| | |
|---|---|
| Abstract checked<br>During the past several years there has | Tóm tắt<br>Trong vài năm qua, đã có sự gia tăng |

| | |
|---|---|
| been an enormous increase of data and IP traffic. The evolution of emerging applications with high throughput requirements is constantly raising the demand for high-bandwidth network technologies. Considering today's technology and network growth, this essentially translates in enhancing the deployment of optical networks. Various techniques have been studied and developed with the more mature being Wavelength Division Multiplexing (WDM) based on Optical Circuit Switching (OCS). Optical Burst Switching (OBS) has been proposed as an alternative switching technique providing more efficient utilization of WDM networks and improved performance under bursty traffic conditions compared to OCS. | rất mạnh về lưu lượng dữ liệu và lưu lượng IP. Sự phát triển của các ứng dụng mới với yêu cầu thông lượng cao làm gia tăng liên tục nhu cầu về các công nghệ mạng băng thông cao. Với sự sự phát triển công nghệ và mạng như ngày nay, chúng ta buộc phải tăng cường triển khai các mạng quang học. Những kỹ thuật khác nhau đã được nghiên cứu và phát triển ngày càng vững mạnh hơn trong đó có thể kể đến là kỹ thuật Ghép Kênh Quang Theo Bước Sóng (WDM) dựa trên Chuyển Mạch Kênh Quang (OCS). Sau đó các nhà nghiên cứu đề xuất kỹ thuật chuyển mạch chùm quang có khả năng tận dụng mạng WDM hiệu quả hơn và cải thiện hiệu suất trong các điều kiện lưu lượng dạng cụm so với OCS.<br>bursty traffic: lưu lượng truyền loạt, lưu lượng thay đổi với biên độ lớn, lưu lượng dạng cụm, Sự phân bố lưu lượng không đều, rất cao trong một khoảng thời gian nhất định và rất thấp trong thời gian còn lại. |
| For the purposes of this graduate project we used the OPNET Modeler tool to simulate and evaluate the performance of an OBS network. More specifically an OBS core router was modeled, while an existing edge OBS router model was extended to include some new attributes. Several reservation and scheduling policies were implemented and the impact of various possible configuration parameters to the overall network performance was evaluated in terms of end-to- end delay and burst loss | Trong luận văn này, chúng tôi đã dùng công cụ OPNET Modeler để mô phỏng và đánh giá hiệu suất của mạng OBS. Cụ thể, chúng tôi mô hình hoá bộ định tuyến lõi OBS, đồng thời mở rộng mô hình bộ định tuyến biên OBS hiện tại để gộp vào một số thuộc tính mới. Chúng tôi thực thi một số chính sách dành trước và chính sách lập lịch và đánh giá tác động của các tham số cấu hình khả dĩ khác nhau đến hiệu suất mạng tổng thể theo độ trễ đầu cuối và xác suất tổn thất burst (chùm). |

| | |
|---|---|
| probability. | |
| Keywords: Optical Burst Switching, Just-Enough-Time (JET), Just-In-Time (JIT), Fiber Delay Lines, LAUC-VF | Từ khoá: Chuyển Mạch Chùm Quang, Giao Thức Just-Enough-Time (JET), Giao thức Just-In-Time (JIT), Đường Trễ Quang, LAUC-VF<br>Just-Enough-Time: giao thức “thời gian vừa đủ”<br>Just-In-Time: Giao thức “vừa đúng lúc” |
| Contents | Nội dung |
| Abstract 3 | Tóm tắt 3 |
| Contents 4 | Mục lục 4 |
| List of Figures 6 | Danh mục hình ảnh 6 |
| List of Tables 8 | Danh mục bảng 8 |
| Chapter 1: Background 9 | Chương 1: Các kiến thức cơ bản 9 |
| 1.1 INTRODUCTION 10 | 1.1 GIỚI THIỆU 10 |
| 1.1.1 Optical Circuit Switching 10 | 1.1.1 Chuyển mạch chùm quang 10 |
| 1.1.2 Optical Packet Switching 10 | 1.1.2 Chuyển mạch gói quang 10 |
| 1.1.3 Optical Burst Switching 11 | 1.1.3 Chuyển mạch chùm quang 11 |
| 1.2 TECHNOLOGY AND ARCHITECTURE 13 | 1.2 CÔNG NGHỆ VÀ KIẾN TRÚC 13 |
| 1.3 BURST ASSEMBLY 14 | 1.3 LẬP BURST 14 |
| 1.4 SIGNALING SCHEMES 14 | 1.4 PHƯƠNG PHÁP PHÁT TÍN HIỆU 14 |
| 1.4.1 JET 15 | 1.4.1 JET 15 |
| 1.4.2 JIT 17 | 1.4.2 JIT 17 |
| 1.4.3 Analytical Delay Model-(JET and JIT) 18 | 1.4.3 Mô Hình Trễ PhânTích-(JET và JIT) 18 |
| 1.5 CONTENTION RESOLUTION 18 | 1.5 GIẢI QUYẾT TRANH CHẤP 18 |
| 1.5.1 Wavelength Conversion 19 | 1.5.1 Chuyển đổi bước sóng 19 |
| 1.5.2 Deflection Routing 19 | 1.5.2 Định tuyến lệch hướng |
| 1.5.3 Burst Segmentation 20 | 1.5.3 Phân đoạn Burst 20 |
| 1.5.4 Segmentation with Deflection 20 | 1.5.4 Phân Đoạn bằng cách làm Lệch 20 |
| 1.5.5 Optical Buffering 20 | 1.5.5 Cơ chế đệm quang học 20 |
| 1.6 CHANNEL SCHEDULING | 1.6 LẬP LỊCH TRÌNH KÊNH |

| | |
|---|---|
| 47 | 47 |
| Figure 23: End-to-End Delay (JIT) 48 | Hình 23: Độ trễ đầu cuối (JIT) 48 |
| Figure 24: Burst Loss Ratio (LAUC-VF) 48 | Hình 24: Tỷ lệ tổn hao Burst (LAUC-VF) 48 |
| Figure 25: End-to-End Delay (LAUC-VF) 49 | Hình 25: Độ trễ đầu cuối (LAUC-VF) 49 |
| Figure 26: End-to-End Delay vs. Load (JET) 50 | Hình 26: Độ trễ và Tải đầu cuối (JET) 50 |
| Figure 27: End-to-End Delay vs. Load (JIT) 51 | Hình 27: Độ trễ và Tải đầu cuối (JIT) 51 |
| Figure 28: End-to-End Delay vs. Load (LAUC-VF) 51 | Hình 28: Độ trễ và Tải đầu cuối (LAUC-VF) 51 |
| Figure 29: Burst Loss Ratio vs. Load 52 | Hình 29: Tỷ lệ tổn hao Burst và Tải 52 |
| List of Tables | Danh mục các bảng |
| Table 1: Implementation Models 33 | Bảng 1: Các mô hình thực thi 33 |
| Table 2: OBS Edge Router Characteristics 34 | Bảng 2: Các đặc trưng của bộ định tuyến biên OBS 34 |
| Table 3: OBS Core Router Characteristics 34 | Bảng 3: Các đặc trưng của bộ định tuyến lõi OBS 34 |
| Table 4: Range of values used for parameterization of offset time 37 | Bảng 4: Khoảng giá trị được dùng để tham số hoá thời gian offset 37 |
| Table 5: Delays assigned at the FDL buffer stages 39 | Bảng 5: Độ trễ được ấn định ở các tầng đệm FDL 39 |
| Table 6: Simulation parameters for parametric studies 42 | Bảng 6: Các tham số mô phỏng của các nghiên cứu tham số 42 |
| Table 7: Simulation parameters for optimum performance 46 | Bảng 7: Tham số mô phỏng để hiệu suất tối ưu 46 |
| Table 8: "Traffic Scaling Factor" values 50 | Bảng 8: Giá trị "Hệ Số Tỷ Lệ Lưu Lượng " 50 |
| Chapter 1: Background | [redacted] |
| 1.1 INTRODUCTION | [redacted] |
| Over the last decade, there has been a phenomenal increase in internet traffic and the variety of internet applications. Various architectures as wavelength-division multiplexing (WDM) and dense-WDM as well as | [redacted] |

several optical switching techniques have been studied to meet the demands for high bandwidth. Three switching technologies have been studied Optical Circuit Switching (OCS), Optical Packet Switching (OPS) and Optical Burst Switching (OBS). In order to highlight the differences between these three we analyze each approach in detail.

