hybrid Burst Assembly. Network Load: 14 E. Output Hurst Parameter = 0.53347

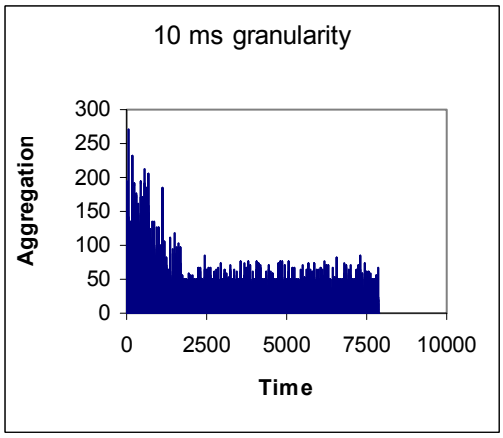
## APPENDIX B




(a)1s granularity

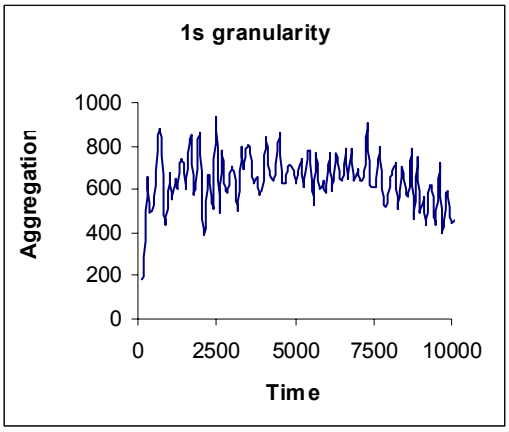



(b) 100ms granularity

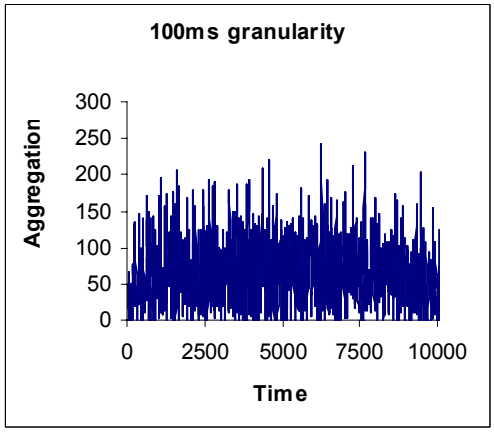



(c) 10ms granularity

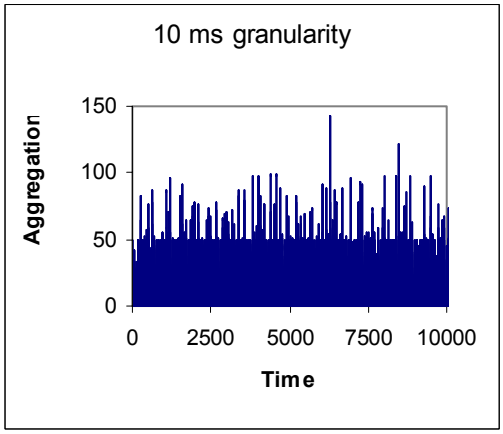
Figure B.1. Traffic characetristics at an egress node. *TTh* = 10μs. Burstification scheme: *TTh* Based Burst Assembly. OBS scheduling scheme: The horizon. Network Load: 13 E. Output Hurst Parameter = 0.5412




(a) 1s granularity

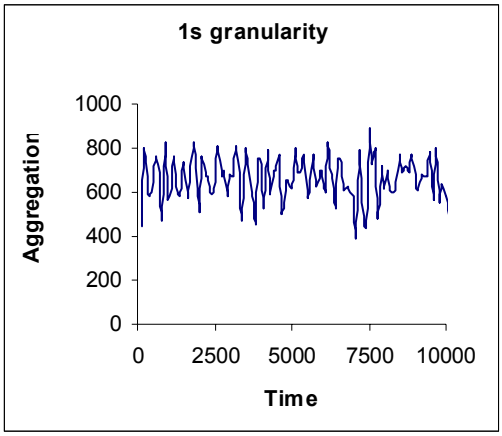



(b) 100ms granularity

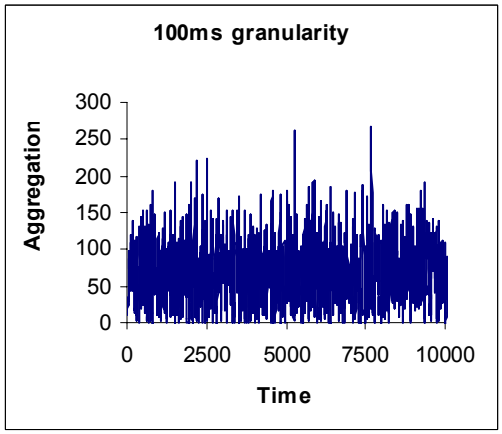



(c) 10ms granularity

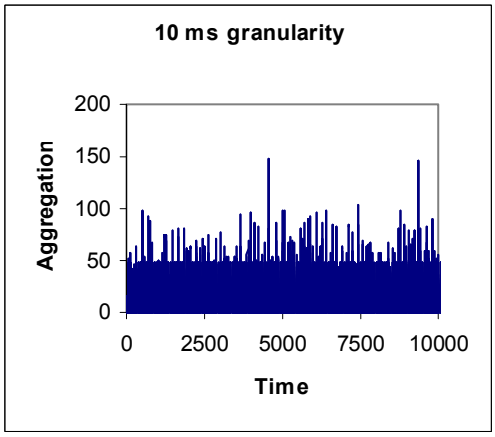
Figure B.2. Traffic characetristics at an egress node. *TTh* = 10μs. Burstification scheme: *TTh* Based Burst Assembly. OBS scheduling scheme: First-Fit VF. Network Load: 13 E. Output Hurst Parameter = 0.5522




