HIGH SPEED MULTI-CHANNEL OPTICAL ROUTER DESIGN IN DENSE WAVELENGTH DIVISION MULTIPLEXING (DWDM) OPTICAL NETWORKS

A Dissertation

Presented to

the Faculty of the Department of Electrical and Computer Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in Electrical Engineering

by

Wenhao Chen

August 2014

HIGH SPEED MULTI-CHANNEL OPTICAL ROUTER DESIGN IN DENSE WAVELENGTH DIVISION MULTIPLEXING (DWDM) OPTICAL NETWORKS

Wenhao Chen

Approved:

Chair of the Committee
Dr. Yuhua Chen, Associate Professor,
Electrical and Computer Engineering

Committee Members:

Dr. E. Joe Charlson, Professor,
Electrical and Computer Engineering

Dr. Pauline Markenscoff,
Associate Professor,
Electrical and Computer Engineering

Dr. Cumaraswamy Vipulanandan,
Professor,
Civil and Environmental Engineering

Dr. Jaspal Subhlok,
Professor,
Computer Science

Dr. Suresh K. Khator,
Associate Dean,
Cullen College of Engineering

Dr. Badri Roysam,
Professor and Department Chair,
Electrical and Computer Engineering

Acknowledgements

First of all, I would like to thank my advisor, Dr. Yuhua Chen, who has been advising me in researches and experiments all these years towards my Ph.D. career. I would also like to thank her for putting great effort in my dissertation. During these four years in Systems Research Lab, Dr. Chen's professional attitude towards research as well as academic work has influenced me and her unique insight has helped me a lot in my Ph.D. study.

Secondly, I would like to thank my parents for raising me and bringing me to excellence. I could not have fulfilled my academic goals without their support and understanding along the way. Third, I would like to thank my wife who accompanies me and loves me unconditionally. Without comfort and advise from her during difficult times, I could not have gone this far today. Also, I would like to thank all of my lab members for their generous help and for creating a pleasant lab environment.

HIGH SPEED MULTI-CHANNEL OPTICAL ROUTER DESIGN IN DENSE WAVELENGTH DIVISION MULTIPLEXING (DWDM) OPTICAL NETWORKS

An Abstract

of a

Dissertation

Presented to

the Faculty of the Department of Electrical and Computer Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in Electrical Engineering

by

Wenhao Chen

August 2014

Abstract

In the era of information explosions, the size and complexity of data is expanding dramatically. To meet this requirement, lots of efforts have been made in both the time domain and the frequency domain. In the wireless communication area, the prevalent Time-Division Multiple-Access (TDMA) used in second generation global system for mobile communications (GSM) network is a good example of manipulating the signal in the time domain in order to share bandwidth over time. Orthogonal frequency-division multiplexing (OFDM) on the other hand, utilizes orthogonal sub-carrier signals to carry data on parallel data channels, achieving high spectrum efficiency. On the optical link side, the Dense Wavelength Division Multiplexing (DWDM) is a promising solution to meet the requirement. By multiplexing different carrier wavelengths onto one single strand of fiber, the link bandwidth can increase exponentially.

But the all optical DWDM network is hard to utilize due to technology limitations. This dissertation aims at solving the current limitations on all-optical network from both switching technology and network architecture aspects. From the perspective of switching technologies, the DWDM Multi-Mode router provides an integrated platform to support three different switching technologies simultaneously. The dynamic reconfiguration capability in DWDM Multi-Mode enables the bandwidth sharing among three switching methods which increases the channel utilization. From the perspective of the applications, the Application-Aware (A^2) optical network features the reverse data path reservation is a good candidate of asymmetric traffic transmission. By creating alternative switching technique towards optical switching network, the A^2 optical scheduler eliminates the setup latency problem in traditional optical router. At the same time, the path reservation can be changed in real-time, increasing the probability of packets delivery.

The 3-D switching opened another dimension in optical network to reduce traffic blocking. A dynamical resource allocation scheme is proposed to assign bandwidth for different traffic flows. The hardware experiments showed the feasibility of the proposed 3-D switching and it is expected to serve as a building block of future optical networks.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

Since the birth of Internet, Internet has undergone through a huge expansion worldwide. From the very beginning, the Internet was very sparse due to the equipment limitations and political issues. However, a campaign in 1982 to increase the awareness of information technology in UK lasted for 12 months and eventually it marked the acceptance of IT to general public. At the same time, a standardized Internet protocol Transmission Control Protocol/Internet Protocol (TCP/IP) came out in 1982 and it represents the introduction of the Internet.

The data transmission traffic has expanded 45-50 percent a year since the debut of the Internet. Since 1982, the speed of wide area network (WAN) modems have increased from 64 Kbps to 200 Mbps; the speed of Ethernet local area networks (LAN) have reached 100 Gbps versus 10 Mbps in 1981 and mobile data reaches the speed of 50Mbs which is forty thousand times faster than the speed in 1981. On one hand, the development of manufacturing technology has almost driven the device speed increase exponentially each year. Thanks to the rapid development of very large scale integrated circuit (VLSI), the PCs and mobile devices such as tablet, smart phone have become necessities in daily lives and they stimulate the development of more and more multimedia applications to enrich people’s life. Nowadays, people are more and more likely to use Internet to stream videos, movies, to make phone calls through Voice over IP (VoIP), and to play online games with friends. However, these network applications running on their devices require tremendous bandwidth and they have brought tremendous traffic load into the network. Global Internet usage through mobile device is growing remarkably. As shown in statistics, the number of global mobile users in 2012 was 1.3 billion versus 1.5 billion desktop users. But this 10% gap is filled in merely two years: In 2014, mobile devices accounted for 55% of Internet usage in the United States in January. Applications occupy 47% Internet bandwidth while 8% traffic comes from mobile browsers. Furthermore, the Internet faces new bandwidth challenges toward supporting the prevalent cloud computing and distributed data processing technology since these services requires high bandwidth in order to corporate with others and update the database frequently.

To support novel applications and accommodate services on the network side, service providers try to accommodate existing equipment and configurations while achieving high traffic throughput. This can be done by upgrading existing devices, upgrading backbone networks as well as adding more flexible switches and advanced traffic management schemes. However, any performance enhancement needs to take cost into consideration and the balance between performance enhancement and cost becomes a major issue which the service providers would concern. On the client's side, various applications and even sessions within applications require different latency and jitter performance. It is an intuitive idea to put different traffic streams that have different priorities onto different transmission lines, but these lines have to have the capability of fast dynamic configuration to better serve end users in order to achieve Quality of Service (QoS) guarantee.

DWDM technology multiplexes a number of optical carrier signals onto a single optical fiber by using different wavelengths of laser light and this technique enables massive bandwidth expansion over one strand of fiber. The DWDM allows hundreds of wavelengths to carry signal each at 100Gbps which is considered an ultimate solution for backbone networks. However, to build an agile network and to provide flexibility in traffic manipulation and management, DWDM requires sophisticated switching and routing schemes between edges. The vendor needs switching technologies to decouple wavelength at certain nodes and to couple information onto other wavelengths. By doing this, wavelength can be dropped and added at any node, which achieves the purpose of exchanging information.

Currently there are three switching technologies existed in the core routers. They are Optical Circuit Switching (OCS), Electrical Packet Switching (EPS) and Optical Burst Switching (OBS). In optical circuit switching, before sending the data message, a path is explicitly set up between the source and the destination. This mechanism has the distinct advantage of guaranteed bandwidth. However, it suffers from severe bandwidth underutilization in the absence of data between the source and destination pairs. In order to address the inefficient resource utilization problem in OCS, EPS breaks down messages into packets and core routers decide the next hop these packets would go through. This technology deploys statistical multiplexing and it makes full use of the bandwidth. Furthermore, the in-sequence arrival of packets is not required in EPS. Although similar concepts can be used to realize optical packet switching, an all optical packet switched network is very hard to utilize for two reasons. For one thing, the core router needs to decode the optical packet header and then decides the next hop of the packet. This has to be implemented in the electrical domain which indicates the signal has to be converted from the optical signal into the electrical signal at each ingress port, and it needs to be converted back to the optical domain at the egress port after the packet is processed. The expensive optical to electrical (O/E) and electrical to optical (E/O) converters are not scalable to be massively deployed in DWDM networks considering the exorbitant price service providers would have to pay. Moreover, optical packet payload needs to be buffered optically while the packet header is being processed. As a result, all optical packet switching is not commercially viable with the current technology.

OBS on the other hand is considered a prominent candidate towards future optical network. In OBS, packets are aggregated into the burst based on destinations, and forwarded on DWDM links. There is no O/E/O (optical to electrical and electrical to optical) conversion at each hop. However, traditional OBS lacks of packet level traffic management in the sense that the traffic is groomed according to the destination address based on the first come first serve criteria. Hence, packets lost their own identity after the data burst is formed. To maintain the packet identity observable is a big challenge in traffic management. Traditionally, adjusting the burst assembling time and the burst length would mitigate the priority issue to some extent. However, in packets with different QoS levels that have the same destination scenario, this mechanism will have limited influence on the jitter management.

This dissertation first analyzes the advantages and disadvantages of the above mentioned switching technologies. Based on the investigations, new network architecture is proposed and discussed to accommodate the fast growing Internet. In Chapter 2, an overview of optical communication systems is provided including DWDM optical switching technologies, three different optical switching types (EPS/OCS/OPS) and their specific signaling in the optical network. A detailed literature review of OBS network such as the burst assembly schemes and burst scheduling mechanisms are discussed. The

Multi-Mode optical switching network which provides a unified approach towards heterogeneous traffic is also discussed in Chapter 2, with illustrations of the function of Multi-Mode router to demonstrate the concurrent support of different switching modes.

The Application-Aware (A^2) Dynamic Optical Switching is proposed in Chapter 3 to address the problem of asymmetric traffic flow in optical network systems. Furthermore, the architecture and scheduler design of the A^2 Switch are demonstrated to show the complete network design. In addition to software simulation on the performance of the proposed A^2 optical switch, the hardware testbed verification from the aspect of signal attenuation and optical path setup are also provided in Chapter 3 to demonstrate the feasibility of the proposed A^2 optical switching.

3-D optical switch architecture is proposed in Chapter 4 to address the traffic blocking issue while maintaining high throughput by adding another dimension to existing optical switching network. This proposed technique changes the optical network fundamentally and the traffic block probability is expected to decrease through the utilization of the 3-D switching architecture. Two variations toward the implementation of the 3-D Optical Switching, namely, Static Elevator Switching and Dynamic Elevator Switching, are also discussed in Chapter 4. The algorithms used at the core routers and edge routers are discussed, along with formal mathematical notation of the 3-D switch.

Chapter 5 mainly focuses on hardware prototyping of the proposed 3-D Elevator Switch. This chapter extends the algorithms in Chapter 4 from theoretical design to practical hardware implementation. The chapter discusses the internal block diagram as well as the flow chart. It also elaborates the edge and core elevator module design in greater details. After discussing the components of every single component in 3-D optical switching network, three sets of hardware experiments are conducted to verify the correctness of the system.

Chapter 6 summarizes the dissertation work. The application aware optical switching addresses the shortcoming OBS in terms of burst assembly and end-to-end latency. The 3-D elevator switching demonstrates its promising potential in solving the traffic blocking problem towards next generation optical networks. The prototyping effort of both application aware optical switching network testbed and 3-D optical switch provides valuable insights for practical adoption of new technologies.

Chapter 2 Background

2.1 Overview of Dense Wavelength Division Multiplexing Technologies

Wavelength Division Multiplexing (WDM) multiplexes a number of optical signals onto a single optical fiber by using different frequency bands of laser light. The WDM system utilizes a multiplexer at the transmitting side and it decouples carrier signals using a de-multiplexer at the receiver end. It features an ultra-high bandwidth transmission without adding new optical fiber. A smooth and cost effective network upgrade is expected since there is no need for overhauling current optical backbone network infrastructure.

In reality, not all wavelengths are suitable for communication if attenuation characters and different types of dispersions are taken into consideration. The transmission distance is limited due to intermodal dispersion, chromatic dispersion, polarization mode dispersion as well as spreading of optical pulses. These limit the bandwidth of the fiber and traffic may not be able to distinguish at the receiver.

The wavelengths having the least effects are most favorable for transmission. These wavelengths are further standardized into six transmission bands. Each transmission band is called a transmission window. Table 1 shows the wavelength bands and their corresponding transmission window.

WDM technology is further categorized into two parts: coarse wavelength division multiplexing (CWDM) and dense wavelength division multiplexing (DWDM). CWDM provides up to 8 channels at the wavelength between 1530 nm to 1565 nm while

DWDM could brought closer to 80 channels with channel spacing of 50GHz in the same transmission window.

Table 1: wavelengths bands and corresponding transmission window

band	transmission window(nm)
O	1260-1360
E	1360-1460
S	1460-1530
C	1530-1565
L	1565-1625
U	1625-1675

2.2 Optical Switching Network

2.2.1 Overview of Switching Networks

Optical switching networks have been commonly used with the increasing traffic originating from end users as well as high bandwidth applications. It could be found not only in wide area network (WAN), which is a long distance back bone network, but also in Metro area network (MAN) and local area network (LAN). Figure 1 is the network hierarchy of most common networks. WANs are connected through backbone long distance fibers. Incoming traffic from WAN is dropped onto MAN. MAN is a ring structure that connects all LANs together. Traffic from LANs will be added to MAN and be sent from MAN to WAN upon request.

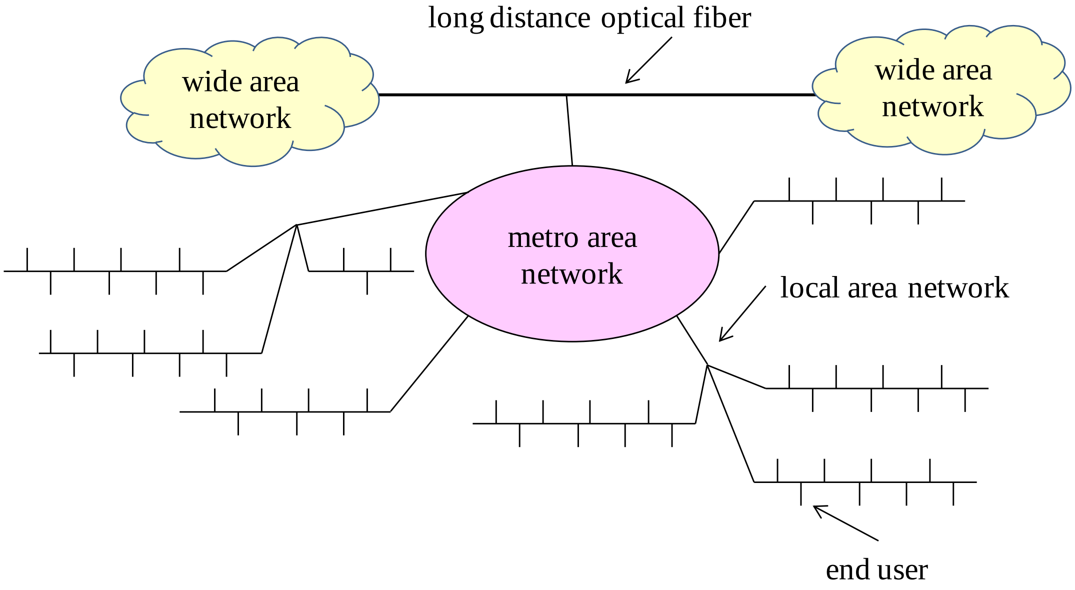


Figure 1: current switching network hierarchy

When delivering the communications connectivity, the parts that actually reaches customers are traditional copper wire subscriber lines, Ethernet cables, coax cables and etc. This is typically the speed bottleneck in communication networks and usually refers to the last mile in order to emphasize the importance of the end users. To mitigate the problem of the last mile, fibers are gradually brought close to end users. Access networks are more and more utilizing optical and wireless media. Fiber network features high bandwidth but the mobility is limited. In contrast, wireless network is bandwidth limited but end users have more freedom of movement. Three switching technologies are discussed below.

2.2.2 Electrical Packet Switching (EPS)

Electrical Packet Switching (EPS) converts optical packets from the optical domain to the electrical domain at each hop using a pair of O/E and E/O converters. Packets at the core nodes can be processed using current popular EPS algorithms. Any traffic congestion or contention could be solved by buffering the packets electronically. The processed packet is attached to one optical channel and it is then converted back to the optical domain at the egress port. Figure 2 shows the router architecture of EPS. There are several disadvantages on EPS: every DWDM channel requires one pair of O/E and E/O converters. The O/E/O converters are expensive which sets a barricade towards the large scale deployment of EPS with a large number of DWDM channels.

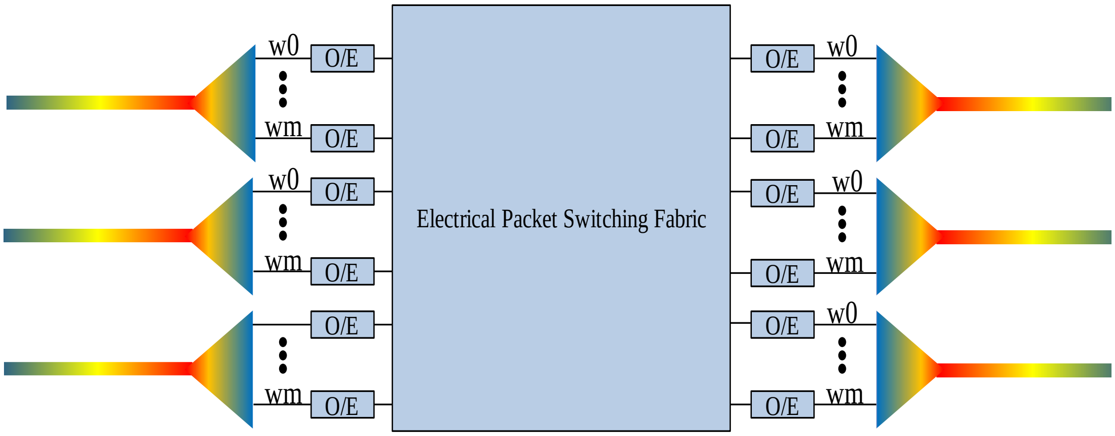


Figure 2: Electrical Packet Switching architecture

2.2.3 Optical Circuit Switching

Optical router in the OCS [1] technique reserves a certain optical path for the communication before the communication itself starts. By configuring the optical circuit at each hop, the upcoming data signal travels through an all-optical light path. An explicit tear-down signal will be generated by the edge router when communication finishes and it would traverse through every hop as the setup signal does. By configuring the core routers, the optical leasing line will be released and will be reused upon the receiving another set up signal in the future.

OCS has several advantages: the all-optical data transmission is achieved, and also the switching capacity is high while maintaining relatively simple QoS mechanism. However, optical circuits require time to set up and tear down and they suffer from severe bandwidth underutilization in the absence of data between end users [2] [3] [4].

2.2.4 Optical Packet Switching (OPS)

Packet switching technology is the most natural thought since it is capable of addressing the bandwidth underutilization problems in OCS. Optical Packet Switching (OPS) divides data into small data packets and these packets are independently transmitted in the network. Figure 3 illustrates the OPS architecture. The edge router sends payload and header of the packet together over an all-optical network. Optical header is extracted at the ingress port of the core router while payload is sent to the fiber delay line to compensate for the optical header processing time and switching fabric configuration time. The updated header and the optical payload are merged together at the egress port. The scheduler tries to solve the contention problem with three options: fiber delay lines, wavelength conversion, and path deflection. If two packets arrive simultaneously, the optical packet switch will first route traffic through a dedicated fiber delay line which is a long fiber that allows the traffic circulating through, making rooms at the output port for other traffic during the time period. If all buffers have been used up, it will route the traffic to other wavelength in the same fiber. If there is no wavelength available for this operation, the traffic will be routed to a different neighbor node. There are some shortcomings in optical packet switching: each buffer could only hold several packets and the traffic bouncing to different paths could cause congestion in other wavelengths. Wavelength conversion also comes at a high cost and it is not viable in commercial deployment.

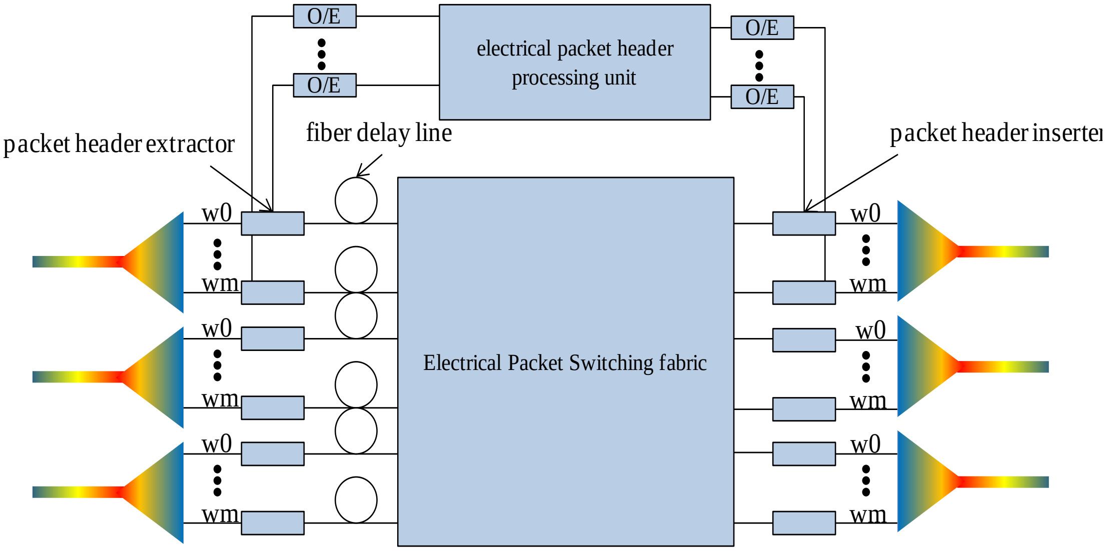


Figure 3: Optical Packet Switching architecture

2.3 Optical Burst Switching (OBS)

Optical Burst Switching (OBS) aims at overcoming the disadvantages of optical packet switching and it is a promising candidate for future optical network. The control header of each burst is sent on an independent control channel before the transmission of the burst. The header configures each core node on the fly, allowing the data burst transmission remains in the optical domain when passing through the OBS router rather than converting back to the electrical form at each hop.

2.3.1 OBS Networks Architecture

The transmission links in the DWDM network takes multiple optical carriers, one of which is used as a control channel to dynamically assign the remaining channels for data bursts. Shortly before the data burst arrives at the core node, a burst header cell (BHC) is sent on an independent control channel. The BHC contains the information of the data burst including the incoming channel and the destination of the data burst. Besides, the BHC also specifies the offset time. The offset time defines the interval time between the transmission of the BHC and the transmission of data burst. The length of the data burst is also included in the BHC and it denotes the duration of the data burst. This information is used by the core routers to reserve and release the optical path for OBS transmission. Figure 4 is an example of multiple BHCs sharing the same control channel, while the data bursts are being transmitted on separated data channels. Wavelength 0 in Figure 4 is used as the control channel and there are *n* wavelengths serving as data channels. The time duration between the BHC and the burst is the offset time. Since it takes time for each core router to utilize the information of BHC and configure the optical path, the offset time is shrinking while the burst is being transmitted along the path.

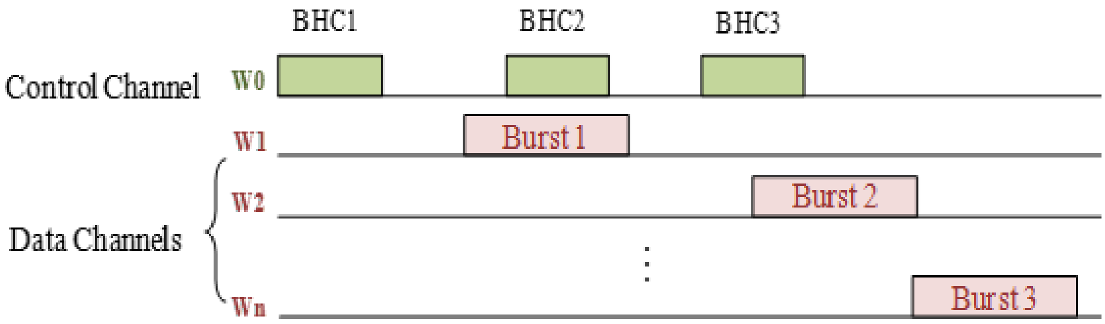


Figure 4: relationship between BHC and Data Burst in OBS

2.3.2 Burst Assembly

Burst assembly is a very important part of OBS. This is a unique feature comparing with OPS, EPS and OCS. By assembling packets into burst under cautious manipulation of burst assembly schemes, the bandwidth efficiency of data transmission is greatly improved. The burst assembly occurs at the edge router before entering the MAN from the DWDM's point of view. Figure 5 is a detailed OBS edge router architecture and it shows the data flow in the OBS edge router. The burst assembly takes place at the end of the Packet Classifier. Traffic from the end user is sorted by the Packet Classifier according to the QoS groups and destination addresses. The burst assembly algorithm dictates data aggregation method and the way of attaching the packet to form a burst. In general, there are two assembling schemes at packet classifier: time based assembly and length based assembly. For the time-based packet assembly, a timer is implemented at each queue to trigger the formation of burst when it times out. On the other hand, the length-based burst assembly will stop aggregating traffic only if a predefined length threshold of the assembly queue is satisfied. Intuitively, longer burst improves network efficiency and it reduces traffic overhead in the whole OBS network. However, shorter bursts are suitable for time sensitive application such as video conference and tele-

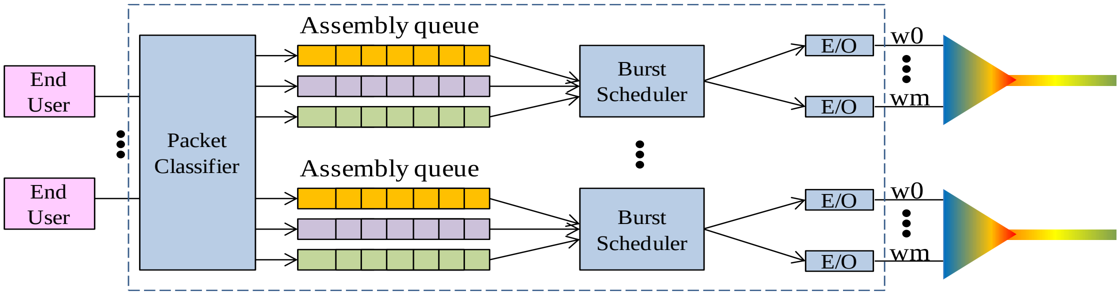


Figure 5: OBS edge router architecture

surgery application. When being implemented, each assembly queue can choose between time-based assembly and length-based assembly scheme or a mixed assembly mechanism based on the application requirements.

Neither time-based burst assembly nor length-based burst assembly could meet the dynamic traffic change requirements in real-time traffic situation. In the case of dynamic traffic, a burst assembly criterion is too rigid to adapt. Therefore, a variety of burst assembly schemes are analyzed to improve the performance. In the time-based burst assembly scheme, authors in [5] proposed an adaptive algorithm to control the burst queue size when traffic rate changes: if the queue size reaches the upper threshold in unit time, the queue size is incremented by one and the size of the next burst is one packet bigger than the previous one. If the queue size is smaller than the lower threshold in unit time, the queue size is decremented by one and the size of the next burst is one packet smaller than the previous one. This scheme dynamically adjusts the burst size step by step and it keeps track of the change of the traffic rate. A significant improvement of traffic latency is achieved in this scheme.

A time and length based mixed-algorithm can also solve the above issue. The adaptive burst formation scheme [6] can be changed according to the fluctuating traffic condition. The adaptive burst formation algorithm exhibits the characteristics of both the time-based and the length-based burst assembly and it stands out for its more acceptable performance in terms of end-to-end latency. The paper also discussed the traffic pattern change during the assembly process. Reference [7] presents two different mathematical models to evaluate the performance of the timer-based and the threshold-based burst assembly schemes. Simulation results shows that the smaller size burst has better link utilization as well as lower blocking probability in most cases.

2.3.3 Burst Scheduling

The goal of the OBS is to enable transparent transmission of optical bursts. The header is transmitted through a separated control channel before the transmission of burst. There are two most popular burst scheduling scheme for the burst scheduler module: just in time (JIT) and just enough time (JET).

JIT provides the burst reservation as soon as the burst header is received by the burst scheduler. The data burst will go through the preconfigured optical path assigned by the burst scheduler. The idea and realization of this mechanism is quite simple. Without taking the offset time or the processing time into consideration, the scheme stands out for its fast burst scheduling and simplicity. Since the wavelength has to be reserved before the arrival of the burst, the performance and efficiency could be low due to the bandwidth underutilization between the channel reservation and the actual burst arrival.

On the other hand, in JET, by calculating the exact burst arrival time, the channel could be set up at the point of data burst arrival, thus avoiding the problem of bandwidth underutilization in JIT. JET provides a fixed offset between the burst header and the data burst at the ingress edge node. Every time a burst scheduler receives the burst header, its offset information is then extracted by the scheduler, and a new offset is written in to the burst header which is the received offset time minus the processing time in the current node. Data channel is then reserved shortly before the arrival of the upcoming burst. Unlike JIT, the delayed reservation can avoid the vacant in data channel when waiting for the burst. However, it usually requires complicated algorithm. Figure 6 shows JIT and

JET scheduling scheme, the *t* axis represents time, W_0 represents control channel and W_1, W_2, W_3 … W_n represent data channels, respectively. For JET in Figure 6(a), the channel reservation time for Burst 1 equals to the length of Burst 1 while JIT in Figure 6(b) reserves a longer period in W_1 for the same burst, which causes the bandwidth underutilization in W_1.

