Li et al. Journal of Cloud Computing: Advances, Systems and Applications (2016) 5:9
DOI 10.1186/s13677-016-0058-8

Journal of Cloud Computing: Advances, Systems and Applications

RESEARCH Open Access

Green spine switch management for datacenter networks

Xiaolin Li, Chung-Horng Lung* and Shikharesh Majumdar

Abstract

Energy consumption for datacenter has grown significantly and the trend is still growing due to the increasing popularity of cloud computing. Datacenter networks (DCNs), however, are starting to consume a greater portion of overall energy in comparison to servers used in datacenters due to advanced virtualization techniques. On the other hand, devices in a DCN often remain under-utilized. There are various DCN architectures. This paper proposes an approach called Green Spine Switch Management System (GSSMS) for Spine-Leaf topology based DCNs. The objective of the approach is to reduce energy consumption used by the network for a Spine-Leaf topology-based datacenter. The primary idea of GSSMS is to monitor the dynamic workload and only keep Spine switches that are necessary for handling the current network traffic. We have developed an adaptive management system to control the number of Spine switches in a Spine-Leaf DCN for efficient energy consumption. Further, we have performed extensive simulation using CloudSim for a number of scenarios. The simulation results demonstrate that our proposed GSSMS can effectively save energy by as much as 63 % of the energy consumed by a datacenter comprising a fixed static set of Spine switches.

Keywords: Datacenters, Datacenter Networks, Spine-Leaf Topology, Resource Management, Energy Efficiency

Introduction

The usage of third party datacenters for provisioning of services is becoming more widespread. More and more small and medium scale enterprises (SMEs) choose to host their services on datacenters managed by datacenter providing companies because of the convenience and decrease in cost in comparison to acquiring and maintaining their own equipment. The service providers endeavor to support reliable, secure, scalable and multi-tenant services with massive datacenters. To serve more and more tenants, the size of a datacenter has to increase. While the size of such a datacenter increases continually, the power consumed by datacenter also increases dramatically. According to [1], from 2000 to 2005, electricity consumed by world datacenters has doubled, and from 2005 to 2010, there is a 56 % increment in power consumption for datacenters across the world, and 36 % for the US datacenters. The report also indicates that, in 2010, electricity used by datacenters 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.

As mentioned in [1], the deceleration of the growth in electricity usage from 2005 is caused by the increased deployment of virtualization in datacenters and the industry's efforts to improve efficiency of datacenter facilities. The sources of the inefficiencies in datacenters include energy non-proportional servers and over-provisioned servers as well as the power infrastructure. The energy non-proportional servers [3] cannot control their energy consumption in accordance with the workload. In other words, servers always consume an almost fixed amount of energy irrespective of whether the workload is low or high. With over-provisioned servers and the supporting power infrastructure, e.g., cooling systems, that are used for handling (the temporary) peak workloads, datacenter devices typically remain under-utilized for most of the time. Researchers have performed a great deal of research on reducing energy consumed by servers and their cooling systems. On the

* Correspondence: chlung@sce.carleton.ca
Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada

other hand, the energy consumed by datacenter *networks* (DCNs) has not received adequate attention although it is 10–20 % of datacenter's total power [2].

Previous studies show that as servers are becoming more energy-proportional, DCNs will consume a greater proportion of the overall power. Although a DCN with a fat tree topology consumes only 12 % of overall power when the datacenter is at its full utilization, with fully energy-proportional servers the the network will consume nearly 50 % of overall power when datacenter is 15 % utilized [3]. Devices in a DCN are typically under-utilized. The utilization of Edge links, aggregation links and core links, remains below 10 % during 95 % of the time, and does not exceed 30 % for more than 99 % of the time [4]. Hence, an effective management of network devices can save energy, and the most efficient way of saving energy consumed by networks is minimizing the number of active network devices [2].

To minimize the number of active switches, a DCN topology must support the ability to shift the traffic on one switch to any other switch on the same layer. Access switches 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 research focuses on aggregation switches in a DCN with a topology called Spine-Leaf (as shown in Fig. 1). A detailed discussion of the Spine-Leaf topology is presented in the next section.

A preliminary report described the basic idea of a novel Green Spine Switch Management System (GSSMS) [20] for Spine-Leaf topology [5, 6] based DCNs. The aim of GSSMS is saving energy by dynamically controlling the number of active Spine switches and maintaining only a minimal set of active Spine switches that is necessary to handle the current workload. This paper extends the experiments and thoroughly investigates the effect of various system parameters.

The main contributions of this paper are summarized:

- A new technique for energy aware Spine switch management is introduced. The technique comprises algorithms to control the number of active Spine switches according to current network traffic.
- A simulation-based performance analysis of the technique for three different traffic patterns is presented.
- Insights into the relationship between various system as well as workload parameters and performance are described.
- A simulation-based analysis of the impact of the various parameters controlling the behavior of the algorithms on performance is presented.
- A set of guidelines for choosing the various parameters controlling the behavior of the algorithms is discussed.

The rest of the paper is organized as follows. The next section presents the description of the Spine-Leaf topology and some approaches for saving energy in

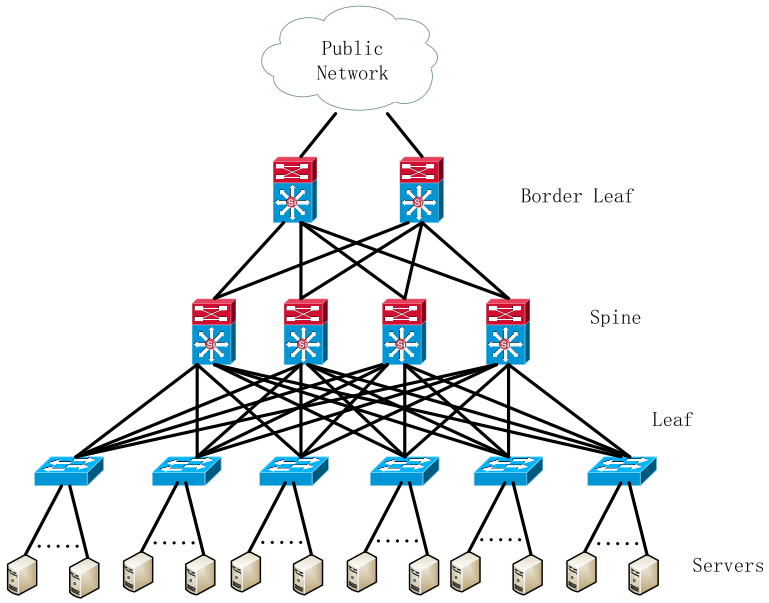

Fig. 1 The Spine-Leaf Topology [5]

datacenters. In Section III the algorithm of GSSMS is introduced. Section IV discusses the simulation methodology and results. The conclusions and future works are presented in Section V.

Related works

Spine-leaf topology

The Spine-Leaf topology, proposed by Cisco [5, 7], is used in massively scalable data centers. As shown in Fig. 1, the Spine-Leaf topology has two types of switches: Spine switch and Leaf switch. Spine switches work as aggregation switches in the traditional 3-tier network architecture. They only connect with Leaf switches and do not connect directly with servers. Every Spine switch connects with all Leaf switches. Leaf switches are access switches. They connect with Spine switches and a number of servers. As shown in Fig. 1, there are also some Leaf switches called Border Leaf switches which are responsible for connecting to public networks.

