8-2017

Study of comparison of OCS and Hybrid switching in FSO data centers

Suraj Yadav
University of Nebraska-Lincoln, ysuraj76@yahoo.com

STUDY OF COMPARISON OF OCS AND HYBRID SWITCHING ON FSO DATA CENTERS

by

Suraj Yadav

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Master of Science

Major: Computer Science

Under the Supervision of Professor Jitender Deogun

Lincoln, Nebraska

August, 2017

STUDY OF COMPARISON OF OCS AND HYBRID SWITCHING ON FSO DATA CENTERS

Suraj Yadav, M.S

University of Nebraska, 2017

Advisor: Jitender Deogun

With the increase in big data applications, it has become the need of the hour to handle data efficiently to handle the growing traffic in the data centers. The popular mechanism is parallel processing using commodity hardware hence it is becoming an interesting research topic to explore new architectures which have high performance, but with increase in data sizes the architecture has to expand which increases the cabling complexity. Hence the study of Free Space Optical (FSO) communication for the data centers is gaining more importance now than ever. We proposed square OWCELL topology which is a new free space optical (FSO) architecture and was accepted in IEEE-ICC [7]. With a view of competing with existing Fat-tree architecture, we are also exploring the usage of hybrid switching over the OCS switching. We also modify a flow level simulator which helps reduce the complexity of simulation by considering study of flows instead of packets. In this tool, we calculate the parameters of flows accepted, flows rejected, flows for which path cannot be found, total flows and their percentages. The Hybrid performs at par or better most of the times as compared to OCS switching by about 3-8% on variation of buffer sizes, number of servers in the network and the congestion.

ACKNOWLEDGEMENTS

I am using this opportunity to express my gratitude to everyone who supported me throughout my Master's program. I am thankful for their aspiring guidance, invaluably constructive criticism and friendly advice. Firstly, I am extremely grateful to my advisor Dr. Jitender Deogun for his constant support during my M.S study and research. His motivation and immense knowledge in the subject helped me all the time during my research.

I would like to thank the rest of my thesis committee of Dr. Ying Lu and Dr. Lisong Xu for their precious time, thoughtful and critical reading of this thesis. I would also like to thank Abdelbaset Hamza and am gratefully indebted to him for helping me all throughout my research and his valuable comments on this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

With the increase in the data exponentially and the cloud services, there is a need to have powerful data centers with high processing throughput and less delay. This has given rise to research into finding efficient data center architectures.

There are two choices for building large scale clusters

1. Use of specialized clusters and hardware with specialized communication protocols so that they can be scaled to thousand nodes for high bandwidth, but they do not use commodity servers

2. This method involves the use of commodity servers and switches. But this cannot be scaled effectively and achieving higher bandwidth gets expensive.

The second method is more popular because it is more cost effective,[16] but this involves the study of various link technologies, data center architectures to achieve parallel processing to improve the performance, power efficiency and most importantly cost effectiveness. Hence it is not unusual to hear cluster sizes of 1000 nodes for universities, research labs & commercial organizations for storage, analysis and net-

working. So, for this reason we try to find a solution using the fastest link technology (free space optics), newly improved architectures and improved switching technology (hybrid switching)

1.2 What is a data center?

A data center is place to interconnect computing facilities like servers, routers, switches and firewall along with other supplemental components for backup, power and air conditionining for efficient processing of huge amount of data.[5]

1.3 Evolution of data Centers:

The data center in early days was referred to as huge computer rooms since they were difficult to maintain and operate and required a unique environment. As the data increased the number of servers required increased, strategies such as racks were devised to organize the servers. Also since the servers increased the energy required increased and hence the cooling. However during the 1980's the evolution of microcomputers took place and computers were started to be deployed everywhere. With the UNIX operating systems in 1970's and Linux compatible PC's the number of users increased, and client-server model was heavily realized for sharing resources. Due to this advancement in technologies networking equipment were available readily and with standardization in the cabling technology the concept of organizing servers in one room became popular and so did the term 'data center'. However, the actual boom came in 1997-2000 when the internet revolution took place with small companies along with the larger ones requiring uninterrupted connectivity and storage.

1.4 Data Center Traffic Characteristics

With the increase in data there is greater interest in the data center research. However, the research interest is mainly on the architecture, minimizing the links the energy consumption etc. Although, it is important to analyze the data center traffic as well to improve the performance. Following are the characteristics of the data center traffics:[2]

1. Most of the flows in the data centers are small in size(less than 10KB) and most of them have duration in the range of a few milliseconds and the number of active flows every second in a data center is around 10000.

2. The traffic in a data center would be huge, but it is intermittent i.e. the traffic is ON/OFF in nature.

3. In the cloud data centers, the majority traffic is intra-rack (80%) while its opposite in the case of university data centers where the majority traffic is inter rack (40-90%)

4. Also, the link utilization in a data center is different on different days of the week and also are different for core, aggregate and edge switches.

1.5 Data Center Design:

The design principles that should be considered when designing a data center are[17]

Incremental Scalability:

Practically if a need arises to increase the number of servers the network designers prefer to add only a few components instead so the architecture must be able to support that.

Cabling Complexity:

When the size of the network is small the number of connections is quite low and hence cabling complexity is not an issue but as the size of network increases and is huge in the case of data centers the cabling gives rise to connection, maintenance and other problems like cooling.

Bisection bandwidth:

This is a measure of the worst case capacity of the network. To study bisection bandwidth the network is cut into two parts, and then the reachability of the destination is studied. Bisection bandwidth is the bandwidth available between these two bisections.

Throughput:

It is a measure of the amount of data that has successfully transmitted. The higher the throughput, the more efficient the network.

Over subscription:

It is defined as the ratio of the ingress bandwidth of the switch to the egress bandwidth. The use of oversubscribed network improves the link efficiency in the data center, and in-turn reduces the cost.

Fault Tolerance:

This is the measure of the effect on the data center when there is a hardware failure. Since hardware failures are common, it is considered an important parameter.

Energy Consumption:

Since the electrical devices require power to operate and with the increase in size of data centers, it is essential now to consider the energy efficiency of the data center. This can be achieved by the use of low power CPUs and routing devices. Software modes by visualizing the power consumption and cooling can also be used.

Costs:

Cost is a major factor. Lower costs can be achieved by using commodity servers and routers in parallel instead of servers with huge processing power and custom routing devices.

Reliability:

The data center must be highly reliable i.e. it must be efficient always because if the performance degrades even for a little duration results in losses.

Security:

Security is essential as the data exchanged between different servers must be isolated and must not be accessible to unauthorized services.

Latency:

Latency is the delay required for the data to be transferred completely from the source node to the target.

Chapter 2

Data center classification

2.1 Based on topologies

2.1.1 Hierarchical topology

As the name suggests, it is called hierarchical because the layers of switches are arranged in levels. At the lowest level are the servers which are connected to ToR's (Top of Racks) switches. The speed of these links is in the range of 1-10Gbps. Aggregate switches are used to interconnect ToR's and their data rates are in the range of 40-100 Gbps. Core switches interconnect the aggregate switches, and the data rate is higher than the aggregate switches but around the same range of 40-100 Gbps.[15] Apart from these, certain links are added additionally in some architectures to minimize the hops to the destination.

2.1.2 Torus topology

In this topology, there are no levels as in the case of hierarchical topology. Instead, the switches are interconnected with each other to form a ring. Torus topology is

not scalable and is preferred in configurations where traffic in the network is less and hence less number of aggregate links is required. Larger networks using torus topology have less connectivity and high delay.[15]

2.2 Based on the switching technique

Based on the switching the data centers are broadly classified into switching and direct data centers.

2.2.1 Switch-based Data centers

A switch based data center consists of multi-level tree of switches which are connected to the end servers. They are popular choice for data center due to low complexity. They can support about 10000 servers. Consider a three level data center, the leaf switches are called as ToR (Top of rack) switches. They are 1Gbps Ethernet ports responsible for transferring data to the servers within the rack. The layer-2 switches are 10 Gbps links which interconnect the ToR switches and these layer 2 switches also called Aggregate switches are interconnected by more powerful switches above at layer 3 known as core switches. However high-end switches generally are costly and hence are replaced by high bandwidth links. The classical example of this is type of architecture is the Fat-tree architecture which is explained in fig 3.1.

The networks employing this architecture are also called as indirect networks.

2.2.2 Direct Data Centers

In this type of data center, the server connects to other servers directly without any routers. Hence servers perform both the processing of the packets as well as routing.

They are used to provide better scalability, fault tolerance and high-end networking capabilities. DCell is an example which employs this technique and is explained in fig. 3.2.

2.3 Based on inter-rack links

With optical fibers having high bandwidth and data-rate properties and more recently the free space optics which offer additional characteristic of wireless connectivity there will come a time in future when electrical links would no longer be used in data center inter-rack communication.

2.3.1 Electrical

As the name suggests high bandwidth electrical cables are used for inter-rack connections, but with increase in traffic in data centers the need the optical data center arised. However electrical links have the advantage of being economical.

2.3.2 Optical

Here the light source is used as the medium of transmission instead of the electric current for data transfer. It can be further classified as wired or wireless which is same as optical fibers and free space optics.

2.3.2.1 Optical Fibers

In this optical fibers are used as the transmission medium due to their high data rates and bandwidth.

2.3.2.2 Free Space optics

With increase in data center architectures, the number of links has increased which reduces the scalability of the network. Free space optical data centers involve the use of wireless light emitting sources which give all the benefits of optical fibers with capability of wireless connectivity but significantly less delay.

Chapter 3

Background

Here we will focus our discussion on existing data center technologies, the type of flow switching technologies and the process of optical switching and the components.

