An Energy Aware Green Spine Switch Management System in Spine-Leaf Datacenter Networks

by

Xiaolin Li

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs in partial fulfillment of the requirements for the degree of

Master of Applied Science

in

Electrical and Computer

Carleton University
Ottawa, Ontario

Abstract

A significant proportion of the operational cost for datacenters is attributed to their energy consumption. Using virtualization techniques in datacenters is enabling the control of electricity use in servers. However, as servers are becoming more energy-proportional, datacenter networks are starting to consume a greater portion of the overall power although networks devices often remain under-utilized. This thesis proposes an energy aware resource management technique for reducing the consumption of energy by the network for a Spine-Leaf topology-based datacenter. The main idea of the system is to keep track of the dynamic workload and enable only switches that are necessary for handling the current network traffic. We have developed an energy aware resource management system for dynamically controlling the number of Spine switches in Spine-Leaf datacenter networks and performed simulation using CloudSim for a number of scenarios. The simulation results show that the system can work effectively to save energy by as much as 63% of the energy consumed by Spine switches in a datacenter comprising a fixed set of 8 Spine switches.

Acknowledgements

I would like to express my sincere thanks to my thesis supervisors, Professor Chung-Hong Lung and Professor Shikharesh Majumdar, for all their guidance and support throughout the completion of this thesis. Furthermore, I would like to than the entire SCE office staff and SCE technical support for all their help.

I also would like to express my gratitude to my beloved parents. They always support me. Without their help, I could not have accomplished my work.

Last but not least, I would like to thank my friends, Zhao Zhao, Susana Cao, Man Si and Fikirte Teka for their kind words and concerns. I'm blessed to have such good people in my life.

List of Abbreviations

ANASS	Average Number of Active Spine Switch
AR	Advance-reservation
BE	Best-effort
CT	Control Threshold
CT-H	the high Control Threshold
CT-L	the low Control Threshold
FBFLY	Flattened Butterfly Topology
FUT	First Utilization Threshold
FUT-H	the high First Utilization Threshold
FUT-L	the low First Utilization Threshold
GSSMS	Green Spine Switch Management System
i.i.d	independent and identically distributed
ISPs	Internet Service Providers
MAWT	Maximum Average Waiting Time
MSDC	Massively Scalable Data Centers
NASS	Number of active Spine switches
NLS	Number of Leaf Switches
NSS	Number of Spine switches
phit	Physical Unit

Pod	Point of Delivery
POPC	Percentage of original power consumed by Spine switches
QDR	Quad Data Rate
REsPoNse	Responsive Energy-Proportional Networks
REsPoNseTE	REsPoNse Traffic Engineering
SLA	Service Level Agreement
STP	Spanning Tree Protocol
SUT	Second Utilization Threshold
SUT-H	the high Second Utilization Threshold
SUT-L	the low Second Utilization Threshold
Tda	Time Duration for Activating
Tdd	Time Duration for Deactivating
VM	Virtual Machine
vPC	Virtual Port Channel

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Datacenters are becoming bigger and their usage is becoming more widespread. Instead of having their own datacenter, more companies choose to subscribe for services from datacenter providers, e.g., Amazon, Google, Microsoft, etc. The service providers endeavor to support reliable, secure, scalable and multi-tenant services with massive datacenters. While the size of such a datacenter increases continually, the power consumed by datacenter increases dramatically as well. According to [Koomey11], from 2000 to 2005, electricity consumed by world datacenters has doubled, but the rate of growth slowed down from 2005 to 2010 – a 56% increment for datacenters across the world, and a 36% increment for datacenters in the US. The report also indicates that, in 2010, electricity used by datacenters across the world is about 1.3% of the total world electricity usage, and for US, it is about 2% of the total US electricity usage. The energy consumed by datacenters still remains substantial and it is important for datacenter service providers to minimize electricity usage for protecting the environment as well as for reducing operational cost.

1.1 Motivation

As mentioned above, the rate of electricity use growth for datacenters slowed down from 2005. The main reason is the increased prevalence of virtualization in datacenters and the industry's efforts to improve efficiency of datacenter facilities [Koomey11]. The sources of the inefficiencies in datacenters include energy non-proportional servers and

over-provisioned servers and power infrastructure [Pedram12]. Energy non-proportional servers mean that servers cannot control its energy consumption in accordance with the workload. In other words, servers always consume a constant amount of energy no matter if the workload is low or high. Over-provisioned servers and power infrastructure, e.g., cooling systems, indicate that in order to handle temporary peak workload, datacenter devices typically remain under-utilized for most of the time. Hence, in order to improve efficiency of a datacenter, researchers have focused mainly on servers and the cooling system, which account for about 70% of a datacenter's total power budget [Heller10]. In recent years, some research efforts have studied methods to improve the energy proportionality of servers, such as [Fan07], [Meisner09] and [Tolia08]. Some papers introduce new policies for task placement or virtual machine migration and placement, which can minimize the number of active servers. Examples include [Goudarzi12], [Liu12] and [Taheri11]. There are also papers about cooling systems, such as [Lee12]. However, not much research has focused on energy saving for the network in datacenters. That is partly because, compared with servers, the network consumes less energy, just 10%-20% of datacenter's total power [Heller10]. However, there are two facts that we should be aware of:

a) Although a datacenter network with a fat tree topology consumes only 12% of overall power when a datacenter is at full utilization, if the servers are full energy-proportional and the datacenter is only 15% utilized, the network will

consume nearly 50% of overall power [Abts10], as illustrated in Figure 1-1.

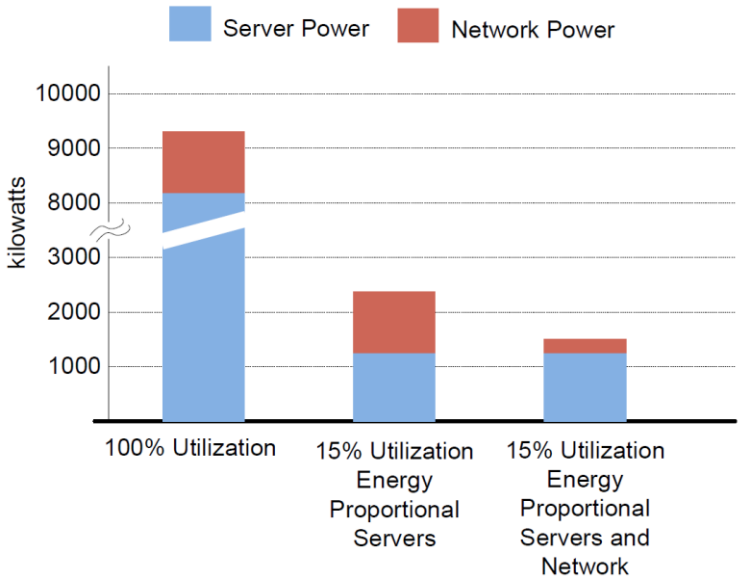


Figure 1-1. Comparison of server and network power [Abts10]

b) No matter for edge links, aggregation links or core links, link utilization is not above 10% during 95% of the time, and does not exceed 30% for more than 99% of the time [Benson10].

The first aforementioned fact shows that as servers are becoming more energy-proportional, the datacenter network will consume a greater portion of the overall power. Therefore, researchers need to consider investigating techniques for saving the power used by the network. The second fact indicates that datacenter network devices are always under-utilized. Hence, effective management of network devices can save energy. Similar to what is being done with servers – by minimizing the number of active servers – we also can find ways to minimize the number of active network devices.

Network devices in this thesis refer to network switches. To minimize the number of active switches, a datacenter network topology must satisfy a certain requirement – the traffic on one switch can be shifted to any other switches on the same layer. Access switches (as shown in Figure 1-2) can never achieve this requirement because they are connected to different servers, and their state (on or off) depends on whether or not all the servers they are connected to are inactive. Therefore, this thesis focuses on aggregation switches (as shown in Figure 1-2) in a datacenter network with a topology called Spine-Leaf (as shown in Figure 1-3). A detailed discussion of the Spine-Leaf topology is presented in Chapter 2. In the Spine-Leaf topology, switches in the Spine layer are aggregation switches, and switches in the Leaf layer are access switches, and every switch in the Spine layer is connected with all the switches in the Leaf layer. The aim of this thesis is to minimize the number of active switches in the Spine layer at a given point in time.

The aim of this thesis is to minimize the number of active switches in the Spine layer at a given point in time. And the scope of the thesis focuses on the design and performance of the proposed system. The protocol running in the Spine-Leaf datacenter networks is not addressed in this thesis. This thesis does not consider the failure mechanism for the proposed system, either.

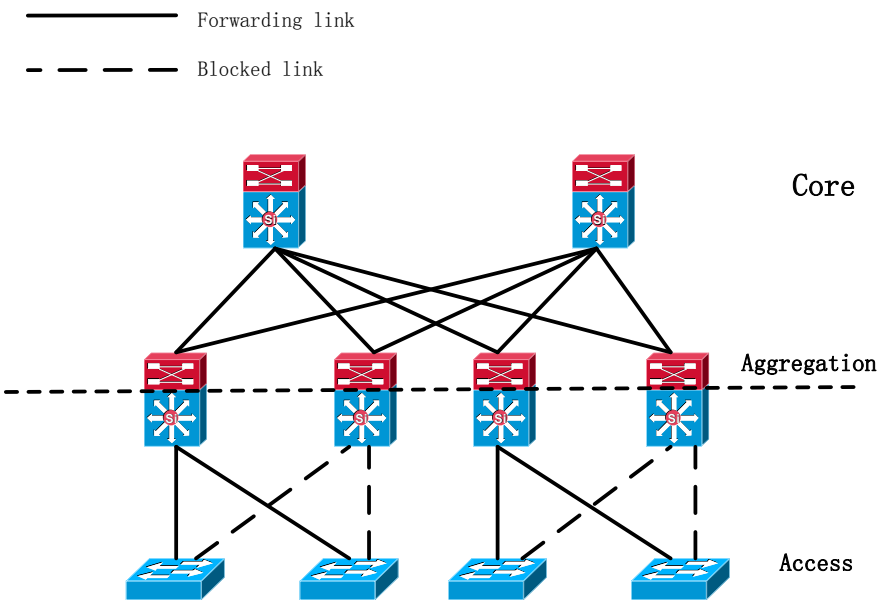


Figure 1-2. Traditional 3-tier Datacenter Network Architecture [Beck13].

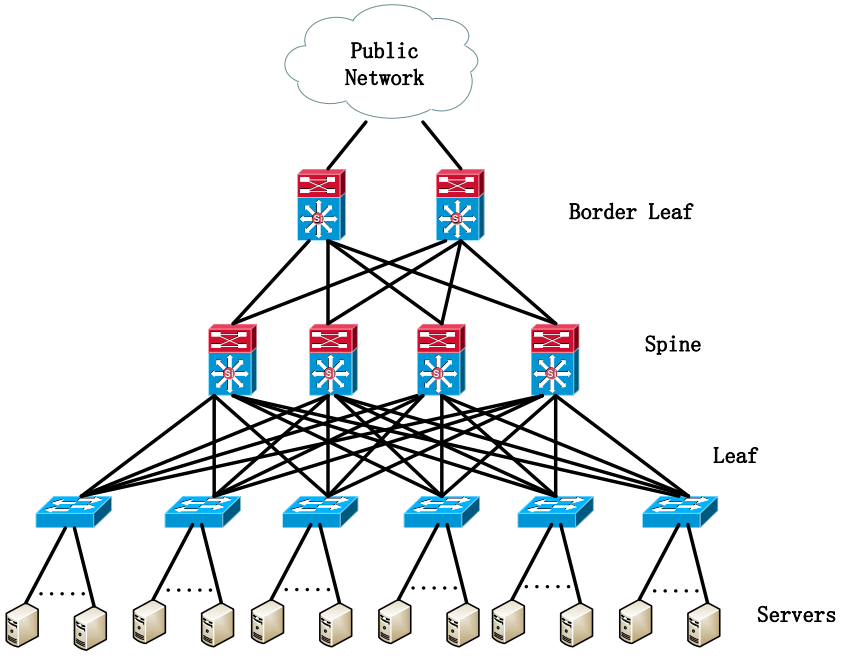


Figure 1-3. The Spine-Leaf Topology [Beck13]

1.2 Contributions

The main contributions of this thesis include:

a) A new technique for energy aware Spine switch management is introduced. The technique comprises algorithms to dynamically control the number of active Spine switches according to the current network traffic.

b) A simulation-based performance analysis of the technique for three different traffic patterns – Uniform traffic, Sine-Wave traffic and Random traffic – is presented.

c) Insights into the relationship between various system as well as workload parameters and performance are described.

d) A set of guidelines for choosing the various parameters controlling the behavior of the algorithms is discussed.

1.3 Dissertation outline

The rest of the thesis is organized as follow: Chapter 2 describes background information and related work. Chapter 3 presents the proposed algorithm in detail, and explains how it works. Chapter 4 shows the setup of simulation and the simulation results. Chapter 5 concludes this thesis and discusses the future work.

Chapter 2: Background and Related Works

This chapter is divided into two sections: (i) an introduction of the Spine-Leaf topology and (ii) related works that present what have been conducted by other researchers in energy saving with datacenter networks.

2.1 The Spine-Leaf Topology

In this section, firstly, a brief introduction of datacenter architecture evolution is provided, explaining why the datacenter network is evolving into the Spine-Leaf topology. The advantages of the Spine-Leaf topology, showing that the Spine-Leaf topology could serve modern datacenter better than the traditional network architecture are outlined. Finally, the description of FabricPath, which is a multipath protocol used in the Spine-Leaf topology, is presented.

2.1.1 Datacenter Architecture Evolution

The traditional 3-tier network architecture, shown in Figure 1-2, has served the industry since mid-1990s, and has been working well for more than 10 years. This architecture is designed for the situation in which the majority of the traffic is between the access and core layer, which is known as “north-south” client-server traffic [Beck13]. In Figure 1-2, the forwarding link is used to transfer data, and the blocked link is backup link which is used to transfer data when the forwarding link fails.

However, as virtualization has become prevalent and big data era is coming, the workload trend has changed and new requirements for datacenters have arisen. Because

of virtualization, virtual machine (VM) and storage migration create a large amount of data that needs to be transferred from one server to another server. The migration generates large volumes of traffic between servers inside datacenters. At the same time, big data is another important trend. Big data refers to diverse data sets that are so large and complex that it is very difficult to process and analyze them in a reasonable time duration and cost with the traditional IT techniques [Beck13]. Therefore, in big data, computing is distributed to many servers, and these servers need to communicate with each other. All these server-server traffic, so called “east-west” traffic [Beck13], makes researchers work on new design for datacenters. One innovation in datacenters is a shift from the traditional 3-tier network architecture to “fat tree” or CLOS based topologies. CLOS network is a multistage switching network proposed in 1952. The Spine-Leaf topology is an example of a CLOS based network [Beck13]. The Spine-Leaf topology is used in massively scalable data centers (MSDC), which is proposed by Cisco [Cisco12] [Beck13].

As shown in Figure 1-3, the Spine-Leaf topology has two types of switches: the Spine switch and the Leaf switch. Spine switches only connect with Leaf switches and do not connect directly with servers. Every Spine switch connects with all Leaf switches. Leaf switches connect with Spine switches and servers. In Figure 1-3, there are some Leaf switches called Border Leaf switches. Border Leaf switches are responsible for connecting to the public network; they work as Core switches in the traditional datacenter

network.

2.1.2 Advantages of the Spine-Leaf Topology

The Spine-Leaf topology has several advantages: (i) making VM placement policy simple, (ii) reducing failures for the network, and (iii) making datacenters easy to scale out.

The advantages of the Spine-Leaf topology are described in detail as follows:

a) Making VM placement policy simple

In the traditional 3-tier datacenter network, server to server connectivity is inefficient. The reason is that in the 3-tier network, it is common to oversubscribe bandwidth in the access layer and the aggregation layer [Beck13]. Oversubscription means that if all servers start transmitting at link capacity, the datacenter aggregation layer and core layer will not be able to handle all the traffic [Dell12]. As a result, available bandwidth between servers in different parts of the datacenter can be different, and available bandwidth may be not enough for some servers [Greenberg08]. Therefore, locations of servers affect the communication between servers. In the datacenter network as shown in Figure 2-1, the traffic between Server1 and Server3 routes through switch S7, S3, S1 or S2, S5 and S9. If Server2 is transmitting data to Server4, and the traffic occupies the majority of the link capacity between switch S3 and switch S5, there may not be enough available bandwidth for the communication between Server1 and Server3. On the other hand, the traffic between Server1 and Server2 is not affected by the traffic between Server3 and Server4. Consequently, if a VM which is going to communicate with a VM

in Server1 needs to be located at a server, but no server in the Server1's rack is available, the better choice is to locate the VM on Server2, not Server3 or Server4. This shows the importance of VMs' locations. Thus, the VM placement policy should manage VMs from all applications very carefully to make sure that the sum of VMs' traffic does not saturate any network link [Greenberg08]. This is a global optimization problem, which is hard to achieve in a reasonable time in practice [Greenberg08].

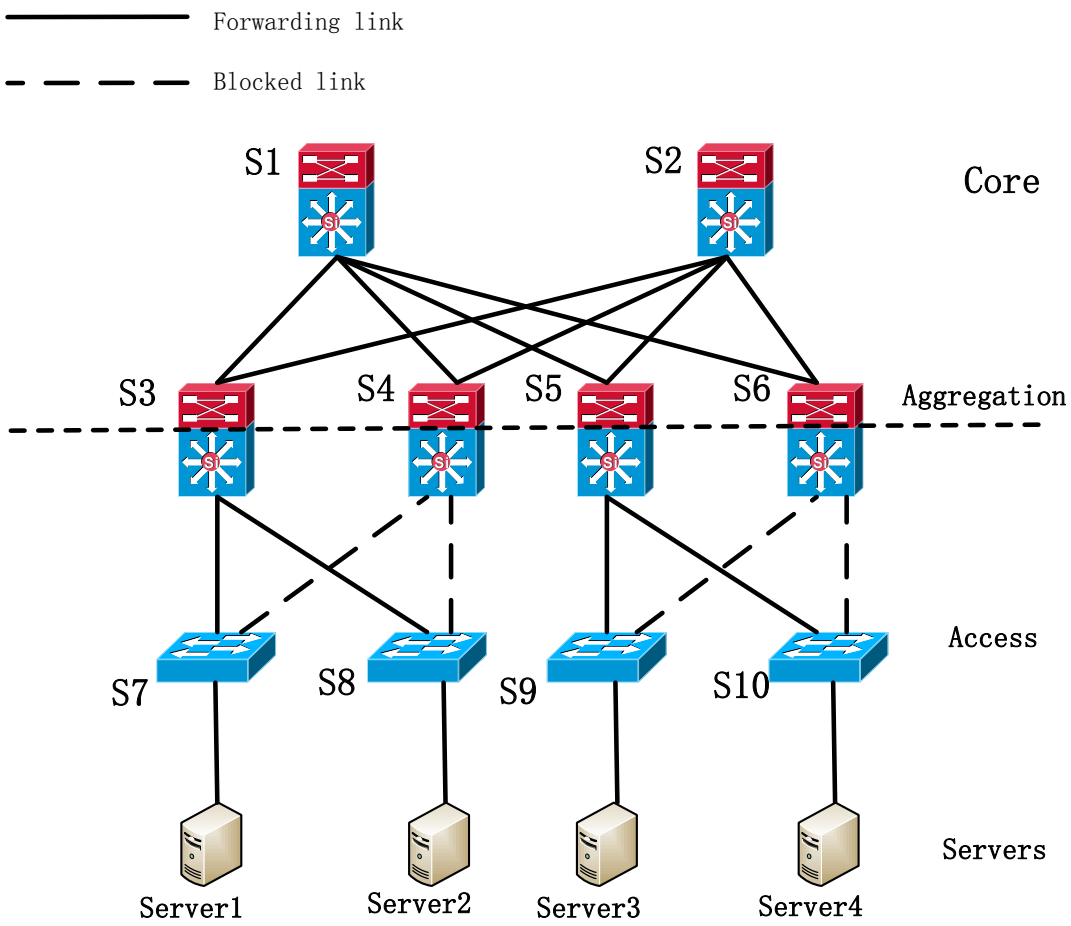


Figure 2-1. Traffic between Servers is affected by Servers' Locations

In contrast, in the Spine-Leaf topology because every Spine switch connects with all

Leaf switches, the traffic between servers in different racks always route through two Leaf switches and one Spine switch. If one link between a Spine switch and a Leaf switch is saturated, such as link between S1 and S5 in Figure 2-2, other Spine switches are available (e.g. Spine switch S2, S3 and S4), and other links are available (e.g., links between S2 and S5, S3 and S5, and S4 and S5). For any server, this feature makes all the servers in other racks equidistant. That means that for any server, all other servers, except the servers in the same rack, have the same distance to it. (The distance here means the number of switches on route). If a VM needs to communicate with Server1 and the VM can be located on Server2 or Server3, locating the VM on Server2 is the same as locating the VM on Server3, because that Server2 and Server3 have same distance to Server1. As a result, when datacenter providers want to place a VM on a server, they do not need to consider on which server the VM should be placed to have better network performance. In other words, datacenter providers do not need to worry about choosing a poor location which can cause higher delay than a good location. This feature makes VM placement policy much simpler.

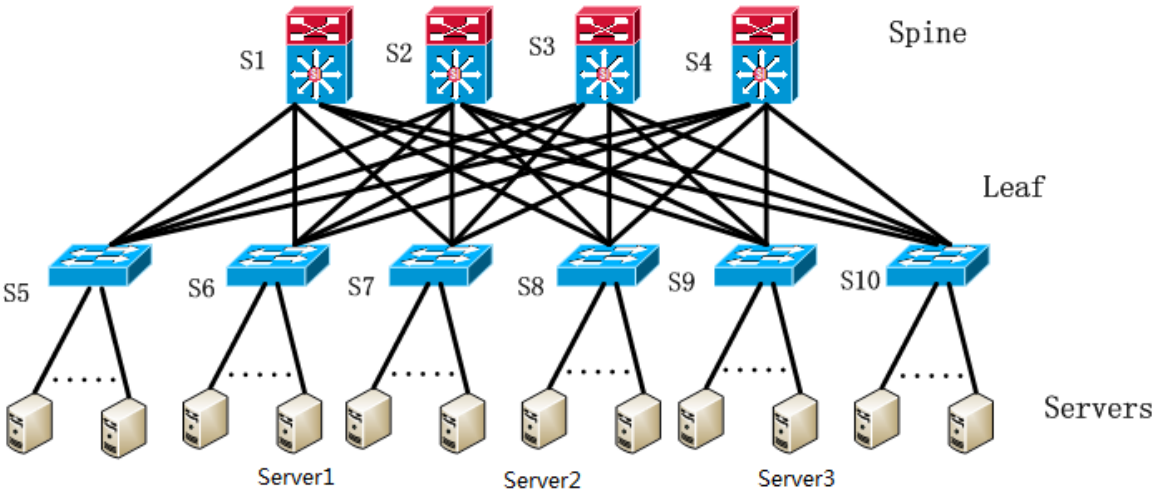


Figure 2-2. Traffic in the Spine-Leaf Topology

b) Reducing failures for the network

Because of the feature of the Spine-Leaf topology (every Spine switch connects with all Leaf switches), the Spine-Leaf topology can reduce failures for the network. If one Spine switch fails, the traffic in the failed Spine switch can be distributed to other Spine switches [Beck13]. As shown in Figure 2-3, if Spine switch S1 fails, the other three Spine switches can take over the traffic and all servers are still accessible. No service will be disrupted because of the failure.

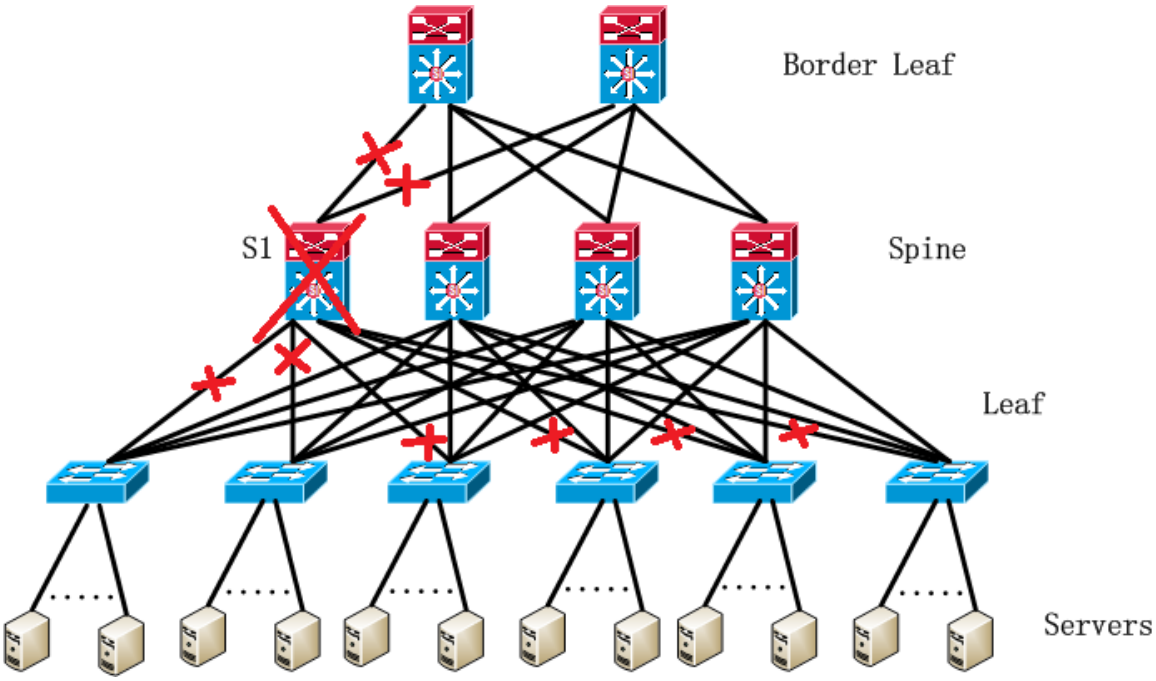


Figure 2-3. If one of Spine switches fails, other Spine switches can take over the traffic in the failed Spine switch

c) Making datacenters easy to scale out

The Spine-Leaf topology makes datacenters easy to scale out [Hedlund12]. If datacenter providers want extra servers in the datacenter, for instance, less than 100 servers, they can simply add one or two new Leaf Switches and new servers without making any change for existing switches or servers in the network. As shown in Figure 2-4, adding one new Leaf switch brings no change to Spine switches and other Leaf switches. However, in this way, the quantity of added servers is limited. Adding more servers and more Leaf switches without adding Spine switches can lead to the increase of oversubscription ratio, which has negative effect on network performance. If hundreds of

servers need to be added, new Spine switches should be added at the same time, as illustrated in Figure 2-5.

