RECONFIGURABLE SCHEDULER FOR STACKABLE OPTICAL FABRIC

A Thesis

Presented to

the Faculty of the Department of Electrical and Computer Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in Electrical Engineering

by

Shalini Srivastava

December 2014

RECONFIGURABLE SCHEDULER FOR STACKABLE OPTICAL FABRIC

Shalini Srivastava

Approved:

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

Committee Members:

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

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

Dr. Cumaraswamy Vipulanandan,
Professor,
Civil and Environmental Engineering

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

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

Acknowledgements

I owe my gratitude to all those who helped me in achieving my academic goals and made this thesis possible.

I would like to thank my advisor Dr. Yuhua Chen, for her great effort in my thesis and my master’s career. Dr. Chen helped me a lot with her inspiring attitude, deep knowledge of the subject, and professional attitude to research work.

I am thankful to the members of my thesis committee, Dr. E. J. Charlson, Dr. Pauline Markenscoff and Dr. Cumaraswamy Vipulanandan for their help and support.

I would like to thank my family members for their continuous support specially my husband for his great support and guidance during my master’s degree program. I could not fulfill my academic goals without his unwavering support and motivational attitude.

I gratefully acknowledge support for this work by the National Science Foundation (NSF) under Grants 0923481 and 0926006.

RECONFIGURABLE SCHEDULER FOR STACKABLE OPTICAL FABRIC

An Abstract

Presented to

the Faculty of the Department of Electrical and Computer Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in Electrical Engineering

by

Shalini Srivastava

December 2014

Abstract

Because of the high bandwidth demand, the optical fabric becomes very important in Internet infrastructure. The switching technologies also pose unique challenges for optical fabric. The optical infrastructure has to face all these challenges created by advanced Internet technologies as well as the challenge of scalability of optical fabric. This thesis proposes a novel stackable structure of optical fabric which is easy to expand and has low installation cost.

In this work, the concept of reconfigurable scheduler for stackable optical fabric has been introduced, and the architecture of this scheduler has been designed. The scheduler considers intrinsic connectivity of physical layers as well as the service requirements of the application layer. In order to verify the design, the prototype was designed and realized on a FPGA board. For evaluating the cost saving of the proposed stackable optical fabric, a general cost model was developed and compared with the flat optical fabric. The stackable fabric was observed to be the trade-off between cost and performance.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

With the advancement of dense wavelength division multiplexing (DWDM) technology in optical fiber communication, the bandwidth available in fiber links has increased tremendously. Meanwhile, the fast growing Internet traffic demands a very high transmission capacity of optical links. It becomes essential to harness the enormous amount of bandwidth of optical fiber efficiently. This needs structural changes in core and edge routers of the optical network. The DWDM technology can support several hundred wavelength channels to be transmitted over a single optical fiber and each channel can transmit the data at10Gb/second. This leads to a data rate of 10Tb/sec in each fiber [1] [2].

Although the electronic switching technologies have the finest granularity, but the DWDM data rate is beyond its capacity. In electronic packet switching (EPS) every packet has to be converted from the optical signal to the electronic signal before processing and then converted back to the optical signal before transmitting. It leads to extra E/O and O/E converters for every channel which becomes the overhead in transmission. These E/O and O/E converters can simply be avoided in optical switching technologies. Existing optical switching technologies such as optical packet switching (OPS) [3], optical circuit switching (OCS) [4], and optical burst switching (OBS) [5] [6] [7] have specific characteristics because of which they are suited for the dynamic conditions of the Internet traffic.

Because of the increasing demand of bandwidth, the switching fabric becomes very important. Due to extensive use of the Internet, there are many switching technologies which support Internet services. All these switching technologies pose unique challenges for optical fabric. Also, various types of Internet applications create complexity for optical switching technologies and optical fabric. The optical infrastructure has to face all these challenges created by advanced Internet technologies. The scalability of optical fabric also plays an important role in selecting the optical components for optical fabric.

The thesis proposes a stackable arrangement of optical fabric. The proposed arrangement is easy to expand to fulfill the high bandwidth demand of the Internet compared to the traditional arrangement of a flat optical fabric, where expansion causes replacement of small flat fabric with a big flat fabric. Stacking reduces the expansion cost by reusing the old optical fabric. The stackable optical infrastructure is an area where there is a lot to explore.

The stacking of optical fabric is a very economical option to grow the optical structure, but it creates some challenges to the switching technologies. The traditional scheduler, which works well for the flat optical fabric has some more specifications to comply with, when it works for the stackable optical fabric. The thesis elaborates these new challenges posed by the stackable optical fabric, and lists the specifications required for a reconfigurable scheduler for stackable optical fabric. The scheduler will have to consider the intrinsic connectivity of the physical layers while scheduling any input request. The specifications also include the option of providing soft service to the users. The soft service would allow the users to have the option of selecting specific output ports, based on the type of application, while transmitting the optical signal through the optical fabric. So the scheduler considers the intrinsic connectivity of the physical layers as well as the service requirements of the application layers. The concept of soft service is very useful if the Internet administrator wants to designate the output ports based on the type of the application the input signal is carrying.

The thesis includes the design of a prototype of the reconfigurable scheduler for stackable optical fabric. Based on the specifications discussed in the Chapter, an algorithm is devised. The prototype has a hardware design in Verilog HDL. The prototype design is also realized on Field Programmable Gate Array (FPGA) development board and tested with a sequence of requests. The response of the prototype exactly matched with the expected results.

The thesis also includes the definition of the cost model of a stackable optical fabric. The cost model of stackable fabric is then compared with the cost model of an equivalent flat optical fabric. It is observed that the stackable optical arrangement reduces the cost of network with some loss of flexibility in connectivity. However, this potential less flexible structure can in fact be useful for virtual network applications. One can use the freedom of isolation by running one’s application on some stacks with some degree of connectivity. This option provides dual advantages of physical isolation as well as low cost of optical fabric and components.

The rest of the thesis is organized as follows. The Chapter 2 provides the background on optical switching techniques and the basic structure of a scheduler. Chapter 3 introduces the concept of stackable optical fabric, describes its various arrangements and mathematical representation of each arrangement. Chapter 4 elaborates the specifications of the reconfigurable scheduler for stackable optical fabric, develops the algorithm for the scheduling stackable fabric. Chapter 5 develops a cost model for the stackable optical fabric. It also describes the prototype verification of the proposed scheduler and its realization on the FPGA board. Chapter 6 concludes the thesis.

Chapter 2 Background

Rapidly growing Internet services are driving the demands for higher transmission capacity and high-speed routers at an unprecedented rate. The advances in dense wavelength division multiplexing (DWDM) technology have made it possible to exploit the huge potential bandwidth of optical fibers. The past decade has seen the great efforts in building hardware-based high-speed electronic routers. However, there is still a serious mismatch between transmission capacity of WDM fibers and the switching capacity of electronic routers. There is still a need to study and work on the architecture of routers, more specifically switching and scheduling which can provide a fast, flexible and economical solution. The Dense wavelength division multiplexing technology multiplexes a number of optical signals into a single optical fiber using different wavelengths. Switching can be done electronically as well as optically. In electronic switching, the optical data is processed electronically, which requires an expensive Optical/Electrical (O/E) and Electrical/Optical (E/O) conversion at core routers. However, this technique is not scalable enough to support hundreds of wavelengths. Every optical data unit is sent through O/E conversion, then it is processed electronically in electronic domain and then converted back into optical data using E/O converters. The disadvantages are expensive E/O and O/E converters. The other disadvantages include the power loss, heat dissipation in those units and limited switching capacities of these electronic switches.

2.1 Optical Switching Techniques

Optical switching technique includes optical packet switching (OPS) [3], optical circuit switching (OCS), and optical burst switching (OBS).

Optical Packet Switching: Each data includes the data packet called the payload and its header, which is attached to the payload. The header includes the information about the payload such as its destination. In OPS, only header is processed electronically. When the optical packet is received at the optical switching node, the header is converted to electronic signal using O/E converters. During this time the payload is buffered optically. When the header is processed and the optical link is established, the payload travels through the optical link. However, the header has to be converted back to the optical data using E/O converters and attached to its payload. This raises the issue of synchronization. Hence, in OPS, optical buffering and synchronization are two major issues. Buffering in the optical domain is performed using fiber delay lines (FDL), which can hold the data only for a limited time [8]. The requirement of optical buffering in OPS is variable buffer time, because this is the time needed for the header processing, as well as contention resolution, which is very difficult to realize using FDL [3] [8] [9] [10] [11] [12].

Another important issue in optical packet switching is synchronization. When the optical link is established and the payload is ready to travel, the header has to be converted back to the optical data and attached to the payload. This is a complicated task to perform in the optical domain. Due to these two issues, the optical packet switching is difficult to deploy.

Optical Circuit Switching: In Optical Circuit Switching, the optical circuit is established between the source and the destination before the optical data is transmitted. Once the light path is setup, the data follows the path without any O/E or E/O conversion. Hence, in this technique, the expensive O/E and E/O conversions are avoided. Also, the issues of buffering and synchronization as in optical packet switching, are not present in this technique. However, the setting up and releasing the light path takes hundreds of milliseconds which is a big amount of time. Optical circuit switching is ideal for data flow, which lasts for a long time, while it is insufficient in bandwidth utilization for short duration traffic.

Optical Burst Switching: In optical burst switching, the data is transmitted in the form of optical bursts [1] [5] [6] [13] [14] [15] [16] [17]. The packets of the same source-destinations are grouped into the bursts at the edge nodes. A bursts header cell (BHC) contains the information about the burst such as its destination, the duration of the burst and its arrival time. The BHC is processed electronically at the core router and is sent on a dedicated wavelength ahead of the actual burst. When arriving at the edge router, the BHC is converted into the electronic data using O/E converters and processed. The light path is established for the data bursts before they arrive. Once the light path is setup, the data burst travels without any further processing or delay. Hence, in this switching technique, FDLs are not required. Since a lot of optical packets are grouped together to form bursts and there is only one BHC for an optical burst, which has to be processed electronically, the O/E and E/O conversions are greatly reduced. Also, the BHCs are sent on a dedicated wavelength other than the actual data burst, unlike optical packet switching. Optical burst switching is an effective means to handle dynamic traffic. This is also used in the proposed scheduler in this thesis. Therefore, this technology is described in detail in the next section.

2.2 Optical Burst Switching

2.2.1 Introduction

In optical burst switching, the BHC is sent ahead of the data burst and on a different wavelength, which greatly reduces the complexity of controlling the data and the header packets. The buffering and synchronization issues are limited in this type of switching technology. The BHC which travels ahead of the data burst and reaches the router first, reserves the wavelength for the data bursts. There are some schemes used to accomplish this task, namely, explicit setup, estimated setup, explicit release and estimated release [18]. Explicit setup configures the light path immediately after it receives the burst header cell while, the estimated setup scheme configures the light path based on the information present in the BHC, such as the offset time to calculate the actual arrival time of the data burst. Similarly, the explicit release scheme sends another BHC to initiate the end of burst transmission, while the estimated release scheme automatically plans the release based on the information present in the burst header cell.

There are two major OBS signaling schemes, just-in-time (JIT) and just-enough-time (JET) [6] [16]. Just-in-time uses explicit setup and explicit or estimate release schemes, while just-enough-time uses estimated setup and estimated release schemes. The explicit schemes simplify the implementation process while the estimated schemes give the best bandwidth utilization. The choice of schemes depends on the priority between the ease of implementation and the maximum bandwidth utilization.

2.2.2 Burst Assembly

The process of aggregating the optical packets into the optical bursts at the edge of OBS network is called burst assembly [2]. The packets with the same destination are aggregated into one burst. The most common burst assembly techniques are the timer based and the maximum burst length threshold based [19] [20]. In the timer based assembly technique the optical burst is aggregated and sent to the optical network at periodic time intervals. In this scheme the bursts arriving at the core router edge can be of variable length.

In the maximum burst length threshold based scheme, there is a limit on the burst length. When the burst length reaches its threshold limit, the burst is formed and sent to the optical network. In this scheme, the fixed sized bursts are aggregated, but they reach the optical network in a non-period manner. The major issue in this scheme is to decide the appropriate burst length for the specific network parameters in order to minimize the packet loss probability in OBS network at the time of contention. In both the schemes, the burst header cell is also generated and is sent on a different wavelength prior to the transmission of the data burst.

The mixed scheme is also incorporated when the optical traffic is light or heavy, in order to improve the performance [15] [19]. The incoming packets may belong to a specific class. High priority bursts can preempt the wavelength reservation of low priority ones. Also, in some schemes, packets with high priority can be placed at the rear edge of the burst. This is done because sometimes during burst contention at intermediate core routers, the head part of the burst is dropped, hence, the high priority packets placed at the rear end of the burst have better chances to be transmitted successfully.

2.2.3 Burst Scheduling

Once the BHC reaches the core router, the scheduler in the core router allocates the wavelength resources for the optical burst, based on the information present in BHC [5] [13] [15] [21] [22] [23] [24]. The two most common schemes are latest available unused channel (LAUC) [5] [13] [15] [25] and latest available unused channel with void filling (LAUC-VF) [15] [26].

LAUC: In the latest available unused channel scheme, when a new BHC arrives, the scheduler checks for its data burst arrival time. It then checks among all the outgoing channels, whether any channel is available at that time. If more than one channel is available at that time, the scheduler selects the latest available channel.

LAUC-VF: In the latest available unused channel with void filling scheme, bursts are allocated in the voids or gaps of any existing consecutive wavelength reservations in order to provide better resource sharing results. If more than one void is available, the scheduler chooses the one that yields the least void. In case no void fits the burst, the scheduler tries to find a free resource across all the channels. The major drawback in this scheme is the time taken in scheduling the bursts which arise from the tracking and decision making for each scheduling operation. Due to the above reason, this scheme is difficult for deployment in large DWDM systems.

Optimal burst scheduling was proposed to avoid the out of sequence arrival of bursts [1]. A constant time burst sequencing (CTBR) scheduler pre-processes the BHCs, holds them in a timing wheel and schedules the BHCs in the sequence of burst arrival times instead of the BHC arrival times. This approach avoids the creation of voids and better utilizes the bandwidth of DWDM channels.

2.3 Optical Components

2.3.1 Multiplexer and De-multiplexer

The primary function of the router is to control and direct the Internet traffic. In DWDM network, the wavelength multiplexing is used to combine and propagate the signals. Wavelength multiplexing is performed using a pair of multiplexer and de-multiplexer. In a bidirectional system, a single device acts as a multiplexer and a de-multiplexer and is present at both the ends [27]. Figure 1 shows a simple structure of a multiplexer and de-multiplexer.

Figure 1: Multiplexer and de-multiplexer

There are several techniques used for multiplexing and de-multiplexing. The refraction and diffraction properties of light are used in combining and separating out the different wavelength signals.