3.1 Optical switching components

An optical switch consists of the following logical components. The first step and last steps are multiplexing and de-multiplexing (WDM) to enable optical fibers to carry multiple data stream. This increases the bandwidth of the links. Currently, each data fiber can support up-to 80 wavelengths with 50GHz spacing each till the speeds of 40Gbps. The brief process is that firstly the data from the incoming fiber is collected and switches to the output fiber on a different wavelength. This is the chief function of the switching fabric along with the buffer which is used to store the data before switching. The switching fabric is all optical, and hence it is referred as all-optical switching. The detailed description of all the components are given below:

3.1.1 Input-Output interface

The main component of the input-output interface is the transceiver. It is a device that inter-converts the data between optical and electrical signal. To be more specific the receiver converts the optical signal to electronic data while the transmitter codes the electronic data to optical signals. The transmitter consists of mainly the light source and the modulator with laser or LED as the main source. On the other hand, the receiver consists of a photo detector and an amplifier. The job of the detector is to detect the optical signals and convert them to electrical signals. Amplification is done only if the signals are weak. The operation of these devices requires power. For example, the transmitter consumes more power if the light source is laser rather than LED also for a transmitter the data rate and encoding schemes used also have a huge impact. A higher data rate and complex encoding scheme consumes more energy. Cost is also an important factor, the cost of the transceiver depends on the tuning ability, wavelength support and distance. The larger the distance support, the tuning ability rather than a fixed wavelength, makes the transceiver more costly. However, a transceiver is required only if there is conversion required from electrical to optical domain i.e. the architecture is a hybrid architecture. However, if the architecture is all optical the there is no need of the transceiver, and Tunable wavelength converter (TWC) is used.[15]

3.1.2 Switch Fabric

SwF is used to switch packets from the input to the output port. The main devices which comprise a switch fabric are as follows:

1. Splitter - It forwards the optical signal to every output port.
2. Micro-electro-mechanical-systems (MEMS) - These are the devices that are

used in guiding the light by the use of mirrors which are controlled in the order of milliseconds.

3. Arrayed waveguide grating router (AWGR) - It changes both the direction and the wavelength of the signal.

4. Optical contention resolution with electronic buffers (OCE) is similar to SwF and is used to resolve the contention in case of signals in AWGR.

5. Wavelength selective switch (WSS) – It is used to combine data from different wavelengths. It consists of 3 ports the input port, drop port and the express port. If selection of 1 wavelength is required from the multiple input wavelengths then that wave is passed from the input port to the express port or else the rest of the wavelengths are passed to the drop port.[15]

3.1.3 Port Buffers

The main function of the buffers is to store the packets if the links are busy. There are 2 types of port buffers, input port buffers and output port buffers. Input port buffers hold the traffic till the SwF is ready to transmit the packet while the output port buffers are used when the output links are busy. Ideally, there is no waiting time for the packets i.e. they are not stored in the input buffers if the incoming data is periodic. However, in optical fibers the light cannot be stored. Hence the concept of delay lines is introduced so that the SwF has enough time to process the traffic. The propagation delay that is introduced is in the order of 200ns for a 40m optical fiber. To summarize there are four different things possible when it comes to port buffers

1. Large electronic port buffers, 2. Small port buffers, 3. No port buffers and 4. Optical delay lines.

While the concept of buffers is used in electrical data centers, delay lines are used

in optical data centers. Concerning power consumption the more the buffers the more power it takes. Header lookup is the basic and most important step when it comes to routing. There are different types of header lookup based on the data center technology used.

Packet Lookup

This is the oldest and the most popular of all the lookup. In this, the packet is decapsulated at every IP layer and the header is read. At the network layer, the header details of the destination are read which is then used to route the packet to the destination. To lookup the destination the routing tables are needed to be updated periodically and hence the routing protocols like RIP & OSPF are used to broadcast updates periodically to which then the routing tables are looked up whenever there is a packet in the switch. However, the process of lookup introduces delay because the destination IP's are 32 bit long addresses and even the longest prefix match takes a particular bit of time. Tertiary content addressable memory is now used to reduce the latency caused during the lookup however it requires more memory and processing.[15]

Label Lookup

To reduce the latency in packet lookup, label lookup was proposed and it has been in use in the protocols such as Ethernet, frame relay and asynchronous transfer mode (ATM) in multi-protocol label switching (MPLS) networks. In this, the labels are first unpacked in by the edge switches and then the MPLS labels are created at the edge router. The interior switches extract the labels which are used for looking up the hash tables and is O(1) complexity process that gives the output port or the combination of output port and label modification. The lookup delay is less in label

switching, but the generation of labels operation requires considerable memory as compared to packet switching networks and introduces a certain setup delay in the network.However the lookup is very efficient and the latency is quite low. [15]

Open Flow lookup

This is the most flexible lookup among all since each service provider can have its own set of protocols to route the flow. It uses a centralized controller which computes the route for the flows and then adds the route to the routing tables of the switches. So when a flow arrives at a switch, the routing table is used and the flow is routed. The lookup is similar to that in MPLS networks however the network designer has an added advantage of deciding the lookup fields which is not the case of label switching but this increases the complexity of the switch. The latency of the open flow lookup is comparable is to that of label lookup but it requires more memory and processing

Optical packet switching (OPS) lookup

This is a technology for the hybrid data centers and used in the LIONS switch. In this the switch has to look-up the destination in a very small duration of time. This duration is typically 190ns or else the packet is dropped and it has to be re-transmitted. This is similar to label processing except the fact that the processing required is fast and the delay is less. However, if there is a need to increase the lookup time then electronic buffers can be used but it has higher energy consumption. The example of this can be Petabit switches.

Wavelength Lookup

This is used in all optical architectures. In this there is a mapping of the input wavelength and the output port. Thus when a signal arrives at a specific wavelength

and port at the input, it can be mapped to the output port and the switching is all optical. The switching is coded while designing the system and hence there is no need of an actual lookup and hence no memory and power are required for its operation.

Finally to summarize the lookup mechanisms in for the parameters of memory, processing, set up delay and latency the table 3.1 is useful.

Table 3.1: Summary of optical parameters for different lookup types

Lookup	**Memory**	**Processing**	**Latency**	**Setup**
Packet	Medium	High	High	Not Required
Label	High	Medium	Medium	Required
OpenFlow	High	Medium	Medium	Required
OPS	High	High	Low	Not Required
Wavelength	Low	Low	Low	Required

3.2 Optical Switching Types

The analysis of a type of traffic that an application generates can be classified as bounded/small or the other type in which there is bulk data transfer between same source and destination for a long period. There are various optical switching technologies which are best suited for one type of applications but not other. They can be classified as:

3.2.1 Optical packet switching (OPS)

With the increase in the internet traffic the average packets in the network were on the rise and there was a need to reduce the delay in the transmission of the packets. Hence the use of the optical fibers increased, but for that the packets had to be converted to

optical domain which required electrical-optical interconnects which inturn required high energy for the operation. Hence the mechanism to directly switching in the optical domain was required. This is called as optical packet switching (OPS).

The main components of an OPS network are multiplexer/de-multiplexer, input interface, output interface, switching fabric, control panel.[3] The input buffer amplifies the signal to restore the quality and then the header is extracted. The control unit then processes the header. The switch fabric generates the wavelength for the data to be transferred at the output port by referring the lookup tables which are constantly updated by the network management system. The packet is passed through the buffer till the new header is found and is programmed to delay the data approximately by the same amount. Finally, the output interface attaches the new header generated in the control panel to the payload which is switched by the switching fabric.[3] Hence for each packet there is a header lookup and hence this type of switching is best suited for applications that generate low traffic.

3.2.2 Optical Circuit Switching (OCS)

Packet switching is used in interconnecting different processing units but with the increase in the size of the networks and increase in the data, these networks were not very scalable. The latency and bandwidth become a constraint for such systems, which is addressed by optical circuit switching networks. As the name suggests the switching is in the optical domain and a circuit is established before the data transfer. Some of the components are similar to those used in wide area networks and SONET. The switches are completely optical and the technology used is micro-electro-mechanical Systems (MEMS) which use arrays of mirrors which move the light from input to output. Since the path is established at the start, the latency of

finding route every time is eliminated and the data can be transferred at complete bandwidth. Also, the speed of light is very high as compared to the speed of electric current and hence physical distance or the link length is also not a major factor when using OCS.[14]

Optical circuit switching is cost effective and energy efficient as compared to the packet switching devices. This is because there no optical to electrical and vice-versa conversions hence the expensive optical trans-receivers are not required. However, one of the drawbacks of OCS is higher connection delay as compared to packets switching networks. Hence it is viable to use OCS only when the transfer of data is for large duration between the same source and destination. One of the early example is the NEC Earth Simulator which is made of 640*640 ports

3.2.3 Optical Burst Switching (OBS)

The problem of using OCS with smaller flows is that the flow sizes are too small and in many cases the overhead and the time required for connection results in degradation of the performance. OBS is a method of aggregating multiple smaller flows into burst and then transmitting it to the destination without buffering at intermediate switches. It combines the advantages of both WDM and packet switching. In OBS only the control packets are processed over each of the switches and also the route is fixed by the control packets but the source does not wait for the acknowledgment of the path being available which reduces the delay to half of the original. The protocol used is JET(Just Enough Time) protocol.[4] A source sends a control packet, which is followed by the burst after a particular time known as the base offset time. JET protocol uses efficient bandwidth utilization by using delayed reservation i.e. the bandwidth of the link is reserved from the time the burst is expected to arrive instead

from the time from which the control packet is reserved and it is reserved till time 'l' where l is the duration of the burst. However, if the bandwidth is not available the burst is dropped and then has to be re-transmitted by the source.

There are two techniques if the one of the links is busy and cannot carry the data then data is dropped and in the second case the if one of the links in the path is busy then the traffic is made to wait and the link status is checked again and data is transmitted only when the links get free.This improves the performance since the amount of traffic dropped will be less.

3.2.4 Hybrid Switching

Hybrid switching is the combination of OCS and OBS switching. we see that OCS works better when the flows are large while OBS is efficient with smaller flows, but the traffic in a network in practical situations may not always be either large or small, instead it is generally a combination of both. Hence to route both these flows efficiently we categorize the flows in the network into small and large flows and then use OCS for large flows and OBS for smaller flows to achieve a greater efficiency in routing as compared to both individually.

3.3 Existing data center topologies

3.3.1 Fat-tree

Fat-tree is the most popular data center architecture which is widely implemented across commercial and educational institutions.

Like the real trees, the architecture of fat-tree gets thicker as we go away from the leaves. It resembles a tree of meshes of a graph. Since the fat-tree gets thicker it

means there is addition of links and in turn bandwidth. Fat-tree is considered as the best architecture in case of the hardware required.[12]

Construction: For the construction of a k-ary Fat-tree, there are k pods each containing two layers of k/2 switches. Lower k/2 ports are connected to the switches and the upper k/2 ports are connected to the aggregation switches. The number of core switches is k*k/4 and each of the core switches is connected to each of the pods. In General, the number of servers in the Fat-tree is given by k*k*k/4.

Fig. 3.1 Shows the construction of a fat-tree with k=4.