As shown above, there are both advantages and disadvantages in JET and JIT. The JIT requires more bandwidth but it has lower algorithm complexity while the JET comes with an advanced yet complicated scheduling algorithm in order to achieve higher bandwidth utilization. The contention arises in the data channel in JET and JIT can be

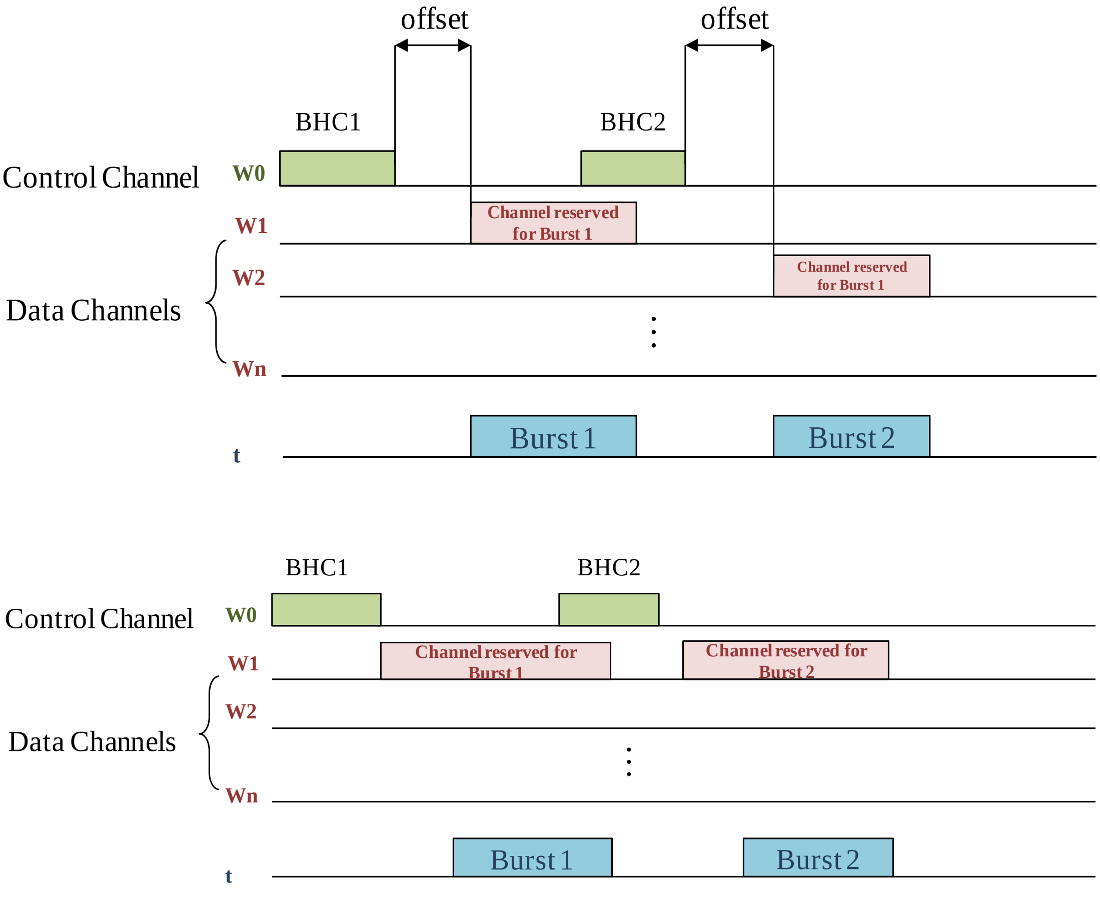


Figure 6: OBS JET and JIT scheme

solved by selective segment dropping [8]. Instead of dropping the whole burst, only the overlapped portion: the tail of the first burst or the head of the second burst is dropped. The significant advantage of burst segmentation (BS) dropping is that there is a good chance that the truncated burst could be delivered to the output ports, thus reducing the packet loss rate.

Reference [9] conducts the performance analysis on JET, JIT and BS for OBS networks. It shows that under the BS scheme, the performance of OBS network can be significantly improved. The experiments also take various traffic scenarios, the number of nodes, delay variations into consideration and it shows that the overall performance of BS is much better than the performance of JET and JIT.

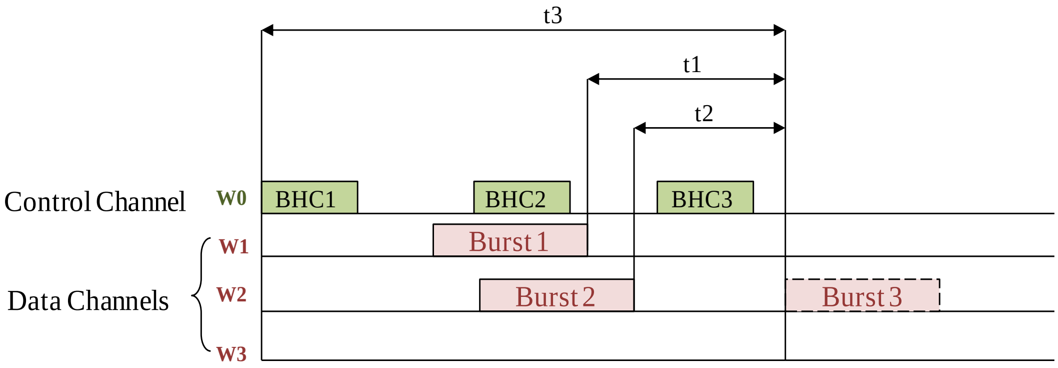


Figure 7: LAUC scheduling algorithm in OBS router

When multiple data channels are available for the upcoming data burst, the scheduling algorithm becomes important due to its impact on the bandwidth utilization. In general, the data channel scheduling can be classified into void filling [10] and non-void filling [11] ones. The non-void filling solution is general called latest available unscheduled channel (LAUC) algorithm[11]. This algorithm tries to find the unscheduled channel for the data burst where the selected channel has the smallest time gap among all channels. As shown in Figure 7, Burst Header 3 wants to reserve data channel for Burst 3.

According to the channel map, three data channels W_1, W_2 and W_3 are available when Burst 3 arrives. However, time gaps between the previous burst and Burst 3 are different: On channels W_1, W_2 and W_3, the time gaps are t_1, t_2 and t_3 specifically. The LAUC scheduler schedules Burst 3 onto the channel that has the smallest time gap. Hence, Burst 3 is scheduled on W_2.

LAUC-VF scheduling algorithm differs from LAUC scheduler in the sense that the scheduler could fill the void between two bursts on data channel. In Figure 8, Burst 4 wants to reserve a data channel. W_2 becomes unavailable using LAUC scheduler since Burst 3 has already occupied the channel till later time and the availability of W_2 will be at the end of Burst 3. However, Burst 4 could be scheduled onto W_2 because there is a large enough gap (void) between Burst 2 and Burst 3. The LAUC-VF will capture this void and fits Burst 4 into it.

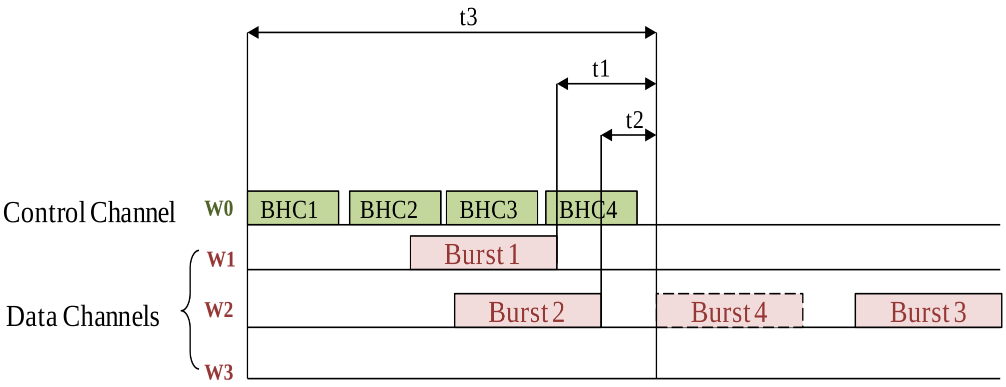


Figure 8: LAUC-VF scheduling algorithm in OBS router

There are advantages and disadvantages for both two scheduling schemes. LAUC scheduler stands out for its simplicity and easy implementation. The scheduler only needs to record the unscheduled time for each channel on the channel map. The drawback of LAUC scheduler lies in the insufficient bandwidth utilization. As a consequence, the burst encounters high loss ratio. The LAUC-VF on the other hand has higher resource utilization. To manage the complexity of the algorithm while achieving the bandwidth efficiency, the LAUC-VF scheduler needs special-purpose parallel process architecture to meet the real-time traffic management requirement. Also, there are many discussions on OBS under limited amount of wavelength converters towards burst blocking probability [12] [13] and a mathematical model is provided in reference [14].

2.4 DWDM Multi-Mode Switching

DWDM Multi-Mode Switching [15] allows wavelength channels in a DWDM network to be individually configured in the EPS, OCS or OBS mode [16] [17] [18] [19] [20]. The most suited switching mode can be applied to transport individual applications, as well as individual message types within an application. The architecture provides an intrinsic support to asymmetric data transfer in the two-way communication. For example, in cloud-based applications, short service request commands are usually transferred from the client computer to the cloud server while high bandwidth multi-media data are transferred from the cloud server to the client computer. In a DWDM multi-mode switching network, the commands are switched using the EPS mode while large data transfer is through the OCS or OBS mode without the need for O/E/O conversion.

At a multi-mode ingress edge router, packets may be received on different line interfaces such as IP, Gigabit Ethernet (GE)/10 GE, and may be classified to be sent using a desired switching mode (EPS, OCS, or OBS). Classification can be performed at the packet level or based on specific settings such as a particular line interface. Based on the results from the classifier, the packets or traffic streams will be processed according to one of the following switching modes: (1) EPS mode - Packets sent in the EPS mode are queued and transmitted as individual packets over EPS channels; (2) OCS mode - Traffic streams in the OCS mode are sent on pre-established optical channels; (3) OBS mode - Packets or flows in OBS mode are assembled into bursts based on the destination edge router address and sent over on-demand optical paths that are created for the bursts. All wavelengths are combined onto the outgoing DWDM link.

DWDM multi-mode network has the following characteristics: Multimodal switching is achieved by allowing utilization of multiple switching modes simultaneously across different wavelength channels. Reconfigurable channels can be used for EPS, OCS or OBS mode based on dynamic traffic demands. Asymmetric data transfer utilizes the best switching mode for the type of message within the application to maximize bandwidth utilization and minimize delay and/or network resource usage.

Among the three switching modes supported by DWDM Multi-Mode network, the EPS mode has the finest control over data being transmitted. On the other hand, the OCS mode provisions the bandwidth at the wavelength level, and its usage is limited to mission critical applications at the expense of bandwidth utilization. The OBS mode assembles packets into bursts, and allows sharing of optical bandwidth over time. With the dynamic configuration capability of wavelength channels, Multi-Mode switching can potentially achieve great scalability by directing majority of traffic over optical paths, reducing the need for expensive O/E/O converters.

2.4.1 Reconfigurable Multi-Mode Core Router Platform

The DWDM Multi-Mode switching technique features a limited amount of O/E/O converters that could be shared among different switching modes through the core optical switching fabric. Therefore, with a relatively small number of electronic switching ports, it is viable to build a cost effective expansion of the network supporting a larger number of DWDM channels.

In DWDM Multi-Mode switching, optical burst switching serves as a middle ground as a unified approach to switch OCS/EPS/OBS in the same platform. Therefore, Multi-Mode switching is also called reconfigurable asymmetric optical burst switching (RA-OBS). The two terms are used interchangeably in this dissertation. In the Multi-Mode-OBS routers, wavelengths channels in the optical fiber are shared among three different switching modes: EPS, OBS and OCS. Scheduling DWDM channels is a challenging problem [21] [22] due to the total number of DWDM channels need to be scheduled and the ultra-high speed requirement of individual channels. Although a separate scheduling scheme can be devised to configure the EPS, OBS and OCS modes respectively, an integrated solution will be more robust and more cost effective.

The problem of allocating a channel for the EPS or the OCS mode can be converted to a special case of the OBS channel scheduling problem, therefore, can be solved by integrated scheduling approaches. Such schemes can also be applied to the universal signaling, switching and reservation framework [23] [24].

2.4.2 DWDM Multi-Mode Switching Networks

The characteristics of Multi-Mode switching network include reconfigurability and asymmetry. Each wavelength can be reconfigured to a different switching mode based on dynamic traffic loads and/or specific application requirements. An application can use a different switching mode for each direction in its two-way communications, taking into account the characteristics of the data in each direction.

The DWDM Multi-Mode switching network is illustrated in Figure 9. In DWDM Multi-Mode switching networks, the optical links between nodes contains three different traffic types simultaneously operating in the network. Wavelengths can be individually configured and reconfigured into one of the three switching modes: EPS, OCS or OBS mode. Multi-Mode switching core routers internally switch data according to the switching mode configured for the wavelengths: (1) EPS mode switches data electronically on a packet by packet basis; (2) OCS mode switches data at the wavelength level using optical circuit switching; (3) OBS mode switches data according to the optical burst switching. The core routers run under the Multi-Mode scheduling algorithm and it enables OBS, EPS and OCS coexistence and shares wavelength among each other. The edge takes care of traffic aggregation and classification. With the packetized-OBS scheme, the edge router could packetize control over OBS mode.

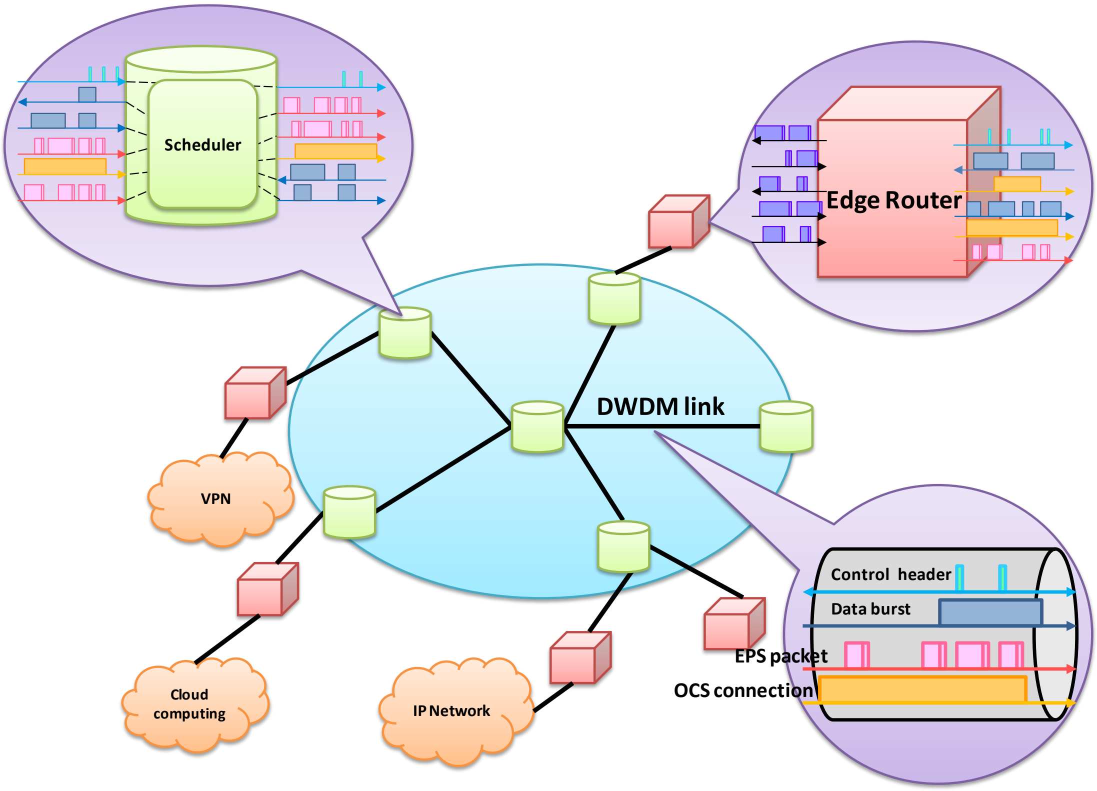


Figure 9: DWDM Multi-Mode switching network

2.4.3 Multi-Mode Switching Router Architecture

The multi-mode switching functions in the RA-OBS network is supported by the RA-OBS core router architecture shown in Figure 10. The incoming wavelengths are separated by optical de-multiplexers. At least one wavelength channel in each DWDM link is configured as the control channel, while the rest can be dynamically reconfigured in any of three switching modes, namely, EPS, OCS and OBS. Control packets for different switching modes can share the same control channel.

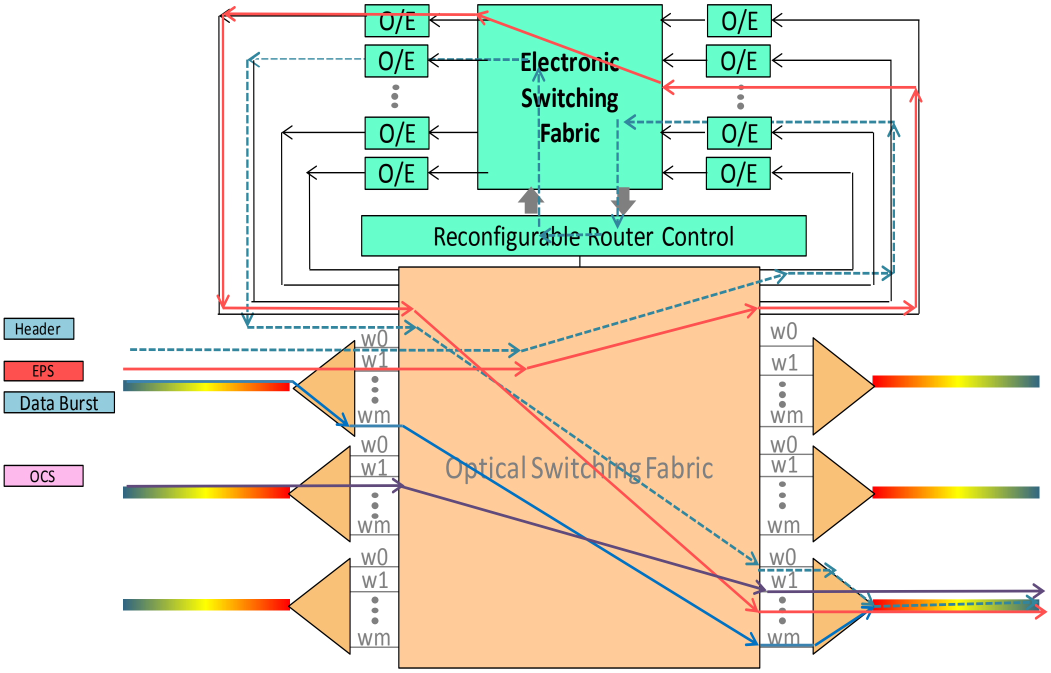


Figure 10: Multi-Mode switching in core router

In the example shown in Figure 10, four wavelengths, W_0-W_3, are configured in different modes in the same fiber connected to Input Port *i*. Wavelength W_0 is the control wavelength to carry the burst header packet (BHP). The control channel W_0 is directed by the optical switching fabric to one of the O/E converters. The BHP is sent to the Reconfigurable Router Control (RRC) unit by the electronic switching fabric, and is processed electronically. After processing, the BHP is forwarded to an E/O converter and is directed to Output Port *k* by the optical switching fabric.

Wavelength W_1 is in the OBS mode carrying data bursts. When the data burst arrives at the core router, an optical path has already been set up by the BHP to direct the burst to Output Port *k* through the optical switching fabric. Wavelength W_2 is set to the EPS mode to carry packets which are switched electronically. An incoming wavelength configured in the EPS mode is routed to one of the O/E converters through a pre-established optical path in the optical switching fabric. The packets are then processed electronically, similar to a traditional electronic packet router. The packets are routed through the electronic switching fabric, and are converted to the output wavelength using an E/O converter. In the figure, the packet is sent on output wavelength W_2 which is routed through the optical switching fabric using a pre-established light path to Output Port *j*. Wavelength W_3 is in the OCS mode where data follows the pre-established light path in the optical switching fabric to Output Port *j*.

As shown above, each of the four wavelengths is switched based the operation of the particular switching mode. The RA-OBS core router architecture can support concurrent DWDM Multi-Mode switching on an integrated router platform. The optical switching fabric not only switches among incoming and outgoing wavelengths, but also interconnects the shared E/O and O/E converter pool. This shared pool of O/E and E/O converters also provide indirect wavelength conversion [25] for OCS and OBS connections by converting the optical signals back into the electrical domain, and retransmit them on a different wavelength using tunable lasers, which is considered most practical solution in the near future[26].

In the Multi-Mode scheduler, the Multi-Mode channel selection and channel update unit can be shared among all three functions: burst scheduling in the OBS mode, connection setup/teardown in the OCS mode, and channel reconfiguration in the EPS mode. The scheduling approach not only provides an integrated solution to the problem, but also enables dynamic wavelength sharing among different switching modes. The operations in each mode are described briefly below. In particular, it shows how to convert the OCS connection setup/teardown and EPS channel reconfigurations problems into a special case of OBS scheduling problem.

First, the integrated Multi-Mode scheduler supports generic OBS burst scheduling. In traditional OBS networks, bursts sent on wavelengths configured in the OBS mode are scheduled onto DWDM channels based on the two parameters carried in the BHP: the offset and the burst length. The offset time defines the time between the transmission of the first bit of the BHP and the first bit of the data burst. The length field specifies the time duration of the burst. Usually a maximum burst length is used to restrict the size of the bursts. The integrated Multi-Mode scheduler inherently support burst scheduling functions [27].

In the case of an OCS connection request, an OCS connection is treated as a burst of infinite length at the connection setup time. The actual connection time is adjusted when a teardown control packet is received. As a result, OCS mode connection setup problem can be converted to an extended OBS channel scheduling problem. The offset time in the extended protocol will support both instant reservations ($\mathrm{offset} = 0$), or delayed reservations ($\mathrm{offset} > 0$).

Similarly, reconfiguring a wavelength in the EPS mode can be mapped to the problem of setting up an OCS connection that requires wavelength conversion. As described earlier, an incoming wavelength in the EPS mode is routed to an O/E converter through the optical switching fabric, while an outgoing wavelength in the EPS mode is routed from an E/O converter to the outgoing link. An EPS connection stays in effect until the next EPS reconfiguration request of the same channel, which can be treated as a connection teardown request in the OCS mode. Therefore, the operation of setting up a wavelength in the EPS mode can also be treated as an extended burst scheduling problem.

Although Multi-Mode switching architecture paves a road for future network scaling, it is arguable that bigger optical pipes are not equivalent to better network services to the end users, especially when the individual control over packets are lost. For example, once a packet becomes a part of a burst during the burst assembly process in OBS, its identity literally disappears until the burst is dissembled at the destination edge router. As more real-time and bandwidth-demanding applications such as video conferencing, on-line gaming and telemedicine emerges in recent years, it becomes more pressing to identify a better solution.

Chapter 3 Application-Aware Dynamic Optical Switching

Multi-Mode Switching offers a consolidated approach to DWDM-based communication and it successfully solves the problem of heterogeneous traffic integration. Using OBS as the middle ground with the pipelined design in hardware implementation, the Multi-Mode switching router could achieve O(1) time complexity. However, there are still questions on the architecture of the Multi-Mode router. First of all, the router does not distinguish the traffic priority and this hampers the implementation of the application layer QoS. Secondly, the offset time introduces a large amount of delay from the perspective of end-to-end latency. When functioning in the OBS mode, packets need to be aggregated at the edge node to form a burst and the burst is held until the header is generated and transmitted. The burst has to wait for an offset time in order to be transmitted transparently in the data channel. All of these protocols and operations severely impact the performance of the application and the network performance is degraded to a large extent.

The Application-Aware (A^2) dynamic optical switching proposed in this chapter fundamentally reconstructs the network from the application's aspect: a new transmission protocol is created to fits the requirement of applications. The A^2 dynamic optical network addresses the offset problem intrinsic in existing Multi-Mode switching scheduler. The proposed A^2 dynamic optical switching also features a novel dual input/output scheduler. With the help of the input scheduler, a data path will be preconfigured for the service when transmission happens in the forward data path.

3.1 Proposed Application-Aware (A^2) Optical Networks

Figure 11 is an overview of optical switching networks running under A^2 dynamic switching scheme, consisting of edge router, core routers, as well as servers and Cloud applications running through IP. These network elements are connected through a strand of DWDM links, forming an A^2 dynamic optical network. At the client side, the edge routers take request from clients as shown on top right part of the figure. Services such as VoIP, VPN and FTP can issue requests to the edge router through dedicated control channels. Data paths of these services are connected to the data port of the edge router so that corresponding packets from the server going out from the data port could reach each client. The edge router at the service side will initiate a data transfer after server finishes the data fetching. The core A^2 routers reserve the data path for the service based on the scheduling protocol and information the previous hop provides. However, the reservation

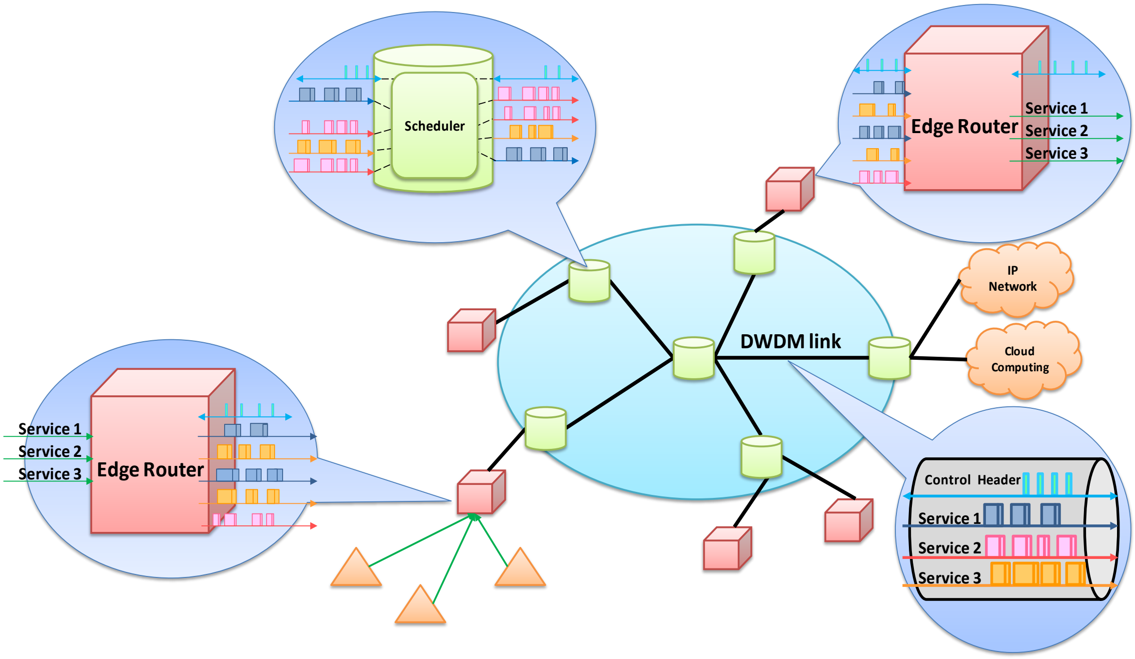


Figure 11: A^2 Dynamic optical switching networks

takes place at the incoming port rather than the outgoing port of the core router. A detail scheduling algorithm is discussed in the later part of this chapter.

The service passing through the DWDM links could be video stream applications, FTP, VPN as well as prevalent data center applications. The application-aware function enables the A^2 router treating each service differently. By assigning the priorities of services at the edge, the A^2 router could reserve the data paths for high priority services. To address the latency issue, the A^2 optical switching network focuses on backward data path reservation which will be shown in the following discussion. In addition, the A^2 router is compatible with Multi-Mode switching router. By enabling the optional forward optical path reservation function block in A^2 router, the A^2 router could be configured as traditional Multi-Mode switching router.

3.2 Application-Aware Optical Path Setup

The Multi-Mode switching router described in previous chapter reserves the optical path for the next hop at each core node: The RRC configures the switching fabric ahead of time to reserve data channel based on the header information. However, the network is not always ideal: traffic congestions and network jitters could happen for many reasons. Unbalanced traffic load, immediate high bandwidth traffic injecting, network disturbance and other network malfunctions could all lead to higher traffic latency than expected. The actual time to transmit data could be longer than expected. The problem arise from this is that since the scheduler is not able to adjust the channel map after the channel reservation, it can potentially lead to the data burst being corrupted and eventually dropped due to the increase of network latency.

Secondly, the offset time is required in the Multi-Mode switching configuration:

every data burst or OCS, EPS transmission has to be configured by the packet header an offset time ahead. The propagation time, scheduler processing time and optical switch configuration time are compensated by the offset time during the transmission of the data burst. This scheduling mechanism has the disadvantage of increased end-to-end latency in the traffic transmission from the perspective of applications. The A^2 router is proposed to solve these two issues.

In general, Internet traffic is mostly asymmetric. For example, it only takes several kilobytes to send the request in order to watch the online video, but the data retrieved from the video server is tremendous. Another example would be data downloading: by clicking the download link, a large amount of data is being transferred from the server side to the client side. If this characteristic is utilized, a reverse path application-aware optical path setup can be used to avoid the disadvantage on the offset time of OBS, thus reducing the latency during path setup when high bandwidth data are being transmitted from the service side to the client side.

The overall architecture of the proposed application-aware optical path setup is shown in Figure 12. In this network, there are several network components: the right hand side is the customized application that could generate the necessary requests to retrieve the data from the service provided on the left-hand side. Requests are usually small but data retrieved are typically fairly large. By manipulating the application request, the reverse path from the service to the application can be configured as soon as the data arrives, eliminating the offset time in the traditional OBS router.

In the A^2 router shown in the middle of the figure, the control channels are connected through the O/E/O converters. Every request packet from the client is converted into the electrical domain. The header information is then extracted by the A^2 router controller. The A^2 router has a database that could look up the time for the header to propagate to the service end. The database also stores properties of each service such as the amount of time allowed to retrieve the data. Using the information stored in the database, the minimum amount of time to configure the path for data from service side to the client side is obtained and traffic arrival time can be determined.