Prism Refraction: The simplest arrangement could be using a prism. Figure 2 explains the working of a prism reflection. When a polychromatic light beam falls on a prism, different wavelengths refract at different angles. Each of these wavelengths, then propagates through a lens. The lens focuses each of these wavelengths to the different points where it enters separate fibers.

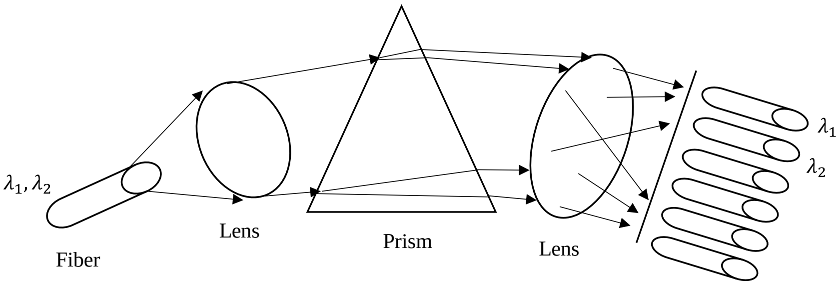


Figure 2: Prism refraction

Waveguide Grating Diffraction: Another arrangement could be using diffraction grating. When a polychromatic light wave falls on a diffraction grating, each wavelength diffracts at a different angle. Each of these wavelength passes through a lens and further focuses on different fibers. Figure 3 shows a wavelength grating diffraction phenomenon.

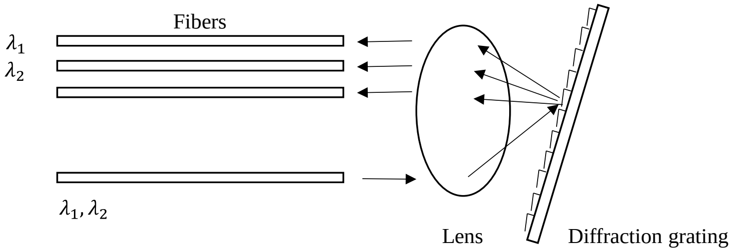


Figure 3: Wavelength grating diffraction

Arrayed waveguide grating: Figure 4 shows arrayed waveguide grating. This type of multiplexing technique is also based on the principle of diffraction. It consists of an array of curved channel waveguides. The channels have the difference in the path lengths. Two input and output cavities are connected at two ends. When a light beam enters at the input cavity, it enters the curved channel waveguide after diffraction. Due to the difference in the optical lengths of different channels, there is a phase delay between the output signals in the output cavity.

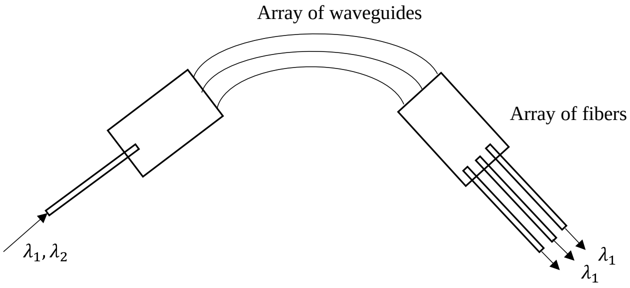


Figure 4: Arrayed wavelength grating

Add/Drop Multiplexer: Optical add-drop multiplexer (OADM) plays a key role in providing reconfiguration flexibility and intelligent handling of network signals. In scheduling certain light paths, the wavelengths need to be dropped and added dynamically. These are done by the optical add-drop multiplexers [28]. Several architectures of the

OADM have been proposed. The very basic function of OADM can be understood using the following diagram as mentioned in Figure 5.

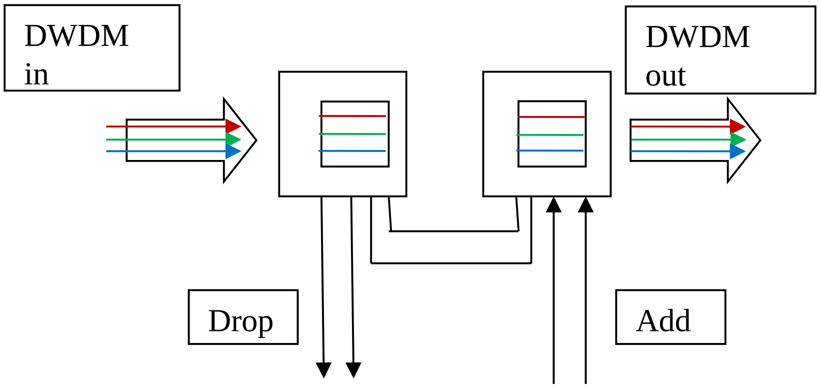


Figure 5: Add-drop multiplexer

2.4 Optical Switching Fabrics

There are various switching fabrics available today and many of them are under study. None of them could be the best for all the optical applications. There are certain parameters taken into account to evaluate and choose optical fabrics for any application [29].

2.4.1 Performance Indicators

The following are the key performance selection criteria of optical switches.

Switching time: This is the time taken by the switch to change its state. It is associated with the reconfiguration rate of the device. Different applications have different switching time requirements.

Crosstalk: This is the interference between two signals propagating on two pairs of cables. It is measured on one cable with respect to the interfering cable.

Reliability: The ability to perform the switching after millions of cycles or a long period of time.

Energy usage: It is the power consumed by the switching device.

Temperature resistance: The switching ability with desired behavior at different temperature conditions.

Insertion loss: It is the signal power loss resulting from the insertion of the switching device. It must be as small as possible.

Extinction ratio: The ratio of the output power in the on-state to the off-state. It should be as large as possible.

Polarization dependence loss: If the switch has different losses at two polarized states, the switch is said to have the polarization dependent loss. It should be as low as possible.

Scalability: The ability to build switches with large port counts from the small port counts without affecting the network performance.

2.4.2 Optical Switching Technologies

Some main optical switching technologies available today are listed below [29].

Optomechanical Switches: This is an oldest optical switching technology commercially available. The light beam is redirected by using electromechanical device. The mechanical devices like a prism or mirror are used to perform switching. The popular Micro-electro-mechanical (MEM) switches are a subcategory of optomechanical switches.

Micro-electro-mechanical Systems (MEMS) optical switches: They are widely used optical switching components and preferred for many applications. They are small in size because they are an assembly of tiny mechanical components. There are lots of variants in their fabrication process. Because of their small size and variations in the fabrication process, they have different characteristics. Hence, they are versatile in nature and most preferred for applications. It has a movable components suspended on a flexible structure above a base layer. The electrostatic or magnetic forces are applied between the base and the elevated component to move the component.

Electro-optic Switches: These switches are made from highly birefringent substrate material. This material exhibits the electro-optic effect which is the change in optical properties of the material with the change in the electrical field. The most common material is Lithium Niobate (LiNbO3). An electrical signal is fed as the control signal into the substrate of the device. This electrical field changes the refractive index of the material. The change in the refractive index manipulates the light through the pre-defined waveguide path to the desired port. These switches are very reliable and extremely fast, but they have a high insertion loss and a high polarization dependence loss.

Thermo-optic Switches: These switches work on the phenomenon of thermo-optical effect. The refractive index of the dielectric changes due to the change in the temperature of material. The two main switches in this category are Interferometric switches and digital optical switches.

Liquid crystal optical switches: This switch works on the principle of polarization control. When the light passes such a material, the polarization state of the light changes. The switch consists of a thin layer of liquid crystal between a pair of parallel glass plates. When voltage is applied between the plates, the polarization state of light changes. This property is used to switch the signal.

Acousto-optic switches: This switch works on the principle of change in refractive index due to the mechanical strain present in surface acoustic waves. This is called acousto-optic effect.

Semiconductor optical amplifier switches: These switches are the arrangement of semiconductor optical amplifiers. These can be used as the gates and let the signal pass through or stop based on the state required. Since they are amplifiers also and can allow the amplification, hence, they can restore the signal level.

Fiber Bragg grating based switches: These switches are based on the Bragg reflectors. It reflects the particular wavelengths and passes all the other wavelengths. It is achieved by creating a systematic variation in the refractive index inside the core of an optical fiber. For manufacturing, usually the germanium-doped silica fiber is used. This kind of fiber is photosensitive and changes its refractive index with the change in UV light exposure. Recent technologies also use polymer fibers for fiber grating.

Chapter 3 Stackable Optical Switching Fabric

3.1 Introduction

As discussed in Chapter 2, the scalability of optical switching fabrics is an important criteria of selection in optical networks. This section focuses on the scalability of optical switching fabric in existing networks. Normally, the scheduler design presumes that all the outgoing channels available in the DWDM links are accessible to all the incoming channels. There is no restriction on interconnectivity of channels. Hence, scheduler can connect all incoming channels to any available outgoing channel. As long as there is single optical fabric present in the core router, this methodology is good. But if there are stacks of optical fabrics instead of a single optical fabric in the DWDM links, such a methodology will not work. In this case scheduler has to have the information about the interconnectivity of input and output ports before scheduling any input signal.

The concept of stackable optical fabric is introduced in this chapter in order to provide an efficient solution for DWDM network upgrade. As the Internet traffic is growing day by day, the DWDM network may need upgrades quite often. To upgrade, the stacking of optical fabrics could be the cost-effective solution rather than replacing the old flat optical fabric with a new flat optical fabric. This thesis proposes a stackable approach which gives a clean, modular structure where the number of channels or ports can be easily added or removed without affecting the original arrangement. Stacking of optical fabric provides an easy option practically as well as economically. Figure 6 shows a flat optical fabric and Figure 7 shows stacks of optical fabric.

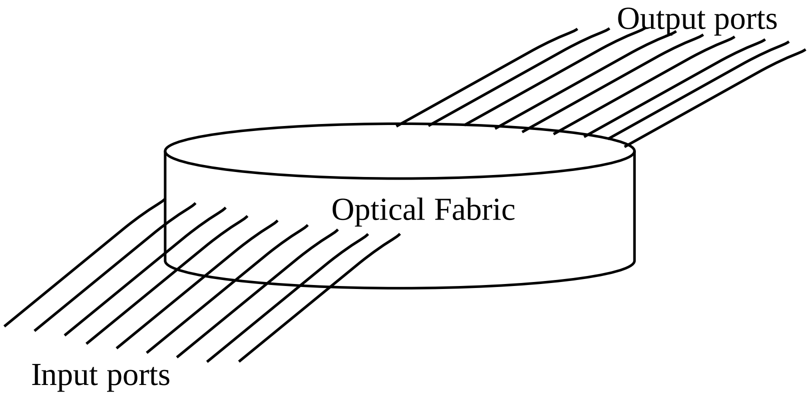


Figure 6: Big optical fabric

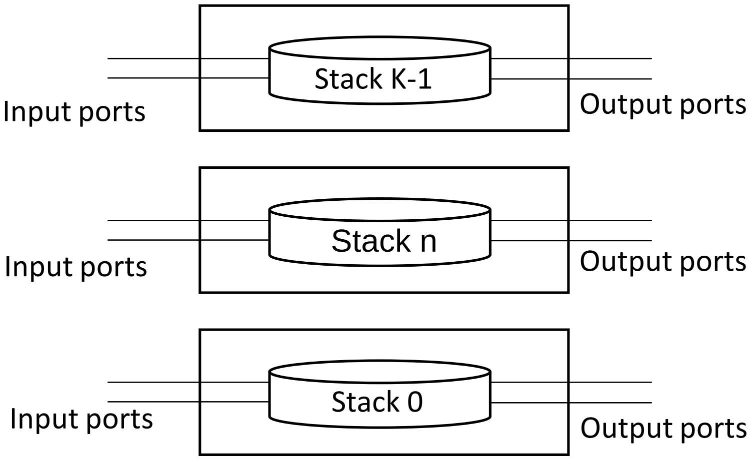


Figure 7: Stacks of optical fabric

The stacking of optical fabric gives the option of expanding optical fabric, without affecting the present configuration and infrastructure. Other benefits include low switching time. The bigger fabric usually encounters a larger switching time of optical switches which is not desirable in many applications. As mentioned earlier the low cost is another benefit of this arrangement. The modular configuration of stacks is highly flexible and very economical to expand.

3.2 Mathematical Representation

Figure 8 illustrates a basic structure of multi-stack optical fabric. It consists of multiple stacks of optical fabric, each of which carries multiple input and output ports. The limitation of this structure is that for any input port, only the output ports of the same stack are accessible for connection. Due to the isolation of stacks, the output ports of the same stack are accessible, but the output ports of other stacks are not. The input signal can come at the input port of any stack. The different output ports are connected to different destinations. At the time of scheduling of input request, the scheduler can schedule the input port to the output port of the same stack.

In Figure 8, there are K stacks ranging from *0* to *K-1*. Each stack caries *N* ports ranging from *0* to *N-1*. The following notation is used to represent the structure: *K* = Number of stacks, *N* = Number of input/output ports. The n^{th} port of k^{th} stack can be represented as $[P]_n^k$.

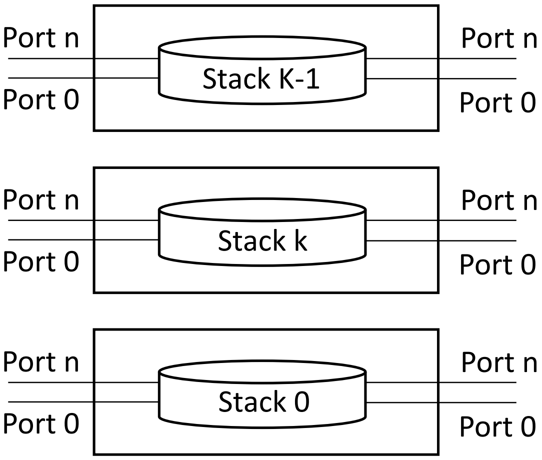


Figure 8: Multi stack optical fabric

An input burst, coming at port $[P]_n^k$, an input port $n\ (0 < n < N-1)$, of stack $k\ (0 < k < K-1)$, can be scheduled and transmitted to port $[P]_{\boldsymbol{m}}^{\boldsymbol{k}}$**,** an output port $m\ (0 < m < N-1)$, of stack $k\ (0 < k < K-1)$. According to the connectivity defined in the multi stack structure, the input port $[P]_n^k$ cannot be connected to an output port other than from stack k.