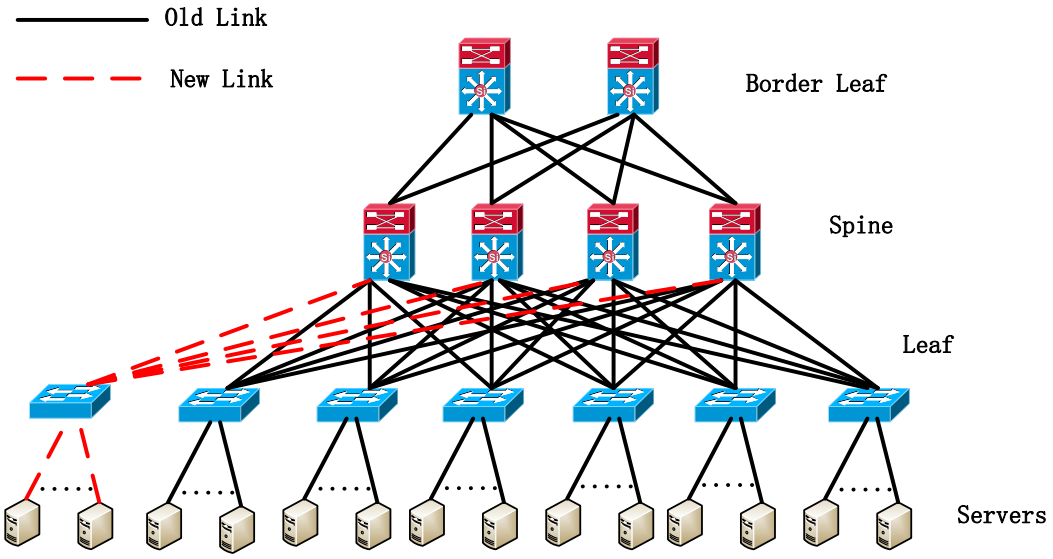


Figure 2-4. Only Add New Leaf Switch in Network

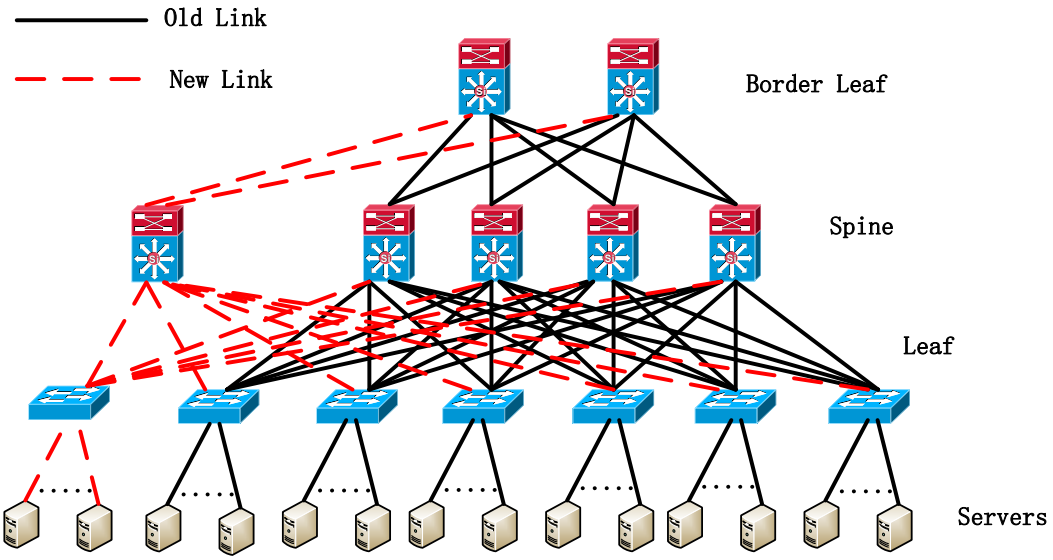


Figure 2-5. Add new Spine Switch in Network

2.1.3 FabricPath [Beck13]

Some protocols used in the traditional 3-tier datacenter network, such as Spanning Tree Protocol (STP), are not suitable for the Spine-Leaf topology. If STP is used in the Spine-Leaf topology, there is still only one path to access a server. Using STP precludes all the advantages of the Spine-Leaf topology. Instead, for the Spine-Leaf topology, datacenter providers can use multipath to scale bandwidth [Alizadeh13] [Beck13]. FabricPath is a multipath protocol used in the Spine-Leaf topology. It basically is multipath Ethernet. It can work on NX-OS (a datacenter operating system proposed by Cisco).

FabricPath combines a number of layer 3 features with current layer 2 attributes to enhance efficiency. In other words, FabricPath makes some capabilities in layer 3 routing available in the traditional layer 2 switching. For instance, FabricPath has the benefits of layer 2, such as low cost, easy configuration and workload flexibility. This means that if VMs and/or applications are needed to move to different physical locations in a datacenter, the datacenter provider can do so in a simple way without requiring VLAN, IP address and other network reconfiguration. Further, FabricPath mitigates the large broadcast domains and broadcast storms inherent in layer 2 networks by employing technologies such as VLAN pruning, Reverse Path Forwarding, Time-to-Live, etc.

The layer 3 features delivered by FabricPath allow datacenter providers to build much larger layer 2 networks. In addition, because FabricPath removes STP and replaces

it with multiple paths between servers in a datacenter, FabricPath provides high availability. This brings increased redundancy because traffic can reach its final destination using multiple paths. FabricPath can not only choose different paths, but also use multiple paths simultaneously for transmitting data, so traffic can span across multiple paths at once. These layer 3 features give FabricPath the capability to use all links between switches to transmit traffic. Therefore, this will bring higher resiliency and bandwidth capacity, which is the most important consideration while scaling requirements are driven up by increasing compute and virtualization capabilities.

In one word, FabricPath provides the good benefits of layer 2 network, while overcomes its drawbacks by having layer 3 features. Compared with the traditional datacenter network, FabricPath has the following advantages:

a) Easier application deployment: Any VLAN can be defined on any Leaf switch with the same configurations and protocols. Therefore, from the viewpoint of applications and servers, network constraints are eliminated.

b) Extended VM mobility range: Because Leaf switches can reach each other at layer 2, simple administration and movement of VMs become possible. As a result, VMs' movement is no longer constrained by the Point of Delivery (Pod) but becomes datacenter wide.

c) Simplicity of configuration: Datacenters with FabricPath have simpler configurations compared with datacenters with STP and Virtual Port Channel

(vPC).

d) Improved reliability: FabricPath has higher redundancy. For instance, in a Spine-Leaf network with 4 Spine switches, the failure of one Spine switch causes just 25% loss of the bandwidth. However, in a datacenter with STP and vPC, the failure of one of four aggregation switches reduces the bandwidth by 50%.

2.2 Energy Saving with Datacenter

Several papers have proposed algorithms for saving energy for datacenter networks. Those algorithms mostly save energy by minimizing the number of network devices, managing port rate, using traffic engineering and managing network device configuration. Finally, this section presents some energy saving algorithms that used for VMs in datacenters.

2.2.1 Saving Energy by Minimizing the Number of Network Devices

In [Heller10], the authors proposed a system called Elastic Tree which is a system for minimizing the power consumption of the datacenter network by shutting down unneeded switches and links. The Elastic Tree is designed for the fat tree topology, which is shown in Figure 2-6.

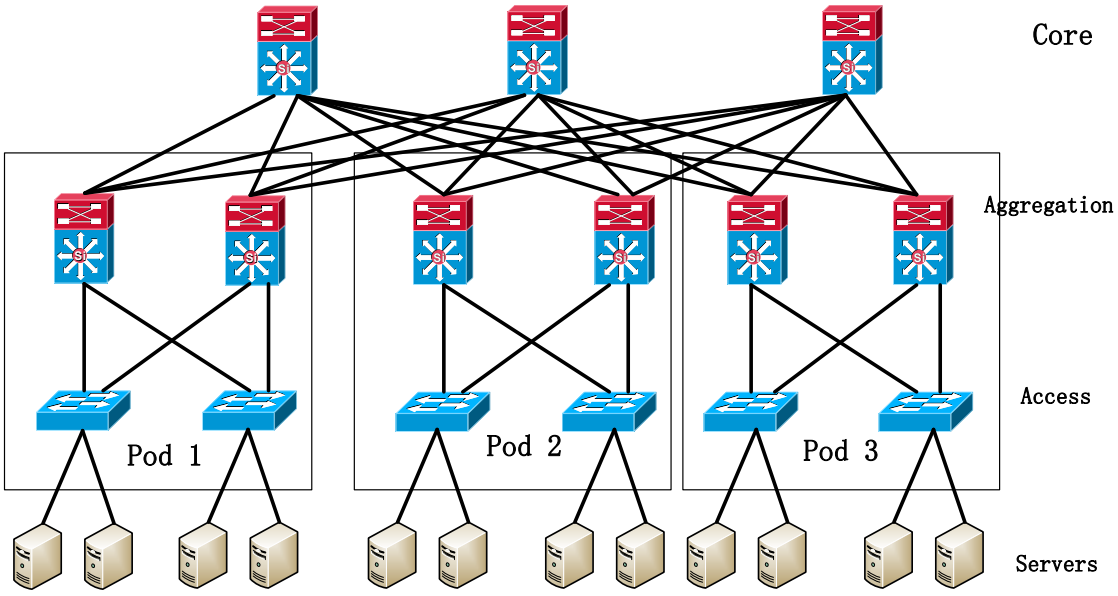


Figure 2-6. Fat Tree Topology [Heller10]

The aim of the Elastic Tree is to manage current non energy-proportional network components according to the network traffic. Its strategy is simple: keep switches and links available only as much networking capacity as required, and shut down those are not needed. Firstly, the Elastic Tree acquires inputs, which include datacenter network topology, the traffic matrix, a power model for each switch and the desired fault tolerance properties. Then based on the inputs, the Elastic Tree chooses the subset of network devices (aggregation switches, access switches, ports, or linecards) that must be active to achieve the required performance and fault tolerance goals.

As shown in Figure 2-7, the Elastic Tree system has three logic modules: optimizer, routing and power control. The optimizer is responsible for finding the network subset with minimum power consumption which can satisfy current traffic with required

performance and fault tolerance. The output of the optimizer is a set of active components. Then the output is sent to both the routing module and the power control module. After receiving the set of active components, the power control module shuts down the ports, linecards, and switches that are not in the set of active components. The role of the routing module is to choose a route for all flows.

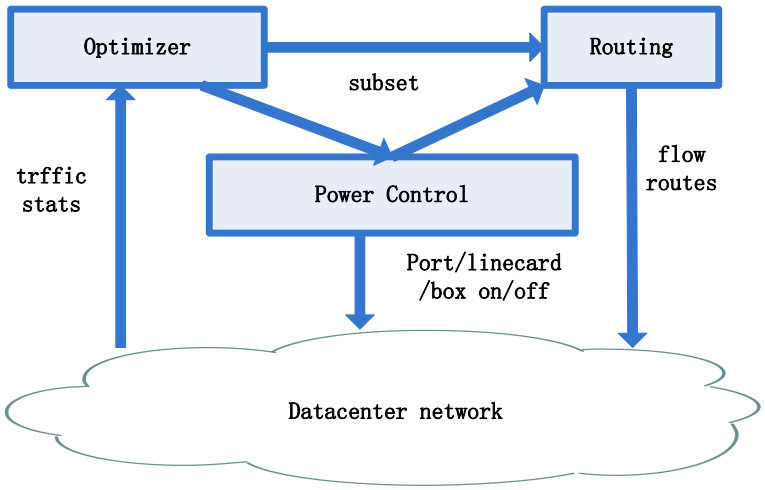


Figure 2-7. Elastic Tree System Diagram [Heller10]

As we can see, a very important factor in the Elastic Tree system is the method used in the optimizer for determining the subset of links and switches that should be active. In the paper, the authors chose a variety of methods for the optimizer: formal model, greedy bin-packer and topology-aware heuristic. The formal model is a multi-commodity flow formulation with binary variables for the power state of links and switches. The formal model minimizes the total network power consumption while satisfying all constraints. The constraints include link capacity, flow conservation, and demand satisfaction. Greedy

bin-packing chooses the leftmost path with sufficient capacity from the evaluated possible paths. The leftmost path refers to the path that is furthest to the left in a single layer. Assuming flows are perfectly dividable, topology-aware heuristic method splits flows and tries to pack every link to full utilization.

Here are two scenarios showing as examples of the output of the optimizer:

a) Scenario 1: Every server is transmitting traffic that must traverse the network core, as depicted in Figure 2-8. Because every server has traffic to network core, one aggregation switch and all access switches are needed for each Pod.

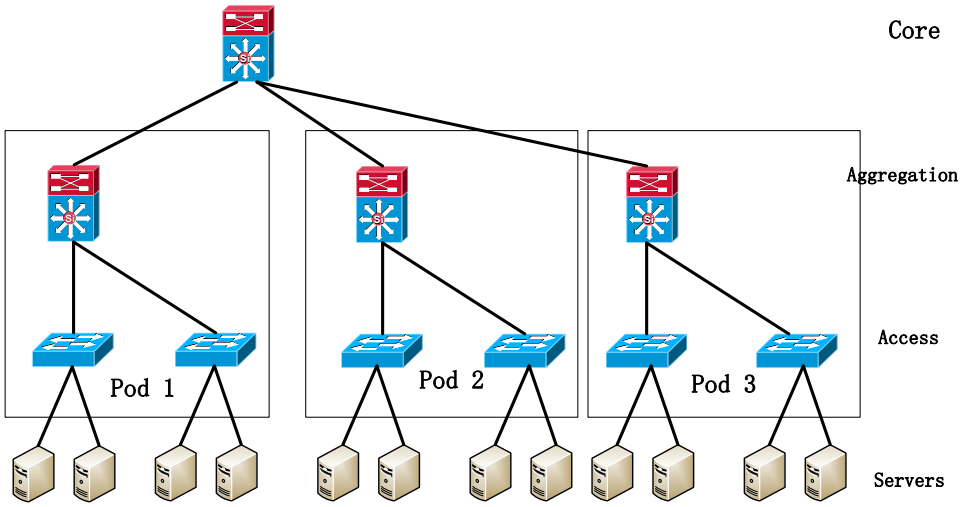


Figure 2-8. Elastic Tree System Scenario 1 [Heller10]

b) Scenario 2: Only Pod 1 and half of the servers in Pod 3 are in use, and servers in use in Pod 3 connect with the same access switch. Servers in Pod 1 communicate with servers in Pod 3. All the links and switches not shown in Figure 2-9 are not in use and are shut down.

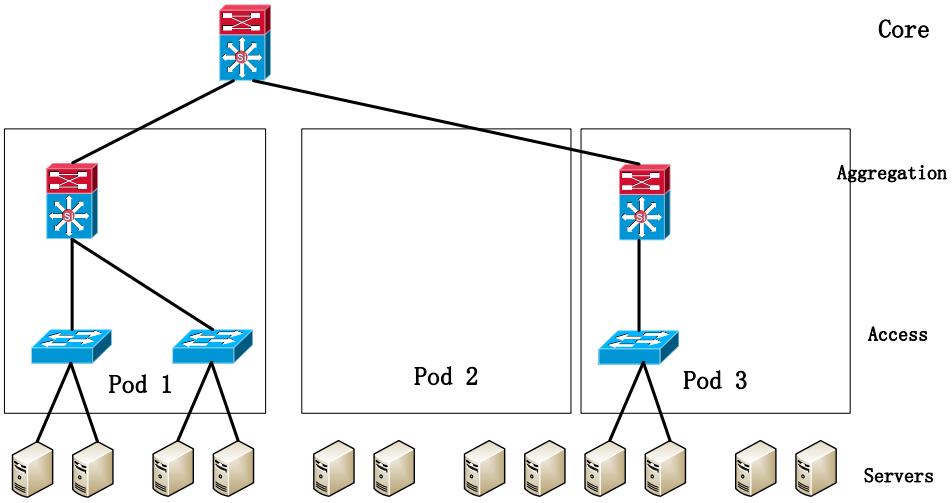


Figure 2-9. Elastic Tree System Scenario 2

The paper describes experiments that show that, on average, it is feasible for the Elastic Tree system to save 25%-40% of the network energy in datacenters.

The Elastic Tree system's strategy is similar to the proposed strategy in this thesis: minimizing the number of active network devices. The differences are as follow:

a) The Elastic Tree system is designed for the fat tree topology. The system proposed in this thesis is designed for the Spine-Leaf topology.

b) The Elastic Tree system considers aggregation switches and access switches as a whole when it makes decisions of turning on/off switches and links. The Elastic Tree system manages aggregation switches and access switches at the same time. In contrast, the system proposed in this thesis only manages Spine switches, which play the aggregation switches' role. The system proposed in this thesis only manages the control for Spine switches. Because a Leaf switch must be

active as long as a server connected with it is active, it is better to give the control for Leaf switches to a server controller (that has all information about all servers, including the state of each server), if the server controller is available.

c) The Elastic Tree system manages links and switches at the same time. The system proposed in this thesis does not have control for links.

One main problem with the Elastic Tree system is that computing the minimum network subset is computationally complex. The resulting network subset from the optimizer may not be the optimal one. For instance, as mentioned in the paper, the output of greedy bin-packing does not work well in some cases.

2.2.2 Saving Energy by Managing Port Rate

Besides managing the number of network devices, some other papers propose methods that save energy by managing port rate.

[Carrega12] focuses on saving energy by traffic merging. This paper presents the design of a hardware called traffic merge network, which is interposed between the connections from servers to switches. The use of the traffic merge network is based on the assumption of a low utilization of most ports in a switch. The traffic merge network is designed to be used in a network topology called Flattened Butterfly (FBFLY). It merges traffic from multiple links prior to feeding the merged traffic to the switch. The merged traffic enters the switch through several ports which are assigned maximum port rate, and

other ports are assigned lower port rate. Lower port rate consumes less than 40% of the power of the maximum port rate.

The traffic merge network proposed in this paper is interposed between the connections from servers to the switches. That means that the traffic from servers passes through the traffic merge network before it enters the ports of switches. Inside the traffic merge network, the N × N merge network ensures that if the port 1 of the switch is available, the traffic, no matter it comes from which server, will enter the switch from port 1. If port 1 is busy and port 2 is idle, the traffic will choose port 2 to enter the switch, and so on. As a result, only first several ports are always busy, having high utilization, and other ports are idle for most of time, having low utilization. To saving power consumed by switches, the authors suggest that maximum port rate should be assigned to those busy ports, and low port rate to non-busy ports.

In the paper, the simulation results demonstrate that with traffic merge network, even with an average load of 50% for each link, there are only 4 active ports that are characterized by the maximum rate, and the traffic merge network can help to save energy up to 22% - 49% for loads between 50% and 5% respectively.

This paper only discusses uplink traffic (traffic from servers to switches) and does not mention downlink traffic (traffic from switches to servers). Because the traffic merge network is interposed between a server and a switch, if there is no other link between the server and the switch, traffic from the switch to the server has to pass through the traffic

merge network too. However the traffic merge network does not demonstrate that it can receive the traffic from switches and send the traffic to the appropriate server. Besides, the FBFLY topology has a great difference with current 3-tier datacenter topology. Whether datacenter providers are willing to pay for the cost of changing their datacenter's topology is a important question.

[Abts10] proposes the FBFLY topology. It also exploits managing port rate to save energy consumed by the datacenter network. The authors combine load prediction with link's dynamic range to ensure that each link has the appropriate link speed to satisfy the traffic load.

Firstly, the paper demonstrates that the FBFLY topology consumes less power than a comparable folded-Clos with equivalent bisection bandwidth and same number of servers. Then the paper introduces a plesiochronous link to explain why one link can have multiple link speeds. The reason is that high-speed channels consist of multiple serialized lanes. These lanes operate at the same data rate, and a physical unit (phit) is striped across all the active lanes. For instance, a maximum link rate of 40Gbps includes four lanes which run at quad data rate (QDR) of 10Gbps each. It is also possible to operate the link with fewer lanes (e.g. 2 lanes for 20Gbps link rate), which can lead to lower data rate and lower power consumption. Finally, the paper indicates how to exploit a link's dynamic range and points out that the essence of exploiting the link's dynamic range is traffic load prediction (estimating the future bandwidth needs of each link periodically) and

reconfiguration of links' data rates to meet requirements. The traffic load prediction in this paper is a simple mechanism: the switch tracks the utilization of its each link over an epoch, and determines the link rate in the next epoch. There is a target utilization for each link. If the actual link utilization is higher than the target utilization, the link rate is doubled up to the maximum rate in the next epoch. Otherwise, the link rate is decreased to half of the current link rate. Another problem with this method is the reactivation latency caused by reconfiguring links. The paper gives two ways to tolerate the reactivation latency: one is to redirect the traffic to another path to destination servers, the other is to continue to allow the traffic to be routed to that link, and when output buffers fill up, drop packets (in a lossy network) or supply back-pressure (in a loss-less network). The second way means every packet routed to the reconfigured link has a longer latency.

The results in the paper illustrate that using link's dynamic range can achieve 36%, 17% and 15% of the power consumption at full link rate for average load of 23%, 6% and 5% respectively. The results also indicate that a network that always operates in the slowest link rate fails to keep up with the traffic load offered.

One of the disadvantages of the techniques proposed in this paper is that the traffic load prediction is too simple to predict the future traffic load accurately enough. The other one is that if the traffic load changes frequently and make the link rate changes every epoch, every packet routing to the link has an extra latency – the reactivation

latency. Therefore, some ways to prevent that the link rate oscillates persistently should be used with the link's dynamic range.

2.2.3 Saving Energy with Traffic Engineering

Some paper suggest methods that aggregate traffic to certain routes and put other route into sleep mode to save energy consumed by datacenter network.

Vasic et al. [Vasic11] proposed a system called Responsive Energy-Proportional Networks (REsPoNse). The system is designed for both Internet Service Providers (ISPs) and datacenters. REsPoNse computes energy-critical paths for the network, and then installs those paths into routing tables. Finally, the system uses online traffic engineering to deactivate and activate network elements on demands.

In the paper, the authors demonstrate that there are energy-critical paths which the majority of node pairs in the network routes their packets through. The main idea of the approach is that because there are energy-critical paths in networks, which can handle all traffic when the demand is low, REsPoNse can keep energy-critical paths always on and put other paths in sleep mode. When the traffic is higher than energy-critical paths' capacity, REsPoNse activates a few more paths, called on-demand paths, to take over the extra traffic. When there is a failure with a network element, REsPoNse provides a path called failover path to take over the traffic on the failed path. There are three types of paths in REsPoNse: always-on path, on-demand path and failover path. These three types of paths are all pre-computed off-line. To compute these three sets of paths, the input

includes the network topology, a power model of the network devices, and traffic matrix estimation (if available). REsPoNse uses an off-the-shelf solver [IBM] to pre-compute the three sets of paths off-line. Always-on paths are obtained by the solver using the off-peak traffic matrix estimation as input. On-demand paths are pre-computed with the peak-hour traffic matrix. Failover paths are a set of paths, in which all paths combined are not vulnerable to a single link failure. That means that if a failure happens, most of the paths in the set are not affected by the failure and still can work.

In the paper, the results illustrate that energy savings are around 30% and 42% for realistic traffic and topology, and REsPoNse can quickly and effectively use the always-on paths and the on-demand path at runtime.

One of the disadvantages of REsPoNse is that the on-demand paths may not work if a sub-path is shared by an always-on path and an on-demand path. Another drawback of REsPoNse is that REsPoNse still keeps some unneeded paths. Even if there is no traffic on the always-on path, REsPoNse keeps the always-on path active all the time. That means that always-on paths consume energy when no traffic passes through them.

Another approach presented in [C.Lee12] uses traffic engineering to achieve energy saving for datacenter networks. The authors used a traffic off-balancing algorithm which behaves oppositely to the load-balancing algorithm, to minimize the number of active network devices.

The strategy of the proposed algorithm is simple. The algorithm shifts traffic to

busy-state network devices. The main idea of the algorithm is to find a busy-state switch with enough available bandwidth for each traffic demand. At the initial state, all Ethernet switches are turned off. Then the algorithm sorts traffic demands in a queue. After having the traffic demands queue, the algorithm tries to distribute traffic demands one by one. The algorithm always tries to distribute the traffic demand to a busy switch as long as the busy switch has enough available bandwidth. If no switch has enough bandwidth for the traffic demand, the algorithm distributes the traffic to a busy switch and turns on a random switch to distribute remainder.

The first drawback of the traffic off-balancing algorithm is that it does not consider network performance. When the algorithm puts too much traffic on one switch, it may violate the SLAs due to possible higher delay it may cause. The second drawback of the algorithm is that that the authors assume that the traffic is perfectly dividable, which may not be true in reality. The third disadvantage is that in some cases, the energy that can be saved is limited. For example, in the fat tree topology with traffic uniformly distributed in all servers, only 50% of energy consumed by switches can be saved at most with the traffic off-balancing algorithm.

2.2.4 Saving Energy by Managing Network Device Configuration

Mahadevan et al., [Mahadevan10] proposed a system using network device configuration management to save energy. The authors proposed a system called Urja. Urja first collects required configurations and traffic information of the network switches and

predicts the power consumption of the switches. Then by analyzing the collected data, Urja lists various configuration and rewiring changes that can be made to the switches to make the network more energy proportional.

Urja has four components: the Measurement based switch power model, Web-based power profiler, Analysis engine and Power management engine. The Measurement based switch power model is a database that stores the power constants related to all switch models, linecard types, etc. The Web-based power profiler is responsible for obtaining the relevant configuration information from all switches in the network. The configuration information includes the switch chassis type, the number and type of active linecards, the number of active ports on each card, the administrative status of each port and the traffic flowing through each port. By using the configuration information and the database, the Web-based power profiler can predict the power consumption of the switches. The Analysis engine analyzes the acquired data to correlate the configuration information to the power consumed and the traffic flowing through switches. Then the Analysis engine generates a list of configuration changes that can be implemented to save energy. Finally, administrators can use the Power management engine to implement some of those suggested configuration changes to the switches.

The algorithm provides some possible configuration changes that can make the network more energy proportional. The possible configuration changes include: disabling unused ports, adapting Port rate, maximizing the number of active ports on a linecard and

using fewer switches by rewiring ports.

The results described in the paper show that the overall energy consumption can be reduced by up to 36% of the energy consumed by the network without the configuration changes.

Urja has some disadvantages. All the possible configuration changes suggest shutting down unused ports or devices without considering the scenario that the traffic increases dramatically. The paper does not present a method to activate those unused ports or devices to maintain network performance when the traffic increases suddenly and exceeds current network capacity. Another drawback of Urja is that it needs a lot of time to study the configuration and traffic of the network before making suggestions. Once the traffic in the network changes, Urja needs time to collect information again and then makes new suggestions.

2.2.5 Other Algorithms for Saving Energy

Algorithms presented in [Goudarzi12], [Liu12], [Taheri11] and [Salehi12] introduce new policies for task placement or virtual machine migration and placement, which can minimize the number of active servers.

Goudarzi and Pedram, [Goudarzi12] proposed an algorithm to generate multiple copies of a VM and place them on different physical servers. The algorithm determines the number of copies for each VM and the physical server for each copy. All the copies for a VM provide the same services. In other words, the algorithm splits a VM into

multiple copies and each copy needs less resource than the original VM. Multiple copies can help to make full use of servers. Therefore, with the algorithm, busy servers have high utilization and the number of underutilized servers increases. By turning off the underutilized servers, the algorithm can minimize the energy consumption.

Liu et al., [Liu12] proposed a greedy task scheduling algorithm which is used to schedule tasks with deadlines. The algorithm assigns tasks to the most energy-efficient servers. The algorithm prefers servers with higher computing capacity, because the energy consumption is proportional to the server's computing capacity. Therefore, the algorithm chooses the servers with higher computing capacity when a task needs to be scheduled. The simulation results demonstrate that the algorithm can minimize the average task queuing time and energy consumption.

Taheri and Zamanifar, [Taheri11] proposed an algorithm for VMs migration. The classic scheduling optimization chooses VMs from over-utilized servers and VMs from under-utilized servers and puts the VMs in a VM pool in the first step. In the second step, the classic scheduling optimization finds physical server for each VM in the VM pool. The proposed algorithm proposed changes the procedure of the classic scheduling optimization. The algorithm selects VMs from over-utilized servers and places them on appropriate physical servers in the first step. In the second step, the algorithm selects VMs from under-utilized servers and places them on appropriate physical servers. The algorithm can help reduce the number of under-utilized servers and avoid useless VMs

migration. By reducing the number of VMs migration, the algorithm helps to reduce the energy consumption of VMs migration.