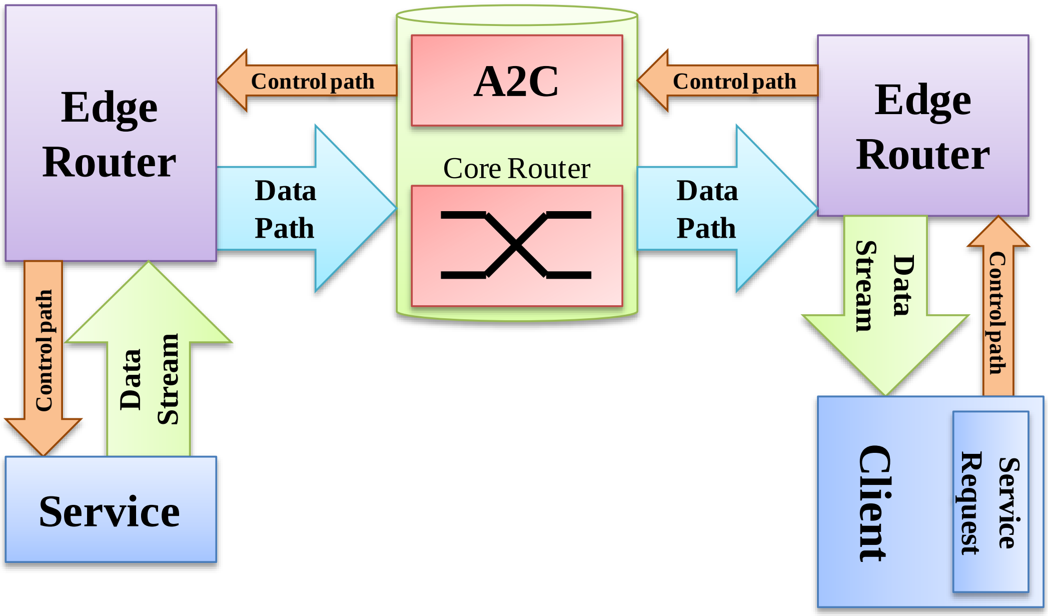


Figure 12: application-aware optical path setup architecture

Application-Aware Controller (A^{2}C) uses the request information and the database information to decide the data arrival time on the reverse data channel. The service type, configuration time, configuration channel are stored in the A^{2}C storage. Since the data retrieve time plus the propagation time is large comparing to the optical switch configuration time which is under one millisecond, by the time the data in the reverse data channel arrives, the optical switch is already configured. Since the reverse data path has been configured ahead of time, from the aspect of the service application, there is no optical data path setup delay comparing with the Multi-Mode OBS router. This end-to-end latency is improved because the design overlaps the time between data retrieval and data path configuration.

In fact, there could have many potential reverse data links, each of them has a different link speed. After reading out the header information and decrying the service, the A^{2}C could choose the optimal link for specific service and schedule the service onto dedicated reverse data path. Furthermore, since there is no offset time between data and header, comparing to the OBS router, the Application-Aware optical path setup eliminates the complicated burst assembly occurred at the edge router. This feature also reduces the burden at the edge to assemble the data burst.

One of the most important features the application-aware optical path setup is the feedback mechanism from the service side to adjust the data path reservation time in case

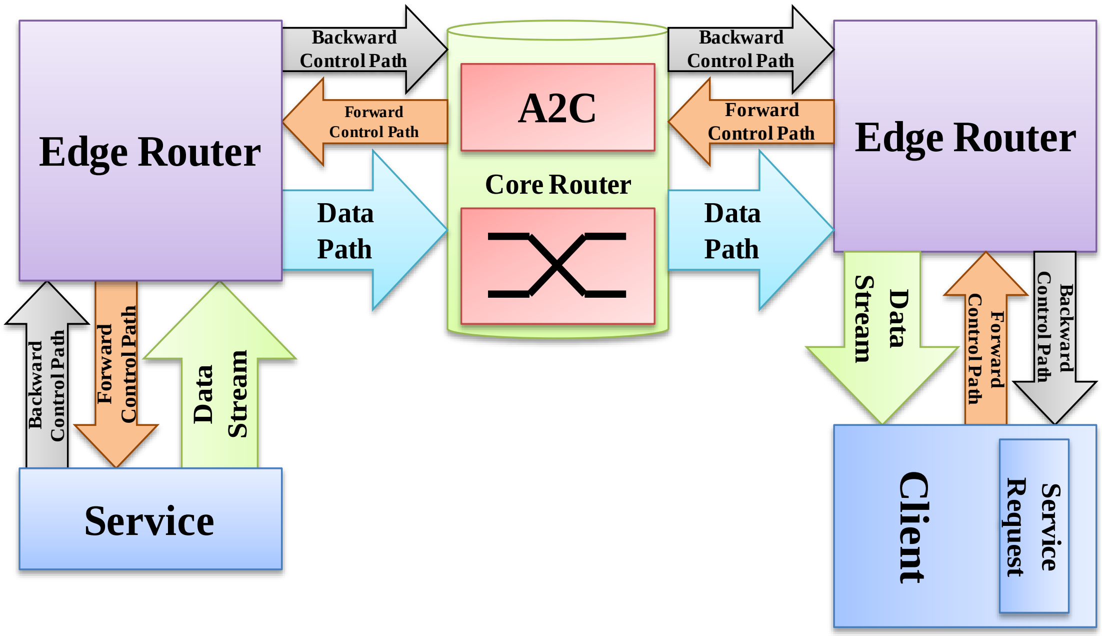


Figure 13: reverse path reservation

of excessive data transmission or unstable network condition, making the data path reservation adaptive and bi-directional. Figure 13 shows the reverse path reservation. The reverse control packets generated by the service are transmitted in a separated control channel. The A^2 router then fetches control packets using the header extraction module in A^2C. After decrypting the header, the A^2C captures the service type information and the request type from the header. If the existing reservation entry has been found in the storage and the scheduler could make an extended reservation in the data channel for this service, the entry is then updated by the A^2C. In case the scheduler cannot make reservations, the request is dropped to prevent the service sends excessive data.

Although previous example only shows one application in the optical network, in real case, there could have several applications sending different requests to the server using dedicated control channels and there are multiple hops in the network. In this case, the database of each A^2 router is then slightly different from others. Figure 14 illustrates an A^2 optical network that has one service, one client and several cascade core routers. When a service request is initiated by the client, the header is then captured by the first router. In addition to calculate the round trip traffic propagation time and the data fetching time from the service side, the router has to consider the header processing time the rest routers consume. Along with the header processing time, the traffic propagation time as well as the data fetching time is also stored in the database of A^2C. A new header containing an updated packet arrival time (previous packet arrival time minus one header processing time minus one hop propagation time) will be regenerated through the header regenerate module if the scheduling configuration is successful. This new header will be propagated to the next hop and the same operation is repeated until the header reaches the destination edge and the server then starts to fetch data from its own storage device. The header information will be recorded by the A^{2}C storage future channel extension use.

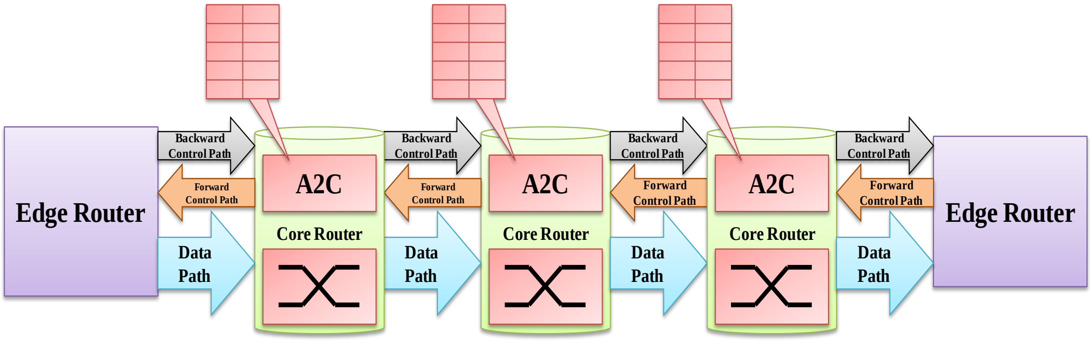


Figure 14: Path reservation of casacated core routers

3.3 Application-Aware Optical Path Controller

The optical path setup and reservation discussed above takes place at each hop. All these tasks are manipulated and managed by the A^{2}C and it plays an important role in information exchange and signal diverting. For one thing, the A^{2}C selects the path for data streams and it coordinates among system resources, switching algorithms, routing strategies as well as network conditions. For another thing, the scheduler in A^{2}C periodically modifies and updates the channel map database, rewrites the header information to aid the path reservation of the next hop.

The general function of the A^{2}C includes header processing, channel selection, header regeneration and optical switching matrix configuration. Figure 15 illustrates the internal block diagram of the A^{2}C in the A^2 router. By complying with the predefined protocol, the typical header packet is divided into different meaningful fields in the header extractor. The header extractor module will analysis every incoming header and the associated registers will be written for A^{2}C.

Database module contains a readable and writable storage device. After powering up, the routing table is generated and stored into the database. The database keeps track of the number of hops from its own router to every service. It also keeps a record of the data fetching time for every service in order to calculate the data arrival time from the service to the current router. In general, the network stays the same after powering up and there is no need to change the critical value in the database. However, the database could be periodically updated thorough if the network is dynamic constructed.

The post header information such as the data arrival time, data length field, incoming and outgoing channel, service type needs to be stored into an individual storage module. This is because when service requires more time to transmit the data packets, as long as the service provides a correct service type in the reverse request, the A^{2}C would search the storage device using the service type field and locate the corresponding reservation entry in the storage. The channel selector/scheduler directly controls the optical switching matrix and it configures the optical switching matrix if a valid configuration signal is

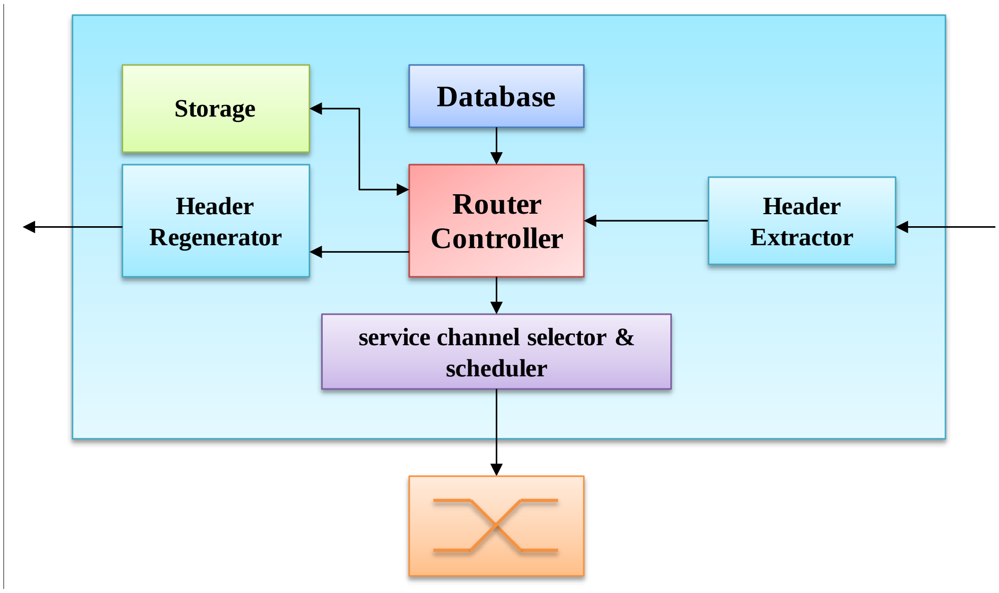


Figure 15: blockdiagram of the A^{2}C in A2 router

received and no conflict is found.

Figure 16 demonstrates the flow chart of the scheduling part. When a new header comes to the extractor which in charges of header receiving and information extraction, the header will be fetched by the extractor. The extracted information contains service type, data duration, and data arrival time from the service. A^{2}C first checks the database for the data fetching time, the propagation time, the number of hops as well as the header processing time of the A^2 router. After calculating the data arrival time and end time, two time stamps are generated to schedule the upcoming transmission onto appropriate data channel. The service type is then forwarded to the service channel selector and scheduler, waiting for the scheduling result. If the scheduling is successful, a new header is generated with an updated time field by the header regenerator module. If the scheduling

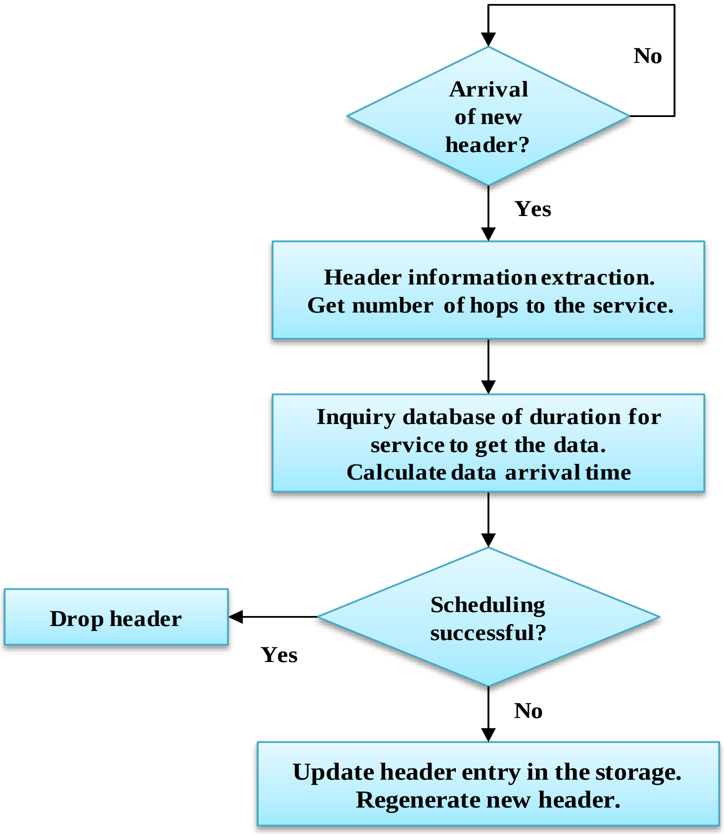

No
Arrival of new header?
Yes
Header information extraction. Get number of hops to the service.
Inquiry database of duration for service to get the data. Calculate data arrival time
Drop header
Yes
Scheduling successful?
No
Update header entry in the storage. Regenerate new header.

Figure 16: flow chart of reverse path reservation

is denied due to the channel unavailability, the header will not be regenerated and the packet is then dropped at this hop.

3.4 A^2 Router Input Scheduler Design

The scheduler in A^{2}C is a critical component in the A^2 router. For one thing, the forward request packet requires the scheduler to reserve the incoming data path of the

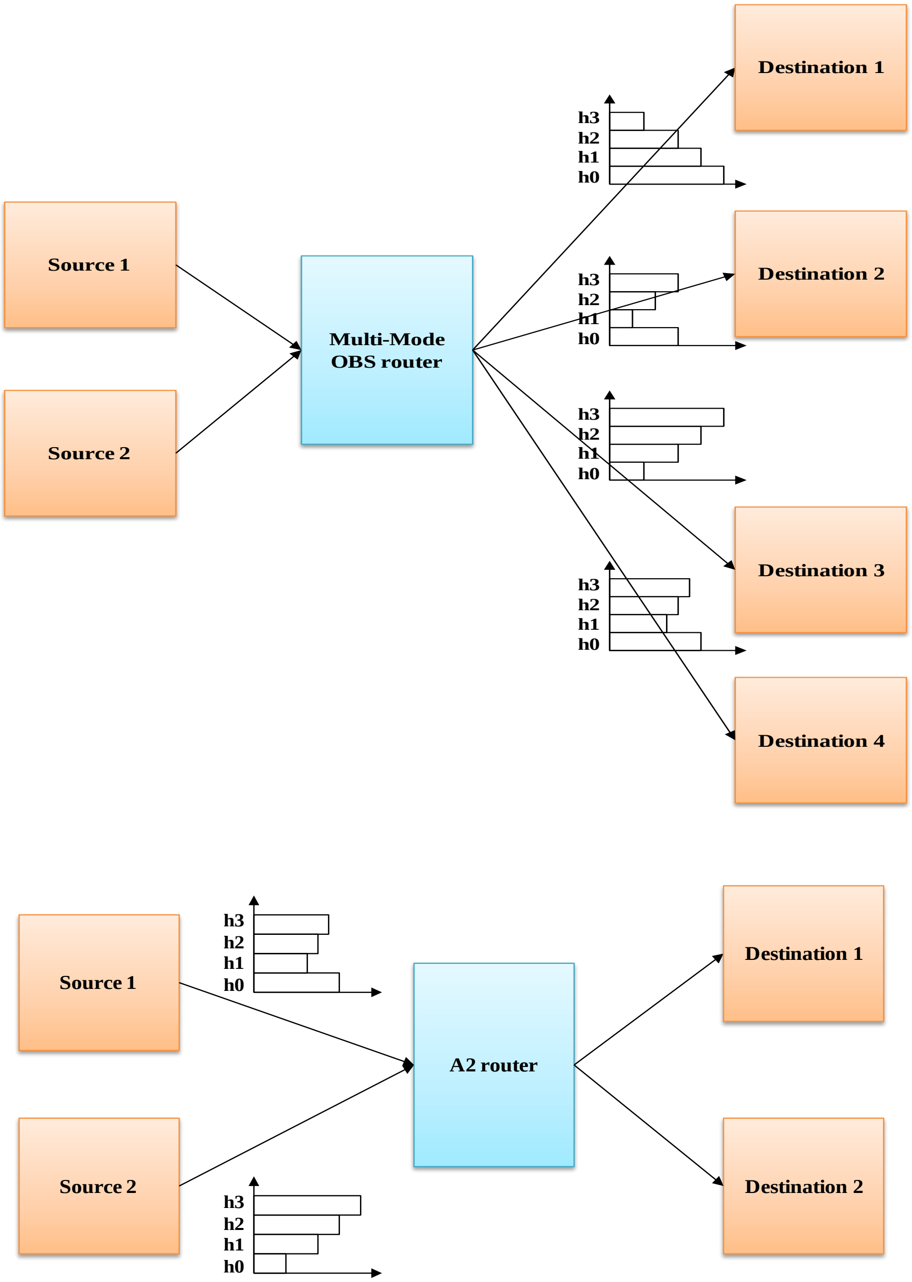


Figure 17: input and output channel scheduler in Multi-Mode OBS and A^2 router

particular service. For the other thing, the scheduler has to handle the backward control packets as well as any kind of reservation extension.

The difference between A^2 router scheduler and the traditional Multi-Mode scheduler is that rather than making output channel reservation in the Multi-Mode scheduler, the A^2 router scheduler focuses on input channel reservation. Figure 17 shows the main difference between output channel reservation and input channel reservation. As mention in the previous chapter, the Multi-Mode OBS supports the channel scheduling of different traffic modes. The idea is an advanced revision of OBS scheduling. The OBS scheduler reserves the outgoing optical paths upon receiving valid control packets. The incoming channel in Multi-Mode OBS router cannot be selected since it is already decided by the previous hop. For each outgoing fiber link that contains several optical channels, there is at least one control channel to propagate the burst header.

The top figure in Figure 17 shows the case where each outgoing fiber link has seven outgoing data channels and one control channel in Multi-Mode OBS network. The Multi-Mode OBS router maintains several schedulers each corresponding to one outgoing link. The scheduler tries to schedule the data path that requires no wavelength conversion first. If the wavelength conversion has to be taken place, the scheduler tries to schedule the packet on to the data channel that has the least available unused channel. The A^2 router scheduler on the other hand reserves the backward data channels upon requests from the previous hop. The bottom figure in Figure 17 shows the case where backward data path reservation occurs. Each of the two clients each has one optical link are connected to the A^2 router. Since the data path is reversed comparing with Multi-Mode OBS router, the A^2 router scheduler reserves the backward data channel at the input side of the data path.

Besides the one control channel (forward control channel) and seven data channels (backward data channels), each optical link also carries one backward control channel. This backward control channel is used for backward data path reservation extension when network condition is not stable.

Several definitions are made to help the explanation of the mechanism of A^2 router scheduler. In Figure 18, define the forward control packets as the control packets from the client to service. Define the forward control channels as the dedicated optical channels carrying these packets. Define the backward control packets as the control packets from the service side. Define the backward control channels as dedicated optical channels carrying these packets. Define the backward data channels as the optical channels carrying the data from the service to the client.

At the very beginning, the scheduler will have information of the outgoing link when request is received for the backward data channel: the header containing the

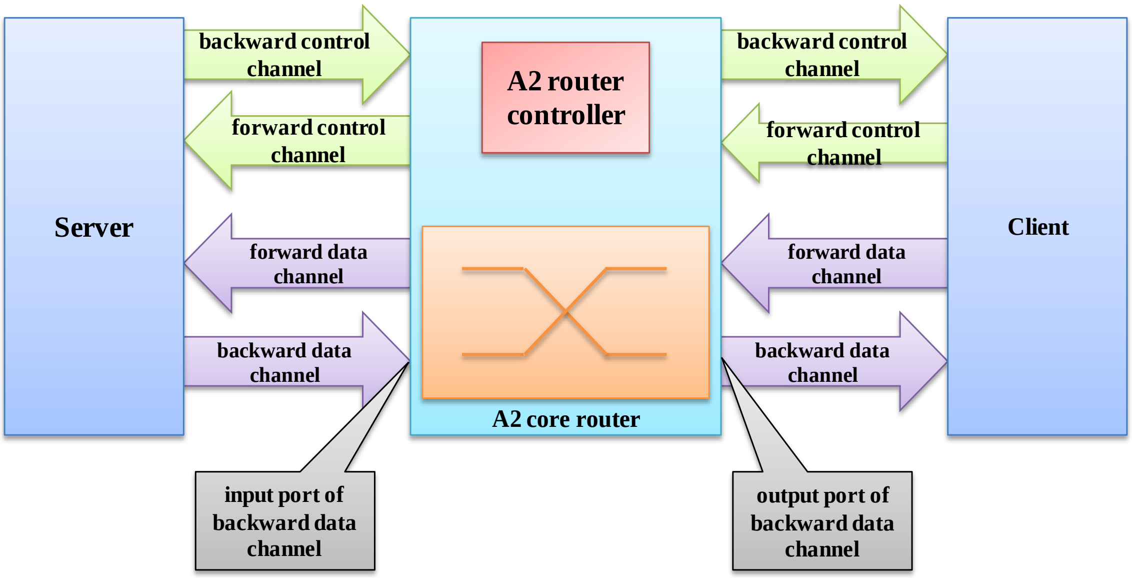


Figure 18: Reverse path reservation

information tells A^2 router to schedule the outgoing link for backward data channel. After decrypting the service type, the scheduler figures out the input link from the service side. At this time, input channel scheduling starts.

When the request is first generated by the edge router at the client side, it is propagated to the next hop core A^2 router first. The core A^2 router reads out the service type information and it utilizes the input backward data link for this service in the future time. The backward data scheduler of this link then utilized the LAUC-VF algorithm to schedule the backward data on to proper optical data path. If the scheduling is successful, this information is written back in to the new header and a new header will be propagated to the next hop. By doing this, the next hop could gain the information of backward data channel chosen by the previous hop and the next hop then knows which output channel the backward data would undergo. At the same time, the core router informs the edge which outgoing channel the service has been granted.

After regenerated header is captured by the next A^2 core router, the core router first decides the backward input link from the database based on the service type provided by the header. Since the regenerated header has the output backward data channel information, the core scheduler then tries to schedule the input backward data channel on to the same wavelength as the output backward data channel. The wavelength conversion is eliminated if the same wavelength has been found. If the scheduler cannot find the same wavelength as the output backward data port, the scheduler will check the possibility of wavelength conversion. When configuration is successful, a new header carrying the input backward data channel on the current A^2 router will be generated by the Header Regeneration Module and it is then transmitted to the next hop through the forward control channel. After propagating through multiple hops, the header eventually reaches the edge on the service side. The service edge reads out the service type information as well as the outgoing data channel and it then fetches data through internal storage device. The data is then packetized and is put in to the backward data channel the header provides.

As to one way path reservation, this mechanism successfully solves the offset time issue traditional OBS router would have. Furthermore, besides the forward control packets which reserve the backward data channel, the A^2 router also support dynamically extension of the service duration. In case the edge cannot transmit data within the reserved time duration, an extension request will be generated by the edge router. This request is then send to the core routers to request for more time to transmit. The core router would try to extend the transmission time by extending the time duration in corresponding channel. If the channel is available for an extended period of time, the request will be accepted and an updated request is then generated by the Header Regenerator Module and is forwarded to the next hop using the backward control channel. Otherwise the request is denied and no modification on the current channel map will be made.

The A^2 router is compatible with Multi-Mode OBS router in the sense that the data path of the edge router can be configured either as the receiving path (forward data channels) for Multi-Mode OBS router or sending path (backward data channels) for application-aware usage. Hence, the edge optical link that connects to the data paths can be configured as either input port of forward data channel or output port of backward data channel.

3.5 Static and Dynamic Service Grouping

Since the exorbitant price wavelength converters cost, it becomes a challenge for the scheduler in the scenario of limited wavelength converters at the outgoing optical fiber links. This means that the incoming channel cannot be tuned to any of the unused channel due to the limited amount of wavelength converters assign to each link.

Due to the cost of wavelength converters, each scheduler only have access to a limited number of wavelength converters or even no wavelength converter depending on the grouping schemes. Therefore, a number of grouping strategies can be chosen to form the A^2 dynamical optical network. The simplest and most straight forward grouping algorithm is to have multiple fibers between each client and server. In this case, each service occupies one optical fiber and the A^2 scheduler use optical links as the switching units rather than using wavelength as the switching units. This switching technique does not utilize the DWDM technology.

The second grouping method makes use of the concept of Multi-Lane OBS. Multiple fibers are still required between clients and servers in the Multi-Lane grouping schemes. However, the traffic can be switched onto the same wavelength of another optical link if the same wavelength on the preferred optical link is not available at the transmission time. This grouping strategy makes use of DWDM while avoiding the need for wavelength converters completely. Instead of requiring expensive optical converters, utilizing a set of dark optical fibers that might have been installed is a cost effective way of achieving the same goal. This consideration can be incorporated into new installations as labor is the dominant factor for network infrastructure expansion. Adding more fibers to existing network infrastructure can be tricky, in which case the service provider must re-engineer the site and interrupt some existing service.

The third grouping method tries to find a balance between cost and performance. In this scheme, only one fiber is connected between client and server but different wavelengths in this fiber can be used for different services. To avoid excessive traffic blocking, each router carries a limited number of wavelength converters for wavelength conversion.

The performance of the third traffic grouping scheme is verified under limited amount of O/E/O converters using the NS-2 simulator. The blocking probability versus the number of electronic switching ports allocated for wavelength conversion is evaluated in the software simulator. The configuration sets every DWDM link to carry 12 backward

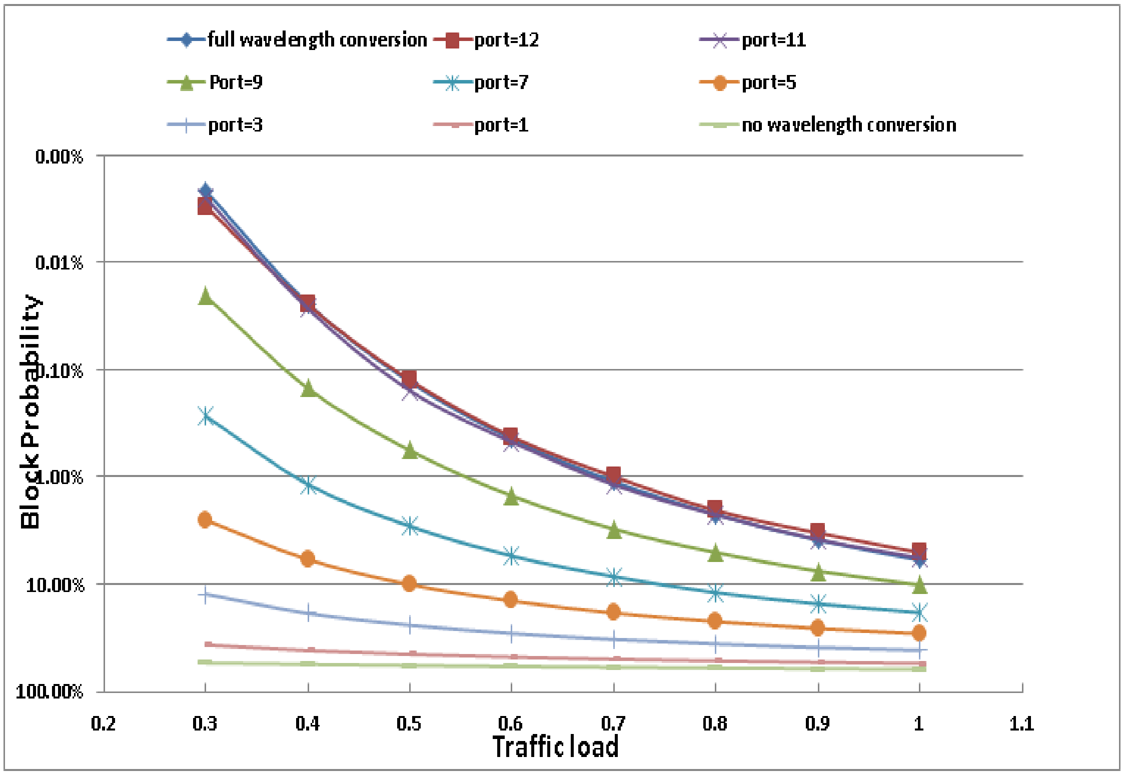


Figure 19: burst blocking probability versus number of O/E/O ports assigned

data channels. The forward control channel is configured so as to avoid artificial traffic drop introduced by control channel scheduling. The bandwidth of each channel is set to 1 Gbps. Traffic arrival follows a Poisson process with traffic rate 1 Gbps during the ON_TIME, traffic rate 0 during OFF_TIME. Define the traffic load as ON_TIME divided by the sum of ON_TIME and OFF_TIME.

The result in Figure 19 shows improvement of blocking probability when more channels are allowed to be shared (via the multi-lane OBS scheme or wavelength converters) in the A^2 router. The result also shows the case where the number of wavelength converters (or shared multi-lane channels) equals to the number of data channels. This is identical to the full wavelength conversion case and it conforms to the Engset formula.

Furthermore, different software simulation configurations are conducted to verify the traffic blocking probabilities of the 50% wavelength sharing case. In these experiments, traffic from several input ports goes to the same destination and the A^2 scheduler tries to schedule the traffic under the 50% wavelength sharing scheme: the scheduler only has half of the wavelength converters than the number of wavelengths. The number of input and output ports is different in each figure. The result shows a deterioration of traffic blocking probability under 50% wavelength sharing scheme. This serves as an indicator when considering the deployment of wavelength converters at each hop in real scenario.