(a)1s granularity

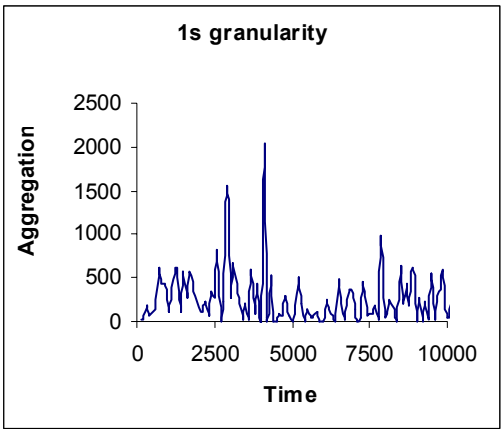



(b) 100ms granularity

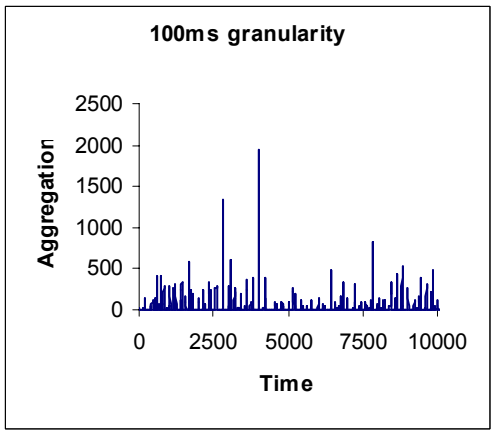



(c) 10ms granularity

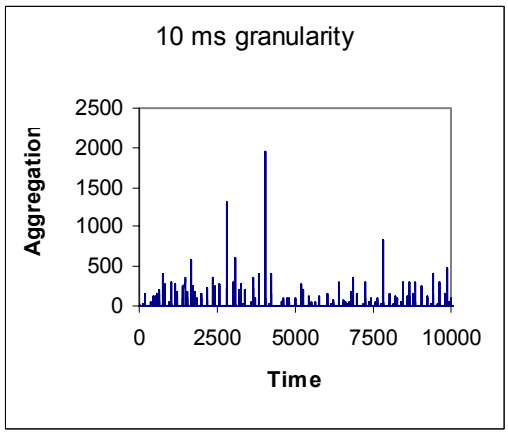
Figure B.3. Traffic characetristics at an egress node. *TTh* = 10μs. Burstification scheme: *TTh* Based Burst Assembly. OBS scheduling scheme: LAUC-VF. Network Load: 13 E. Output Hurst Parameter = 0.5610




(a)1s granularity

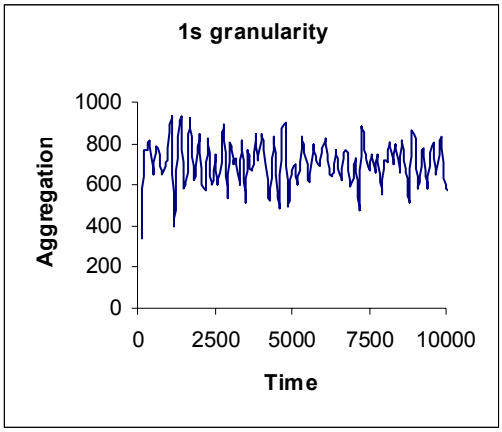



(b) 100ms granularity

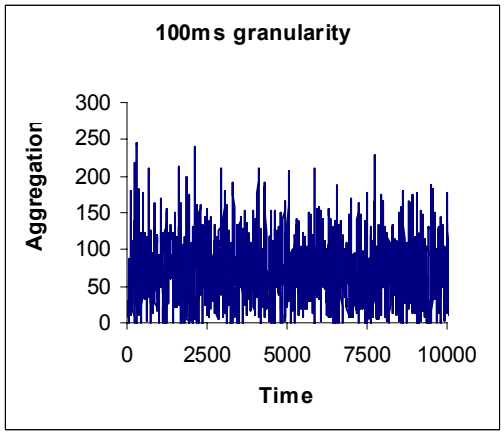



(c) 10ms granularity

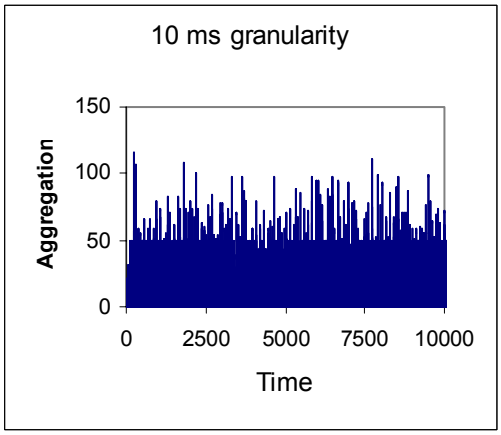
Figure B.4. Traffic characetristics at an egress node. *TTh* = 10μs. Burstification scheme: *TTh* Based Burst Assembly. OBS scheduling scheme: Group Scheduling. Network Load: 13 E. Output Hurst Parameter = 0.6111




