Reducing Energy Consumption in SDN-based Data Center Networks Through Flow Consolidation Strategies

Marcelo da Silva Conterato
Pontifical Catholic University of Rio Grande do Sul
Porto Alegre, RS, Brazil
marcelo.conterato@acad.pucrs.br

Tiago Coelho Ferreto,
Pontifical Catholic University of Rio Grande do Sul
Porto Alegre, RS, Brazil
tiago.ferreto@pucrs.br

Fábio Rossi
Federal Institute of Education, Science and Technology Farroupilha
Alegrete, RS, Brazil
fabio.rossi@iffarroupilha.edu.br

Wagner dos Santos Marques
Pontifical Catholic University of Rio Grande do Sul
Porto Alegre, RS, Brazil
wagner.marques.001@acad.pucrs.br

Paulo Silas Severo de Souza
Pontifical Catholic University of Rio Grande do Sul
Porto Alegre, RS, Brazil
paulo.silas@acad.pucrs.br

ABSTRACT

In the last decade we noticed a growth on studies regarding energy savings in data centers. The main reasons include political factors such as compliance with global protocols of conscious energy consumption, financial incentives such as tax reduction, and environmentally driven by concerns about sustainability issues such as emission of heat and gases harmful to the ozone layer. Most works aim to reduce the energy consumption of servers and cooling systems. However, network devices comprise also a significant slice of the total Data Center energy consumption, and most studies often neglect that. In this paper, we propose techniques to define flow paths in an SDN-based Data Center network respecting flow bandwidth requirements, while also enabling changing the operation state of network devices to a state of lower energy consumption in order to reduce the total consumption of the network layer. We evaluate the proposed techniques using different ratios of link demand oversubscription in a fat-tree topology with different POD sizes. Results show savings of up to 70% regarding energy consumption in the network layer.

CCS CONCEPTS

• **Networks → Programmable networks**; **Data center networks**;
• **Applied computing → Data center**;

KEYWORDS

Data center network, energy-saving, consolidation, software defined networks.

SAC '19, April 8–12, 2019, Limassol, Cyprus

ACM ISBN 978-1-4503-5933-7/19/04…$15.00
https://doi.org/10.1145/3297280.3297420

ACM Reference Format:
Marcelo da Silva Conterato, Tiago Coelho Ferreto,, Fábio Rossi, Wagner dos Santos Marques, and Paulo Silas Severo de Souza. 2019. Reducing Energy Consumption in SDN-based Data Center Networks Through Flow Consolidation Strategies. In *The 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19), April 8–12, 2019, Limassol, Cyprus.* ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.1145/3297280.3297420

1 INTRODUCTION

Enterprises like Google, Facebook, Yahoo!, and Amazon host their applications in cloud environments based on large-scale Data Centers. Such infrastructures offer specialized platforms for storing, searching, and processing an enormous volume of data. However, as more processing is needed, higher is the energy consumption. High energy consumption is one of the most prominent limitations of today's data centers, as they are on the wrong track to global sustainability issues. Several countries offer tax incentives to service providers in order to reduce energy consumption. This action is necessary due to the environmental impact these large-scale infrastructures may cause, such as heating and emission of gases such as carbon dioxide, harmful to the atmosphere and responsible for the greenhouse effect.

Studies indicate that the total energy consumption of all Data Centers in the world increased by 56% between 2005 and 2014, with total energy costs reaching billions of dollars [12]. Energy consumption is expected to grow in the future, increasing total cost of operation (TCO) [17]. Consequently, there are several works that address this issue in the literature, wherein most of them propose new techniques of energy savings on the servers, through server consolidation strategies, power management techniques (e.g., DVFS - Dynamic Voltage and Frequency Scaling), and also through intelligent management of the cooling of the computational environment. However, there are a reduced number of works aiming on the network layer. It may occur due to the limitations of the physical substrates of today's networks, which in most cases remain on traditional arrangements. They inherit the drawbacks from the TCP/IP protocol stack, such as lack of performance isolation, increased security risks, poor management flexibility, limited support for new network innovations, and increased use of energy [4].