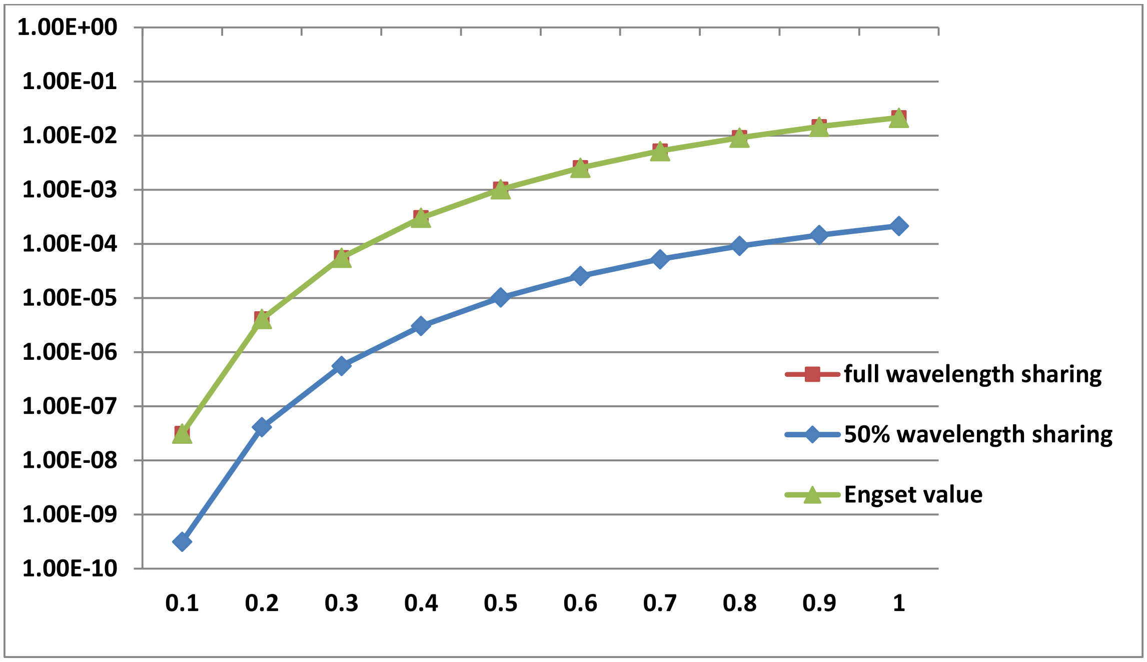


Figure 20: burst blocking probability. 12-in-8-out, 4 wavelength converters

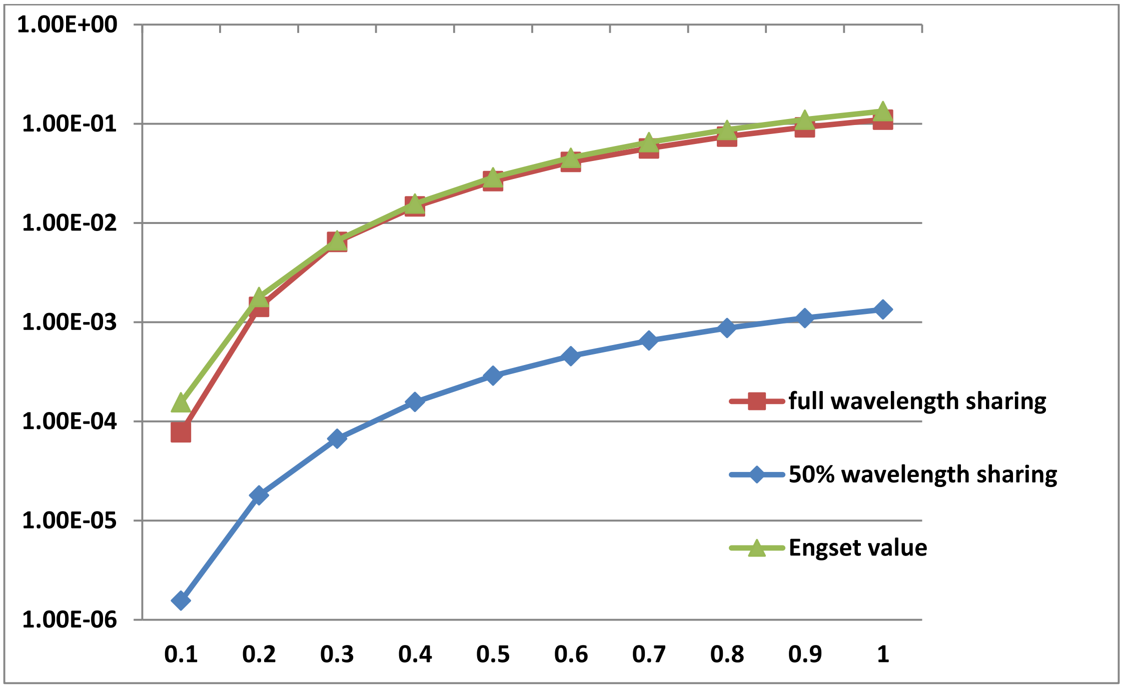


Figure 21: burst blocking probability. 8-in-4-out, 2 wavelength converters

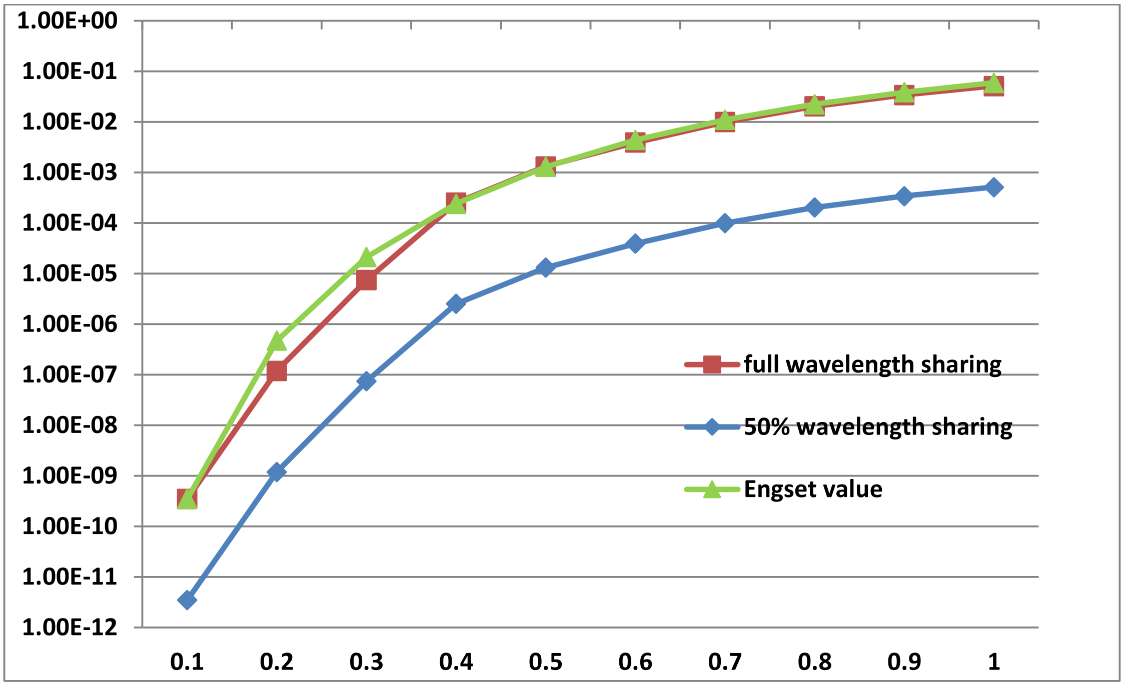


Figure 22: burst blocking probability. 32-in-12-out, 6 wavelength converters

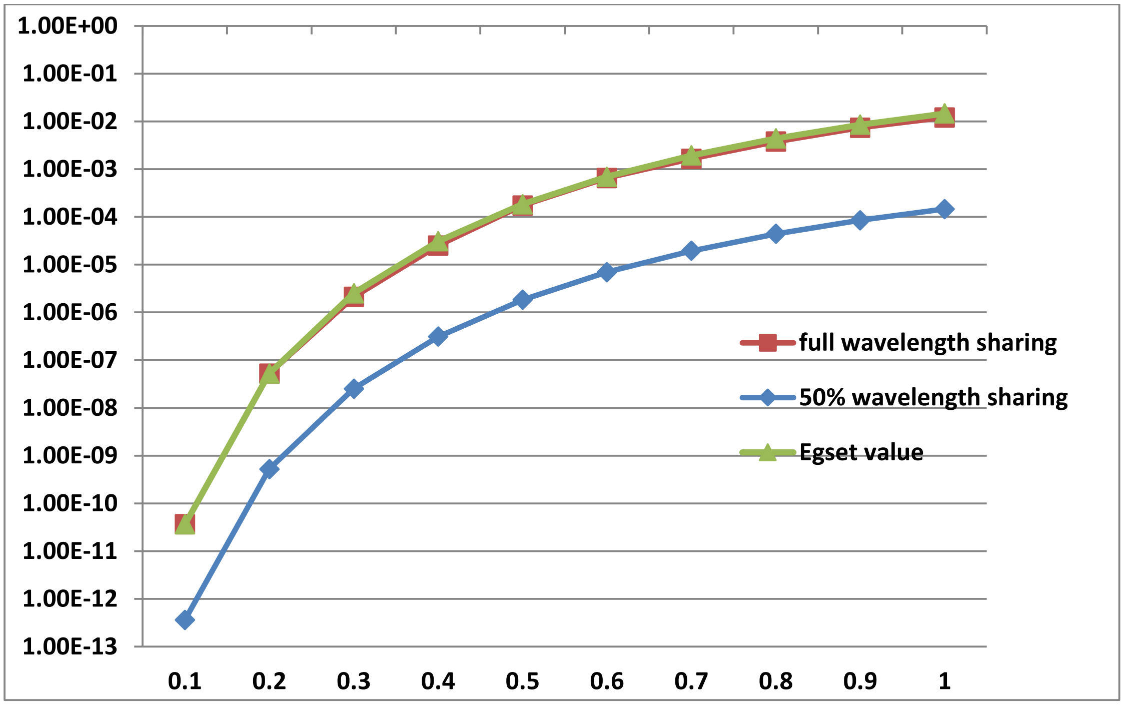


Figure 23: burst blocking probability. 20-in-12-out, 6 wavelength converters

The wavelength converters of A^2 router can be shared among different optical links rather than be shared within one optical links. The O/E/O conversion can be share per link (SPL), which means the wavelength converters are share within one optical link or share per node (SPN), which means the wavelength converters are shared among multiple optical links in the A^2 router. This becomes important considering asymmetric traffic. If the traffic is asymmetric but the scheduler uses SPL grouping scheme, the wavelength converters assigned to idle links cannot be reconfigured to aid the scheduling of the busiest links, causing high traffic drop rate at the busy link. On the other hand, SPN grouping scheme can solve this problem by reassigning other wavelength converters onto the busy optical link. The difference between SPL and SPN becomes prominent when traffic is highly biased or asymmetric. This case is analyzed where one core node needs to deliver traffic to two edge nodes but each edge node has different traffic demand. The experiment gradually increases the traffic load and keep the traffic biased, the bias ratio is 1:9 which is the throughput of destination one versus the throughput of destination two. From Figure 24, the heavier the traffic load, the better the SPN grouping scheme.

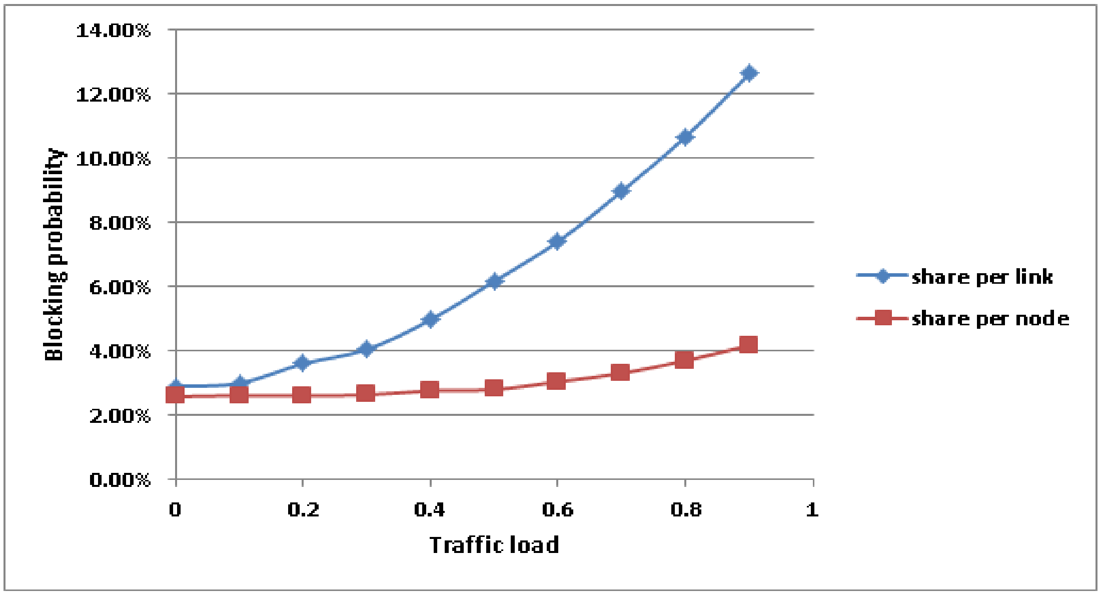


Figure 24: burst blocking probability in shaer per link and share per node configuration

3.6 Hardware Prototyping

3.6.1 Prototyping of router scheduler

For the A^2 router, the most critical aspect is the speed of scheduling. The proposed integrated Multi-Mode scheduling algorithm has been verified in a hardware testbed. The proposed algorithm is implemented in FPGA hardware using Verilog Hardware Description Language (HDL). The FPGA hardware is tested along with the optical switching node in the optical switching testbed. To verify the capability of handling

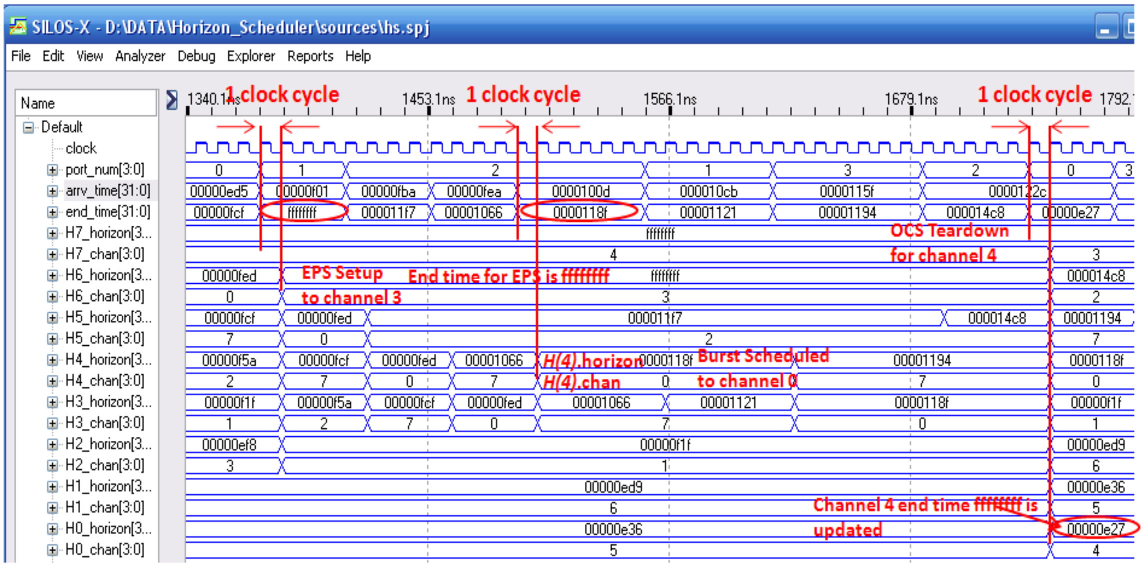


Figure 25: circuit simulation for concurrent EPS, OCS and OBS burst scheduling

Multi-Mode switching connections, several traffic patterns were generated for the circuit simulation of the integrated scheduler module. Figure 25 illustrates the results of hardware simulation of the concurrent operations of EPS setup, OCS teardown and OBS burst scheduling. The configuration scheduling request can be handled by the scheduler with a fixed number of clock cycles (O(1) runtime). In the pipelined implementation, the request can be completed in one clock cycle, independent of the total number of channels (8 in this example).

3.6.2 Prototyping of Dynamic Switching Optical Network

The previous circuit simulation has been verified on the Altera DE2 FPGA board. The decoupling of the scheduling from high speed data path in the A^2 router allows the high performance scheduler to be implemented using low cost FPGA boards coupled with optical switching fabric controller for the core router, as traffic passes through the core router optically. The edge router would require higher processing power due to traffic processing.

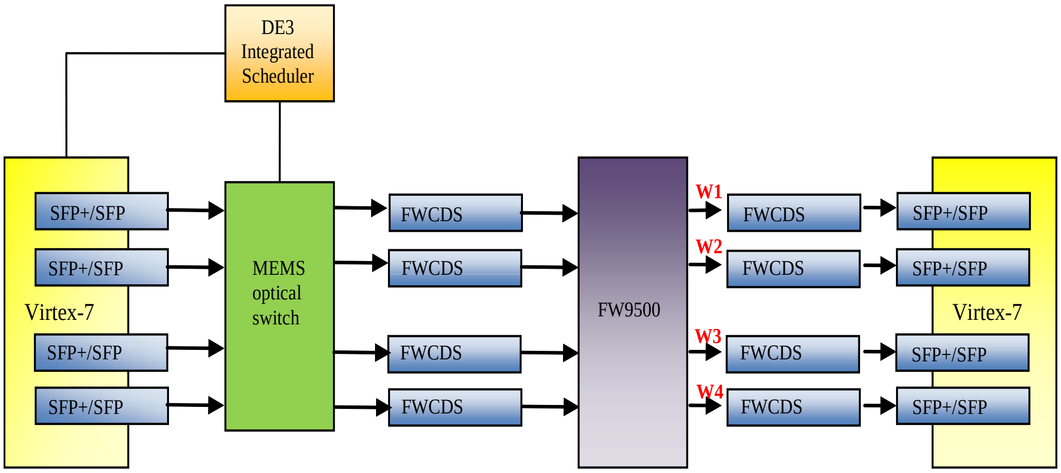


Figure 26: hardware testbed with integrated scheduler controlling custom-built MEMs

As for reconfigurable router controller, a dedicated high performance processing module is desirable: Upon checking and forwarding the necessary packets, the controller needs to be able to configure the switching fabric. In general, packet processing takes place in the embedded system rather than with dedicated circuit, contributing to processing delay. External and Internal memory capacity is another concern when data buffering happens. For prototyping, a hardware board with better communication modules and better memory capacity is preferred. DE3 FPGA evaluation board comes with Altera Cyclone III EP3C120F780 FPGA and its memory rich peripheral components make it a good candidate to implement the router controller.

Figure 26 shows the system architecture and topology of the prototype Multi-Mode Optical Network. Fujitsu FLASHWAVE 9500 Packet Optical Networking Platform (FW9500) is an advanced commercial optical switch. It consists of network interface cards, switching fabric, optical wavelength tuners, optical multiplexers/de-multiplexers (MUX/DEMUX), wavelength selection switches (WSS), and optical amplifiers. Incorporated with the traffic aggregation device, FLASHWAVE Compact Density Switch (FWCDS), different traffic flows are tuned onto different wavelengths in FW9500.

Each Xilinx FPGA board supports 10 SPF+ channels operating at 10 Gbps per channel, acting as edge traffic nodes. Traffic flows are queued individually at the edge based on the configuration profile predefined. Unique traffic flows are assigned onto individual optical channels. The optical channel is attached onto one of the FWCDS and is tuned dynamically onto different wavelengths using the FW9500 IFP5-ETA card.

The constructed optical switch is capable of achieving sub-millisecond dynamic configuration using off-the-shelf Microelectromechanical systems (MEMs) optical switching components and the prototype integrated scheduler described earlier. By dynamically configuring the MEMs components, different traffic flows can share the same wavelength over time, thus achieving the goal of dynamical reconfigurable routing.

3.6.3 Prototyping of Application-Aware Optical Network

The optical switching time is a critical factor to achieve fast and reliable switching performance in Application-Aware optical network. Due to manufacturing limitation, there is a 500 us switching delay in the MEMs switch used in the prototype. In the meantime, there are 70 clock cycles control packet processing overhead at the edged node and in the integrated scheduler. This adds 1.12 us to the configuration time with FPGA circuit running at the clock frequency of 125 MHz. Nevertheless, the total delay of setting up an optical data path is well below 1 ms, achieving practical sub-millisecond dynamic optical switching. With the cascaded connection of multiple optical switches and the parallel controller design, such a sub-millisecond switching target can be maintained for a large number of optical ports.

An application triggered application-aware dynamic optical path setup experiment is conducted to demonstrate its capability. More specifically, in Figure 27, three routers are in the same subnet while router 1 and router 2 are isolated by customer VLAN tag from router 3. Server connected to 3Com switch streams two separate medical image traffic simultaneously. The experiment swaps the SVLAN tag by routing the traffic to FWCDS1 and FW9500. At the egress port, iPads function as clients and they get the traffic from FWCDS2. The high resolution medical images are displayed at the both client sites.

On the server side, requests from iPads automatically trigger the dynamical setup and teardown of the optical path by sending control packets to DE3 FPGA board, which configures the optical switch upon receiving the control packets. At the same time, server would fetch the necessary medical image data while the path is being configured. This demonstrates the idea in A^2 dynamic optical network and it has successfully demonstrated the viability of sub-millisecond dynamic optical path setup. From the user's perspective, the path setup time is invisible.

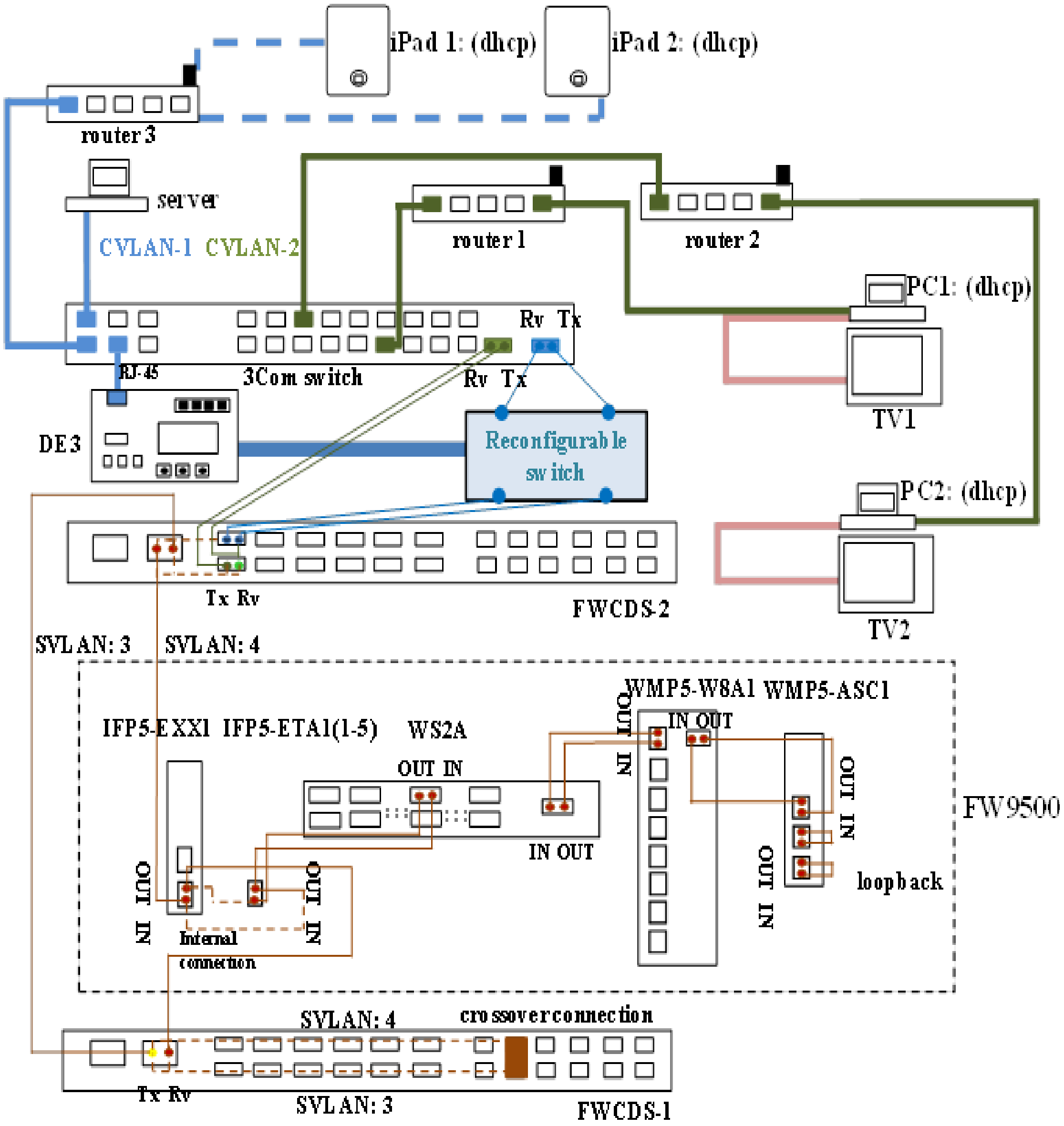


Figure 27: system architecture of application triggered dynamic optical path setup

3.6.4 Optical Signal Attenuation in A^2 Optical Switching Network

Adding optical MEMs switches requires optical signal coupling between two or more optical components, which inevitably introduces optical power loss. By measuring the optical power loss per hop, the maximum number of cascaded optical switches can be determined based on performance metrics. Table 2 shows the optical power monitoring results averaged over 15 minutes periods at different insertion points. The optical switch insertion attenuation is measured individually between FWCDS and FW9500 10GigaEthernet interface (IFP5-EXX), FW9500 Wavelength Tuner (IFP5-ETA) and the ingress port of optical MUX (WS2A), as well as the egress port of optical MUX and the ingress port of wavelength selection switch (WMP5-W8A1C). The experiments showed that optical power loss varying between 0.2dB and 0.4dB with single optical switch insertion point.

Table 2: optical power measurement

Device Insertion Point	IFP5-EXX (A) 1-4-2	IFP5-ETA (B) 1-5-E1	WMP5- W8A1 C (C) 1-15-C1-12	WMP5- W8A1 E (D) 1-15-E1-12	WMP5-ASC1 C (E) 1-14-C1
	OPR (dBm)	OPT (dBm)	OPR (dBm)	OPT (dBm)	OPR (dBm)
No Insertion Point	-6.8	4.0	-4.0	-19.8	-20.3
Between CDS and A	-7.1	4.0	-4.0	-19.8	-20.2
Between B and C	-6.8	4.0	-4.4	-19.5	-19.8
Between D and E	-6.8	4.0	-3.9	-19.8	-21.0

Cascaded connections of multiple optical switches fit into the A^2 optical switching network scenario where multiple hops are connected to form an optical path for the data stream. In order to guarantee the signal attenuation within the allowable range, three additional optical couplers are added at the egress port of MEMs optical switch. The measurements show 0.1 dB – 0.2 dB attenuation introduced by each coupler.

Figure 28 illustrates the experiment that shows the optical power attenuation where 1 to 5 MEMs switches are cascaded. An average of 0.3dBm optical power loss can be observed. The experiment serves as a guide when building optical network using MEMs switching as the switching matrix in A^2 router.

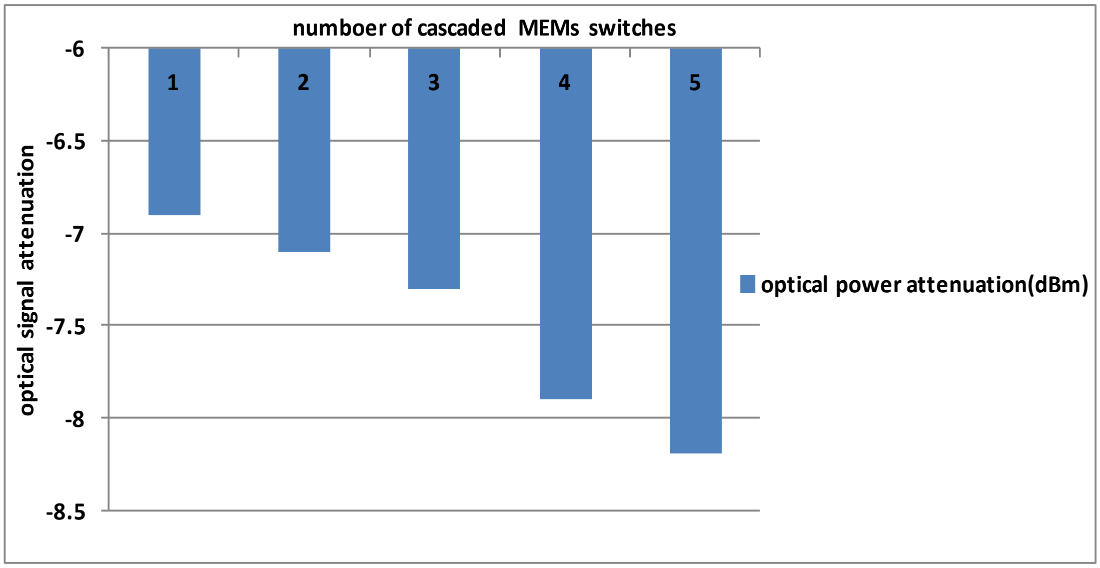


Figure 28: optical power attenuation measurement of cascaded MEMs switches

3.7 Summary

The Multi-Mode switching offers unified solution to support multiple switching modes on the same DWDM platform, solving the issue of integrating switching among different traffic modes. The application-aware dynamic optical switching on the other hand provides application level provisioning, creating a novel optical switching architecture that maximizes application level Quality of Experience (QoE). Dynamic optical path setup in the proposed application-aware optical networks has been demonstrated with hardware prototyping testbed, which realizes functions of the core and edge routers. The static and dynamic service grouping strategies have been explained and verified. The result shows the accuracy of the simulation configuration and it leads to a better understanding of the principles of the traffic sharing in the A^2 optical network.