Salehi et al., [Salehi12] proposed an algorithm that uses preemption of the lower priority requests to save energy. The paper assumes that there are two levels of priority: advance-reservation (AR) and best-effort (BE). To avoid starvation for BE requests, the algorithm defines a maximum average waiting time (MAWT) for BE requests. When an AR request arrives, if the risk of violating MAWT for BE request is low, the AR request acquires resource by preempting BE requests. Otherwise, the AR request acquires resource by switching on more resources.

Chapter 3: Energy Saving with Green Spine Switch Management

This chapter proposes a Green Spine Switch Management System (GSSMS) to minimize the number of active Spine switches in order to save energy consumed by a datacenter network. As mentioned in Chapter 2, this GSSMS is designed for use in a Spine-Leaf datacenter topology. A typical Spine-Leaf topology example is shown in Figure 1-3.

This chapter starts with an introduction of GSSMS framework, and follows on with a detailed description of the two modules: Routing Module and Spine Switch Controller. Finally, several scenarios that show how the system works are provided.

3.1 Green Spine Switch Management System Framework

GSSMS is a system for dynamically managing the energy consumed by Spine switches according to traffic load in datacenters. GSSMS comprises of three modules: Routing, Spine Switch Controller, Network Monitor and Power Control, as shown in Figure 3-1.

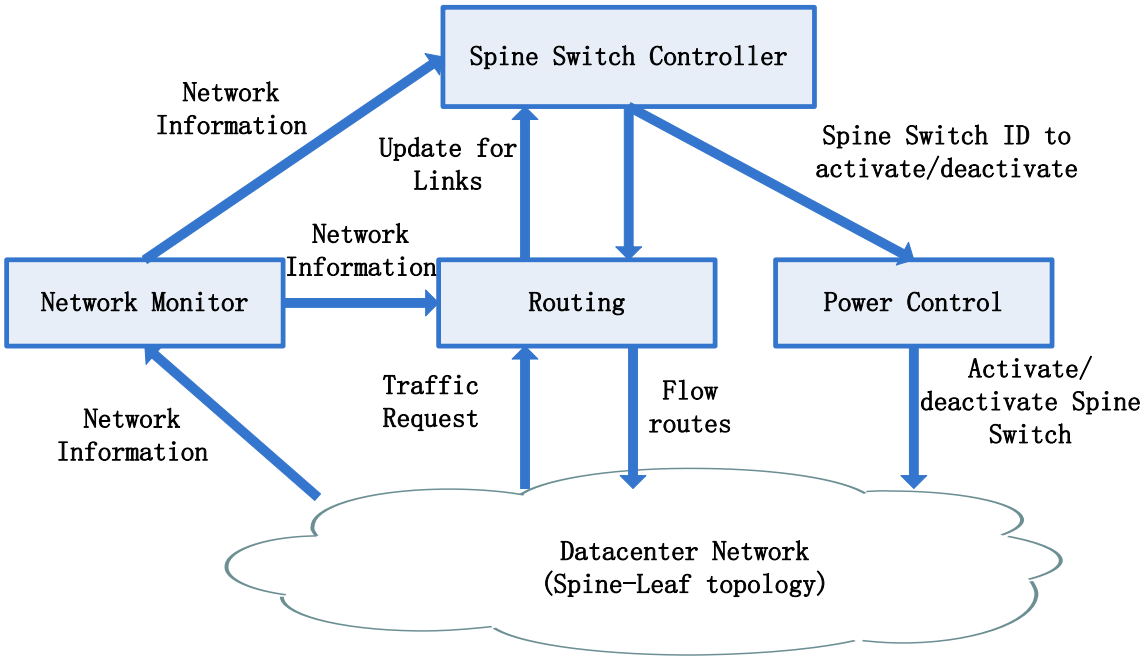


Figure 3-1. The Green Spine Switch Management System Diagram

The Routing module is responsible for choosing one Spine switch from the available active Spine switches to forward traffic that comes from Leaf switches. The Routing Module's policy is choosing the active Spine switch with the highest link utilization (the link connects the Spine switch and the Leaf switch) that has enough available bandwidth to handle the new flow. After the new flow is taken over by a Spine switch, the Routing Module sends the update of the link utilization to the Spine Switch Controller. The Spine Switch Controller module is used to monitor the utilizations of all links and all Spine switches, and make the decision of activating or deactivating a Spine switch and the decision of which Spine switch should be activated or deactivated as well. After the Spine Switch Controller decides to activate or deactivate a Spine switch, it sends the Spine switch ID to both the Routing Module and the Power Control module. The Routing

Module updates the state of Spine switches (active or inactive) based on the message from the Spine Switch Controller. The Network Monitor module is responsible for collecting network information and sending the network information to the Routing module and the Spine Switch Controller module. The Power Control module is in charge of toggling the power states (active or sleep) of Spine switches. After receiving the Spine switch ID from the Spine Switch Controller, the Power Control module puts the Spine switch into the appropriate power state.

The sequence diagram is shown in Figure 3-2. The Datacenter Network sends requests to the Routing Module. If the request asks for a route for a new flow, the Routing Module sends the route back to the Datacenter Network and sends a link utilization update message to the Spine Switch Controller. If the request indicates that a flow ends, the Routing Module sends the corresponding update message to the Spine Switch Controller. The Spine Switch Controller receives the update messages from the Routing Module and monitors the utilization of links. According to the utilization of links, the Spine Switch Controller makes the decision of activating or deactivating Spine switches, and sends the decision to the Routing Module and the Power Control.

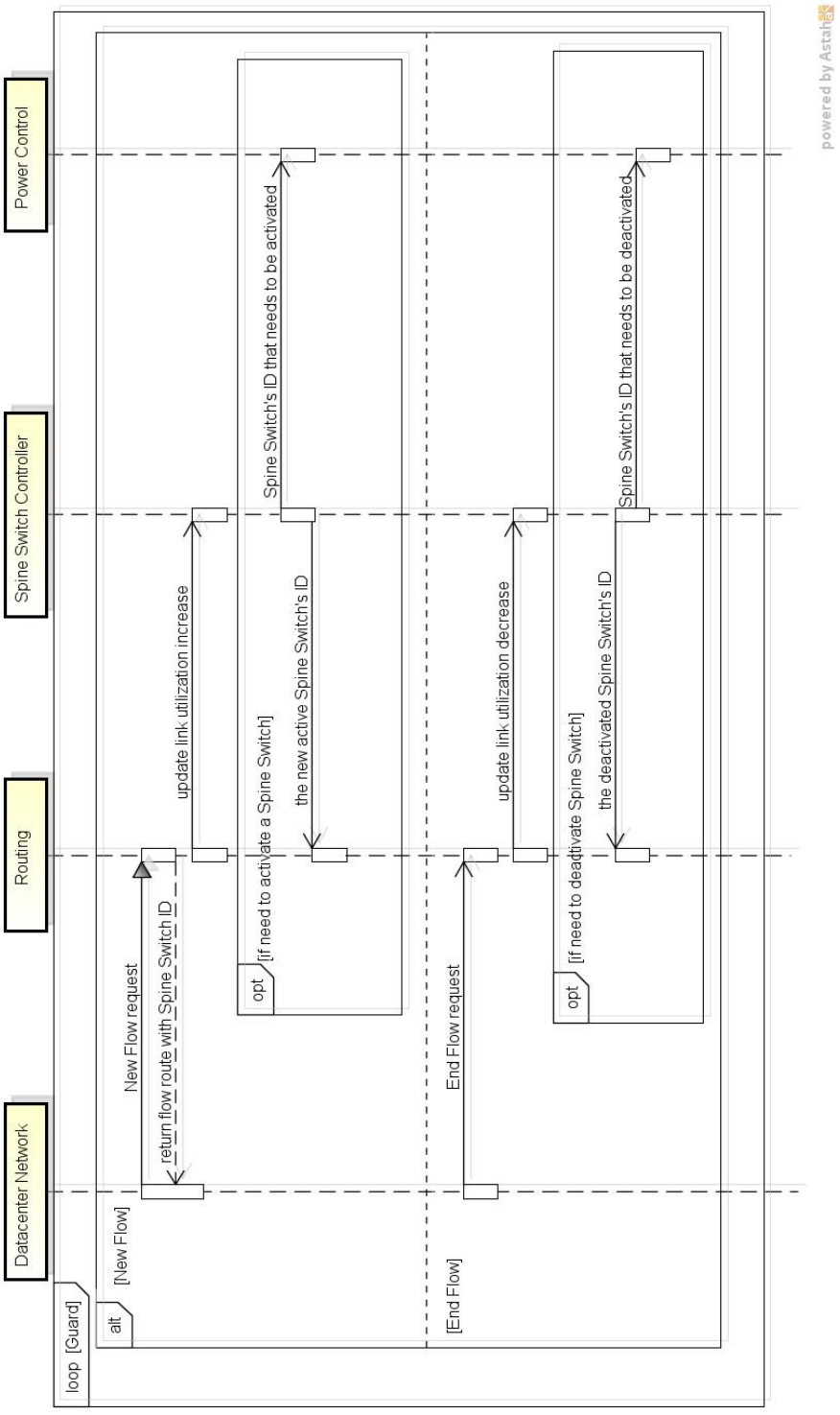


Figure 3-2. Sequence Diagram for the Green Spine Switch Management System

3.2 Routing Module

The Routing Module has two responsibilities: (i) choosing an active Spine switch for new flow and (ii) sending link utilization update to the Spine Switch Controller. The flow chart of the Routing Module is shown in Figure 3-3.

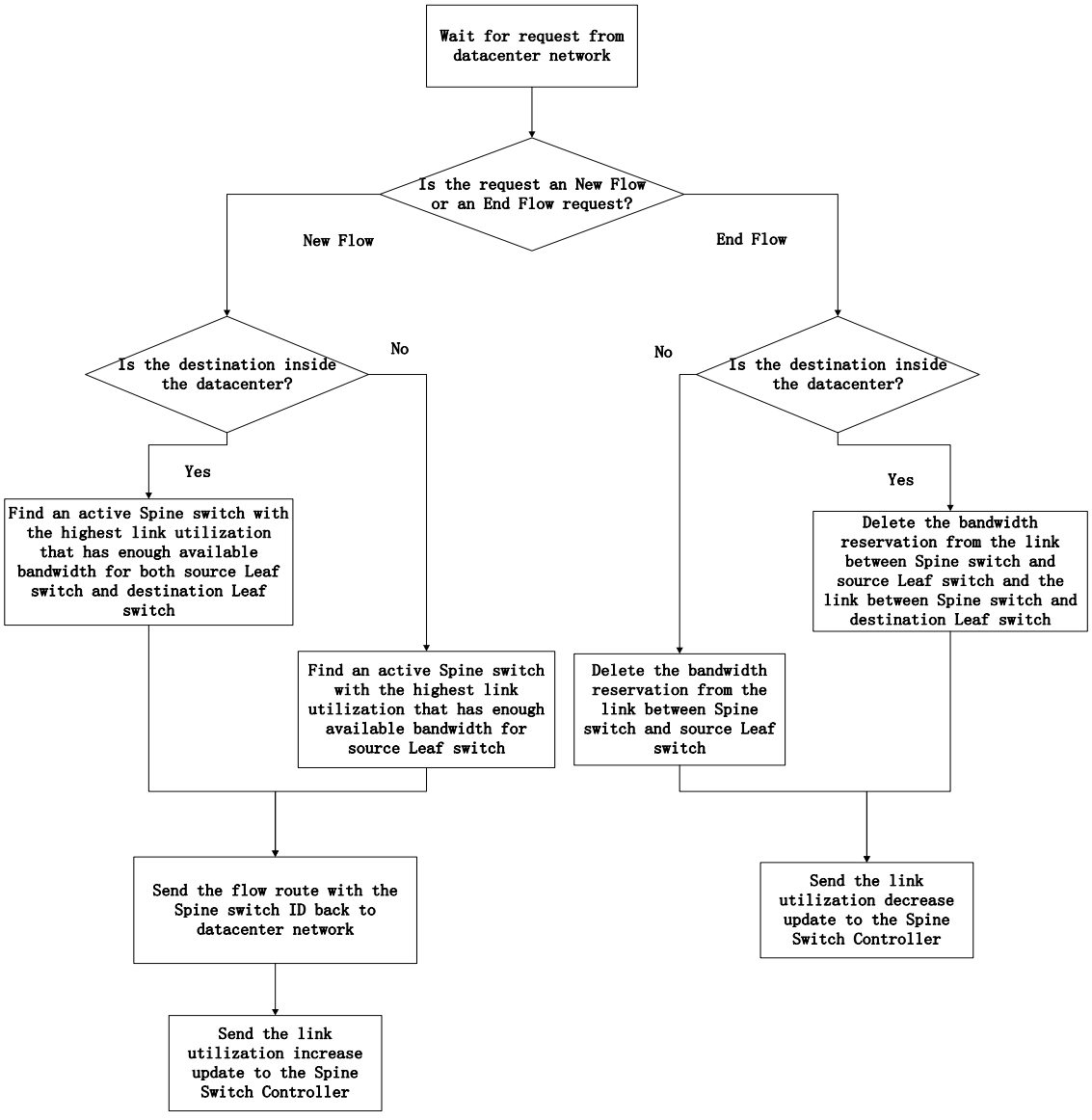


Figure 3-3. Flow Chart of the Routing Module

The requests from the datacenter network are of two types:

1. New Flow Request: the source Leaf switch asks for bandwidth reservation for a new flow.
2. End Flow Request: the source Leaf switch asks to release the bandwidth reservation for the finished flow.

After the Routing Module receives a request, no matter which type, it first needs to determine whether the flow's destination is outside or inside the datacenter. These two kinds of requests should be treated in different ways. First, flow with a destination outside the datacenter only needs bandwidth reservation on one link between a Spine switch and a Leaf switch. Second, flow with a destination inside the datacenter needs bandwidth reservation on two links between a Spine switch and two Leaf switches. Figure 3-4 shows two flows from server D: one is flow D-B-A going outside of the datacenter, and the other one is flow D-B-A-C-E which stays inside of the datacenter. As shown in the figure, flow D-B-A only needs to acquire bandwidth reservation on link B-A; however, flow D-B-A-C-E needs to obtain bandwidth reservation on two links (e.g. link B-A and link A-C).

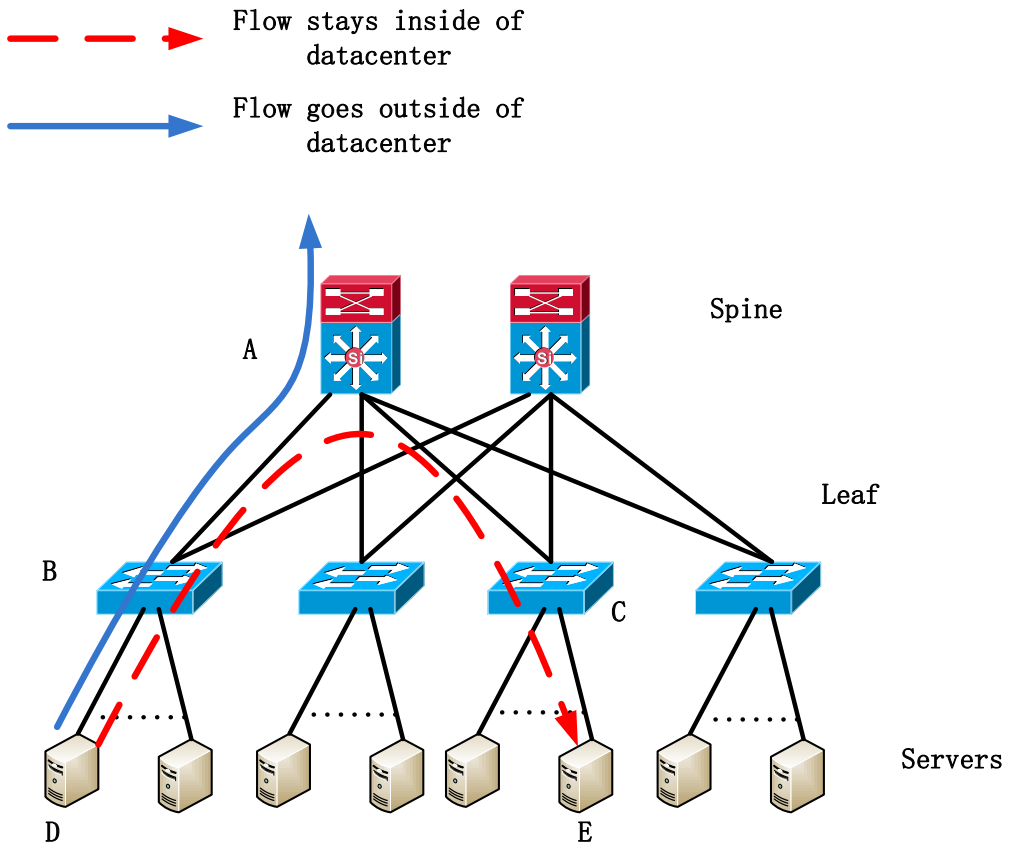


Figure 3-4. Flow Going Outside of Datacenter and Flow Staying Inside of Datacenter

When the request is a New Flow request, after knowing the destination of the flow, the Routing Module can decide which active Spine switch should be allocated to the new flow. The allocation policy is simple; inside the Routing Module, for each Leaf switch, all utilizations of links are sorted in non-increasing order in a queue. If the new flow goes outside of the datacenter, the Routing Module just needs to pick the first Spine switch that has enough available bandwidth in the link utilization queue. Otherwise the Routing Module needs to choose the first Spine switch that has enough available bandwidth for two links; one is the link between the source Leaf switch and the Spine switch, and the

other one is the link between the Spine switch and the destination Leaf switch. Finally, the Routing Module sends the flow route with the Spine switch ID back to the datacenter network, and sends a message to inform the Spine Switch Controller that the utilization of the link between source Leaf switch and the chosen Spine switch has increased.

When the request is an End Flow request, besides the destination of the flow, the Routing Module can also acquire information about the Spine switch allocated for the flow. Therefore, if the flow's destination is out of the datacenter, the Routing Module only needs to remove the bandwidth reservation on the link between the source Leaf switch and the Spine switch. Otherwise, the Routing Module needs to release the bandwidth reservation on the link between the source Leaf switch and the Spine switch and the bandwidth reservation on the link between the Spine switch and the destination Leaf switch. Finally, the Routing Module sends a link utilization update to inform the Spine Switch Controller that the link utilization has decreased.

3.3 Spine Switch Controller

The Spine Switch Controller module is a key component of GSSMS. In this section, a detailed description of the control algorithm and the important controller parameters is provided, which is then followed by a presentation of the control flow for the Spine Switch Controller.

3.3.1 Control Algorithm

The Control Algorithm has two components: an Algorithm for Activating a Spine Switch

and an Algorithm for Deactivating Spine Switches. The parameters used in these components are described before explaining the operations of these components.

The parameters of the Control Algorithm are:

a) The high First Utilization Threshold (FUT-H): works with SUT-H, CT-H and Tda to make the decision of activating a Spine switch.

b) The low First Utilization Threshold (FUT-L): works with SUT-L, CT-L and Tdd to make the decision of deactivating Spine switches.

c) The high Control Threshold (CT-H): works with FUT-H, SUT-H and Tda to make the decision of activating a Spine switch.

d) The low Control Threshold (CT-L): works with FUT-L, SUT-L and Tdd to make the decision of deactivating Spine switches.

e) The high Second Utilization Threshold (SUT-H): works with FUT-H, CT-H and Tda to make the decision of activating a Spine switch.

f) The low Second Utilization Threshold (SUT-L): works with FUT-L, CT-L and Tdd to make the decision of deactivating Spine switches.

g) The Time Duration for Activating (Tda): works with FUT-H, SUT-H and CT-H to make the decision of activating a Spine switch.

h) The Time Duration for Deactivating (Tdd): works with FUT-L, SUT-L and CT-L to make the decision of deactivating Spine switches.

3.3.1.1 Algorithm for Activating a Spine Switch

FUT-H, CT-H, SUT-H and Tda are used for activating a Spine switch. For each Leaf switch, when the link utilizations of a given number of links connected with the given Leaf switch exceed FUT-H, and remain higher than SUT-H for the given time duration Tda, a Spine switch is activated. The logic is presented in Algorithm 1.

Algorithm 1: Increasing the number of active Spine switches

1. $Na = \lceil$number of active Spine switches $\times$ *CT-H*$\rceil$
2. **Foreach** Leaf switch
3. **If** the link utilization of *Na* links connected with the given Leaf switch $\geq$ *FUT-H* && the link utilization of *Na* links connected with the given Leaf switch $\geq$ *SUT-H* for a duration $\geq$ *Tda*
4. Activate a Spine switch
5. **End If**
6. **End Foreach**

3.3.1.2 Algorithm for Deactivating Spine Switches

FUT-L, CT-L, SUT-L and Tdd are used for deactivating Spine switches. For each Leaf switch, when the link utilizations of a given number (Nd) of links connected with the given Leaf switch fall below FUT-L, the Leaf switch's ID is put into a list *timingmap*. When the size of *timingmap* is equal to the number of Leaf switches, it indicates that all Leaf switches have Nd links with utilization below FUT-L. If the size of *timingmap* is equal to the number of Leaf switches for the given time duration Tdd, a number of Spine switches are deactivated. The logic is presented in Algorithm 2.

Algorithm 2: Decreasing the number of active Spine switches

1. $Nd = \lceil$number of active Spine switches $\times$ *CT-L*$\rceil$
2. **Foreach** Leaf switch
3. **If** the link utilization of *Nd* links connected with the given Leaf switch $\leq$ *FUT-L*
4. Add the Leaf switch's ID into *timingmap*
5. **End If**
6. **End Foreach**
7. **If** *timingmap*.size() == number of Leaf switches
8. Start timing *T*
9. **Foreach** Leaf switch
10. **If** the link utilization of *Nd* links connected with the given Leaf switch $\geq$ *SUT-L* and $T \leq Tdd$
11. Remove the Leaf switch's ID from *timingmap*
12. Stop timing *T*
13. **End If**
14. **End Foreach**
15. **End If**
16. **If** *timingmap*.size() == number of Leaf switches for $T \geq Tdd$
17. Deactivate at least *Nd* Spine switches
18. **End If**

3.3.2 Controller Parameters

In this section, the detailed description of how the thresholds and time durations work is provided.

First Utilization Thresholds (FUTs) and Control Thresholds (CTs)

To decide when a Spine switch should be activated or deactivated, the Spine Switch Controller uses two pairs of thresholds. One pair of thresholds is for the utilization of

links, called FUTs. The other pair of thresholds is for the number of links with utilization beyond FUTs, called CTs. In both pairs of thresholds, there is one high threshold for activating a Spine switch and one low threshold for deactivating Spine switches. FUTs are the thresholds for each link between a Spine switch and a Leaf switch. When the utilization of one link exceeds FUT-H, it may be caused by an increase in traffic. The traffic increase trend can lead to the situation that the link may not have enough available bandwidth for new flows in the near future. When the utilization of one link is below FUT-L, it may be caused by a decrease in traffic. The traffic decrease trend can lead to the situation where the link's utilization may decrease to a lower value in the near future. CTs are used to demonstrate the conditions for activating and deactivating a Spine switch. For each Leaf switch, when the utilizations of Na links have exceeded FUT-H, a Spine switch should be activated. CT-H is responsible for determining the value of Na used in the activating of a Spine switch. When the utilizations of Nd links have been under FUT-L, a Spine switch should be deactivated. CT-L decides the value of Nd used in the deactivating of Spine switches.

With these two pairs of thresholds, the policies for activating and deactivating slightly differ. The Spine Switch Controller takes one Leaf Switch as one traffic source. As demonstrated in Algorithm 1, the activating policy makes decision to activate a Spine switch based on the traffic load from each source. That means if the traffic load from one source demonstrates that one more active Spine switch is needed, the Spine Switch

Controller will decide to activate a Spine switch, even though the traffic from other sources do not indicate the need of a new active Spine switch. There is a scenario shown in Table 3-1. In this scenario, FUT-H is 80%, and CT-H is 90%. Currently, there are five active Spine switches and six Leaf switches in the datacenter network. That means when utilizations of all five (calculated by Equation 3.1) links that connect to the same Leaf switch exceed FUT-H, the Spine Switch Controller will make the decision to activate one Spine switch

$$\begin{aligned} \text{Na} &= \lceil \text{the number of active Spine switches} \times \text{CT} - \text{H} \rceil \\ &= \lceil 5 \times 90\% \rceil \\ &= 5 \end{aligned} \tag{3.1}$$

As shown in the Table 3-1, only the link utilizations of Leaf Switch 3 show that one more active Spine switch is needed because the utilizations of all five links that connect to Leaf Switch 3 are equal to or higher than FUT-H of 80%. In this scenario, the Spine Switch Controller will activate a Spine switch, and disregard about the fact that traffic from other Leaf switches do not need a new active Spine switch.

Table 3-1. Scenario for Activating Policy

Link Utilizations (%)	Spine Switch 1	Spine Switch 2	Spine Switch 3	Spine Switch 4	Spine Switch 5
Leaf Switch 1	70%	80%	90%	60%	10%
Leaf Switch 2	90%	80%	90%	50%	15%
Leaf Switch 3	82%	90%	80%	85%	88%
Leaf Switch 4	60%	10%	0%	0%	0%
Leaf Switch 5	90%	70%	12%	0%	0%
Leaf Switch 6	70%	90%	30%	0%	0%

In contrast, as demonstrated in Algorithm 2, the deactivating policy makes decision to deactivate a Spine switch based on the traffic from all sources. That means only when traffic from all sources demonstrates that they all have at least a given number of idle links, the Spine Switch Controller will make the decision to deactivate Spine switches. Table 3-2 shows a scenario of deactivating policy. In this scenario, FUT-L is 20%, and CT-L is 30%. Assuming that there are five active Spine switches and six Leaf switches in the datacenter network, when every Leaf switch has two (calculated by Equation 3.2) links with utilization below FUT-L, the Spine Switch Controller can make the decision to deactivate Spine switches.

$$\begin{aligned} \mathrm{Nd} &= \lceil \text{the number of active Spine switches} \times \mathrm{CT} - \mathrm{L} \rceil \\ &= \lceil 5 \times 30\% \rceil \\ &= 2 \end{aligned} \tag{3.2}$$

From Table 3-2, we can see that each Leaf switch has at least two links with utilization lower than FUT-L of 20%. Therefore, the Spine Switch Controller can deactivate two Spine switches in this case. Another scenario for the deactivating policy occurs with the same FUT-L of 20%, but with a CT-L of 20%, instead of 30% from the previous scenario. This scenario also has the link utilizations in Table 3-2. In this scenario, the deactivating policy only asks for one (instead of two) link utilization lower than FUT-L, even though all Leaf switches have two links with the utilization lower than FUT-L. In this scenario, the Spine Switch Controller will choose to deactivate two Spine

switches because every Leaf switch has two links with utilization below FUT-L.