The Spine-Leaf topology has many advantages that include the following:

- Overcoming oversubscription: In the traditional 3-tier datacenter network, one access switch only connects to one core switch; therefore, the traffic capacity between servers under different access switches depends on the link capacity between the access layer and the core layer. Sometimes the available bandwidth between the access layer and the core layer is not enough to handle traffic spikes, which may lead to unexpected oversubscription. One way to mitigate the oversubscription problem is to use higher link capacity between the access layer and the core layer than that between servers and the access layer. However, oversubscription using the tree-based topology can still exist and result in the blocking server-to-server connectivity problem which can have severe performance impact on data centers due to increasing east-west traffic. In contrast, in the Spine-Leaf topology, one Leaf switch connects to multiple Spine switches; therefore, the traffic capacity is not limited by the link capacity of a single link between the Leaf layer and the Spine layer. When one Spine switch alone is unable to handle the traffic spikes, there are other Spine switches available to share the traffic load. With high-performance switches incorporating high link capacity, a Spine switch can connect to hundreds of Leaf switches (e.g., 384 or 768) and a Leaf can connect to 32 Spine switches, which give rise to an oversubscription ratio of up to 1:1 (i.e., no oversubscription) [7, 21]. The Spine-Leaf topology represents a reasonable balance among various design considerations, including development cost, east-west vs. north-south traffic, complexity and the number of cables needed [21].
- Providing a predicable amount of delay: The topology only consists of two layers, the Spine layer and the Leaf layer. Further, a Leaf switch is connected to each and every Spine switch. As a result, the delay or latency for traffic inside the data center, i.e., east-west traffic, is predicable and a non-blocking server-to-server connectivity can be realized [7].
- Improving robustness for the network: Because 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 routed through the failed Spine switch can be distributed to other Spine switches [5].
- Making scaling out easy on datacenters: If the datacenter providers need additional servers in the datacenter, 100 servers or lower for example, they can simply add one or two new Leaf Switches and new servers without making any change to existing Spine switches or servers in the network. If more than 100 servers need to be added or when oversubscription occurs, an additional Spine switch may be added and connected to every Leaf switch. Alternatively, multiple Spine switches may be added at the same time, if needed. The process can be repeated until the either port capacity of a Spine switch gets exhausted or oversubscription becomes an issue. If the ports for Spine switches are exhausted, then as suggested in [7], an additional super Spine switch connecting Spine switches may be considered.

To the best of our knowledge, there is no research reported in the literature addressing the energy reduction for Spine-Leaf topology. However, the Spine-Leaf topology has many advantages, such as simplifying VM placement, reducing network failures and making datacenters easy to scale out [5].

Some protocols used in the traditional 3-tier DCN, such as Spanning Tree Protocol (STP), are not suitable for the Spine-Leaf topology. Instead, for the Spine-Leaf topology, datacenter providers can use multipath to scale bandwidth [5]. 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.

Energy saving approaches in datacenters

Researchers have proposed some algorithms for saving energy in DCNs. These algorithms save energy by minimizing the number of network devices, managing port rate or using traffic engineering schemes.

In [2], the authors proposed Elastic Tree which is a system for minimizing the DCN power consumption by shutting down unneeded switches and links. Elastic Tree chooses a subset of network devices that must be active to achieve the required performance and fault tolerance objectives based on the traffic matrix, DCN topology and the power model for each switch. Kakadia et al. [8] proposed a SDN based method for the fat tree topology to incrementally calculate the network devices required to support the current load. Preter et al. [9] presented a spanning tree based algorithm in the SDN paradigm to reduce the network electric energy consumption. Both the fat tree topology and the spanning tree based approaches suffer from the blocking of a server-to-server communications problem at high traffic, rendering the system inefficient for handling the increasing level of east-west network traffic inside of a data center. As stated in Section I, the blocking problem can be mitigated or avoided using the Spine-Leaf topology.

Both [3] and [10] proposed to save energy by managing the port rate. Abts et al. [3] proposed a topology called Flattened Butterfly (FBFLY) and exploited managing the port rate with FBFLY to save energy consumed by the DCN. The authors combined load prediction with link's dynamic traffic range to ensure that each link has the appropriate link speed to satisfy the traffic load. The approach presented in [10] focused on saving energy by traffic merging with FBFLY. The authors presented the design of a hardware called traffic merge network, which 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.

Traffic engineering methods have been used in [11] and [12] to save energy. Vasic et al. [11] proposed a system which pre-computes energy-critical paths off-line for the network, and then uses online traffic engineering to deactivate and activate network elements on demand. In [12], the authors used a traffic off-balancing algorithm which behaves oppositely to the load-balancing algorithm to minimize the number of active network devices. Shi et al. [13–15] have proposed approaches managing resource in wireless network.

In this paper, we propose an algorithm that minimizes the number of active Spine switches to dynamically manage the number of Spine switches needed according to the traffic load in datacenters. Based on an investigation on energy consumption, a high end switch can save tremendous energy (e.g., 84 %) when it is in the Hibernation mode [19]. The main difference between our algorithm and aforementioned approaches is that our algorithm uses the Spine-Leaf topology. Our current focus is not specifically for SDN. But the concept can be integrated with SDN, as our approach also makes use of a controller to Spine switches, as shown in Section III.

Green spine switch management system

The basic concepts underlying GSSMS are summarized. When the network traffic increases and the active Spine switches do not have enough available bandwidth for the traffic, GSSMS activates an additional Spine switch that is available; when the network traffic drops and one or more active Spine switches are idle, GSSMS deactivates those Spine switches for energy saving.

GSSMS comprises of four main modules: Routing, Spine Switch Controller, Network Monitor and Power

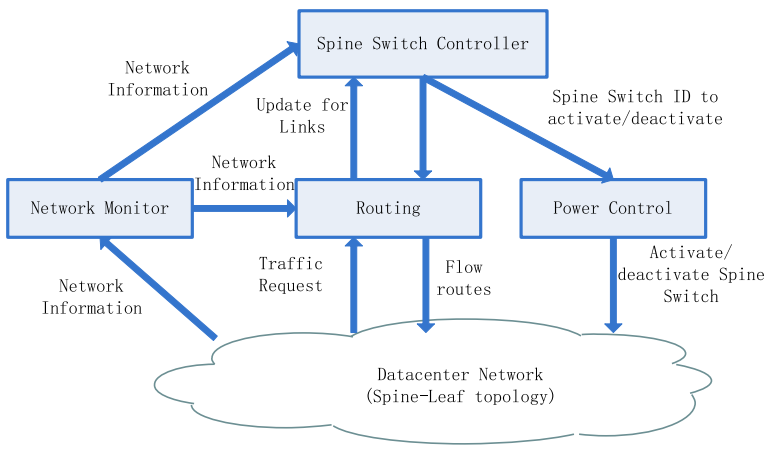

Fig. 2 The Green Spine Switch Management System Diagram

Control, as shown in Fig. 2. 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 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. The Network Monitor module is responsible for collecting network information. The Power Control module is in charge of toggling the power states (active or sleep) of Spine switches.

Routing module

The Routing Module selects the active Spine switch with the highest link utilization (the link connecting the Spine and the Leaf switches) that has enough available bandwidth to handle the new flow. The algorithm is shown in Algorithm 1.

The Routing Module's inputs are the requests coming from the DCN. The requests are of two types: (i) New Flow Request – the source Leaf switch asks for bandwidth reservation for a new flow and (ii) End Flow Request – the source Leaf switch asks to release the bandwidth reservation for the finished flow.

As shown in Algorithm 1, for both types of requests, the Routing Module determines whether the flow's destination is outside or inside the datacenter. Requests will be treated in different ways based on their destination. A flow with a destination outside the datacenter only needs bandwidth reservation on one link between a Spine switch and a Leaf switch. A flow with a destination inside the datacenter needs bandwidth reservation on two links between a Spine switch and two Leaf switches.

Algorithm 1: Routing

Input : Requests from the datacenter network

Output: flow routes and link utilization update

```
1.  receive a request
2.  switch(request)
3.      case New Flow:
4.          if (request.dest is outside datacenter)
5.              find the proper Spine switch
6.              make the bandwidth reservation on the link
7.          else find the proper Spine switch
8.              make bandwidth reservations on two links
9.          send the flow route & link utilization increase update
10.     case End Flow:
11.         if (request.dest is outside datacenter)
12.             delete the reservation on one link
13.         else delete the reservations on two links
14.         send the link utilization decrease update
```

For a New Flow request, the Routing Module finds the proper Spine switch to transmit the new flow. 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 select 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. Then the Routing Module makes the bandwidth reservations on both the links. Finally, the Routing Module sends the flow route with the Spine switch ID back to the source Leaf switch, and sends a message to update the Spine Switch Controller with the increased utilization of the link between source Leaf switch and the chosen Spine switch.