### 1.1.1 Optical Circuit Switching

In circuit switching networks all-optical wavelength paths are established between pairs of nodes. The establishment of optical paths involves several tasks as topology and resource discovery, routing, wavelength assignment and signaling and resource reservation. Transmission starts only after an end-to-end path reservation is acknowledged. Allocated resources are exclusively available to the end-to-end connection for its entire duration.

Optical circuit switching has the largest transmission granularity among the three switching techniques. However it is not bandwidth efficient unless the duration of transmission is greater than the circuit establishment period. Moreover, it has its own limitations to support dynamic traffics. It is clear that if traffic is varying dynamically, then sending this traffic over static optical paths would result in the inefficient utilization of bandwidth. On the other hand, if we attempt to set up optical paths in a very dynamic manner, then the

[redacted]

network state information will be constantly changing, making it difficult to maintain current network state information. Thus, as traffic becomes more dynamic and bursty in nature, alternative approaches may be needed to transport data across networks [1].

### 1.1.2 Optical Packet Switching

Ideally, in order to provide the highest possible utilization in the optical core, nodes would need to provide packet switching at the optical level. In all-optical packet switched networks, packets are switched and routed independently through the network entirely in the optical domain without conversion back to electronics at each node. Such networks allow for greater degree of statistical multiplexing on optical fiber lines and are better suited for handling bursty traffic than optical circuit switched networks.

In order for optical packet switching to be practical, fast switching times are required. While switching speeds are expected to improve in the near future, current technology is not yet mature enough to support optical packet switching. Another challenge in optical packet switching is synchronization of packets at switch input ports, required to minimize contentions. Such synchronization is typically difficult to achieve.

Optical packet switching (OPS) uses a

[redacted]

[redacted]

| | |
|---|---|
| hop-by-hop store and forward scheme and needs buffering and processing at each intermediate node. This can result to contention which occurs when two or more packets contend for the same output port at the same time. Contention is difficult to handle in optical packet switched network as it is difficult to handle buffers in the optical domain. Some approaches to resolve contention that have been studied is the use of fiber delay lines to delay a packet for some time and deflection routing to route the contending packets to an output port other than the intended output port. | [redacted] |
| The major advantage of optical packet switched networks is that in these networks packets are transmitted at a full link speed, while in the burst switching, bursts are transmitted only at a channel speed of a Time Division Multiplexing (TDM) link. | [redacted] |
| Optical packet switching networks are flexible, bandwidth efficient and have the least transmission granularity but require practical cost-effective scalable implementations of optical buffering and optical header processing, which are several years away. Thus, all-optical packet switching is likely to be infeasible in the near future due to technological constraints [1]. | [redacted] |
| 1.1.3 Optical Burst Switching<br>Optical Burst Switching (OBS) has been proposed in the late 1990s as a novel optical network architecture | [redacted] |

directed towards efficient transport of IP traffic. OBS is promising approach between Optical Circuit Switching and Optical Packet Switching. OBS that does not require optical buffer and intermediate transmission granularity is emerging as the new switching technique to be used in next generation optical networks.

In Optical Burst Switching, IP packets with a common destination and some other common attributes, like QoS requirements, arriving at the same ingress node are aggregated into large bursts, each being switched and routed individually. OBS can reduce switching time as only a single setup message (burst header control message) and possibly a trailer are associated with each burst. The transmission of each burst is preceded by the transmission of the setup message, whose purpose is to inform each intermediate node of the upcoming data burst so that it can configure its switch fabric in order to switch the burst to the appropriate output port. The setup message can be transmitted on the payload wavelength or on separate control wavelengths. An OBS source node does not wait for confirmation that an end-to-end connection has been set up;
instead it starts transmitting a data burst after a delay (offset time), following the transmission of the setup message [1],[13].

The burst header packet (BHP) apart

[redacted]

[redacted]

[redacted]

from the routing information contains OBS specific information as its payload which includes burst offset-time, data burst duration/length, data channel carrying the burst, the bit rate at which the data burst is sent, and QoS, among others. The offset time allows for the control packet to be processed and the switch to be set up before the burst arrives at the intermediate node; thus no electronic or optical buffering is necessary at the intermediate nodes while the control packet is being processed. The use of the offset time is shown in the following figure:

Figure 1: The use of the offset time [1]

As there is no buffering, bursts may use bandwidth resources along several links and still be blocked and lost without completing their routes in case of congestion or output port conflict. OBS aims to achieve end-to-end optical path speed with high link utilization.

In OBS the reserved resources at each switch and output link port are held for the duration they are needed for switching and transmission of individual bursts. Thus, the resources can be allocated more efficiently. It is worth noting that in OBS networks, where link propagation delay is significantly larger than burst transmission time, multiple bursts could simultaneously propagate not only the same route but also along the same link and wavelength.

[redacted]

[redacted]

It is obvious that OBS has the advantages of both optical circuit switching and optical packet switching while avoids some serious disadvantages. Since data is transmitted in large bursts, optical burst switching reduces the technological requirement of fast optical switches that is necessary for optical packet switching. Moreover, OBS bursts are not buffered at the switches while packets are. Additionally, due to reservations, OBS bursts use the path links in a time-synchronized manner while packets use them asynchronously. Thus, OBS networks allow for a greater degree of statistical multiplexing and are better suited for handling bursty traffic than optical circuit-switched networks.

Despite all the previous advantages there are several issues that need to be considered before optical burst switching can be deployed successfully in working networks. In particular, these issues include burst assembly, signaling schemes, contention resolution, burst scheduling and quality of service.

### 1.2 TECHNOLOGY AND ARCHITECTURE

An optical burst-switched network consists of edge routers, which are responsible for burst assembly and disassembly and core routers, which are mainly responsible for header processing and burst routing. These routers are interconnected via fiber links. Each fiber link is capable of

[redacted]

[redacted]

[redacted]

supporting multiple wavelength channels using wavelength division multiplexing (WDM).

A block diagram of a simple OBS network is shown in the following figure which consists of optical core routers and electronic edge routers connected by WDM links [1], [2].

Figure 2: A simple OBS network [1]

We have ingress edge nodes and egress edge nodes. The ingress assembles packets from the client systems into bursts. It is also responsible for routing, wavelength assignment and scheduling of bursts at the edge node. The assembled-bursts are transmitted all optically over the OBS core network to their destination without any intermediate buffering at nodes. The egress, upon receiving a burst, disassembles the bursts back into packets and forwards the packets to the destination end systems.

We will focus on edge routers and core routers when we describe the architecture of our model network.

### 1.3 BURST ASSEMBLY

A burst assembly scheme is required to determine how packets are assembled into bursts. When packets arrive at the ingress edge node from some client systems they are stored in electronic buffers according to their destination and class. Then the burst assembly mechanism will place these packets into bursts in a specific way. Issues coming along are when to

[redacted]

[redacted]

[redacted]

[redacted]

| | |
|---|---|
| assemble a burst, how many packets to include in a burst and what types of packets to include in a burst. | [redacted] |
| There are several types of burst assembly techniques with most common the timer-based and threshold-based policies. In timer-based approach, a burst is created and sent into the optical network at periodic time intervals. This scheme is used to provide uniform gaps between successive bursts from the same ingress node into the core networks. The length of the burst varies as the load varies. In threshold-based approach, a limit is placed on the maximum number of packets contained in each burst. Thus, fixed size bursts will be generated at the network edge at non-periodic time intervals. The burst assembly scheme will affect the burst length as well as the amount of time that a packet must wait until being transmitted. | [redacted] |
| In general, the shorter the average burst length, the lower the dropping probability. However, if the average length of the burst is short, the whole switching time will increase, which leads to a waste of bandwidth and increases network delay. Thus there is a tradeoff between average burst length, dropping probability and network end to end delay. | [redacted] |
| Both timer and threshold approaches are similar and there are cases that they are both used interchangeably in some OBS network architectures as in | [redacted] |

the model of our study. Actually using both timeout and threshold together provides the best of both schemes, and burst generation is more flexible than having only one of the above. This is because if we calculate the optimum threshold value, the minimum burst length and use a timeout value based on the packet's delay tolerance, we can ensure that we have minimum loss while satisfying the delay requirement [1].

## 1.4 SIGNALING SCHEMES

A signaling scheme is required for reserving resources and configuring switches for an arriving burst at each node. The signaling scheme in an optical burst-switched network is typically implemented using burst header packets transmitted either on the same wavelength as the data burst or on a different wavelength from the burst itself. The header packet travels along the same route as the burst, informing each node along the route to configure its optical switch to accommodate the arriving burst at the appropriate time.

Several variations of optical burst switching signaling protocols are possible, depending on how and when the resources along a route are allocated for a burst. In particular, a signaling scheme can be characterized by the following characteristics: one-way, two-way or hybrid reservation; source- initiated, destination-initiated

[redacted]

or intermediate-node-initiated reservation; persistent or non-persistent reservation; immediate or delayed reservation; explicit or implicit release of resources; centralized or distributed signaling.