The growth in Data Centers' capacity requires the conscious use of energy, demanding a new, more reconfigurable network architecture, so that new features can be easily added within the network, such as: traffic orchestration, new routing schemes, or energy consumption monitoring. When it comes to solutions to save energy in networks, different strategies are possible, and among them, it is interesting to investigate if further optimizations in the network topology can address this issue.

Software-Defined Networking (SDN) appeared as a key paradigm to achieve such improvement. It provides a (logically) centralized control point in the network, enhancing programmability, and provides remote management between infrastructures using a single open protocol. This structure allows network and business applications to work together with the help of analytics and to reconfigure the network policies according to the changing user experience and application performance. In this context, network design and architecture remain the same while applications and systems progress to an advanced level [22].

SDN [14] physically separates the data plane from the control plane on packet forwarding devices (switches). A logically centralized controller (control plane) independently controls every single stream on the network by installing custom flow rules on the switches that are responsible for routing (data plane). The rules are composed of fields that combine data flow information along with an instruction field that details the actions to be taken on the flow and the counters that maintain these flow statistics. OpenFlow [18] is one of the main implementations of the SDN paradigm. SDN has the advantage of providing central visibility into the network, enabling multiple traffic engineering schemes to be implemented, as well as centralized control, offering a compelling design option for Data Center networks.

In this paper, we propose a flow mapping algorithm aiming at reducing the energy consumption of the network layer in SDN-based Data Centers. The algorithm focus on consolidating network flows based on bandwidth demands and mapping them to the existing network infrastructure. Network elements (switches and links) are dynamically put in a low energy consumption mode when not used, and the speed on switch ports are adjusted based on the required bandwidth demand, according to mapped flows. We analyzed the impact of using different heuristics to perform the consolidation and different oversubscription ratios to increase energy savings.

The paper is organized as follows: Section 2 presents a background on SDN and Data Center networking and describes related works regarding minimizing energy consumption in the network layer; Section 3 describes our algorithm to reduce energy consumption in the network layer; Section 4 presents an evaluation of our algorithm along with a discussion on the results obtained. Finally Section 5 concludes the paper and presents future work.

2 BACKGROUND

Software-Defined Networking (SDN) [13] is a paradigm that provides a more dynamic, manageable, and adaptive network. It can handle high bandwidth and the dynamic behavior of today's applications, making networks both flexible and efficient. In such paradigm, network control is programmable, enabling easy configuration, holistic management, and quick network resources optimization. Additionally, it allows administrators to dynamically adjust the traffic flow across the network to meet new application demands.

According to Budhiraja et al. [6], a Data Center is a centralized repository, whether physical or virtual, for the storage, management, and dissemination of data and information organized around a specific knowledge base or belonging to a particular company. The Data Center Network (DCN) [4] is the communication infrastructure that can be described by the network topology, routing/switching equipment, and protocols.

In a conventional topology, the Top-of-Rack Switch (ToR) in the access layer provides connectivity to the servers allocated in each rack. Each aggregation switch in the aggregation layer (sometimes referred to as the distribution layer) forwards the multi-access layer (ToR) traffic to the core layer. Each ToR switch is connected to multiple aggregation switches for redundancy. The aggregation layer provides secure connectivity between aggregation switches and core routers connected to the Internet.

Even with topologies that offer dynamicity and speed to network applications such as Fat-Tree, it may not be sufficient to meet specific demands. Al-Fares et al. [3] introduce the concept of oversubscription in Data Center networks as a means to reduce the total cost of the project. The term oversubscription is defined as the ratio of the aggregate bandwidth between the final hosts to the full bandwidth of a particular communication topology. For instance, an oversubscription of 1:1 indicates that all hosts can potentially communicate with other hosts with their total bandwidth. However, knowing exactly where to act on the network flows requires a network paradigm that allows a holistic view of the physical substrate and the flows that travel on it.