The proposed Multi-Mode scheduler has been implemented in FPGA hardware and the A^2 sub-millisecond optical path setup testbed demonstrates the correctness of A^2 optical network architecture. A truly dynamic optical network can be built with off-the-shelf optical components.

Chapter 4 3-D Optical Switch

4.1 Motivation

Currently, the market offers a variety of optical switches. Most of those products aim at optical circuit switching networks. The switching technology allows traffic to be sent through optical fiber medium with relatively low cost. This technology meets the industrial needs by leasing static optical lines to customers. The performance is good in terms of throughput and reliability but when taking a closer look at the network component, there are some problems.

First of all, adding an optical fiber comes at a very high cost. A significant amount of truck roll is required when adding or modifying the current optical switching configuration. It is a complex task to reengineer the entire traffic station in order to add a new optical path or reroute the existing traffic to a different destination. Due to the complex engineer work, it also brings a tremendous operational cost. In fact, in many commercial optical switching plans, it is very hard to accommodate new network traffic on the go. Much of the ongoing traffic has to be turned down in order to do a small modification.

From another point of view, the network needs to be upgraded periodically to meet the latest technology. Current commercial switches utilize optical switching fabric with small port counts in the optical switching panel. However, the entire optical switching fabric has to be replaced if it is upgraded to support more ports. This type of optical switch is not cost efficient and upgrade friendly architecture is an important design metric for the next generation optical switch.

Designing a new agile optical switching platform that features dynamic fast customized modification becomes an urgent issue. In Figure 29, the orange circles represent the proposed elevator switch and L1-L4 represents various traffic paths. Traffic with different destinations at the ingress node can be differentiated by optical device. The edge router diverts traffic onto network ports based on the traffic path number. At the same time, the router tries to configure the outgoing traffic path for this traffic. As shown in green lines in Figure 13, when traffic from L2 comes to the optical switch, the switch will try to make connections to the destination. If the path setup is successful, the traffic will be routed to the corresponding network ports. The traffic is then transmitted through the configured path. Otherwise, the traffic can be routed to an intermediated node based on the predefined routing criteria. When traffic arrives at the destination, the optical switch decouples the signal and delivery packets to the final client port.

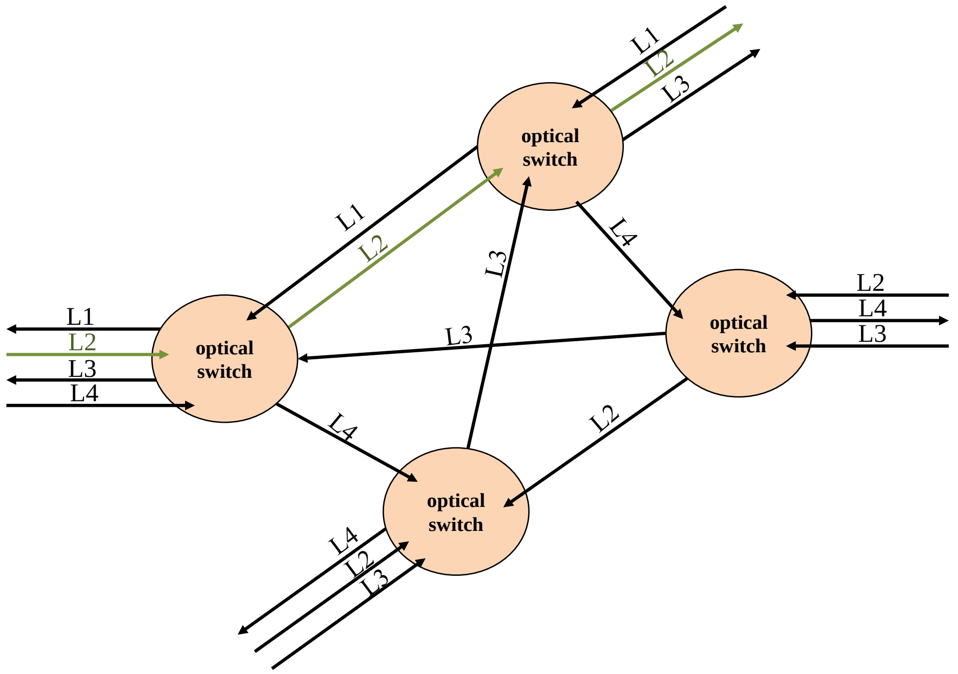


Figure 29: optical switching system

4.2 Proposed 3-D Switch

In fact, every traffic aggregation point only has one receptor and it corresponds to one wavelength tuner. Problem stands out when multiple network traffic point to the same destination.

Figure 30 illustrates an example of the above problem: in order to handle two traffic streams that go to one destination, another pair of wavelength tuner and two traffic aggregators need to be used. But as known to all, the wavelength tuner comes at a high cost. Moreover, the resource is occupied by the extra channel but in general these channels are only in use from time to time, causing bandwidth underutilization.

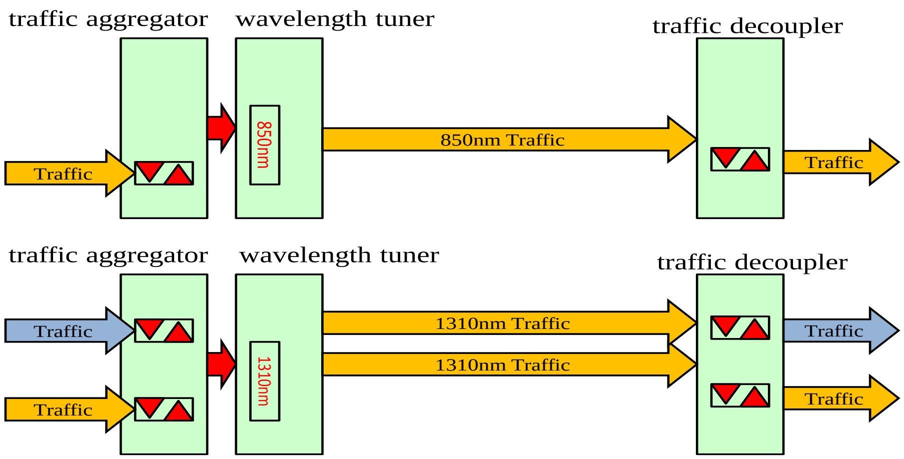


Figure 30: example of two traffic stream with one destination

In essence, if the traffic is not interfered with each other across the space domain, building a switch that could handle multiple traffic streams layer by layer is more desirable. The layer allows the traffic source to even be virtual source contains a number of sub sources inside the traffic itself.

Figure 31 shows the idea: the wavelength could be shared over three traffic sources. Rather than one traffic receptor, the traffic aggregator now could have as many receptors as possible as long as the QoS on every traffic stream can be satisfied. By allowing traffic to share the wavelength over time, the extra resource needed is reduced.

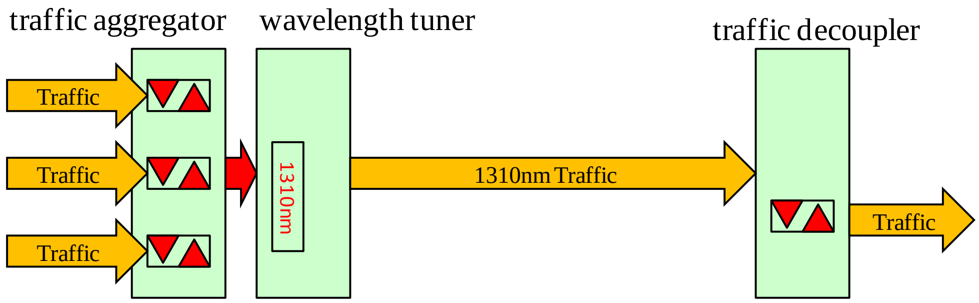


Figure 31: example of multiple traffic streams with one destination

To further extend this problem, suppose there are *M* traffic sources and every source wants to reach one of the *M* destinations. To make the switch fully connected, *M***M* channels are required. Besides, the traffic has to be redirected periodically when different destinations need to be reached.

The proposed 3-D Switching can eliminate the need of multiple wavelength tuners to its best effort. In the meantime, this mechanism could greatly increase the cost efficiency ratio comparing with traditional optical switching approach. In the proposed elevator switch routers, a link consists of multiple wavelengths, each of which carries multiple wavelengths. Any of the input wavelengths can be switched to the same wavelength on any desired outgoing port. By carefully planning the size of each lane, high statistical multiplexing can be achieved with a relative low cost.

Figure 32 illustrates a general 3-D Optical Switching Network where every 3-D switch is connected by multiple links. In this figure, each link at the edge carries *M* wavelengths *λ1,…,λw*. The edge link can be constructed using a collection of optical cables or using DWDM technology to multiplex different optical carrier signals.

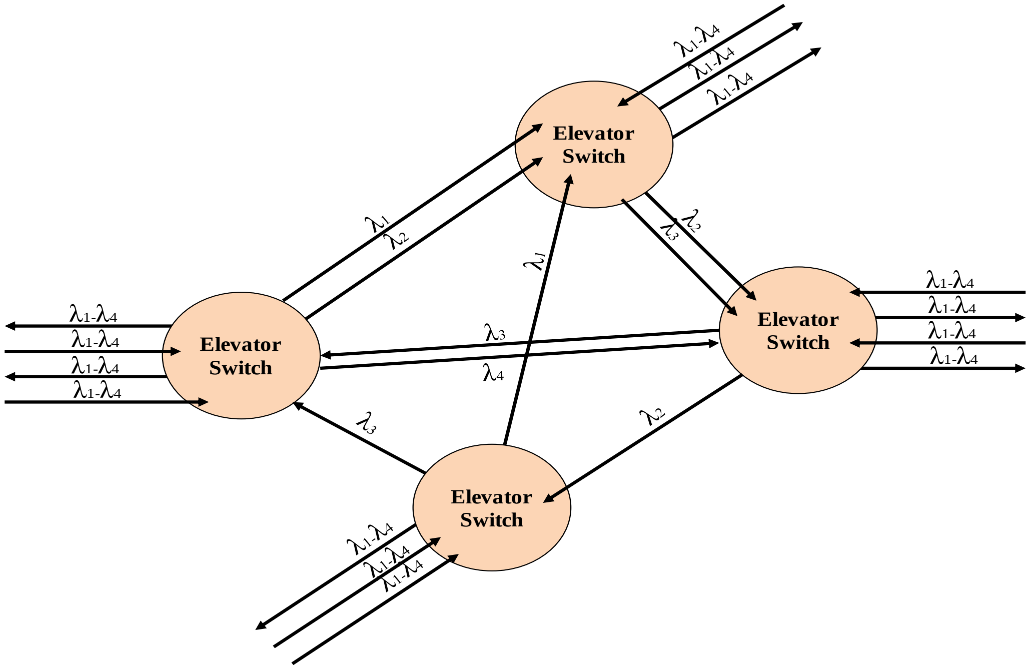


Figure 32: 3-D switching system

The 3-D switch control protocol is compatible with the mechanism used in DWDM Multi-Mode Switching. On receiving a control header, the 3-D switch selects an idle channel on the outgoing link for the data corresponding to this header. The header information occupies a separate wavelength for each outgoing destination. In addition to the general control header information such as the length of the data, the offset information, the incoming wavelength and the quality of service information of the data, the DWDM Multi-Mode control header contains traffic type information. They are EPS, OBS or OCS. As well, the Link ID specifying which lane the traffic was sent will be carried in the header.

The proposed elevator switch system provides statistical multiplexing for same wavelength from different traffic links without using excessive wavelength converters. Since the signaling is compatible with DWDM Multi-Mode switching, which incorporates OBS signaling, the proposed 3-D Switching System can be easily adapted to traditional OBS routers without adding additional components. From this point of view, the 3-D Switch is a cost effective way to realize OBS networks. Comparing the OBS network which needs extensive wavelength conversions in OBS router, the 3-D Switching System only needs multiple fibers that running through the network. Therefore, the cost of Elevator Switching System is much lower than the traditional OBS router with wavelength conversion. In addition to savings on the wavelength converters, the 3-D Switching System is expendable, which indicates that to upgrade the Elevator Switch, the user only needs to expand the switching matrix rather than replacing the entire switching system.

The overall Elevator Switch architecture can be used for the DWDM Multi-Mode router. To formalize the architecture, following notations are used:

L: Number of input/output links per port

W: Number of wavelengths per fiber.

F: Fibers counts per link.

Lij: The *j-th* fiber on the incoming/outgoing link *i*.

$\lambda_{ij}(e)$: Wavelength e on *Lij*.

N: Numer of ports of the optical switching plane

O: Numer of incoming/outgoing ports

M: Numer of switch layers

S_w: The *w-th* optical switching layer

P_{wd}: The *d-th* port on the *w-th* optical switching layer

Figure 33 illustrates the high level system block diagram of the 3-D switch. In this figure, each of the multi-layer optical switches is connected to L incoming as well as L outgoing links. Each port contains P fibers and each fiber carries W wavelengths. A burst from wavelength $\lambda_{ij}(e)(1<i<P, 1<j<L, 1<e<W)$ can be routed to $\lambda_{ij}(e)(1<i<P, 1<j<L)$ on any of the outgoing fiber.

There are different configuration mechanisms towards building the optical crossbar switch. Figure 33 is one way to construct the optical matrix. At the input of the 3-D switch, fibers from each link are fed to their respective optical demultiplexer (DEMUX). The DEMUX separates the optical signals on different wavelength. Individual single band optical signals are fed to different layers of the switching matrix.

At the outgoing port of the optical switching matrix, wavelengths at different layer are then combined together using optical multipler (MUX). Since each switching matrix layer is connected to one of the W wavelengths, there are W switching matrix layers in total for all the wavelengths in one fiber. Therefore,

$$M = W. \tag{1}$$

Based on the connectivity rule, wavelength $\lambda_{ij}(e)(1<i<P, 1<j<L, 1<e<W)$ on any given fiber is connected to the e-th layer of switch matrix S_e, it is intuitive that $w=e$. Based on the connectivity rule, the total number of links determine the minimum number of ports on each switch matrix layer equals

$$O * N = \lfloor d \rfloor. \tag{2}$$

Wavelength $\lambda_{ij}(e)(1<i<P, 1<j<L, 1<e<W)$ on L_{ij} is connected to $P_w(O\text{-}1)*N+j$.

The output ports are the reverse of the incoming ports, and the same connectivity method can be applied. The configuration derived above can be used for scheduling as well as for building switching matrix.

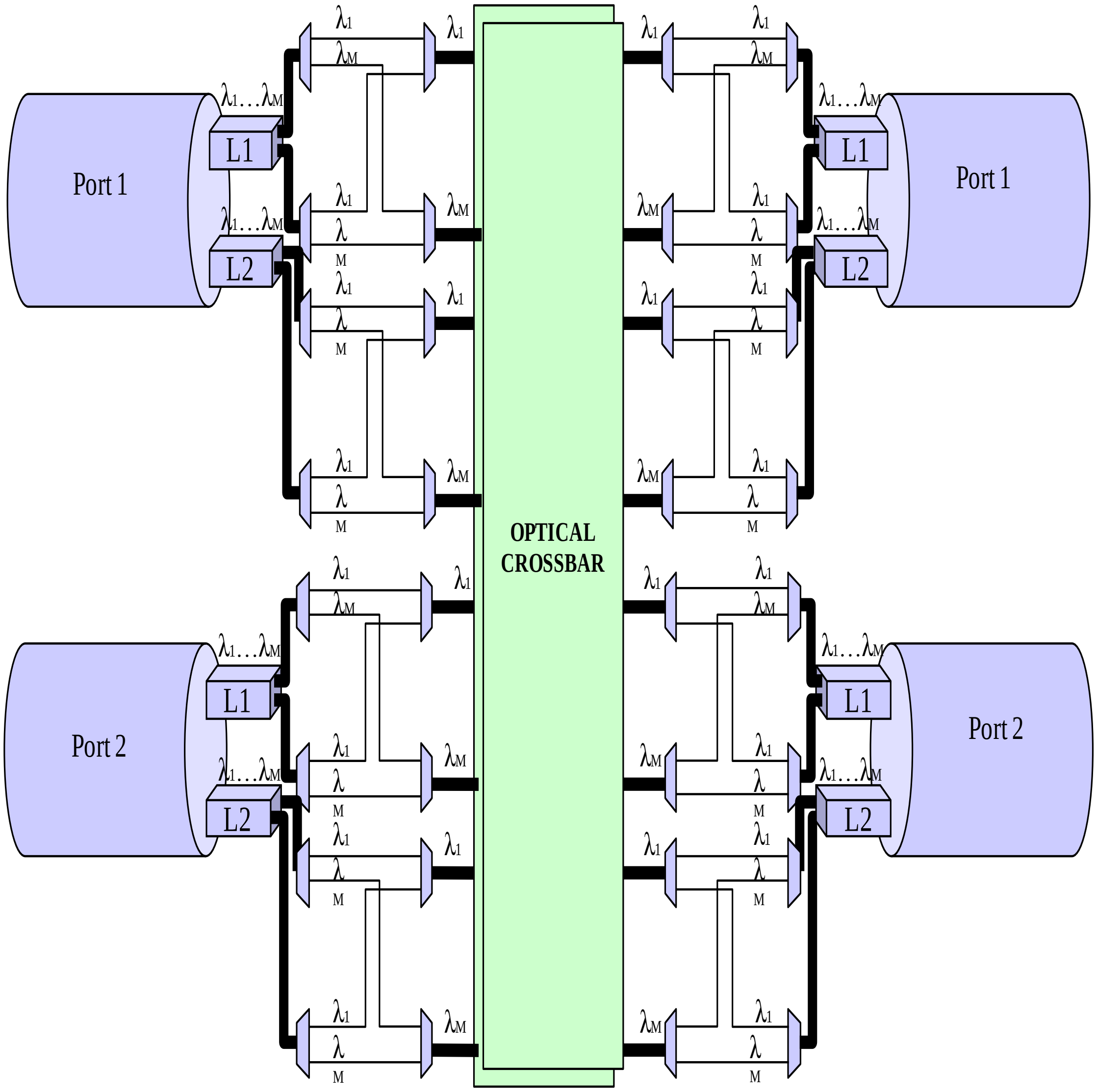


Figure 33: 3-D switching block diagram

The 3-D switch edge router multiplexes different traffic streams in the time domain. Not only the traffic could go through the line card according to destination, but also the switching fabric could be dynamically reconfigured in real time.

The 3-D switching consists of three main parts. They are ingress/egress edge, switching fabric matrix and optional routing controller in advanced dynamic reconfigurable elevator switching system.

Figure 10 illustrates the general architecture of the 3-D Switching System where an *M*-row *N*-column switching fabric matrix is connected to *N* edges. Each switching fabric matrix layer is constructed by 2×2 MEMs switch cell where these cells together form an $N \times N$ switch.

The formation of the switching layer could utilize either blocking or non-blocking mechanism depending on the application occasions and network scenarios. Let the number of MEMs switches needed at each layer be *k*, and the total number of switches required is $M \times k$.

Every edge has *M/2* optical transceivers connects to the switching fabric side and *M/2* optical transceivers connect to the client side. The edge contains *M/2* levels and at least one dynamic reconfigurable (DRP) port. The *N*-input dynamic configure controller in switching fabric matrix accepts configuration commands issued at every edge node. It configures the state of every MEMs switch in parallel through $M \times k$ I/O pins when valid commands and available paths are both captured.

4.3 Static 3-D Switching

We propose two types of 3-D switches: static 3-D switch and dynamical 3-D switch. In the static pipelined 3-D switching system as shown in Figure 34, the edge node functions as traffic aggregator as well as traffic classifier. On one hand, the ingress portion of the switching system edge aggregates traffic through preconfigured client ports. The client refers to general Internet traffic both from personal computers, cellular devices and complex servers and cloud structures. On the other hand, every destination in the switching system edge classifies traffic as two groups based on the destination address located in the traffic header: local group and remote group. If the associated destination address assigned to the elevator cell is identical to the destination address of the incoming traffic, the egress edge will mark the traffic flow as local group. In this case, traffic will be forward to the attached outgoing optical transceiver directly. If the pipelined packet classifier could not find a matching destination entry, the traffic is considered as remote

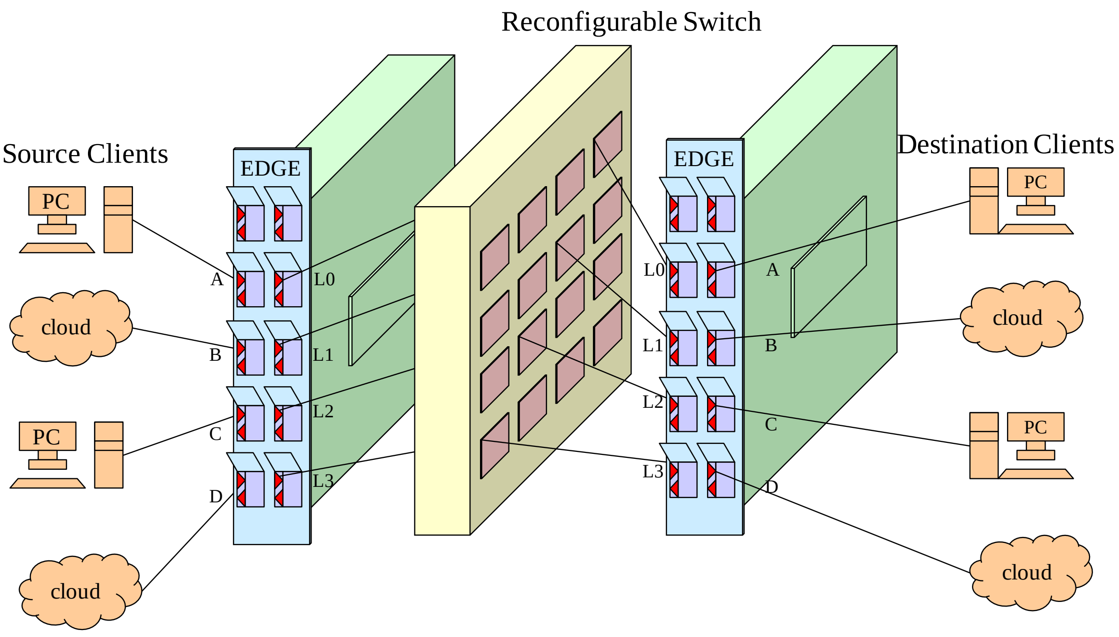


Figure 34: static pipelined switching system

group and will be forward to the next elevator cell and so on so forth. By implementing this scheme, traffic with different destinations will be circulated in the pipe until the matching destination found.

Because the switching fabric matrix is manipulated in layers and every edge could have access to any of the layers, we are expecting a much less blocking probability with low resources utilization. Comparing with the Benes network that requires *Nlog2N - N/2* 2×2 crossbar switches to building a rearrangeably non-blocking network, the proposed switching mechanism only need a fraction of the crossbar switches under the support of ingress/egress edge node.

Figure 35 is a comparison of number of basic element MEMs switches need to build an identical scale switch between Bens rearrangably network and the proposed 3-D switching network. When topology enlarges, the number of MEMs elements needed increases exponentially in bens network. However, using proposed scheme the element number increase with the functional of logarithm.

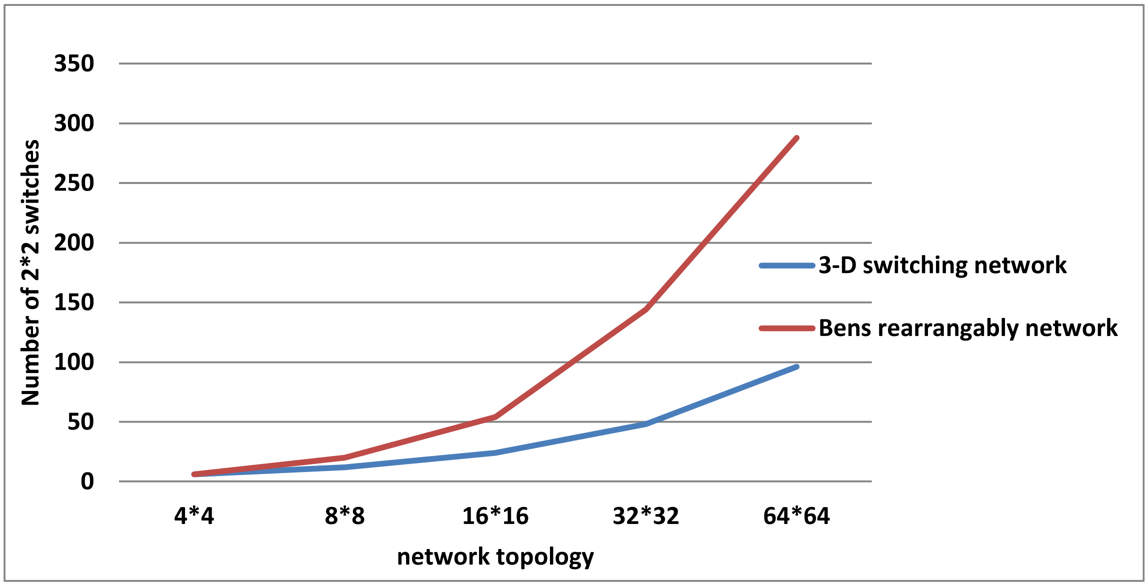


Figure 35: MEMs switches usage comparison between bens network and proposed 3-D network

4.4 Dynamic Elevator Switching

Since every elevator cell is capable of connecting to one of the several layers in the optical switching matrix, it tries to setup the optical path and deliveries traffic at its best effort whenever all of the following three conditions are met: the current optical path is not occupied; there is no optical path in other layers that could reach the designated destination; the designated destination could be reached by dynamically configuring the optical path at current switching fabric matrix layer.

Besides the two main tasks, in the advanced dynamic reconfigurable elevator switching system, the switching fabric matrix controlled by edge node will setup and tear down the optical path on demand. The path setups is not necessary one to one mapping which means the high priority traffic could have more than one optical path, making the connection more flexible in asymmetric traffic case. Borrowing the concept in Multi-

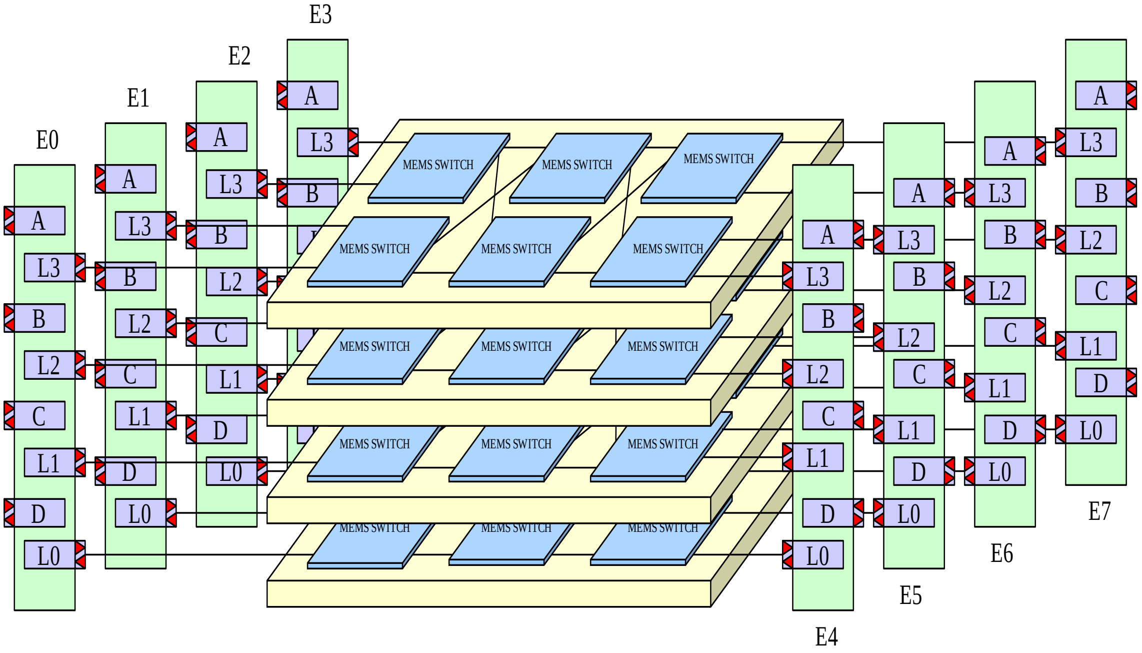


Figure 36: Overview of elevator switching architecture

Mode OBS, different modes of traffic can coexists in the same platform without conflict.

Consider the case in Figure 37, every line card has 9 ports and there is at least one Dynamic Reconfigurable Port. The reconfigurable router controller (RRC) ports keep track of the current configuration of the MEMs switch and it takes the valid configuration command from DRP. The MEMs switch can be configured by the RRC so that the traffic can be routed to the destination. Suppose at time t1, there is a packet goes in to port D at Edge 0, and its corresponding destination is E7 with destination port A. The packet header will be scanned in one clock cycle time. At the same time, the RRC will check its own routing table. There are three conditions that could happen.

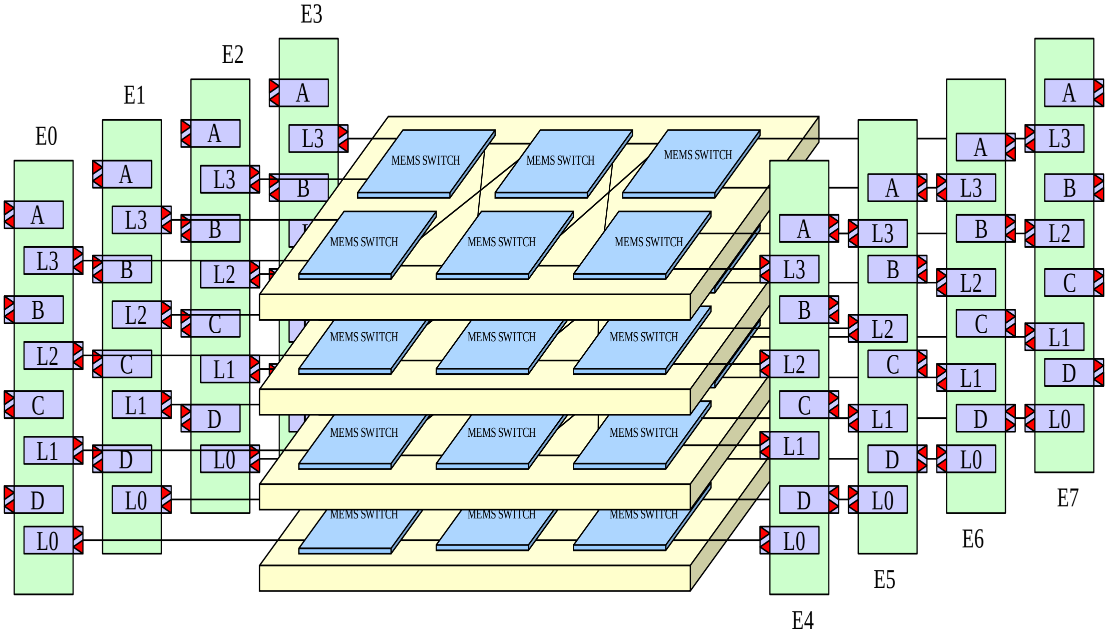


Figure 37: Overview of 3-D switching architecture

If any route between E0 and E7 has been found through the table lookup, the corresponding network port in E0 will be configured and traffic will be routed to that port. For example, if there is a path between E0 and E7 through port L0 on E0 to port L0 on

E7, the bottom switch layer in MEMs switching fabric is selected during this routing period. Traffic will be forward from port D on E0 to port L0 on E0 in line card E0.