Common signaling schemes for reserving resources in OBS networks are Just-Enough-Time(JET), Just-In-Time(JIT), Tell-And-Go (TAG) and Tell-And-Wait (TAW). We will focus our theoretical approach on JET and JIT as these are the signaling schemes that we implemented in our network simulator. The analytical models assume Poisson arrivals, but are valid for arbitrary burst length distributions and arbitrary offset length distributions. The models also account for the processing time of setup messages and the optical switch reconfiguration times.

### 1.4.1 JET

The Just-Enough-Time (JET) signaling technique adopts delayed reservation and implicit release. The delayed reservation scheme operates as follows: "an output wavelength is reserved for a burst just before the arrival of the first bit of the burst; if, upon arrival of the setup message, it is determined that no wavelength can be reserved at the appropriate time, then the setup message is rejected and the corresponding burst is dropped[5]." Implementation of delayed reservation leads to higher bandwidth utilization. Also it leads to the generation of idle voids between the scheduled bursts on

[redacted]

the data channels. Moreover, in an implicit release technique, the control message (BHP) has to carry additional information such as the burst length and the offset time. This technique results in lower burst loss ratio due to the absence of any delay between the actual ending time of the burst and the arrival time of the release control message at each node.

The JET signaling technique is illustrated in the following figure [1]:
Figure 3: JET signaling technique [1]

[redacted]

A source node first sends a burst header packet (BHP) on a control channel towards the destination node. The BHP is processed at each subsequent node in order to establish an all-optical data path for the corresponding data burst. If the reservation is successful, the switch will be configured prior to the burst's arrival. Meanwhile, the burst waits at the source in the electronic domain. After a predetermined amount of time (offset time), the burst is sent optically on the chosen wavelength. The offset time is calculated based on the number of hops from source to destination, and the switching time of the node.

[redacted]

Offset time is calculated as offset_time = h*BHP_proc_Del +Switch_Reconf_delay, where h is the number of hops between the source and the destination, BHP_proc_Del is the per-hop burst header processing time, and Switch_Reconf_delay is the switching reconfiguration time. If at

[redacted]

| | |
|---|---|
| any intermediate node, the reservation is unsuccessful, the burst will be dropped [5]. | [redacted] |
| The information necessary to be maintained for each channel of each output port of every switch for JET comprises of the starting and the ending time of all scheduled bursts, which makes the system rather complex. On the other hand, JET is able to detect situations where no transmission conflict occurs, although the start time of a new burst may be earlier than the finishing time of an already accepted burst, i.e. a burst can be transmitted in between two already reserved bursts. Hence, bursts can be accepted with a higher probability in JET. | [redacted] |
| 1.4.2 JIT<br>JIT is similar to JET except that JIT employs immediate reservation and explicit release instead of delayed reservation and implicit release. JIT is very simple as it does not involve complex scheduling or void filling algorithms and it is amenable to hardware implementation.<br>Immediate reservation works as follows: "an output wavelength is reserved for a burst immediately after the arrival of the corresponding setup message; if a wavelength cannot be reserved at that time, then the setup message is rejected and the corresponding burst is dropped[5]." Immediate reservation is simple and practical to implement, but incurs | [redacted] |

| | |
|---|---|
| higher blocking probability due to inefficient bandwidth allocation. In an explicit release technique, a separate control message is sent following the data burst, from the source towards the destination in order to release or terminate an existing reservation. This technique results in lower bandwidth utilization and increased message complexity.<br>The operation of JIT is illustrated in the following figure ([5]).<br>Figure 4: JIT signaling technique [5]<br>Let t be the time a setup message arrives at some OBS node along the path to the destination user. As the figure shows, once the setup message arrives at the node, a wavelength is reserved for the upcoming burst and the operation to switch the burst is initiated. When this operation completes at time t + BHP_proc_Del + Switch_Reconf_delay, the switch is ready to carry the burst and the burst actually arrives at the node. Because the channel is reserved from the moment the setup message arrives at the OBS node until the burst exits this node, the end time calculated for the burst, which is also the time that the channel becomes free again is equal to: t+ duration_of_burst + BHP_proc_Del + Switch_Reconf_delay. This is shown clearly in the following illustration where we assume a single channel and scheduled bursts in the time domain.<br>Figure 5: JIT protocol in a single | [redacted]<br>[redacted]<br>[redacted]<br>[redacted] |

| | |
|---|---|
| channel [5]<br>In the JIT protocol the control packet is not aware of the burst length and reserves the relevant link bandwidth (if available) for the entire burst as soon as it arrives at the switch. Studies have shown that JIT performs worse than JET in terms of burst loss probability. | [redacted] |
| 1.4.3 Analytical Delay Model-(JET and JIT)<br>In Just-Enough-Time (JET) and Just-In-Time (JIT), the end-to-end delay is given by the sum of the burst aggregation time, the offset time, the burst transmission time, and the data burst propagation time.<br>Where ta = tttMp + [5]<br>1.5 CONTENTION RESOLUTION<br>Contention will occur if multiple bursts from different input ports are destined for the same output port at the same time. In an OBS network with higher load, contention increases and hence the number of dropped bursts increases, leading to a serious loss of performance. Several methods can be used to lower the burst dropping probability, such as wavelength conversion, deflection routing and optical buffering.<br>In OBS networks, burst loss due to contention is a major issue. The contention of bursts depends on the physical topology and resources available such as the number of wavelengths and the network connectivity and also the burst length and the traffic load. | [redacted] |
| We will have a short discussion for | [redacted] |

| | |
|---|---|
| some critical proposed schemes for contention resolution that exist in literature and then we will focus on the optical buffering and specifically the use of fiber delay lines which is something that we implemented in our network simulator [1], [2].<br>1.5.1 Wavelength Conversion<br>In WMD, several wavelengths run on a fiber link that connects two optical switches. Wavelength conversion is the process of converting the wavelength of an incoming channel to another wavelength at the outgoing channel, if we assume that we have more than one bursts (N) destined to the same output port at the same time. If there was no wavelength conversion, only one burst would be transmitted through the specific output port and the other N-1 would be dropped. Now, using wavelength conversion, all N bursts can be transmitted, but on N different wavelengths. | [redacted] |
| In optical burst switching with wavelength conversion, contentions are reduced by utilizing additional capacity in the form of multiple wavelengths per link. A contenting burst may be switched to any of the available wavelengths on the outgoing link. | [redacted] |
| This solution is still not used due to the high cost and the immaturity of technology. Only a small range of conversions are possible. There are four different wavelength conversion categories: full conversion, limited | [redacted] |

conversion, fixed conversion and sparse wavelength conversion.

### 1.5.2 Deflection Routing

In deflection routing, contention is resolved by routing data to an output port other than the intended output port. This seems to be a very useful technique in optical burst switching networks as there is no optical buffering to keep the contenting bursts until the original path is freed up again.

In deflection routing (or else hot-potato routing), a deflected packet or burst typically takes a longer route to its destination, leading to increased delay and a degradation of the signal quality. Furthermore, it is possible that the packet or burst may loop indefinitely within the network, adding to congestion. Additional mechanisms must be implemented to prevent excessive path lengths. Furthermore, in order to prevent looping for the given network, the nodes at which deflection can occur as well as the options for the deflection port must be limited.

### 1.5.3 Burst Segmentation

In existing optical burst switching approaches, when contention between two bursts cannot be resolved through other means, one of the bursts will be dropped, even though he two bursts may overlap in a very small amount of time. Thus, in some applications, it may be desirable to lose a few packets from a given burst rather than the whole burst. This is what burst

[redacted]

[redacted]

segmentation does.

Burst segmentation minimizes packet losses by partitioning the burst into segments and dropping only those segments which contend with another burst. A significant advantage of burst segmentation is that it allows bursts to be preempted by other bursts. This implies some sort of priorities when handling contentions.

### 1.5.4 Segmentation with Deflection

A basic extension of burst segmentation is to implement segmentation with deflection. Rather than dropping one of the overlapping segments of a burst in contention, we can either deflect the entire contenting burst or deflect the overlapping segments of the burst to an alternative output port other than the intended (original) output port. This technique increases the probability that the burst will reach the destination (performance improvement).

A disadvantage of this method is that a burst may encounter looping in the network or may be deflected multiple times, thereby wasting network bandwidth. This situation will cause increased packet loss especially when we have high loads. Also, due to deflection, the burst may also traverse a longer route, thereby increasing the total processing time.