For an End Flow request, besides the destination of the flow, the Routing Module can also acquire information about the Spine switch which transmits the flow. 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 two links – one is between the source Leaf switch and the Spine switch and the other one is between the Spine switch and the destination Leaf switch. Finally, the Routing Module sends a link utilization update to the Spine Switch Controller.

Spine switch controller

The Spine Switch Controller module which is the key component of GSSMS uses eight parameters:

1. The high First Utilization Threshold (FUT-H)
2. The low First Utilization Threshold (FUT-L)
3. The high Control Threshold (CT-H)
4. The low Control Threshold (CT-L)
5. The high Second Utilization Threshold (SUT-H)
6. The low Second Utilization Threshold (SUT-L)
7. The Time Duration for Activating (Tda)
8. The Time Duration for Deactivating (Tdd)

FUT-H and FUT-L are used for starting the timer for activating or deactivating decision making, respectively. SUT-H and SUT-L are used to decide if the respective timer for activating and deactivating Spine switches should stop. CT-H and CT-L are ratios of the number of links between a Leaf switch and Spine switches that cross those aforementioned thresholds. Tda is the time duration used for activating a Spine switch, while Tdd is the time duration used for deactivating one or more Spine switches. Tda and Tdd are

used to avoid frequent changes in the number of Spine switches due to temporary spikes in traffic.

Algorithms for activating/deactivating spine switches

The algorithm for activating a Spine switch is presented in Algorithm 2. For each Leaf switch, when the link utilizations of a given number (*Na*) 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.

Algorithm 2: Increasing the number of active Spine switches

```
1. Na = round-up (number of active Spine switches × CT-H)
2. foreach Leaf switch
3.    if the link utilization of Na links connected with the given Leaf
      switch ≥ FUT-H
4.       start timer T
5.    if T ≥ Tda
6.       Activate a Spine switch
```

Algorithm 3: Decreasing the number of active Spine switches

```
1. Nd = round-up (number of active Spine switches × CT-L)
2. foreach Leaf switch
3.    if the link utilization of Nd links connected with the given Leaf
      switch ≤ FUT-L
4.    add the Leaf switch into timingmap
5. if timingmap.size() == number of Leaf switches
6.    start timer T
7.    foreach Leaf switch
8.       if the link utilization of Nd links connected with the given
         Leaf switch ≥ SUT-L and T ≤ Tdd
9.          remove the Leaf switch from timingmap and stop timer T
10. if timingmap.size() == number of Leaf switches for T ≥ Tdd
11.   deactivate at least Nd Spine switches
```

The algorithm for deactivating Spine switches is presented in Algorithm 3. 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 is put into a list *timingmap*. If all Leaf switches are in *timingmap* for the given time duration Tdd, a number of Spine switches are deactivated.

Operations of the spine switch controller

The main operation performed by the Spine Switch Controller comprises two steps: (i) making decision of starting the timer for activating/deactivating a Spine switch and (ii) checking the time duration.

Algorithm 4 presents the steps for making the decision for starting the timer used in Algorithm 2 and 3. The input of the Spine Switch Controller is the link utilization update, including link utilization increase update and link utilization decrease update.

Algorithm 4: Making decision of starting the timer

Input : link utilization updates

```
1. switch(update)
2.    case increase update:
3.       if the utilization of the updated link ≥ FUT-H and the
         number of busy links of the Leaf switch ≥ Na
           start the activating timer for the Leaf switch
5.       else if the utilization of the updated link ≥ SUT-L and the
         number of idle links of the Leaf switch < Nd
6.          remove the Leaf switch from timingmap and stop the
               deactivating timer if needed
7.    case decrease update:
8.       if the utilization of the updated link ≤ FUT-L and the
         number of idle links of the Leaf switch ≥ Nd
9.          add the Leaf switch into timingmap
10.         if (timingmap.size == number of Leaf switches)
11.         start the deactivating timer
12.      else if the utilization of the updated link ≤ SUT-H and the
         number of busy links of the Leaf switch < Na
13.         stop the activating timer for the Leaf switch
14.   Checking the time duration
```

For 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. Then the Spine Switch Controller updates the information related to the timer for deactivating Spine switches and switches OFF the timer if the requirement of deactivating is not satisfied any more.

Algorithm 5: Checking the time duration

Output: Spine switches' ID to activate or deactivate

```
1. foreach Leaf switch
2.    if the activating timer for the Leaf switch is on for Tda
3.       add a Spine switch and send the Spine switch's ID to the
         Routing Module and the Power Control
4. if the deactivating timer is on for Tdd
5.    foreach Leaf switch
6.       if (min_num > num_of_idle_links[leafswitchID])
7.          min_num= num_of_idle_links[leafswitchID]
8.    choose min_num Spine switches with the lowest utilization to
      deactivate
```

For a link utilization decrease update, the Spine Switch Controller updates the link utilization and checks if the timer for deactivating Spine switches should start. The Spine Switch Controller has one timer for deactivating Spine switches and a timingmap (shown in line 6&9 in Algorithm 4, 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. Then the Spine Switch Controller updates the information related with the timer for activating a Spine switch and switches OFF the timer if the requirement of activating is not satisfied any more.

After confirming the timer's state, the Spine Switch Controller checks the time duration, as shown in Algorithm 5. Each Leaf switch has a timer for activating a Spine switch. If any one of those timers is on and Tda is exceeded, the Spine Switch Controller makes the decision of adding a new one. The Spine Switch Controller sends the Spine switch's ID to the Power Control to inform it that the Spine switch should be activated, and sends the Spine switch's ID to the Routing Module to tell it that traffic can be distributed on the new active Spine switch.

If the timer for deactivating Spine switches is on and Tdd is exceeded, the Spine Switch Controller calculates the number (*min_num*) of Spine switches that should be deactivated (as shown in lines 5–7 in Algorithm 5) and determines which Spine switches should be deactivated. The Spine Switch Controller sends the Spine switches' ID to the Power Control Module to inform that the Spine switches should be deactivated, and sends the Spine switch's ID to the Routing Module to inform that traffic is not to be distributed on those Spine switches.

Backup spine switch

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 DCN. In this scenario, there is no active Spine switch to transmit the extra traffic; and the DCN 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 transmit traffic when there is no other active Spine switch with enough available bandwidth, as shown in Fig. 3.

After GSSMS decides to activate a Spine switch, there is an expected delay (1–3 min as noted for EnergyWise in [16]) before the Spine switch can accept new flows. During this time duration, GSSMS distributes new flows on the Backup Spine Switch. After the newly activated Spine switch becomes active, GSSMS distributes new flows on the Spine switch. GSSMS never move existing flows from one Spine switch to another.

Simulation setup and results

We use CloudSim [17] for simulation. The traffic model used in this paper is the ON/OFF model. The ON/OFF traffic model is observed in real DCNs [4]. This paper uses the same distribution used in [18] for the durations of ON/OFF period – Pareto distribution. In the simulation, the DCN configuration used is adopted from Cisco's practices [7]. The DCN consists of 16 Leaf switches and 8 Spine switches, and the link capacity is set to10Gbps. One Leaf switch connects to 15 servers. One server has 10 VMs and each VM is a traffic source.

Metrics used in simulation evaluation include: the average number of active Spine switches (ANASS), the percentage of power consumed by GSSMS over the total number of Spine switches (POPC) and the percentage of failures (PF). They are calculated by Eqs. (1), (2) and (3). In Eq. 2, N represents the total number of Spine switches, and we assume that a switch in the Hibernation mode can lead to a significant power saving, e.g., 84 % as reported in [19].