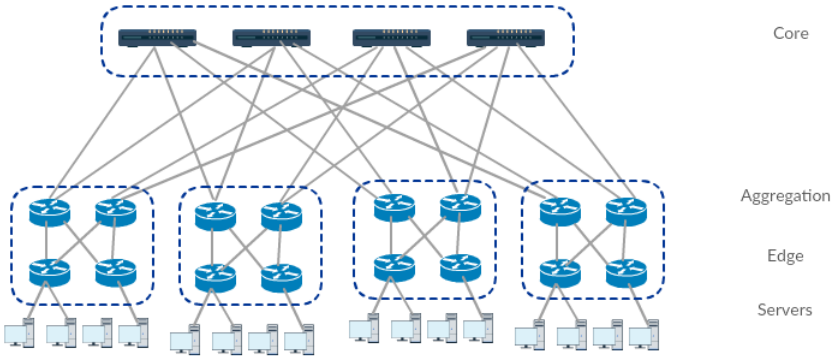

Figure 3.1: Fat-tree Architecture

In a regular tree, a parent produces two children, each of which in turn produces two children each. Hence the width of the tree increases with the increase in the number of children and since each child is connected to only one parent, the communication bandwidth for each node remains the same. Hence the efficiency does not depend on the position at which they are placed i.e. nodes can be placed at any arbitrary position. However, the problem with this structure is that the same branch children cannot directly communicate instead they can communicate only through their parent. Similarly, the children and grandparent cannot communicate directly without involving a parent.[13] Hence if two children want to communicate to the

grandparent, they can't communicate in parallel and if one link goes down the entire communication of the branch is broken. However, these problems do not exist in the fat-tree since the routers are connected to each other over multiple paths and the higher level nodes are connected to the lowest level nodes directly too. Hence, if one link fails the effect is quite less on the other nodes as in the case of a normal tree.[8]

3.3.2 DCell:

This is a server centric architecture. It is denoted by DCell k, and n, k are the important parameters, where n is the number of servers a switch is connected to. It is generally around 8 and k is the level. The DCell0 is the building block of the architecture. It has n servers and each of them is connected to the switch.[6] A DCell1 is formed by interconnecting n+1 Dcell0 cells. Every server is connected to a server in the other cell such that no two servers in the same cell are connected to the servers in the same cell. For a level 1 architecture the servers are named as [level id, Server id]. Thus the servers with tuples [i, j-1] and [j, i] are connected with a link for every i and every j>i

A Dcell1 with n=3 is shown in the Fig. 3.2 The switches are represented by rectangles and the servers by circles. The dotted circle enclosed shows the formation of DCell0. The servers are named as [cell level, server number]. The lines represent the connection between them

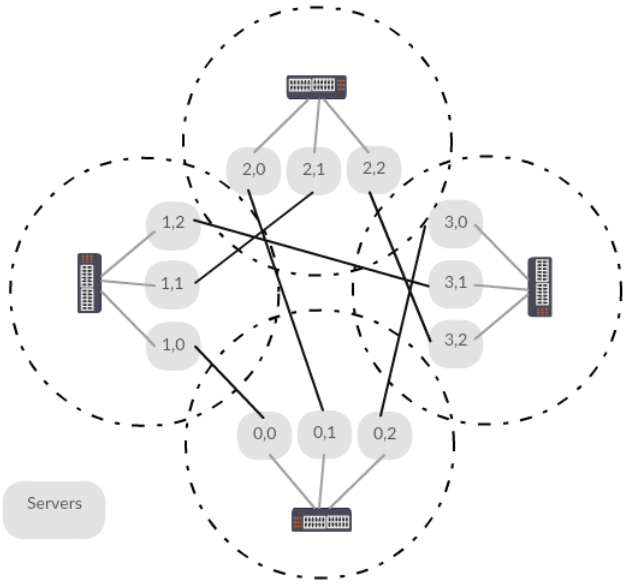

Figure 3.2: DCell Architecture

DCell Routing:

In this Routing the divide and conquer approach is used. When computing path form source to destination in different cells, first an intermediate link (n1, n2) is found and then the same approach is followed for the path between source to n1 and n2 to destination. The final path of the routing is combination of the sub-paths. The pseudo code for the routing can be shown

DCellRouting(src, dst)

pref =GetCommPrefix(*src, dst*)

m=len(*pref*)

if(*m*==*k*)

return(*src, dst*)

(n1, n2)=GetLink(*pref*,s_{k-m}, d_{k-m});

path1= DCellRouting(*src, n1*);

path2= DCellRouting(*n2, dst*);

return *path1+(n1, n2)+path2*;

Here the method GetCommPrefix returns the common prefix between source and destination and the GetLink method finds the links between the two sub-DCells. Hence if (i, j) are the two sub-DCells then the link that inter connects them are (i, j-1) and (j, i) and the results show that the DCell Routing was better than Shortest Path Routing.

Chapter 4

Methodology

4.1 Problem Definition

To compare the performance of Hybrid switching to OCS switching for new Cellular based topological configurations.

4.2 Assumptions

The topologies under the study have applications in data centers with a minimum of 16k servers which means that the amount of data flow through the network is considerably high and hence the open flow simulator simulations which is based on the principle of flows rather than packets will give an almost actual behavior of the network.

4.3 Selection of Tool

The first choice for simulations was ns2 and ns3 due to their popularity and stability.

The problem with packet simulators is that they require a detailed packet level programming which may be a bit complex but certainly not impossible. However, in the case of data center communication, a number of packets coming together for the same destination is huge, and logically binding the packets into streams/flows eliminates the need for studying packet level characteristics and justifies the use of flow based simulation.

Also, ns2 and ns3 simulators have no built in modules for optical communication. Hence the optical switches, optical buffering and optical communication channel was to be built and then integrated for results which are a time intensive task and can be avoided by the use of open flow simulator.

So we made use of simulator described in [10] but doing a lot of modifications like changing the flow creation to simulate the long and short flows, developing the modules for buffer generation and management, developing topologies and routing required for optical communication.

4.4 Selection of Topology

For the comparison, we used the optical data center topologies of type OWCell namely squared, octagonal and cross octagonal cell architecture [7] which are referred in 5.1, 5.2 and **??**

4.5 Selection of Routing

The routing used are XY, octagonal, cross-octagonal routing described in 5.1, 5.2 and **??** respectively.

Chapter 5

Proposed data center topologies

5.1 OWCell:

As proposed in [7],OWCell is a free space optics (FSO) type of architecture which means that the switches are connected wireless by using line of sight (LOS) wireless links. Cell of racks (CoR) is used, which is racks arranged along the vertices of a regular polygon. If the number of vertices is 4, 6, 8 the polygons are square, hexagon & octagon respectively. Several CoR's are connected along the edges to form a graph that represents the OWCell architecture which is denoted by C(n, t, S) where n is the number of ToR's in the network, t is the order of the graph and S the servers per rack. In the diagrams, the circle represents the racks and the solid line represents the connection between them.

The different parameters for an OWCell is represented in table 5.1

Property	**n=4d**	**n=6**
No. of CoR's	t^2	$3t^2$-3t+1
No. of ToR's(N)	(n-2)t^2+2t	$9t^2$-3t
No. of Servers	(((n-2)t^2+2t)*S	($9t^2$-3t)* S
No. of Links	t^2*n(n-1)/2	$45t^2$-45t+15
Bisection Width	**t even**: (n-1)t **t odd**: (n-1)(t-1)+$(0.5n)^2$	8t - 5
Diameter	2t -1	2t-1
Max. Degree	2(n-1)	2(n-1)
Min. Degree	n-1	n-1

Table 5.1: OWCELL properties for square and hexagonal cells

5.1.1 Square Cell Architecture

The Fig. 5.1 shows the 4x4 square cell network. It can be represented as C(4, 1, S) with 4 representing the square configuration, 1 means the total levels of the mesh network and since we are considering only the single layer mesh networks it is 1 and 'S' representing the number of servers in each rack. In this the racks are placed along the vertices of a tilted square such that they are connected to each other with wireless optical links represented by the solid black lines. The capacity of each link is 1 GHz. Each top of rack(ToR) can route the flows to the destination and the direction depends on the position of destination from the current node. Since the racks are placed sufficiently apart they are easy to configure.

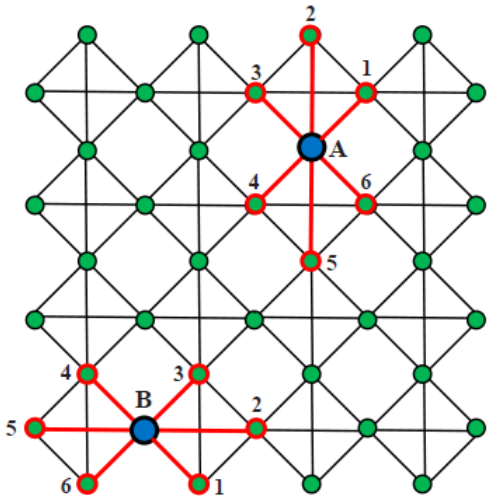

Figure 5.1: Square cell'ed architecture

Square Routing:

The routing used to determine the path is called as square routing. In this 2 random servers with each parameter (x, y, z) is selected where x, y represent the co-ordinates of the server and z represents the arbitrary server number. Here for example in fig.5.2 consider 2 racks which are represented by blue color circles and the red lines represent the possible path to the neighbors. The algorithm can be defined as:

Input: Source and Destination server

Output: Path P from source to destination

while Current ≠Dest do

$\triangle x$= $Dest.x$- $Current.x$

$\triangle y = Desty - Current.y$

if $Current.y$ *is even* **then**

$Next \leftarrow route\_even(Current, \triangle x, \triangle y)$

else

$Next \leftarrow route\_odd(Current, \triangle x, \triangle y)$

$Path \leftarrow Append(Path, Next)$

$Current \leftarrow Next$

Return Path;

5.1.2 Octagonal Cell

This can be represented as C(8, 1, S) with 8 representing the octagonal configuration, 1 means the total levels of the mesh network and since we are considering only the single layer mesh networks it is 1 and 'S' representing the number of servers in each rack which is variable depending on the size of the configuration. In this topology, the racks of servers are arranged along the edges of each rack in the cell connected to each other with 1 GHz links represented by the solid lines as shown in fig 5.2. The advantages of using this topology are that the servers are arranged closely and hence the cell diameter is very small as compared to that of the square cells if considering the same architechtural configuration. The racks can be divided into two categories

1. **Edge racks:** The racks which are common to the cells.
2. **Middle racks:** The racks which belong to a single cell.