### 1.5.5 Optical Buffering

In optical networks, since there is no RAM-like buffering available, fiber delay lines (FDLs) can be utilized to delay packets for a fixed amount of

[redacted]

[redacted]

[redacted]

| | |
|---|---|
| time. By implementing multiple delay lines in stages or in parallel, a buffer is created that can hold a burst for a variable amount of time [2]. | [redacted] |
| In any optical buffer architecture, the size of the buffers is severely limited, not only by signal quality concerns, but also by physical space limitations. To delay a single burst for 1 ms requires over 200 km of fiber. Due to the size limitation of optical buffers, a node may be unable to effectively handle high load or bursty traffic conditions.<br>Optical buffers are either single-stage, which have only one block of delay lines, or multistage which have several blocks of delay lines cascaded together, where each block contains a set of parallel delay lines. Optical buffers can be further classified into feed-forward, feedback and hybrid architectures. In feed- forward architecture, each delay line connects an output port of a switching element at a given stage to an input port of another switching element in the next stage. In feedback architecture, each delay line connects an output port of a switch at a given stage to an input port of a switch in the same stage or in previous stage. In hybrid architecture, feed-forward and feedback buffers are combined [1]. | |
| Optical buffers can also perform | [redacted] |

looping in a single fiber for as much time as needed to delay a burst in order it to be transmitted successfully.

According to the position of the buffers, packet switches are essentially categorized into three major configurations: input buffering, output buffering and shared buffering. In input buffering, a set of buffers is dedicated for each input port (we use input buffering at our core node routers). In output buffering, a set of buffers is dedicated for each output port. In shared buffering, a set of buffers can be shared by both input ports and output ports. Input buffering has poor performance due to head-of-line blocking. Output buffering and shared buffering can both perform better. However, output buffering requires a significant number of FDLs as well as larger switch sizes. With shared buffering, on the other hand, all output ports can access the same buffers. Furthermore, buffers can be either configured as degenerate buffer or non-degenerate buffer.

In addition to buffering bursts optically, it is also possible to buffer bursts electronically. Electronic buffering can be accomplished by sending the bursts up to the electronic switching or routing layer. The disadvantage of an approach like this is that each node must have electronic switching or routing capabilities, resulting in higher network costs, more complex architectures and RAMs working under optical

[redacted]

[redacted]

networks speeds.

Delay lines may be acceptable in prototype switches, but are not commercially viable. Furthermore, buffering approaches using FDLs increase the complexity of the optical layer [4], [6], [7].

[redacted]

## 1.6 CHANNEL SCHEDULING check 2

When a burst arrives to a node, it must be assigned a wavelength on the appropriate outgoing link. The primary objective of channel scheduling is to minimize the "voids" in each channel's schedule, where a void is the idle space between two bursts which are transmitted over the same output wavelength.

When a BHP arrives at a core node, a channel scheduling algorithm is used to schedule the coming burst to an outgoing data channel. The scheduler in the core router retrieves from the burst header packet all the needed information for burst scheduling including the burst arrival time and duration of the unscheduled burst. Depending on the algorithm used for scheduling, the latest available unscheduled time (LAUT) may be required to be maintained or easily derived from BHP information. This is because it may need the LAUT to schedule a newly arriving burst. A "gap" is the idle period between a newly arriving burst to be scheduled and an already scheduled burst.

[redacted]

Data channel scheduling algorithms can be broadly classified into two categories: with and without void

[redacted]

filling. These two categories of algorithms differ mainly on the type and amount of information that they need to maintain at a no about a channel. For example, in void filling algorithms, the starting and the ending time for each burst on every data channel must be maintained with other important information something that does not hold for non-void-filling algorithms.

Traditional non-void filling scheduling algorithms are the following:

- First Fit Unscheduled Channel (FFUC)
- Latest Available Unscheduled Channel (LAUC)

Traditional void filling scheduling algorithms are the following:

- First Fit Unscheduled Channel with Void Filling (FFUC-VF)

- Latest Available Unscheduled Channel with Void Filling (LAUC-VF)

We are going to analyze LAUC-VF as we implemented this algorithm in our network simulator [1].

### 1.6.1 Latest Available Unscheduled Channel with Void Filling (LAUC-VF)

The LAUC-VF scheduling algorithm, being one of the most important void-filling algorithms, maintains the starting and ending times for each unscheduled data burst on every data channel. The goal of this algorithm is to utilize voids between two data burst assignments. The criterion to choose

[redacted]

[redacted]

an outgoing channel that will hold the burst is to have a void that minimizes the gab.

It is assumed that we use a WDM link with M different channels. Based on our description of optical buffers in 5.6, it is also assumed that the optical buffer has B FDLs with ith FDL being able to delay Qi time, 1 < ĩ < B. For FDL 0, its delay time is Q0= 0. To simplify the description, we assume Qi = i*D where D is a given time unit. It is further assumed that the Switch_reconf_delay is negligible; hence the data burst arrival time to the optical switching matrix is equal to its departure time if FDL 0 is used.

The goal of LAUC-VF algorithm is to minimize voids by selecting the latest available unused data channel for each arriving data burst. Unscheduled data channels can also be referred as unused data channels. Given the arrival time t of a data burst with duration L to the optical switching matrix (we explain in more detail the architecture of OBS core router in 2.2), the scheduler first finds the outgoing data channels that are available for the time period (t, t+L). If there is at least one such data channel (out of M), the scheduler selects the latest available data channel, i.e., the channel having the smallest gap between t and the end of the last data burst just before t. In order to understand the concept of the algorithm we show an example in

[redacted]

figure 6 which we describe underneath ([2]).

Figure 6: An example of LAUC-VF algorithm [2]

In this example, we have 5 different data channels namely Di,.., D5. The new burst arrives at time t. As we can see from the figure three out of the five channels are eligible unused data channels for the requesting time t and they are able to carry the burst. These channels are D1, D2 and D5. Data channel D3 cannot carry the data burst as the available void is too small comparing with the burst size. Moreover, channel D4 is ineligible to carry the burst as it is busy at time t. Data channel D2 is chosen to carry the data burst as É - ÍJ < í - Í! < í - í5. If this was not the case and all the channels were not able to carry the burst at time t, the scheduler at the core will try to find an available data channel at time t+D[i.e., available for the time period of (t + D, t + D + L)]. If again there is no available channel, the scheduler searches for a channel at time t+2*D[i.e., available for the time period of (t + 2*D, t + 2*D + L)]. If no channels are found eligible up to time t + B*D [i.e., for the time period of (t + B*D, t + B*D+L)], the arriving data burst and the corresponding BHP are dropped. Bursts are dropped when this limit is reached because B*D is the maximum time that a data burst can be buffered in the FDL Buffer [4], [9].

The formal description of the LAUC-VF algorithm is presented below. "available_channel(t)" is a function

[redacted]

[redacted]

which searches for the eligible latest available unused channel (LAUC) at time t (t:time that the data burst arrives at the optical switching matrix) and returns the selected outgoing data channel to carry the data burst if found, otherwise return value -1. We denote as j the outgoing data channel selected to carry the data burst.

Be gi 71 {LAUC — VF algorithìn\
Step 2: j = Avaiỉabỉe_charmeỉ(t);
{seỉected data channel Ị to carry the burst and i the seỉected FDL; stop;}
eỉse

Figure 7: Formal description of LAUC-VF algorithm [2]

We have to mention that an exhaustive search among the FDL stages is used in the LAUC-VF algorithm, thus Qi takes all the possible delay times according to i (FDL stage in optical buffer), vvhere 1 < i < B. In general, for a given time t, the data channels can be classified into unscheduled data channels and scheduled channels. In the first category the channels have no scheduled bursts after t (e.g D5 in the above example) and in the second category the opposite, i.e. there are already scheduled bursts after t (e.g D1, D2, D3 and D4). The above LAUC-VF algorithm as well as the simplified version that we implemented in our network topology do not distinguish between scheduled and unscheduled data channels.

[redacted]

# Chapter 2: Framework of Study

At first we will give an extensive description of the two main elements of our network, edge routers and core routers.

## 2.1 EDGE ROUTERS

The simplified functional architecture of OBS edge routers that we used is shown in figure 8. An edge router consists of two main modules, the EdgeRouting module and the BurstQueue module. IP packets enter the OBS Edge Router via its electronic interfaces. In the EgdeRouting module the packets are sorted based on two criteria: a) destination of OBS Edge Router and b) QoS class of Service. According to the destination, each packet is forwarded to the appropriate (exactly one) BurstQueue module. Within the burst buffer, whereby each of the N sub- queues correspond to exactly one out of N QoS classes supported by the OBS Edge Router. The BurstQueue module performs bursts assembly (time- or length-based or mixture of these two), scheduling of bursts and management of concurrent access to shared channels (This is the general description of the edge router-in our simplified version we use only one shared channel.) [2].