If no route between E0 and E7 has been found through table look up but there are idle layers in the MEMs switching fabric, the RRC will add one entry into the routing table and configure the optical switch fabric immediately. In the meantime, the corresponding network port in that layer in E0 will be configured and traffic will be routed to that port. For example, if the bottom switch layer in MEMs switching fabric is occupied but the second layer can be configured, the path will be routed through RRC and a route between E0 and E7 through port L1 on E0 to port L1 on E7 is established. Traffic will then be forward from port D on E0 to port L1 on E0 in line card E0.

If no route between E0 and E7 is found through table lookup and all layers are occupied after one inquiry iteration, the packet will be dropped since no resource is available for traffic transmission. After transmitting through MEMs switching, traffic needs to reach corresponding port on E7. When network port on E7 receives traffic that is tagged with destination D at its own edge, it will be forwarded to destination D directly. In these three steps, the traffic arrives at the destination correctly. Table 3 is the corresponding 16×16 elevator switching routing table in Figure 37. One may observe that any port of the ingress edge in Figure 11 is able to access any port of the egress edge with

Table 3: 16x16 elevator switching routing table

ingress	egress	ingress	egress	ingress	egress	ingress	egress
E0L0	E4L0	E1L0	E4L1	E2L0	E4L2	E3L0	E4L3
E0L1	E5L1	E1L1	E5L2	E2L1	E5L3	E3L1	E5L0
E0L2	E6L2	E1L2	E6L3	E2L2	E6L0	E3L2	E6L1
E0L3	E7L3	E1L3	E7L0	E2L3	E7L1	E3L3	E7L2

correct manipulation of the small scale non-blocking optical switches at every layer.

4.5 Dynamic Reconfigurable Router Controller Algorithm

The DRP controller stores a separate routing table for all the channels. The configuration takes place at each layer when a valid configuration command has been received. To simplify the problem, assume that MEMs switch system is composed of Multiple 4×4 non-blocking sub-switching system. A general configuration algorithm is provided.

First of all, the RRC needs to store the channel map. The channel map predefines which MEMs switch in the sub-switching system needs to be configured in order to route path from the ingress port to the egress port. For 4×4 non-blocking sub-switching, there are two possible routes to route the path from one ingress port to one egress port. Therefore, there are two sets of channel maps available, each set stores one routing path.

Figure 38 shows a sub-switching system and Table 4 is the corresponding channel map. The first two columns in Table 4 are indicators of the corresponding input and output ports. The columns from A to F in Table 4 represent the configuration status on 2×2 MEMs switches in Figure 38. A1 in the column means that the MEMs switch should be configured and 0 means there is no need to configure the MEMs switch. The columns

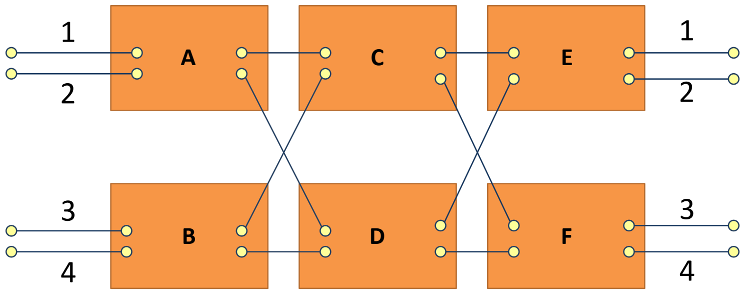


Figure 38: 4x4 non-blocking sub-switching system

from SA to SF in Table 4 represent the configuration mode on 2 × 2 MEMs switches in Figure 38. A1 in the column means the MEMs switch should be configured to the cross state and 0 means to configure the MEMs switch to the bar state.

Table 5 shows the two channel maps associated with the 4 × 4 non-blocking sub-switch systems. When a configuration request is captured by RRC the in/out port from the packet will be captured. The information is used in routing table lookup. If the RRC finds out the range of the in and out port from the packet exceeds the allowable routing range, a deny signal will be transmitted back to DRP and the packet will be dropped since the configuration command is invalid. There are three steps to configure the switching fabric if a correct input and output port pair is found from the routing table.

Beside the pre-stored channel map, there are two basic information types the RRC would have to use when configuring every MEMs switch: whether the MEMs switch needs to be configured and the state of the MEMs switch should be configured to. The RRC has another table called Status Table. Table 6 is an illustration of this table. The RRC keeps the record of configuration status and state status in every switch and it will update this table periodically whenever a configuration takes place

Table 4: 16x16 elevator switching request entry

IN	OU T	A	B	C	D	E	F	SA	SB	SC	SD	SE	SF
1	1	1	0	1	0	1	0	0	0	0	0	0	0

.

Table 5: 16x16 elevator switching channel map table

IN	OUT	A	B	C	D	E	F	SA	SB	SC	SD	SE	SF
1	1	1	0	1	0	1	0	0	0	0	0	0	0
1	2	1	0	1	0	1	0	0	0	0	0	1	0
1	3	1	0	1	0	0	1	0	0	1	0	0	0
1	4	1	0	1	0	0	1	0	0	1	0	0	1
2	1	1	0	1	0	1	0	1	0	0	0	0	0
2	2	1	0	1	0	1	0	1	0	0	0	1	0
2	3	1	0	1	0	0	1	1	0	1	0	0	0
2	4	1	0	1	0	0	1	1	0	1	0	0	1
3	1	0	1	1	0	1	0	0	0	1	0	0	0
3	2	0	1	1	0	1	0	0	0	1	0	1	0
3	3	0	1	1	0	0	1	0	0	0	0	0	0
3	4	0	1	1	0	0	1	0	0	0	0	0	1
4	1	0	1	1	0	1	0	0	1	1	0	0	0
4	2	0	1	1	0	1	0	0	1	1	0	1	0
4	3	0	1	1	0	0	1	0	1	0	0	0	0
4	4	0	1	1	0	0	1	0	1	0	0	0	1

IN	OUT	A	B	C	D	E	F	SA	SB	SC	SD	SE	SF
1	1	1	0	0	1	1	0	1	0	0	0	1	0
1	2	1	0	0	1	1	0	1	0	0	0	0	0
1	3	1	0	0	1	0	1	1	0	0	1	0	1
1	4	1	0	0	1	0	1	1	0	0	1	0	0
2	1	1	0	0	1	1	0	0	0	0	0	1	0
2	2	1	0	0	1	1	0	0	0	0	0	0	0
2	3	1	0	0	1	0	1	0	0	0	1	0	1
2	4	1	0	0	1	0	1	0	0	0	1	0	0
3	1	0	1	0	1	1	0	0	1	0	1	1	0
3	2	0	1	0	1	1	0	0	1	0	1	0	0
3	3	0	1	0	1	0	1	0	1	0	0	0	1
3	4	0	1	0	1	0	1	0	1	0	0	0	0
4	1	0	1	0	1	1	0	0	0	0	1	1	0
4	2	0	1	0	1	1	0	0	0	0	1	0	0
4	3	0	1	0	1	0	1	0	0	0	0	0	1
4	4	0	1	0	1	0	1	0	0	0	0	0	0

Table 6: 16x16 elevator switching status table

A	B	C	D	E	F	SA	SB	SC	SD	SE	SF
0	0	0	0	0	0	0	0	0	0	0	0

To configure a single layer of MEMs switch, the in and out ports of the incoming packet will be scanned by RRC and two entries from channel map shown in Table 5 is then selected as the candidate entries. Each entry is then divided into two parts towards two separate procedures.

The first part of the one of the two channel map entries is from column one to column six: it indicates which of the switches needs to be configured. The RRC will perform an AND operation between these six bits and the first part of the status table. The reason for doing the AND operation is to detect whether there is any possible conflict in configuring the MEMs switch. For example, if there is a one in any of the six outcome bits, then it is possible to have a configuration conflict. The reason is that from the status table, bit one means the MEMs switch has already been configured, but the channel map may tell the RRC that this MEMs switch needs to be configured. However, it is also possible that the RRC will configure this MEMs switch to the same state as the MEMs switch current stays, which leads to the second step.

The second step is to detect the possible configuration conflict mentioned previously. The RRC takes the second part of the channel map entries, and it will perform an XOR operation between this part and the second part of the channel map entry. The idea of XOR operation is to identify a possible configuration conflict. If there is no configure change on current MEMs switch, the outcome of XOR operation should be zero. Otherwise, the outcome of this operation should be one.

The third step is to finally determine whether the configuration can be successfully accepted or not. An AND operation is performed in this step on the previous two outcomes. If the outcome is zero on all bits, there is no configuration conflict. Otherwise there will be a conflict and the channel scheduling cannot be completed.

The AND operation tests the condition that if the previous configured MEMs switch needs to be reconfigured and if this MEMs switch is going to be configured to a new status. If the result is yes (one appears in the result array), then there is going to be a configuration conflict since there is no resource to configure the path without losing the existing configured route. The packets have to be dropped at this layer.

One thing worth mentioning is that if FPGA is chosen to execute this function, Step one, two and three can be processed with minimum delay by using a simple combinatory logic circuit. Moreover, when configuring the MEMs switch, the switches are configured simultaneously rather than being configured one by one. This can dramatically decrease the configuration time of the 3D Elevator switch.

Figure 39 shows a flowchart of the proposed algorithm. As mentioned above, first, the RRC checks whether the incoming packet is out of range or not, and then the message is split into two parts for special processing. In the end, the AND operation is performed and the result is then fed into the decision module. The decision module takes care of the MEMs switching system configuration and status table update.

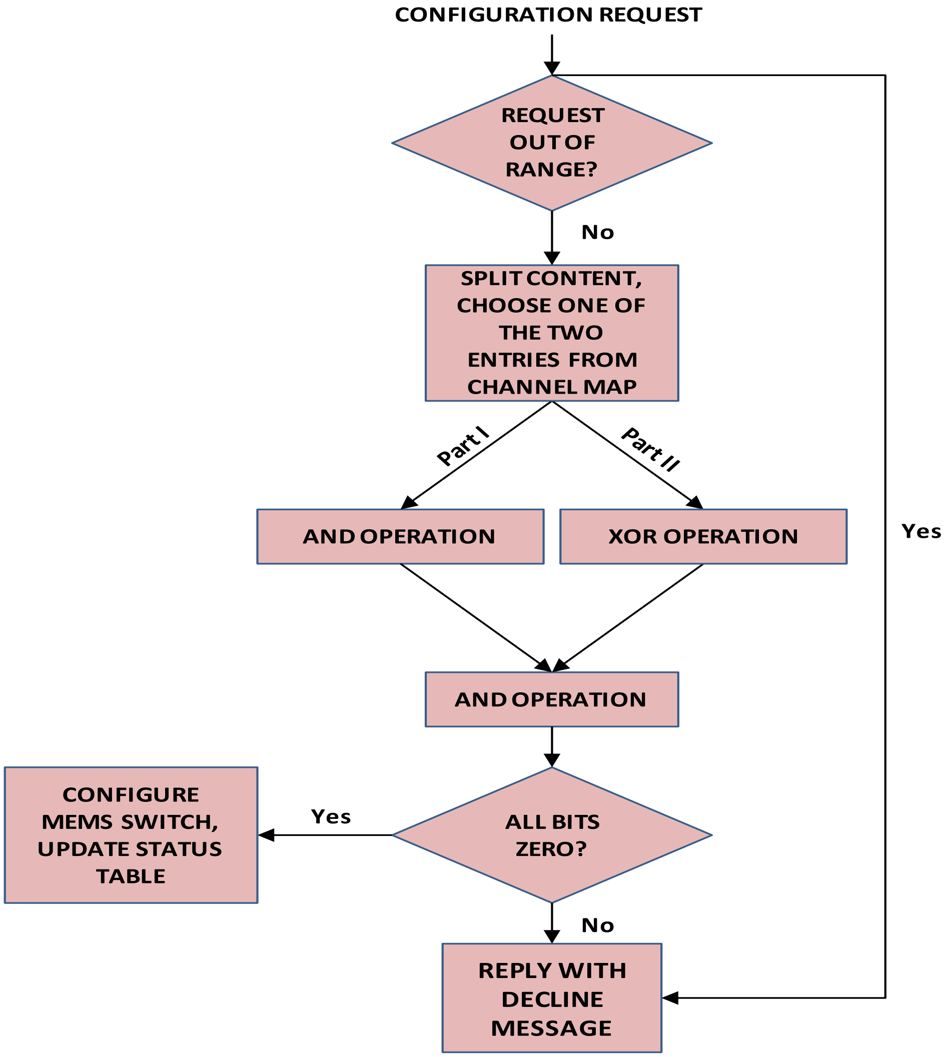

CONFIGURATION REQUEST
REQUEST OUT OF RANGE?
No
SPLIT CONTENT, CHOOSE ONE OF THE TWO ENTRIES FROM CHANNEL MAP
Part I
Part II
AND OPERATION
XOR OPERATION
Yes
AND OPERATION
CONFIGURE MEMS SWITCH, UPDATE STATUS TABLE
Yes
ALL BITS ZERO?
No
REPLY WITH DECLINE MESSAGE

Figure 39: Flowchart of Controller Configuration Algorithm

4.6 3-D Elevator Switch Edge Router Design

The design of the edge node is another important aspect of the 3-D switch. First, the design has to guarantee the ultra-high network speed as well as low latency. Secondly, the internal processing speed needs to be designed carefully to cope with the different line speed between network and client ports. Because the header of every packet is scanned to determine the outgoing port, it brings in a design challenge to eliminate the processing latency as much as possible.

In every line card, there are three types of ports. Client port aggregates traffic from the end user and a classifier screens the traffic, and then it sends the traffic to dedicated queue. DRP port sends commands to RRC while gathering the incoming traffic information. Network ports are different from the client port in the sense that the network port has much higher throughput than the client port. The follow section demonstrates one way of building a working version of 3-D switching, since the traffic will circulate around the edge line cards. This method is called 3-D elevator switching.

4.6.1 Client Cell Module Design

In general, from the customers' point of view, it is not necessary to know the corresponding transceiver the traffic should be sent to or whether the destination is reachable or not. Traffic from each ingress port usually has multiple destinations inherently.

Figure 40 is a block diagram on the edge cell. The classifier at the input side of the edge will verify the destination address and traffic with different destination addresses will be put into corresponding buffer. The associated destination address will be sent through the Central Process Module and the message will be fitted into the predefined format. DRP then conveys this information to RRC. At the same time, DRP receives a success or failure configuration result from the RRC. The message is then fed back to the classifier for packets drop or traffic reroute. The classifier then dynamically reconfigures the RRC if successful routing path is found by RRC.

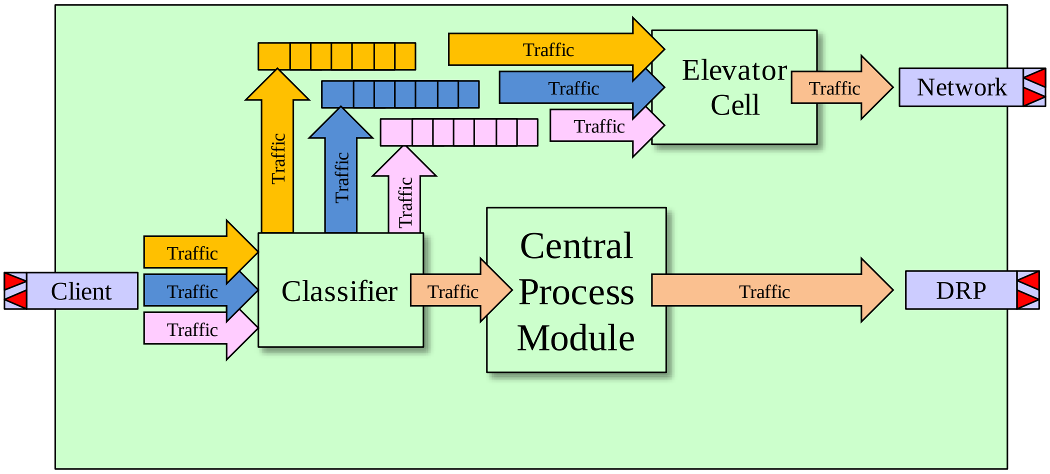


Figure 40: client cell block diagram

4.6.2 Network Cell Module Design

When sending traffic from network cell, the network would worry about the source and destination on packets in the queue but the network speed is the most concern in the network cell design. In order to achieve the goal of different traffic speed between the client port and the network port, the buffer within network and client should be set asynchronously and there should have two different clock domains for both client port and network port. The minimum speed the network cell should support is

$$\frac{num.\ client\ ports\ \times num.sources\ per\ client\ port\ \times source\ link\ speed}{number\ of\ network\ ports}. \quad (3)$$

4.6.3 Elevator Cell Module Design

The elevator cell in Figure 41 plays a critical role in the switch. The elevator cell is essentially a multiplexer that selects the outgoing path for traffic flows. Figure 41 illustrates one of the four elevator cell slices at edge node. Port A, B, C, D on the left-hand side is client and the L0 port at the right-hand side is the network port. There are 4 different destinations associated with each source port. Every destination has their own dedicated buffer to store corresponding packets. At each ingress transceiver port, elevator cells will capture traffic from the buffer that has the identical destination address to the elevator cell itself. The elevator cell then moves up to the next ingress transceiver buffer at the end of each operation. The elevator cell sends all traffic to corresponding network optical transceiver port and it moves back to the original position in the end. By doing this, traffic with the same destination but different sources can be aggregated at the client line card. This mechanism puts traffic into different vertical layers and the RRC

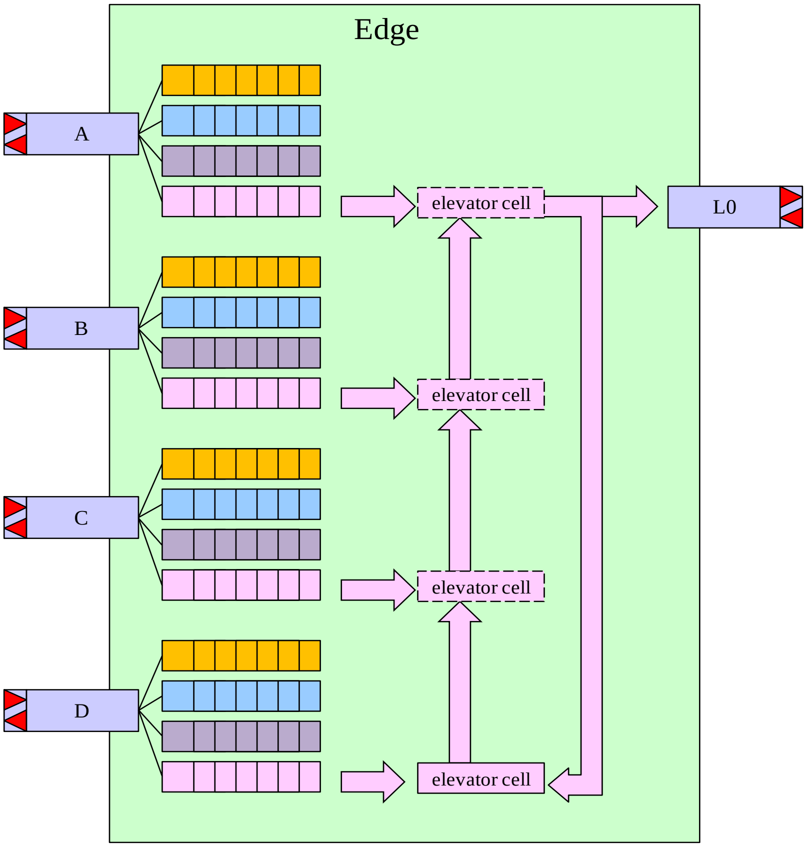


Figure 41: elevator cell block diagram

dynamically creates multiple paths at each layer. The high blocking probability issue in the traditional optical network can be eliminated to a large extent since traffic can be routed through other layers if one layer is not available at a particular time.

4.6.4 Formal Notation of 3-D Elevator Switch

Previously, the elevator design has been discussed. However, Figure 41 only illustrates a special case where there are 4 ingress port each has multiple traffic flow with four different destinations. On the network side only one network port has been shown in this case. In fact, the design of an agile network is another important goal. It is desirable that a network switch system can be parameterized towards different user applications in order to fit different scenarios.

Begin with the source side of the elevator module. Define *SC* as the number of edge source ports at the client side. Define the following symbols:

D: Number of traffic flows at every source.

J: Number of Network ports.

K: Number of Source Links.

M: Number of queues every network port attached

N_{ik}: *i-th* elevator at level *k* elevator modules

S_j: size of elevator service bit for level j_{th} elevator

As shown in Figure 41, each traffic flow at the source node should have one unique elevator module to collect the traffic for its particular destination. Hence

$$\sum_{i=0}^{n} N_i^1 = D. \tag{4}$$

Each elevator module only serves one traffic flow at a time. During the next time slot, the elevator elevates to the “higher floor” and it collects the traffic flow with identical destination address on that floor. There should have an indicator to indicate which floor is in service and which are not. It can be known from the algorithm that the size of indicator equals to the number of source links. Then,

$$S_1 = K. \tag{5}$$

On the network side, every network port should be configured to be attached to different traffic queues. Every network port has M queues attached. Assuming every network port fetches traffic from the same number of traffic queues, the total number of traffic queues (TQs) is

$$TQs = M * J. \tag{6}$$

Since one network port is attached to multiple traffic queues, another level of elevator module is required to process the traffic selection. Then

$$\sum_{i=0}^{m} N_i^2 = J. \tag{7}$$

Define the size of service indicator to be

$$S_2 = M. \tag{8}$$

Assuming every client port has identical number of queues. The number of queues equals to TQ, and

$$D = TQs = M * J. \tag{9}$$

Figure 41 only shows a portion of a full setup. Figure 42 is a complete configuration from Figure 41. There are four level 1 elevators each attached to one source node. Every source node has 4 buffers. On the network side, there is only one network port associated with this line card. This port has one level 2 elevator module taking care of the traffic scheduling from four buffers. Using the mathematical model to describe the scenario in Figure 42, it is an elevator switch system with parameter *K=4, n=4, D=4, m=1, J=1, M=4*. From Equation 2 to Equation 8, the size of the level 1 elevator switch indicator is 4. *TQ* can be obtained from equation 8 and the result is identical to *D*.

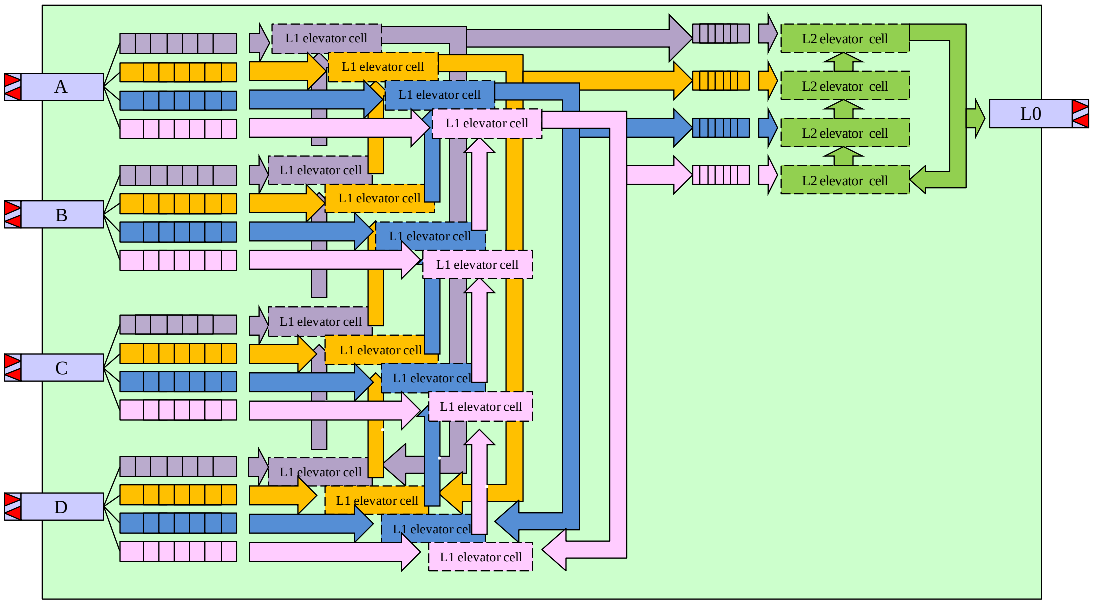


Figure 42: elevator cell block diagram

In real application, the number of network ports is more flexible. The number of queues fed to each network port is parameterized through the top level module. However, to make the configuration symmetric, every network port expects to have the same number of traffic queues.

Figure 43 is an extension from Figure 42. At the ingress side, there are four level one elevators each attached to one source node. The source node has four buffers associated with different destinations. Level one elevator module periodically scans the incoming buffers that have identical destination tag. When the corresponding queue at the source node is empty, the elevator module skips the traffic fetch operation at the current level and it elevates to the next “floor”, which is the corresponding traffic queue at the next source node. If there is traffic waiting for transmission, the elevator module directly forwards traffic to the corresponding outgoing traffic queue. Traffic management mechanisms can be integrated into traffic fetch operation in order to achieve QoS. For example, there could be a weight parameter to adjust the priority of each traffic queue: for higher priority traffic queue, the weight can be higher than the one that contains lower priority traffic.

The level two elevator module then forwards the traffic from the outgoing traffic queue to the network traffic port. One may find that for the level two elevator modules, their behaviors are slightly different from level one elevator module: rather than taking the traffic from the source traffic queue that has the same destination, each level two elevator module is in charges of one network port and it takes traffic that goes to the same

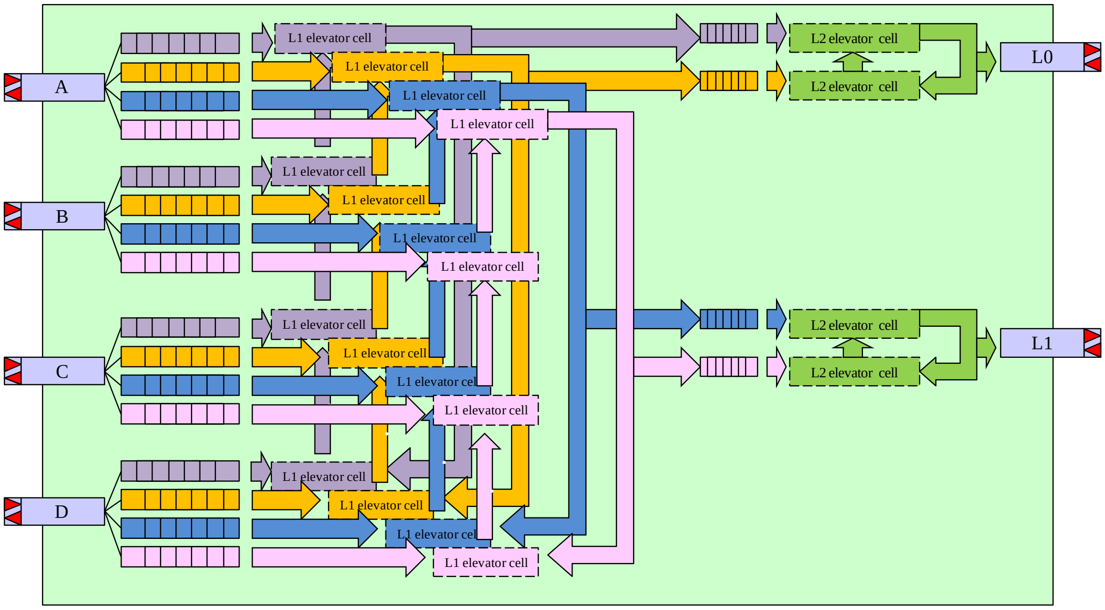


Figure 43: elevator module extended block diagram

network port. There is a central routing table and every elevator switch module keeps track of that table. When constructing the level two elevator modules, the Central Process Module checks the routing table and assigns outgoing traffic queues to different network ports. By doing this, the outgoing traffic can be correctly routed to its designated network port.

Figure 43 has four outgoing queues containing four different destinations. Through scanning the routing table, the Central Process Module assigns each queue to its corresponding network port based on the routing algorithm. The routing algorithm could be custom designed to fit the application.

4.7 Summary

The proposed 3-D optical switch paves a new path for optical switching network in the sense that the switching techniques could be expanded into three dimensions. By allowing traffic switching among different layers, the 3-D optical switch has greatly increased the flexibility of the network and it demonstrates its potential for future optical networks. This chapter also shows two practical ways of implementing the proposed 3-D optical switch: the static elevator switching and dynamic elevator switching. The elevator switch controller configuration algorithm shows the path decision could be achieved using combinatorial logic which is O(1) time complexity. Also the formal notation of the 3-D Elevator Switch shows the parameterized 3-D elevator module towards hardware implementation.

Chapter 5 Hardware Prototyping of the 3D Elevator Switch

The previous chapter focuses on the system architecture of the 3D elevator switch network design and the system level design of the central module. The elevator switch can adapt to any required link speed as long as the central processing module is sufficiently fast to process traffic at each level of the elevator module. The elevator switch edge link can be implemented using different technologies. The primary consideration in designing the elevator switch is the processing speed. In addition, the packet forwarding speed is essential for the network side since multiple traffic streams may go through the same network port, making the design more challenging. Despite the challenges, this chapter describes the hardware prototyping of the proposed 3-D switch, demonstrating the feasibility of the design.