3.2.1 Physical Connectivity

As mentioned in Section 3.1, the stackable optical fabric has restricted connectivity between input and output ports. Figure 9 represents the structure of a multi stack fabric. In the stackable arrangement of fabric, not all output ports are reachable by all input ports. The output ports which are in the same stack can be reached by the corresponding input ports, but the output ports which are in different stacks cannot be reached by the input ports. They have no connectivity between them because they do not have the physical connection. The mathematical representation of port is defined as $[P]_n^k$, the n^{th} port in k^{th} stack. If function $f(i,j)$ represents the physical connectivity between two ports, its value would be

$$f(i,j) = \{\ [P]_m^i,\quad [P]_{\boldsymbol{n}}^{\boldsymbol{j}}\ \} = \begin{cases} 1 & if\ i = j, \\ 0 & if\ i! = j. \end{cases} \tag{1}$$

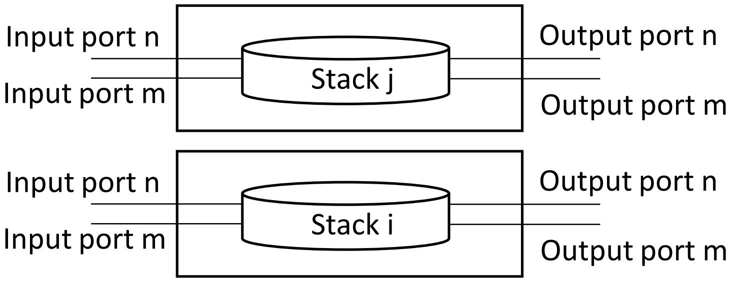


Figure 9: Basic structure of multi stack optical fabric

3.2.2 Hard Connectivity

By introducing the stackable optical fabric, there is a restricted on connectivity between the input and output ports, which might not be acceptable by many applications. As the Internet based applications are increasing, the demand for the connectivity is increasing as well. With the stackable optical structure, some restrictions on connectivity are inevitable, which arises due to the isolation of different stacks. There is a solution to avoid this restriction and to further improve the connectivity between stacks. The ports of different stacks can be connected externally using an optical fiber in order to provide more connectivity between the stacks.

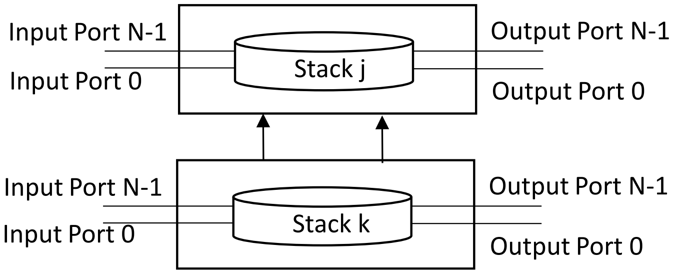


Figure 10: Hard connectivity between optical stacks

In Figure 10, the external connections give an option to schedule and allow the input signal of stack k to transport through the output port of stack j. In a practical scenario, direct fiber connections can be made between different stacks. Hence, improving the connectivity between input and output ports tremendously. In the later sections, it can be observed that by hard wiring different stacks, the number of interconnects can be improved dramatically in the stackable fabric.

For example the input burst $[P]_n^k$, coming at port n $(0 < n < N-1)$, of stack k $(0 < k < K-1)$. The available output ports for connection, before the hard connectivity is $[P]_m^k$, port m $(0 < m < N-1)$, on stack j $(0 < j < K-1)$.

If stack i and stack j are connected using hard wire, the available output ports for connection after the hard connectivity will be $[P]_m^j$ and $[P]_m^k$, Port m $(0 < m < N - 1)$, on stack j $(0 < \mathrm{j} < \mathrm{K} - 1)$ and Port m $(0 < m < N - 1)$, on stack k $(0 < k < K - 1)$.

This concept can be expanded where different stacks of an optical fabric are connected in several different ways. The configuration, in which the stacks are connected to each other in the fabric has an impact on switching time and the scheduling preference. This property can be defined as the degree of connectivity between input and output ports. Let α represent the degree of connectivity between input and output ports.

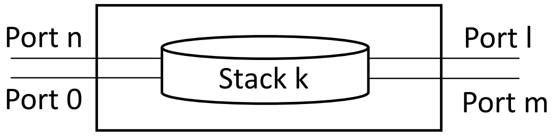


Figure 11: Degree of connectivity =1

Degree of connectivity between input and output ports is related to the number of stacks involved between these ports. As in Figure 11, the degree of connectivity between $[P]_n^k$ and $[P]_m^k$ is one. It can be represented as

$$\alpha\ ([P]_n^k\ \ [P]_m^k)\ = 1.$$

Also the following are true:

$$\alpha\ ([P]_n^k\ \ [P]_l^k)\ = 1,$$

$$\alpha\ ([P]_o^k\ \ [P]_m^k)\ = 1, \text{ and}$$

$$\alpha\ ([P]_o^k\ \ [P]_l^k)\ = 1.$$

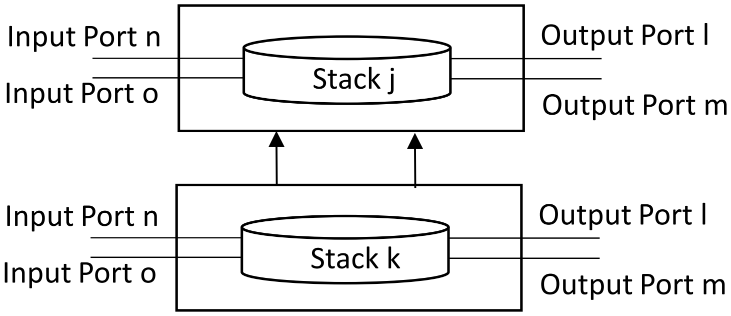


Figure 12: Degree of connectivity =2, between optical stacks

Figure 12 connects two different stacks *j* and *k*. The degree of connectivity between input ports of stack k $[P]_n^k$ and output port of stack j $[P]_l^j$ can be represented as

$$\alpha\ ([P]_n^k\ \ [P]_l^j)\ = 2.$$

Also, the degree of connectivity between $[P]_n^k$ and $[P]_m^j$ is,

$$\alpha\ ([P]_n^k\ \ [P]_m^j)\ = 2.$$

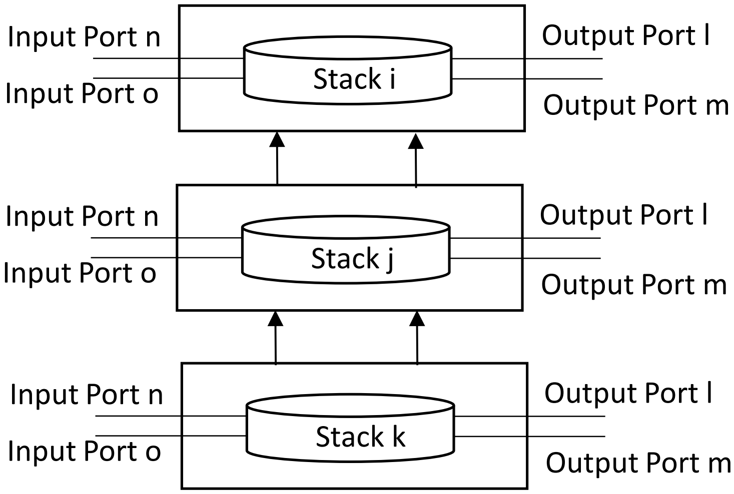


Figure 13: Degree of connectivity =3, between optical stacks

In Figure 13, the input port $[P]_n^k$ is connected to output port $[P]_m^i$ using three stacks among them. As a result, the degree of connectivity between these two ports can be represented as

$$\alpha\ ([P]_n^k\ \ [P]_m^i)\ = 3.$$

For ports $[P]_n^k$ and $[P]_m^j$, the degree of connectivity would be

$$\alpha\ ([P]_n^k\ \ [P]_m^j)\ = 2.$$

3.2.3 Multiple Degree Connectivity

For the practical purpose, the optical fabric can be visualized as having multiple degrees of connectivity between ports of two stacks. The two ports can be connected physically by direct connection between two stacks or by indirect connection via some other stack. Figure 14 below has two different degrees of connectivity between $[P]_n^i$ input port *n* of stack *i* and $[P]_m^k$ output port *m* of stack *k*.

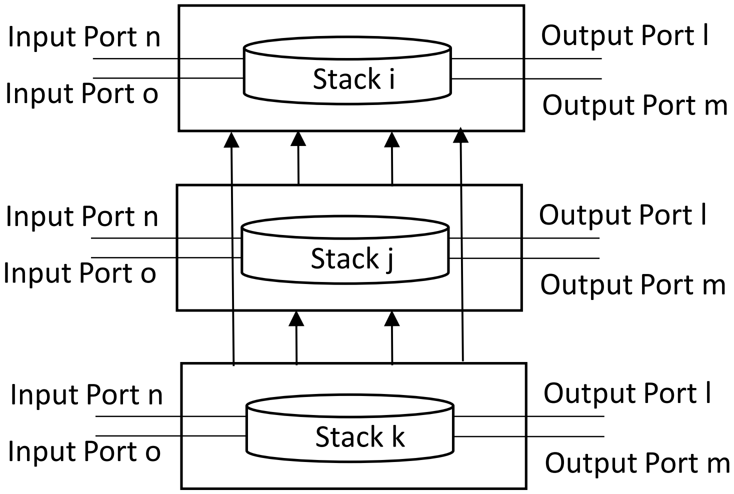


Figure 14: Multiple degree of connectivity between optical stacks

These degrees of connectivity can be represented as

$$\alpha\ ([P]_n^i\ \ [P]_m^k)\ = 2, \text{ and}$$

$$\alpha\ ([P]_n^i\ \ [P]_m^k)\ = 3,$$

via stack *j* between stack *i* and stack *k*.

The multiple degree of connectivity between two stacks gives more opportunity to make a connection between input and output ports. The point to be noted is that, as the degree of connectivity increases, the cumulative optical loss increases. The user always wants to reduce optical losses, and hence, picks the minimum degree of connectivity between two ports. So for any incoming burst, if the scheduler finds multiple degrees of connectivity, it should prefer to connect using minimum degree of connectivity and reduce the optical losses.

3.2.4 Concept of Soft Service

There is one factor which needs to be taken into account as far as the connectivity of input and output ports are concerned. For the input signal coming at any input port, there could be the preference or restriction on some output ports. Some applications may want to allow the input burst going to some specific output ports or sometimes want to restrict the input burst going to some specific output ports. This service can help provide flexibility to applications and can be used for multiple reasons. Define this service as Soft Service. The concept of soft service can be used for many reasons, one among which could be the security of data. This is one more aspect of ports connectivity apart from the physical and hard connectivity. Though in one stack, all the input ports could be connected to all output ports of the same stack, the network administrator or application provider can impose some restrictions and can restrict some output ports from being connected for some particular input signal. Ports can be designated based on the application. This usage is optional and can be requested by a “service bit” present in the incoming signal. If the “service bit” is not present, there is no soft restriction on the port's connectivity. On the contrary, if the "service bit" is present, there is a service restriction on ports.

Let function $g(x)$ represents the soft connectivity, where x is the "service bit" present in the incoming signal. Hence,

$g(0)$ → "Service bit" is not present. No restriction is imposed due to service bit. Each input port can be connected to all the output ports present in the stack.

$g(1)$ → "Service bit" is present. Hence, there is a service dependence. It will be checked further for allowable output ports using "service ID" or "service parameter" present in the request.

Sample values of the function could be

$g(0) = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ all the connections are possible, the default value of the function, and

$g(1) = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ all the connection are possible, except one output port.

3.2.5 Overall Connectivity

The overall or total connectivity between two ports can be defined as the product of hard and soft connectivity. Let δ be the total connectivity.

Total connectivity = hard connectivity * soft connectivity

Referring to Sections 3.3.1 and 3.3.4,

$\delta = f(i,j) * g(x)$, when expanded it becomes

$$\delta = \{\ [P]_m^i, \quad [P]_n^j\ \} * g(x). \tag{2}$$

3.3 Summary

If one chooses the option of stackable optical structure, there are few factors which need to be taken into account before building such an infrastructure. The simplest stackable structure has its physical connectivity defined where the input and the output ports of the same stacks are reachable. Different stacks of the structure are totally isolated with each other and hence, no connection is possible between them.

In order to improve the connectivity of the infrastructure, different stacks can be connected manually using external optical fibers. There are a variety of possible ways to build a manual connection. In order to keep track of those different ways, the degree of connectivity is introduced as a property of the stackable optical infrastructure.

The application provider may want to have some flexibility in connectivity. Input signal may not go to a particular output port even though the physical connection is possible. This could be motivated by security reasons. The optional soft service introduced in this chapter provides this capability. By combining all the parameters defined above, one can have a complete picture of a stackable optical fabric.

Chapter 4 Reconfigurable Scheduler for Stackable Optical Fabric

4.1 General Concept

Chapter 3 describes the concept of stackable optical fabric. Though stackable approach is modular, economic and easy to expand, it is observed that replacing big fabric with stacks of small optical fabric reduces the connectivity between the ports. Hence, the concept of hard connectivity is integrated so that the connectivity between the ports can be improved further.

The concept of soft connectivity is also integrated in the above concept. It is to offer the application providers the capability of choosing output ports of their choice or restrict some output ports from being connected. The network administrator may want to reserve some output ports for some special inputs hence, restricting them for regular usage.

After the concept of stackable optical fabric finalizes, there is a need for a scheduler which can work with stackable optical fabric. The scheduler should be able to schedule the requests taking into account the physical, hard, and soft connectivity between the ports. The scheduler should also consider the degree of connectivity between ports which was introduced to improve the interconnectivity. The "reconfigurable scheduler for stackable optical fabric" is proposed to meet such requirements.

To deploy the stackable optical fabric in the core routers, the scheduler needs to be able to schedule the input requests taking into account the internal or external connectivity between the ports. It should be able to pre-store the intrinsic connectivity of the ports and then be able to schedule the input requests based on the pre-stored information. It should also have the capability to incorporate the soft connectivity information present in the input requests as discussed in Section 3.3.4. It should interpret the “soft service” information from the BHC and act based on that information. In this way, it can restrict some output ports from being connected. These restricted output ports can be designated by the network administrator for some specific application or service.