Figure 8: Block diagram of OBS edge router

We use an OBS core that contains 16 BurstQueue modules. Each BurstQueue module holds IP packets with a unique OBS Edge Destination

[redacted]

| | |
|---|---|
| Router. Thus, this node can be used in a network scenario with 16 other OBS edge routers at maximum.<br>The "EdgeRouting" process model handles reception of IP packets and OBS bursts separately. In the former case, it places each received IP packet to the proper BurstQueue module (which is actually a queue). As such, this process handles route lookup (in the OBS domain) of an IP packet, as well as sorting of IP packets according to the OBS Edge Router Destination and potentially also according to class of service (if different class of services are supported by the OBS network). The implication of sorting is that the IP packet is eventually forwarded to the BurstQueue module that is responsible for serving the respective OBS Edge Router Destination. In the case of arrival of bursts, the process simply decapsulates the IP packets and forwards them through the electronic interface to their destination.<br><br>"BurstQueue" is a queue module implementing the functionality of the "Burst Assembler" and "Scheduler" blocks depicted in the previous figure. As such, the implementing process is responsible for:<br><br>a) receiving IP packets/ready-to-go bursts that are destined to remote OBS Edge Router X,<br>b) assembling bursts from waiting IP packets according to the burst assembly algorithm and parameters used | [redacted]<br><br>[redacted] |

| | |
|---|---|
| c) store bursts into assembled burst in burst queues<br>d) schedule the transmission of bursts waiting in the burst queues through the optical interface | [redacted] |
| The BurstQueue module supports a separate queue for each of the service classes supported by the OBS router through the use of "subqueues" functionality supported by the OPNET Modeler queue module data structure. As mentioned earlier each BurstQueue module comprises N (w <5, i.e. five classes of services are supported at maximum in the existing edge router module) subqueues, where N is the number of service classes supported by the Edge Router. | [redacted] |
| In this version of the simulator, a single burst assembly algorithm is implemented, namely the mixed timeout/burst-threshold algorithm: a burst is formed either when the assembly timeout timer (bộ đếm thời gian quá hạn) has expired or when the burst size has for the first time exceeded the burst threshold. When one of the two previous situations occurs, a burst is assembled and ready to travel across the network. We can certainly simulate timeout-only or threshold-only burst assembly techniques by setting extreme values to the burst assembly parameter that is to be excluded from being a criterion for burst forming. Assembled bursts are then queued in OPNET Modeler data structures called "Segmentation and Reassembly (SAR) | [redacted] |

| | |
|---|---|
| Resegmentation Buffers". These are specialized queue data structures that ease the creation of encapsulated packets i.e. packets that comprise of smaller packets encapsulated together under a common header. It is straightforward that there is a 1:1 correspondence between (IP packet) subqueues and (burst) SAR Resegmentation Buffers within a single BurstQueue object. | [redacted] |
| It also attaches a timestamp to each arriving BHP packet, which records the arrival time of the associated data burst to the optical switching matrix. The timestamp is the sum of the BHP arrival time; the burst offset time and the delay of input FDL.According to scheduling policy used, "Scheduler" state of the BurstQueue module peaks a non-empty QoS class of service and pops a burst from the head of the selected subqueue. It attaches a timestamp to the BHP packet which records the arrival time of the associated data burst to the next nodes' optical switching matrix (see details about OSX in 2.2). The timestamp depends on the reservation scheme we use. If we use JET, it is the sum of the BHP arrival time; the switch reconfiguration delay and the burst offset time. The "scheduler" transmits first the related Burst Header Packet and then sends the actual burst (based on the scheduling scheme we use in core routers). | [redacted] |
| 2.2 CORE ROUTERS | [redacted] |

A general diagram of the core router architecture that we use in our network simulator is shown in figure 9. This architecture is also used in [2] where it is related to multiple channels (WDM). In our implementation we use this basic structure but we have not implemented WDM channels. We use one single channel for both the transmission of BHP control messages and data bursts. We are going to describe this general architecture focusing on the modules on which we implemented our scheduling and reservation policies [2].

Channel Mapping Figure 9: General Architecture of OBS core router [2]

We assume that the dimensions if the core router are N x M. We have a WDM link having K channels with k control channels and K-k data channels, L Ị í -V. Furthermore, we have two different channel groups in each direction of the WDM link, the Data Channel Group (DCG) and the Control Channel Group (CCG). Consequently, CDG consists of K-k data channels and CCG of k control channels. In general DCGs' and CCGs' channels can be physically carried either on the same fiber or on different fibers.

As depicted, a core router consists of input fiber delay lines, an optical switching matrix, a switch control unit (SCU), and routing and signaling

[redacted]

[redacted]

[redacted]

processors. Data channels are passed through the Input FDL lines and are connected to the optical switching matrix. The control channels are terminated at the SCU.

The input FDLs are used to delay the arriving bursts for specified amounts of time (different for each FDL stage) in order to allow the SCU to have enough available time to process the associated BHPs and perform other important actions. As we mentioned in the general description of OBS networks, the burst header contains all the necessary routing information. This information is used by the switch control unit at each hop to configure the optical switching matrix to switch the data burst optically. Data bursts are always in the form of optical signals in core routers. The optical buffers of FDLs with the several levels of delay, are used to resolve data burst contentions on outgoing data burst channels. Each optical buffer has B WDM FDLs with ith FDL being able to delay Qi time, l<i<Bễ and it is assumed that Qi<Q2<-- <Qb- In figure 9 each FDL has (K- k) wavelengths. By default there is always an FDL with zero delay time, denoted by 0 with Q0 = 0. The design of optical switching matrices is very important and there are several issues that need to be considered in order to do so, such as implementation complexity, cost, burst loss ratio etc [2].

[redacted]

| | |
|---|---|
| Routing and Signaling processors run routing and other control protocols for the OBS network. Routing processor searches the entire network and based on the information that it gathers, it creates and maintains a routing table. Furthermore, it computes also the forwarding table for the SCU. The SCU looks at this table and decides on which outgoing DCG and CCG to forward each arriving burst and its corresponding BHP. If there is an available CCG and DCG channel for the arriving burst (either for the time the burst arrives at the core node or after a specific delay assigned by a FDL), the SCU directly "locks" the FDL stage required and configures the Optical Switching Matrix to let the data burst pass through. Otherwise, the data burst is dropped.<br>Another situation where the data burst will be dropped is when the data burst enters the optical switching matrix before its corresponding BHP has been processed. It is obvious that this happens because data bursts are optical analog signals and if no path is set up when they enter the optical switching matrix they have no information so they are simply lost. We have to mention that the delay specified by the FDL Input Buffer should be properly engineered such that almost no bursts are lost due to arrival at next optical switching matrix before their BHPs.<br>Now we are going to focus a little bit more on the architecture of the switch control unit as this is where we | [redacted]<br>[redacted] |

implemented our signaling schemes and scheduling algorithms. The block diagram of a switch control unit is depicted in figure 10. We focus on the description of centralized configuration where all the schedulers have the same switch controller.

Figure 10: Block diagram of a Switch Control Unit (SCU) [2]

[redacted]

First we give a short description of SCUs building blocks. The packet processor (PP) performs decapsulation functions. The forwarder performs the forwarding table lookup to decide on which outgoing CCG to forward the BHP and on which outgoing DCG to forward the corresponding data burst. The mapping of logical channels to physical wavelengths is done in the forwarder. Then it simply forwards BHPs across the switch in a certain order.

[redacted]

The scheduler is responsible as the name implies, for scheduling. It schedules both the switch of the data burst on an outgoing data channel and the transmission of its BHP on an outgoing control channel. In this specific module we describe, there is one scheduler for each DCG and CCG pair. Each scheduler needs to keep track of the busy/idle periods of its related outgoing DCG and CCG. The scheduler first reads the timestamp and the data burst duration information from a BHP in order to be aware of the scheduled start time (entering the optical switching matrix) and the corresponding duration of the

[redacted]

| | |
|---|---|
| data burst. It then searches for an available outgoing data channel time slot to carry the data burst, making l use of the FDLs to delay the data burst if necessary. Once an available outgoing DCG channel is found, the scheduler knows exactly the time that the burst will depart from the optical switching matrix. If so, it schedules the time to send out the BHP on the outgoing CCG, trying to resynchronize the BHP with the data burst. After successful scheduling of the transfer of the burst, the scheduler will send the configuration information to the switch controller which will in turn configure immediately the optical switching matrix properly to let the data burst pass through. The configuration information includes incoming data channel identifier, outgoing data channel identifier, time to switch the data burst, duration of the data burst and the FDL buffer stage identifier.<br>As we mentioned earlier all schedulers have the same switch controller. Each scheduler is bidirectionally connected to the switch controller. So the scheduler waits for the switch controllers respond. If the configuration information processing at the switch controller is successful an acknowledgement is sent back to the scheduler. The scheduler then first updates the state information of the DCG and CCG. It then modifies the BHP information (including updating offset time for the next core node) and passes it along with the time-to-send | [redacted]<br>[redacted]<br>[redacted]<br>[redacted] |

BHP information to the BHP transmission (Tx) module. It is now ready to process the next BHP.