5.1 10G Edge Single Channel Design

5.1.1 Xilinx Virtex 7 Board

Although lack of rich peripheral support, field-programmable gate array (FPGA) has a very high processing speed and rich RAM blocks to implement complex digital computations. The FPGA is an integrated circuit designed to be configured by a designer after its manufacturing. The FPGA configuration is generally specified with hardware description language (HDL). FPGAs are especially popular for prototyping the hardware circuit design. Once the circuit has been verified, ASIC chips are then produced using the designed circuit as the blueprint.

As to the elevator switch, since there are multiple client ports as well as network ports, the candidate FPGA development board should support a number of transceivers. Targeting to build the optical switch exceeds 10 Gbps per channel, the speed of the transceiver should be at least 10 Gbps per channel in the design. In addition, the memory-rich board is preferred since the routing table as well as the channel mapping table are memory intensive. A large amount of fast memory is also required for traffic storage and buffering when necessary.

Virtex 7 is the leading commercial FPGA chip from Xilinx. Comparing with other FPGAs, it features massive logic cells, high system speed as well as low power consumption. Virtex 7 has up to 2 M logic cells which is sufficient to hold the user design logic. On the communication side, it supports up to 2.8Tbs serial bandwidth, with up to 96*13.1GGTs. This outstanding throughput performance meets the design requirement of the 3-D switch well. The recent optical landscape continues to focus on the 10 Gbps and the cutting edge application is extending to 100Gbs or even 400Gbps option. For the 10 Gbps speed, the popular (small form factor plus) SFP+ optics are directly supported by the Virtex 7 GTH transceivers. The 100G CFP2 are simultaneously supported by the Virtex 7 series transceivers and this could be used in the future expansion design.

One of the design considerations is the power consumption. Since the 28nm Virtex 7 Series FPGA is fabricated on a low-power process, it offers lower total power consumption without sacrificing the performance of the FPGA. Comparing with competitor’s contemporary integrated chips, it only consumes a fraction of the power of other FPGA alternatives. Using the default power optimization setting, the Virtex 7 chip saves about 7.5% power with no performance degradation while other FPGAs save only

2% power with 1% performance degradation. Since the Elevator Switch requires multiple channels as the network and client ports, power saving without scarifying much performance is essential. A full power optimization is preferred in the design, where the Virtex 7 chip saves 18% power with 1% performance degradation rather than 12% power saving with 9% performance degradation other vendors claim.

Although there is a limit on the number of channels to be supported by a single FPGA chip, the Xilinx Virtex 7 board has excellent performance on both throughput and power consumption needed for the 3-D switch. Xilinx Virtex 7 stands out as a good candidate to implement the Elevator Switch and future design upgrade is available through utilizing the rest of the resource on the FPGA chip.

5.1.2 Xilinx Virtex-7 GTX Transceivers

The elevator switch requires both network ports and client ports to be accessible from the FPGA chip. Supporting multiple high-speed SFP+ ports in the FPGA is the fundamental requirement for prototyping the 3-D elevator switch. There are two pairs of electrical differential signals attached to each of the SPF+ interface and these two electrical differential signals are converted to optical signals by the SPF+ module. These two differential signals are directly driven by one of the GTH transceivers on Xilinx Virtex 7 FPGA.

The GTH transceiver takes care of the clock recovery, signal processing in the physical layer. As the speed goes higher, the coding/encoding scheme becomes especially important in the physical layer. Any data misalignment could impact the stability of the entire system or making it unable to function correctly. The GTH transceiver features easy configurability and highly integrated with the logic resource of the Virtex-7 FPGA.

Its goal is to provide a stable physical layer solution on the high-speed telecommunication design. At physical coding sub-layer, the transceiver supports 4-byte internal data path with 64b/66b encoding and decoding. The PMA sub-layer of the transceiver supports clock recovering as well as decision feedback equalization. The GTH transceivers are generic transceivers that could support standards such as PCI Express Reversion 1.0 to 3.0, Serial Advanced Technology Attachment(SATA), 10GBASE-R, 10Gb Attachment Unit Interface, 100Gb Attachment Unit Interface, 40Gb Attachment Unit Interface which will be used in the elevator system design.

Figure 44 is the block diagram of Xilinx GTH transceivers. There are 20 GTH

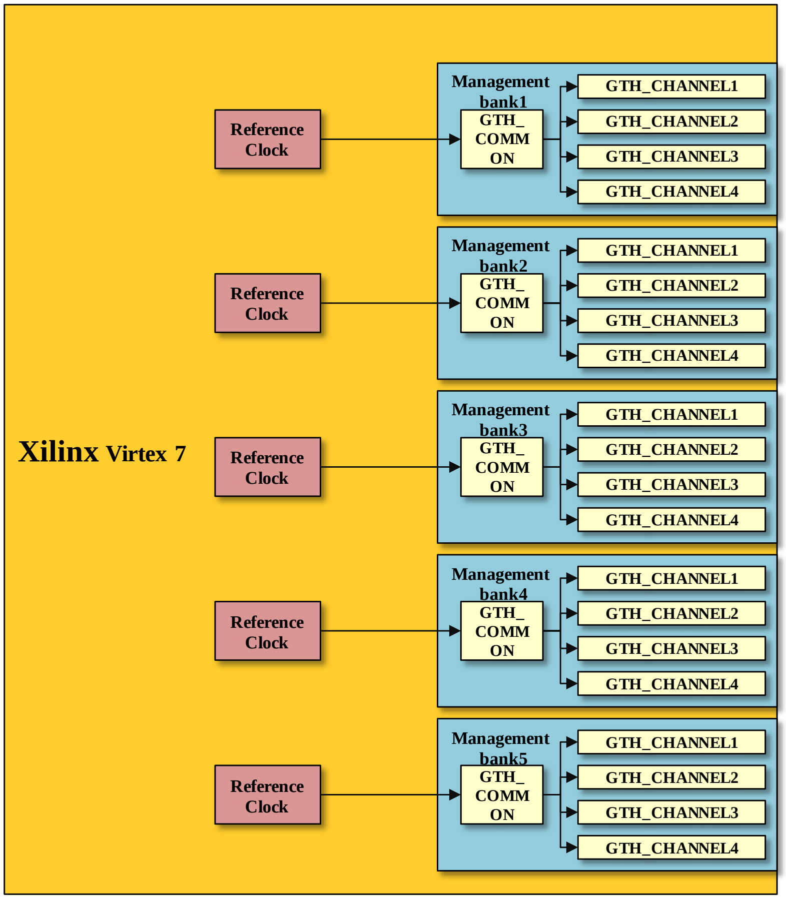


Figure 44: Xilinx Virtex 7 GTH transceiver Block Diagram

transceivers in Xilinx Virtex-7 chip. The transceivers are located at the edge of the chip and are grouped together for better performance. There are also five management banks with four transceivers in each bank. Each bank has its unique recovery clock and this clock is fed to four transceivers as their independent recovery clocks. The recover clock is used as the receiving clock and it is also used as the transmission clock.

5.1.3 Hightek Xilinx Virtex-7 Evaluation Board

In addition to the high-speed transceiver considerations described above, memory capacity is of paramount importance in terms of prototyping the 3-D switch to absorb any traffic rate variation. The Virtex 7 FPGA chosen for the design only has limited on-chip memory, which is used for critical information such as configuration commands and channel maps. With 10 Gbps channel rate, one has to resort to off-chip memory for

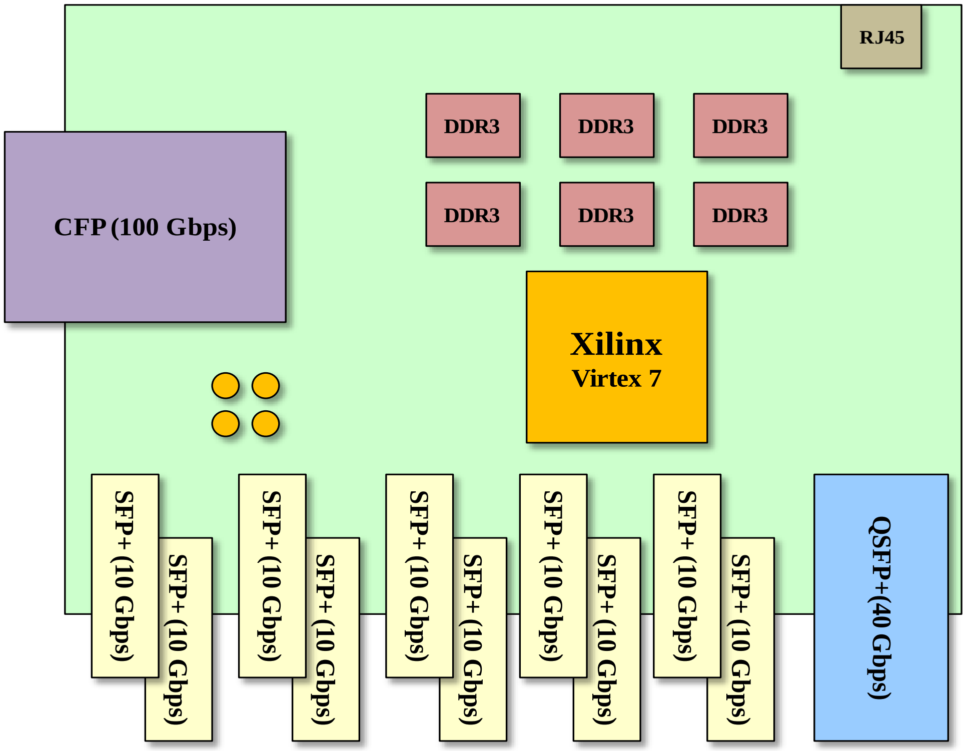


Figure 45: HG707 Evaluation board Block Diagram

packet buffering. There are two design considerations in choosing the off-chip memory: memory capacity and memory bandwidth. The memory capacity should be large enough to store the maximum number of packets specified by the 3-D switch. The memory bandwidth should be high enough to support sustained link rate.

The FPGA developed board chosen for the design should be rich in external memory and features Xilinx Virtex 7 FPGA chip. The HG707 FPGA board from High-tech Global is a good fit for the elevator switch design. Figure 45 is the block diagram of the HG707 board. The board is powered by the Xilinx Virtex-7 X690T FPGA that could deliver the most configuration blocks for building 10G/40G/100G elevator switching systems. 100G optical ports (10G*10 Gbps) can be used to design both the client ports and network ports of the elevator switch. The QSFP+ 40Gbps and CFP 100Gbps port on the board are reserved as the future upgrades towards the network traffic port. There are 3GB (512MB/chip*6chips) DDR3 on-board external memory which meets the task of traffic storage discussed above. Besides, there is a 100/1000Mb Ethernet controller with RJ45 interface, which is sufficient for sending/receiving control commands for the elevator switch.

5.1.4 Hightek Systems Transceiver Core

The Hightek Transceiver core is an integrated PCS intellectual property (IP) core designed for 10 Gbps application. The IP core contains two main modules: the 10G data link layer module (10GEMAC), physical layer PCS module (10G PCS) as well as the packet generator module for verification purpose. There are multiple registers in each module as the configuration register. Figure 46 shows the frame work of the IP core. The 10G EMAC transmits data to 10G PCS using 64-bit XGMII Interface under the clock speed of 125MHz. The GTH Transceiver module contains its own circuit to translate the traffic from 64B/66B to 40B. By doing this, the 10G PCS could interface with GTH transceivers using 40-bit Serdes interface. User can also choose to use the Xilinx's core for this task. User can choose to feed traffic to 10G EMAC from the build-in Packet Generator or the Custom Design. Adopting the 64-bit AXI4-Streaming User Interface, the traffic speed can reach 10 Gbps under 125MHz clock rate.

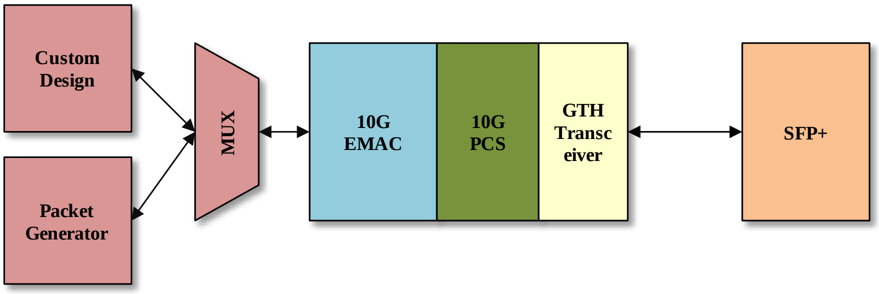


Figure 46: HG707 GTX Transceiver Core

Reference clocks for each bank are generated through the synthesizer. Figure 47 illustrates the clock distribution of the 4-port Elevator Switch System. The synthesizer in

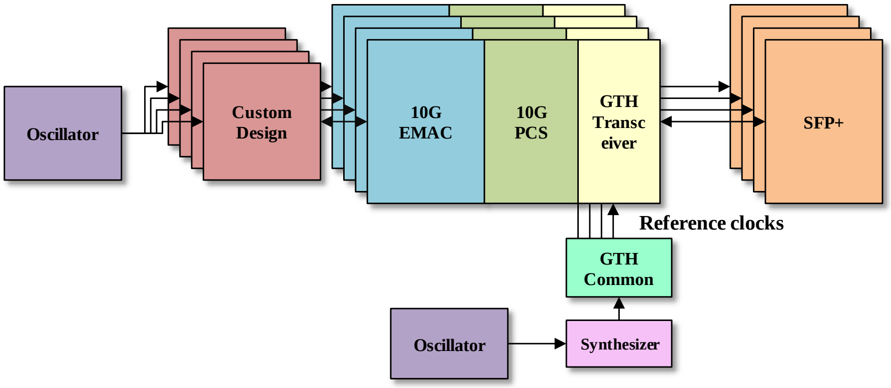


Figure 47: overview of top-level 3-D elevator switching system

the figure could produce reference clock from the base frequency provided by the oscillator if configured through the I^{2}C interface. It must be provided in advance in order to successfully reset the circuit. Therefore, an I^{2}C module that has AXI4-Lite Memory-Mapped Slave Interface is also included in the design. The system clock is provided by another individual oscillator.

5.1.5 Elevator Switch Peripheral System Design

Each module has one AXI4-Lite Memory-Mapped Slave Interface so that the master device could access their registers specifically. Figure 48 shows the AXI4-Lite address map. The AXI4-Lite address consists of sixteen bits and these bits are further divided into two parts: the chip select address and the chip address. Since there are multiple modules, the first four-bit address is separated from the rest of the bits as the chip select to differentiate each module. The rest twelve bits are the base address of each module.

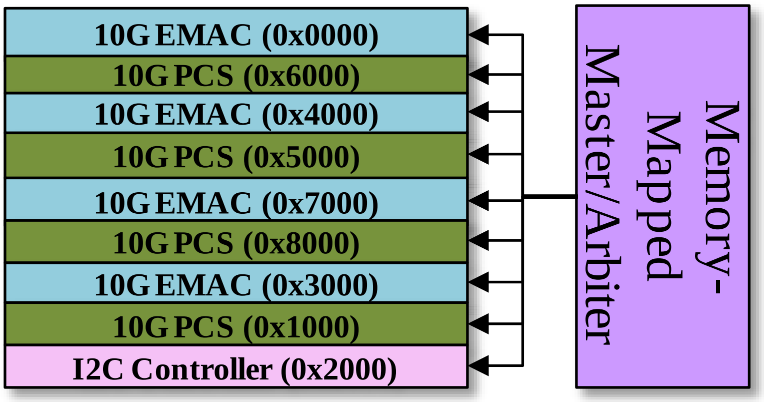


Figure 48: AXI4-Lite Master/Arbiter address space

When powering up the circuit, the I^{2}C controller configures the synthesizer to generate the reference clock for all transceivers. The information is stored in a predefine read-only memory and it is automatically executed after powering up. System goes into the reset state upon the detection of correct reference clock. The reset includes PCS reset, Transceiver reset as well as EMAC reset. If any of the components encounter an error during transmission, the entire system will reset to ensure correctness.

Two experiments were conducted to verify the correctness of the design. The first experiment is to verify the behavior of the single channel operation. A single mode fiber is plugged into both Tx and Rx sides of the SFP+ module to create a loopback condition for the test traffic. 7Gbps traffic is launched using the internal Packet Generator attached at the 10G EMAC side. On the receiving side, there is a separate monitoring module to count the number of receiving frames and calculate real-time throughput. A UART-based software is connected to the module and it repeatedly reads the register to calculate the traffic bandwidth. Figure 49 shows the single channel operation. The red line and blue line are representation of the traffic transmission speed and receiving speed, the time step is 1 second with 10 seconds per column. During the experiment, the transceiver side optical fiber was plugged in and out periodically to test the reliability of the system. It can be seen from the figure that the receiving module keeps up with the transmission module quite well: Once the transmission is resumed, the receiving side recovers quickly

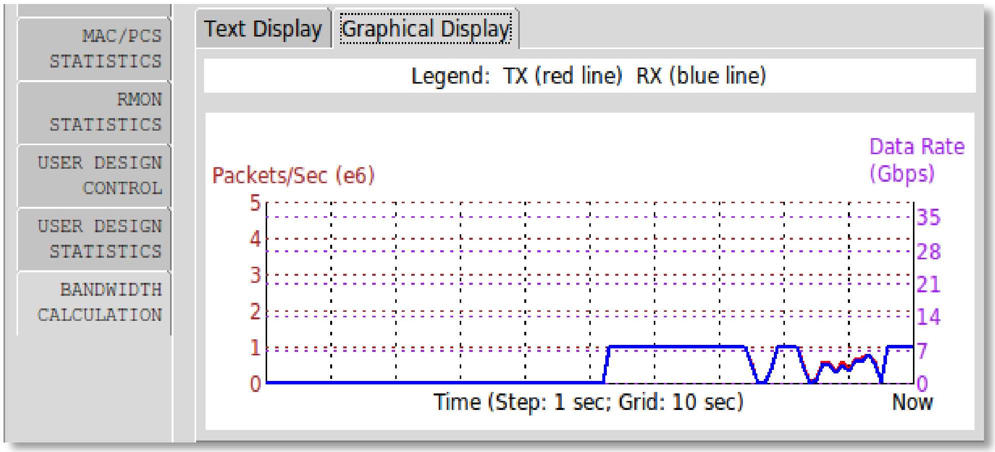
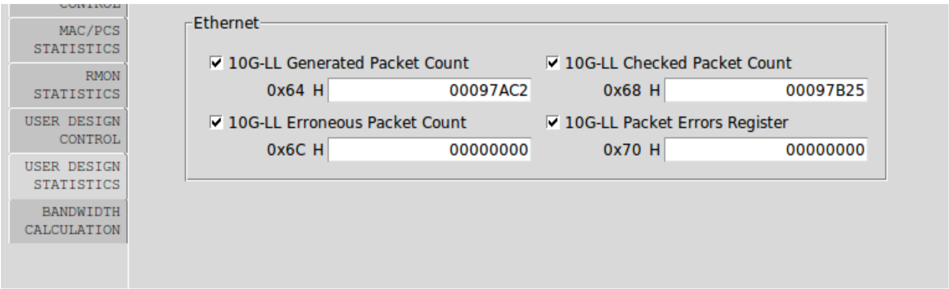


Figure 49: single channel operation

to minimize the packet lost during the recovery period. The above experiment shows the stability of the Tx and Rx module using traffic generated by the Packet Generator with a series of identical packets.

In addition to the above mentioned loopback test, a single channel transmission test has been conducted to verify the stability of the receiver module. The SFP+ capable PC directly sends random traffic to the FPGA board with a traffic rate around 500Mbps, the Packet Monitor then records the correct packets received in a separate register for read out. In case the packet fails the cyclic redundancy check, the error recording register will increment by one. Figure 50 and Figure 51 show both the traffic throughput and the content of the register. It can be seen from the figures that the PC traffic speed reaches



Ethernet

☑ 10G-LL Generated Packet Count
0x64 H 00097AC2

☑ 10G-LL Checked Packet Count
0x68 H 00097B25

☑ 10G-LL Erroneous Packet Count
0x6C H 00000000

☑ 10G-LL Packet Errors Register
0x70 H 00000000

Figure 51: error register status

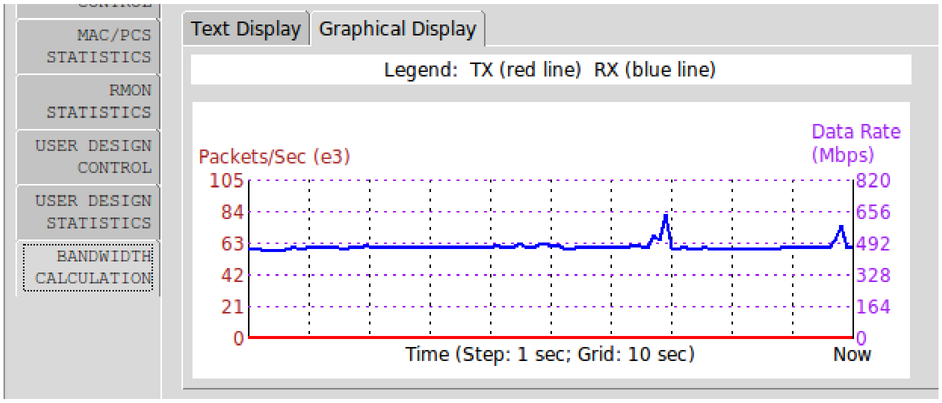


Figure 50: PC traffic throughput

around 500Mbps with no error. This experiment demonstrated the stability of the single channel communication module.

The FPGA development board comes with ten SPF+ channels. As described above, four transceiver channels forms one management bank and ten channels in total occupies three management banks. These management banks use three reference clocks that have

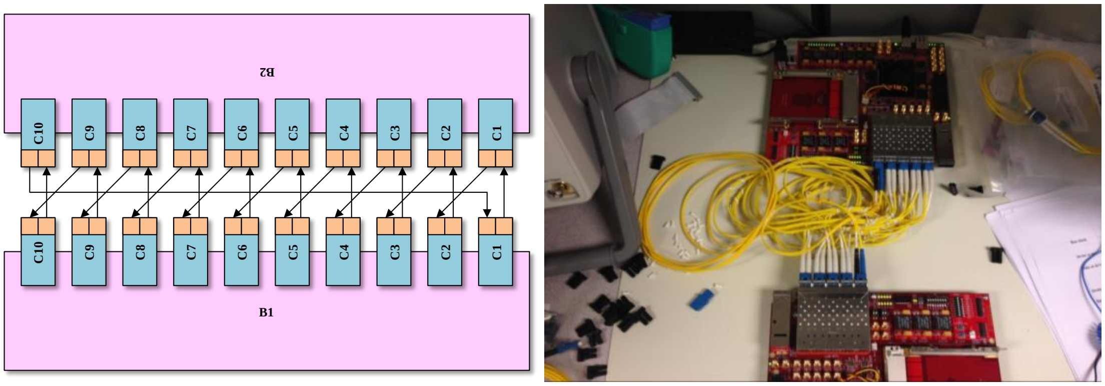


Figure 52: daisy chain experiment configuration

the same clock frequency from the synthesizer. A daisy-chain experiment is conducted to test the performance of multiple ports. In this experiment, two boards with 10 SFP+ channels per board are daisy chained together and this makes the throughput of the single board achieves 100Gbps. Figure 52 shows the configuration of the experiment: The source traffic is generated from Channel 1 on Board 1. Whenever the Channel 1 of Board 2 receives the traffic from Channel one of Board 1, it directly forwards the traffic to its Tx port. Traffic the goes back to Port 2 of Board 1 from Port 1 of Board 2, and the pattern continues until traffic reaches Port 10. At the end of the iteration, the traffic goes back from Port 10 of Board 2 back to Port 1 of Board 1. A traffic monitoring module in this port collects the data and then calculates the traffic rate.

The setup has been verified using the internal packet generator. During the experiment, traffic is launched from Channel 1 (PCS address 0x6000) and daisy chained through Channel 2 (PCS address 0x5000), Channel 3 (PCS address 0x8000) and Channel 4 (PCS address 0x1000). The read out value on traffic receiving monitor at Channel 4 is shown in Figure 53. A steady 7 Gbps incoming traffic is detected without error. This experiment has proved the capability and correctness of aggregated 100Gbps traffic transmission.

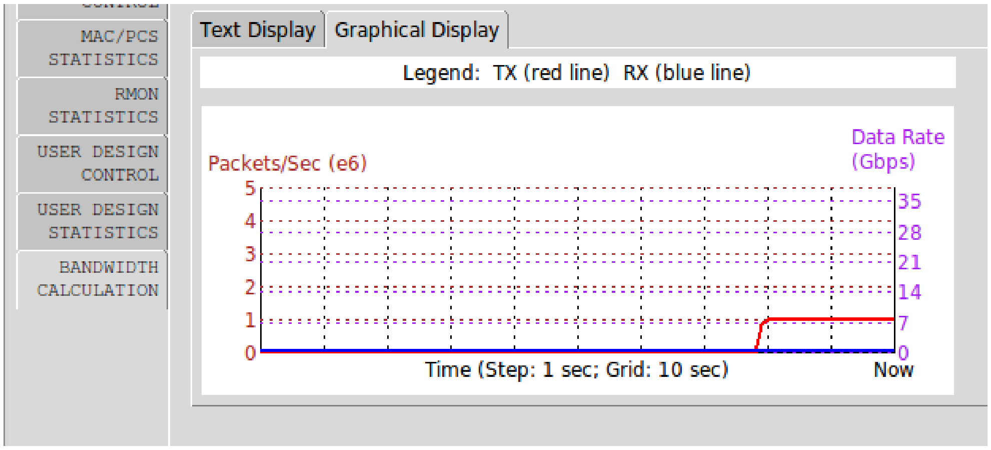


Figure 53: channel 4 traffic monitor status

5.2 Parameterized Elevator Module

After validating the transceiver module, the next goal is to design the Elevator Switch module. The EMAC module for each channel can be configured to accept frames contain unicast addresses or broadcast addresses. As to ports of the elevator switch, each port should accept different destination addresses and the central processing module then forwards the packets to different elevator switch modules. Because of this requirement, the promiscuous mode of EMAC has been disabled and a customized address filter is created for the purpose of collecting packets of different destinations. The flowchart of the traffic filter is shown in Figure 54: If an incoming packet is detected, the packet header will be scanned. If the destination information in the packet header matches the entry in the address filter, the packet will be forwarded to its corresponding buffer;

otherwise, the packet will be dropped. By doing this, traffic with different destinations are separated and are put into different traffic queues.

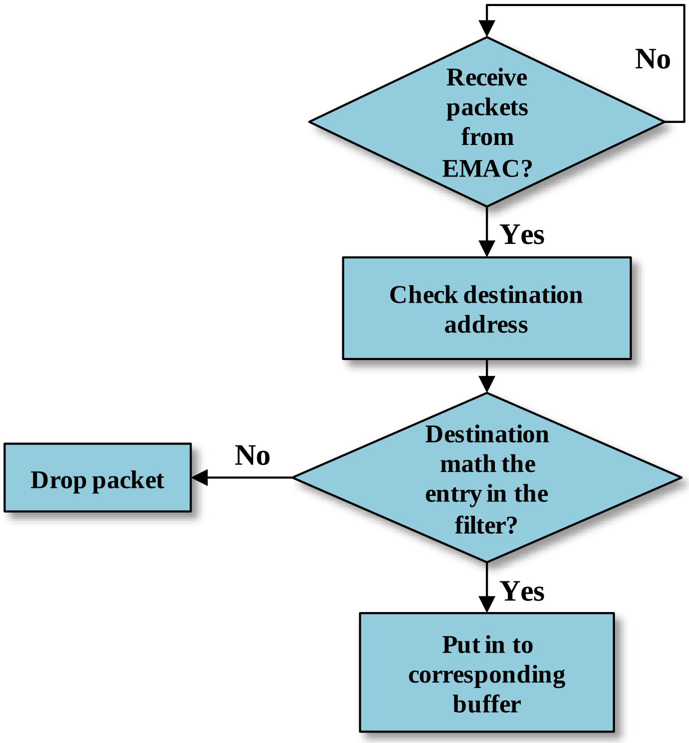
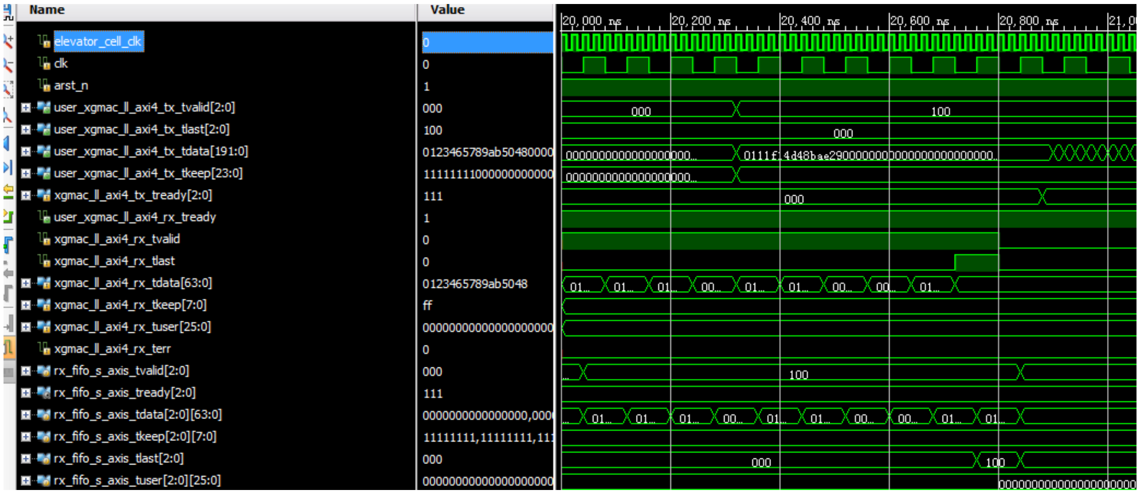

Receive packets from EMAC?
No
Yes
Check destination address
Destination math the entry in the filter?
No
Drop packet
Yes
Put in to corresponding buffer

Figure 54: traffic filter flow chart

The timing issue is a critical design consideration in the design to avoid impact on the performance of the system. At the same time, synchronization is a big challenge in



Name	Value
elevator_cell_clk	0
clk	0
arst_n	1
user_xgmac_ll_axi4_tx_tvalid[2:0]	000
user_xgmac_ll_axi4_tx_tlast[2:0]	100
user_xgmac_ll_axi4_tx_tdata[191:0]	0123465789ab50480000
user_xgmac_ll_axi4_tx_tkeep[23:0]	111111110000000000000
xgmac_ll_axi4_tx_tready[2:0]	111
user_xgmac_ll_axi4_rx_tready	1
xgmac_ll_axi4_rx_tvalid	0
xgmac_ll_axi4_rx_tlast	0
xgmac_ll_axi4_rx_tdata[63:0]	0123465789ab5048
xgmac_ll_axi4_rx_tkeep[7:0]	ff
xgmac_ll_axi4_rx_tuser[25:0]	00000000000000000000
xgmac_ll_axi4_rx_terr	0
rx_fifo_s_axis_tvalid[2:0]	000
rx_fifo_s_axis_tready[2:0]	111
rx_fifo_s_axis_tdata[2:0][63:0]	0000000000000000,000
rx_fifo_s_axis_tkeep[2:0][7:0]	11111111,11111111,111
rx_fifo_s_axis_tlast[2:0]	000
rx_fifo_s_axis_tuser[2:0][25:0]	00000000000000000000

Figure 55: circuit simulation of asynchronous traffic buffer

the sense that traffic queues from different client ports are actually working under their individual clocks. The synchronization issue has to be overcome in order for the elevator module to access traffic queues that have the same destination but from different ports. Therefore, in the elevator design, an asynchronous buffer scheme is adopted and the elevator module works on its own separated clock. By doing this, the elevator cell could control its own speed to balance the traffic between client ports and network ports. Figure 55 shows different clock domains in the Elevator system. Different background colors represent different clock domains. Client ports are under the same clock domain. The Processing Module determines the service object and the asynchronous buffers take the traffic from its own EMAC module. Figure 36 is a circuit simulation of the asynchronous FIFO. The Tx side data could be read out using faster clock while the Rx side use a slower clock to buffer data.