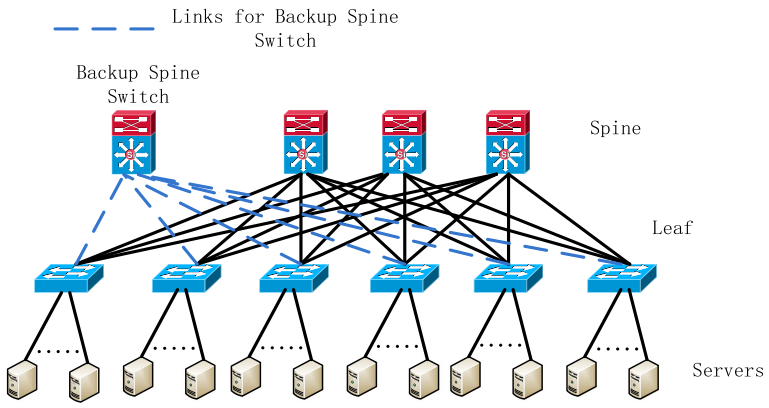

Fig. 3 Backup Spine Switch

$$\mathrm{ANA}SS = \frac{\sum \text{number of active Spine switches} \times \text{duration}}{\text{total duration}} \quad (1)$$

$$\mathrm{POP}C = \frac{\mathrm{ANASS} + (\mathrm{N}-\mathrm{ANASS}) \times 0.16}{\mathrm{N}} \times 100\% \quad (2)$$

$$\mathrm{PF} = \frac{\text{the number of failed flows}}{\text{the number of all flows}} \quad (3)$$

A failure is said to occur when a VM intends to send data, but there is not enough available bandwidth for the data flow. In GSSMS, the minimum number of active Spine switches is two. One of the two Spine switches is the backup Spine switch.

Input traffic patterns

This paper 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.

Uniform traffic

For the uniform traffic [2], the traffic rate is fixed for each traffic source. Based on the traffic's destination, this paper considers three types of uniform traffic: Near traffic, Far traffic and Half-Far/Half-Near traffic.

Near traffic (Near): For each data flow, the flow's source and destination connect with the same Leaf switch.

Far traffic (Far): For each data flow, the flow's source and destination connect with different Leaf switches.

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

These traffic types are adapted from [2]. The simulation results for the Uniform traffic are presented in Fig. 4. In Fig. 4a, for the Near traffic case, ANASS is two all the time, because the traffic is not routed through Spine switches for Near traffic. Thus, ANASS is not affected by the traffic change. For Far traffic, as expected, ANASS increases as the traffic rate for each traffic source increases. 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 Half-Far/Half-Near traffic, ANASS also increases as the traffic rate for each traffic source increases. The reason is same as the reason provided for Far traffic. Compared with the Far traffic case, for a given traffic rate, the total traffic of the DCN is approximately half of that in the Far traffic scenario. Therefore, ANASS is approximately half of ANASS in the Far traffic scenario. Figure 4a also shows that POPC is determined by the traffic routing through Spine switches. For Near traffic, there is no traffic routing through Spine switches. Therefore, only two Spine switches are active and POPC is 37 % all the time. For Far traffic, POPC increases while the traffic rate for each traffic source increases. That means that the saved energy decreases as the traffic rate increases. When the traffic rate is 30Mbps, POPC is 37 %, which means GSSMS saves 63 % energy in comparison to a static system. When the traffic rate is 310Mbps, POPC is 100 %, which means GSSMS cannot save energy. For Half-Far/Half-Near traffic, GSSMS can save energy from 42 to 63 %. GSSMS can save more energy for Half-Far/Half-Near traffic at a given traffic rate because Half-Far/Half-Near traffic has lower traffic routing through Spine switches.

In Fig. 4b, failures appear when the traffic rate increases (e.g., traffic rate from 150Mbps to 190Mbps, from 210Mbps to 230Mbps and from 250Mbps to 290Mbps) while ANASS remains the same value. For instance, the four points shown in Fig. 4b: A, B, C and D, but the values of PF are small. Because GSSMS does not activate Spine switches immediately, it is possible that the network does not have enough available bandwidth for the traffic at a given time; hence failures occur on the system. The increase in PF is caused by the increase in the traffic rate. With the same ANASS, the network in the case of a higher traffic rate has less available bandwidth; as a result, more flows cannot receive adequate bandwidth. If the traffic rate keeps increasing, ANASS increases, such as the points E, F, and G shown in Fig. 4b. The increase of ANASS means that the available bandwidth of the network increases. Therefore, the network can handle the traffic without failures most of the time.

Sine-wave traffic

For the Sine-Wave traffic, the traffic rate for each traffic source varies as a sine wave [2].

$$\text{Traffic Rate} = {}^{1}/_{2} \times \text{max rate} \times (1 + sin(t)) \quad (4)$$

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 a similar meaning as their respective Uniform traffic counterparts.

NASS in Fig. 5 represents the number of active Spine switches. Figure 5 shows the simulation results for Far traffic. The number of Spine switches changes as the traffic rate for each traffic source changes. The results show that NASS changes according to the total traffic of the DCN. As the total traffic of the DCN increases, more Spine switches are activated in the DCN. As the total traffic of the datacenter decreases, NASS of the DCN is observed to decrease.

The Half-Far/Half-Near traffic gives rise to a similar results, except that the value of NASS at a given point in time seems to be smaller than that achieved with the Far traffic scenario.