[redacted]

We conclude with some assumptions that we made for our network model regarding the core routers architecture. First we assume one single channel thus; data bursts and BHPs are carried on the same channel. An illustration of burst transmission in our OBS network is shown in 11. Bursts do not content with BHPs as BHPs are very small comparing to data bursts. So there is no problem regarding BHP and data burst contentions. Each FDL has one wavelength and by default we also use the FDL with zero delay time. Moreover, in our simple implementation we use one single scheduler that performs the actions described above. We have also a state that occurs whenever a burst is transmitted. This state is called "Update Switching Table" and manages the concurrent access of the various schedulers to the channel, releasing the "locks" from the FDLs when they are not occupied anymore and deallocates the already used memory which is not required anymore from the optical switching matrix.

[redacted]

## 2.3 SIMULATION FRAMEWORK-IMPLEMENTATION check 1

All the implementation of our work is done in OPNET Modeler. OPNET Modeler is a commercial network simulator which is used to analyze the

[redacted]

| | |
|---|---|
| behavior and performance of communication networks by performing discrete event simulations. Using OPNET, one can design models, execute simulations, collect and analyze data from the simulation outputs.<br>We use OPNET to simulate networks at their lowest level of hierarchy. Thus, a typical such network comprises of only nodes and links. Some examples of node models that we also used in our implementation are workstations, switches, routers and servers. Node models consist of smaller building blocks called modules. | [redacted] |
| Each module is a finite state machine (FSM) which is used to represent the logic and the behavior of the model. It consists of various states and transitions between them according to each incoming event. Modules are connected through packet streams, statistic wires and logical associations.<br>In this study, we developed some new components and combined them with the existing components in the network architecture. Furthermore, we implemented some parts of the already existing components that were not supported beforehand. To be more specific, this part of our work is implemented by writing some C/C++ code to some parts of the FSM of the node model "Core Router", basic model unit of the core router architecture. | [redacted] |
| We will describe the network topology | [redacted] |

that we worked on as well as some OBS edge and core nodes characteristics. Following this, we will analyze the three different scenarios that we implemented.

### 2.3.1 Network Topology

The topology of the OBS network that we worked on is depicted in the following figure. The original topology had only two edge stations that were used to send and receive traffic. In our scenario we added four edge routers more in order to test the parallel sending and delivery of bursts from and to several end systems. We also added some VoIP and FTP workstations as well as some ftpservers.

Figure 12: Network Topology

Following is a list containing all the implementation model names that we used from the Object Palette menu of the OPNET modeler with a short description for each model.

Table 1: Implementation models

OBS edge and core routers are connected between each other with fiber links employing 16 wavelengths at 2,5Gbps each. Although in our implementation we used only one wavelength, we use this link in order for the network to be compatible with WDM for future integration. IP routers are connected to their paired OBS edge routers via bidirectional PPP link. We used Ethernet switches, Ethernet workstations and servers for

[redacted]

[redacted]

[redacted]

[redacted]

| | |
|---|---|
| creation of IP traffic that will flow through the OBS network. Each OBS edge node of this model can connect to exactly one IP router and at most two OBS core routers. Some OBS edge and core routers characteristics are mentioned in the following tables: | [redacted] |
| OBS EDGE ROUTER CHARACTERISTICS<br>Parameter Value Description<br>OBStoIPtxStrm 15 | [redacted] |
| Index of the transmitter leading from the Edge Routing Processor to the IP router connected to the IP Edge Router.<br>Burst Assembly Parameters<br>2 Number of supported QoS classes | [redacted] |
| Scheduling policy FCFC (First-Come-First- Served)<br>Scheduling scheme used for both the forwarding of IP packets to burst assemblers as well as the forwarding of deburstified IP packets to the electronic transmitter. | [redacted] |
| Burst Scheduling MethodOldest Burst First Method to be used to schedule bursts among burst queues of different class of services.<br>BHP Offset Time 0.00018<br>Timeout offset (in seconds) between a BHP and a data burst.<br>Burst Assembly Method Timer-based and threshold based policies<br>We use this mixed policy to create | [redacted] |

bursts and send them to the scheduler in order to be scheduled.

Table 2: OBS Edge Router Characteristics

[redacted]

OBS CORE ROUTER CHARACTERISTICS

Parameter Value Description

BHP Processing Delay promoted Time it takes for the setup message to be processed in each core router

[redacted]

Switch

Reconfiguration

Delay promoted Time it takes for the OXC configuration in order to be ready to carry the burst.

[redacted]

Routing Scheme Single Shortest Path (Hop-based-static)

To determine the path from source to destination.

Table 3: OBS Core Router Characteristics

[redacted]

Regarding routing we have two different planes, routing in the OBS domain and routing in the IP domain. Computation of static routing tables in the OBS domain is done automatically through implementation in the respective node models (already implemented). However, routing in the IP domain is still handled manually. When we say routing in the IP domain we mean routing between end systems and IP access routers. In order to do this, after assigning IP addresses to all end systems and routers we filled up the static routing tables of all IP access routers by hand. In each static routing

[redacted]

table we included the IP addresses of all end systems of the network, the IP address assigned to the interface of each IP router that is connected with each paired OBS edge router and the IP address assigned to the interface of each IP router that is used to reach specific end systems electronically.

We also added configuration objects to our network ("Application Definition" and "Profile Definition" objects from object palette) to specify the application traffic that will exist on the network. We created two applications, FTP and VoIP and two application profiles, FTPUser and VoIPUser respectively. In figures 13 and 14 we give the description characteristics of each application definition. We assign two Type Of Service (TOS) values for the applications which are equal to 0 for the FTP application and 192 for the VoIP application. The TOS value of each application have to match the TOS attribute value of the OBS Edge Router forwarding class we use to carry flows of this application in the OBS domain. In the Profile Configuration we created two VoIPUser profiles (VoIPUser, VoipUserl) and three FTPUser profiles (FTPUser, FTPUser1, FTPUser2). These profiles have similar characteristics except from the fact that they all have different start times. To finish this procedure, we added traffic in each workstation by modifying the attributes "Applications: Supported Profiles" and "Applications: Destination Preferences" of the end systems accordingly.

[redacted]

Figure 13: Description characteristics of FTP

[redacted]

Now that we have specified the network, traffic and statistics to collect we run our simulation with three different scenarios in order to compare the different scheduling policies regarding burst loss and delay.

Our basic implementation is focused on two modules of the switch control unit (figure 10) namely the "Scheduler" and the "Switch Controller". It is very important to mention that for the purposes of our study we assumed that we only have one channel for the transferring of bursts and their corresponding BHPs (no WDM implementation). We created three different scenarios each implementing a different reservation and scheduling technique. We describe the three different scenarios below:

[redacted]

### 2.3.2 Scenario 1: JET

The original simulator did not support any reservation scheme, or any scheduling policy. It had something simple implemented just to forward the bursts to the next hop without any concerns about the reservation of the resources (channel) or the scheduling

[redacted]

| | |
|---|---|
| method followed. So we started from that. We implemented JET reservation scheme, as stated in literature and as explained above (1.4.2), calculating correctly the desirable times: start time of the burst handling, duration and corresponding end time, for each burst at each core node of the OBS network.<br>As mentioned in 1.4.2, the offset time which is a very important factor for the implementation of JET is calculated from the formula: offset_time = h*BHP_proc_Del +Switch_Reconf_delay, where h is the number of hops between the source and the destination, BHP_proc_Del is the per-hop burst header processing time, and Switch_Reconf_delay is the switching reconfiguration time. The offset time in an essential issue in each optical burst switching network. For our implementation we used a fixed h (equal to 7) as we use routing scheme Single Shortest Path and we use average length_of_path = h for our calculations. This could be also a variable calculated explicitly for each path. We observed that we had to do a parametric study for offset time (i.e. for BHP processing delay and switch reconfiguration delay), based on our network characteristics. The offset time must be compatible each time with our network characteristics (modules, scheduling and reservation policies etc.). Thus, in order to derive the optimum value of the offset time | [redacted]<br>[redacted] |

for the corresponding network scenario, we performed a parametric study using the "DES Parametric Study tool" in OPNET Modeler. So we run some simulations with a range of values different for the BHP processing delay and the switch reconfiguration delay which are shown in the following table:

Table 4: Range of values for parameterization of offset time

We used a list, different for each core router, in order to store there a data structure representing all the important information about each data burst as: burst id (unique for each burst), ingress router address, egress router address, output port, wavelength (zero for our simulation), start time, end time and duration. This list is sorted based on start times. This is very important because we need to check whether the channel is available at the time that a burst wants to reach this core node. Our scheduling policy regarding JET reservation scheme is actually a search in the whole list of elements in order to find if the wavelength is available at the specific time intervals. Based on the outcome of this scheduling policy, we either forward or drop the burst.