(a)1s granularity

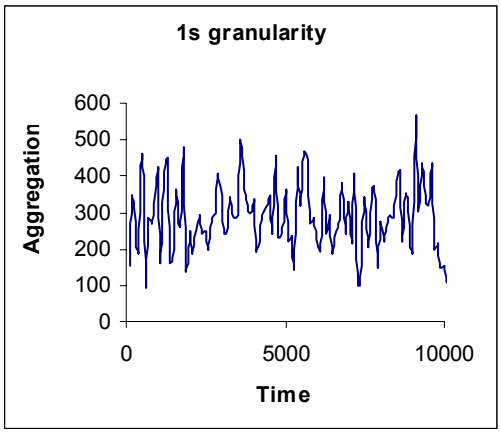



(b) 100ms granularity

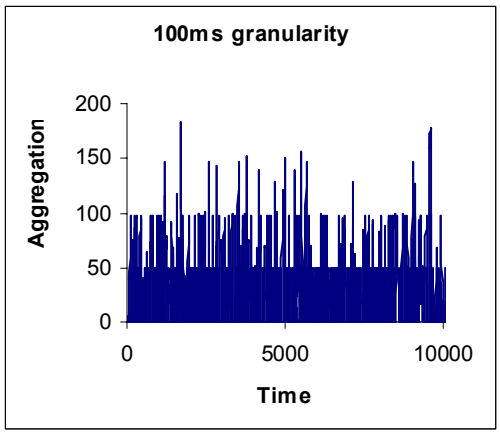



(c) 10ms granularity

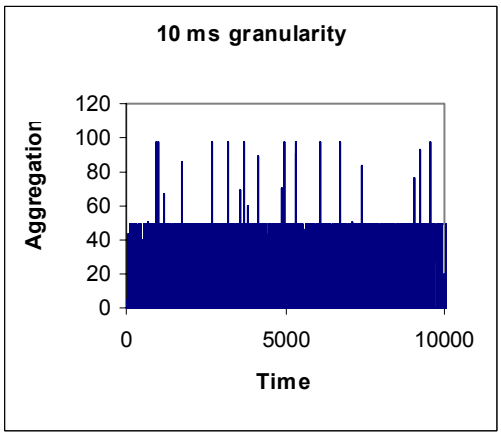
Figure B.5. Traffic characetristics at an egress node. *TTh* = 10μs. Burstification scheme: *TTh* Based Burst Assembly. OBS scheduling scheme: DFMOC-VF. Network Load: 13 E. Output Hurst Parameter = 0.5665




(a)1s granularity

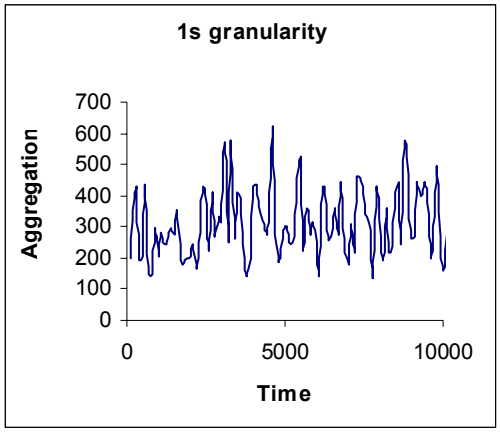



(b) 100ms granularity

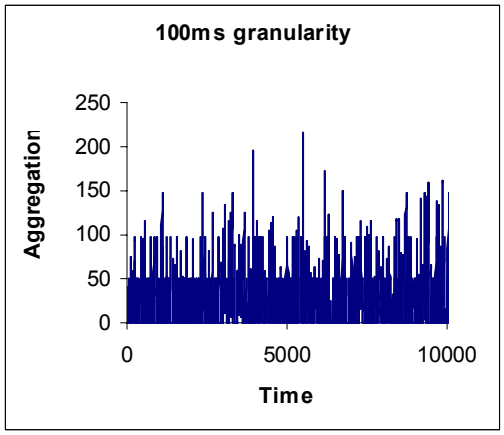



(c) 10ms granularity

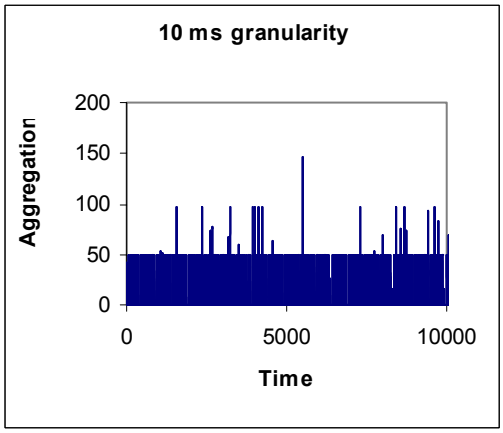
Figure B.6. Traffic characetristics at an egress node. *TTh* = 16μs. Burstification scheme: *TTh* Based Burst Assembly. OBS scheduling scheme: The horizon algorithm. Network Load: 13 E. Output Hurst Parameter = 0.4987




(a)1s granularity

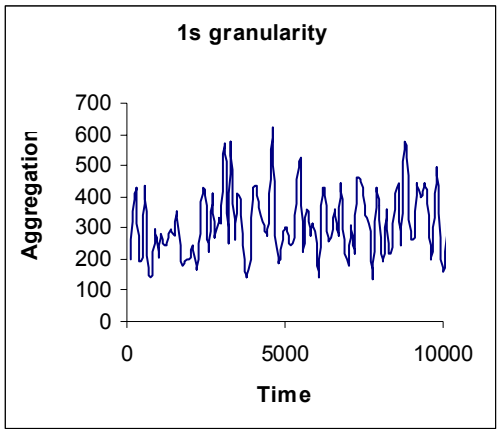



(b) 100ms granularity

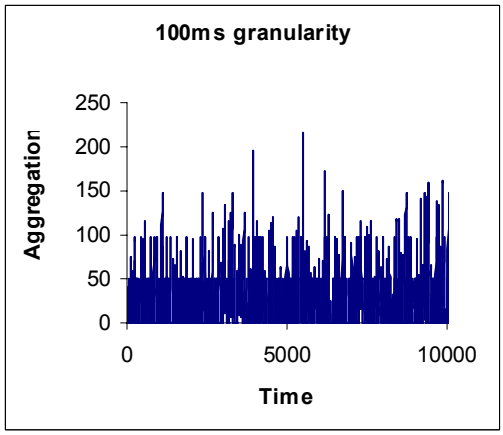



(c) 10ms granularity

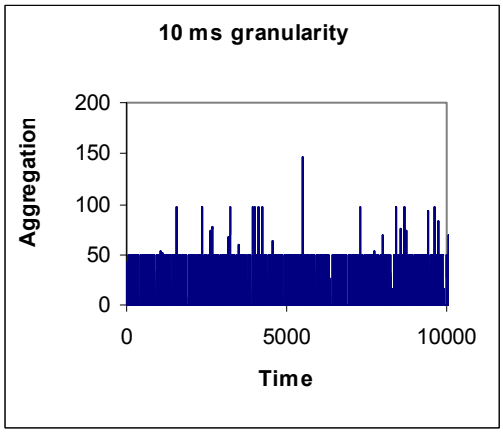
Figure B.7. Traffic characetristics at an egress node. *TTh* = 16μs. Burstification scheme: *TTh* Based Burst Assembly. OBS scheduling scheme: First-Fit VF. Network Load: 13 E. Output Hurst Parameter = 0.4933