For the purpose of routing the middle racks are redundant. Only the edge racks route the flow depending on the destination.

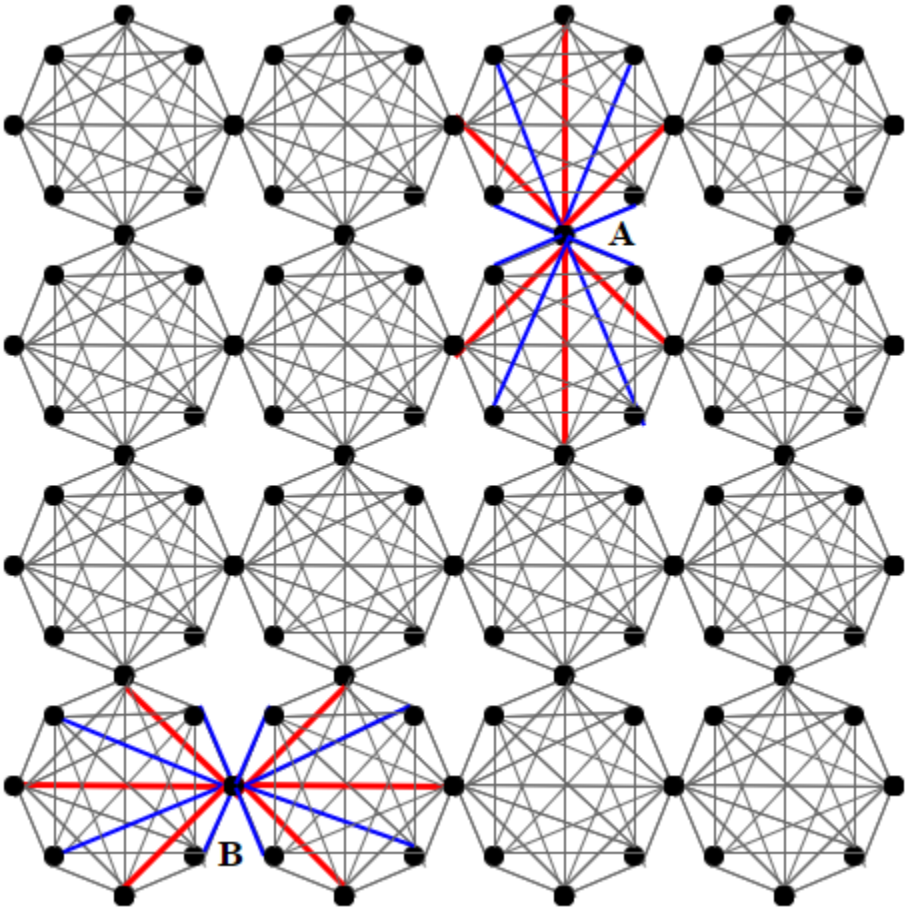

Figure 5.2: Octagonal cell architecture

Octagonal routing:

The routing used to determine the path in a octagonal cellular network is called as octagonal routing. In this two random servers with parameters (x, y, z) are selected where 'x', 'y' are the co-ordinates of the server and 'z' is the arbritrary server number. The algorithm can be defined as:

Input: Source and Destination server

Output: Path P from source to destination

while *Current* $\neq$*Dest* do

$\triangle x$= $Dest\_x$ - $Current\_x$

$\triangle y$ = $Dest\_y$ - $Current\_y$

$find\_next(Current, \triangle x, \triangle y)$

if *next is x* **then**

$Next \leftarrow route\_x(Current)$

if *next is y* **then**

$Next \leftarrow route\_y(Current)$

else

$Next \leftarrow Current$

$Path \leftarrow Append(Path, Next)$

$Current \leftarrow Next$

return path;

5.1.3 Cross-Octagonal architecture:

This architecture is similar to that of octagonal architecture except that the middle racks participate in routing to by interconnecting the middle racks of adjacent cells as shown in the fig. 5.3.

The advantages that is, the octagonal topology has the same number of routing racks like in the case of square cells, which becomes a bottle neck and in-turn results in lower performance. Hence by adding additional links in this architecture, we are trying to increase the number of routing racks and thereby improving performance.

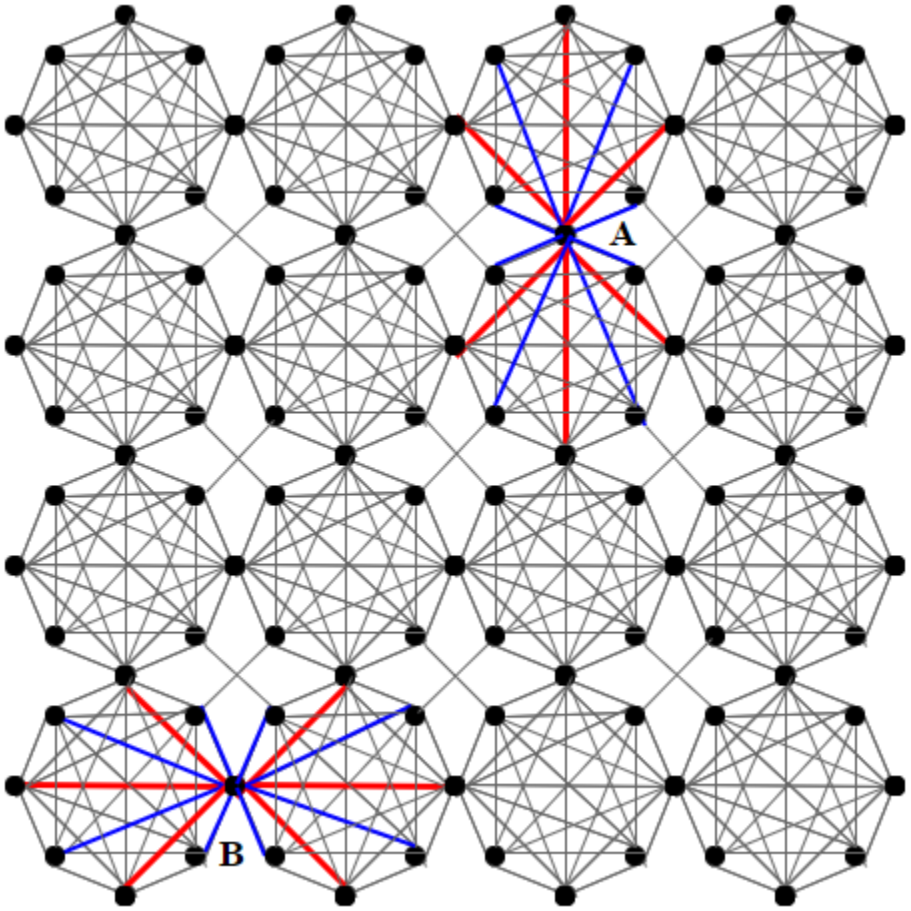

Figure 5.3: Cross-Octagonal cell architecture

Cross-Octagonal Routing:

Most of the routing is similar to that of the octagonal routing except that in addition to octagonal routing here the middle racks are involved in routing too. The algorithm can be defined as:

Input: Source and Destination server

Output: Path P from source to destination

while *Current* $\neq$*Dest* do

$\triangle x$= $Dest\_x$ - $Current\_x$

$\triangle y$ = $Dest\_y$ - $Current\_y$

$find\_next(Current, \triangle x, \triangle y)$

if *next is x* **then**

$Next \leftarrow route\_x(Current)$

if *next is y* **then**

$Next \leftarrow route\_y(Current)$

else

$Next \leftarrow route\_middle(Current)$

$Path \leftarrow Append(Path, Next)$

$Current \leftarrow Next$

return path;

5.1.4 Modified cross octagonal routing

In this we modify the middle rack to have more routing options. This is done because we think that the bottle neck in normal octagonal and square cellular networks is the less routing capabilities. Hence by adding cross links in cross octagonal cellular network the performance increased. Here in this topology we are tryig to see if adding more links to the middle racks as shown in the fig. 5.4 will have impact on the performance.

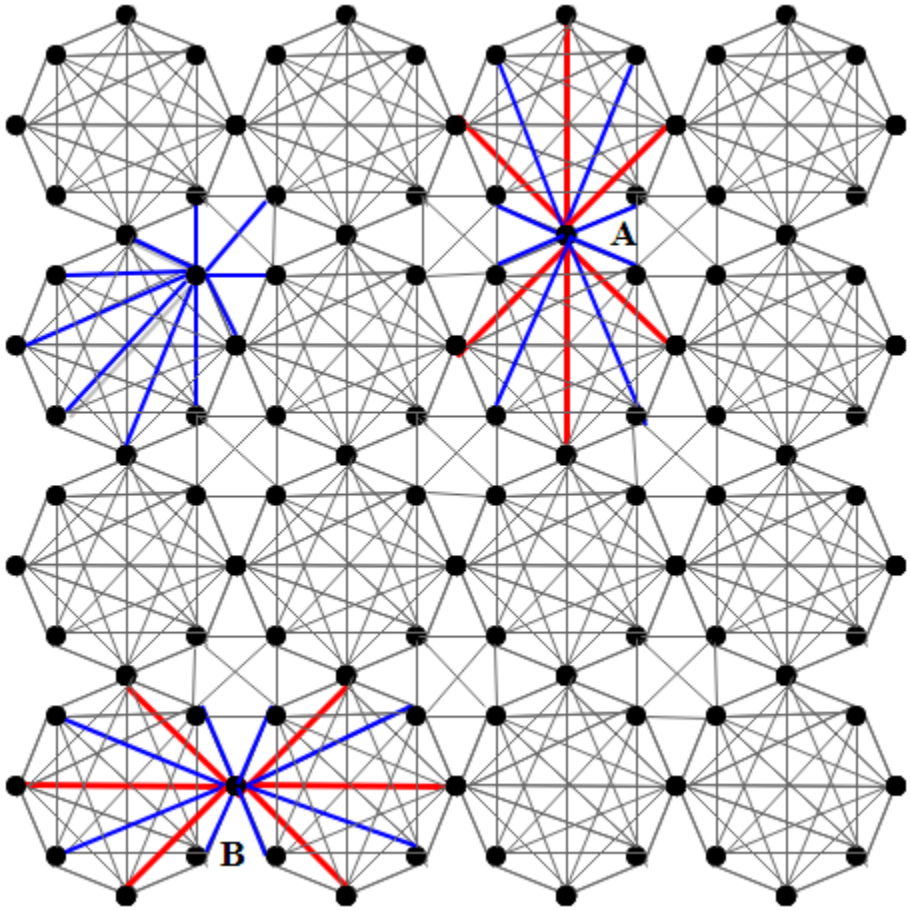

Figure 5.4: Modified cross-Octagonal cell architecture

Chapter 6

Simulation:

6.1 Functional design

The tool can be divided into several parts depending on the functions as shown in the fig. 6.1.