Following the reservation scheme, if the reservation is successful we schedule an event at the core router

[redacted]

after the burst and its corresponding BHP leave the router and move towards the next hop router. This event actually searches the list with the scheduled bursts, removes the correct element (of already serviced burst) and deallocates its memory.

Because we have bursts coming from different nodes and going to different destinations, we are waiting to detect situations that we have burst contention and thus burst loss. We will analyze this at the results section.

[redacted]

### 2.3.3 Scenario 2: JIT

Following the theoretical details mentioned in 4.2.2 we implemented also the JIT reservation scheme in our OBS core nodes. Following the previous description for the JET scenario, we used a similar scheduling policy in order to check if the channel is available for each corresponding burst we have, in order to send the burst to the next node.

Regarding the optimum values for BHP processing delay and switch reconfiguration delay we used the optimum values that we computed from the parametric study in Scenario 1.

The sorted list we used to store the scheduled elements remains the same as before. We also use the same technique for the removal of the element from the list after the corresponding burst leaves the commensurate core node.

[redacted]

### 2.3.4 Scenario 3: LAUC-VF

[redacted]

In order to implement the LAUC-VF algorithm we first implemented the FDL buffers in our network. AS mentioned in the background section, FDLs are very important as they can be used in order to delay a burst for a specific amount of time in order for it to be scheduled normally. Several are the cases when the start time of a burst is slightly smaller than the end time of the previously scheduled burst. This is where FDLs come to solve the contention by delaying the arriving burst for some time in order not to content with the previous one.

In our implementation each core node has six different FDL stages (including the default 0 FDL stage) and a burst can use the FDL stage of its interest only if it is not already occupied by a previously scheduled burst. This explicit use of the fdl stages is implemented with the use of mutexes. We assume that we have the technology required to delay the bursts for the times stated in the scenario. The delays assigned at each FDL stage are shown in the following table:

Table 5: Delays assigned at the FDL buffer stages

Following the parametric analysis for the optimum values of BHP processing delay and switch reconfiguration delay that we discussed in scenariol, now that we have a different scenario with fiber delay lines we need to do again the parameter study in order to find the

[redacted]

| | |
|---|---|
| new optimum values. The range of values we used for this study remains the same as in JET but the optimum values change.<br>After implemented and tested the FDL mechanism we implemented the LAUC-VF algorithm using the times we have calculated by applying the JET reservation mechanism (start time, duration and end time). In 1.6.1 we described the LAUC-VF algorithm assuming that we have WDM links. Now that we have one single link (channel) we modified slightly the original algorithm in order to fit it in our network simulation. | [redacted] |
| Figure 15 depicts the single channel of our core network topology with some already scheduled bursts in it. In this version of the LAUC-VF algorithm we search in this channel whether the newly arrived burst fits in any available void with or without the use of the Fiber Delay Lines. As the original version of LAUC- VF that we described in 1.6.1 an extensive search is performed in all the stages of the optical buffer to check whether there is an available void for the newly arriving burst to be scheduled. With the way we implemented the reservation technique for the single channel, it is very easy for someone in the future to integrate this method-algorithm to support WDM network.<br>Example of scheduled voids in a single channel | [redacted] |

| | |
|---|---|
| Chapter 3: Results Conclusions<br>3.1 Results check 1<br>Before we actually show the results of our study and the conclusions derived from them, we sum up the assumptions that we used in our network scenario: | [redacted] |
| ❖ We use one single channel for the transmission of both the Burst Header Packet and the data burst (Single wavelength (0)). | [redacted] |
| ❖ We assume that we have the corresponding technology of fiber delay lines to accommodate the delay times we defined. | [redacted] |
| ❖ The values for the switch reconfiguration delay and the BHP processing delay we used are the optimum (-thus optimum offset time-) values (within a specific range) that we computed during our parametric studies. | [redacted] |
| ❖ Despite the fact that we use optical links of 2,5 Gbps, we make the assumption that only 500 Kbps are available for our services in the network. The rest are occupied from other services that are not of our interest. Thus our data rate is equal to 500 Kbps. | [redacted] |
| ❖ No wavelength conversion at nodes. | [redacted] |
| ❖ Retransmission of lost bursts is not considered. | [redacted] |

| | |
|---|---|
| ❖ All the times, regarding arriving time of new bursts, duration etc. are calculated in msec.<br>❖ The number of hops between the source and the destination is considered being constant in this scenario and specifically equal to 7. | [redacted] |
| The focus of our study is on Burst Loss Ratio due to data channel congestion as well as on End-to-End Delay which are two very important statistics regarding networks performance. | [redacted] |
| We separate our results in three sections which contain parametric studies for scenario1 and scenario3, performance of JET, JIT and LAU-VF using the optimum offset values and estimation of end to end delay and burst loss ratio versus increasing load.<br>The global statistics that we used from the "Choose Individual DES Statistics" box are OBS. Burst Loss Ratio (Global).time average which is the percentage of BHPs dropped in the entire network, throughout the duration of the simulation, and Ethernet Delay (sec).time average which represents the end to end delay of all packets received by all the stations. | [redacted] |
| 3.1.1 Parametric Studies<br>Based on the calculation (depending on our available data rate) of the burst duration we picked specific ranges for the corresponding BHP processing delay and switch reconfiguration delay | [redacted] |

(2.3.2) which are mentioned again in the following table. We also cite the other parameters that we used in the Configure/Run Discrete Event Simulation (DES) box:

| Attribute | Value |
|---|---|
| Common | |
| Duration | 3 minutes |
| Seed | 128 |
| Values per Statistic | 100 |
| Update Interval | 50000000 events |
| Inouts | |
| Global Attributes-> | |
| Traffic: | |
| Traffic Scaling Factor | 1.0 |
| Traffic Scaling Mode | Only Background traffic |
| Object Attributes-> | |
| Switch Reconfiguration Delay JET/LAUC-VF: | 0.01 - 0.4 by 0.1 |
| BHP Processing Delay | 0.001 - 0.5 by 0.090 |

Table 6: Simulation parameters for parametric studies

We performed first a parametric study in OPNET in order to find the optimum values for both the BHP processing delay and switch reconfiguration delay, i.e. the optimum offset time. Parametric studies in OPNET allow us to take results for the same statistic across multiple simulations. Thus, across these 24 runs we derived the following graphs for Burst Loss Ratio (global statistic) and End-to-End delay in the

[redacted]

[redacted]

[redacted]

network. #of experiment depicts the # of the specific run and then from the "Preview Simulation Values" box of OPNET we derive the specific values for BHP processing delay and switch reconfiguration delay. Optimum values for the Switch Reconfiguration Delay and the Switch Reconfiguration delay in JIT will be the same derived from the parametric study in JET. Moreover, we follow the same procedure to perform a parametric study using LAUC-VF because it that case we use fiber delay lines (FDLs) and thus optimum value for the offset time will be different than JET, JIT.

Figure 16: Burst Loss Ratio (parametric study-JET)

Figure 17: End-to-End Delay (parametric study-JET)

As we can see from the graphs the optimum values for both attributes (in this specific range of values) were noted at the 21st "run" of the simulation which corresponds to:

BHP Processing Delay = 0.451 ms

Switch Reconfiguration Delay = 0.01 ms

From these values we derive the optimum offset time for the existing topology: Offset time = length_of_path *BHP_processing_delay + switch_reconfiguration_delay = 3.167 ms

Scenario 3: LAUC-VF

■ OBS.Burst Loss Ratio (Global) .time average

[redacted]

0.40 0.35 0.30 0.25 0.20 0.15 0.10 0.05

Experiment #

Figure 18: Burst Loss Ratio (parametric study -LAUC-VF)

Figure 19: End-to-End Delay (parametric study- LAUC-VF)

As we can see from the graphs the optimum values for both attributes (in this specific range of values) were noted at the 6th "run" of the simulation which corresponds to:

BHP Processing Delay = 0.091 ms Switch Reconfiguration Delay = 0.11 ms

From these values we derive the optimum offset time for the existing topology:

Offset time = length_of_path *BHP_processing_delay + switch_reconfiguration_delay = 0.747 ms

As we observed from the above corresponding graphs of both JET and LAUC-VF implementation the offset time is a very special characteristic of our network. In the parametric study we were increasing each time the constituents of the offset time and as we can see from the burst loss ratio graph or the end- to-end delay graph, there is no specific pattern for the optimum offset meaning that there is no breaking point after which we would have worse burst loss ratio or even end-to-end delay. On the contrary we have variations in both statistics that we measure.