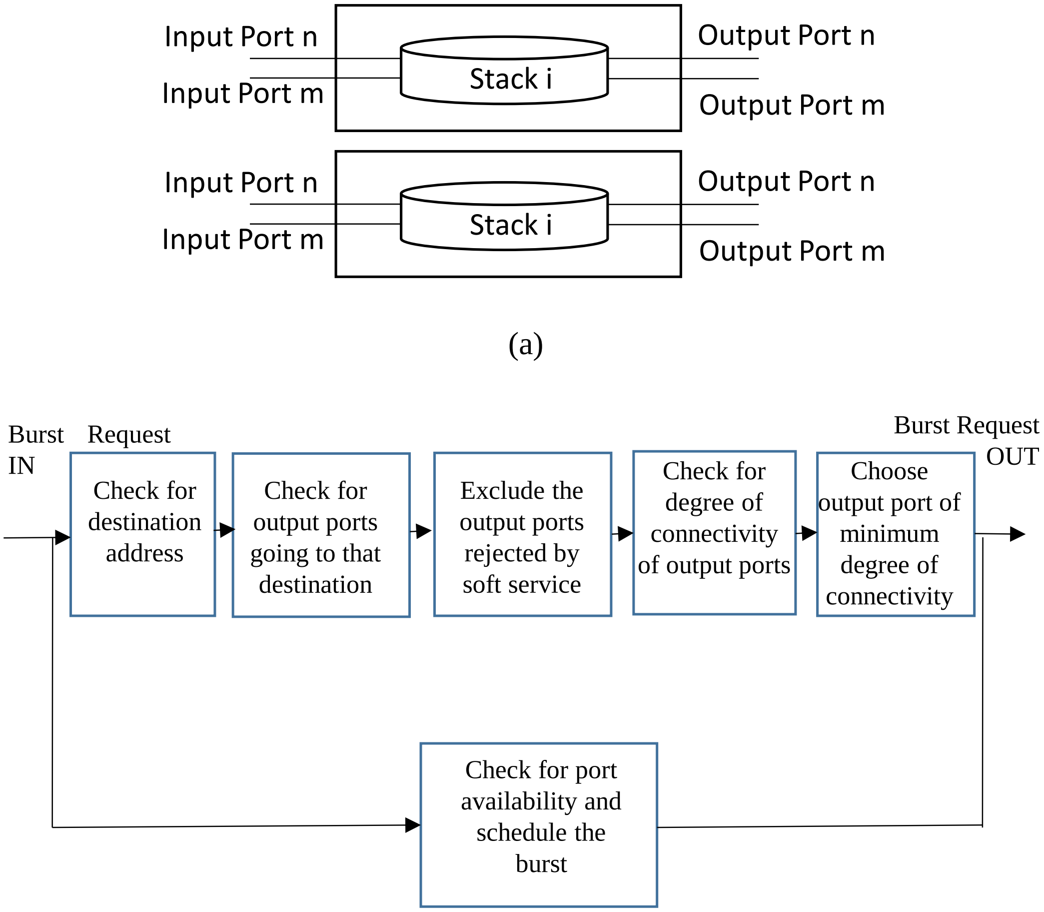
The scheduler should also be able to handle various degrees of connectivity between the ports. In summary, at the time of input burst arrival, the scheduler should be able to check between the two ports:

a) Where they are physically connected,
b) How many different types of connections are possible between them, and
c) The degree of connectivity.

Based on the above information, the scheduler should be able to verify the possibility of scheduling the request on different stacks for different configurations concurrently. Later, it should choose the best possible connection with the lowest degree of connectivity between input and output ports. The whole process should be referred to a central database which is dynamically updated after each scheduled event. This central database should be easy to maintain and update. It should be well structured so that at the time of future updates of optical fabric, the update of the database is easy and flawless.

4.2 Operation

The stackable optical fabric has some intrinsic properties attached to it. Some of them are the ways different stacks are connected to each other, the connectivity of input ports to all the output ports, the presence of hard wiring between stacks and so on. The need is to pre-record this information in the form of a central database before the basic operation of a reconfigurable scheduler for stackable optical fabric can be achieved. The proposed scheduler would refer to this information on the fly and update them when needed. Figure 15 shows the example block diagram of the proposed scheduler, which uses the internal and external connectivity information so that it can schedule the input request efficiently and effectively.



(a)

(b)

Figure 15: (a) Stackable optical fabric (b) Scheduler for stackable optical fabric

In order to schedule a new input request, it observed that as the burst request comes, the scheduler checks for the target destination information in the request. The central database contains the information about different destinations and the lists of output ports going to that destination. The scheduler refers to that information to figure out which output ports are going to that particular destination.

The scheduler then checks for the “soft service” requirement. If the input request contains this information, the scheduler makes reference to the central database again for soft service requirements. This information restricts some output ports to be connected for that particular input request. Now scheduler finalizes the allowed output ports for making a connection to that particular input port at which the input request came.

At this stage, the scheduler checks for the degree of connectivity between each input-output port pair. It tries to schedule the input request on an output port, which is available and has minimum degree of connectivity. The benefits of a minimum degree of connectivity have been discussed in Section 3.2.3.

4.3 Algorithm of the Scheduler

4.3.1 Algorithm of Single Stack Scheduler with No Soft Service

The single stack structure is the simplest stackable optical fabric to work with. It is the basic building block of a multi-stack fabric. A scheduler designed for the single stack structure can serve as a building block for the target reconfigurable scheduler for stackable optical fabric. As already discussed in Section 3.2.1, all types of stacks have some intrinsic physical connectivity defined. During the scheduling phase, the scheduler has to refer to this information before making a connection between input and output ports. The basic structure of a router can be visualized as routers connected to different routers in a network. The output ports are going to different destinations. The optical burst or Internet request can come at any channel or input port. The input request has the information about the output destination.

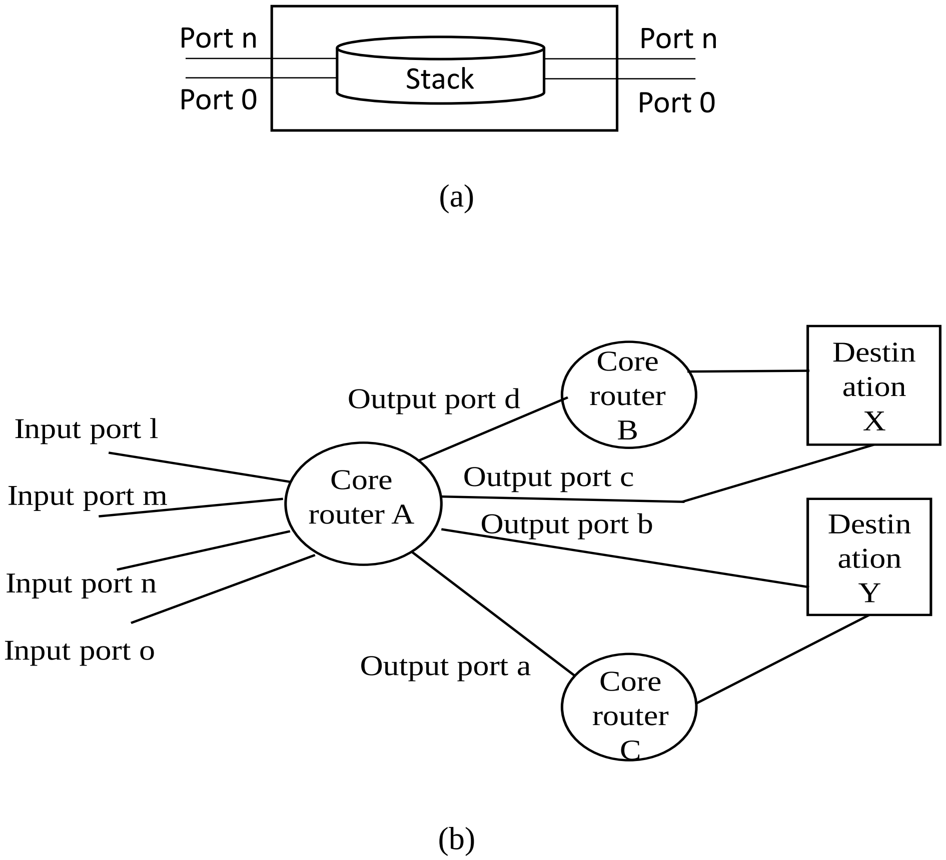


Figure 16: (a) Single stack fabric (b) basic scheduler for single stack fabric

Assume a single stack optical fabric, stack k, which has N input channels $(0 < n < N - 1)$, and M output channels$(0 < m < M - 1)$. The input burst request comes at input port $[P]_o^k$, the input port o of stack k. The burst has the following information: Input channel = $[P]_o^k$; Output destination address = X; Burst offset = W; Burst length = L. Before making a connection, the scheduler needs to know which output ports are going to destination X. Based on this, the scheduler would try to connect the input port to that output port. The scheduler needs an Index Table which keeps track of the connectivity information of output ports and the target destinations.

Table 1: Destination Index Table

Destination Address	$[P]_a^k$	$[P]_b^k$	$[P]_c^k$	$[P]_d^k$
X = 001100	0	0	1	1
Y = 001001	1	1	0	0

From the example shown in Figure 16, it can be seen that destination X can be accessed by output ports *c* and *d*. While scheduling the request, the scheduler will refer to Table 1, the destination index table. Since there is no “soft service” in the burst request, there is no service restriction on the output ports. The scheduler can pick any available output ports between $[P]_c^k$ and $[P]_d^k$. After this decision is made, the scheduler will make a connection between the input and output ports.

To summarize the function of a single stack scheduler, consider that in a single stack, any input port can be connected to any output port. There is no restriction on the physical or internal connectivity, external connectivity and the soft connectivity. This is the simplest configuration possible for a stackable scheduler. The central database will maintain a table called “destination index table”. This will have the information on how different output ports are connected to different destinations. The scheduler will refer to this database before scheduling a request. Once it obtains the list of allowed output ports, the scheduler will make a selection based on the availability of output ports.

4.3.2 Algorithm of Single Stack Scheduler with Soft Service Requirement

Consider a single stack fabric same as in Section 4.3.1. The additional feature to comply with, is the soft service requirement present in the input request. The scheduler described in Section 4.3.1 is used as a building block. The only challenge is to cater to the soft service requested by the application vendor.

The procedure to figure out the allowed output ports as discussed in Section 4.3.1 will remain the same. Once the list of allowed output port is final, the “service bit” present in input request is checked. The “service bit” present in BHC is introduced by the application vendor, if he needs to put some restriction on the output ports. This is discussed in detail in Section 3.2.2. The central database has pre-stored mapping between the “service bit” and different output ports. This is presented in Table 3, the service restriction table. The scheduler at this stage will refer to the service restriction table and update its list of allowed output ports. After it has the updated list of allowed output ports, the scheduler will check the availability of output port at the time of burst arrival. As soon as it finds any output port available, it schedules the input request. After scheduling the ports, the scheduler hardware will make the actual connection between input and output ports.

The above concept is illustrated in the following example. Assume a single stack optical fabric stack k which has N input channels $(0 < n < N - 1)$, and M output channels $(0 < m < M - 1)$. The burst request comes at $[P]_o^k$, the input port o of stack k. The burst has the following information. Input channel = $[P]_o^k$; Output destination address = X; Burst offset = W; Burst length = L; Service bit = Q. The scheduler will refer to the destination index table, as described in the previous section, to check for the connectivity of output ports and destinations. After it finalizes the output ports, it will process the service request need of the input request.

Table 2: Destination Index Table for Soft Service

Destination Address	$[P]_a^k$	$[P]_b^k$	$[P]_c^k$	$[P]_d^k$
X = 001100	0	0	1	1
Y = 001001	1	1	0	0

From the example shown in Figure 16, it can be seen that destination *X* can be accessed by the output ports *c* and *d*. The scheduler will refer to the Table 2, the destination index table. Since there is a “service bit” present in the burst request, the scheduler will refer to the Table 3, the service restriction table. The service restriction table will restrict or allow the output ports depending on the service bit present in the BHC. In the example, the allowed ports from the destination index table are $[P]_c^k$ and $[P]_d^k$. Based on the service restriction table, the port $[P]_d^k$ is excluded from the list of allowed output ports. Hence, the allowed output port for connection is now $[P]_c^k$. The scheduler will check the availability of this output port and schedule the input burst, if the output port $[P]_c^k$ is available at the burst arrival time.

Table 3: Service Restriction Table for Soft Service

Service bit	$[P]_a^k$	$[P]_b^k$	$[P]_c^k$	$[P]_d^k$
Q = 001	0	0	1	0
R = 011	1	1	1	0
S = 010	0	0	1	1

To summarize, in order to support the functionality of a single stack scheduler with soft service request, the scheduler defined in Section 4.3.1 is taken as a building block. As the burst request comes, the scheduler checks for the destination information in the burst request. After that, it refers to the destination index table to check how many output port are connected to that particular destination, and prepares a list of allowed output ports. Later, it refers to a service restriction table where it gets the information about the ports allowed for any service bit entry. After referring to this table, the scheduler modifies its list of allowed output ports. Now, for all allowed output ports, the scheduler checks for their availability, and schedule the input request as soon as it identifies the available output port.

4.3.3 Algorithm of Multiple Stacks Scheduler with External Connections and Soft Service Request

Consider an example where there is a multi-stack optical fabric. It also has hard wired external connections. Different stacks of the optical fabric are connected by external fibers in a way that they have different degrees of connectivity between different stacks. The input request coming to this fabric also has a “soft service” request from the application vendor. A scheduler which can work in line with the requirements of this multi-stack optical fabric is described as follows. The challenges for this scheduler are to support multiple stacks with external connections of different degrees of connectivity, and soft service requests at the same time.

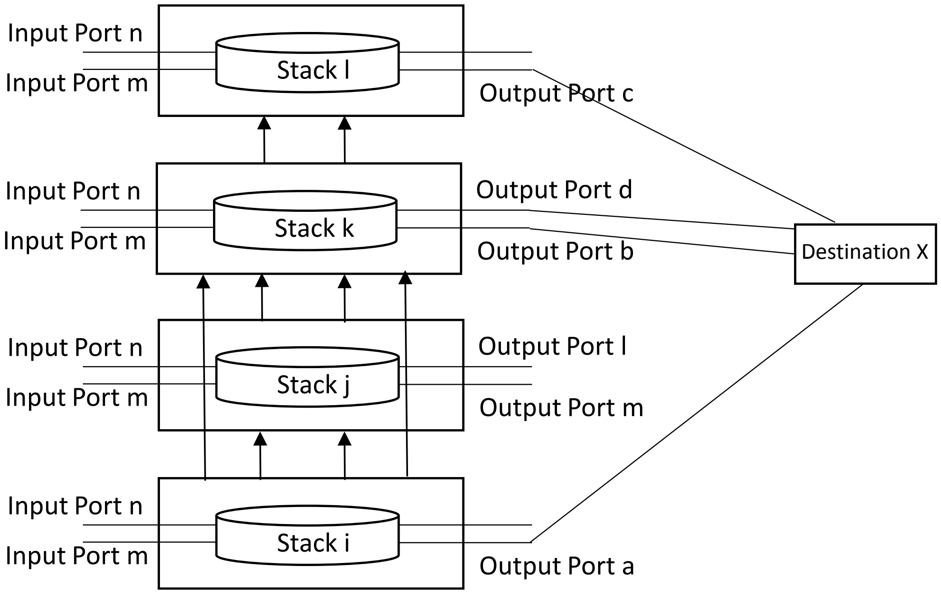


Figure 17 Multi Stack Optical Fabric in a Network

This scheduler needs to address the challenges to work with the physical connectivity of different stacks and external connectivity with different degree of connectivity. The scheduler discussed in Section 4.3.2 will be considered as a building block. The additional features such as the hard or external connectivity between different stacks with different degrees of connectivity needs to be supported by the scheduler. At the time of the request arrival, the scheduler will refer to the destination index table and the service restriction table as in Section 4.3.2. After this stage, it will have an allowed list of output port for making connections.

The operations of the scheduler is described in the following example where there are multiple stacks in the optical fabric. The stacks are stack i, stack j, stack k, and stack l as in Figure 17. The burst request comes on Port m of stack i. The burst has the following information: Input channel = $[P]_m^i$; Output destination address = X; Burst offset = W;

Burst length = *L;* Service bit = *R*. As mentioned in Table 4, the destination index table, the target destination *X* is accessible by following ports.