Main:

This module consists of the code which does the integration with other modules such as the network, IO and Algorithm.

IO:

This module takes care of reading the values from the parameter files and storing the results in the text file.

Network:

This module is the heart of the tool which does the major simulation of flows and buffers. For this, it invokes the appropriate classes from the algorithm module.

Random:

This contains the type of distribution which can be used in flow generation.

Algorithm:

This module consists of the algorithms that implemented. It is invoked to find the path between the source and the destination.

Topology:

This part contains the topologies which are implemented as part of studying the optical data centers.

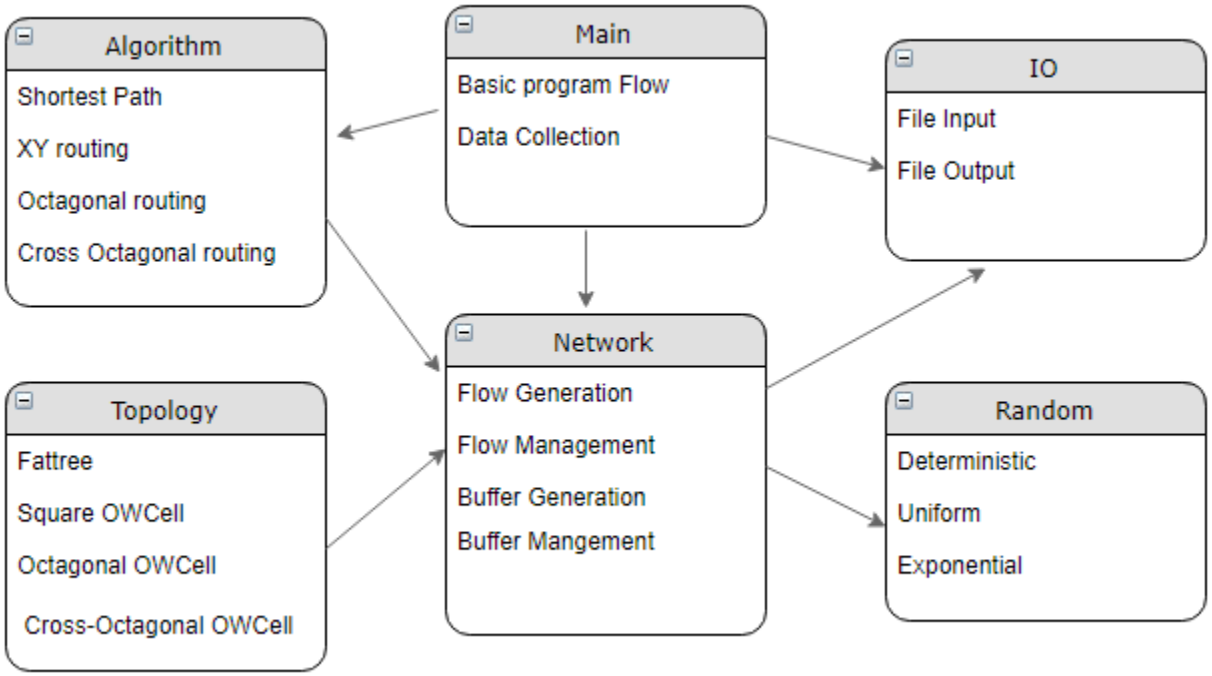

Figure 6.1: Functional Diagram

6.2 Working of the tool

The tool used for the simulation is flow based simulator.[12] It is developed in C++ which can be used to measure the performance of networks.

Flow Simulation:

The traffic is modeled in terms of flows which are independent of each other i.e. new flows do not have any effect on the existing flows. There are two types of flows which are generated, long flows and short flows which have a uniform distribution. The differentiation is based on the duration. The typical duration of a short flow is 100 to 2000 msec while for the large flows is 2000 to 5000msec. This is done to emulate the data center traffic. The advantage of considering flows is that we don't have to consider the packet level details and the assumption is that the flow consists of several packets which are routed along the same path from the source to the destination. The tool is triggered by the arrival of a new flow, whenever a new flow is generated all the flows before the current time are evaluated and the expired flows are pruned. The new flow is generated after the inter-arrival duration of the first flow. The inter-arrival duration is an exponential function to determine the congestion in the network with values ranging between 1-10msec.

Buffer Simulation:

To facilitate the hybrid switching the concept of buffers is introduced. Every optical switch in the network is supposed to store the smaller flows based on the source and destination and then burst are generated when the buffer capacity is reached. The buffers are stored/emptied for during the OBS/Hybrid switching mode of operation .The small flows will not be stored in buffers in OCS mode, and are routed directly.

Link simulation:

Links are the path that exists among the switches, or between a server and switch. Each link has a maximum capacity which is reduced each time there is a flow requests

the bandwidth and the capacity is increased when the flow is terminated. So after the route from the source to destination is determined by the algorithm, bandwidth is requested for each link along the path which can be upto 50% of the total link capacity.

Flow Rejected/ Flow Blocked:

When a path is found from source to destination, the flow requests capacity form the links. If every link along the path has the capacity, then the flow is accepted. However, if the requested capacity for any of the particular link along the path is less than the available capacity, then the flow is blocked. However when the path is not found for the given source and destination, then the flow is rejected.

Updating Network

Whenever a new flow is generated the interval-time is added to the current time and all the flows which have time period less than that of current time i.e. have completed the duration before the current time are pruned and the corresponding link capacity is restored.

Topology/Routing

The simulator is parameterized to for any network or routing algorithm. Any network can be created by specifying the number of nodes, switches and the links between them. Similarly, any routing algorithm can be created and be used.

The Overall operation of the simulator is shown in the Fig. 6.2

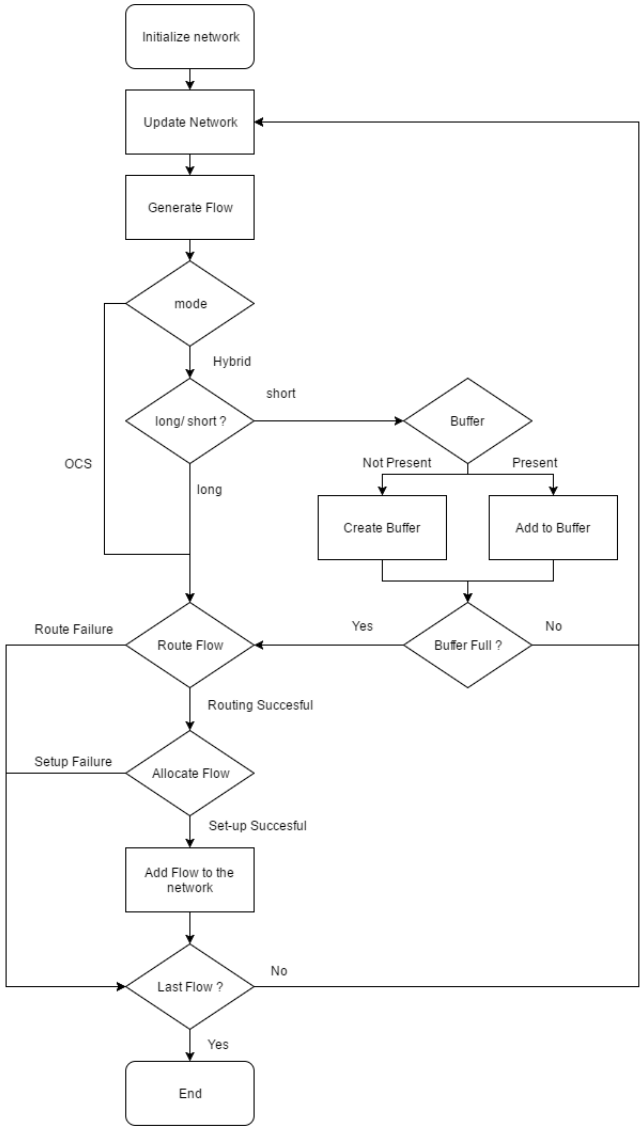

Figure 6.2: Process Flow

6.3 Features:

Data Traffic:

The traffic emulated by this simulator is similar to a practical data center which has ON/OFF traffic. Also, the traffic is inconsistent and not uniform. For this the

data in divided into large and small flows and the cutoff is 25MB.

Efficient Data routing:

This tool efficiently routes the non-uniform data flows with long flows being routed using circuit switching and smaller flows being buffered and routed using buffer triggering or time triggering.

Routing Modes:

The tool offers three routing modes.

1. The traditional **OCS switching mode** in which both the short and long flows are routed by circuit switching.

2. The **OBS switching mode** in which both the short and large flows are routed by storing them in the buffers.

3. The **Hybrid switching mode** in which the long flows are routed by OCS and shorter flows are routed by storing them into buffers and using one of trigger modes.

In the hybrid and OBS switching modes the flows are stored and are aggregated as per the destination which can be triggered by three possible modes:

1. Time Triggered mode: In this mode, the flows are stored only for certain period of time, which when expires the total number of flows at that time are combined and are routed to the destination. The typical values are 10ms.

2. Buffer Triggered mode: In this mode, the flows are triggered when the buffer is full which can be parameterized for different values from the parameter file. Typical values are 30-50MB

3. Hybrid Trigger mode: In this mode, the flows are triggered by either time triggered or buffer triggered mechanism whichever occurs earlier.

Scalability:

This tool can be used be used for any topology and routing i.e. simulations would work for any new topology and routing.

6.4 Input Parameters:

Flows:

It determines the total number of flows injected for the simulation.

E.g. Flows 10000 means 10000 flows are injected.

Buff_Type:

It determines the type of Buffer triggering used.

E.g. Buff_Type 0 means Hybrid Trigger, Buff_Type 1 means Buffer Triggered, Buff_Type 2 means Time Triggered.

Sim_Type:

It determines the type of simulation used.

E.g. Sim_Type 0 means OBS and OCS simulation, Sim_Type 1 means OCS simulation only

Topology:

It determines the type of topology to be used for simulation with its input parameters.

E.g. Format is Topology <topology_name> <input_parameters>

Lcapacity:

It determines the normalized link capacity.