(a)1s granularity

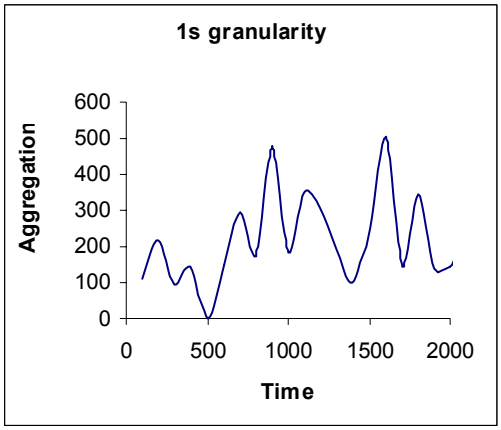



(b) 100ms granularity

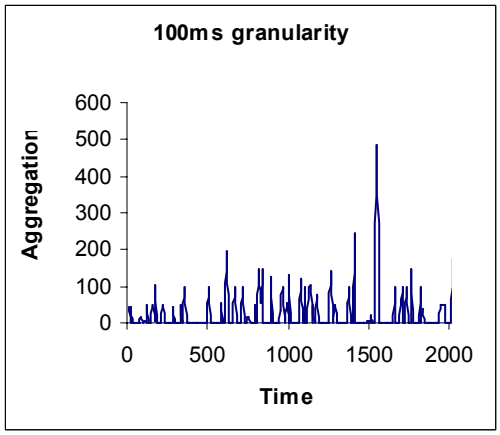



(c) 10ms granularity

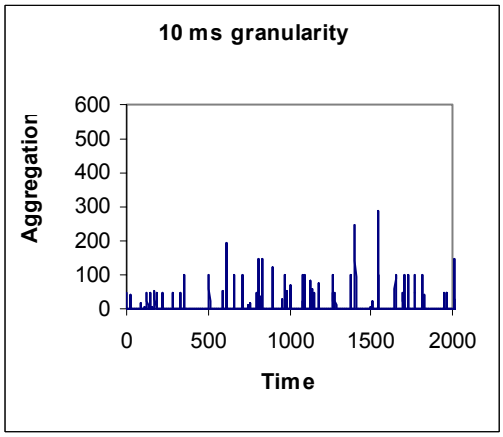
Figure B.8. Traffic characetristics at an egress node. *TTh* = 16μs. Burstification scheme: *TTh* Based Burst Assembly. OBS scheduling scheme: LAUC-VF. Network Load: 13 E. Output Hurst Parameter = 0.5001




(a)1s granularity

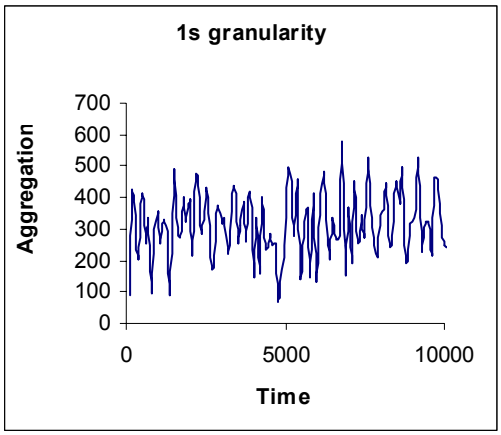



(b) 100ms granularity

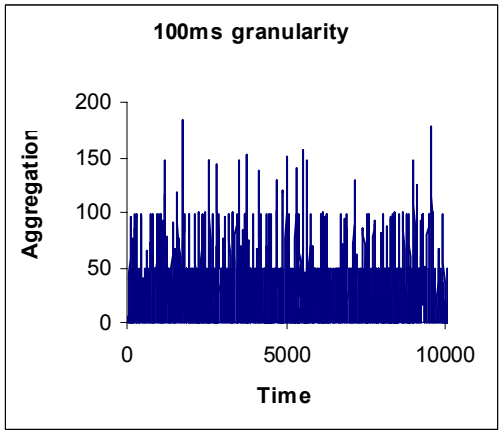



(c) 10ms granularity

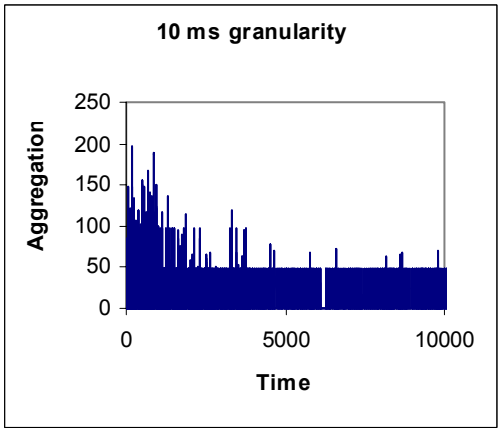
Figure B.9. Traffic characetristics at an egress node. *TTh* = 16μs. Burstification scheme: *TTh* Based Burst Assembly. OBS scheduling scheme: Group Scheduling. Network Load: 13 E. Output Hurst Parameter = 0.5650