Allowed output ports= $[P]_a^i$,$[P]_b^k$, $[P]_c^l$ and $[P]_d^k$.

Table 4: Destination Index Table for External Connections

Destination Address	$[P]_a^i$	$[P]_b^k$	$[P]_c^l$	$[P]_d^k$
X = 001100	1	1	1	1
Y = 001001	1	1	1	0

The burst request has the service bit = R, which excludes $[P]_d^k$ from the connection. Hence, allowed output ports after considering Table 5, service bit are $[P]_a^i$, $[P]_b^k$ and $[P]_c^l$.

Table 5: Service Restriction Table for External Connections

Service bit	$[P]_a^i$	$[P]_b^k$	$[P]_c^l$	$[P]_d^k$
Q = 001	0	0	1	0
R = 011	1	1	1	0
S = 010	0	0	1	1

Now the scheduler will refer to a new input-output connectivity table which is pre-stored, The example input-output connectivity table is shown in Table 6.

Table 6: Input-Output Connectivity Table for External Connections

Input ports	$[P]_a^i$	$[P]_b^i$	$[P]_m^j$	$[P]_b^k$	$[P]_d^k$	$[P]_c^l$
$[P]_m^i$	1	1	2	2,3	2,3	3,4
$[P]_n^i$	1	1	2	2,3	2,3	3,4
$[P]_a^j$	2	2	1	2	2	3

This table represents how all the input ports are connected to all the output ports of stacks and with which degree of connectivity. For the example taken above, the degree of connectivity found is as follows:

$$\alpha\ ([P]_m^i\ \ [P]_a^i)\ = 1,$$

$$\alpha\ ([P]_m^i\ \ [P]_b^k)\ = 2, 3, \text{ and}$$

$$\alpha\ ([P]_m^i\ \ [P]_c^l)\ = 3, 4.$$

The scheduler will check for the availability of these ports and pick the output port which has the lowest degree of connectivity.

The functionality of the multi-stack scheduler with external connections between the stacks, and soft service requests are summarized as follows. The scheduler defined in Section 4.3.2 will be taken as a building block. As soon as the request comes at any input port, the scheduler checks for its destination and refers to the destination index table as defined in Table 4. It makes a list of allowed output ports. Then it checks whether the service bit is present in the burst request. If service bit is present, then it refers to the service restriction table as in Table 5. The service restriction table allows or restricts some output ports based on the value of service bit. Hence, the list of allowed output port is modified after this. Later it checks for the input-output connectivity table as defined in Table 6 and figures out how the input port is connected to different allowed output ports and with which degree of connectivity. It stores this information in the list of allowed output ports. Now it sorts the list of allowed output ports based on the degree of connectivity. Finally, it checks each output ports from the list of allowed output ports one by one for its availability at the burst arrival time and schedule the request when it gets the first output port available. After this, it populates the output port for the time defined by the burst length in the burst request.

The external connections made between different stacks improve the number of interconnects. Also, if two stacks have multiple degrees of connectivity, the lowest degree of connectivity has the minimum number of stacks involved between input and output ports. Hence, the switching time and optical losses of optical components are less in this case. The scheduler algorithm always picks the lowest degree of connectivity between the stacks. The next section describes the flow chart of a reconfigurable scheduler for stackable optical fabric. The soft service request is optional and the scheduler has a decision making point for this, in the flow chart. The flow chart and pseudo code of scheduler described in the next section executes exactly the same sequence of events as the algorithm of the scheduler discussed in the previous section.

4.4 Flow Chart

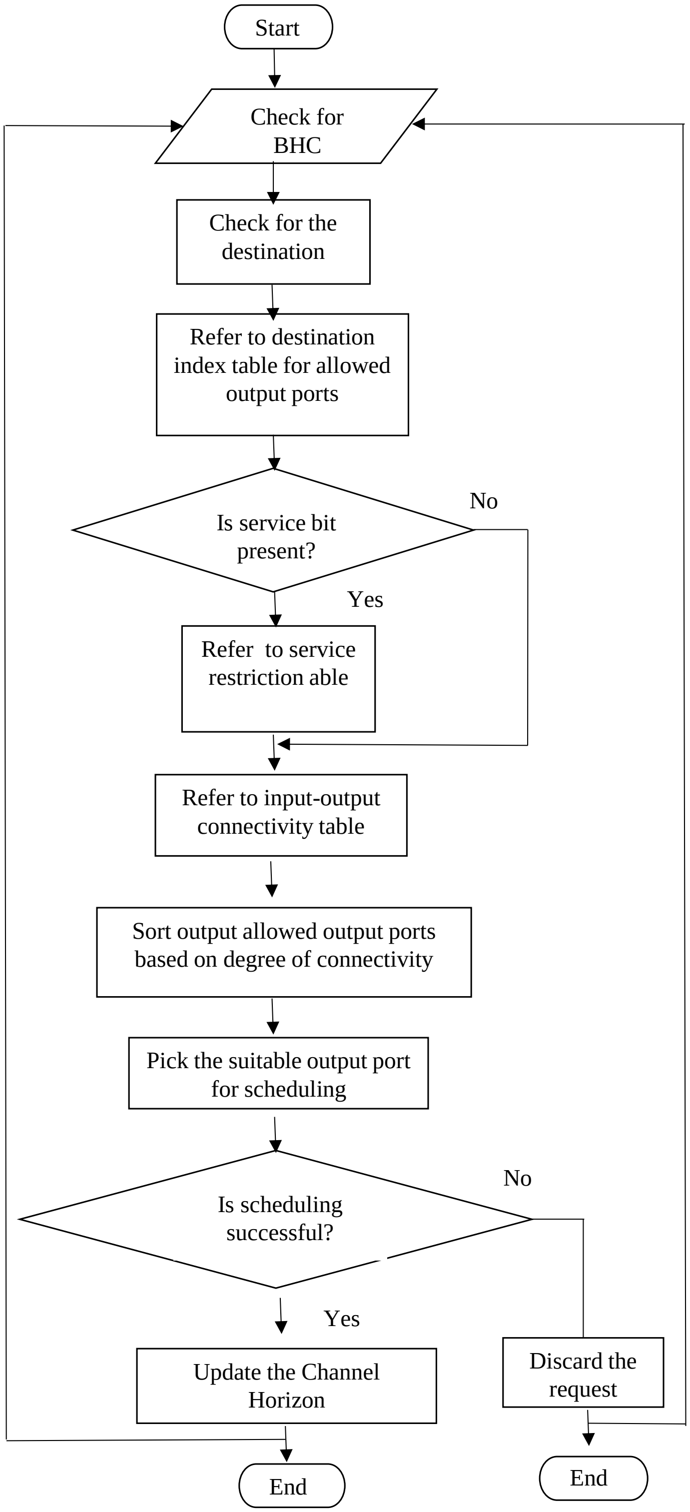


Figure 18: Flow chart of stackable scheduler

The flow chart of a reconfigurable scheduler for stackable optical fabric is described in Figure 18 . The scheduler continuously checks for the BHC arrival. As soon the BHC arrives, the scheduler extracts some information from the BHC such as its destination, the burst arrival time, the burst length and the input port at which the burst will arrive. Then it refers to the destination index table to list all the allowed output ports connected to that destination. The example destination index table is shown in Table 4.

It then checks, in the input request, whether it contains the soft service request or not. If yes, then it refers to service restriction table and modifies the list of allowed output ports based on the restriction present in the service restriction table; otherwise, it goes forward to refer to the input-output connectivity table to check how the input ports are connected to the output ports. The example service restriction table is shown in Table 5.

The input-output connectivity table as shown in the Table 6, has the connectivity information between the input and output ports along with the additional information about the degree of connectivity. The degree of connectivity is described in Section 3.2.2. Once the list of allowed output ports is finalized, it is sorted based on the degree of connectivity.

Now the scheduler checks all the allowed output ports for their availability at the time of burst arrival. As soon as it gets an output port available at the time of the burst arrival, it finalizes that port and sends a signal to configure the optical fabric. It occupies that output port for the time of burst length defined in BHC. If all the output ports mentioned in the list of allowed output ports are busy at the time of the burst arrival and no suitable output port is available for connection, it discards the input request. In both the cases, a successful scheduling and an unsuccessful scheduling, the scheduler moves on to check for the new BHC request.

4.5 Pseudo Code

The pseudo code for the proposed scheduler is described in this section. Let “BHC_in” be the burst request which is called the burst header cell and arrives an “offset time” earlier than the actual burst. The burst request contains the following information about the actual burst, the burst destination, burst offset, burst length, burst input, and service_bit, if soft service is requested from the vendor.

There is a centralized database, which is predefined and stored in the memory. It has the intrinsic connectivity information of stackable optical fabric defined in it. This database contains tables such as DestinationIndexTable, ServiceRestrictionTable and inputOutputConnectivityTable. The database is described in the form of tables as defined in Section 4.3.3. The allowed_out_port is the temporary register which stores the list of allowed output ports for connection. This register gets updated dynamically as the code progresses. After storing the burst related information in different registers, the scheduler puts values in allowed_out_port register, after reading the information from DestinationIndexTable. Then, it checks if the “service bit” is present in the request or not. If the “service bit” is present, it updates the allowed_out_port register again, based on the information in ServiceRestrictionTable. Otherwise, it just moves on with previously stored allowed_out_port register file.

In order to check how the input and output ports of different stacks are connected in the fabric, the scheduler refers to inputOutputConnectivityTable. It updates its allowed_out_port register again, based on the values in inputOutputConnectivityTable. After updating the contents of allowed_out_port register, the scheduler sorts this file in the ascending order of the degree of connectivity. For each entry in the allowed_out_port file, the scheduler checks for the availability of the output ports at the time of burst arrival. As soon as it finds a suitable output port, it selects that output port for the connection. It occupies that output port for the time of burst_length. If the scheduler does not find any suitable output port available for the connection, it discards that input request and moves on to check for a new BHC request. It repeats the whole process for each incoming BHC request.

The algorithm described as the following pseudo code.

```
while (BHC_in! = 0) {
        channel_select = 0;
        burst_destination = BHC.destination;
        burst_offset = BHC.offset;
        burst_length = BHC.lenght;
        burst_input = BHC.input;
                allowed_out_port = read(DestinationIndexTable);
if (bhc_service_bit = 1)
        begin
                allowed_out_port = read(ServiceRestrictionTable);
        end
                allowed_out_port = read(inputOutputConnectivityTable);
                allowed_out_port_sort = sort (allowed_out_port);

for (allowed_out_port_sort)
        find_channel(bhc_input, allowed_out_port_sort);
```

```
        if  (find_channel != -1)
                return (find_channel);
                update_channel(find_channel);
        else
                return(-1);
        if (find_channel != -1)
                channel_select = 1;
        else channel_select = 0;
}
```

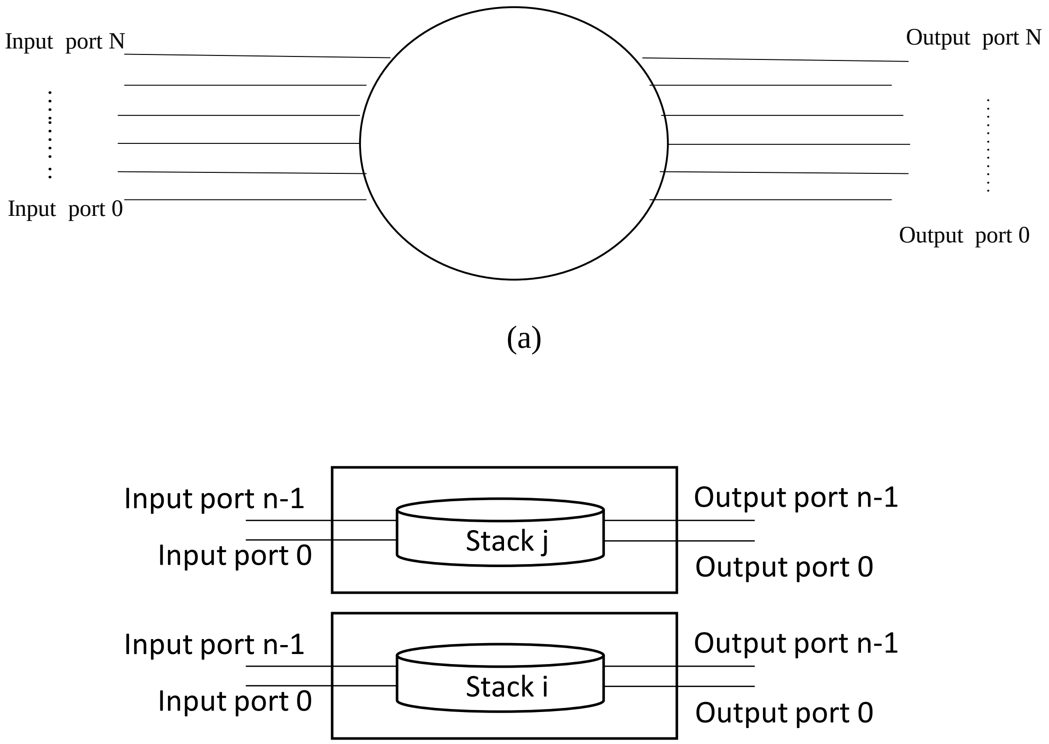
The pseudo code described above, covers all the combination of stackable optical fabric; optical fabric with single stack, multi-stack with no soft service requirement, multi-stack with soft service requirement, externally connected multi-stack with and without soft service requirement. It has similar processes as the scheduling algorithm defined in previous sections. In the next chapter, the cost of a stackable optical fabric is evaluated and compared with the equivalent flat optical fabric, followed by a prototype implementation of the scheduler.

Chapter 5 Analysis and Prototyping

5.1 Cost Model of Stackable Optical Fabric

In this thesis a stackable optical fabric for DWDM network is proposed. This option provides a flexible, easy to implement and cost effective optical fabric structure. However, with this approach, some connectivity between the ports is reduced. In the stackable fabric, input ports cannot connect to all output ports, as in the case of equivalent big flat optical fabric. This section compares both flat and stackable optical fabrics for the possible number of interconnections, if the number of input ports is same in both the cases. The interconnections between input and output ports are considered as the cost of the network. In case of big flat fabric, the number of interconnects is maximum, and hence, the cost is the maximum. Such optical fabric shows best performance with higher cost. There is a trade-off between the cost and connectivity.

The number of interconnects or the cost also increases as the external connections between the stacks or the degree of connectivity is introduced, with improved connectivity. Because of this the stackable optical fabric is a flexible option to choose between cost and connectivity, one can select the minimum number of stacks suited to the application and thereby pay the optimum cost of the optical fabric. On the other hand, flat fabric does not provide such flexibility.