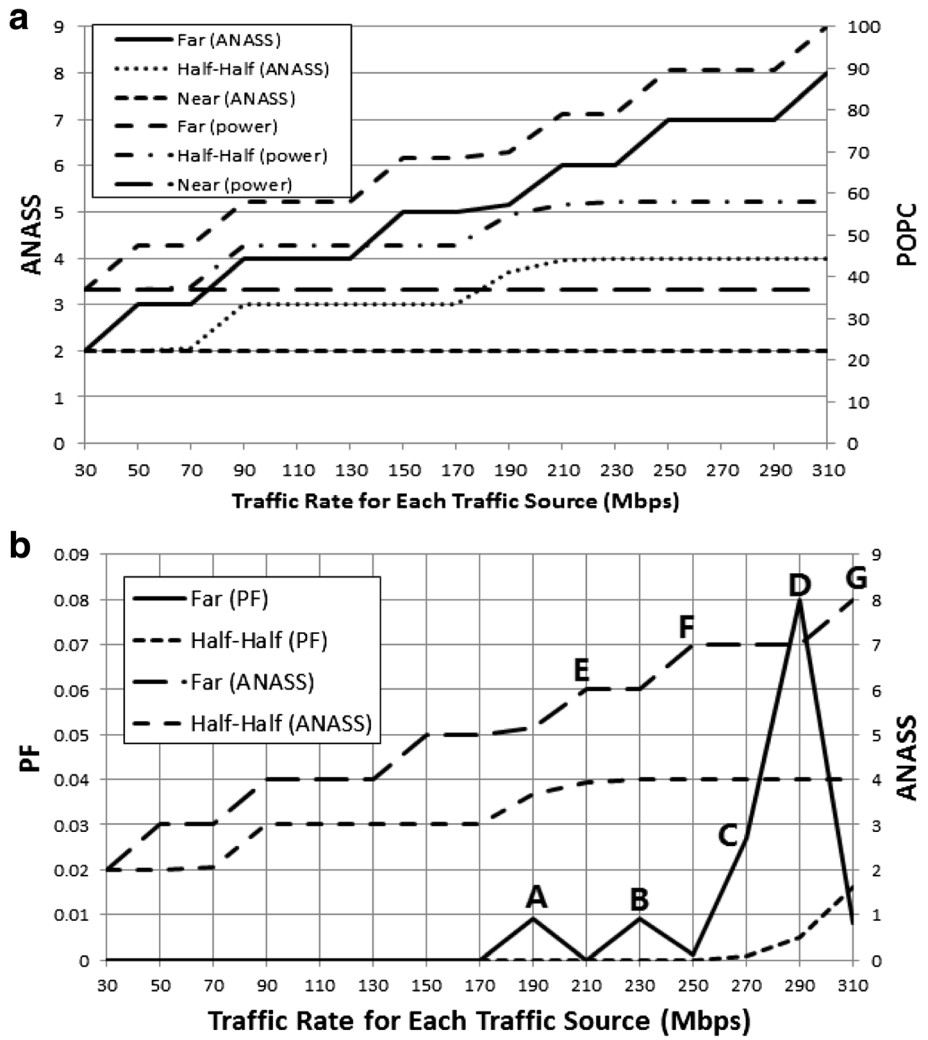

Fig. 4 Simulation Results of the Uniform Traffic

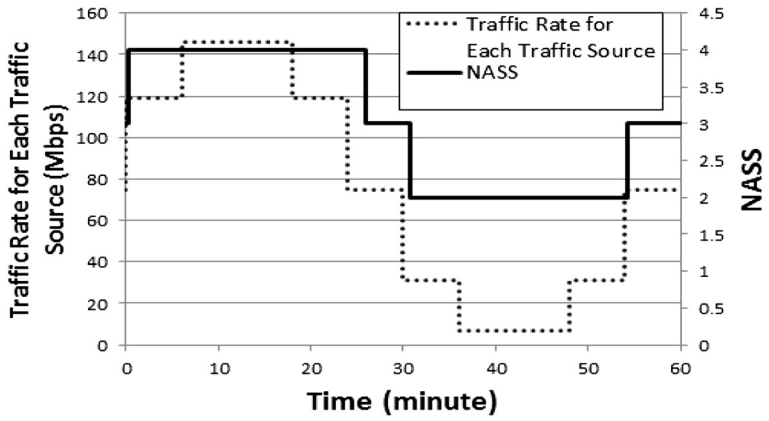

Fig. 5 Simulation Result for Far Traffic for the Sine-Wave Traffic

Random traffic

Random traffic is used to simulate the scenario observed in datacenters where 95 % of the time, the traffic is 10 % of the DCN capacity, and during 5 % of the time, the traffic is higher than 10 % of the DCN capacity [4]. In the simulation setup, each link has a capacity of 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.

The results for Random traffic are shown in Fig. 6. The traffic bursts last for approximately 1 min and GSSMS increases the number of active Spine switches for the duration of the traffic bursts and decreases the number of active Spine switches when the traffic bursts finish. Time duration Tda is 5 s and Tdd is 10 s in the simulation. NASS increases approximately 5 s after the traffic rate for each traffic source increases. NASS decreases approximately 10 s 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.

For Sine-Wave traffic and Random traffic, we expect that the shapes of ANASS and POPC graphs would be similar to those in Fig. 4a.

Comparison between GSSMS and fixed numbers of spine switches

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

Figures 7 and 8 present the simulation results for Uniform-Far and Uniform-Half-Far/Half-Near (Uniform-Half/Half) traffic. NSS in Fig. 7 represents the number of Spine switches used in a datacenter that does not use GSSMS and deploys a fixed number of Spine switches. In Fig. 7, compared to the datacenter with six Spine switches, GSSMS can save energy when the traffic rate is < 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 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 deploying a fixed number of eight Spine switches, GSSMS can save energy when not all the Spine switches are active and has an acceptable PF at the same time. In Fig. 8, compared to the datacenter with a fixed set of four Spine switches, GSSMS produces an ANASS lower than four when the traffic rate is < 210Mbps. When the traffic rate is > 210Mbps, ANASS for GSSMS is four and it produces a PF that is similar to the datacenter using a fixed set of four Spine switches.

The simulations for Sine-Wave-Far (Sine-Far) and Sine-Wave-Half-Far/Half-Near (Sine-Half/Half) traffic produce similar results, except that the value of ANASS and PF at a given traffic rate is smaller than that for the Uniform-Far and Uniform-Half/Half scenarios.

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 (<0.09) and reduce PF significantly (100 to 82 %) by having one or more active Spine switches when the PF of GSSMS is compared with the PF of the datacenter with a fixed set of two, four and six Spine switches.

Effect of system parameters

A detailed simulation-based investigation of the impact of various system and workload parameters on performance was performed.

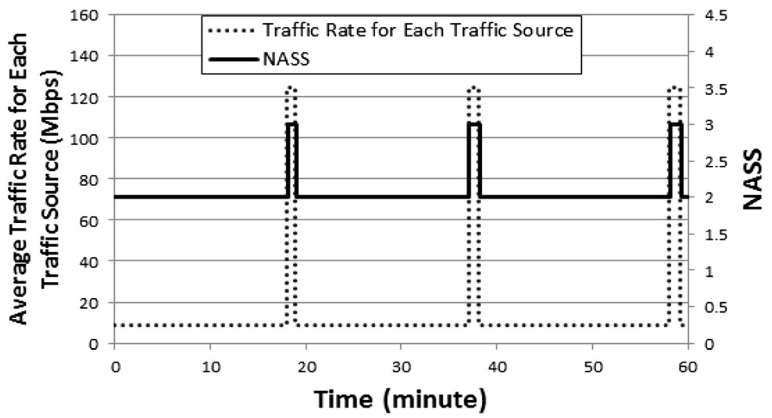

Fig. 6 Simulation Result of the Random Traffic

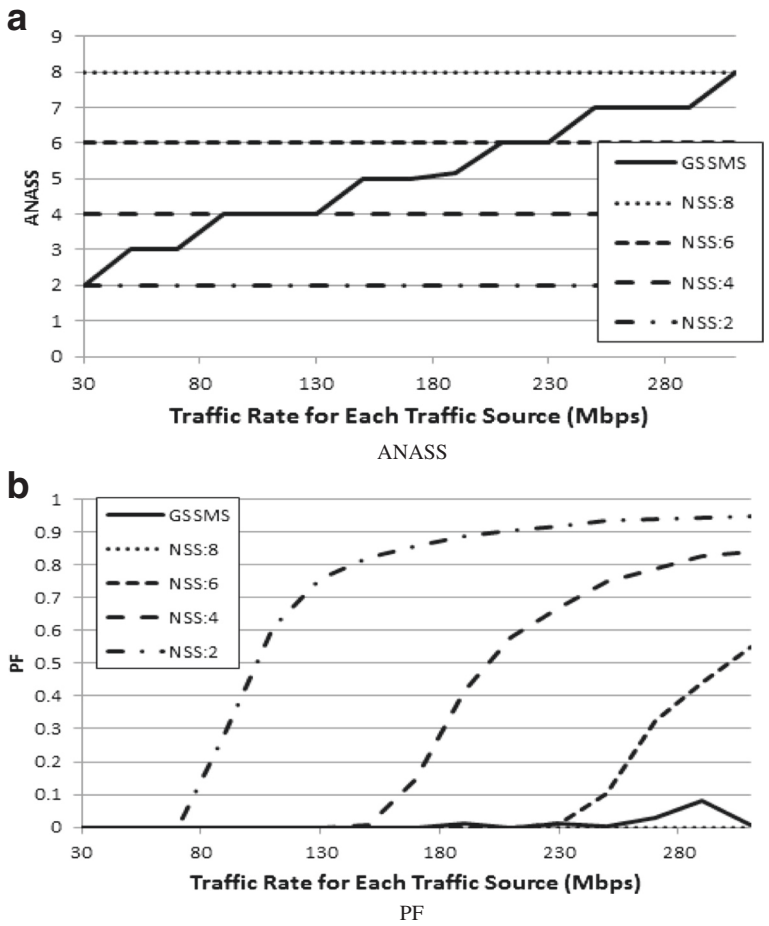

Fig. 7 Simulation Results for Uniform-Far Traffic. NSS-n: the number of Spine switches, n, used without GSSMS

The system parameters are the aforementioned parameters: FUT-H, FUT-L, CT-H, CT-L, SUT-H, SUT-L, Tda and Tdd.

Figure 9 presents the simulation results which demonstrate the impact of FUT-H. Figure 10 shows the impact of 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.

The simulation results shown in Fig. 9a 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 Fig. 9b 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, which 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.