(a)1s granularity

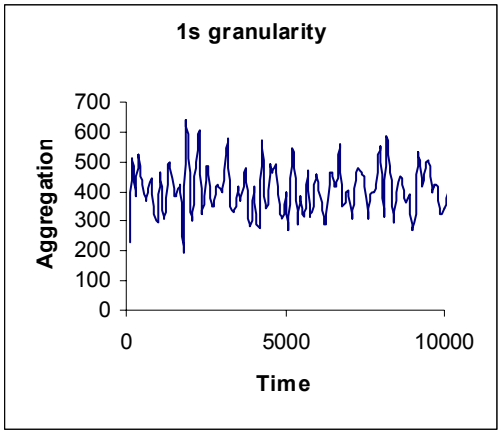



(b) 100ms granularity

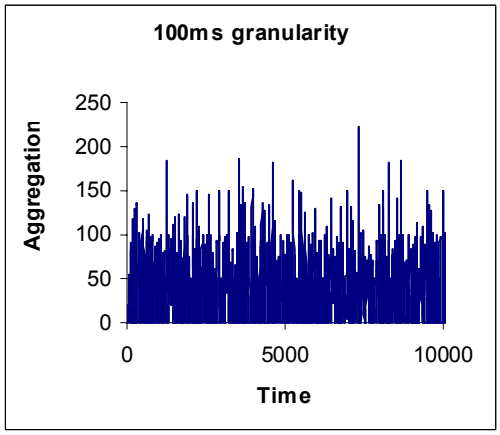



(c) 10ms granularity

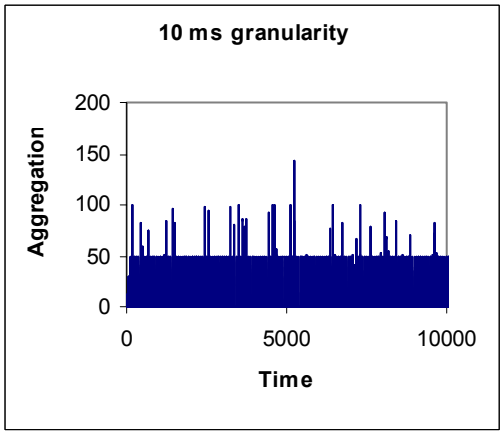
Figure B.10. Traffic characetristics at an egress node. *TTh* = 16μs. Burstification scheme: *TTh* Based Burst Assembly. OBS scheduling scheme: DFMOC-VF. Network Load: 13 E. Output Hurst Parameter = 0.5087




(a)1s granularity

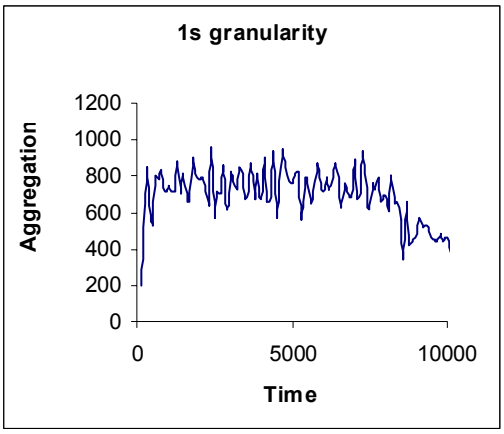



(b) 100ms granularity

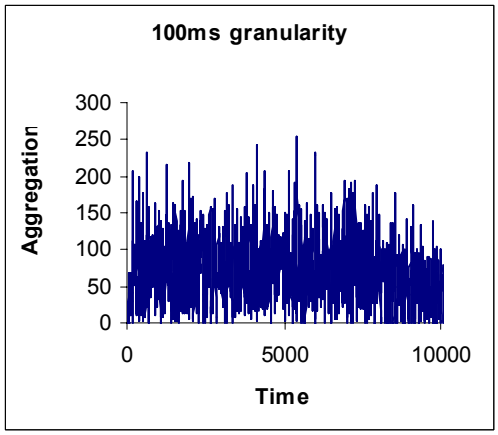



(c) 10ms granularity

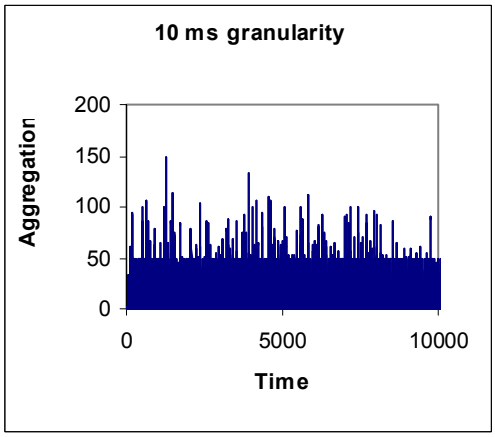
Figure B.11. Traffic characetristics at an egress node. *TTh* = 10μs. Burstification scheme: Hybrid Burst Assembly. OBS scheduling scheme: The horizon algorithm. Network Load: 13 E. Output Hurst Parameter = 0.5327




(a)1s granularity

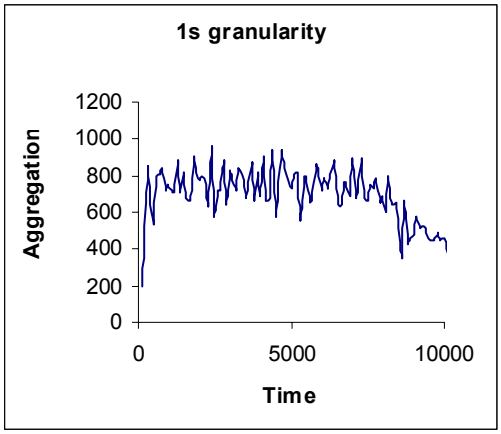



(b) 100ms granularity

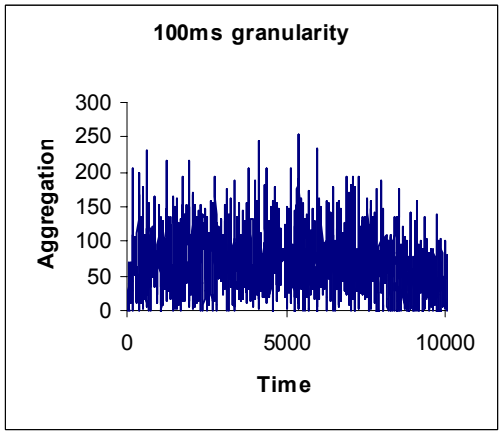



(c) 10ms granularity

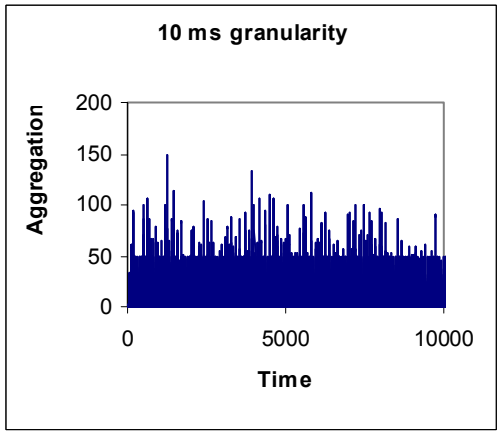
Figure B.12. Traffic characetristics at an egress node. *TTh* = 10μs. Burstification scheme: Hybrid Burst Assembly. OBS scheduling scheme: First-Fit VF. Network Load: 13 E. Output Hurst Parameter = 0.5862