E.g. The format is Lcapacity C <Normalized_capacity>

Endpoints:

It determines the process of choosing source and destination

E.g. Endpoints U means that the process is uniform while E means that the process is exponential.

Farrival:

It determines the inter-arrival time between two flows. Inter-arrival time is the time after which the second flow starts. It is exponential range with the mean 10ms to emulate the behavior of a practical data center.

E.g. Farrival E <time in ms>

Processing Time:

It is the average processing delay that each flow experiences. It is higher for hybrid networks due to buffer storage.

E.g. Processing Time <time in ms>

Fduration:

It is a uniform range which a flow can take. To distinguish between the large flows we have two types Fduration1 and Fduration2.

Fduration1 U 2000 5000 means uniform distribution in the range 2000-5000ms i.e. for large flows (>25MB)

Fduration2 U 0.8 2000 means uniform distribution in the range 0.8-2000ms i.e. for short flows (<25MB)

Fbandwidth:

It is the range of the links in every path that each flow can reserve.

E.g. Fbandwidth U 10 50 means each link along the path can reserve up-to 50% of link capacity with distribution being uniform.

Algorithm:

It mentions the routing algorithm to be used.

E.g. Algorithm <Algorithm_name>

Buffer:

It determines the size at which the buffer becomes full. If we are using time trigger this value will not be used for calculations.

E.g. Buffer <Capacity_in_MB>

Time:

It is the time up-to which the buffers will be stored.

E.g. Time <Time_limit>

6.5 Output Values:

Flow Total :

It is the average of total number of flows of all the iterations that are evaluated by the system

E.g. Flow Total <Total_Flows>

FlowAccepted:

It is the average of total number of flows of all the iterations which are successfully routed by the simulator.

E.g. FlowAccepted <Number_of_flows_accepted>

FlowBlocked:

It is the average of number of flows of all the iterations for which the route cannot be found. If the routing is proper the number is generally 0.

E.g. FlowBlocked <Number_of_flows_blocked>

FlowRejected:

It is the average of total number of flows of all the iterations which are dropped due to one or more links in the path being full.

E.g. FlowRejected <Number_of_flows_rejected>

Data Total:

It is the average of the total data in MB over all the iterations of all the flows.

E.g. Data Total <Data_of_Total_flows>

Data Accepted:

It is the average of the total data in MB accepted over all the iterations.

E.g. Data Accepted <Total_data_in_accepted flows>

Data Blocked:

It is the average of the total data in MB

E.g. Data Blocked <Total_data_in_Blocked_flows>

RejectSetup:

It is the average of total data over all the iterations in the flows that were dropped due to insufficient capacity in the Links.

E.g. RejectSetup <Total_data_in_Rejected_flows>

AvgHopcount:

It is the average of the average of the number of hops required to reach the destination.

E.g. AvgHopcount <number_of_hops>

Chapter 7

Experiments and Results:

In the First Experiment try to study the effect of changing the buffer size of the Hybrid network and compare its performance with the similar configuration but in the OCS mode for different server sizes. To start with we run we select the OWCELL square architecture, and we run the experiment by using the OCS switching mode and comparing it with different buffer sizes (10MB to 50MB) in hybrid mode.

Study 1.

The first experiment is for 16K servers, and the variation of change percent change in accepted flows to the total number of flows is in the Graph 7.1

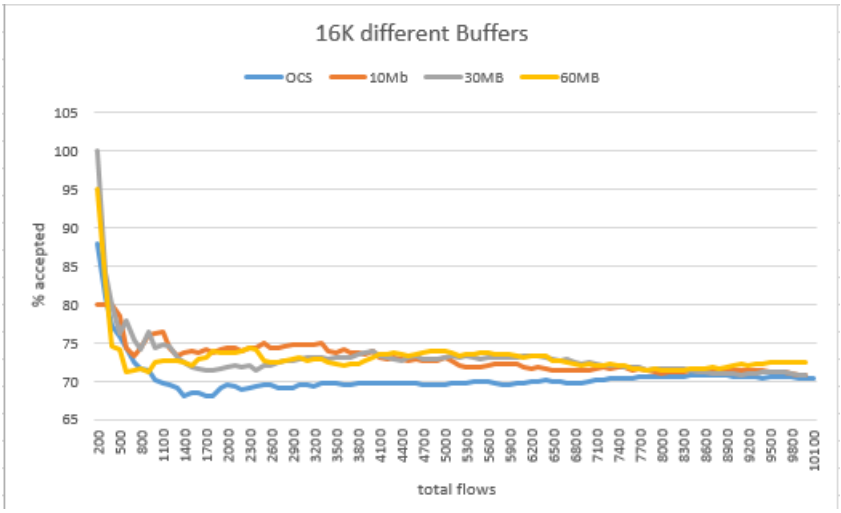

Figure 7.1: Flows vs Percent data accepted for 16k servers

Here we see that hybrid switching performs better than OCS switching in the start till about 8k flows and then the performance of hybrid switching is same as that of OCS switching. This can be explained by the fact that initially the buffers are empty and hence storing the flows causes less number of lows to be dropped but as the traffic becomes uniform the buffers are filled and emptied at constant rate.

In the graph 7.2 we see that hybrid switching performs better than the OCS switching. This is because even when we increase the architecture size, the worst performance of the hybrid switching would be that it stores only 1 flow, which is the maximum what OCS can switch at a given time. However, the performance would increase if the average number of flows per buffer increases.

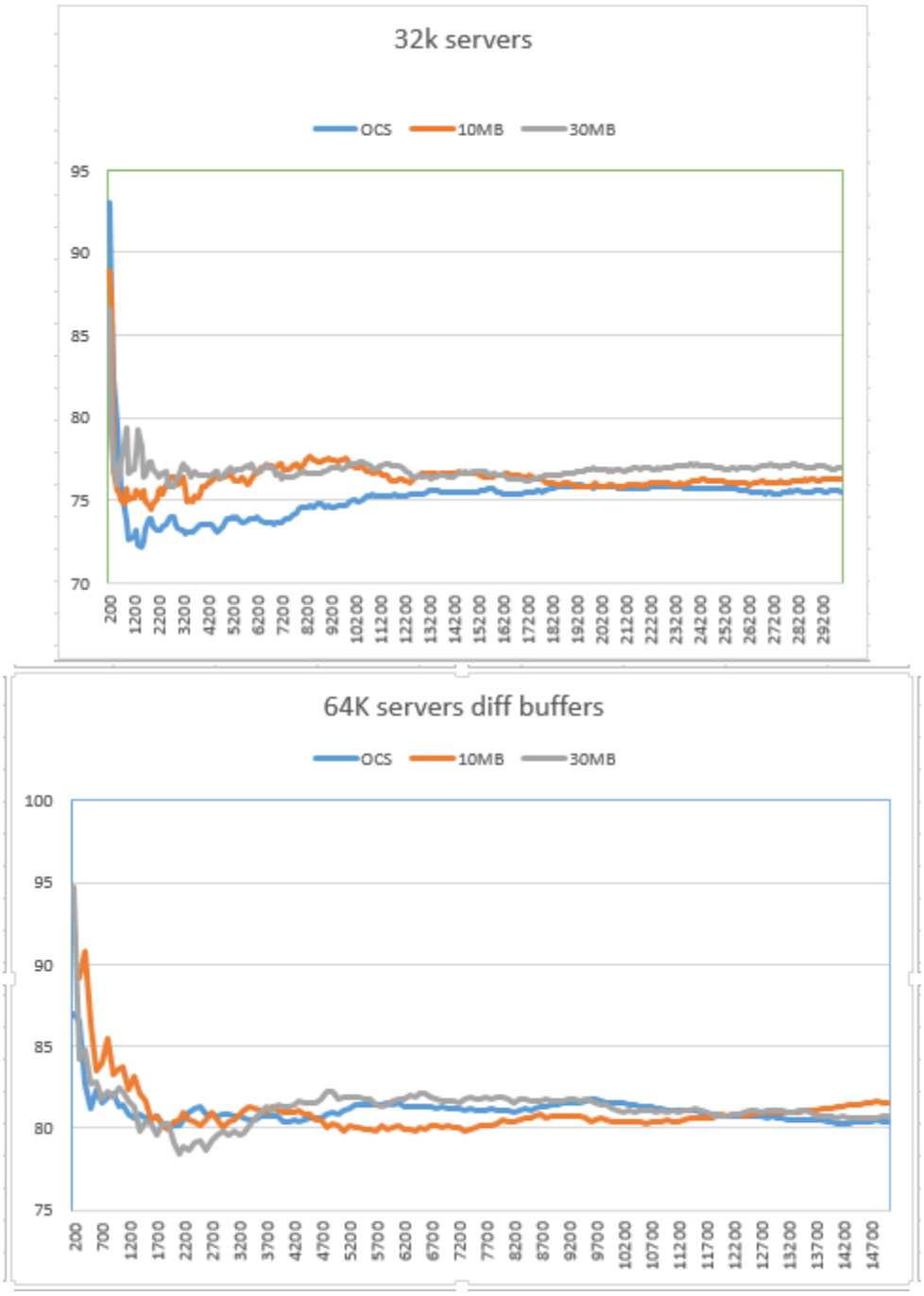

Figure 7.2: Flows vs Percent data accepted for 32k and 64k servers

Study 2.

The graph 7.3 shows the percentage difference between the performance of the hybrid and OCS for different data center sizes for a square OWCELL architecture. The buffer size is kept to be the same at 30MB.

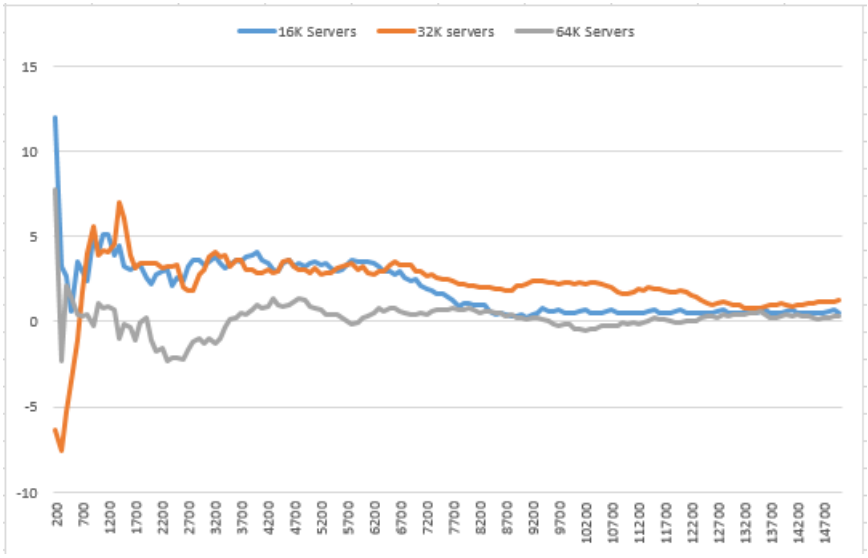

Figure 7.3: Percentage difference between OCS and Hybrid vs Flows

Here we see that as we increase the size of the data center keeping the buffer size constant, the difference between the performance decreases as we increase the size of data center. This behavior can be explained by the following table. As we increase the size of the data center the number of ToR's increases and hence the probability of filling the buffers decreases, also the number of links in the network increases hence the performance of OCS increases too so as that of hybrid and hence the difference decreases as we increase the number of servers.