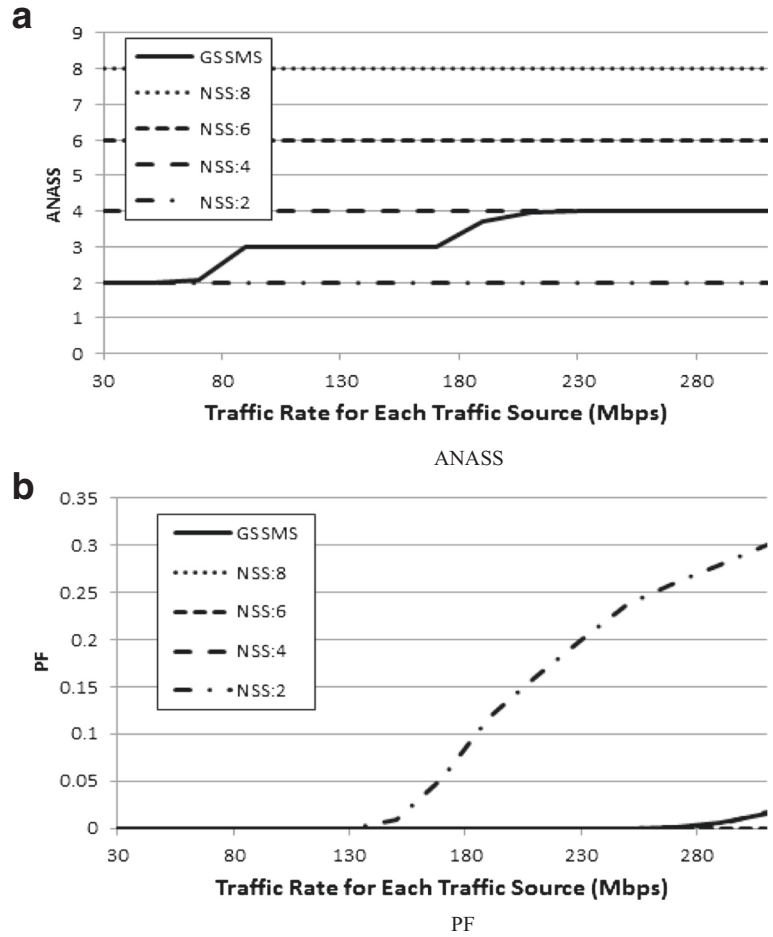

Fig. 8 Simulation Results for Uniform-Half-Far/Half-Near Traffic

In Fig. 9a, 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 9a also shows that ANASS of Uniform-Far traffic for given FUT-H and FUT-L is higher than ANASS of Sine-Wave-Far 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 IV.A, 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.

Figure 10a demonstrates 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

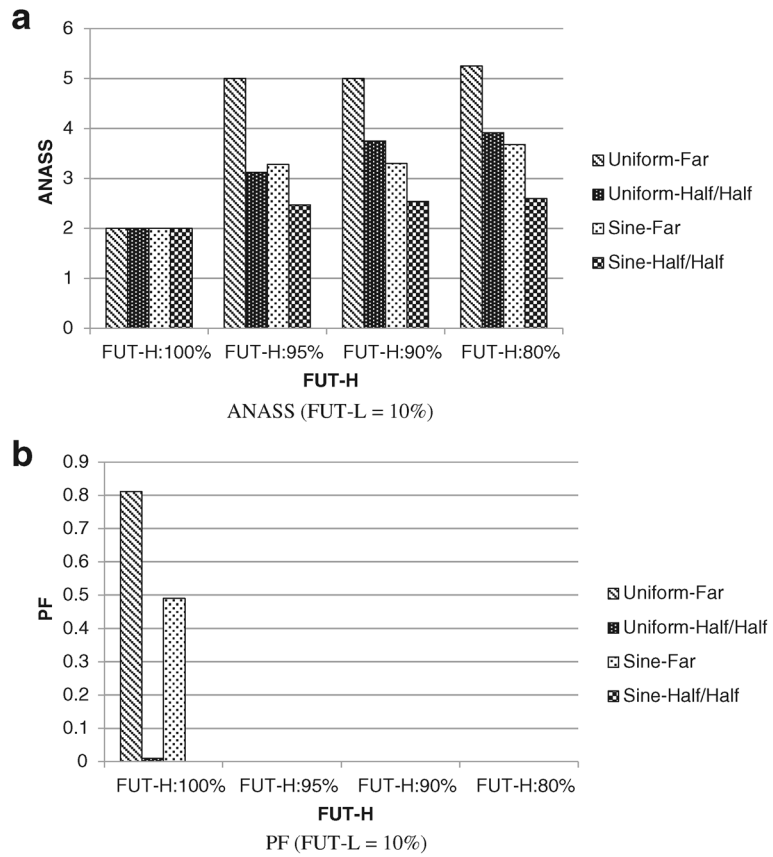

Fig. 9 The Impact of FUT-H on Performance

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 a 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 Fig. 10b 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 10a 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 before, the difference between ANASSs

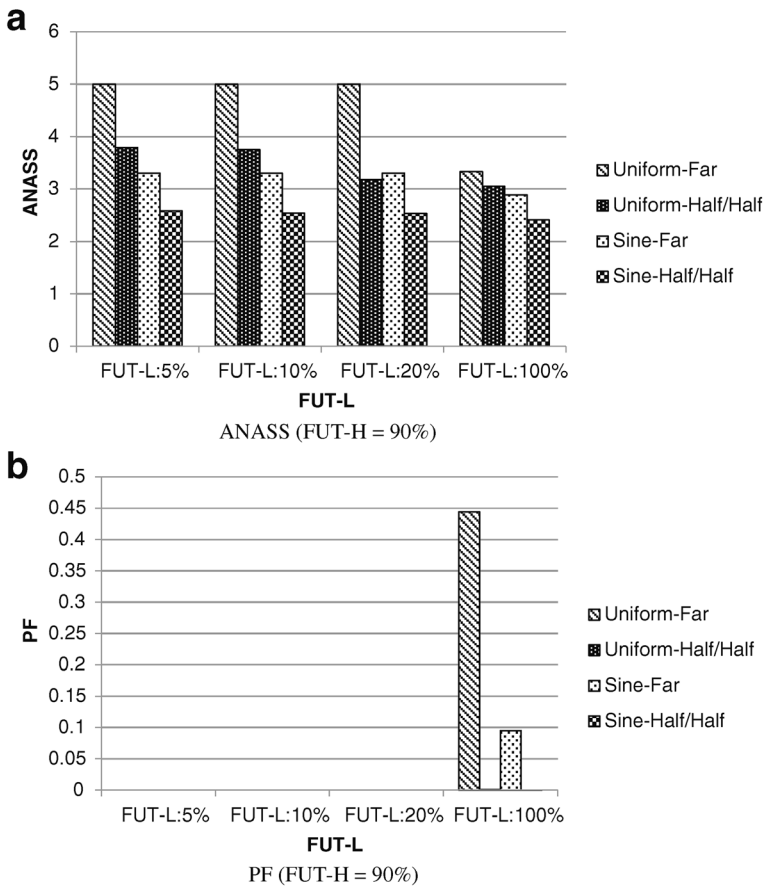

Fig. 10 The Impact of FUT-L on Performance

for different traffic patterns is caused by the difference between the traffic routing through Spine switches in the different traffic pattern cases.

Simulation results show that SUTs have similar impact on performance with FUTs. The difference is that SUTs have less impact on performance than FUTs, especially on PF. In the simulation for SUT-H, the highest PF is 0.005, and the highest PF is 0.0002 in the simulation for SUT-L. The highest PF in the simulation for FUTs are 0.8 and 0.44 respectively.