(a)1s granularity

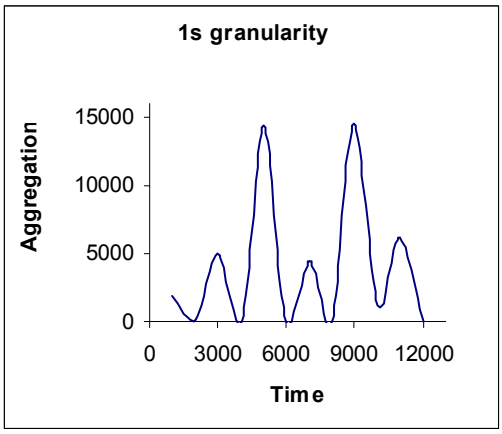



(b) 100ms granularity

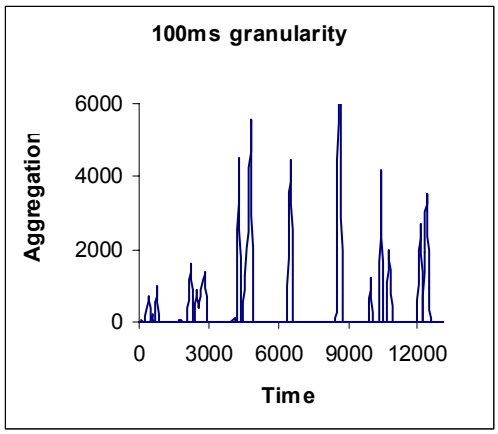



(c) 10ms granularity

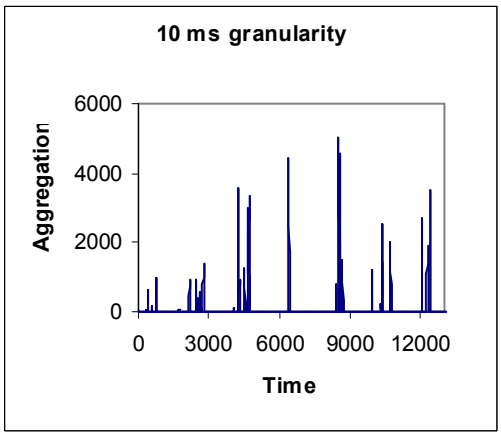
Figure B.13. Traffic characetristics at an egress node. *TTh* = 10μs. Burstification scheme: Hybrid Burst Assembly. OBS scheduling scheme: LAUC-VF. Network Load: 13 E. Output Hurst Parameter = 0.5817




(a)1s granularity

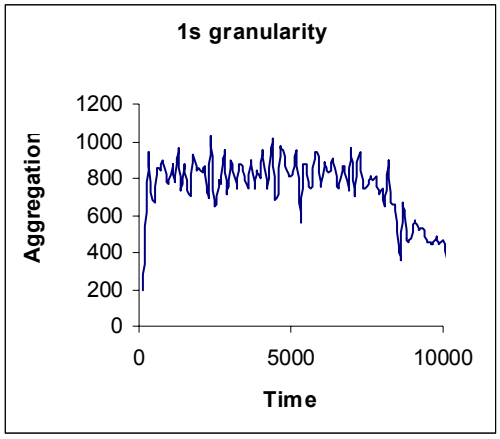



(b) 100ms granularity

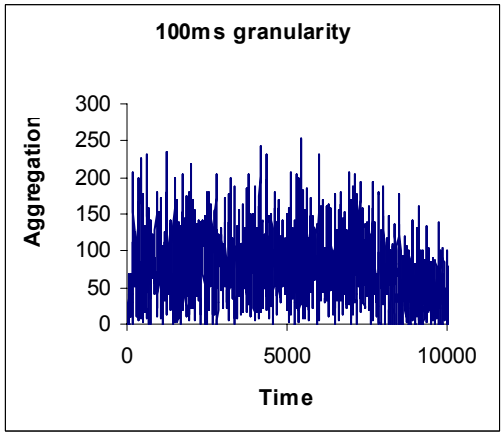



(c) 10ms granularity

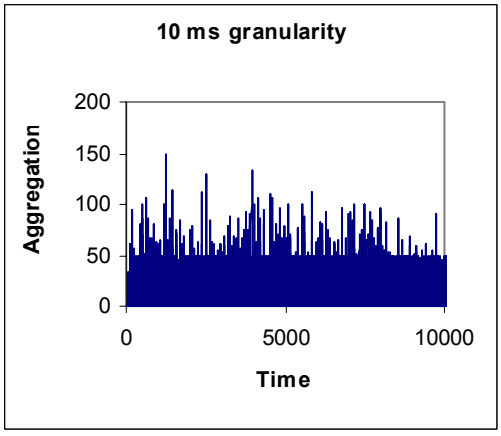
Figure B.14. Traffic characetristics at an egress node. *TTh* = 10μs. Burstification scheme: Hybrid Burst Assembly. OBS scheduling scheme: Group Scheduling. Network Load: 13 E. Output Hurst Parameter = 0.7102