Table 3-2. Scenario for Deactivating Policy

Link Utilizations (%)	Spine Switch 1	Spine Switch 2	Spine Switch 3	Spine Switch 4	Spine Switch 5
Leaf Switch 1	80%	80%	60%	0%	10%
Leaf Switch 2	90%	80%	90%	15%	1%
Leaf Switch 3	72%	90%	80%	5%	0%
Leaf Switch 4	60%	10%	0%	0%	0%
Leaf Switch 5	90%	70%	12%	0%	0%
Leaf Switch 6	70%	90%	30%	0%	0%

Time Duration and Second Utilization Thresholds (SUTs)

Just the activating and deactivation policies described above are not enough for the Spine Switch Controller. Because when the traffic load changes frequently, the number of active Spine switches may fluctuate, as shown in Figure 3-5. This fluctuation can make the datacenter network unstable because every time the number of active Spine switches changes, all switches in the network have to learn the topology of the network again. A system in which the change in the number of Spine switches is not very frequent (e.g., the time between two changes is longer than half an hour) is thus desirable. The desired output of the Spine Switch Controller with the same traffic load in Figure 3-5 is shown in Figure 3-6. Unlike the fluctuations in Figure 3-5, the number of active Spine switches demonstrates relative stability.

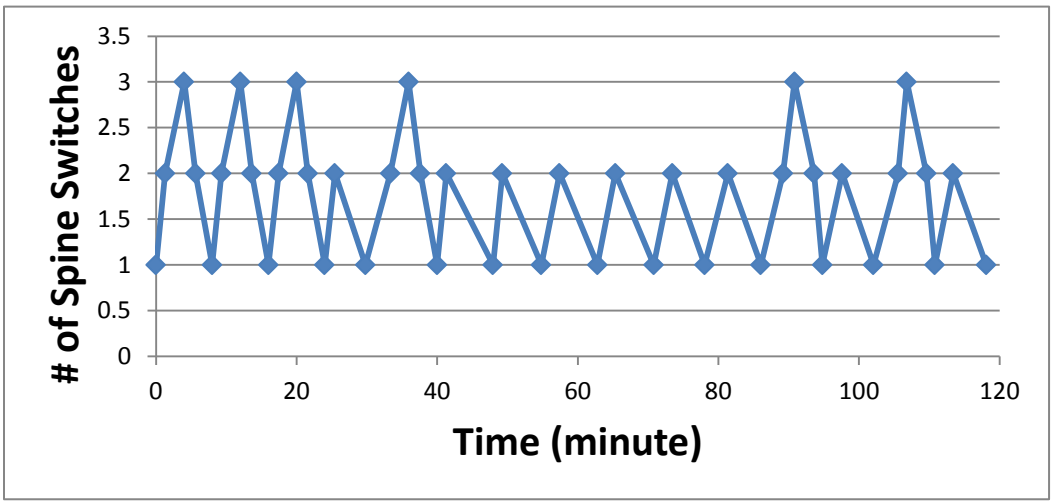


Figure 3-5. Fluctuation of the Number of Active Spine Switches

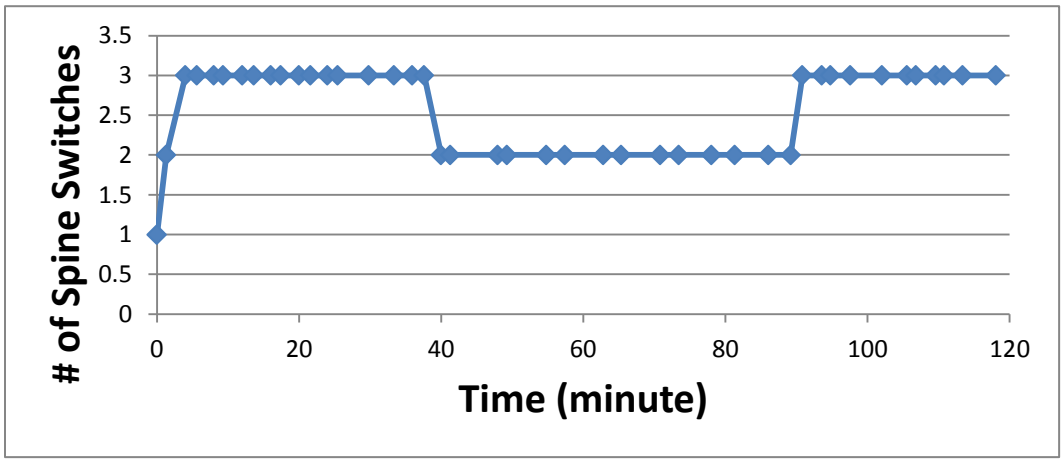


Figure 3-6. Desired output of the Green Spine Switch Management System

To avoid persistent vibration and instability of the number of active Spine switches, the Spine Switch Controller uses time durations (Tda and Tdd) and another pair of associated thresholds (SUTs). The time durations are used to filter the instant traffic changes. Once the number of link utilizations higher than FUT-H reaches the given number Na, the Spine Switch Controller no longer decides to activate a Spine switch

immediately. Instead, the Spine Switch Controller starts timing. For the activating algorithm, within the time duration Tda, if the traffic load remains high and does not decrease significantly, the Spine Switch Controller will make the decision of activating an additional Spine switch. This ensures that the number of Spine switches is not changed due temporary spike in traffic. A traffic decrease trend has the same problem. Therefore the Spine Switch Controller uses SUTs to permit the traffic to decrease slightly for a traffic increase trend and permit the traffic to increase slightly for a traffic decrease trend. SUT-H must be lower than FUT-H, and SUT-L must be higher than FUT-L. As a result, the Spine Switch Controller works as follow:

As demonstrated in Algorithm 1, when the number of links with utilization that is equal to or higher than FUT-H reaches the given number Na, the Spine Switch Controller starts timing. Within Tda, the Spine Switch Controller checks whether the number of links with utilization that is equal to or higher than SUT-H reaches Na or not. If the number of the links is never below Na, the Spine Switch Controller will decide to activate a Spine switch when the time duration Tda is up.

For the traffic decrease trend, the policy works in a similar way, except the time duration Tdd that is used is longer than Tda to make it longer to make the decision of deactivating.

Here are two scenarios:

1. Scenario 1:

As shown in Figure 3-7, when the number of links with utilization that is equal to or higher than FUT-H reaches the given number Na at time point Ta, the timer for activating a Spine switch starts. After the time point Ta, the number of links with utilization that is equal to or higher than SUT-H does not fall below Na during Tda. Therefore, after Tda, a Spine switch will be activated.

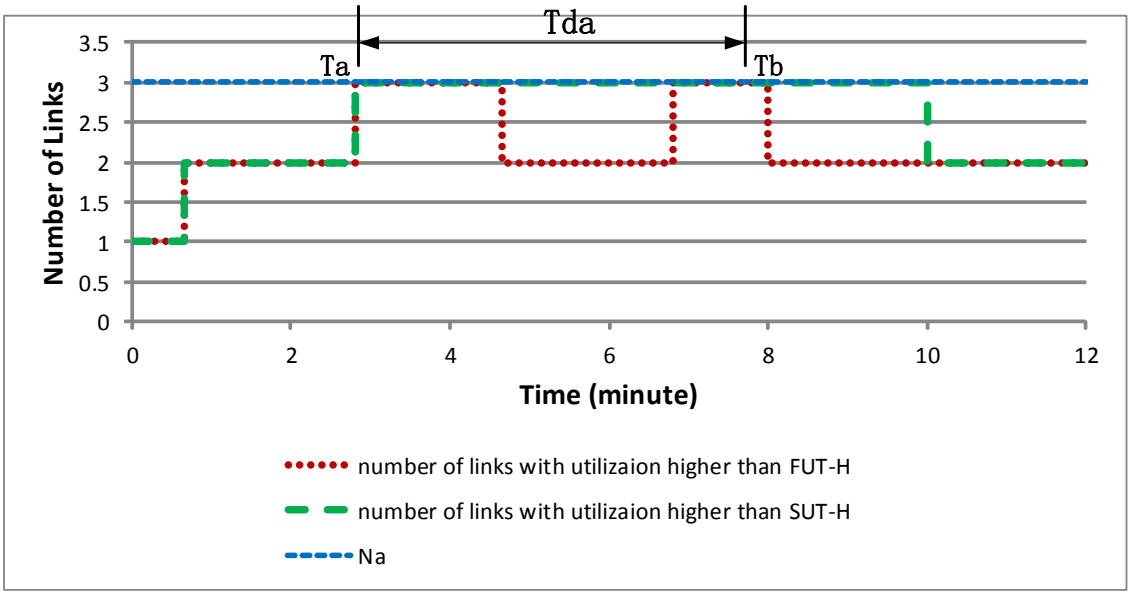


Figure 3-7. Scenario One for the Duration Policy

2. Scenario 2:

In Figure 3-8, after the timer starts, the number of links with utilization that is equal to or higher than SUT-H decreases below Na. Therefore, the timer stops at the time point Tb. Because the number of links with utilization that is equal to or higher than FUT-H never reach Na after Tb, timer for activating a Spine switch does not start again.

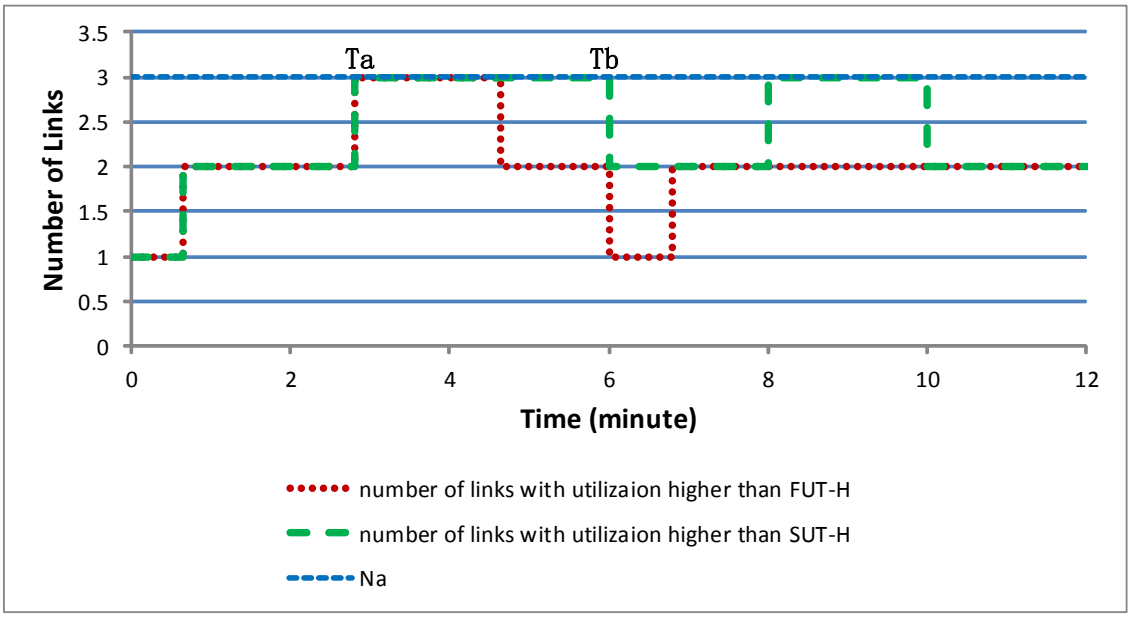


Figure 3-8. Scenario Two for the Duration Policy

Because of the use of the time durations (Tda and Tdd), GSSMS has a specific deployment: a backup Spine switch, shown in Figure 3-9. Because the Spine Switch Controller does not activate a Spine Switch immediately when the traffic load increases, it is possible that during the respective time duration, the traffic load exceeds the capacity of the datacenter network. In this scenario, there is no active Spine switch to take over the extra traffic; and the datacenter network will drop all extra traffic, which is unacceptable. Therefore, GSSMS chooses one active Spine switch as the backup Spine switch, which is always on. The Backup Spine Switch's responsibility is to take over traffic when there is no other active Spine switch with enough available bandwidth is available.

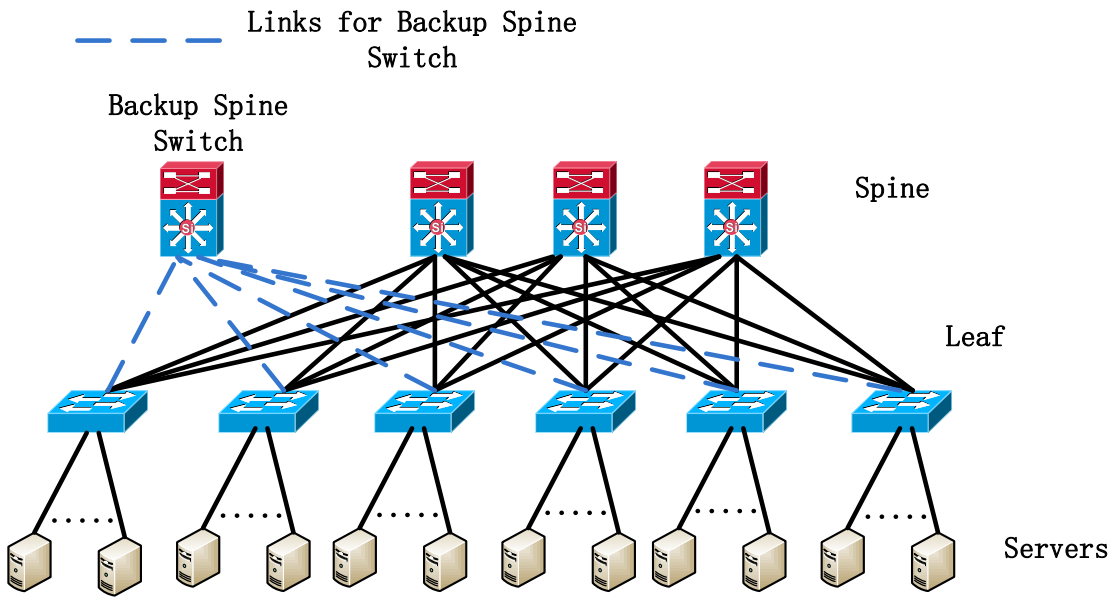


Figure 3-9. Backup Spine Switch

3.3.3 Control Flow of the Spine Switch Controller

Figure 3-10 and Figure 3-11 demonstrate the control flow of the Spine Switch Controller. As we can see, the control flow can be divided into two main parts: (i) making decision of starting timing and (ii) checking duration.

3.3.3.1 Making Decision of Starting timing

There are two steps for making decision of starting timing.

a) Step 1: When the timer is OFF, the Spine Switch Controller decides if the timer should start using the link utilization updates.

   After receiving a link utilization update, the first thing that the Spine Switch Controller does is determining whether it is a link utilization increase update or a

link utilization decrease update. Link utilization increase update means a new flow has come, and link utilization decrease update means a flow has finished. A new flow coming indicates that it may trigger the timer for activating a Spine switch. A flow finishing may trigger the timer for deactivating a Spine switch. Therefore, the Spine Switch Controller checks different thresholds according to the type of link utilization update. The source Leaf switch ID comes with the link utilization update. When the update is a link utilization increase update, the Spine Switch Controller needs to check the number of links with utilization that is equal to or higher than FUT-H for the source Leaf switch to determine if the timer for activating a Spine switch needs to start. If the number of links reaches Na, the timer for activating a Spine switch starts. Because any Leaf switch can trigger the timer for activating, Leaf switches have separate timers. The Spine Switch Controller records the start time for them individually. In contrast, for the timing for deactivating Spine switches, there is only one timer for the Spine Switch Controller. The Spine Switch Controller has a *timingmap* (shown in line 4 in Algorithm 2, which is a list of Leaf switch IDs) to record the Leaf switches that satisfy the requirement of deactivating Spine switches. Only when all Leaf switches' IDs are in *timingmap*, the Spine Switch Controller starts the timer for deactivating Spine switches.

b) Step 2: The Spine Switch Controller updates the information related with the

timer for deactivating Spine switches and decides if the timer should be OFF.

After step 1 is done, if the update is a link utilization increase update, the Spine Switch Controller checks if the number of the links reaches Nd for the source Leaf switch. Otherwise the Spine Switch Controller checks if the number of the links reaches Na for the source Leaf switch. If the number of the links does not reach Nd or Na, the respective timer stops. After confirming the timer's state, the Spine Switch Controller checks the time duration, which is presented in the next section.

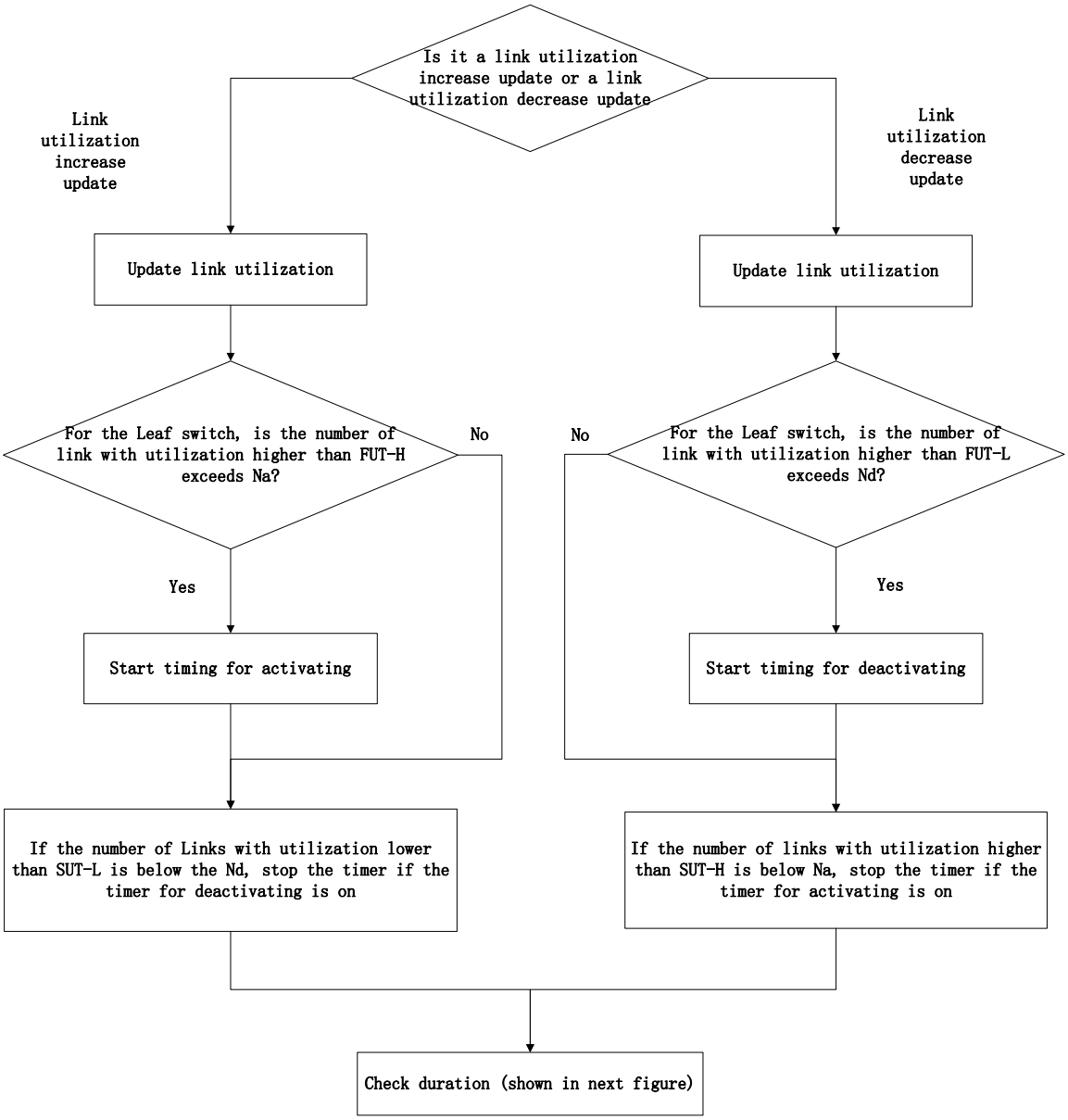


Figure 3-10. Control Flow for Spine Switch Controller

3.3.3.2 Checking Time Duration

The control flow of checking time duration is shown in Figure 3-11. This part has no relationship with the link utilization updates described in Figure 3-10. No matter which type of link utilization update is received, this part will be executed in the same way. As

shown in Figure 3-11, the first step of checking time duration is to find out whether the timer for activating a Spine switch is ON or the timer for deactivating Spine switches is on. Each of these outcomes leads to a separate branch in the flow chart:

a) If the timer for activating a Spine switch is on, the Spine Switch Controller will check the recorded start time (if available) for each Leaf switch individually. Once one Leaf switch's start time demonstrates that the time duration Tda is exceeded, the Spine Switch Controller will decide to add a new active Spine Switch. As shown in the flow chart in Figure 3-11, the Spine Switch Controller has to figure out the Spine switch's ID as well. Before explaining how to acquire the Spine switch's ID, the Waiting for Deactivating queue should be explained. In GSSMS, when the Spine Switch Controller decides to deactivate a Spine switch with non-zero utilization, it does not deactivate the Spine switch immediately, but puts the Spine switch into the Waiting for Deactivating queue to wait for all the flows on that Spine switch to finish. When the Spine switch is in the Waiting for Deactivating queue, the Routing Module treats it as an inactive Spine switch, and does not locate flows on it. However, the Power Control module treats the Spine switch as an active Spine switch because the Spine switch has outstanding flows to handle. The Spine switches in the Waiting for Deactivating queue will not be deactivated until their utilization is zero. Therefore, when the datacenter network needs additional active Spine Switches while the Waiting for Deactivating queue

is not empty, the Spine switches in the Waiting for Deactivating queue are the best candidates because they are still active. Otherwise, when the queue is empty, the Spine Switch Controller randomly chooses a Spine switch to activate.

b) If the timing for deactivating Spine switches is on, the Spine Switch Controller will check the timer for deactivating. If the time duration Tdd is exceeded, the Spine Switch Controller will check the number of links with utilization that is equal to or below FUT-L for each Leaf switch. Then the Spine Switch Controller chooses the minimum number from these numbers of links as the number of Spine switches that should be deactivated. For example, if the link utilizations are as those shown in Table 3-2 and FUT-L is 20%, Leaf Switch 4 and 5 have three links with utilization lower than 20% respectively, and other Leaf switches have two links with utilization lower than 20% respectively. In this scenario, the Spine Switch Controller chooses two as the number of Spine switches that should be deactivated. Then the Spine Switch Controller chooses the Spine switches with lowest utilization to deactivate. As mention earlier, if those Spine switches' utilizations are non-zero, the Spine switches will be put into the Waiting for Deactivating queue, otherwise their ID will be sent to the Routing Module and the Power Control module. Finally, if the Waiting for Deactivating queue is not empty, the Spine Switch Controller will check the utilization of Spine switches in that queue. If there is no flow on those Spine switches, the Spine Switch

Controller will deactivate them.

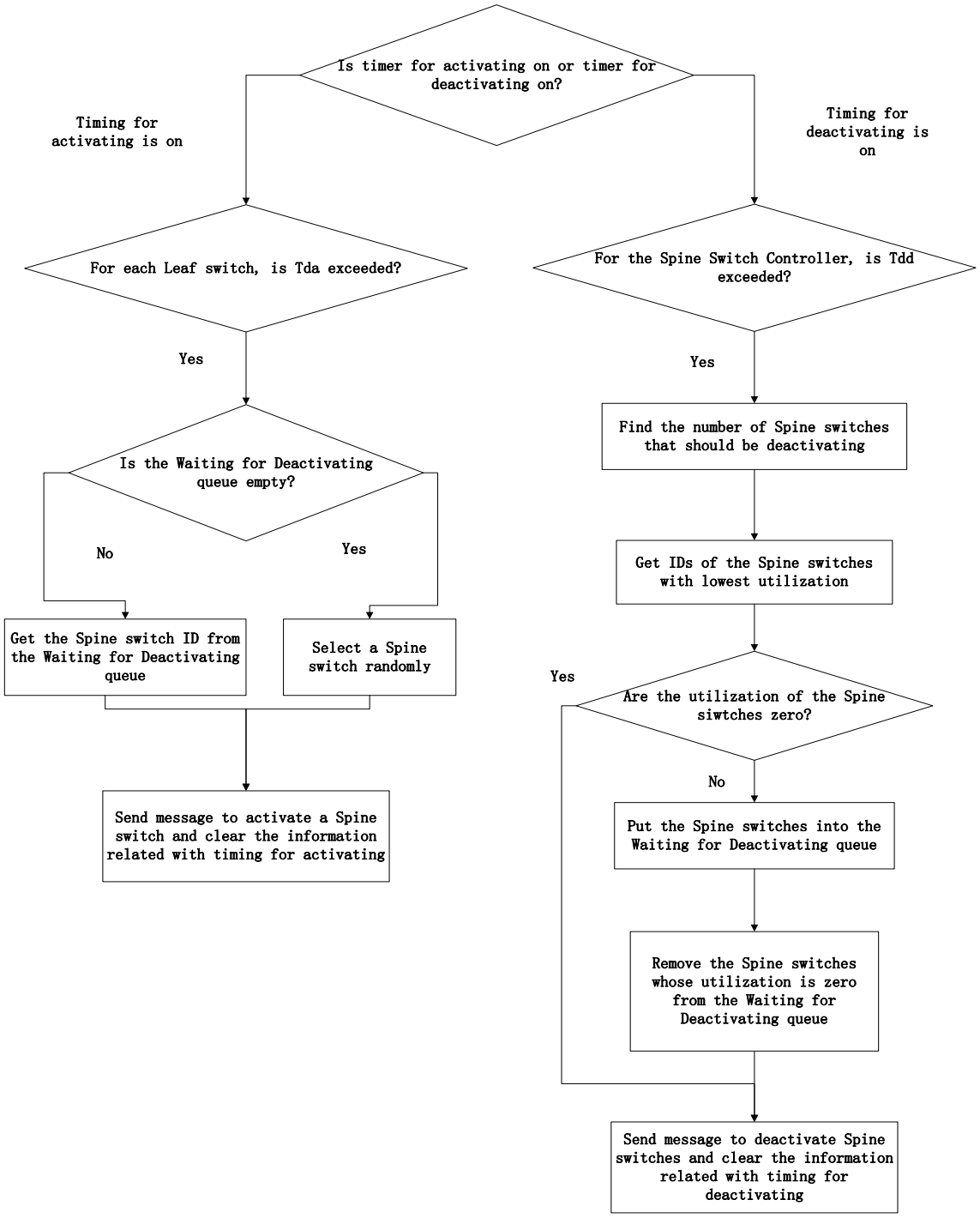


Figure 3-11. Workflow for Checking Duration

3.4 Scenarios of GSSMS

This section presents a set of representative scenarios that illustrate how the proposed algorithms adjust the number of active Spine switches based on the utilizations. All scenarios in this section are based on the same datacenter network (Spine-Leaf topology with four Leaf switches), the same capacity of links between a Spine switch and a Leaf switch (10Gbps) and the same parameter values for GSSMS. For the scenarios that will activate a Spine switch, the current number of active Spine switches is one, and for the scenarios that will deactivate a Spine switch, the current number of active Spine switches is two. A list of scenarios is presented:

a) Scenario 1: Scenario for Starting Timer for Activating a Spine Switch

b) Scenario 2: Scenario for Checking Time Duration for Activating a Spine Switch

c) Scenario 3: Scenario for Interrupting the Timer for Activating a Spine Switch

d) Scenario 4: Scenario for Starting Timer for Deactivating a Spine Switch

e) Scenario 5: Scenario for Checking Time Duration for Deactivating a Spine Switch

f) Scenario 6: Scenario for Interrupting the Timer for Deactivating a Spine Switch

The parameter values used for all aforementioned six scenarios are as follow:

a) FUTs: FUT-H is 80%, FUT-L is 20%

b) CTs: CT-H is 90%, CT-L is 20%

c) SUTs: SUT-H is 60%, SUT-L is 40%

d) Tda: 1 second

e) Tdd: 2 seconds

Scenario 1: Scenario for Starting Timer for Activating a Spine Switch

Link utilizations are shown in Table 3-3. The timer of activating a Spine switch is initially OFF.