(a)

(b)

Figure 19: (a) Flat optical fabric (b) Equivalent stackable optical fabric

As discussed in the previous section, the number of interconnections can be considered as a cost. Figure 19 (a) shows a network with flat optical fabric, where there are *N* ports on each side. Figure 19 (b) shows a network with equivalent stackable optical fabric, where there are *n* ports on each side. The equivalent stackable fabric has *S* number of stack having *n* ports on each side, where

$$n = \frac{N}{S}. \tag{3}$$

If *C* represents the number of interconnections possible in any optical fabric, then as mentioned in Figure 19 (a),

$$C_{Flat} = N^2, \tag{4}$$

where *N* is the number of ports on each side of the flat optical fabric.

Consider the equivalent stackable optical fabric, which has n number of ports on each side. There are *S* stacks and *N* ports distributed in *S* stacks. Referring to Equation (3), $n = \frac{N}{S}$. Consider the case where each stack has *D* degree of connectivity. Referring to Figure 19 (b), the total number of interconnections in the stackable optical fabric will be

$$C_{Stack} = [n + (D - 1)]^2 * S.$$

Replacing n with $\frac{N}{S}$

$$C_{Stack} = [\frac{N}{S} + (D - 1)]^2 * S, \tag{5}$$

where: *n* = number of ports each side, *S* = number of stacks, *D* = degree of connectivity.

5.1.1 Low Degree of Connectivity

Figure 20 shows the plot of the number of interconnections against the number of ports. The first curve (stack = 1) shows the number of interconnections for the flat optical fabric which is considered as a reference. The other curves represent the optical fabric with stack number 2, 3, and 4. The horizontal axis shows the number of ports and the vertical axis represents the number of interconnects.

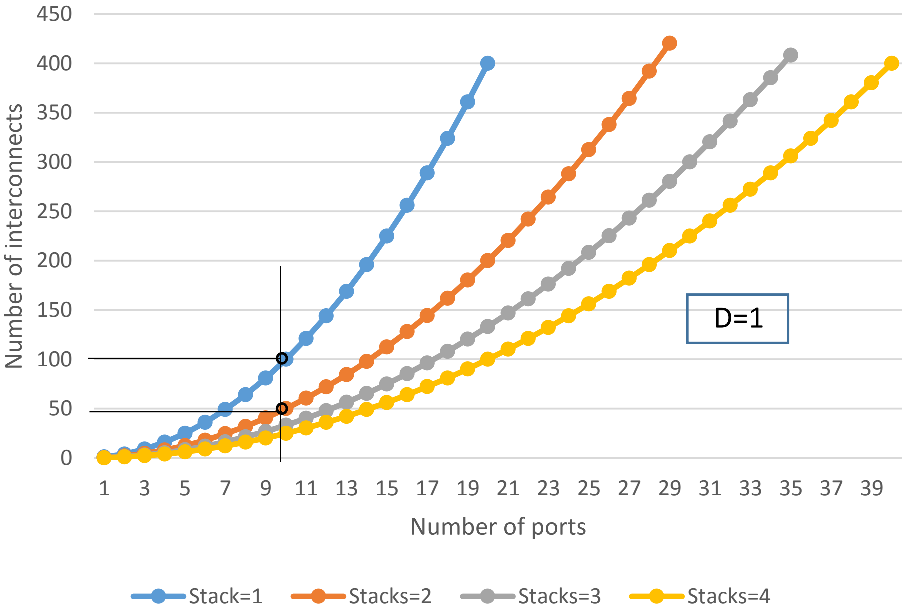


Figure 20: Number of interconnects versus number of ports with D=1

For all the curves, the degree of connectivity is considered as 1. As the stacks in optical fabric increases, the number of interconnects for a particular number of ports or the cost drops down as compared to a flat fabric. As an example, if the number of ports is 10, then $N = 10$. Referring to Equation (4),

$$C_{Flat} = 100.$$

For the equivalent stackable optical fabric with the number of ports $N = 10$, the number of stacks $S = 2$, the degree of connectivity $D = 1$. Referring to Equation (5),

$$C_{Stack} = [\frac{N}{S} + (D - 1)]^2 \ * \mathrm{S},$$

$$C_{Stack} = [\frac{10}{2} + (1 - 1)]^2 \ * 2, \text{ and}$$

$$C_{Stack} = 50.$$

As the number of stack increases, the C_{Stack} drops more. This trend is visible in the plot of Figure 21.

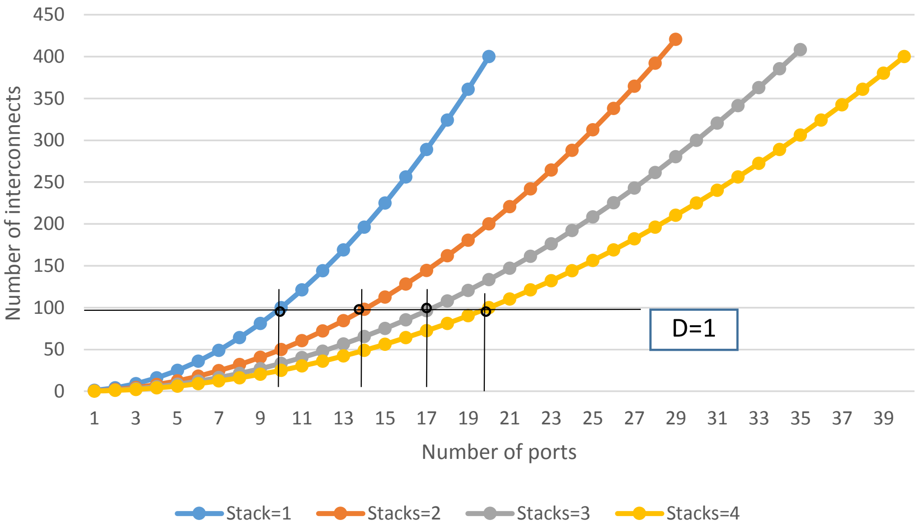


Figure 21: Equivalent number of interconnections for different stack sizes

The above figure explains that the same number of connectivity as of flat fabric ($C_{Flat} = 100$) with the number of ports *N* = *10*, can be seen in different stackable structures which has more number of ports. If the number of stacks are *S*= *2*, then *N* = *14*; if the number of stacks are *S* = *3*, then *N* = *17*; and if the number of stacks *S* = *4*, then *N* = *20*. The above analysis is done for degree of connectivity *D* = *1*.

5.1.2 Higher Degree of Connectivity

Degree of Connectivity = 2

When the external connections are added between the stacks, the number of connectivity of the stackable optical fabric increases tremendously. The degree of connectivity represents the number of stacks involved between the input and output ports.

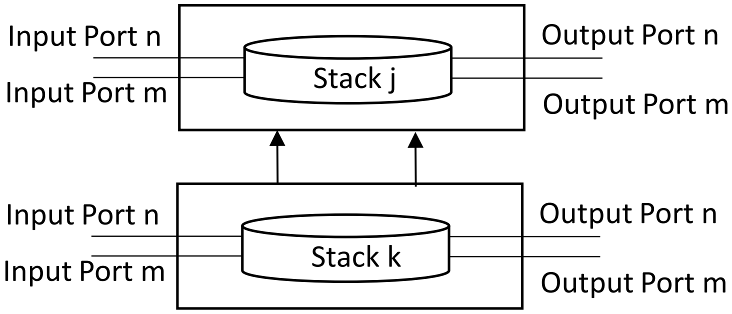


Figure 22: Multiple stacks of optical fabric with D = 2

Figure 22 shows how different stacks are manually connected by the external optical fibers. Between stack *k* and stack *j*, *D = 2*. Referring to the Equation (5),

$C_{Stack} = [\frac{N}{S} + (D-1)]^2 * $ S, when expanded, it becomes

$$C_{Stack} = [\frac{N}{S} + 1]^2 * S. \qquad (6)$$

These are the number of interconnects in a stackable optical fabric. In case of a flat fabric, the number of interconnects are N^2 as mentioned in Equation (4). Figure 23 shows the plot between the numbers of interconnects and the number of ports when the degree of connectivity is 2. The first curve (stack = 1) shows the plot of equivalent flat fabric while the other plots are for stack number 2, 3, and 4. The flat fabric has the maximum number of interconnects for any given number of input ports. As the number of stack increases, the number of interconnects decreases for a given number of input ports.

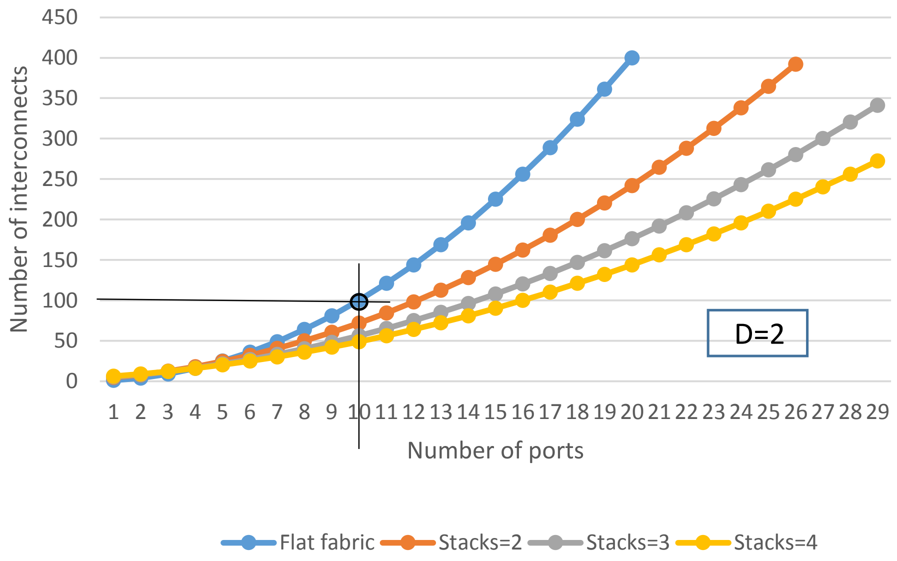


Figure 23: Number of interconnects versus number of ports at D=2

Degree of Connectivity = 3

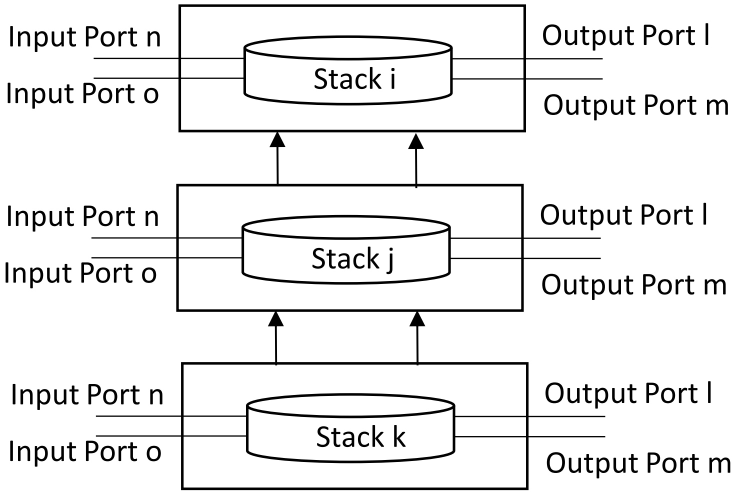


Figure 24: Multiple stacks optical fabric with degree of connectivity = 3

Figure 24 shows multiple stacks connected to each other with a degree of connectivity of 3. The external connections between the stacks increase the connectivity to a great extent. As mentioned, $D = 3$. Referring to the Equation (5),

$$C_{Stack} = [\frac{N}{S} + (D-1)]^2 * \text{S}.$$

After substituting the value of D, it becomes

$$C_{Stack} = [\frac{N}{S} + 2]^2 * S. \tag{7}$$

These are the number of interconnects in a stackable optical fabric. The number of interconnects in a flat fabric is N^2 as mentioned in Equation (4).

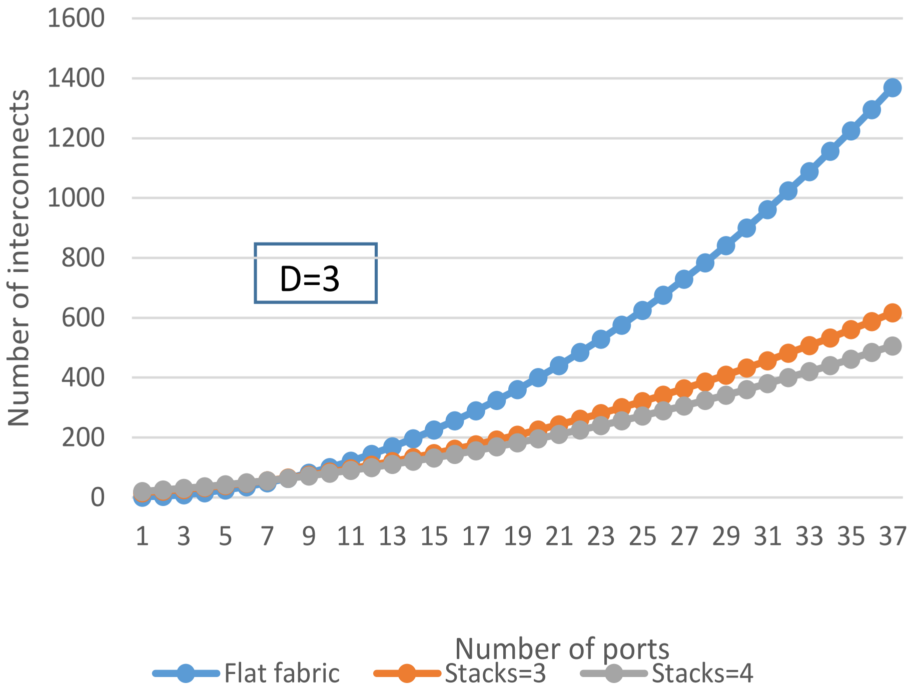


Figure 25: Number of interconnects versus number of ports at D=3

Figure 25 shows the plot between the numbers of interconnects and the number of ports when the degree of connectivity is 3.