(a)1s granularity

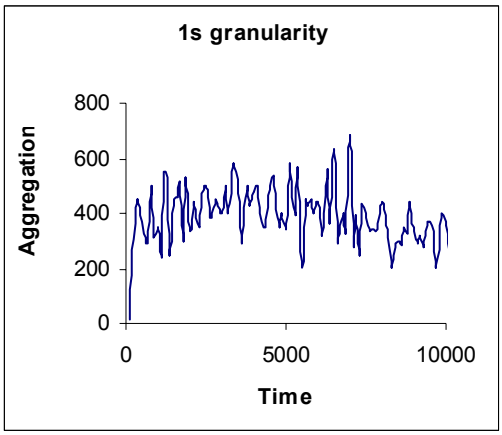



(b) 100ms granularity

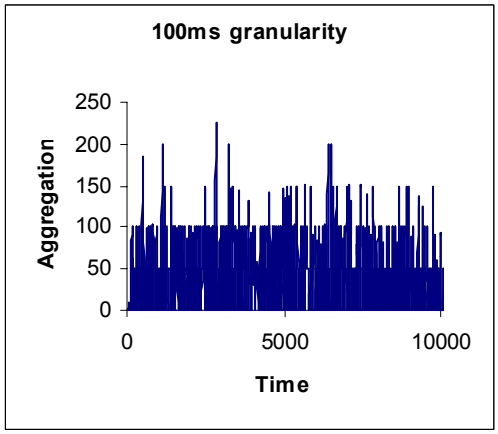



(c) 10ms granularity

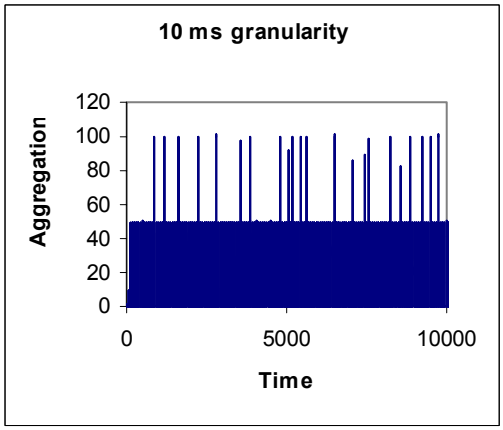
Figure B.15. Traffic characetristics at an egress node. *TTh* = 10μs. Burstification scheme: Hybrid Burst Assembly. OBS scheduling scheme: DFMOC-VF. Network Load: 13 E. Output Hurst Parameter = 0.6012




(a)1s granularity

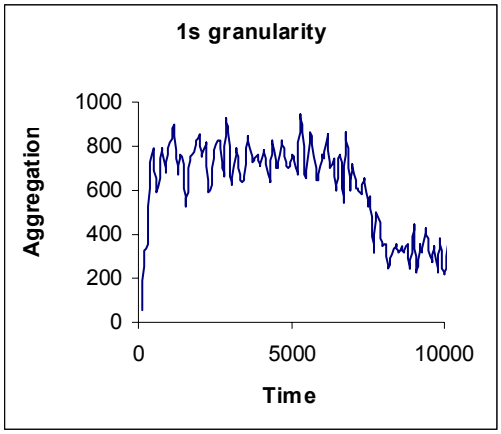



(b) 100ms granularity

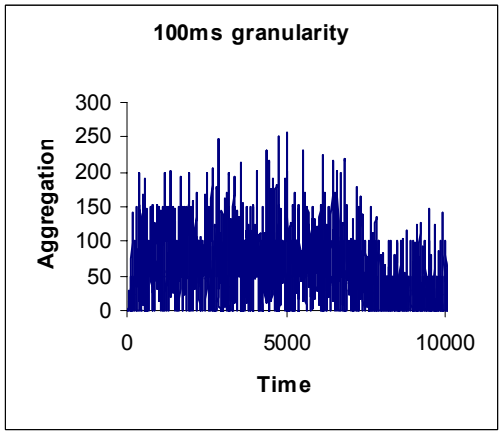



(c) 10ms granularity

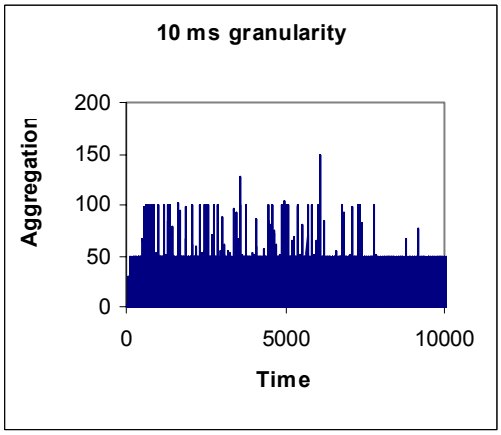
Figure B.16. Traffic characetristics at an egress node. *TTh* = 16μs. Burstification scheme: Hybrid Burst Assembly. OBS scheduling scheme: The horizon algorithm. Network Load: 13 E. Output Hurst Parameter = 0.5314




(a)1s granularity

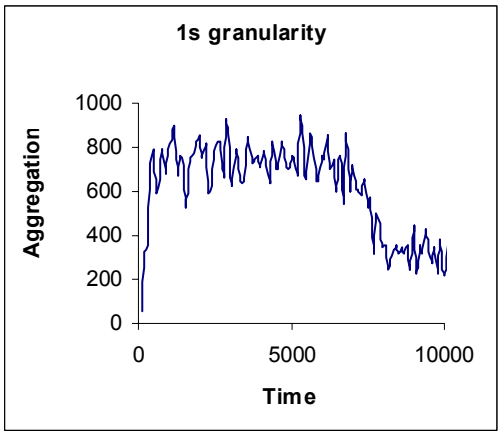



(b) 100ms granularity

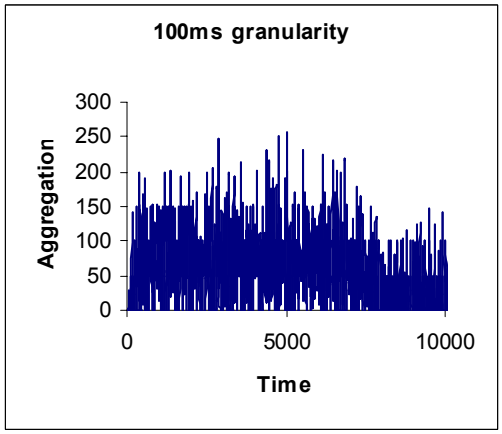



(c) 10ms granularity

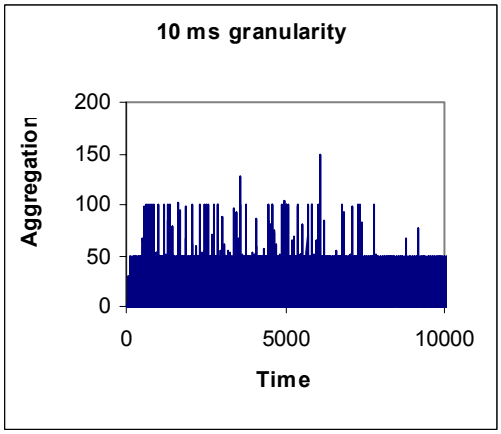
Figure B.17. Traffic characetristics at an egress node. *TTh* = 16μs. Burstification scheme: Hybrid Burst Assembly. OBS scheduling scheme: First-Fit VF. Network Load: 13 E. Output Hurst Parameter = 0.56417