Table 3-3. Link Utilizations for Scenario 1

Link Utilizations (%)	Spine Switch 1	Backup Spine Switch
Leaf Switch 1	76%	0%
Leaf Switch 2	60%	0%
Leaf Switch 3	78%	0%
Leaf Switch 4	70%	0%

A new flow comes from Leaf Switch 3 with flow rate 400Mbps. The Routing Module will choose Spine Switch 1 to take over this new flow because Spine Switch 1 has enough available bandwidth and has the highest utilization. After the link takes over this new flow, the utilization of the link between Spine Switch 1 and Leaf Switch 3 increases to approximately 81%. The utilization of the link is higher than FUT-H. Each Leaf switch has a timer. Therefore the Spine Switch Controller sets the timer ON for Leaf Switch 3 to indicate that the timer for activating a Spine switch has started. However because other Leaf switches' link utilizations do not exceed FUT-H, the timers for other Leaf switches are still OFF.

Scenario 2: Scenario for Checking Time Duration for Activating a Spine Switch

Link utilizations are shown in Table 3-4. The Leaf Switch 1's timer of activating a Spine

switch is initially on, but the time duration Tda is not exceeded.

Table 3-4. Link Utilizations for Scenario 2

Link Utilizations (%)	Spine Switch 1	Backup Spine Switch
Leaf Switch 1	97%	10%
Leaf Switch 2	60%	0%
Leaf Switch 3	78%	0%
Leaf Switch 4	70%	0%

A new flow comes from Leaf Switch 1 with flow rate 400Mbps. As shown in Table 3-3, the utilization of the link between Spine Switch 1 and Leaf Switch 1 is 97%. That means that the link only has approximately 300Mbps available bandwidth, which is not enough for the new flow. As mentioned above, the Leaf Switch 1's timer for activating is on, but the time duration Tda is not exceeded. It is still not the time to activate a Spine switch. Therefore the only choice is to assign the new flow on the Backup Spine Switch, which has enough available bandwidth. If during Tda, the utilization of the link between Spine Switch 1 and Leaf Switch 1 remains above SUT-H, GSSMS will have a new active Spine switch after Tda.

Scenario 3: Scenario for Stopping the Timer for Activating a Spine Switch

Link utilizations are shown in Table 3-5. The timers of Leaf Switch 1 and 2 for activating a Spine switch are initially on, but the time duration Tda is not exceeded.

Table 3-5. Link Utilizations for Scenario 3

Link Utilizations (%)	Spine Switch 1	Backup Spine Switch
Leaf Switch 1	97%	10%
Leaf Switch 2	61%	0%
Leaf Switch 3	78%	0%
Leaf Switch 4	70%	0%

A flow on Leaf Switch 2 with flow rate 500Mbps has finished. After processing this End Flow request, the utilization of the link between Spine Switch 1 and Leaf Switch 2 decreases to about 56%, which is lower than SUT-H of 60%. As a result, Leaf Switch 2's timer for activating a Spine switch is turned OFF while Leaf Switch 1's timer still remains on. Therefore, for the Spine Switch Controller, the timer for activating a Spine switch is still on.

Scenario 4: Scenario for Starting Timer for Deactivating a Spine Switch

Link utilizations are shown in Table 3-6. The timer for deactivating a Spine switch is initially OFF.

Table 3-6. Link Utilizations for Scenario 4

Link Utilizations (%)	Spine Switch 1	Spine Switch 2	Backup Spine Switch
Leaf Switch 1	97%	22%	0%
Leaf Switch 2	60%	0%	0%
Leaf Switch 3	80%	5%	0%
Leaf Switch 4	90%	8%	0%

A flow on Leaf Switch 1 with flow rate 300Mbps has finished. Before this End Flow request arrives, except Leaf Switch 1, all Leaf switches have one link with utilization

below FUT-L. After processing this End Flow request, the utilization of the link between Spine Switch 1 and Leaf Switch 1 decrease to approximately 19%, which is lower than FUT-L. That means that all Leaf switches have one link with utilization below FUT-L. Therefore, the Spine Switch Controller starts the timer for deactivating a Spine switch.

Scenario 5: Scenario for Checking Duration for Deactivating a Spine Switch

Link utilizations are shown in Table 3-7. The timer for deactivating a Spine switch is initially on, but the time duration Tdd is not exceeded.

Table 3-7. Link Utilizations for Scenario 5

Link Utilizations (%)	Spine Switch 1	Spine Switch 2	Backup Spine Switch
Leaf Switch 1	97%	18%	0%
Leaf Switch 2	98%	0%	0%
Leaf Switch 3	80%	5%	0%
Leaf Switch 4	90%	10%	0%

A new flow comes from Leaf Switch 1 with flow rate 500Mbps. Because the available bandwidth (approximately 300Mbps) on Spine Switch 1 is not enough to handle this flow, the new flow is assigned to Spine Switch 2, and the utilization of the link between Spine Switch 2 and Leaf Switch 1 increases to approximately 23%, which is higher than FUT-L of 20%, but lower than SUT-L of 40%. That means that after processing this New Flow request, the timer for deactivating a Spine switch still remains on. If none of the links connected to Spine Switch 2 has utilization that is equal to or higher than SUT-L during the time duration Tdd, the Spine Switch Controller will

deactivate Spine Switch 2 when Tdd is exceeded. The Spine Switch Controller chooses Spine Switch 2 to deactivate because compared with Spine Switch 1, it has lower utilization.

Scenario 6: Scenario for Stopping the Timer for Deactivating a Spine Switch

Link utilizations are shown in Table 3-8. The timer for deactivating a Spine switch is initially on, but the time duration Tdd is not exceeded.

Table 3-8. Link Utilizations for Scenario 6

Link Utilizations (%)	Spine Switch 1	Spine Switch 2	Backup Spine Switch
Leaf Switch 1	97%	22%	0%
Leaf Switch 2	99%	38%	0%
Leaf Switch 3	80%	5%	0%
Leaf Switch 4	90%	8%	0%

A new flow comes from Leaf Switch 2 with flow rate 500Mbps. Because the link between Spine Switch 1 and Leaf Switch 2 only has approximately 100Mbps of available bandwidth, which is not enough for the new flow, the link between Spine Switch 2 and Leaf Switch 2 takes over the new flow. The utilization of the link between Spine Switch 2 and Leaf Switch 2 increases to approximately 43%, which is higher than SUT-L of 40%. That means that Leaf Switch 2 does not have any link with utilization below SUT-L. Therefore the timer for deactivating a Spine switch is stopped and reset to zero.

3.5 Summary

In this chapter, a detailed description of GSSMS is provided, including the framework of

the system and the workflows for the Routing Module and the Spine Switch Controller. The GSSMS framework reveals the relationship between the three logic modules: Routing, Spine Switch Controller and Power Control. The workflow of the Routing Module demonstrates the algorithm used to locate flows. The control flow of the Spine Switch Controller presents the algorithm used to make decisions for activating and deactivating. Finally, six scenarios are provided to demonstrate how GSSMS operates for six key example system states.

Chapter 4: Simulation and Results

This chapter consists of five sections. The first section discusses the simulation setup and design, which introduces the ON/OFF traffic model used in the simulation and provides the description of the simulation procedure. The second section discusses the input traffic patterns used as input in the simulations. The third section presents the simulation results for the input traffic patterns, which illustrate that the proposed GSSMS can work well for the input traffic patterns. The fourth section presents the comparison between GSSMS and datacenter with fixed numbers of Spine switches. The fifth section focuses on performance analysis, which reveals the impact of the system and workload parameters on performance. The sixth section provides some guidelines for choosing the control parameters.

4.1 Simulation Setup and Design

This thesis uses CloudSim [CLOUDS] to simulate a datacenter network with GSSMS. The traffic model used in this thesis is the ON/OFF traffic model. The ON/OFF traffic model is observed in real datacenter networks [Benson10]. In the ON/OFF traffic model, the traffic source in the network has two states: ON state (the traffic source originates a data flow) and OFF state (the traffic source does not send data). For each traffic source, the ON state and the OFF state appear successively. The durations of ON periods and OFF periods are independent and identically distributed (i.i.d.) random variables. This thesis uses the same distribution used in [Calabretta13] for the durations of ON/OFF

period – Pareto distribution. It is assumed that for different ON periods, a given traffic source can have different traffic rates.

In the simulation, as shown in Figure 4-1, the datacenter network consists of 16 Leaf switches and 8 Spine switches adopted from Cisco's practices [Cisco12]. The link capacity between a Leaf switch and a Spine switch is 10Gbps. One Leaf switch connects with 15 servers. One server has 10 VMs and each VM is a traffic source.

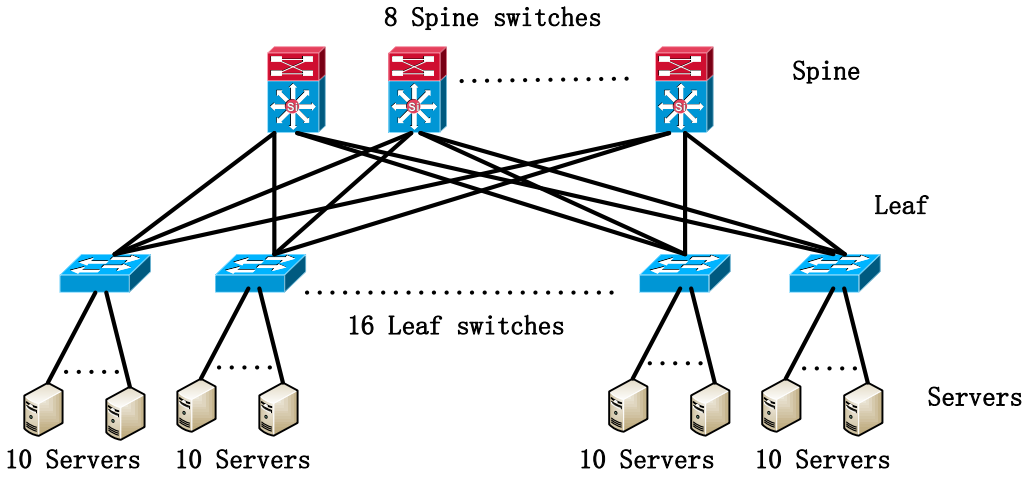


Figure 4-1. Simulation Setup

There are four components in the simulation model: Datacenter Manager, Broker, AppCloudlet and GSSMS. The Datacenter Manager is responsible for managing servers in the network and allocating servers for VMs. The Broker creates VMs and AppCloudlets. The AppCloudlet is the application running on VMs. AppCloudlet is responsible for generating ON and OFF events.

The sequence diagram of the simulated system is shown in Figure 4-2. Firstly, the Broker creates VMs. After a VM is created, it is located on one server by the Datacenter

Manager. Each VM creates one AppCloudlet. After the AppCloudlet is created, it generates the first ON event and sends the ON event to the Datacenter Manager. Then, when the Datacenter Manager receives an ON event, if the flow's source and destination connect with the same Leaf switch, the Datacenter Manager sends a New Flow request to GSSMS. When the Datacenter Manager receives an OFF event, if the flow's source and destination connect with the same Leaf switch, the Datacenter Manager sends an End Flow request to GSSMS. After sending the request to GSSMS, the Datacenter Manager asks the AppCloudlet to generate the next ON/OFF event. Finally, if GSSMS receives a New Flow request from the Datacenter Manager, it returns the route for the new flow to the Datacenter Manager and updates the available bandwidth. If GSSMS receives an End Flow request, it updates the available bandwidth without returning anything.

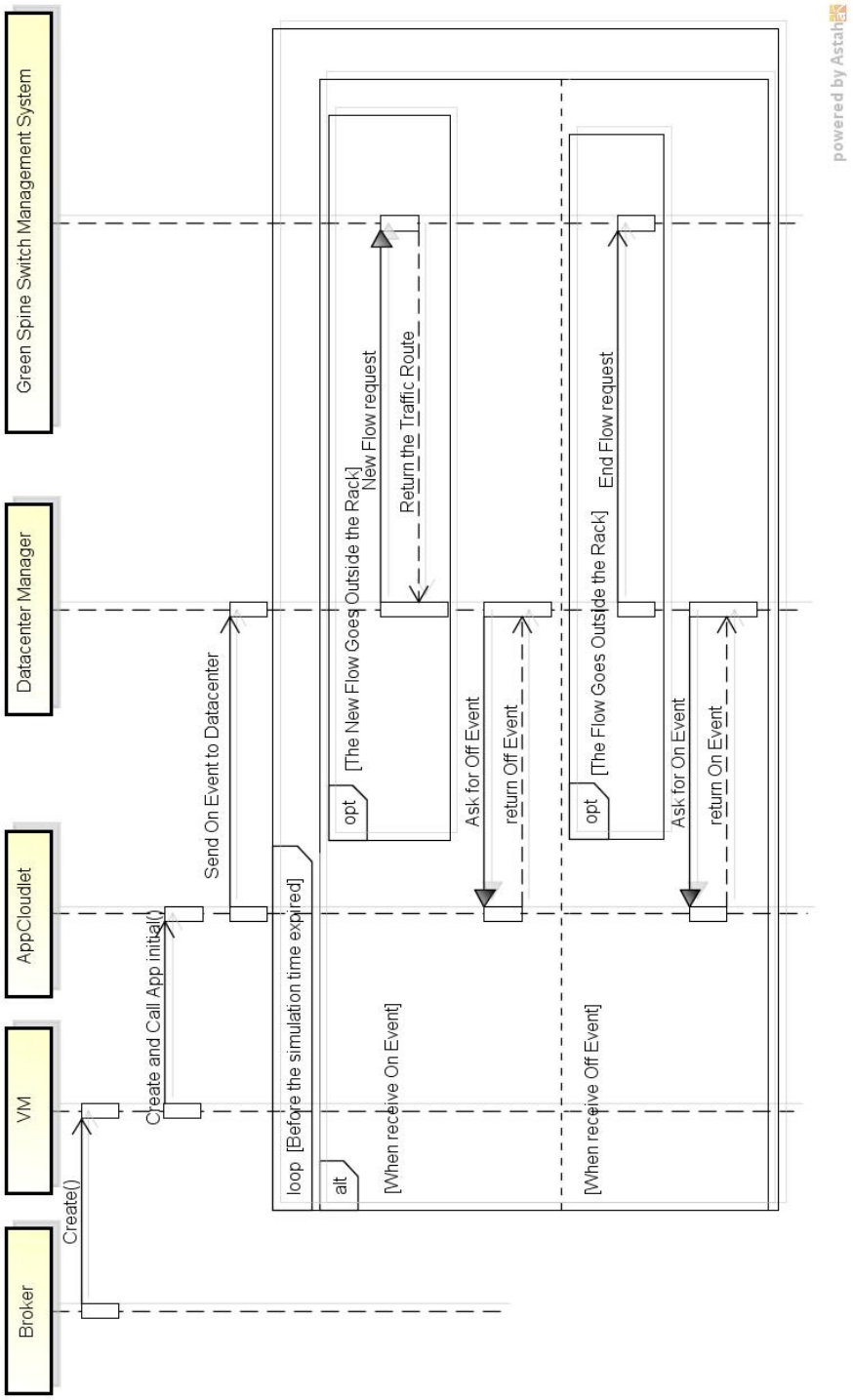


Figure 4-2. The Sequence Diagram of the Simulation

4.2 Input Traffic Patterns

This thesis uses three types of traffic pattern for GSSMS's input. These three types of traffic pattern are: Uniform Traffic, Sine-Wave Traffic and Random Traffic.

4.2.1 Uniform Traffic

The uniform traffic [Heller10] means that the traffic rate is fixed for each traffic source. According to the traffic's destination, this thesis has three types of uniform traffic: Near traffic, Far traffic and Half-Far/Half-Near traffic.

a) Near traffic: For each data flow, the flow's source and destination connect with the same Leaf switch. That means that the traffic between the source and destination servers only needs to be routed through one Leaf switch.

b) Far traffic: For each data flow, the flow's source and destination connect with different Leaf switches. That means that the traffic between the source and destination servers has to needs to be routed through one Spine switch.

c) Half-Far/Half-Near traffic: 50% of the traffic is Near traffic, and 50% of the traffic is Far traffic.

4.2.2 Sine-Wave Traffic

For the Sine-Wave traffic, the traffic rate for each traffic source varies as a sine wave. This thesis assumes that

$$\text{Sine} - \text{Wave Traffic Rate} = ½ \times \text{max traffic rate} \times \left(1 + \sin\left(\left(\frac{\pi}{5}\right) \times \text{roundup}\left(\frac{t \times 10}{\text{execution time}}\right)\right)\right) \ldots (4.1)$$

As the equation used in [Heller10], the Equation 4.1 uses 10 discrete values from sine wave and the given maximum traffic rate to calculate the traffic rate at a given time t. In the simulation, the execution time is 60 minutes. Therefore the traffic rate looks like the one shown in Figure 4-3.

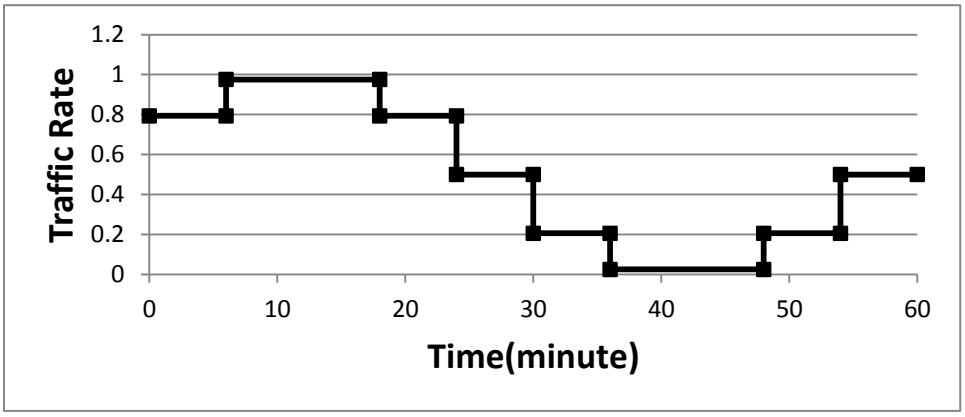


Figure 4-3. The Sine-Wave Traffic

There are three types of Sine-Wave traffic: Near traffic, Far traffic and Half-Far/Half-Near traffic. These three types of Sine-Wave traffic have the same meaning as the three types of Uniform traffic.

4.2.3 Random Traffic

In [Benson10], the authors observed that during 95% of the time, the traffic in datacenters is below 10% of the datacenter network capacity. The random traffic is used to simulate the scenario where 95% of the time, the traffic is 10% of the datacenter network capacity, and during 5% of the time, the traffic is higher than 10% of the datacenter network

capacity. In the simulation setup, each link has 10Gbps. Therefore, in the random traffic scenario, the traffic rate for each traffic source is 9Mbps for 95% of the time, and uniformly varies within the range between 50Mbps and 200Mbps for the rest 5% of the time.

4.3 Simulation Results for Input Traffic Patterns

In this section, the simulation results for the three input traffic patterns are presented. The simulation results show that GSSMS can perform well for the three input traffic patterns.

4.3.1 Uniform Traffic

The simulation results of the uniform traffic are shown in Figure 4-4 and Figure 4-5. In Figure 4-4, ANASS represents the average number of active Spine switches, which is defined by Equation 4.2, and POPC means the percentage of original power consumed by Spine switches, which is defined by Equation 4.3. In Equation 4.3, N represents the static number of Spine switches. [Miercom13] reports that Cisco switches in Hibernation mode reduce the power consumed for the Catalyst 2960-X model by 84%. Therefore, this thesis assumes that a switch in the Hibernation mode can save the consumed power by 84%. In other words, a switch in the Hibernation mode consumes 16% of the power consumed by an active switch. In Figure 4-5, PF refers to the percentage of failures. PF is calculated by Equation 4.4.

$$\text{ANASS} = \frac{\sum number\ of\ active\ Spine\ switches \times duration}{total\ duration} \quad \text{..... (4.2)}$$

$$\text{POPC} = \frac{ANASS+(N-ANASS)\times 0.16}{N} \times 100\% \text{.....} \quad (4.3)$$

$$\text{PF} = \frac{\text{the number of failed flows}}{\text{the number of all flows}} \text{.....} \quad (4.4)$$

Failures in the simulation mean that when a VM wants to send data, there is not enough available bandwidth for the data flow. In GSSMS, two is the minimum number of active Spine switches for configuration. One of the two Spine switches is the backup Spine switch.

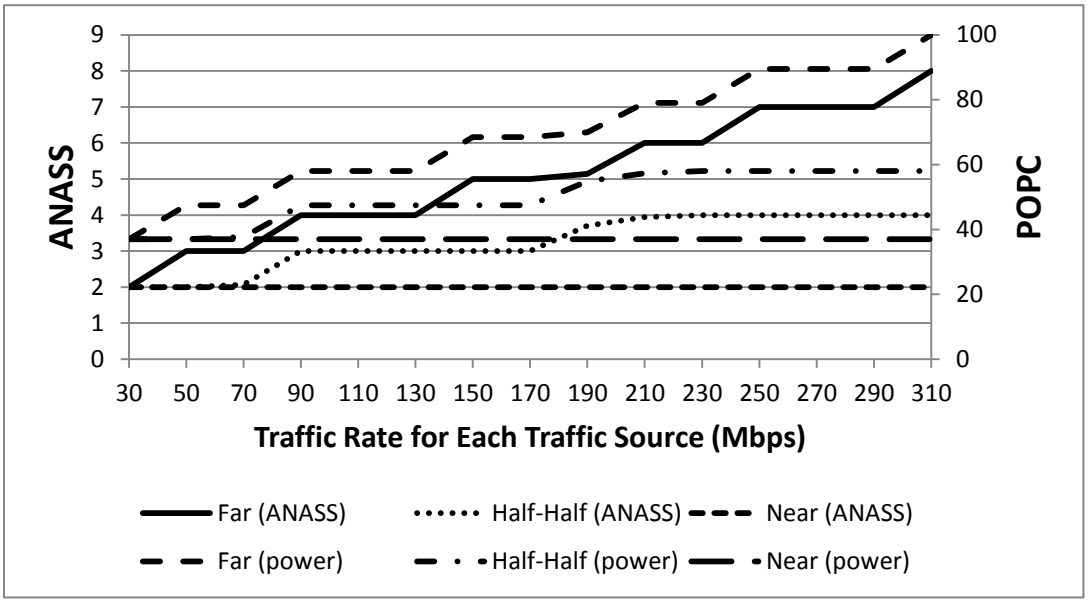


Figure 4-4. The Simulation Results of the Uniform Traffic

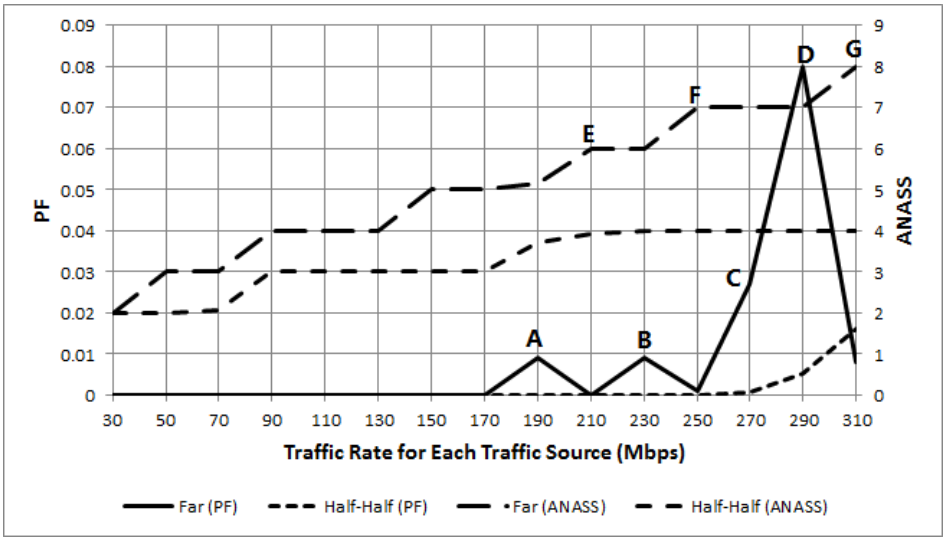


Figure 4-5. The Simulation Results for PF

As we can see from Figure 4-4, for the Near traffic case, ANASS is two all the time. Because the traffic in the datacenter network is not routed through Spine switches for Near traffic, the total traffic routing through Spine switches is always zero in this scenario. Therefore, ANASS is not affected by the change of the traffic rate.

For the Far traffic case, ANASS increases as the traffic rate for each traffic source increases. In this scenario, the total traffic routing through Spine switches increases while the traffic rate for each traffic source increases. The increase of the total traffic causes the increase of ANASS.

For the Half-Far/Half-Near traffic case, ANASS also increases as the traffic rate for each traffic source increases. The reason is that the total traffic routing through Spine switches increases while the traffic rate for each traffic source increases. Compared with Far traffic, for a given traffic rate, the total traffic of the datacenter network is

approximately half of that in the Far traffic scenario. Therefore, ANASS is approximately half of ANASS in the Far traffic scenario.

In Figure 4-5, the simulation results for PF are presented. Failures appear when the traffic rate increases while ANASS remains the same value. For instance, the four points shown in Figure 4-5: A, B, C and D. For A, ANASS is five for the traffic rate range 150Mbps to 190Mbps. Failures happen when the traffic rate is 190Mbps. For B, ANASS is six for the traffic rate range 210Mbps to 230Mbps. Failures happen when the traffic rate is 230Mbps. For C and D, ANASS is seven for the traffic rate range 250Mbps to 290Mbps. Failures happen when the traffic rate is 270Mbps and 290Mbps. While the traffic rate for each traffic source increases, the average traffic of the network calculated by Equation 4.3 and the traffic rate of instantaneous traffic increase together. Because GSSMS does not activate Spine switches for the instantaneous traffic, it is possible that the network does not have enough available bandwidth for the instantaneous traffic when the traffic in the network almost reaches the network capacity. While the network does not have enough available bandwidth for the instantaneous traffic, failures happen. When the traffic rate continues to increase, ANASS increases, such as the points E, F, and G shown in Figure 4-5. The increase of ANASS means that the available bandwidth of the network increases. Therefore, the network can take over the instantaneous traffic without failures.

$$\text{average traffic} = \text{average number of concurrent flows} \times \text{the traffic rate} \ldots (4.3)$$

4.3.2 Sine-Wave Traffic

For the Sine-Wave traffic, the simulation result for Near traffic is shown in Figure 4-6. NASS refers to the number of active Spine switch. Two active Spine switches is the minimum number of Spine switches required in GSSMS.