Figures 11 and 12 present the impact of CT-H and CT-L on performance. Like FUTs, CT-H is used for activating a Spine Switch in response to an increase in

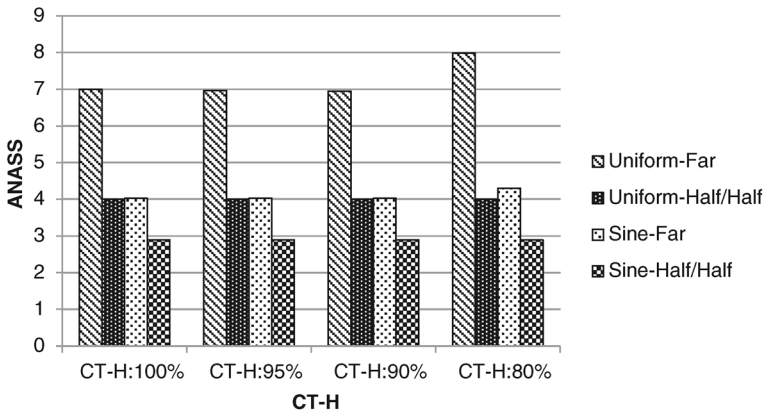

Fig. 11 The Impact of CT-H on Performance (CT-L = 20 %)

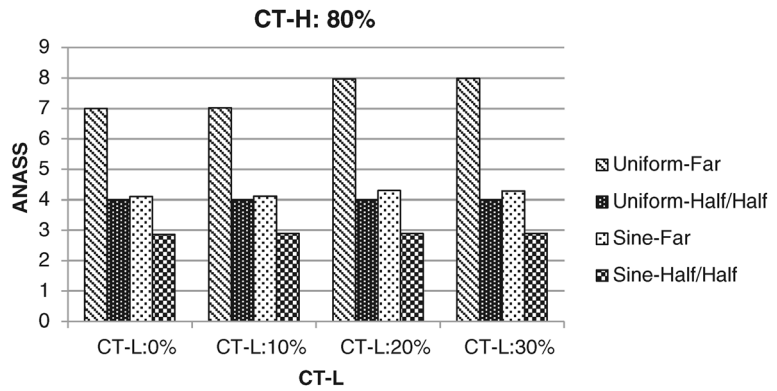

Fig. 12 The Impact of CT-L on Performance

traffic. CT-L is used for deactivating Spine switches when traffic reaches a predetermined low value.

The simulation result for the Far traffic case in Fig. 11 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 11 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, Fig. 11 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 simulation results in Fig. 12 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).

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 Fig. 13a 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 s, ANASS decreases as Tda increases. The reason is same with Uniform traffic. The simulation results shown in Fig. 13b demonstrate that when Tda is 100 s or 20 s, 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.

There is another difference between Uniform traffic and Sine-Wave traffic shown in Fig. 13a. 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 s. 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 s, the number of active Spine switches decreases to two 200 s after the decrease of the traffic

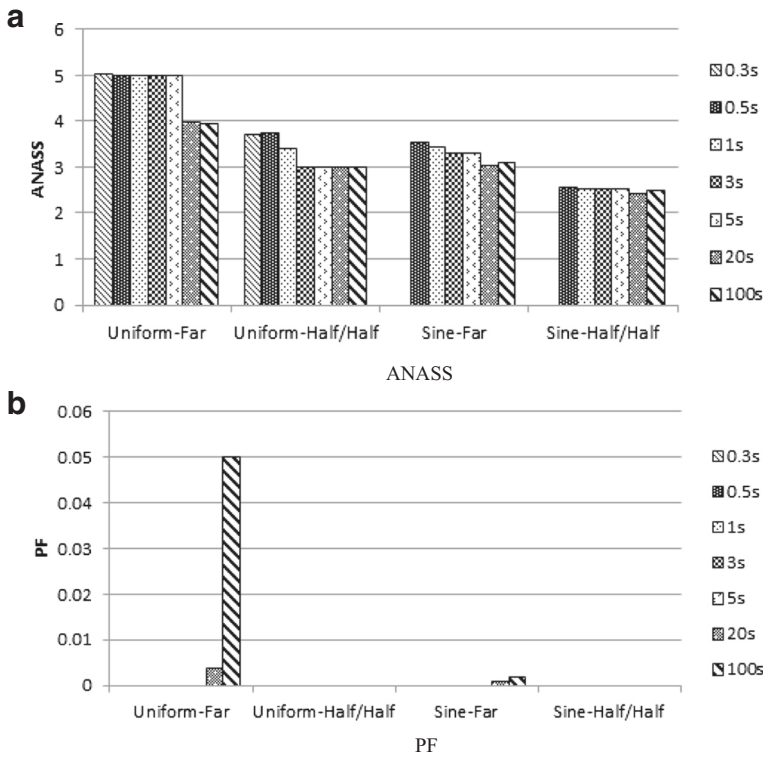

Fig. 13 The Impact of the Time Duration on Performance

rate. The 200 s delay is the reason for the increase of ANASS.

The workload parameters include the number of Leaf Switches, ON/OFF Duration-ON and ON/OFF Duration-OFF.

Figure 14 presents the impact of the number of Leaf switches (NLS). 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 24 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

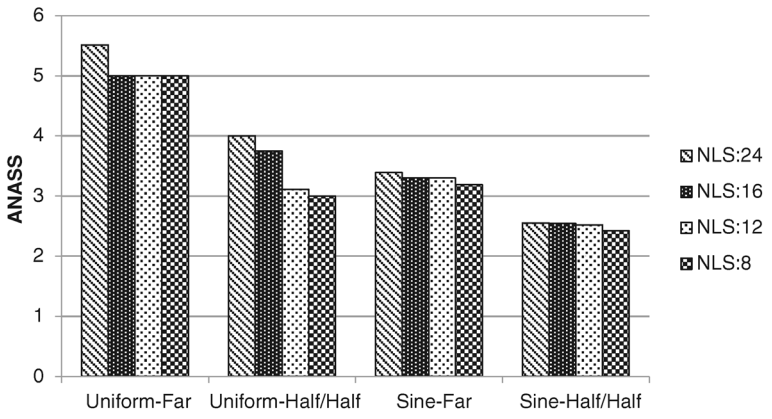

Fig. 14 The Impact of the Number of Leaf Switches on Performance

exceeding the link capacity increases, the probability of GSSMS activating a Spine switch increases.

The simulation results in Fig. 15 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 DCN decreases. ANASS decreases when the total traffic decreases.

The simulation results in Fig. 16 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 DCN increases. As a result, ANASS increases when the total traffic increases.

Guidelines for choosing the control parameters

A few guidelines for choosing the parameters have been developed and are presented 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. FUT-H is to be chosen to be low enough such that the desired value of PF is achieved. SUT-H can then be chosen such that the lowest possible value of ANASS for the selected FUT-H is achieved.

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 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.

Conclusions and future directions

This paper proposed GSSMS to save energy consumed by the DCN using the Spine-Leaf topology which has received increasing attention in practice. GSSMS can dynamically manage the number of Spine switches according to the current DCN traffic. The purpose is to save energy consumption without a significant decrease in reliability when the traffic intensity is low. The GSSMS algorithm used six parameters to determine the number of active Spine switches in a DCN. The threshold parameters 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 DCN, 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 the number of Spine switches is not enough to handle the increased traffic in the DCN.

The simulation results show that GSSMS can work efficiently for different input traffic patterns. In comparison to Far traffic, GSSMS can save more energy for the Half-Far/Half-Near traffic because the Half-Far/Half-Near traffic

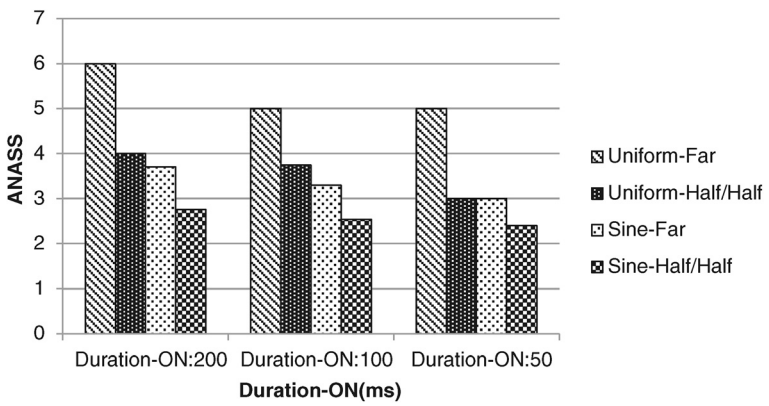

Fig. 15 The Impact of the ON Duration on Performance (Duration-OFF = 20 ms)