(a)1s granularity

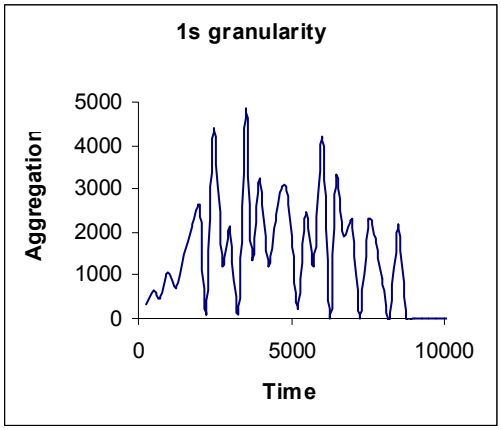



(b) 100ms granularity

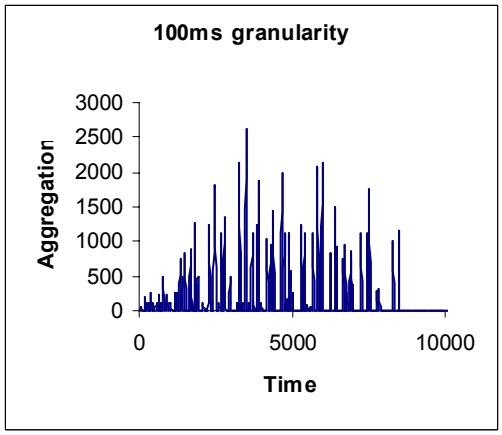



(c) 10ms granularity

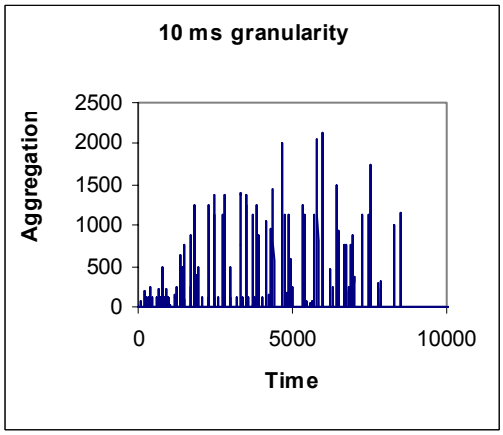
Figure B.18. Traffic characetristics at an egress node. *TTh* = 16μs. Burstification scheme: Hybrid Burst Assembly. OBS scheduling scheme: LAUC-VF. Network Load: 13 E. Output Hurst Parameter = 0.56631




(a)1s granularity

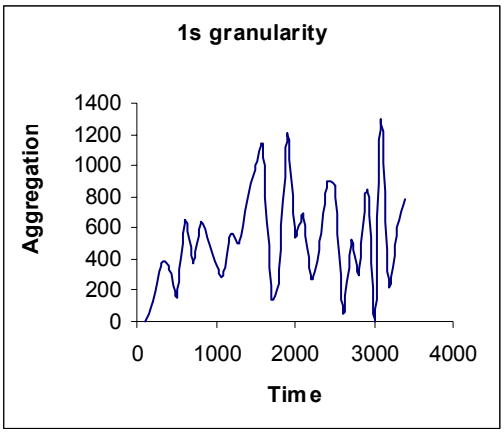



(b) 100ms granularity

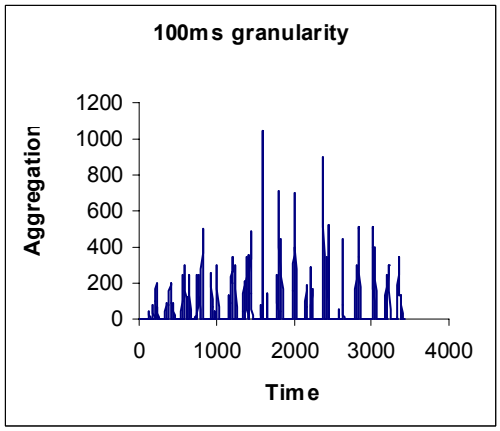



(c) 10ms granularity

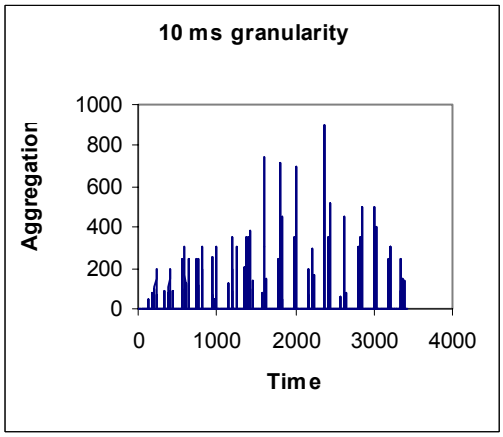
Figure B.19. Traffic characetristics at an egress node. *TTh* = 16μs. Burstification scheme: Hybrid Burst Assembly. OBS scheduling scheme: Group Scheduling. Network Load: 13 E. Output Hurst Parameter = 0.7341




(a)1s granularity




(b) 100ms granularity




(c) 10ms granularity

Figure B.20. Traffic characetristics at an egress node. *TTh* = 16μs. Burstification scheme: Hybrid Burst Assembly. OBS scheduling scheme: DFMOC-VF. Network Load: 13 E. Output Hurst Parameter = 0.68427

## BIOGRAPHY

Burak KANTARCI was born in İstanbul in 1981. Upon graduating Vefa Anatolian Highschool he started studying in Computer Engineering Department in Istanbul Technical University. Upon completing BSc program of Computer Engineering in 2002, he started the MSc program of Computer Engineering in Istanbul Technical University. From June 2002 to June 2004 he worked as information system coordinator at Univesity of Bahcesehir, Istanbul. Since June 2004, he has been working as a research and teaching assistant at Istanbul Technical University in Department of Computer Engineering. His area of interests are optical switching, QoS issues, adaptive network protocols, and wireless sensor sytems.