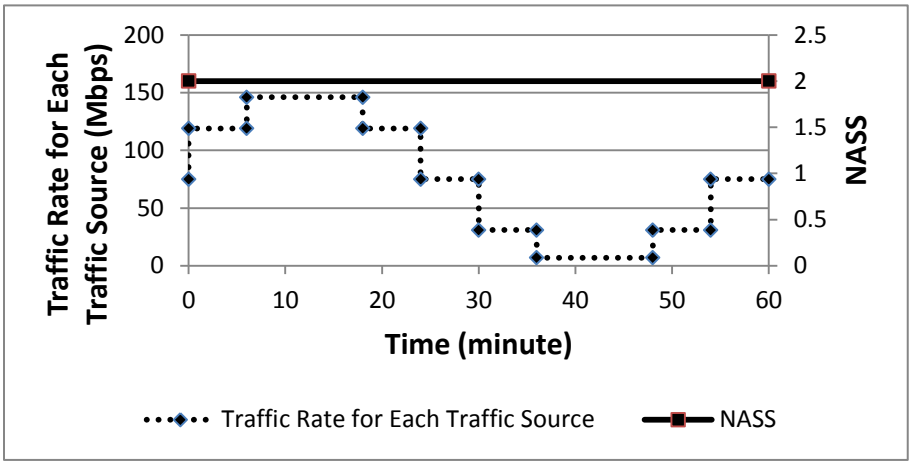


Figure 4-6. The Simulation Result for Near Traffic for the Sine-Wave Traffic

As we can see in Figure 4-6, the number of Spine switches does not change because the traffic in the datacenter network is not routed through Spine switches for Near traffic.

The simulation result for Far traffic is shown in Figure 4-7.

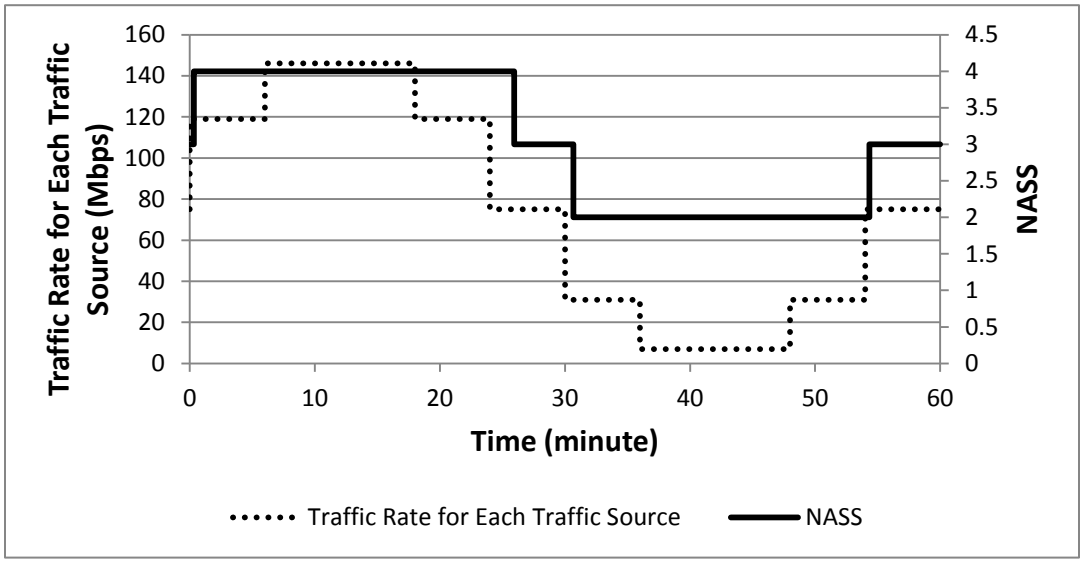


Figure 4-7. The Simulation Result for Far Traffic for the Sine-Wave Traffic

As we can see in Figure 4-7, the number of Spine switches changes as the traffic rate for each traffic source changes. The result shows that NASS changes according to the total traffic of the datacenter network. As the total traffic of the datacenter network increases, more Spine switches are activated in the datacenter network. As the total traffic of the datacenter decreases, NASS of the datacenter network decreases too.

The simulation result for Half-Far/Half-Near traffic is shown in Figure 4-8.

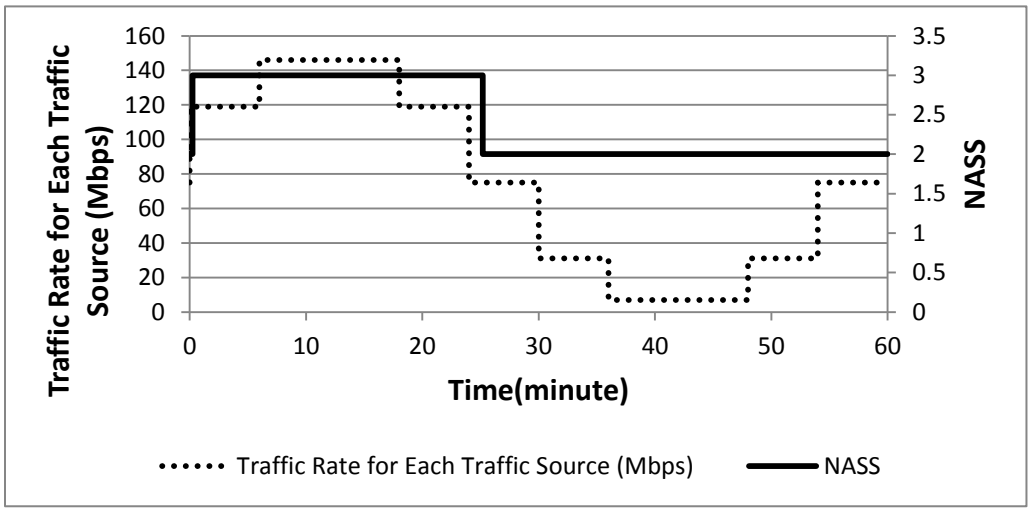


Figure 4-8. The Simulation Result for Half-Far/Half-Near Traffic for the Sine-Wave Traffic

We can see that the simulation result of Half-Far/Half-Near traffic is similar to the simulation result of Far traffic. Because the total traffic of the datacenter network in Half-Far/Half-Near traffic scenario is lower than that in the Far traffic scenario, the value of NASS at a given point in time seems to be smaller than that achieved with the Far traffic scenario.

4.3.3 Random Traffic

The simulation result for Random traffic is shown in Figure 4-9.

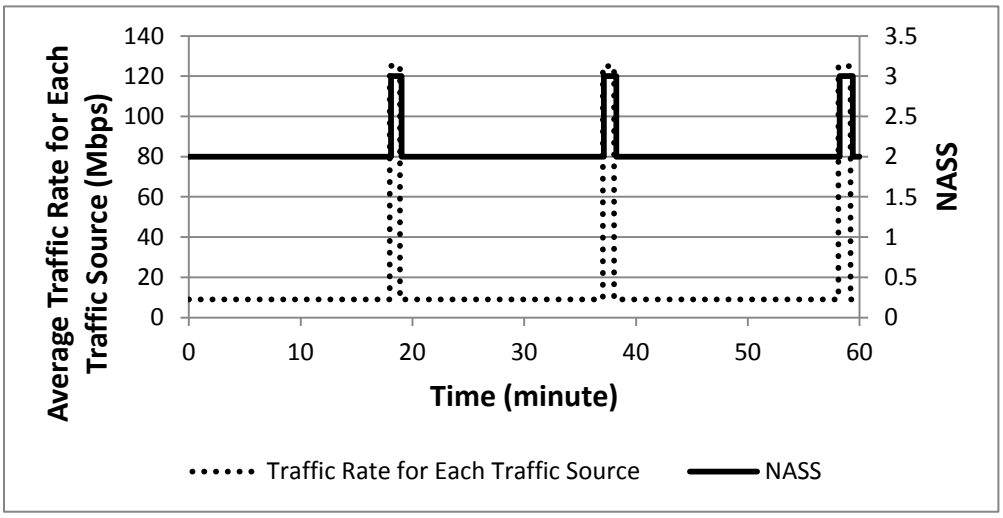


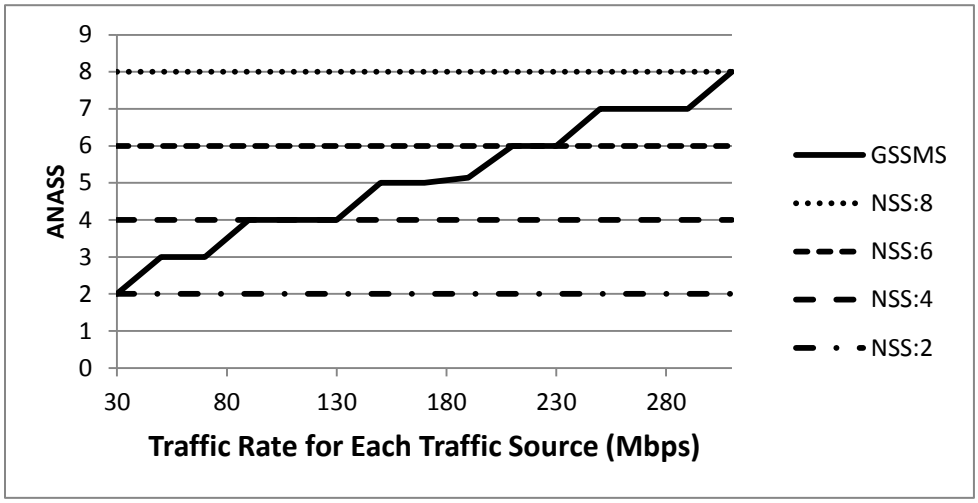
Figure 4-9. The Simulation Result of the Random Traffic

As shown in Figure 4-9, the traffic bursts last for approximately 1 minute and GSSMS increases the number of active Spine switches for the duration of the traffic bursts and decrease the number of active Spine switches when the traffic bursts finish. The time duration Tda is 5 seconds and Tdd is 10 seconds in the simulation. NASS increases approximately 5 seconds after the traffic rate for each traffic source increases. NASS decreases approximately 10 seconds after the traffic rate for each traffic source decreases. Because the x-axis values in the graph are in minutes, it is very difficult to visualize these delays clearly in the figure.

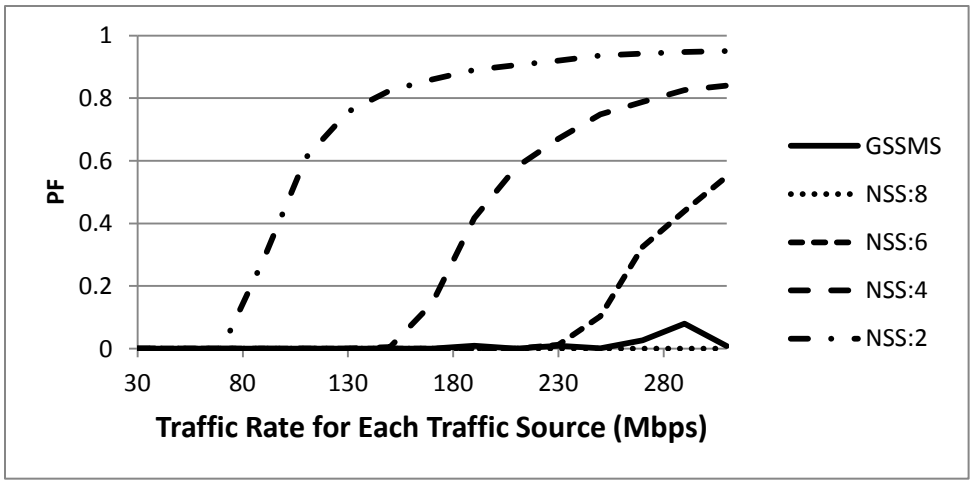
4.4 Comparison between GSSMS and Fixed Numbers of Spine Switches

In this section, the simulation results present the difference between GSSMS and a datacenter with fixed numbers of Spine switches in ANASS and PF. Because, as discussed earlier in Section 4.3.1, the number of Spine switches never changes for the Near traffic case, the simulation results for Near traffic are not discussed.

Figure 4-10 shows the simulation results for Uniform-Far traffic. NSS in Figure 4-10 represents the number of Spine switches used in a datacenter that does not use GSSMS and deploys a fixed number of Spine switches.



(a) ANASS



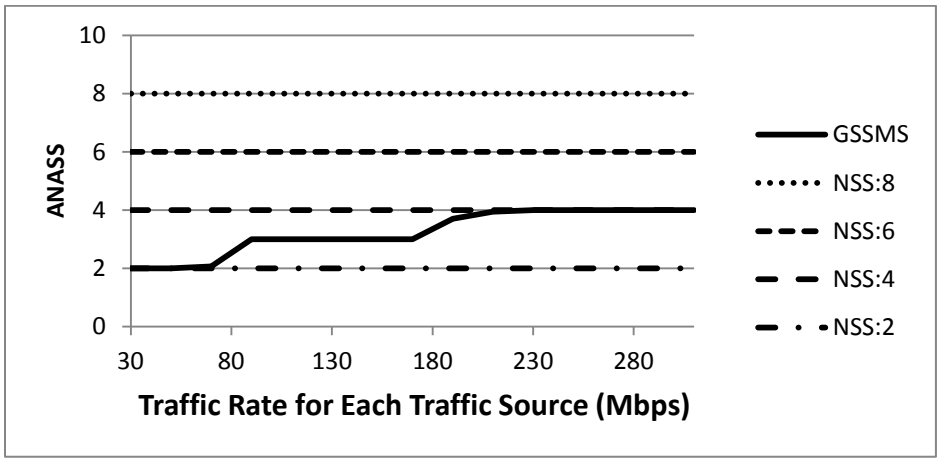
(b) PF

Figure 4-10. Simulation Results for Uniform-Far Traffic

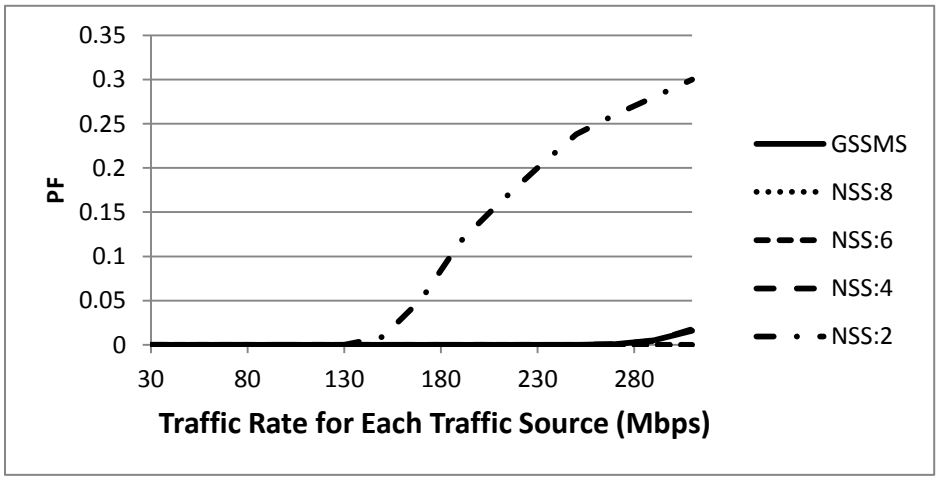
Compared with the datacenter with a fixed set of 8 Spine switches, GSSMS can save energy when not all the Spine switches are active. However, GSSMS has a small number

of failures when traffic rate is higher than 180Mbps. Compared with the datacenter with a fixed set of 6 Spine switches, GSSMS can save energy when the traffic rate is lower than 210Mbps. When the traffic rate is higher than 250Mbps, GSSMS leads to an ANASS that is higher than six. Although GSSMS consumes more energy than the datacenter with a fixed set of six Spine switches when the traffic rate is higher than 250Mbps, it has a much lower PF than that with six Spine switches. When the traffic rate is 270Mbps, PF of GSSMS is only 0.03 while PF of the datacenter with six Spine switches is more than 0.3. Compared with the datacenter with a fixed set of 4 and 2 Spine switches, when the traffic rate is high (e.g., 190Mbps), GSSMS leads to an ANASS that is higher than four, but has much fewer failures.

Figure 4-11 shows the simulation results for Uniform-Half/Half traffic.



(a) ANASS



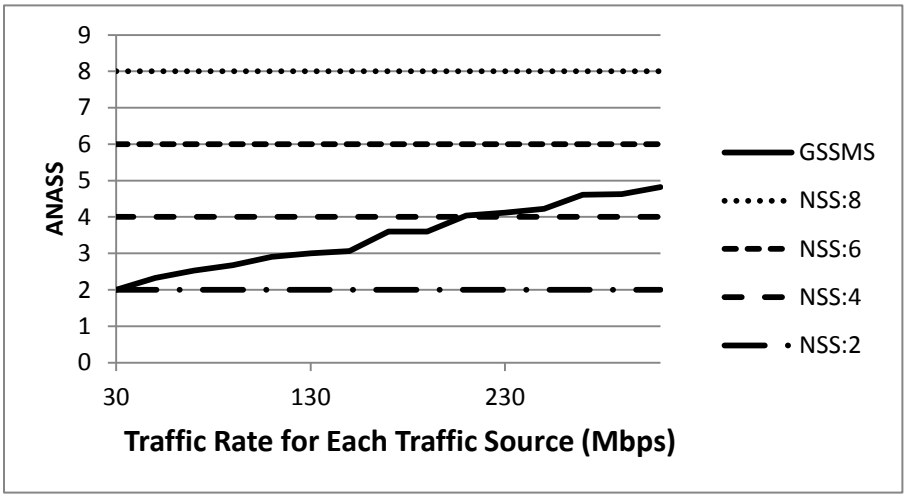
(b) PF

Figure 4-11. Simulation Results for Uniform-Half/Half Traffic

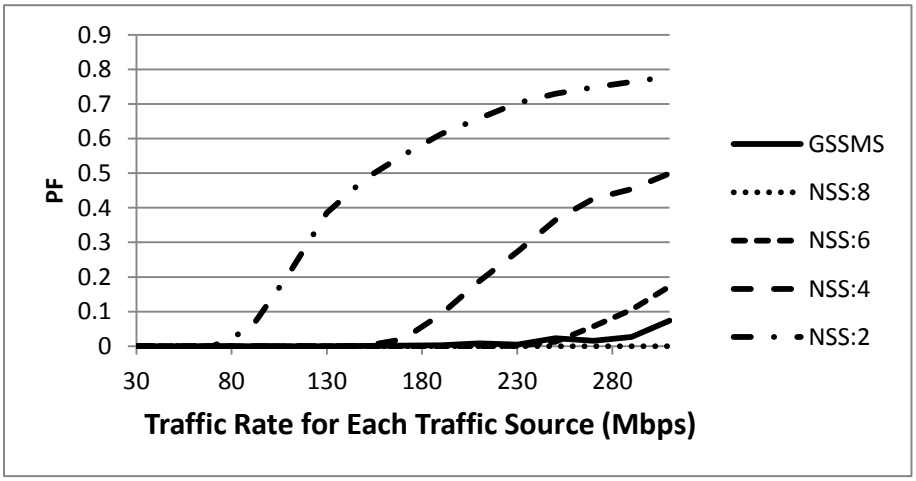
Compared with the datacenter with a fixed set of 8, 6 and 4 Spine switches, GSSMS has an ANASS smaller or equal to 4 and saves energy consumed by Spine switches. In Figure 4-11, compared with the datacenter with a fixed set of 4 Spine switches, GSSMS produces an ANASS lower than 4 when the traffic rate is lower than 210Mbps. When the traffic rate is higher than 210Mbps, ANASS for GSSMS is 4 and it produces a PF that is similar to the datacenter using a fixed set of 4 Spine switches. On the other hand, GSSMS

consumes more energy than the datacenter with a fixed set of 2 Spine switches when the traffic rate is higher than 70Mbps, but has much fewer failures than the datacenter with a fixed set of 2 Spine switches when the traffic rate is higher than 170Mbps.

Figure 4-12 shows the simulation results for Sine-Wave-Far traffic.



(a) ANASS



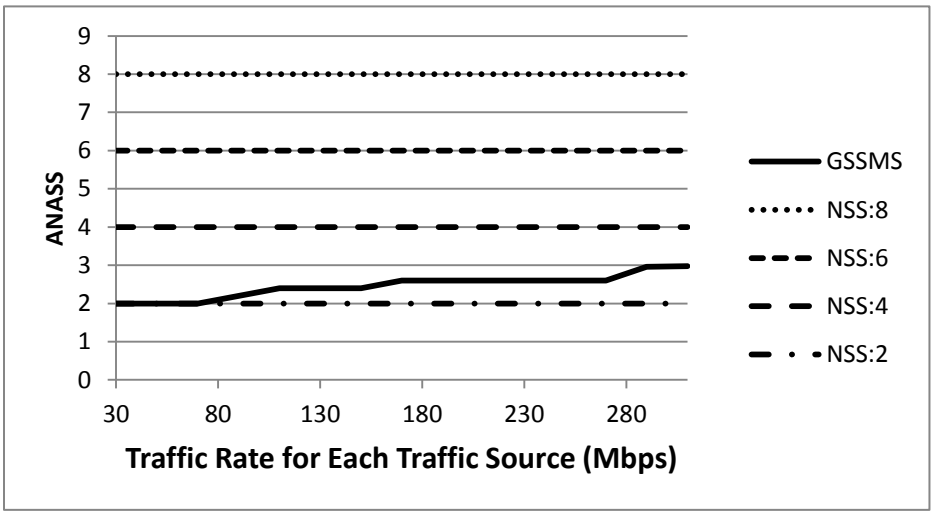
(b) PF

Figure 4-12. Simulation Results for Sine-Wave-Far Traffic

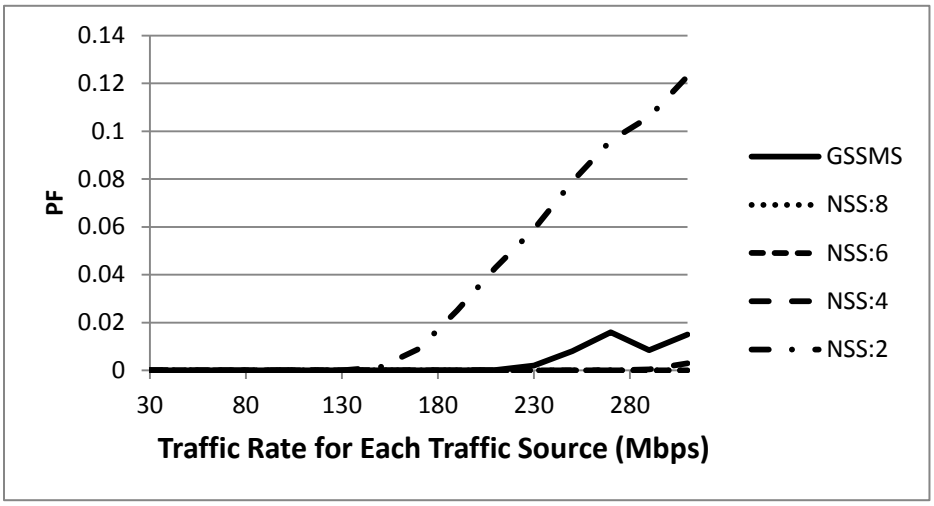
As shown in Figure 4-12, the simulation results for Sine-Wave-Far traffic are similar

to the simulation results for Uniform-Far traffic. The traffic rate for each traffic source is the max traffic rate in Equation 4.1. When the max traffic rate is 280Mbps, ANASS of GSSMS is less than five. GSSMS can save energy compared with the datacenter with a fixed set of 8 and 6 Spine switches. At the same time, GSSMS has a smaller PF than that with 2, 4 and 6 Spine switches. Compared with the datacenter with a fixed set of 2 and 4 Spine switches, GSSMS has a smaller PF because it has an ANASS larger than 4. Compared with the datacenter with a fixed set of 6 Spine switches, GSSMS has both a smaller ANASS and a smaller PF. The reason is that the traffic rate varies in the Sine-Wave traffic case. As shown in Section 4.2.2, the traffic rate at a given time can reach the max traffic rate during only one fifth of the simulation time. When the traffic at a given time reaches the max traffic rate, e.g. 280Mbps, the datacenter with a fixed set of 6 Spine switches does not have enough bandwidth to handle the total traffic routing through Spine switches and leads to a PF of 0.1. On the other hand, the number of active Spine switches of GSSMS can reach 7 when the traffic rate at a given time reaches the max traffic rate, and GSSMS has a small number of failures (PF is 0.02). Because the number of active Spine switches of GSSMS decreases while the traffic routing through Spine switches at a given time decreases, GSSMS has an ANASS lower than 5 when the max traffic rate is 280Mbps.

Figure 4-13 shows the simulation results for Sine-Wave-Half/Half traffic.



(a) ANASS



(b) PF

Figure 4-13. Simulation Results for Sine-Wave-Half/Half Traffic

As shown in Figure 4-13, the simulation results for Sine-Wave-Half/Half traffic are similar to the simulation results for Uniform-Half/Half traffic. GSSMS has an ANASS smaller or equal to 3. Thus, GSSMS consumes more energy than the datacenter with a fixed set of 2 Spine switches only. When the traffic rate for each traffic source is high, e.g.

280Mbps, the PF of GSSMS is less than 0.02, much lower than the PF of the datacenter with a fixed set of 2 Spine switches. Compared with the datacenter with a fixed set of 8, 6 and 4 Spine switches, GSSMS can save energy with a increase in PF.

The simulation results reveal that compared with the datacenter with a fixed numbers of Spine switches, GSSMS can save energy consumed by Spine switches with an acceptable increase in PF (less than 0.09). When the network traffic is higher than 180Mbps, GSSMS can reduce PF significantly (100% to 82%) by having one or more active Spine switches compared with the datacenter with a fixed set of two, four and six Spine switches.

4.5 Performance Analysis: Effect of System Parameters

In this section, the impact of the parameters on performance is presented. The parameters include FUTs, CTs, SUTs, the Time Duration Tda and the number of Leaf switches. Because for the Near traffic case, the number of Spine switches never changes, the simulation results for Near traffic are not discussed. Because the simulation with large Tda value cannot reveal the impact of the parameters clearly, the default value of Tda is 0.5 seconds in this section.

The values of GSSMS parameters are shown in Table 4-1 (bold ones are the default values):

Table 4-1. Values for GSSMS Parameters

Parameters	Values
FUT-H	100%, 95%, **90%**, 80%
FUT-L	100%, 20%, **10%**, 5%
CT-H	100%, 95%, **90%**, 80%
CT-L	30%, **20%**, 10%, 0%
SUT-H	90%, 80%, **70%**, 60%
SUT-L	100%, 30%, **20%**, 5%
Tda	0.3s, **0.5s**, 1s, 3s, 5s, 20s, 100s
Tdd	0.6s, **1s**, 2s, 6s, 10s, 40s, 200s

The values of GSSMS parameters shown in Table 4-1 were determined from pilot experiments. A value that produced a reasonable performance was determined for each parameter. A range of values was used to study the impact of the parameters on performance.