Studies have shown that in general

[redacted]

[redacted]

offset time is a very important factor and its optimum value is very difficult to be determined. In general the offset time is network based, which means that its value depends on the whole network. Other parameters that affect offset time calculation are burst length parameters, optical router size, size of data channels, size of control channels, processing speed of the scheduler in the SCU, processing speed of the switch controller in the SCU, burstification process, size of the network etc. All these parameters should be carefully considered in the network design.

In real-life networks someone could implement dynamic offset time setup in order to achieve better performance.

### 3.1.2 Performance with optimum values

We attach the parameters that we used in the Configure/Run Discrete Event Simulation (DES) box:

Table 7: Simulation Parameters for optimum performance

Applying the optimum values for each scenario we get the following graphs regarding burst loss ratio and end-to-end delay.

Figure 21: End-to-End Delay (JET)

Figure 23: End-to-End Delay (JIT)

Figure 25: End-to-End Delay (LAUC-VF)

From the above graph we can

[redacted]

conclude that JIT performs worse in terms of burst loss ratio both from JET and from LAUC-VF. Additionally, LAUC-VF performs much better than JET and this is how we can understand the benefit of having fiber delay lines used for optical buffering in our core routers. These results come to verify the already proven performance of those three in literature. Thus, LAUC-VF can reduce the burst loss in a network which is actually the goal of the network designer. Regarding End-to-End Delay we can see from the graphs that it is constant throughout the duration of the experiment.

### 3.1.3 Performance with increasing loading conditions

Parametric studies also examine results for the same statistic from different scenarios corresponding to different traffic growth simulation. After we found the optimum values for BHP processing delay and switch reconfiguration delay for JET and LAUC-VF, we used these values and run simulations with duplicate scenarios (adding more traffic load - both explicit traffic and background traffic) in order to derive some statistics about burst loss ratio versus load and end-to-end delay versus load. To be more specific, we used the same parameters as stated in Table 7, with the only difference that in each different scenario we were changing the "Traffic Scaling Factor" attribute of the simulation console and the

[redacted]

values we used are included in table 8:

Table 8: "Traffic Scaling Factor" values

Consequently, as traffic in the network increases the behavior of both burst loss ratio and end-to-end delay for the different scenarios are depicted below. We have merged the three Burst Loss Ratio graphs into one in order to compare them.

End-to-End Delay

■ Elhernet.Deley (secXtiirie average

1.00450

Figure 27: End-to-End Delay vs. Load (JIT)

Figure 28: End-to-End Delay vs. Load (LAUC-VF)

Figure 29: Burst Loss Ratio vs. Load

What we observe is that End-to-End Delay is more sensitive regarding load than Burst Loss Ratio. From the figure 26 for JET we can see that after experiment #3 which corresponds to "Traffic Scaling Factor" of the simulation console equal to 9.0, End-to-End Delay starts growing rapidly. Thus, this point is a breakpoint for JET. From figure 27 for JIT we can see that the corresponding breakpoint is at experiment#4 which corresponds to "Traffic Scaling Factor" equal to 13.0. Moreover a similar pattern exists for End-to-End Delay for LAUC-VF (figure 28) with breakpoint at experiment #9 which corresponds to "Traffic Scaling Factor" equal to 37.0. Thus, in the given load scale that we

[redacted]

took our experiments we can see that for the three scenarios we have a breakpoint after which the End-to-End Delay starts growing rapidly. The same does not imply for Burst Loss Ratio as we can see in figure 29. Burst Loss Ratio has variations and the difference is very small from its scenario optimum value. So there is no breakpoint for Burst Loss Ratio within the specific load scale after which the statistic starts growing rapidly. This means that Burst Loss Ratio is less sensitive regarding increasing load conditions and retains approximately the optimum burst loss ratio for a large scale of load conditions. Probably if we could extend our traffic loading conditions at some point we would find a breakpoint for this statistic for all three scenarios. The reason that we used this time scale is because as we saw in the graphs at some points in this range we have breakpoints for End-to-End Delay for all three scenarios so we did not want high End-to-End delay. Thus, keeping End-to-End Delay low was our metric for determining the loading scale to perform our performance evaluation.

## 3.2 CONCLUSIONS

Some of the conclusions derived out of this study are mentioned in this section.

- During the last decade, the evolution of emerging applications

[redacted]

| | | |
|---|---|---|
| with high throughput requirements is constantly raising the demand for high-bandwidth network technologies.<br>• Optical Burst Switching technique is a very promising solution providing more efficient utilization under increasing bursty traffic conditions.<br>• There are several issues that need to be resolved before Optical Burst Switching networks can be applied successfully in real-life networks. Most important are burst assembly, signaling schemes, contention resolution, burst scheduling and quality of service.<br>• The hybrid type of burst assembly (both timer based and threshold based) is the best as in this case the burst generation is more flexible.<br>• Just-In-Time (JIT) signaling scheme is characterized by immediate wavelength reservation where Just-Enough-Time (JET) is characterized by delayed wavelength reservation.<br>• Offset time is a very special characteristic and it is basically network based. There are several parameters that need to be considered regarding estimation of offset -ti me in a specific network and some of these are burst length parameters, optical router size, size of channels, processing speed of the scheduler in SCU, processing speed of the switch controller in the SCU, burstification process and size of the network. | | [redacted] |

| | |
|---|---|
| • A lot of research has been made in the field of contention resolution in order to minimize the burst loss ratio. Some techniques that have been proposed are wavelength conversion, deflection routing, burst segmentation, segmentation with deflection and optical buffering.<br>• Optical buffering is a very promising solution because using fiber delay lines (FDLs) we can be able to delay a burst for a specific amount of time in order for the outgoing wavelength to be available to carry the burst. However optical buffer size is limited due to signal quality concerns and physical space limitations.<br>• Channel Scheduling algorithms have the objective to minimize the voids in each channels schedule. There are two categories algorithms with void filling and algorithms without void filling with most known LAUC-VF, FFUC- VF and LAUC, FFUC.<br>• JIT performs worse in terms of burst loss ratio both from JET and from LAUC-VF.<br>• LAUC-VF performs much better than JET. This shows us the benefit of having fiber delay lines. Consequently LAUC-VF can reduce the burst loss ratio in the network.<br>• Regarding performance of JET, JIT and LAUC-VF against increasing loading conditions End-to-End Delay is more sensitive than Burst Loss Ratio. Actually BLR retains its optimum value for a large scale of loading conditions whether End-to- | [redacted] |

| | |
|---|---|
| End Delay after a specific point starts increasing rapidly.<br>Chapter 4: Future Work Extensions<br><br>Through the end of this study several ideas come along for future integration of the existing topology and research in the optical burst switching domain. These are mentioned below:<br><br>❖ Following the conclusions in 3.2.1 one very important aspect would be for someone to perform a dynamic offset setup which will unconditionally provide better performance.<br>To be more specific, someone could implement a protocol which would collect some specific traffic statistics and based on recent history of the network should determine a value for the offset time that would probably have the best performance.<br>This would be a very important solution to the offset time specification problem especially in real-life networks.<br>❖ Implementation of WDM technology supporting multiple wavelengths.<br>❖ Extension of LAUC-VF algorithm in order to be performed in WDM technology. It will be very interesting to see the differences in burst loss ratio when we increase the number of data channels.<br>❖ Deflection routing is also implemented in the existing network topology. Thus a good idea would be | [redacted]<br><br>[redacted]<br><br>[redacted]<br><br>[redacted]<br><br>[redacted] |

| | |
|---|---|
| to evaluate the scheduling performance of our schemes when we have k multiple paths.<br>❖ Variable number of hops from source to destination^ difference in optimum offset time values.<br>❖ Perform also some BHP scheduling in OBS core nodes. Then combine LAUC-VF with the most promising BHP scheduling algorithms and evaluate this combination in terms of burst loss ratio and end-to-end delay.<br>❖ It would be interesting to evaluate the performance of reservation and scheduling policies if the network implies differentiation in services and thus different QoS parameters. | |