Study 3:

Another trend that we see that is the number of flows at which the performance of the hybrid data center peaks. In the graph 7.3 we see that this increases as the size of data center increases.

This can be explained by the fact that as the size of data center increases the number of ToR's increases and hence the number of buffers and the probability of filling the buffers decreases. This means that the same buffer which was getting filled faster will take longer to fill and hence the increase in the number of flows required to reach the maximum value. Table 7.1 gives an estimation on the number of ToR's in the data center and the flows at which the performance peaks.

Table 7.1: Server connections and performance parameters

Total servers	**Number of ToR's**	**Number of links**	**peak performance**
16k	256	111360	900
32k	529	230115	1400
64k	1024	445440	4800

Study 4:

In this study, we study the effect of congestion on the data center network. In this, we change the congestion in the same data center architecture and also see the effect on data center with the change in the congestion in the network. Here we choose the data center of size 8k servers and 16k servers and the buffer is kept constant at 40 MB with increasing congestion in the network. Here we see that the performance of the data center decreases with increase in the data center. This is expected as the increase in the traffic causes the links to saturate faster and hence the performance

of the links decreases, and there is decrease in the overall percentage of the flows accepted. This is the common across both the architecture however if we see the change in performance we will find that the change is observed more in the case of smaller architecture this is because as we increase the architectures the number of links increase which allows to accept the increase more gradually than that with less number of links. This can be observed in the graph 7.4.

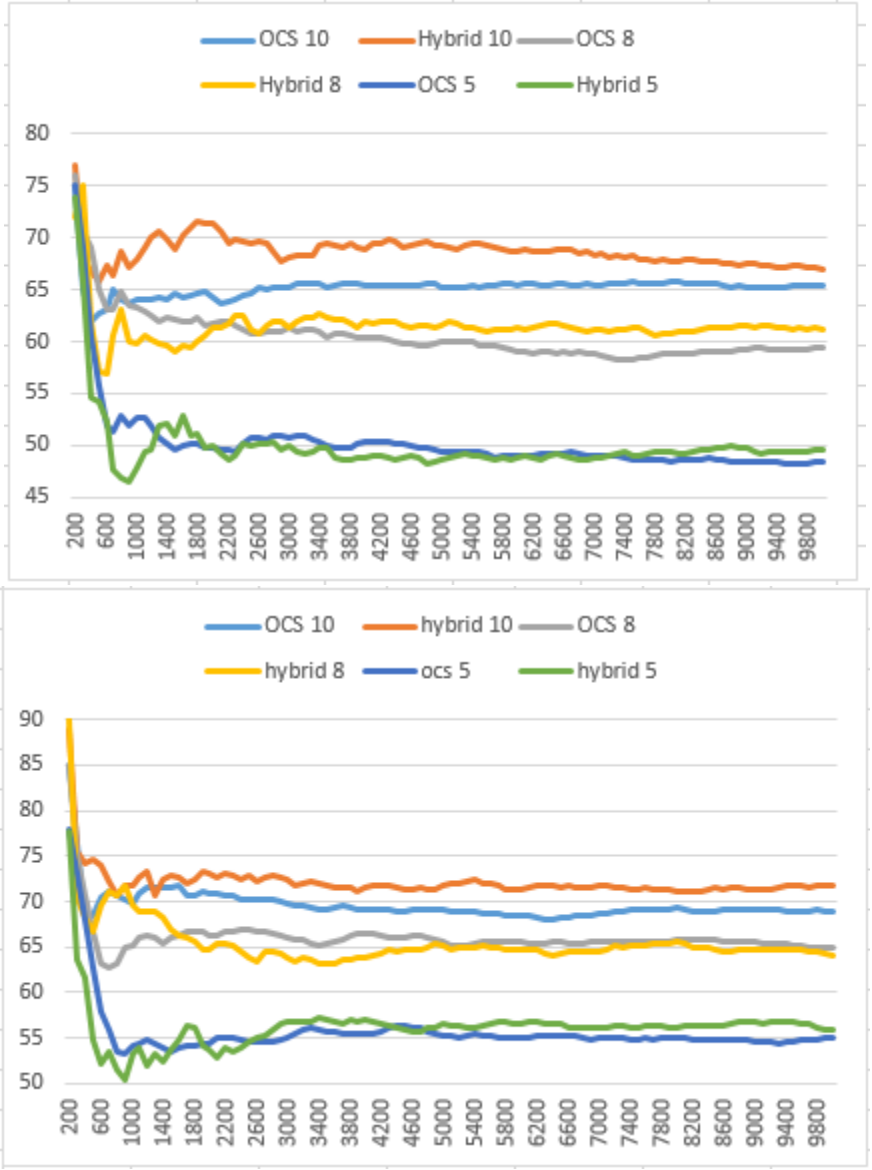

Figure 7.4: Effect of Congestion on network

Study 5:

In this study, we compare the performance of square cell data centers to that of octagonal and cross-octagonal cell data centers. In the graph 7.5, we can see that the octagonal and cross octagonal performs better than that of square cell data center.

This is because the number of links in the octagonal data center is higher and hence the average congestion per links in less in the octagonal cell is less.

In another case, the cross-octagonal cell performs better than that octagonal because adding cross links increases the number of paths possible. In Octagonal cell topology, only the racks on the edges are the route to adjacent cells which changes when we add cross links thereby removing the bottleneck of the topology architecture. This also explains why smaller configurations will perform better for cross octagonal configuration and difference getting smaller as we increase the number of servers.

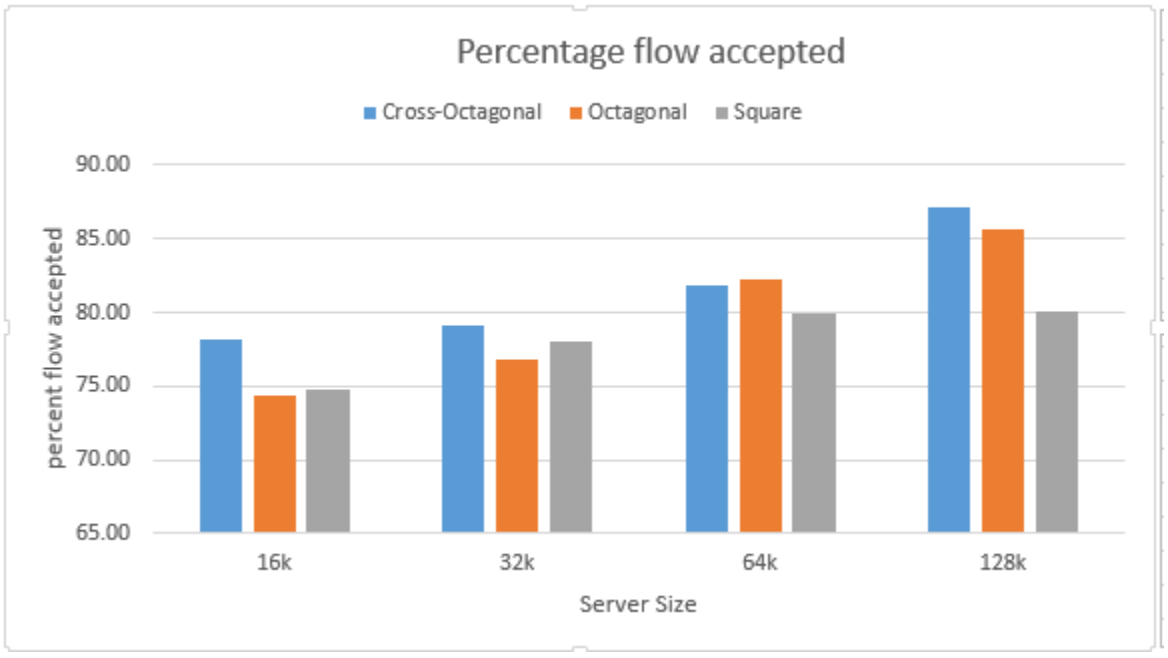

Figure 7.5: Comparison of Cross Octagonal, Octagonal and Square Celled architecture

Study 6:

In this study, we examine the cell diameters of the square cell and octagonal cell architecture. In the fig. 7.5 we see that cross octagonal cell architecture performs better than that of square cell architecture in general, but for certain configurations(32k

& 64k servers) we see that the performance is comparable. But still using cross-octagonal cell is better because the diameter of the architecture is far less which can be shown in the graph

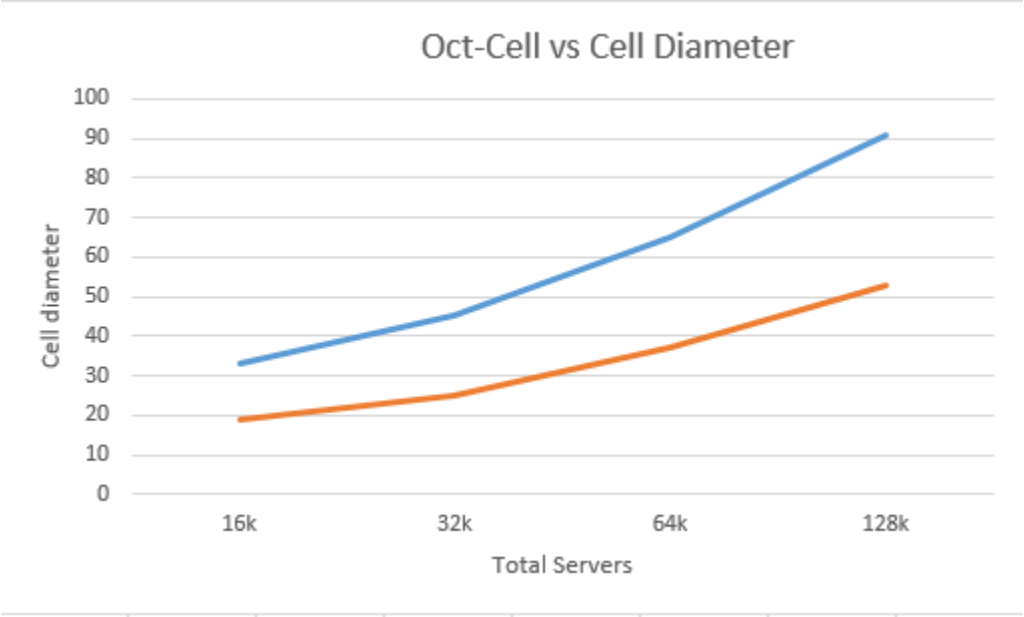

Figure 7.6: Comparison of Octagonal and Square Celled architecture diameter

Study 7:

In this study, we compare the performance of Hybrid switching over Optical circuit switching for cross, octagonal & square cell architecture. For this comparison, we use 8k servers as shown in the graph 7.5 and plot the difference in the percentage of flows accepted using Hybrid Switching and OCS switching. Here we can see that the cross-octagonal cell architecture performs better than that of both square and octagonal cell architecture while octagonal performs similar to that of the square one. This is because although in graph 7.5 the octagonal performs better in terms of percentage flows accepted but in terms of difference in performance they are similar because the number of path possible from each node is 2 in both architectures, while in cross

octagonal cell architectures even the middle nodes will perform routing additional to the edge nodes thereby removing the bottle neck in square and octagonal cell architecture.

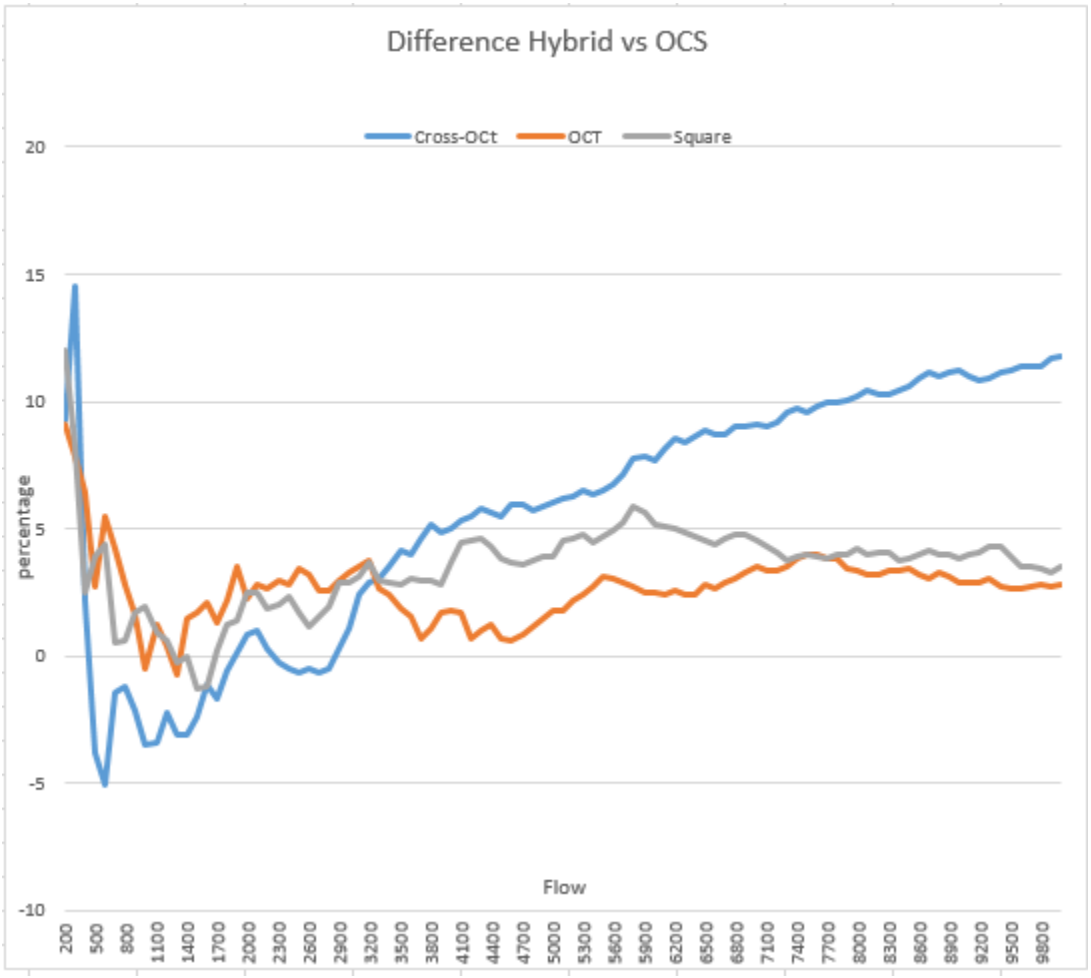

Figure 7.7: Comparison of performance of square, octagonal and cross-octagonal cell architecture

Chapter 8

Conclusion and Future Work

In this thesis, we try to study the optical techniques in Data Centers. OWCELL square architecture which is a new data center network proposed has been already introduced in [7] is studied for performance in the optical circuit switching and the hybrid mode which comprises of optical burst switching along with optical circuit switching. We are currently have published a technical paper in IEEE-ICC about the introduction of square cell architecture and its performance and are currently in the process to introduce the octagonal and cross-octagonal architectures and their advantages over the square cell architecture.

With various studies, it is seen that the Hybrid switching can outperform the traditional optical circuit switching(OCS) by about maximum 8% by choosing appropriate buffer size. Even with its worst performance it is not bad as compared to OCS. However. the general observation is that the peak performance comes for smaller data centers, and the larger data centers are much more stable and do not show much change with the change in traffic conditions. The performance of cross cell architecture is high due to the increasing number routers and thereby increasing the number of possible path explaining its superior performance over square cell architecture.

The performance of hybrid switching for larger data centers can be improved further by improving the routing. Currently, the OWCELL has torus topological structure with octagonal and square cells only. Additional changes can be made to octagonal structure as shown in fig. 5.4 which may impact performance. Also, it would be interesting to see the addition of aggregate and core switches to make it a hierarchical structure which would mean better routing and less number of hops which lowers the probability of link capacity getting exhausted which should be better exploited by Hybrid switching rather than OCS only which would be part of the future study.

Bibliography

[1] Kevin J Barker, Alan Benner, Ray Hoare, Adolfy Hoisie, Alex K Jones, Darren K Kerbyson, Dan Li, Rami Melhem, Ram Rajamony, Eugen Schenfeld, et al. On the feasibility of optical circuit switching for high performance computing systems. In *Proceedings of the 2005 ACM/IEEE conference on Supercomputing*, page 16. IEEE Computer Society, 2005.

[2] Theophilus Benson, Aditya Akella, and David A Maltz. Network traffic characteristics of data centers in the wild. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 267–280. ACM, 2010.

[3] D. J. Blumenthal. Optical packet switching. In *The 17th Annual Meeting of the IEEELasers and Electro-Optics Society, 2004. LEOS 2004.*, volume 2, pages 910–912 Vol.2, Nov 2004.

[4] Yang Chen, Chunming Qiao, and Xiang Yu. Optical burst switching: a new area in optical networking research. *IEEE Network*, 18(3):16–23, May 2004.

[5] James Glanz. Power, pollution and the internet. *The New York Times*, 22, 2012.

[6] Chuanxiong Guo, Haitao Wu, Kun Tan, Lei Shi, Yongguang Zhang, and Songwu Lu. Dcell: A scalable and fault-tolerant network structure for data centers. *SIGCOMM Comput. Commun. Rev.*, 38(4):75–86, August 2008.

[7] Baset Hamza, Suraj Yadav, Suraj Samal, Jitender Deogun, and Denis Alexander. Owcell: Optical wireless cellular data center network architecture. *IEEE-ICC*, pages 1–6, 2017.

[8] James R Heath, Philip J Kuekes, Gregory S Snider, and R Stanley Williams. A defect-tolerant computer architecture: Opportunities for nanotechnology. *Science*, 280(5370):1716–1721, 1998.

[9] Krishna Kant. Data center evolution: A tutorial on state of the art, issues, and challenges. *Computer Networks*, 53(17):2939–2965, 2009.

[10] Tom Kleiberg, Bingjie Fu, Fernando A Kuipers, Piet Van Mieghem, Stefano Avallone, and Bruno Quoitin. Desine: a flow-level qos simulator of networks. In *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, page 10. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

[11] Frank Thomson Leighton. New lower bound techniques for vlsi. *Mathematical Systems Theory*, 17(1):47–70, 1984.

[12] Charles E Leiserson. Fat-trees: universal networks for hardware-efficient supercomputing. *IEEE transactions on Computers*, 100(10):892–901, 1985.

[13] Radhika Niranjan Mysore, Andreas Pamboris, Nathan Farrington, Nelson Huang, Pardis Miri, Sivasankar Radhakrishnan, Vikram Subramanya, and Amin Vahdat. Portland: A scalable fault-tolerant layer 2 data center network fabric. *SIGCOMM Comput. Commun. Rev.*, 39(4):39–50, August 2009.

[14] George Porter, Richard Strong, Nathan Farrington, Alex Forencich, Pang Chen-Sun, Tajana Rosing, Yeshaiahu Fainman, George Papen, and Amin Vahdat. Integrating microsecond circuit switching into the data center. *SIGCOMM Comput. Commun. Rev.*, 43(4):447–458, August 2013.

[15] Ganesh C Sankaran and Krishna M Sivalingam. A survey of hybrid optical data center network architectures. *Photonic Network Communications*, pages 1–15, 2016.

[16] Amin Vahdat, Mohammad Al-Fares, and Alexander Loukissas. Scalable commodity data center network architecture, July 9 2013. US Patent 8,483,096.

[17] Kaishun Wu, Jiang Xiao, and Lionel M Ni. Rethinking the architecture design of data center networks. *Frontiers of Computer Science*, 6(5):596–603, 2012.

[18] Lisong Xu, Harry G Perros, George Rouskas, et al. Techniques for optical packet switching and optical burst switching. *IEEE communications Magazine*, 39(1):136–142, 2001.

[19] Jian Zhen. Five key challenges of enterprise cloud computing. *Cloud computing journal*, 16, 2008.