Energy consumption in the Data Centers has become a key component for the sustainable growth of some paradigms, such as cloud computing. As servers become more efficient in terms of energy consumption, the focus shift to the network layer. Several works have addressed techniques that aim to optimize energy consumption in DCNs through network programmability.

Recent studies also focus on reducing energy consumption in Data Center networks through the concept of proportional energy. They assume that the energy consumption of a Data Center network depends mainly on the wiring capacity of the links that interconnect these equipments, where the suspend mode is used to indicate that a particular component of the network can be switched off when there is no traffic through it.

2.1 Energy-aware approaches for Data Center networks

Heller et al. [10] present a energy optimizer for the network, to continuously monitor Data Center traffic, where a set of network elements are selected and must remain active to ensure performance tolerance and failure. Unnecessary ones are disabled. The authors demonstrated that traffic flows in a DCN can be consolidated into a small set of links and switches, which are sufficient to meet the demand for bandwidth. The analyzed methods decide which subsets of links and switches can be turned off. The authors estimated energy savings of about 50%.

Abts et al. [1] propose several ways to design a high-performance DCN whose energy consumption is proportional to the amount of traffic, i.e., the amount of energy consumed is proportional to the traffic intensity on the network. The work demonstrates that a DCN based on the Flattened Butterfly topology results in a network more energy efficient and therefore with lower operating costs. The future bandwidth requirements of each link are periodically estimated and dynamically reconfigured to meet these requirements. The authors present results of reduction of the global energy consumption of up to 60%.

Huong et al. [11] propose to reduce energy consumption in the DCN based on traffic engineering, focusing on optimizing the energy consumption of the network components, by designing a control system network. This system adapts dynamically to the set of network components (active), where patterns of total traffic that pass through the Data Center are verified and, based on these standards, actions are defined to be performed by these devices. The optimization algorithm proposed by the authors showed up to 35% in energy reduction.

Wang et al. [25] propose to build a system based on the ElasticTree [10] topology that can be flexible and adapt the topology of the DCN to the energy consumption that satisfies traffic requirements. The goal is to consolidate traffic flows based on analysis of the correlation between flows in a DCN. Another feature of this work is the correlation of traffic consolidation with the adaptation of the link rate to increase energy savings. It uses linear programming to determine the rate of consolidation on each data link in the DCN, and proposes a heuristic to find solutions for configuration and consolidation. In the paper, the authors reported results of energy savings in the DCN of up to 46%.

Vu et al. [24] propose an extension for switches that enable the OpenFlow protocol to support different power saving modes. The expansion includes new messages in the OpenFlow protocol stack and new routines integrated into the controller that adds the function of managing them, including the ability to turn on/off and enable/disable equipment ports. It implements the built-in controller functions with an OpenFlow switch based on NetFPGA technology [1] [11].

Zhang and Ansari [26] propose a hierarchical model of energy optimization. Using a Fattree topology, the authors evaluate the possibility of disconnecting and connecting some switches and links hierarchically, without violating connectivity and QoS constraints. They propose some heuristics based on different switch elimination criteria to solve the problem of hierarchical optimization.

Adnan and Gupta [2] propose an online algorithm to dynamically dimension the DCN in order to achieve energy proportionality. The proposed algorithm controls the level of redundancy to guarantee robustness as well as energy efficiency. The role of the optimizer at each level is to find a minimum network subset meeting agreed levels of performance and fault tolerance. In this work, the authors report the reduction of 80% in the energy consumption of the network.

[1] http://netfpga.org

Villarreal et al. [23] propose to reduce the energy consumption of Data Centers by applying a rules-based network equipment management model that makes consumption proportional to traffic. They carried out the extension of the CloudSim simulator [7] to simulate the energy consumption in the network. From a traditional DCN topology, based on the proposed management model, they evaluated the application of the rules of disconnecting and connecting switches and links. The model also takes into account the allocation of virtual machines and some migration strategies for server consolidation. In the experiments, the results indicate that energy saving depend on the configuration of data centers, the type of network equipment and workloads, and may provide savings of about 50% in network energy consumption and 5% in total energy consumption of the data center.