5.1.3 Change in Number of Interconnects with Degree of Connectivity

It is observed that with the change in degree of connectivity, the number of interconnects increases. In Figure 26, it is shown that how the number of interconnects increases with the change in the degree of connectivity. As in Equation (5),

$$C_{Stack} = \left[\frac{N}{S} + (D-1)\right]^2 * \text{ S and}$$

$$\frac{\partial C_{Stack}}{\partial D} = 2D * S + 2N - 2S. \qquad (8)$$

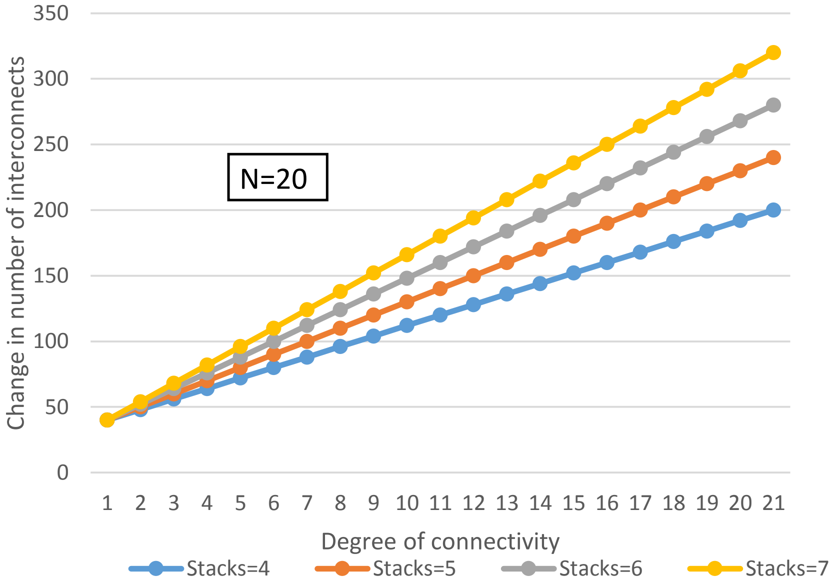


Figure 26: Change in number of interconnects with degree of connectivity at N=20

Equation (8) is a linear equation, which shows that the change in the number of interconnects changes linearly with the change in the degree of connectivity. The change is more as the number of stacks increases.

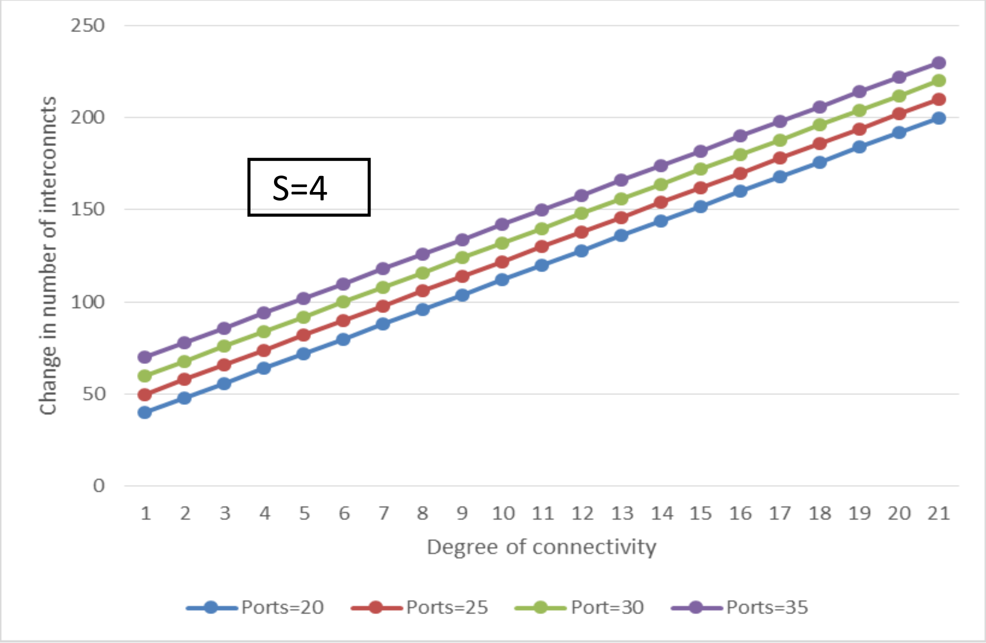


Figure 27: Change in number of interconnects with degree of connectivity at S=4

Figure 27 is a different plot using Equation (8) where the number of stacks is fixed and the number of ports is varying. The change in the number of interconnectivity increases as the number of ports increases.

5.1.4 Change in Number of Interconnects with Number of Stacks

Now the change in the number of interconnection with respect to the change in the number of stacks is evaluated. Referring to Equation (5),

$$C_{Stack} = [\frac{N}{S} + (D-1)]^2 * S \text{ and}$$

$$\frac{\partial C_{Stack}}{\partial S} = [1 + D^2 - 2*D - \frac{N^2}{S^2}]. \qquad (9)$$

Figure 28 shows the plot of the change in the number of interconnects with the change in the number of stacks. The plots are for different degrees of connectivity while the number of ports is fixed.

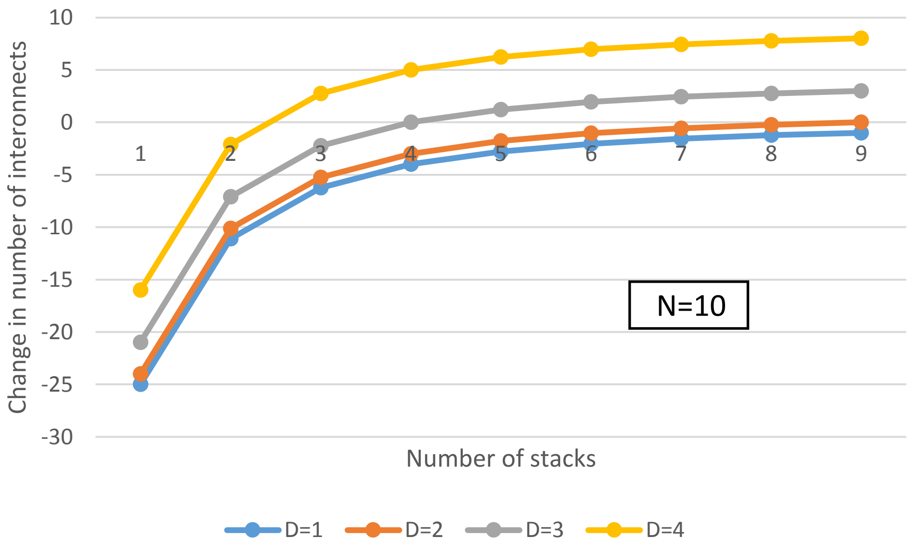


Figure 28: Change in number of interconnects with number of stacks at N=10

Figure 29 shows the plot of the change in the number of interconnects with the change in the number of stacks. These different curves are for different numbers of ports while the degree of connectivity is fixed. All the curves are merging near *x*-axis, which shows that after a certain point if the number of ports in a stack is increased, there would not be much change in the number of interconnects.

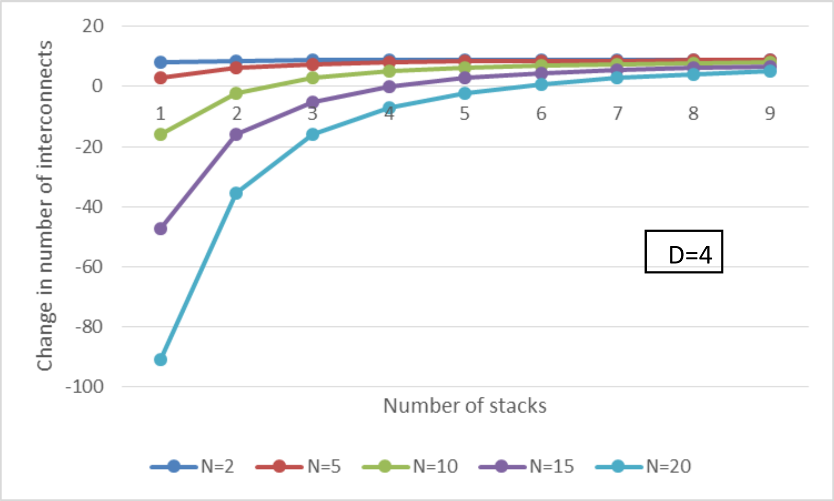


Figure 29: Change in number of interconnects with number of stacks at D=4

In all the above plots, it is observed that the number of interconnects is maximum in a flat fabric, which drops when flat fabric splits into small stacks of optical fabric. The number of interconnects increases again as the degree of connectivity increases, but they are still smaller than the number of interconnects in an equivalent flat fabric. The number of interconnects represents the cost of a network. Hence, the stackable option is a cost effective alternative. The user has a flexibility to balance between cost and performance.

With this proposed optical fabric option, one can always take advantage of the natural isolation provided between the stacks. If one wants to create a mapping application on a virtual network, they may want to keep it separate and not interacting with other applications. If there is a flat structure available, some other application running on the top of it may influence it. However, if one has the isolation provided by stackable optical fabric, work can be assigned on a few slices of the substrate of a virtual network with some degrees of connectivity.

It is seen that stackable optical fabric can reduce the cost while providing the flexibility in terms of providing connectivity supporting the slicing in virtual networks. This can provide a balance between cost and performance in terms of connectivity, and supporting the target application which will be mapped to a substrate. The substrate would be mapped to one or more stacks, but would never be mapped to the entire fabric. Providing unnecessary larger degrees of connectivity is a costly investment. To the other extreme, the flat structure means that every port can communicate to all other ports. In the virtual networking environment, it is never true because the target application needs to communicate to a subset of output ports. Connecting more output ports will increase the cost unnecessarily. The stackable optical fabric structure provides intrinsic isolation between unnecessary interactions.

5.2 Prototyping

The proposed scheduler design is implemented in Verilog-HDL (hardware description language). As mentioned in Section 4.3, the algorithm of the scheduler, there are three pre-stored databases needed to describe any stackable optical fabric. These are the destination index table, the service restriction table and the input-output connectivity table. In prototyping, these tables are defined in the memory to store the optical fabric connectivity.

Table 7 shows how output ports are connected to different destinations. Table 8, the service restriction table, shows how the service bit present in the BHC imposes restrictions on the output port’s connectivity and allows some specific output ports for connections.

Table 7: Destination Index Table for Prototype

Destination Address	$[P]_a^k$	$[P]_b^k$	$[P]_c^k$	$[P]_d^k$
X = 001100	0	0	1	1
Y = 001001	1	1	0	0

Table 8: Service Restriction Table for Prototype

Service bit	$[P]_a^i$	$[P]_b^k$	$[P]_c^l$	$[P]_d^k$
Q = 001	0	0	1	0
R = 011	1	1	1	0
S = 010	0	0	1	1

Table 9: Input-Output Connectivity Table for Prototype

Input ports	$[P]_a^i$	$[P]_b^i$	$[P]_m^j$	$[P]_b^k$	$[P]_d^k$	$[P]_c^l$
$[P]_m^i$	1	1	2	2,3	2,3	3,4
$[P]_n^i$	1	1	2	2,3	2,3	3,4
$[P]_a^j$	2	2	1	2	2	3

Table 9 shows how different stacks are connected to each other and the degree of connectivity between them. In simple words, it shows that for one input port, what the connectivity of all the output ports of the optical fabric is. In the prototype, these three tables can be pre-stored in Read-Only-Memory (ROM). The scheduler will refer to them when any BHC comes for scheduling.

The basic block diagram of the prototype is shown in Figure 30. It has a scheduler, a BHC processor, a display and an optical fabric. A BHC processor processes and sends BHCs to the scheduler. With the help of a First-In First-Out (FIFO) Memory, the scheduler reads these BHCs one by one. The BHC stored in BHC processor will have various elements such as Burst_offset, Burst_length, Burst_destination and burst_soft_bit defined for each BHC. The scheduler reads the BHCs until the FIFO gets empty. Another hardware component Random Access Memory (RAM) will be used to maintain the status of the output ports. It has two elements for each output port: port availability and port horizon. These elements will be updated by the scheduler block every time the BHC gets scheduled at any output port. The port availability signal is a single bit signal with values "0" or "1", while the port horizon is 32-bit digit representing the time the output port is scheduled.

The scheduler sends the "port availability" signal to the optical fabric block as shown in Figure 30. This signal will be used to create the proper configuration for optical components attached. The scheduler is also connected to 7-segment LED displays to show the status of BHC scheduling. One LED display can be used to show the status of BHC, whether scheduled or not. It can have values "FAIL" or "SUCCESSFUL". Other LED can be used to display the input port at which request is coming. Other LED can be used to display output port if the BHC is scheduled successfully. One more LED display can be used to display the horizon of output port scheduled.

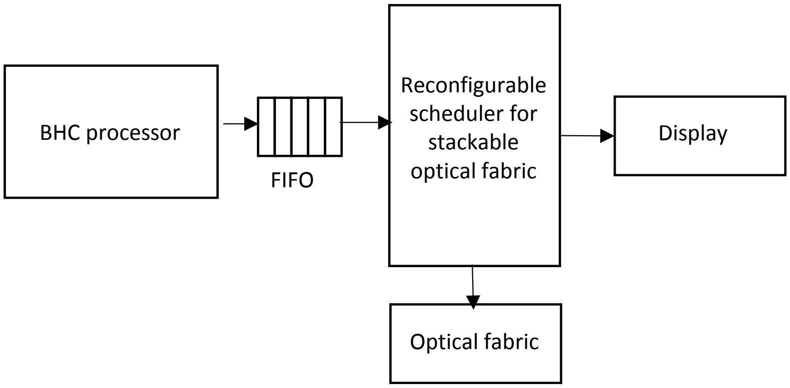


Figure 30: Block diagram of scheduler

The prototype is used to verify the proposed design. The scheduler receives the BHC request from the BHC processor through a FIFO. The result from the scheduler will be sent to the optical fabric to generate proper configuration signals to control the optical fabric. The scheduler processes BHCs until the FIFO gets empty. When the scheduler receives BHC, it will read its burst_destination and determines the list of allowed output ports. The scheduler will store this information in an allowed_ports_register. After this, the scheduler will check if the “service bit” is present in BHC or not. If the “service bit” is present, the scheduler will refer to the service restriction table. After reading this table, the scheduler will modify its allowed_ports_register. Now the scheduler will refer to the input-output connectivity table. The table shows how the input port at which the burst will come is connected to all other output ports. The scheduler will refer to its allowed_ports_register and will check for each output port’s connectivity to the input port. For scheduling a burst request, the scheduler will check for the output port’s availability at “current time + Burst_offset time” and pick the output port with the minimum degree of connectivity. The output port’s availability will be checked from the “port availability” bit for each output port. A “0” shows that the port is available and a “1” shows that the port is occupied. The other element, (32-bit) port horizon, shows for how long the port is occupied.

After selecting one output port, the scheduler will update its “port availability” bit. The scheduler will update the availability bit from “0” to “1” from the “Burst_offset time” to “Burst_offset + Burst_length” time. It will also update the port horizon to “Burst_offset + Burst_length” time. The scheduler will then display “SUCCESSFUL” on display, if the scheduling is successful. If scheduling is unsuccessful, it will display “FAIL” as the status. It will also display the input port in both the cases. It will display the output port only if the scheduling is successful.

In the prototype, an 8-channel scheduler is implemented in the hardware using Verilog hardware description language. Figure 30 shows the block diagram of the hardware implemented. The BHC processor used is a simplified BHC processor, because the focus of the prototype was on the design of a scheduler. The BHC contains all the necessary information needed. The scheduler pre-stores the information about the input-output port’s connectivity. The scheduler also has the information about the availability of output ports.

The scheduler reads a new BHC from BHC processor as long as FIFO is not empty. It checks for Burst_destination in the BHC. It checks from the output port availability database for the availability of the destination. If the output port requested is available, the scheduler schedules that particular output port for the Burst_length time. Then it displays the input port, output port and scheduler status as “SU” for successful action or “FA” for an unsuccessful action.