The values of workload parameters are shown in Table 4-2 (bold ones are the default values):

Table 4-2. Values of Workload Parameters

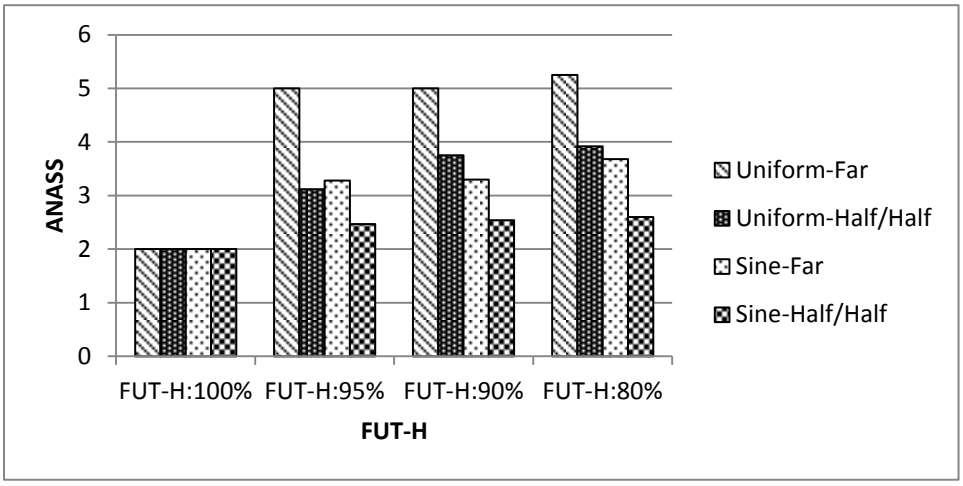
Parameters	Values
Number of Leaf Switches	24, **16**, 12, 8
ON/OFF Duration-ON	200ms, **100ms**, 50ms
ON/OFF Duration-OFF	40ms, **20ms**, 10ms

4.5.1 The Impact of FUTs on Performance

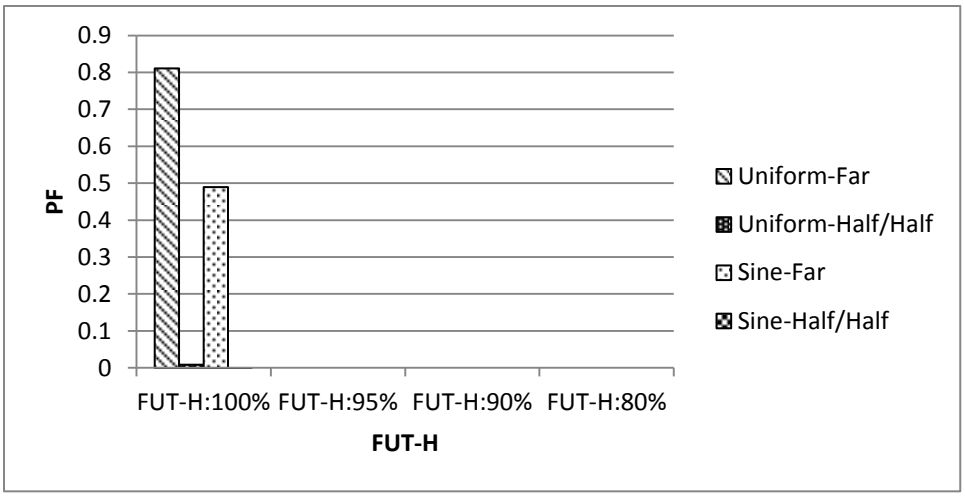
As mentioned in Chapter 3, there are two FUTs: FUT-H and FUT-L. FUT-H is used for activating a Spine Switch in response to an increase in traffic. FUT-L is used for deactivating Spine switches in response to a decrease in traffic.

4.5.1.1 The Impact of FUT-H on Performance

Figure 4-14 shows the simulation results which demonstrate the impact of FUT-H.



(a) FUT-L = 10%



(b) FUT-L = 10%

Figure 4-14. The Impact of FUT-H on Performance

The simulation results shown in Figure 4-14 (a) demonstrate that ANASS increases as FUT-H decreases. The reason is that compared with GSSMS with a high FUT-H,

GSSMS with a low FUT-H is more likely to activate a Spine switch under the same situation. For instance, consider a situation in which the traffic increases to 96% of a link capacity and then decreases to 70% of a link capacity within the time duration Tda. In this scenario, when the traffic increases to 96% of a link capacity, the traffic can trigger the timer of activating a Spine switch for both GSSMS with FUT-H of 95% and GSSMS with FUT-H of 80%. However, when the traffic decreases to 70% of a link capacity, the timer of GSSMS with FUT-H of 95% turns to OFF (SUT-H is 20 lower than FUT-H, and 70% is lower than SUT-H of 75%). On the other hand, the timer of GSSMS with FUT-H of 80% is still ON because 70% is higher than SUT-H of 60%. As a result, after the time duration Tda, GSSMS with FUT-H of 80% activates a Spine switch while GSSMS with FUT-H of 95% does not.

The simulation results shown in Figure 4-14 (b) illustrate that when FUT-H is 100%, PF of Far traffic increases dramatically. A 100% FUT-H means that only when the active Spine switches are fully used, GSSMS can trigger the timer for activating a Spine switch. As a result, when the current active Spine switches do not have enough available bandwidth for the traffic, a Spine switch cannot be activated in time. Not activating a Spine switch in time leads to the occurrence of failures. PF of Half-Far/Half-Near traffic increases slightly because compared with the Far traffic case, the Half-Far/Half-Near traffic case has lower traffic. To summarize, the simulation results demonstrate that high FUT-H leads to a low ANASS and a high PF compared with a low FUT-H.

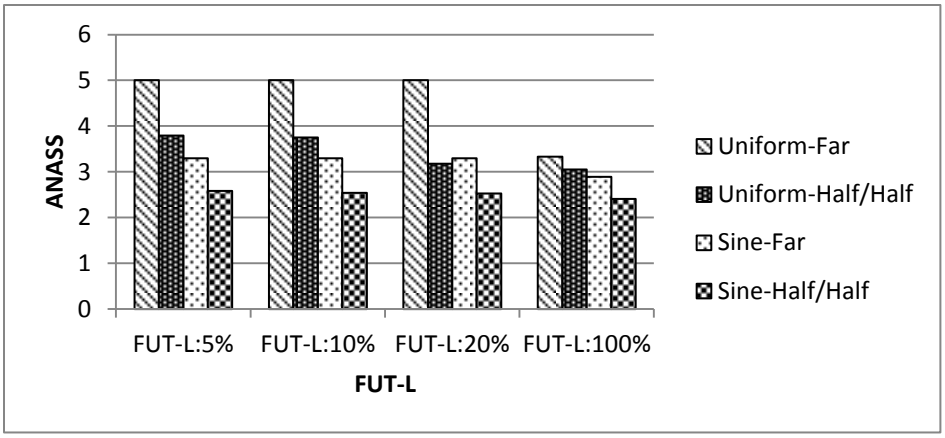
In Figure 4-14 (a), ANASS of Uniform-Far traffic for given FUT-H and FUT-L is higher than ANASS of Uniform-Half-Far/Half-Near traffic except for 100% FUT-H, and ANASS of Sine-Wave-Far traffic is higher than ANASS of Sine-Wave-Half-Far/Half-Near traffic except for 100% FUT-H. The reason is that compared with Far traffic, Half-Far/Half-Near traffic has less traffic routing through Spine switches. The number of active Spine switches of GSSMS at a given time is determined by the traffic routing through Spine switches. Thus, the number of active Spine switches of GSSMS for Half-Far/Half-Near traffic at a given time is lower than that for Far traffic. For the 100% FUT-H case, there is no difference in ANASS for the four traffic patterns because 100% is too high and GSSMS never has a chance to increase the number of active Spine switches during the simulation.

Figure 4-14 (a) also shows that ANASS of Uniform-Far traffic for given FUT-H and FUT-L is higher than ANASS of Sine-Wave-Far Uniform-Half-Far/Half-Near traffic except for 100% FUT-H, and ANASS of Uniform-Half-Far/Half-Near traffic for given FUT-H and FUT-L is higher than ANASS of Sine-Wave-Half-Far/Half-Near traffic except for 100% FUT-H. This is caused by the difference in traffic routing through Spine switches in the Uniform traffic case and the Sine-Wave traffic case. As introduced in Section 4.2, in the Uniform traffic case, the traffic rate for each traffic source is fixed. In the Sine-Wave traffic case, the traffic rate at a given time varies as a sine wave, and the traffic rate assigned as the traffic rate for each traffic source is the maximum traffic rate.

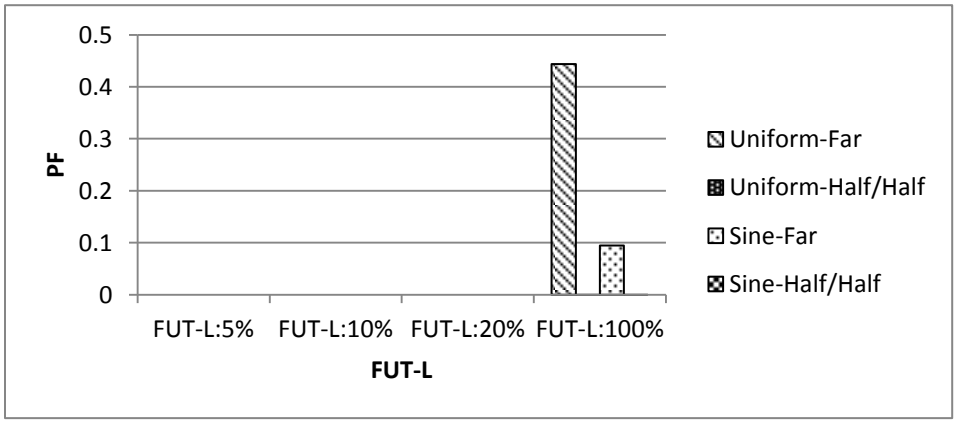
The traffic rate at a given time can reach the maximum traffic rate during only one fifth of the simulation time. Therefore, for Uniform traffic and Sine-Wave traffic with the same traffic rate for each traffic source, the traffic routing through Spine switches in the Uniform traffic case is higher than that in the Sine-Wave traffic case.

4.5.1.2 The Impact of FUT-L on Performance

Figure 4-15 shows the simulation results which demonstrate the impact of FUT-L.



(a) FUT-H = 90%



(b) FUT-H = 90%

Figure 4-15. The Impact of FUT-L on Performance

The simulation results shown in Figure 4-15 (a) demonstrate that ANASS decreases as FUT-L increases except for the Uniform-Far traffic case. The reason is that compared with GSSMS using a low FUT-L, a GSSMS using a high FUT-L can deactivate a Spine switch more easily. For instance, consider a situation in which the traffic decreases to 3% of a link capacity and then increases to 26% of a link capacity within the time duration Tdd. In this scenario, when the traffic decreases to 3% of a link capacity, the traffic can trigger the timer of deactivating a Spine switch for GSSMS with FUT-L of 5% and GSSMS with FUT-L of 20%. However, when the traffic increases to 26% of a link capacity, the timer of GSSMS with FUT-L of 5% turns to OFF (SUT-L is 10 higher than FUT-L, and 26% is higher than SUT-L of 15%). On the other hand, the timer of GSSMS with FUT-L of 20% is still ON because 26% is lower than SUT-L of 30%.Therefore, GSSMS with FUT-L of 20% has higher probability of deactivating a Spine switch than GSSMS with FUT-L of 10% under the same situation. For Uniform-Far traffic, the utilizations of all links are much higher than FUT-L. GSSMS does not have chance to deactivate a Spine switch for the Uniform-Far traffic case. Therefore, the simulation results for Uniform-Far traffic cannot show the impact of FUT-L.

The simulation results shown in Figure 4-15 (b) illustrate that when FUT-L is 100%, PF of Far traffic increases dramatically. A 100% FUT-L means that GSSMS can start the timer for deactivating Spine switches at any time and can always deactivate Spine switches after the time duration Tdd. As a result, GSSMS deactivates Spine switches

even when the Spine switches are needed to handle the traffic. The inappropriate deactivation of Spine switches leads to the occurrence of failures. PF of Half-Far/Half-Near traffic increases slightly for an FUT-L of 100% because compared with the Far traffic case, the Half-Far/Half-Near traffic case leads to a lower traffic. To summarize, the simulation results demonstrate that high FUT-L leads to a low ANASS and a high PF compared with a low FUT-L.

Figure 4-15 (a) also shows that ANASS of GSSMS for Uniform-Far traffic for given FUT-H and FUT-L is always higher than that for Uniform-Half-Far/Half-Near traffic, and ANASS of GSSMS for Sine-Wave-Far traffic for given FUT-H and FUT-L is always higher than that for Sine-Wave-Half-Far/Half-Near traffic. ANASS of GSSMS for Uniform-Far traffic for given FUT-H and FUT-L is always higher than that for Sine-Wave-Far traffic and ANASS of GSSMS for Uniform-Half-Far/Half-Near traffic for given FUT-H and FUT-L is always higher than that for Sine-Wave-Half-Far/Half-Near traffic. As discussed in Section 4.5.1.1, the difference between ANASSs for different traffic patterns is caused by the difference between the traffic routing through Spine switches in the different traffic pattern cases.

4.5.2 The Impact of the Control Thresholds on Performance

Similar to FUTs, there are two CTs: CT-H and CT-L. CT-H is used for activating a Spine Switch in response to an increase in traffic. CT-L is used for deactivating Spine switches when traffic reaches a predetermined low value.

4.5.2.1 The Impact of CT-H on Performance

The simulation results for analyzing the impact of CT-H on performance are shown in Figure 4-16.

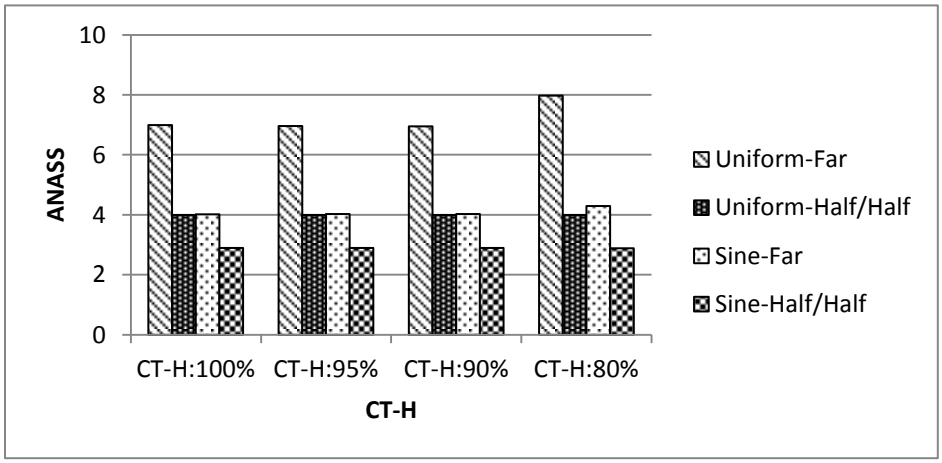


Figure 4-16. The Impact of CT-H on Performance (CT-L = 20%)

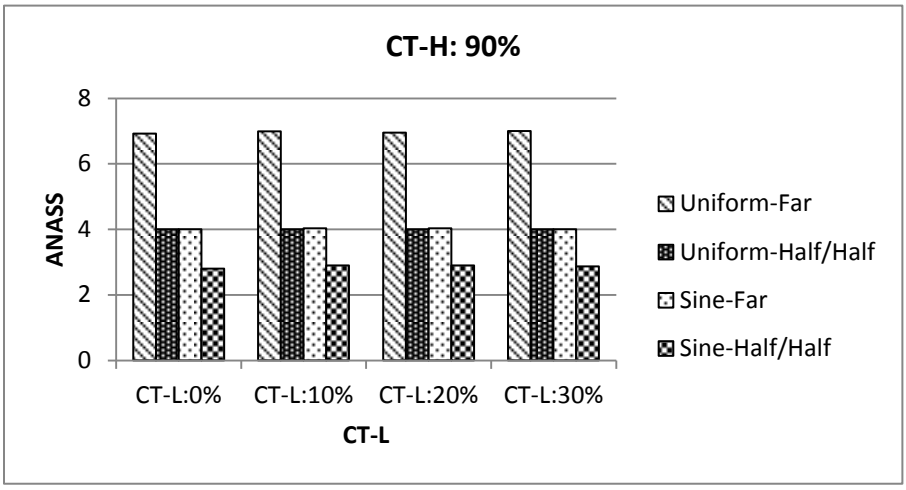
The simulation result for the Far traffic case illustrates that ANASS increases while CT-H decreases. The reason is that a lower CT-H value is easier to reach. For example, consider a situation in which there are six active Spine switches in the network and a Leaf switch has five links with utilization higher than FUT-H. In this situation, if CT-H is 90%, the given number Na is six, and Na is five if CT-H is 80%. That means that when CT-H is 80%, GSSMS sets the timer for activating a Spine switch ON; when CT-H is 90%, the timer for activating a Spine switch is still OFF. Figure 4-16 shows that the performance for Half-Far/Half-Near traffic is not very sensitive to CT-H. The reason is that the number of active Spine switches is too small which makes Na be the same when CT-H has different values.

Changes in CT-H do not seem to have much impact on PF. The simulation results demonstrate that when CT-H is 100%, it only has slight impact on the Sine-Wave Far traffic case (PF is 0.0002). The reason for the occurrence of failures is that GSSMS cannot activate a Spine switch in time because of the high CT-H. The reason of no failures in the Uniform Far traffic case is that seven active Spine switches are enough to handle the traffic without failure.

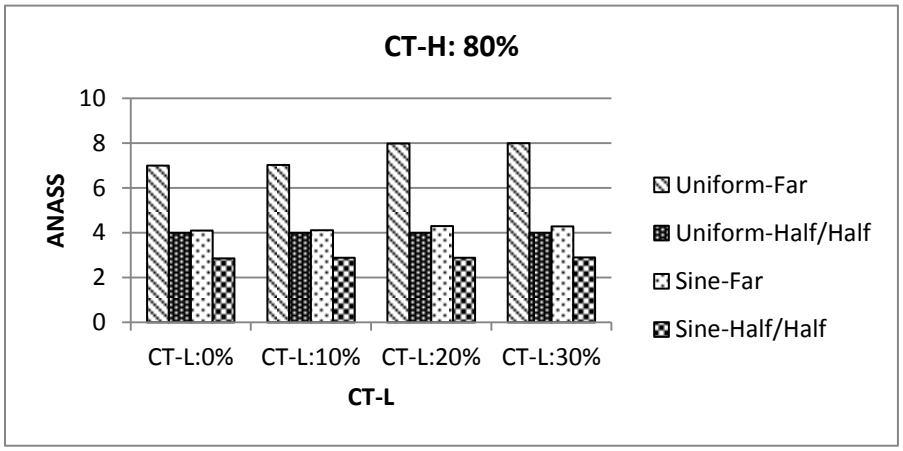
Similar to the previous results in Section 4.5.1.1, Figure 4-16 shows the same difference between Far traffic and Half-Far/Half-Near traffic and the same difference between Uniform traffic and Sine-Wave traffic. The reason is discussed in detail in Section 4.5.1.1.

4.5.2.2 The Impact of CT-L on Performance

The simulation results for analyzing the effect of CT-L on performance are shown in Figure 4-17.



(a) CT-H = 90%



(b) CT-H = 80%

Figure 4-17. The Impact of CT-L on Performance

Figure 4-17 (a) depicts the simulation results for a CT-H of 90%. The simulation results shown in Figure 4-17 (b) are the simulation results for a CT-H of 80%. The simulation results for CT-H of 90% do not show any significant change in performance when CT-L is varied. The reason is that GSSMS does not have a chance to deactivate a Spine switch when CT-H is 90%. As we can see, for a given CT-L, ANASS for a CT-H of 90% tends to be lower than that achieved with a CT-H of 80%. That means that the link utilizations in the simulation for CT-H of 90% never decrease below FUT-L. The

simulation results for CT-H of 80% demonstrate that ANASS increases while CT-L increases. The reason is that a lower CT-L value is easier to reach. For instance, consider a situation in which there are six active Spine switches in the network and all Leaf switches have one link with utilization lower than FUT-L. In this situation, if CT-L is 10%, the given number Nd is one, and Nd is two if CT-L is 30%. That means that when CT-L is 10%, GSSMS sets the timer for deactivating a Spine switch ON; when CT-L is 30%, the timer for deactivating a Spine switch is still OFF.

The change of CT-H does not have much impact on PF. The simulation results demonstrate that when CT-L is 0%, it only has a slight impact on the Sine-Wave Far traffic (PF is 0.0001).

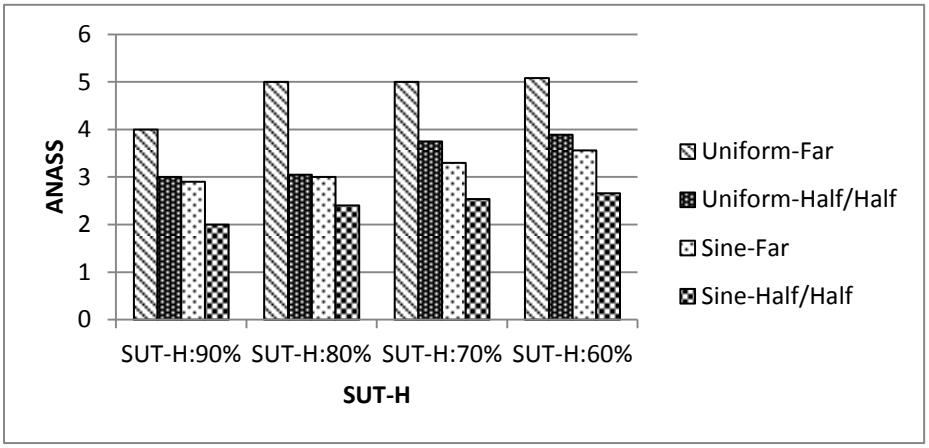
Similar to the previous results presented in Section 4.5.1.1, the results in Figure 4-17 demonstrate the same difference between different traffic patterns. The reason is discussed in detail in Section 4.5.1.1.

4.5.3 The Impact of SUTs on Performance

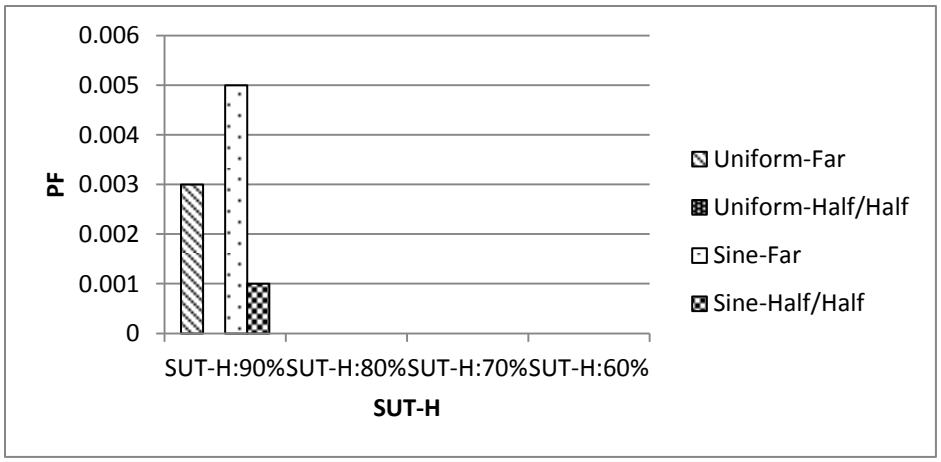
As in the case of FUTs, there are two SUTs: SUT-H and SUT-L. SUT-H is used for activating a Spine Switch in response to an increase in traffic. SUT-L is used for deactivating Spine switches in response to a decrease in traffic.

4.5.3.1 The Impact of SUT-H on Performance

Figure 4-18 illustrates the simulation results which demonstrate the impact of SUT-H.



(a) SUT-L = 20%



(b) SUT-L = 20%

Figure 4-18. The Impact of SUT-H on Performance

The simulation results shown in Figure 4-18 (a) demonstrate that for any given traffic model ANASS increases while SUT-H decreases. The reason is that with a low SUT-H, GSSMS allows the total traffic to decrease more during the time duration Tda without stopping the timer for activating a Spine switch. Therefore, GSSMS with a low SUT-H can activate a Spine switch more easily. For instance, consider a situation in which the traffic increases to 96% of a link capacity and then decreases to 70% of a link capacity within the time duration Tda. In this scenario, when the traffic increases to 96% of a link

capacity, the traffic can trigger the timer of activating a Spine switch for both GSSMS with SUT-H of 60% and GSSMS with SUT-H of 80%. However, when the traffic decreases to 70% of a link capacity, the timer of GSSMS with SUT-H of 80% turns to OFF (70% is lower than SUT-H of 80%). On the other hand, the timer of GSSMS with SUT-H of 60% is still ON because 70% is higher than SUT-H of 60%. As a result, after the time duration Tda, GSSMS with SUT-H of 60% activates a Spine switch while GSSMS with SUT-H of 80% does not.

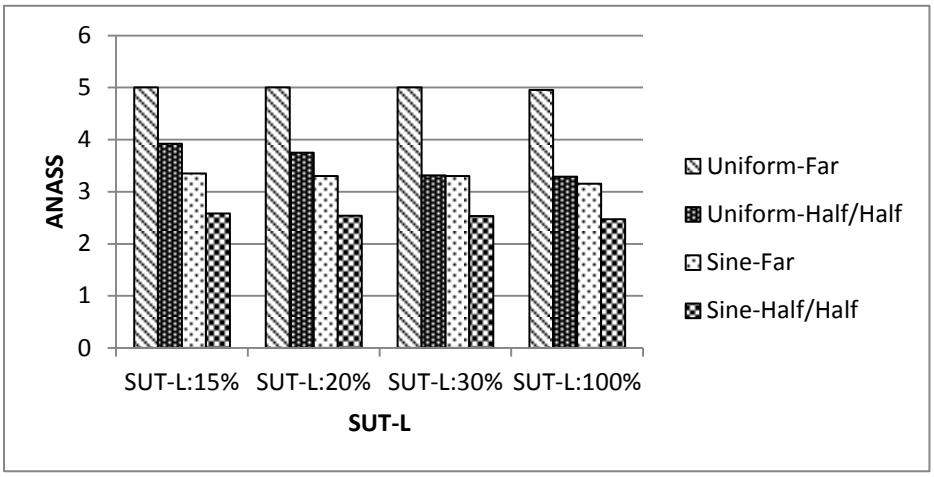
The simulation results shown in Figure 4-18 (b) illustrate that when SUT-H is 90% which equals to FUT-H, the failures occur. When SUT-H becomes equals to FUT-H, it means that once traffic decreases to the value below FUT-H, the timer for activating a Spine switch stops. Therefore, with a high SUT-H, GSSMS identifies traffic increase as instantaneous traffic sometimes. As a result, some times GSSMS cannot activate a Spine switch in time, and failures occur.

Similar to the previous results described in Section 4.5.1.1, Figure 4-18 shows the same difference between Far traffic and Half-Far/Half-Near traffic and the same difference between Uniform traffic and Sine-Wave traffic. The detailed explanation is in Section 4.5.1.1.

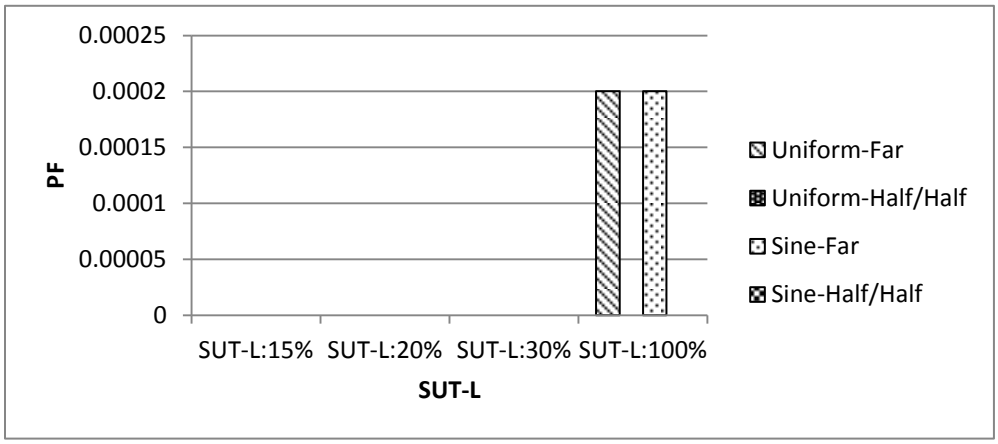
4.5.3.2 The Impact of SUT-L on Performance

Figure 4-19 illustrates the simulation results which demonstrate the impact of FUT-L on

performance.