Figure 56 shows the internal design of the elevator switch. The Level One elevator switch is a Processing Module controlled by the multiplexer that selects different traffic streams from the client side. The user defined QoS mechanism could be implemented in the Processing Module so that customer traffic queues can be treated based on the QoS scheme when selecting the traffic stream. The Processing Module could put more weight on high priority traffic streams such as video conference, telesurgery command. On the other hand, the background data transfer could be deferred without much penalty. Before assigning traffic queues to elevator switches, the Processing Module checks the routing map stored in the on-chip memory, determines the QoS schemes and finally selects corresponding input traffic queues for every elevator switch. The main differences between Level One elevator switch and Level Two elevator switch is that the former serves traffic streams that have the same destination address but the Level Two elevator switch serves the traffic streams that would transmit to the same network adaptor.

5.2.1 Integrate the Parameterized Elevator Module

Based on the discussion in the previous chapter, there are several parameters to determine the structure of the elevator switch module. Assuming that every client port has identical traffic queue configuration (the number of traffic queues at each client port are the same and the destination addresses of each client port are the same). The number of Level One elevator switch is then determined by the number of destinations at the client port. The Level Two elevator switch fetches traffic that would transmit through the same network adaptor. Since each Level Two elevator switch corresponds to one network port, the number of network ports determines the number of Level Two elevator switches. Based on these observations, an elevator module has been designed using Verilog HDL. By specifying those parameters, the module can be either configured as the client elevator module or the network elevator module.

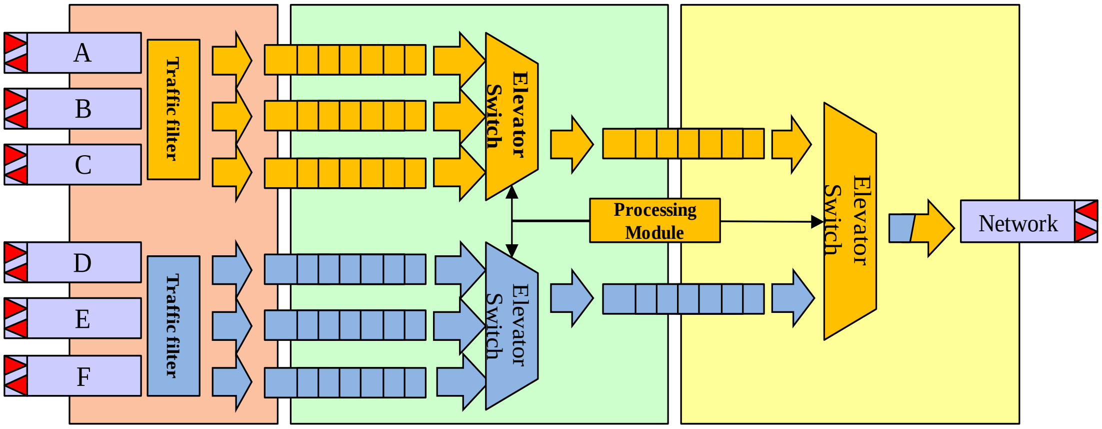


Figure 56: elevator switching internal system design

5.3 Hardware Experiment

5.3.1 Configuration 1 (3-in-1-out)

Based on the guidelines, a 3-in-1-out hardware experiment is conducted to test the feasibility of the elevator switch. Figure 57 is the test configuration of the first test setup. In this test, there is only one traffic queue at each client port to store the client packets. Hence, the test takes three traffic queues to a same destination from the client ports A, B and C. At the network side of Board 1, there is one network port to transmit the outgoing traffic. Since all traffic goes to the same destination, one network port takes the traffic from Board One and sends the traffic to the destination port - Port D. Board Two takes the mixed traffic sent from client ports on Board One. Since the traffic only has one destination, there is only one traffic queue in the network port of Board Two. The Processing Module then controls the elevator switch and it takes the traffic from the Network traffic queue, forwarding the traffic to the destination port.

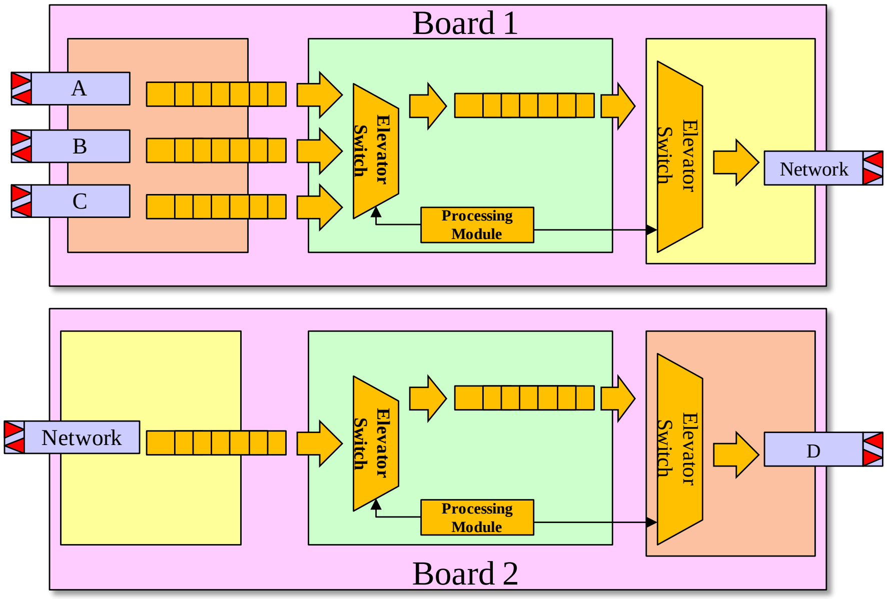


Figure 57: 3-in-1-out elevator switching system hardware configuration

A 1080 pixel high definition video clip with traffic rate of 4.5Mbs is chosen as the experimental traffic at each client port. The source divides the video streams into chunks and the user datagram protocol (UDP) is used to encapsulate the video stream into network traffic. On the client side, SFP+ 850nm SR optical form factors are used as the optical ports. The number of Level One elevator switch is one since there is only one traffic queue that has the same destination per source link. On the network side, the number of Level Two elevator switch equals to the destination count which is also one. Figure 58 is the screen capture of real-time HD traffic at port D from port A (video on the left hand side), B (video on the middle) and C (video on the right hand side).

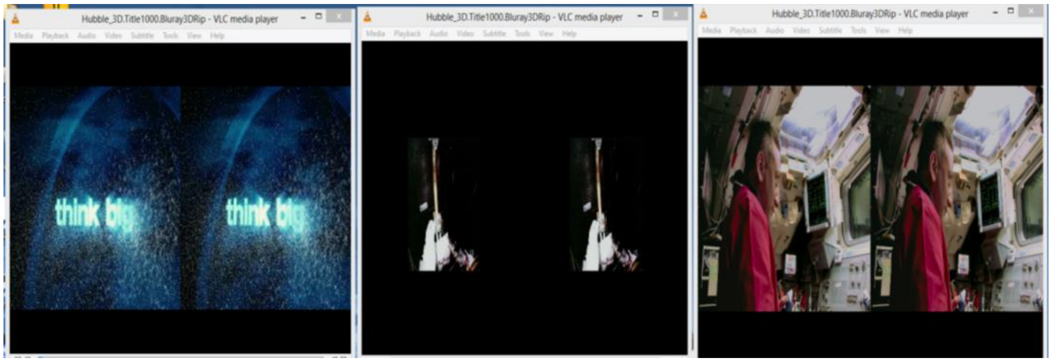

Hubble_3D.Title1000.Bluray3DRip - VLC media player
Media Playback Audio Video Subtitle Tools View Help
think big
think big
Hubble_3D.Title1000.Bluray3DRip - VLC media player
Media Playback Audio Video Subtitle Tools View Help
Hubble_3D.Title1000.Bluray3DRip - VLC media player
Media Playback Audio Video Subtitle Tools View Help

Figure 58: real-time traffic at port D

5.3.2 Configuration 2 (3-in-3-out)

Consider the case where there are multiple traffic streams each of which tries to reach different destinations. A 3-in-3-out test configuration experiment is conducted in test configuration 2 to verify the this scenario. In the second experiment, there are three traffic queues at each client port and every traffic queue stores the traffic going to different destinations. However, the client only streams one of the traffic to each port. For example, in Figure 59 the traffic stream labeled in purple is sent from port B. But the purple traffic stream are not sent from port A and port C. Port A instead sends traffic stream labeled in yellow while port C sends traffic stream labeled in blue. Other unused queues are marked white and they remain empty during the experiment. In the hardware implementation, the client elevator modules are instantiated from one design, and unused traffic queues are kept idle during the entire experiment.

Since there are three traffic queues at each client port, three Level One elevator cells are needed in order to process three different traffic streams. The Processing Module selects which queue to serve and also controls the high-speed switch multiplexer to the corresponding queue based on its internal service protocol. On the network side of the Board 1, only one network port is designed to transmit three different traffic types. The

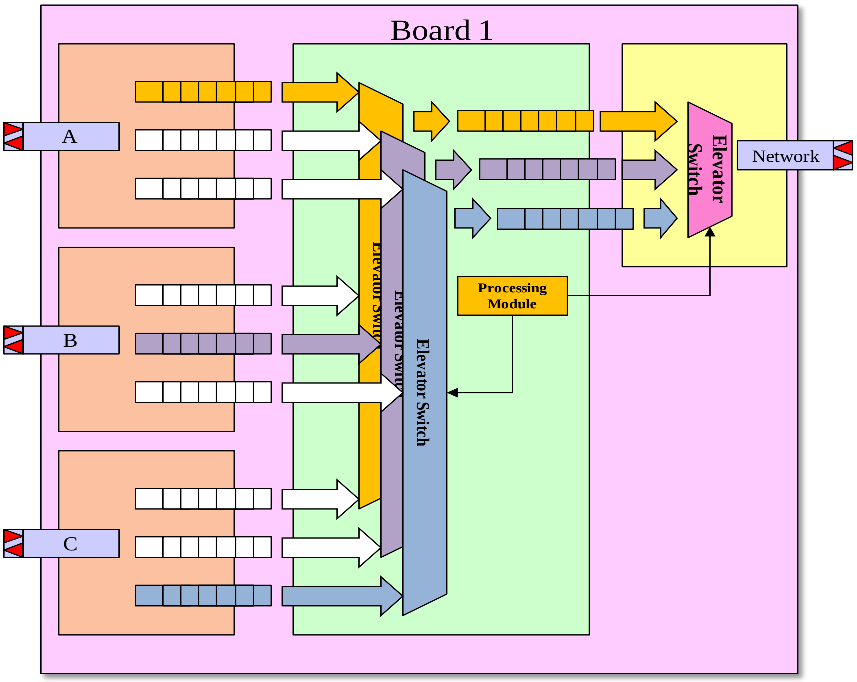


Figure 59: 3-in-3-out elevator switching system hardware configuration on board 1

destination count equals to one and *DPDL* equals to three. According to the formulas in the previous chapter, only one Level Two elevator cell is needed in the design and the number of service bits is three. To simplify the switching part in the hardware experiment, the hardware experiment uses round robin algorithms to periodically select different traffic types from traffic queues.

On the network side of Board Two as shown on Figure 60, three traffic queues are created to store traffic for port D, port E and port F, respectively. The Level One elevator switch differentiates traffic according to the destination and it forwards the traffic to different client ports. There are three Level Two elevator switches on Board Two and each of them collects one traffic stream from the Level One elevator. The traffic stream is

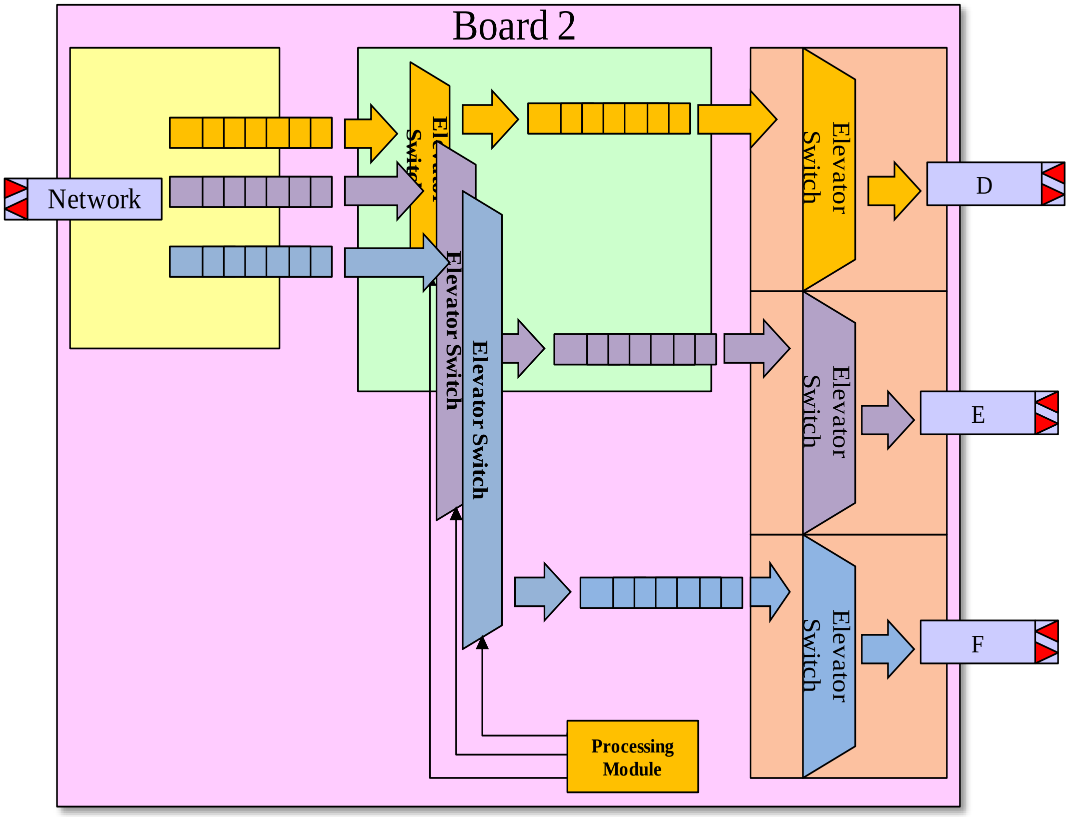


Figure 60: 3-in-3-out elevator switching system hardware configuration on board 2

then decoded and repackaged into video streams at the output port of port D, port E and port F.

The experiment also uses 1080 pixel high definition video clip with traffic rate of 4.5Mbs. The average traffic rate at the network port is 13.5 Mbs since three traffic streams are aggregated at the network port. The IP addresses of six client ports are within the same subnet in order to make the communication goes through the personal computer (PC). Three steady 1080 video clips are generated from port A, port B and port C. A mixed traffic streams are observed at the network port. By running the client software at the network port of Board 1, videos from port A, B and C are captured correctly.

5.3.3 Configuration 3 (9-in-9-out)

A full 9-in-9-out experiment is conducted to demonstrate a fully working 3-D Elevator Switch. In this experiment, nine traffic streams containing three destinations are generated at the client port. Every client port has three traffic streams that go to different destination and client A, B and C are instantiated from the same client design. In the 9-in-9-out configuration, all traffic queues at port A B and C are used as shown in Figure 61. The *DPSL* equals to three since three traffic streams are generated per source port in the experiment. The major difference on Board 1 between 3-in-3-out and 9-in-9-out is that the Level One elevator service bits are 3 instead of 1 due to the difference in the source count.

In the previous 3-in-3-out case, only one packet queue is used per source link. Hence, the service bit is set to one to make the Level one elevator switch to collect traffic only from one traffic queue. Figure 59 demonstrates this where traffic queues filled in white color stays empty in the 3-in-3-out experiment. Each elevator switch only collect traffic from one traffic queue at all times. However, the third configuration fully populates the scheme and 9 traffic streams are put into 9 traffic queues as shown in Figure 61. Since the Level One elevator has to decide which queue out of three queues needs to be serviced, the service bits of level one elevator switch is then set to 3.

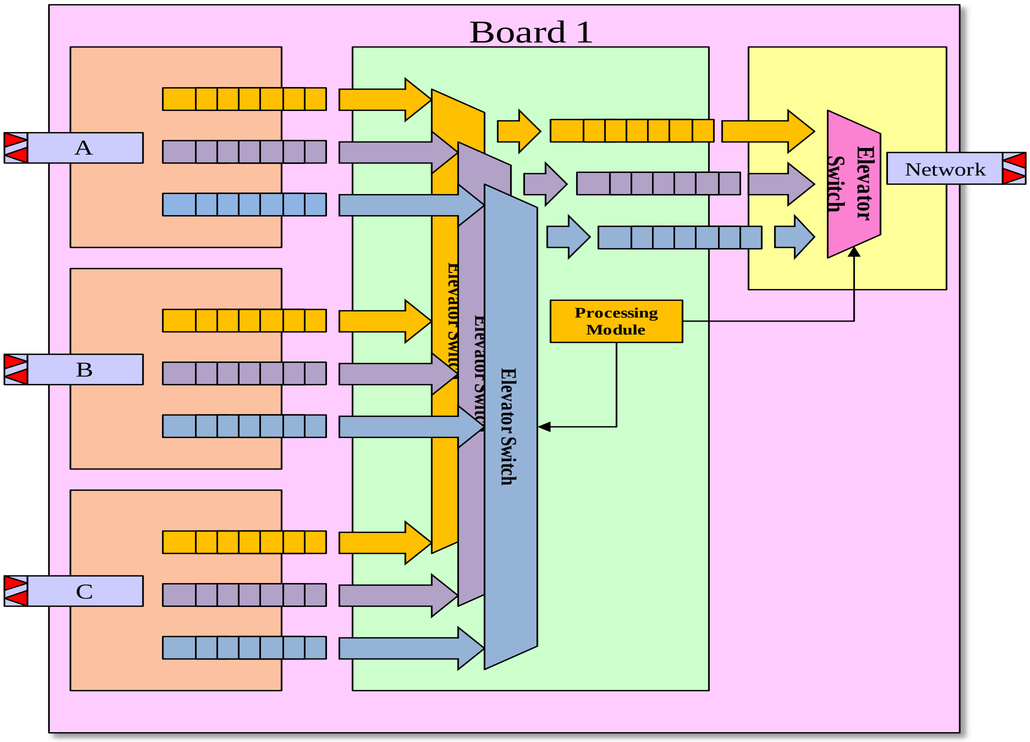


Figure 61: 9-in-9-out elevator switching system hardware configuration on board 1

Board 2 of the third configuration is similar to Configuration 2. However, Because Port A, B and port C are streaming three different traffic streams to port D, E and port F, every client port at Board 2 displays three videos rather than only one video as in 3-in-3-out case. The configuration on Board Two is shown in Figure 60. Nine original traffic streams are combined into three traffic streams according to the traffic destinations. The Processing Module controls the Level One elevator switch on Board Two and it forwards the traffic to their designated destination. The PC connecting to each port then resorts the traffic streams according to the source address and reassembles the network traffic into video streams. Three different traffic streams are successfully observed from each PC and there are a total of nine video clips play simultaneously in the experiment.

5.4 Summary

These experiments have demonstrated the feasibility of transforming architecture of the proposed 3-D switch into practical hardware design. Although some of the components need to be carefully designed in order to achieve traffic throughput requirements, the experiments achieve the idea of 3-D switching using the elevator-like method to transport traffic between different layers. The entire elevator switching design is based on the architecture described in Chapter 4. Router designers can choose to customize the parameterized elevator module in the design. The QoS algorithm can also be configured within the module to test the efficiency of each method. Besides, the system is reconfigurable for other network configurations to be deployed using the same design for future studies on traffic latency, throughput, and the different QoS schemes, which would greatly assist the development towards future optical networks.

Chapter 6 Conclusion

The optical network under the DWDM technology is believed to be the future of the networks due to the huge bandwidth DWDM offers. Again, the service provider is seeking solutions in optical networks when expanding their service areas.

Despites the recent advances in technology, there are still a lot needs to be investigated in optical switching technologies in terms of high speed information exchange, traffic throughput as well as network flexibility and scalability. OBS serves as the intermediate agent to address the disadvantages of OCS and EPS. Nevertheless, OCS schemes are still preferred in some time-sensitive applications. EPS, on the other hand, is a mature technology. However, it is facing challenges in terms of network scalability.

The dissertation has examined these three technologies with an extra focus on OBS due to its importance in the DWMD Multi-Mode router. The Multi-Mode Optical Switching technology uses OBS as a middle ground and it builds a unified platform for EPS, OCS and OBS. This idea meets various requirements of applications and the ability for seamless resource sharing has greatly improved cost efficiency in building DWDM routers.

The importance of application stands out especially in telesurgery and emergency scenarios. The ground breaking idea of A^2 optical switching network exams the optical network from the applications' perspective and it aims at building an optical network that could sense the priority on application level, solving the QoS problem in existing optical switching network. A formal protocol of the A^2 router has been proposed. From the perspective of the applications, the Application-Aware (A^2) optical network featureing the reverse data path reservation is a good candidate of asymmetric traffic transmission. By creating alternative switching technique towards optical switching network, the A^2 optical scheduler eliminates the setup latency problem in traditional optical router. At the same time, the path reservation can be changed in real-time, increasing the probability of packets delivery. The proposed A^2 optical switching has been verified through both software simulation and hardware implementation.

Furthermore, this dissertation has introduced the concept of 3-D switching, creating a new dimension for network resource sharing. The proposed 3-D switching can greatly improve the performance in terms of traffic connectivity as well as the number of required basic elements to build a large switching matrix. In addition, a practical elevator-switching using the idea of 3-D switching has been proposed. Both the dynamic and static approaches have been discussed. A hardware prototype has been constructed, and the hardware experiments have verified the feasibility of 3-D switching. It is expected to serve as a building block of future optical networks.

Bibliography

[1] E.W.M. Wong and T. S. Yum. “Maximum Free Circuit Routing in Circuit Switched Networks.” *Proc. IEEE Infocom’90, vol. 3*, pp. 934-937, Jun. 1990.

[2] M. Wang, S. Li, E.W.M. Wong and M. Zukerman. “Blocking Probability Analysis of Circuit-Switched Networks with Long-Lived and Short-Lived Connections.” *IEEE/OSA Journal of Optical Communications and Networking, ISSN 1943-0620*, Volume 5, Issue 6, pp. 621-640, 2013.

[3] A. V, S. Li, M. Wang, E.W.M. Wong and M. Zukerman. “Computation of Blocking Probability for Large Circuit Switched Networks.” *IEEE Communications Letters, ISSN 1089-7798*, Volume 11, Issue 11, pp. 1892-1895, 2012.

[4] T. Venkatraman and S. Suresh. “Blocking of Multirate Circuits in Multichannel Optical Networks.” *Proceedings of SPIE, ISSN0277-786X*, Volume 4874, Issue 1, 07/2002.

[5] S. Oh and M. Kang. “A Burst Assembly Algorithm in Optical Burst Switching Networks.” *Proceedings of the OFC*, pp. 771-773, 2002.

[6] C. Yuan, Z. Zhang, Z. Li, Y. He and A. Xu. “A Unified Study of Burst Assembly in Optical Burst Switching Networks.” *Photonic Network Communications*, Volume 21(3), pp. 228–237, 2011.

[7] J. Choi, H. Vu, C.W. Cameron, M. Zukerman and M. Kang. “The Effect of Burst Assembly on Performance of Optical Burst Switched Networks.” *Information Networking*, 2004, Vol.3090, pp.729-739.

[8] V.M. Vokkarane, J.P. Jue and S. Sitaraman. “Burst Segmentation: An Approach for Reducing Packet Loss in Optical Burst Switched Networks.” *Proceedings of the IEEE International Conference on Communications (ICC)*, 28 April–2 May 2002, vol. 5, pp. 2673–2677.

[9] M.S. Alam, S. Alsharif and P. Panati. “Performance Evaluation of Throughput in Optical Burst Switching.” *International Journal of Communication Systems*, 26 SEP 2010, Volume 24, Issue 3.

[10] L. Tancevski, A. Ge, G. Castanon, and L. Tamil. “A New Scheduling Algorithm for Asynchronous, Variable Length IP Traffic Incorporating Void Filling.” *Proc. OFC’99*.

[11] Y. Xiong, M. Vandenhoute and H. Cankaya. “Design and Analysis of Optical Burst-Switched Networks.” *Proc. SPIE’99 Conf.*, All Optical Networking: Architecture, Control, Management Issues, Boston, MA, Sept. 19-22.1999, vol. 3843, pp. 112–119.

[12] Z. Rosberg, A. Zalesky, H.L. Vu and M. Zukerman. “Analysis of OBS Networks with Limited Wavelength Conversion.” *IEEE/ACM Transactions on Networking*, 2006, ISSN 1063-6692, Volume 14, Issue 5, pp. 1118 – 1127.

[13] S. Li, M. Wang, E.W.M. Wong, V. Abramov and M. Zukerman. “Bounds of the Overflow Priority Classification for Blocking Probability Approximation in OBS Networks.” *IEEE/OSA Journal of Optical Communications and Networking*, ISSN 1943-0620, 2013, Volume 5, Issue 4, pp. 378 – 393.

[14] Z. Rosberg, L.V. Hai, M. Zukerman and J. White. “Blocking Probabilities of Optical Burst Switching Networks based on Reduced Load Fixed Point

Approximations.” *Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies, ISSN 0743-166X, 2003, ISBN 0780377524*, Volume 3, pp. 2008-2018.

[15] H.C. Leligou, K. Kanonakis, T. Orphanoudakis and J.D. Angelopoulos. “Traffic Aggregation for Slotted OBS Systems.” *ELMAR*, 2005. 47th International Symposium, June 2005, pp. 319-322, 8-10.

[16] Y. Chen and W. Tang, “Concurrent DWDM Multi-Mode Switching: Architecture and Research Directions.” *IEEE Communications Magazine*, May 2010, vol. 48, no. 5, pp. 57-65.

[17] J. S. Turner. “Terabit burst switching.” *J. High Speed Networks*, 1999, vol. 8, no. 1, pp 3-16.

[18] C. Qiao and M. Yoo. “Optical burst switching (OBS): A new Paradigm for an Optical Internet.” *J. High Speed Networks*, Jan. 1999, vol. 8, no. 1, pp. 69-84.

[19] Y. Chen and P. Verma. “Optical Burst Switching – An Emerging Core Network Technology.” *Internet Networks – Wired, Wireless, and Optical Technologies*, K. Iniewski, Ed. CRC Press, Taylor & Francis Group, pp. 265-290, 2010.

[20] G. Wu, T. Zhang, J. Chen, X. Li, and C. Qiao. “An Index-Based Parallel Scheduler for Optical Burst Switching Networks.” *IEEE/OSA Journal of Lightwave Technology*, vol. 29, no. 18, Sept. 2011.

[21] Y. Chen, J. Turner, and P. Mo. “Optimal Burst Scheduling in Optical Burst Switched Networks.” *IEEE/OSA Journal of Lightwave Technology*, vol. 25, no. 8, pp. 1883-1894, Aug. 2007.

[22] G. Wu, T. Zhang, J. Chen, X. Li, and C. Qiao. “An Index-Based Parallel Scheduler for Optical Burst Switching Networks.” *IEEE/OSA Journal of Lightwave Technology*, vol. 29, no. 18, Sept. 2011.

[23] C. Qiao, W. Wei, and X. Liu, “Extending generalized multiprotocol label switching (GMPLS) for polymorphous, agile, and transparent optical networks (PATON),” *IEEE Communications Magazine*, pp. 104-114, Dec. 2006.

[24] X. Liu, C. Qiao, W. Wei and T. Wang. “A Universal Signaling, Switching and Reservation Framework for Future Optical Networks.” *IEEE/OSA Journal of Lightwave Technology*, vol. 27, no. 12, pp. 1806-1815, June 2009.

[25] R. Ramaswami and G. Sasaki. “Multiwavelength Optical Networks with Limited Wavelength Conversion,” *IEEE/ACM Transactions on Networking*, vol.6, no.6, pp.744-754, Dec 1998.

[26] R. Ramaswami. “Optical Networking Technologies, What Worked and What didn’t.” *IEEE Communication Magazine*, vol. 44, no. 9, pp. 132-139, Sept, 2006.

[27] Y. Chen, J. Turner, and P. Mo. “Optimal Burst Scheduling in Optical Burst Switched Networks,” *IEEE/OSA Journal of Lightwave Technology*, vol. 25, no. 8, pp. 1883-1894, Aug. 2007.