The scheduler reads incoming BHCs from the BHC processor one by one. For some requests, it could schedule the burst if it finds the suitable output port. For some requests, it could not find the suitable output port and hence, could not schedule the burst. The Figure 31 shows the simulation waveforms for a successful scheduling of the incoming burst and Figure 32 shows an unsuccessful or failed scheduler action due to unavailability of requested output port.

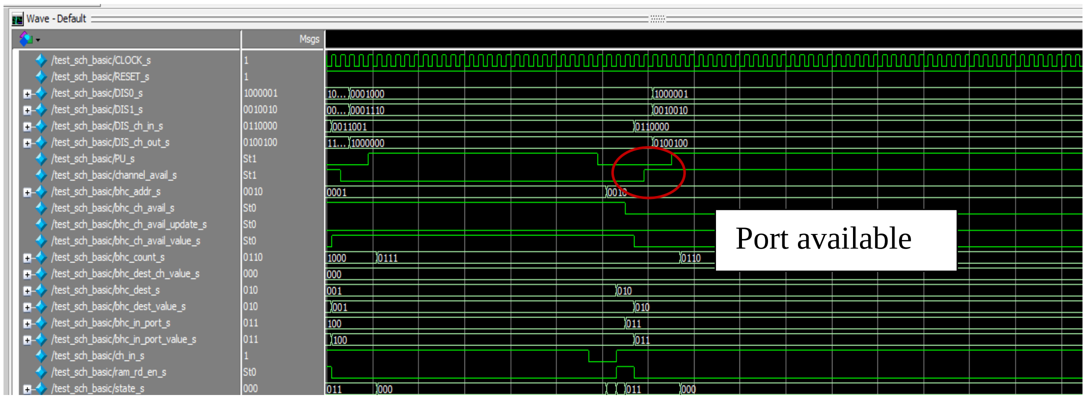


Figure 31: Successful scheduling

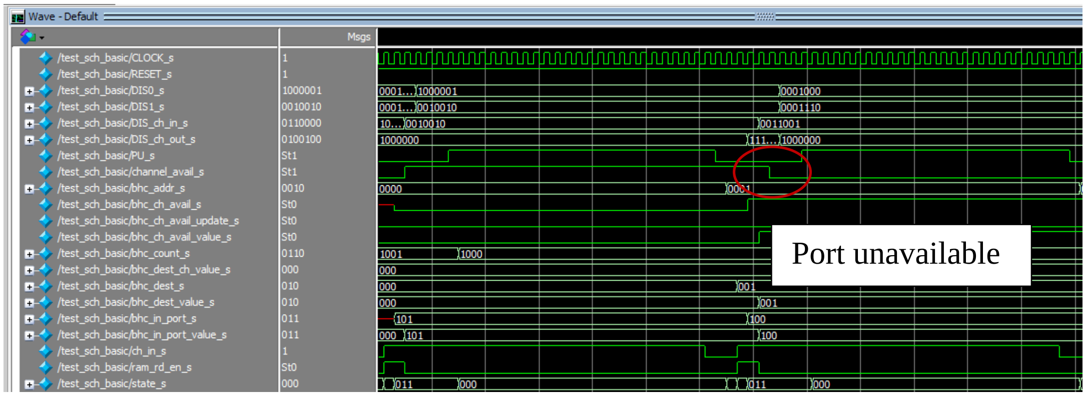


Figure 32: Unsuccessful scheduling

The circuit has been synthesized using Quartus II on Altera Cyclone II EP2C35F672C6 field programmable gate array (FPGA). The design is downloaded on FPGA and tested with a sequence of input requests. The clock frequency taken was 50 MHz. For each BHC, the result matched with the expected results.

The basic scheduler is designed and verified on Altera FPGA board. The behavior of scheduler designed was observed as expected. All types of input cases were used in the form of BHCs. The scheduler response to all the BHCs were exactly as expected. For some BHCs scheduler could schedule the burst and for some BHCs it could not. For those it could successfully schedule, it reported the input port, the output port and the final status as “successful”. For those it could not schedule, it reported the input port and the final status as “failed”.

In the above prototype, the destination index table and soft service were not implemented completely. The scheduler design can be easily extended to include the complete destination index table and the soft service. Referring to Figure 18, the flow chart of the stackable scheduler, the basic scheduler maintains a list of allowed ports. It modifies this list when it refers to the destination index table and the service restriction table. The basic scheduling algorithm and the selection of ports comes later in the flow when it finalizes its list of allowed output ports. The concept of destination index table and soft service can easily be added to the prototype. With the inclusion of the destination index table and the soft service, only the list of allowed ports will be modified. The rest of the algorithm will remain the same.

From the results obtained in the prototype design, it can be concluded that the stackable optical structure and its scheduler work as expected. It covers all the specifications of a reconfigurable scheduler for stackable optical fabric described in Section 3.3. The benefits of stackable optical fabric are already described in Chapter 3, which is mainly the ease of scalability and cost effectiveness. There was no existing scheduler to support a stackable optical fabric. The proposed stackable scheduler works exactly in line with the requirements and the prototype design confirms the feasibility of such a design. Apart from the scheduling action for the stackable optical fabric, the scheduler also provides the soft service to the vendors so that they can restrict some output ports from being connected. This is an additional feature of the proposed scheduler in order to provide some privacy for the data transmission. Also the cost model discussed at the start of this chapter demonstrates that the stackable structure offers a flexible structure at lower cost. As discussed earlier, the stacks can also offer significant privacy to the applications on virtual networks and software defined networks.

As demonstrated in the cost model and the prototype design, the proposed stackable optical structure, if implemented, would provide a low cost, scalable and flexible solution to the future Internet infrastructure.

Chapter 6 Conclusion

There is rarely any field in everyday life which is unaffected by the use of Internet. The use of Internet for data as well as for applications are sure to expand, which leads to the demand of a robust and expandable Internet infrastructure.

This thesis has proposed a stackable arrangement of optical fabric. The explicit benefit of this approach is expandability at lower cost. The other benefit of this approach is the freedom of isolation. In many applications of virtual networks, it may be desired to keep the application separate from other applications. It is possible with stackable optical fabric. One can assign work on a few slices of the substrate of a virtual network to some stacks of optical fabric. In this way, physical isolation can be obtained with relatively low cost.

The stackable arrangement of optical fabric provides flexibility in terms of connectivity of ports. It is observed that with stackable fabric, the number of interconnects is limited. Due to physical isolation between the stacks, there is restricted connectivity between input and output ports. Connectivity can be further improved by adding external connections between stacks.

The scheduler, which can work with the stackable optical fabric should have some special characteristics. In Section 3.3, the specifications for such a scheduler are defined. It is observed that no existing scheduler can work with stackable optical fabric because in stackable structure, not all the input ports are connected to all the output ports as in flat fabric. The scheduler should know before scheduling, that the input port at which burst is coming belongs to which stack and what the output ports of that stack of optical fabric are. If the external connections were made between the stacks in order to improve the connectivity, the scheduler should know about it in detail. Further, in Section 3.2.4, the concept of soft service has been added. This is an option for vendors to use by which they can prevent some output ports from being connected. They can use the benefit of the soft service in order to connect some high priority applications on some highly secure output ports.

Section 4.1 defines the general concept of a stackable scheduler. Later in Section 4.3, the algorithm of such a scheduler has been discussed. In the algorithm, there is a core scheduling block which takes BHCs at the input ports and it has a list of allowed output ports. It checks all the output ports for its availability and schedules the input request as soon as it gets a first output port available. The list of allowed output ports is a golden list of reference. This list is made referring three elements: the destination defined in BHC, the soft request if present in BHC, and the physical connectivity of the input output ports. Once the scheduler schedules the input request on an output port, it occupies that output port for a fixed time defined by burst offset and burst length in BHC.

Based on the algorithm defined in Section 4.3, a prototype has been designed in Verilog hardware description language. The design has been tested in the Modelsim simulator, and has been downloaded on Altera FPGA. It has been tested on the FGPA hardware board. The behavior of the scheduler was as expected. The successful implementation of prototype confirms the feasibility of such a scheduler design.

The cost model of the stackable fabric has been studied in Section 5.1. The number of interconnects has been considered as the parameter of cost. The flat fabric has the maximum number of interconnects, hence, it is the most expensive one. The equivalent stackable optical fabric with no external connections has the minimum number of interconnects, hence, is the cheapest one. The stackable optical fabric with external connections lies in between. The user can use a trade-off between cost and performance and choose a suitable structure for the optical fabric.

The proposed stackable optical fabric offers a flexible solution for different Internet applications. It offers the privacy and the ports reservations with the added advantage of low cost. This is extremely useful for applications on virtual networks where privacy is offered by physical isolation of stacks. The flexibility in connectivity provides a wide range of options based on the load on optical components. The proposed stackable optical fabric can play a major role in Internet infrastructure.

References

[1] Y. Chen, J. S. Turner and P.-F. Mo, "Optimal Burst Scheduling in Optical Burst Switching Network," *Journal of Lightwave Technology,* vol. 25, no. 8, pp. 1-12, 2007.

[2] Y. Chen, C. Qiao and X.Yu, "Optical Burst Switching: A New Area in Optical Networking Research," *IEEE Network 18,* pp. 1-13, 2004.

[3] D. Blumenthal, P. Prucnal and J. Sauer, "Photonic Packet Switches: Architectures and Experimental Implementation," *Proc. IEEE,* vol. 82, no. 11.

[4] K. Barker, A. Benner, R. Hoare, A. Hoisie, A. Jones, D. Kerbyson, D. Li, R. Melhem, R. Rajamony, E. Schenfeld, S. Shao, C. Stunkel and P. Walker, "On the Feasibility of Optical Circuit Switching for High Performance Computing Systems," in *Proc. 2005 ACM/IEEE Conference on Supercomputing*, Seattle, 2005.

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

[6] C. Q. a. M. Yoo, "Optical Burst Switching (OBS) - A new paradigm for an optical Internet," *Journal of High Speed Networks,* vol. 8, no. 1, p. 69–84, 1999.

[7] 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, 2010, pp. 265-290.

[8] D. K. Hunter, M. C. Chia and I. Andonovic, "Buffering in optical packet switches," *Journal of Lightwave Technology,* vol. 16, no. 12, p. 2081–2094, 1998.

[9] L. Tamil, F. Masetti, T. McDermott, G. Castanon, A. Ge and L. Tancevski, "Optical IP routers: Design and Performance Issues Under Self-Similar Traffic," *Journal of High Speed Networks,* vol. 8, no. 1, p. 59–67, 1999.

[10] B. Wen and K. M. Sivalingam, "Routing, Wavelength and Time-Slot Assignment in Time Division Multiplexed Wavelength-Routed Optical WDM Networks," *in Proc. INFOCOM,* p. 1442–1450, 2002.

[11] S. Yao, S. Dixit and B. Mukherjee, "Advances in Photonic Packet Switching: An overview," *IEEE Communication Magazine,* vol. 38, no. 2, p. 84–94, 2000.

[12] W. D. Zhong and R. S. Tucker, "Wavelength Routing-Based Photonic Packet Buffers and Their Applications in Photonic Packet Switching Systems," *Journal of Lightwave Technology,* vol. 16, no. 10, p. 1737–1745, 1998.

[13] J. S. Turner and Y. Chen, "WDM Burst Switching for Petabit Capacity Routers," *in Proc. IEEE Mil. Communication Conference,* p. 968–973, 1999.

[14] J. Y. Wei and R. McFarland, "Just-in-time Signaling for WDM Optical Burst Switching Networks," *Journal of Lightwave Technology,* vol. 18, no. 12, p. 2019–2037, 2000.

[15] Y. Xiong, M. Vandenhoute and H. C. Cankaya, "Control Architecture in Optical Burst-Switched WDM Networks," *IEEE Journal on Selected Areas in Communications,* vol. 18, no. 10, pp. 1838-1851, 2000.

[16] M. Yoo and C. Qiao, "Just-enough-time (JET): A high Speed Protocol for Bursty Traffic in Optical Networks," in *Proceeding of IEEE/LEOS Conf on Technologies For a Global Information Infrastructure*, New York, 1997.

[17] C. Qiao and M. Yoo, "Optical Burst Switching (OBS)–A New Paradigm for an Optical Internet," *Journal of High Speed Networks,* vol. 8, no. 1, p. 69–84, 1999.

[18] D. Sarrazin, H. Jordan and V. Heuring, "Fiber Optic Delay Line Memory," *Aplied Optics,* vol. 29, pp. 627-637, 1990.

[19] X. Yu, Y. Chen and C. Qiao, "Study of Traffic Statistics of Assembled Burst Traffic in Optical Burst Switched Networks," *Proceeding of Opticomm,* p. 149–159, 2002.

[20] A. Ge, F. Callegati and L. Tamil, "On Optical Burst Switching and Self-Similar Traffic," *IEEE Communications Letters*, vol. 4, p. 98–100, 2000.

[21] J. Xu, C. Qiao and G. X. J. Li, "Efficient channel scheduling in optical burst switched networks," *in Proc. IEEE INFOCOM*, vol. 3, p. 2268–2278, 2003.

[22] J. Xu, C. Qiao, J. Li and G. Xu, "Efficient Burst Scheduling Algorithms in Optical Burst-Switched Networks Using Geometric Techniques," *IEEE Journal in Selected Areas in Communications*, vol. 22, no. 9, p. 1796–1881, 2004.

[23] V. M. Vokkarane and J. P. Jue, "Segmentation-Based Nonpreemptive Channel Scheduling Algorithms for Optical Burst-Switched Networks," *Journal of Lightwave Technologies*, vol. 23, no. 10, p. 3125–3137, 2005.

[24] J. Teng and G. N. Rouskas, "Wavelength Selection in OBS Networks Using Traffic Engineering and Priority-Based Concepts," *IEEE Journal of Selected Areas of Communications*, vol. 23, no. 8, p. 1658–1669, 2005.

[25] Y. X. M. Vandenhoute and H. Cankaya, "Design and Analysis of Optical Burst-Switched Networks," in *Proc. SPIE'99 Conference. All Optical Networking: Architecture, Control, Management Issues*, Boston, 1999.

[26] L. Tancevski, A. Ge, G. Castanon and L. Tamil, "A new Scheduling Algorithm for Asynchronous, Variable Length IP Traffic Incorporating Void Filling," in *Proc. OFC'99*, San Diego, 1999.

[27] Cisco Systems, *Introduction to DWDM Technology*, 2001.

[28] R. Ramaswamy, "Optical Networking Technologies: What Worked and What Didn't," *IEEE Communications Magazine*, pp. 132-139, 2006.

[29] G. I. Papadimitriou, C. Papazoglou and A. Pomportsis, "Optical Switching: Switch Fabrics, Techniques, and Architectures," *Journal of Lightwave Technology*, vol. 21, no. 2, pp. 384-405, 2003.