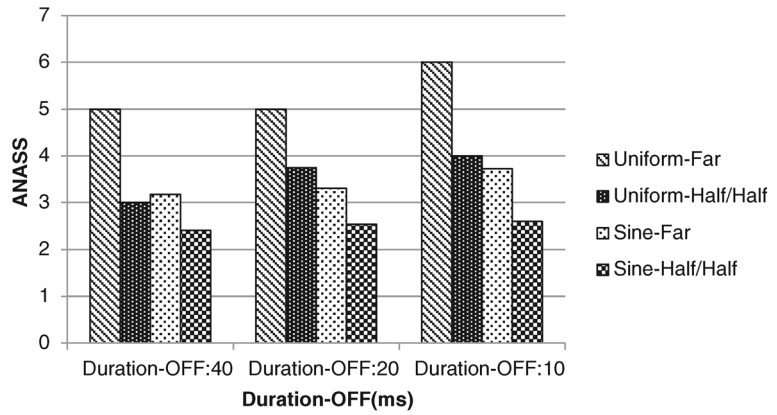

Fig. 16 The Impact of the OFF Duration on Performance (Duration-ON = 100 ms)

results in lower traffic through Spine switches. Compared with Uniform traffic, GSSMS can save more energy for Sine-Wave traffic because the traffic routed through Spine switches with the Sine-Wave traffic is lower than that achieved with the Uniform traffic. Specifically, GSSMS can save energy (up to 63 %) with a slight increase (0.08) in PF (shown in Fig. 4) or reduce PF significantly by dynamically adjusting the number of active Spine switches (see Fig. 7). Similar conclusions are expected when DCNs have different number of Leaf and Spine switches than those used in our simulations.

The simulation results also show the limitations of GSSMS. First, if the traffic burst exceeds the link capacity, activating one Spine switch may not be enough to handle the traffic burst. The current algorithm needs to wait for another monitoring duration to activate an additional Spine switch if the high traffic demand persists. To handle the scenario more effectively, the algorithm can be extended by incorporating another high threshold. Once this high threshold value is exceeded, two or more Spine switches can be activated at the same time. Second, if the traffic is complex (e.g., when the traffic is mixed with slow increase and spikes which can last longer than Tda occasionally), it is challenging to find a Tda that works efficiently for the complex traffic. As shown in simulation, a small Tda works efficiently for large traffic bursts, whereas a large Tda works efficiently for the stable traffic.

Further research is warranted for addressing the issues outlined in the previous paragraph. Directions for future research include the following:

(i). Investigating the effect of activating/deactivating multiple Spine switches at the same time.
(ii). Investigating the use of a variable Tda to improve GSSMS's performance when subjected to other traffic patterns forms an interesting direction for future research.
(iii). Investigating the use of Network Function Virtualization (NFV) for the proposed approach: Although GSSM is not specifically targeted for the SDN or the NFV paradigm, the proposed approach could be seen as one of the network functions for a datacenter network. Hence, it is worthy of further investigation using NFV for GSSM.

Abbreviations

DCN, Datacenter network; FBFLY, Flattened Butterfly; GSSMS, Green Spine Switch Management System; STP, Spanning tree protocol.

Authors' contributions

The work is mostly based on XL's Master's research and thesis, which was co-supervised by CL and SM. All authors contributed to each aspect of the technical areas and the writings. XL designed and implemented the simulation on CloudSim. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Received: 1 January 2016 Accepted: 21 June 2016
Published online: 13 July 2016

References

1. Koomey J. Growth in Data center electricity use 2005 to 2010 [Online]. Analytics Press. Available: http://www.analyticspress.com/datacenters.html, [Last Accessed on 19 May 2016]
2. Heller B, Seetharaman S, Mahadevan P (2010) Elastic Tree: Saving Energy in Data Center Network. Proc. of 7th USENIX Conf. on Net. Sys. Design and Imple. (NSDI). San Jose, CA, USA, pp 17-17
3. Abts D, Marty MR, Wells PM, Klausler P, Liu H (2010) Energy Proportional Datacenter Networks. Proc of the 37th Int Symp Comput Architecture. Saint-Malo, France, pp 338–347
4. Benson T, Akella A, Maltz D (2010) Network Traffic Characteristics of Data Centers in the Wild. Proc of 10th ACM SIGCOMM Conf Internet Measurement Melbourne, Australia, pp 267–280
5. Beck P et al (2013) IBM and Cisco: Together for a World Class Data Center. IBM Redbooks. [Online], Available: http://www.redbooks.ibm.com/redbooks/pdfs/sg248105.pdf, [Last Accessed on 4 Jul 2016]
6. Alizadeh M, Edsall T (2013) On the Data Path Performance of Leaf-Spine Datacenter Fabrics. Proc of IEEE 21st Symp High-Performance Interconnects (HOTI), San Jose, CA USA, pp 71–74
7. Cisco. Cisco's Massively Scalable Data Center. [Online], Available: http://www.cisco.com/c/en/us/solutions/enterprise/data-center-designs-data-center-networking/landing_msdc.html, [Last Accessed on 19 May 2016]

8. Kakadia D Varma V (2012) Energy Efficient Data Center Networks – A SDN based Approach. Bangalore, IBM Collaborative Academia Research Exchange (I-CARE) IISC, Report No: IIIT/TR/2012/-1
9. Prete L, Farina F, Campanella M, Biancini A (2012) Energy Efficient Minimum Spanning Tree in OpenFlow Networks. Proc of European Workshop on Software Define Networking, Darmstadt Germany, pp 36-41
10. Carrega A, Singh S, Bruschi R, Bolla R (2012) Traffic Merging for Energy-Efficient Datacenter Networks. Proc of Int Symp on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), Genoa Italy, pp 1–5
11. Vasic N, Bhurat P, Novakovic D, Canini M, Shekhar S, Kostic D (2011) Identifying and Using Energy-Critical Paths. Proc of the 7th Conf on Emerging Networking Experiments and Technologies Article No. 18 Tokyo, Japan
12. Lee C, Rhee JK (2012) Traffic Off-Balancing Algorithm: Toward Energy Proportional Datacenter Network. Proc Opto-Electronics Commun Conf (OECC), Busan, Korea, pp 409–410
13. Shi Z, Beard C, Mitchell K (2013) Analytical Models for Understanding Space, Backoff, and Flow Correlation in CSMA Wireless Networks. Wireless Networks 19(3):393–409
14. Shi Z, Beard C, Mitchell K (2008) Tunable Traffic Control for Multihop CSMA Networks. Proc of the 28th Military Communications Conf (MILCOM), San Diego, CA, 1–7
15. Shi Z, Beard C, Mitchell K (2011) Competition, Cooperation, and Optimization in Multi-Hop CSMA Networks. Proc of the 9th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN) Paphos, Cyprus, pp 117–120
16. Cisco. Configuring EnergyWise. [Online], Available: http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst2960x/software/15-0_2_EX/energywise/configuration_guide/b_ew_152ex_2960-x_cg/b_ew_152ex_2960-x_cg_chapter_010.pdf, [Last Accessed in on 19 May 2016]
17. The Cloud Computing and Distributed Systems (CLOUDS) Laboratory, University of Melbourne, http://www.cloudbus.org/cloudsim/
18. Calabretta N, Centelles RP, Di Lucente S, Dorren TJS (2013) On the Performance of a Large-Scale Optical Packet Switch Under Realistic Data Center Traffic. Opt Commun Networking 5(6):565–573
19. Miercom Report. Cisco Catalyst 2960-X/2960-XR Switches. Nov. 2013, Available: http://miercom.com/pdf/reports/20131112.pdf, [Last Accessed on 19 May 2016]
20. Li X, Lung C, Majumdar S (2015) Energy Aware Green Spine Switch Management for Spine-Leaf Datacenter Networks. Proc IEEE Int Conf Communications (ICC), London, UK, pp 116–121
21. Cisco. Massively Scalable Data Center (MSDC) Design and Implementation Guide. Oct. 2014, http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Data_Center/MSDC/1-0/MSDC_Phase1.pdf, [Last Accessed on 19 May 2016]