(a) SUT-H = 70%



(b) SUT-H = 70%

Figure 4-19. The Impact of SUT-L on Performance

Figure 4-19 (a) shows that ANASS decreases while SUT-L increases except for the Uniform-Far traffic case. The reason is that with a high SUT-L, GSSMS allows the traffic to increase more during the time duration Tdd without stopping the timer for deactivating a Spine switch. Thus, GSSMS with a high SUT-L deactivates a Spine switch less aggressively. For the Uniform-Far traffic case, the utilizations of all links are much higher than FUT-L. GSSMS does not have chance to deactivate a Spine switch for the

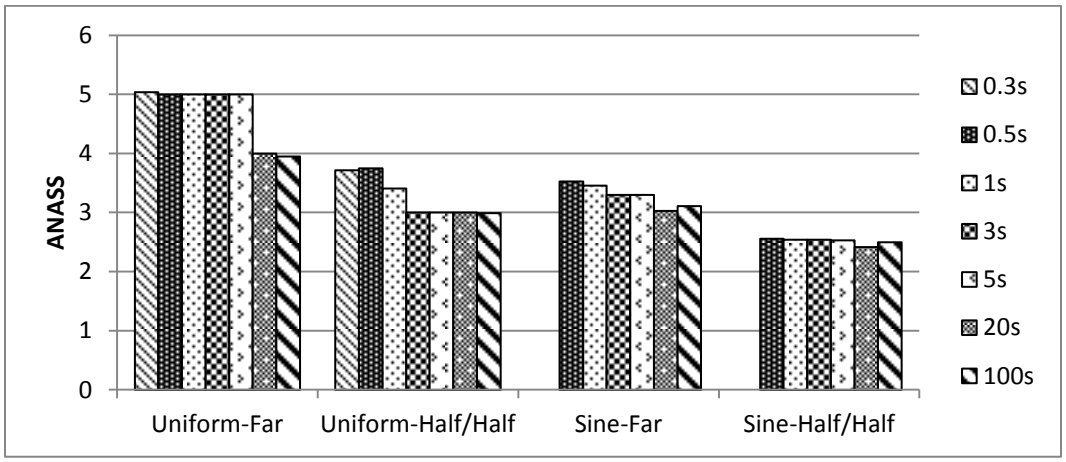
Uniform-Far traffic case. Therefore, the simulation results for Uniform-Far traffic cannot show the impact of SUT-L.

The simulation results shown in Figure 4-19 (b) illustrate that when SUT-L is 100%, the Far traffic case has a small number of failures. With a 100% SUT-L, once the timer for deactivating a Spine switch is triggered, GSSMS can deactivate a Spine switch after the time duration Tdd. That means that it is possible that GSSMS deactivates a Spine switch even when the Spine switch is needed to handle the traffic. The inappropriate deactivation of Spine switches causes failures.

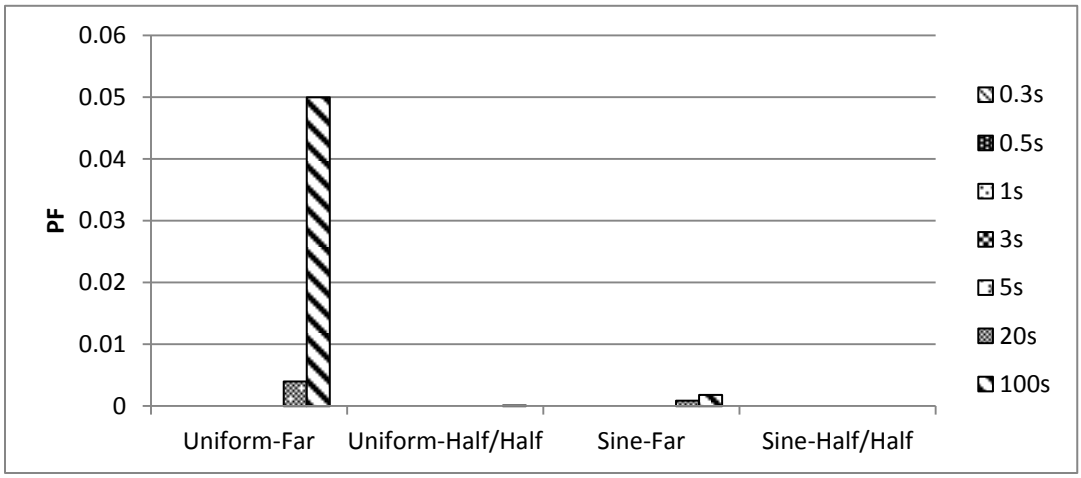
Figure 4-19 reveals the same difference between the Far traffic and the Half-Far/Half-Near traffic as demonstrated in Section 4.5.1.1. In addition, the figure also shows the same difference between the Uniform traffic and the Sine-Wave traffic as previous results. The reason is discussed in detail in Section 4.5.1.1.

4.5.4 The Impact of the Time Duration on Performance

The simulation result showing the impact of the Time Duration Tda is presented in Figure 4-20.



(a)



(b)

Figure 4-20. The Impact of the Time Duration on Performance

The Time Durations Tda and Tdd are used to filter out the instantaneous traffic. Tdd is held at twice the value of Tda in the simulation. That Tdd is longer than Tda is good for the situation that the traffic increases in a short time after it decreases. The simulation results shown in Figure 4-20 (a) illustrate that ANASS decreases as Tda increases for the Uniform traffic. For Uniform traffic, ANASS for the small Tda is higher than that for the large Tda. The reason is that with small Tda, GSSMS activates a Spine switch for the short time traffic increase while GSSMS with large Tda does not activate a Spine switch

for such a short time traffic increase. For Sine-Wave traffic, when Tda is smaller than 100 seconds, ANASS decreases as Tda increases. The reason is same with Uniform traffic. The simulation results shown in Figure 4-20 (b) demonstrate that when Tda is 100 seconds or 20 seconds, the Far traffic case has failures. The reason is that if Tda is too long, and GSSMS can be too late for activating a Spine switch and some packets may be dropped.

Similar to the previous results, Figure 4-20 demonstrates the same difference between the Far traffic and the Half-Far/Half-Near traffic and the same difference between the Uniform traffic and the Sine-Wave traffic. The detailed explanation is provided in Section 4.5.1.1.

There is another difference between Uniform traffic and Sine-Wave traffic shown in Figure 4-20 (a). As Tda increases, ANASS of GSSMS for Uniform traffic decreases while ANASS of GSSMS for Sine-Wave traffic decreases first and then increase slightly. For Sine-Wave traffic, ANASS increases when Tda is 100 seconds. For the Sine-Wave traffic, the traffic rate for each traffic source changes with time as a sine wave, and the number of the active Spine switches is the minimum number two for half of the simulation time because of the low traffic rate part (the traffic rate is lower than half of the maximum traffic rate) in Sine-Wave traffic. When Tda is 100 seconds, the number of active Spine switches decreases to two 200 seconds after the decrease of the traffic rate. The 200 seconds delay is the reason for the increase of ANASS.

4.5.5 The Impact of the ON/OFF Duration on Performance

The impact of the ON/OFF duration on performance is presented in this section.

4.5.5.1 The Impact of the ON Duration on Performance

The simulation results showing the impact of the ON Duration are presented in Figure 4-21.

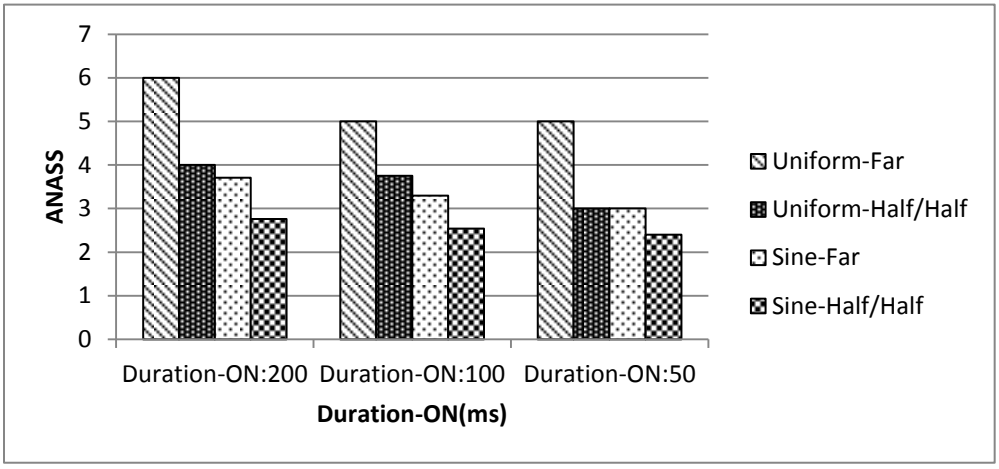


Figure 4-21. The Impact of the ON Duration on Performance (Duration-OFF = 20ms)

The simulation results demonstrate that, as expected, ANASS decreases while the ON Duration decreases. The reason is that when the ON Duration decreases, the number of the concurrent flows decreases, thus the total traffic in the datacenter network decreases. ANASS decreases when the total traffic decreases.

As the previous results, Figure 4-21 shows the same difference between the Far traffic and the Half-Far/Half-Near traffic and the same difference between the Uniform traffic and the Sine-Wave traffic. The reason is discussed in detail in Section 4.5.1.1.

The simulation results showing the impact of the OFF Duration are presented in

Figure 4-22.

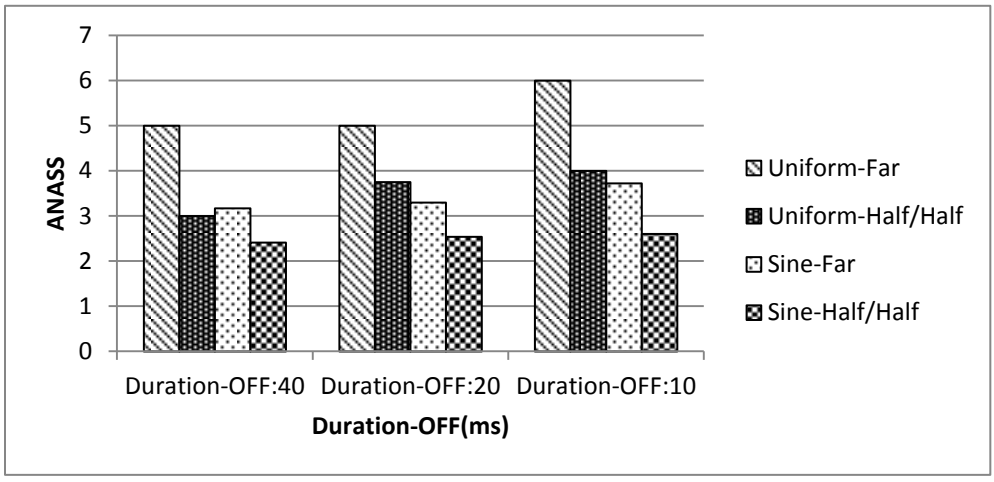


Figure 4-22. The Impact of the OFF Duration on Performance (Duration-ON = 100ms)

The simulation results demonstrate that ANASS increases while the OFF Duration decreases. The reason is that when the OFF Duration decreases, the number of the concurrent flows increases, thus the total traffic in the datacenter network increases. As a result, ANASS increases when the total traffic increases.

Figure 4-22 also shows the same difference between the Far traffic and the Half-Far/Half-Near traffic and the same difference between the Uniform traffic and the Sine-Wave traffic as the previous results presented in Section 4.5.1.1. The detailed explanation is provided in that section.

4.5.6 The Impact of the Number of Leaf Switches on Performance

The simulation results showing the impact of the number of Leaf switches (NLS) are presented in Figure 4-23.

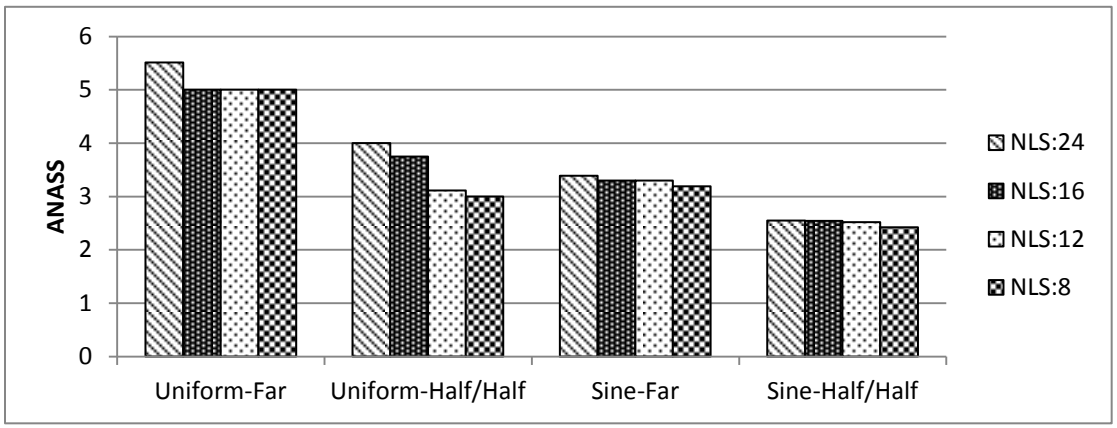


Figure 4-23. The Impact of the Number of Leaf Switches on Performance

The simulation results illustrate that ANASS decreases as NLS decreases. The reason is that with more Leaf switches, the probability of traffic concentrating on one Leaf switch is higher. For instance, assuming that the traffic on each Leaf switch is i.i.d and the probability of the traffic on one Leaf switch exceeding the link capacity is p, eight Leaf switches in the network means that the total probability of the traffic on one or more Leaf switches exceeding the link capacity is $8 \times p$. If the network has twenty four Leaf switches, the total probability of the traffic exceeding the link capacity is $24 \times p$. GSSMS activates a Spine switch as long as any one Leaf switch satisfies the requirement of activating an additional Spine switch. Therefore, if the total probability of the traffic exceeding the link capacity increases, the probability of GSSMS activating a Spine switch increases.

Figure 4-23 demonstrates the same difference between different traffic patterns as the previous results discussed in detail in Section 4.5.1.1.

4.6 Guidelines for Choosing the Control Parameters

A short description of a set of guidelines for choosing the various parameters that control the behavior of GSSMS is provided in this section. Only those parameters that have demonstrated an impact on performance during the simulation studies are considered. Although the exact values of these parameters depends on the other system and workload parameters including the pattern of network traffic in the datacenter, a general set of rules that can aid in making such parameter choices is discussed. A simulation-based study can be used to choose appropriate parameter values for a given datacenter.

FUT-H and SUT-H: jointly control the activating of Spine switches on the system. If FUT-H is too high (100% in the simulation), PF would be very high (0.8 in the simulation). FUT-H is to be chosen to be low enough such that the desired value of PF is achieved. With a low FUT-H, ANASS would be higher than ANASS of GSSMS with a high FUT-H, and thus GSSMS with a low FUT-H would save less energy than GSSMS with a high FUT-H. Therefore, FUT-H should be chosen to be the value that can lead to desired PF and the lowest possible value of ANASS. If SUT-H is too high, e.g. equal to FUT-H, it would be very hard for GSSMS to activate additional Spine switch in time and lead to an increase in PF. If SUT-H is too low (e.g., lower than 50%), GSSMS would activate unnecessary Spine switch. Thus, SUT-H should be chosen such that the desired PF and the lowest possible value of ANASS for the selected FUT-H can be achieved at the same time.

FUT-L and SUT-L: jointly control the deactivation of the Spine switches on the system. Once again, values of these two parameters need to be chosen in such a way that PF does not exceed the desired value and ANASS is minimized. Note that higher values for both of these parameters are expected to lead to a lower ANASS. However, the system administrator needs to be aware between the potential tradeoff between ANASS and PF while making a choice of these parameters.

Tda and Tdd: smaller values of these parameters tend to increase the sensitivity of GSSMS to change in traffic intensity, but may also lead to frequent changes (e.g., the number of Spine switches changes twice or more in half an hour) in the number of Spine switches leading to an increase in system overhead. Values of these parameters that strike an effective compromise between sensitivity and undesirable frequency of changes in the number of Spine switches need to be used.

4.7 Summary

This chapter presents the simulation environment and results for GSSMS in detail. The simulation results in this chapter demonstrate that GSSMS can work well for the three types of input traffic patterns. This chapter also presents the simulation results that reveal the impact of parameters on performance. The explanations of the impact are also provided. A set of guide-lines that can help the system administrator to choose the parameter values is discussed.

Chapter 5: Conclusion and Future work

This chapter concludes this thesis and discusses some potential future work.

5.1 Conclusion

Nowadays, datacenters consume a huge amount of energy. Researches have devoted their efforts to improve the efficiency of servers. However, not enough attention has been spent on the efficiency of datacenter networks. This thesis focuses on energy saving for datacenter networks. To save energy consumed by datacenter networks, this thesis choses Spine-Leave topology, which was introduced in Chapter 2, as the datacenter network topology.

This thesis proposed GSSMS, a system that can dynamically manage the number of active Spine switches according to the traffic of the datacenter network. The purpose is to save energy consumption without significant decrease in reliability when the traffic is low. The GSSMS algorithm used six parameters to determine the number of active Spine switches in a datacenter network based on the network traffic. The thresholds FUTs and SUTs are used to control the activating and deactivating of Spine switches. The time duration Tda and Tdd are used to avoid frequent changes in the number of active Spine switches. Unlike the traditional datacenter network, which has a fixed number of switches, GSSMS uses two Spine switches when the network traffic is lower than the capacity of two Spine switches, and activates additional Spine switches when there is not enough Spine switches to handle the increased traffic in the datacenter network. Chapter 3

presented the algorithms used in GSSMS and a detailed process of GSSMS. Chapter 4 demonstrated the detailed information about the simulation for GSSMS. Chapter 4 also presented the simulation results and explained all simulation results. This thesis considered three traffic patterns, which are adapted from [Heller10], as the input traffic. The three traffic patterns are Uniform traffic, Sine-Wave traffic and Random traffic. Both the Uniform traffic and the Sine-Wave traffic in turn, have three types – Far traffic, Near traffic and Half-Far/Half-Near traffic. The simulation results show that GSSMS can work efficiently for the three input traffic patterns. For the Uniform traffic and the Sine-Wave traffic, in comparison to the Far traffic, GSSMS can save more energy for the Half-Far/Half-Near traffic, because the Half-Far/Half-Near traffic has a lower traffic routing through Spine switches. Compared with Uniform traffic, GSSMS can save more energy for Sine-Wave traffic because the traffic routing through Spine switches of Sine-Wave traffic is lower than Uniform traffic's traffic routing through Spine switches. Specifically, GSSMS can save energy (up to 63% in a datacenter with 8 Spine switches) with a slight increase (0.08) in PF or reduce PF significantly by dynamically adjusting the number of active Spine switches. Similar conclusions are expected when datacenter networks have different number of Leaf switches and Spine switches.

5.2 Future work

GSSMS has some limitations. There are some directions that can be further researched to improve GSSMS.

GSSMS in this thesis activates only one Spine switch at a time. If the traffic burst exceeds the link capacity, activating one Spine switch may not be enough to hand the traffic burst. The activating algorithm which can activate more than one Spine switch according to the increase rate of the traffic should be considered. For instance, when the traffic increase is higher than half of the link capacity within 0.5 seconds, GSSMS can activate two Spine switches after a time interval of Tda.

The time duration Tda is a very important parameter for GSSMS. A small Tda works efficiently for big traffic bursts, whereas a large Tda works efficiently for the stable traffic. That means that if the traffic is complex (e.g., when the traffic is increasing slowly and the traffic also has spikes which can last longer than Tda occasionally), it is very hard to find a Tda that works efficiently for the traffic. Using a variable Tda may be investigated for further studies. With such a Tda, GSSMS would change Tda appropriately based the rate of change of traffic.

There are some other factors that should be considered. For instance, the average response time of request in the network with GSSMS and the impact of the size of buffer, which warrant further investigation.

References

[Abts10] D. Abts, M. R. Marty, P. M. Wells, P. Klausler, and H. Liu, “Energy Proportional Datacenter Networks”, Proc. of the 37th Annual International Symposium on Computer Architecture(ISCA), New York, NY, USA , June 2010, pp. 338-347

[Alizadeh13] M. Alizadeh, T. Edsall, “On the Data Path Performance of Leaf-Spine Datacenter Fabrics”, Proc. of IEEE 21st Annual Symposium on High-Performance Interconnects (HOTI), San Jose, CA, USA, Aug. 2013, pp. 71 - 74

[Beck13] Pall Beck, Peter Clemens, Santiago Freitas, Jeff Gatz, Michele Girola, Jason Gmitter, Holger Mueller, Ray O'Hanlon, Veerendra Para, Joe Robinson, Andy Sholomon, Jason Walker, Jon Tate, “IBM and Cisco: Together for a World Class Data Center”, IBM Redbooks, July 31, 2013.

[Benson10] T. Benson, A. Akella, D. Maltz, “Network Traffic Characteristics of Data Centers in the Wild”, Proc. of the 10th ACM SIGCOMM Conference on Internet Measurement, New York, NY, USA, 2010, pp. 267-280

[Calabretta13] N. Calabretta, R. Centelles, S. Lucente, H. Dorren, “On the Performance of a Large-Scale Optical Packet Switch Under Realistic Data Center Traffic”, Optical Communications and Networking, Volume 5 Issue 6, June 2013, pp. 565 – 573

[Carrega12] A. Carrega, S. Singh, R. Bruschi, R. Bolla, “Traffic Merging for Energy-Efficient Datacenter Networks”, Proc. of Performance Evaluation of Computer and Telecommunication Systems (SPECTS), Genoa, Italy, July 2012, pp. 1-5

[Cisco12] Cisco, “Cisco’s Massively Scalable Data Center Overview” [Online], December 21, 2012, Available: http://www.cisco.com/c/en/us/solutions/enterprise/data-center-designs-data-center-networking/landing_msdc.html, last accessed in June, 2014

[CLOUDS] The Cloud Computing and Distributed Systems (CLOUDS) Laboratory, University of Melbourne, http://www.cloudbus.org/cloudsim/. Last accessed in July 2014.

[C.Lee12] Chankyun Lee and June-Koo Kevin Rhee, “Traffic Off-Balancing Algorithm: Toward Energy Proportional Datacenter Network”, Proc. of Opto-Electronics and Communications Conference (OECC), Busan, Korean, July 2012, pp. 409 - 410

[Dell12] Dell, “Distributed Core Architecture Using Dell Networking Z9000 and S4810 Switches” [Online], Available: http://www.dell.com/learn/us/en/555/shared-content~solutions~en/documents~distributed-core-architecture-using-z9000-s4810.pdf, last accessed in June 2014

[E.Lee12] E. K. Lee, H. Viswanathan, and D. Pompili, “VMAP: Proactive Thermal-aware Virtual Machine Allocation in HPC Cloud Datacenters”, Proc. of High Performance Computing (HiPC), Dec. 2012, pp. 1-10

[Fan07] X. Fan, W. Weber, and L. A. Barroso, “Power Provisioning for a Warehouse-sized Computer”, Proc. of the 34th Annual International Symposium on Computer Architecture, New York, NY, USA, May 2007, pp. 13-23

[Goudarzi12] H. Goudarzi and M. Pedram, “Energy-Efficient Virtual Machine

Replication and Placement in a Cloud Computing System", Proc. of IEEE Cloud Computing (CLOUD), Honolulu, HI, USA , June 2012, pp. 750-757

[Greenberg08] A. Greenberg, J. Hamilton, D. A. Maltz, P. Patel, "The Cost of a Cloud: Research Problems in Data Center Networks", Newsletter of ACM SIGCOMM Computer Communication Review, Volume 39 Issue 1, January 2009, pp. 68-73

[Hasan12] Masum Z. Hasan, Edgar Magana, Alexander Clemm, Lew Tucker and Sree Lakshmi D. Gudreddi. "Integrated and Autonomic Cloud Resource Scaling". IEEE Network Operations and Management Symposium, April 2012, pp. 1327-1334.

[Hedlund12] Brad Hedlund, "Basic introduction to the Leaf/Spine data center networking fabric design" [Online Video], Available: http://bradhedlund.com/2012/10/24/video-a-basic-introduction-to-the-leafspine-data-center-networking-fabric-design/, 24 October, 2012, last accessed in June 2014

[Heller10] B. Heller, et al., "Elastic Tree: Saving Energy in Data Center Network", Proc. of USENIX Conference on Networked Systems Design and Implementation, San Jose, CA, April 2010

[IBM] CPLEX Optimizer, http://www.ilog.com/products/cplex/, last accessed in July 2014

[John10] Nicholas John, "A Simpler Data Center Fabric Emerges for The Age of Massively Scalable Data Centers" [Online], June 2010, Available: http://lippisreport.com/2010/07/a-simpler-data-center-fabric-emerges-for-the-age-of-mass

ively-scalable-data-centers/, last accessed in June 2014

[Koomey11] Jonathan Koomey. "Growth in Data center electricity use 2005 to 2010" [Online]. Analytics Press. August 1, 2011 Available: http://www.analyticspress.com/datacenters.html last accessed in June, 2014

[Liu12] N. Liu, Z. Dong, R. Rojas-Cessa, "Task and Server Assignment for Reduction of Energy Consumption in Datacenters", Proc. of IEEE International Symposium on Network Computing and Applications (NCA), Aug. 2012, pp. 171-174

[Mahadevan10] P. Mahadevan, S. Banerjee, and P. Sharma, "Energy Proportionality of an Enterprise Network", Proc. of the 37th Annual International Symposium on Computer Architecture (ISCA), New York, NY, USA, June 2010, pp. 338-347

[Meisner09] D. Meisner, B. T. Gold, and T. F. Wenishch, "PowerNap: Eliminationg Server Idle Power", Proc. of the 14th International Conference on Architectural Support for Programming Languages and Operating Systems, New York, NY, USA, March 2009, pp. 205-216

[Miercom13] Miercom Report, "Cisco Catalyst 2960-X/2960-XR Switches" [Online], November 2013, Available: http://miercom.com/pdf/reports/20131112.pdf, last accessed in December, 2014

[Pedram12] Massoud Pedram, "Energy-Efficient Datacenters", Computer-Aided Design of Integrated Circuits and Systems, vol. 31, no. 10, Oct. 2012, pp. 1465 – 1484

[Salehi12] M. A. Salehi, P. R. Krishna, K. S. Deepak, R. Buyya, "Preemption-Aware

Energy Management in Virtualized Data Centers", Proc. of IEEE Cloud Computing (CLOUD), Honolulu, HI, USA , June 2012, pp. 844-851

[Taheri11] M. M. Taheri and D. Zamanifar, "2-Phase Optimization Method for Energy Aware Scheduling of Virtual Machines in Cloud Data Centers", Proc. of Internet Technology and Secured Transactions (ICITST), Abu Dhabi, Dec. 2011, pp. 525-530

[Tolia08] N. Tolia, Z. Wang, M. Marwah, C. Bash, P. Ranganathan, and X. Zhu, "Delivering Energy Proportionality with Non Energy-Proportional Systems: Optimizing the Ensemble", Proc. of the Conference on Power Aware Computing and Systems, Berkeley, CA, USA, 2008, pp. 2-2

[Vasic11] N. Vasic, D. Novakovic, S. Shekhar, P. Bhurat, M. Canini, D. Kostic, "Identifying and Using Energy-Critical Paths", Proc. of the Seventh Conference on Emerging Networking Experiments and Technologies, New York, NY, USA, Dec 2011