Nam et al. [19] propose a new consumer economy scheme that can flexibly control and route traffic based on power profiles on network devices. They used the OpenFlow protocol in conjunction with SDP-based Netflash cards to implement the DCN. The energy consumption profiles of the network were defined based on the information of each device. According to the authors, the proposal of using power profiles in the network for topology optimization processes and flow routing makes it flexible to work with heterogeneous devices from different vendors.

3 FLOW MAPPING ALGORITHM FOR REDUCING ENERGY CONSUMPTION

Although the servers are increasingly proportional in energy terms, the Data Center network will consume a more significant share of total Data Center energy. Therefore, searching for techniques to save energy used by network devices becomes critical. Just as it is done with servers by minimizing the number of active servers, we can also find ways to reduce the number of active network devices (switches and links).

In this work, strategies are proposed to configure the Data Center network to reduce energy consumption, using the SDN paradigm. These mapping strategies aim to: (i) consolidate flows based on the proposed energy consumption model, (ii) adapt ports' speed to save energy, and (iii) dynamically disconnect elements from networks that are not being used.

The consolidation of traffic provides an efficient approach to save energy in the Data Center network. The proposed mapping algorithms are based on the Bin-packing Problem [9] in combination with the Dijkstra (Shortest Paths) algorithm. The Bin-packing Problem asks for the minimum number k of identical bins of capacity C needed to store a finite collection of weights $w_1, w_2, w_3, \ldots, w_n$ so that no bin has weights stored in it whose sum exceeds the bin's capacity. This is considered an NP Hard problem. The First-fit, Best-fit, and Worst-fit heuristics were used, combining the strategies of reduction of energy consumption through disconnecting links and switches that have not had mapped flows, and adapting the speed of the connections, measuring the impact of the use of different oversubscription factors, so that the energy consumption in the Data Center network can be reduced.

Energy savings can be addressed by relying on the energy consumption of the intermediate network devices that can consume less energy or even be turned off. According to the measurements made in [16] and [25], the idle energy consumption of a 48-port switch ranges from 70W to 150W. About 40 watts or more of energy consumption can be added if the switch is working at its full capacity. These measurements also indicate that each switch port only consumes 1-2 Watts. On the other hand, Benson et al. [5] show the usage characteristics of the links in a Data Center and found that the actual production Data Centers have an average use of the links in the aggregation layer of about 8% for 95% of the time, while the average use of the links in the edge layer and the links in the core layer are approximately 20% and 40%, respectively. Leveraging low-utilization links provides ample space to consolidate data streams, so fewer links and switches are enough to support existing data traffic.

From the results presented in [16] and [25], the energy consumption equation of a switch can be defined as

$$PC_{Switch} = PC_{Chassis} * \sum_{i=1}^{Ports} PC_{Port(i)} \tag{1}$$

where:

- PC_{Switch}: total energy consumed by a switch.
- $PC_{Chassis}$: energy consumed by the switch chassis, based on the type of *Chassis* (it can be Edge, Aggregation, or Core).
- *Ports*: number of active ports on the switch.
- $PC_{Port(i)}$: energy consumption that each type of port will have based on its speed (1000/100/10 Mbps) or its state (on or off).

The equation consists of calculating the total energy consumption of each switch within of the Data Center network topology by checking the key parameters that influence consumption. The energy consumption of the switch chassis, the number of ports in use on each line-card, and the speed of each port that can be 10 Mbps, 100 Mbps or 1 Gbps, are taken into account in the equation.

For instance, considering a linear topology with 10 switches, where each switch has 1 server connected to 100 Mbps and between the switches, there is a 1 Gbps link. Based on the parameters defined, we know that the chassis of each switch consumes on average 60W, each 1 Gbps port has a consumption of 1W and each port of 100 Mbps has an average consumption of 0.3W. Thus, we can verify that the topology consumption shown in Figure 1, based on the model is 621 Watts, since there are 18 ports with a speed of 1 Gbps interconnecting the switches and 10 ports with speed of 100 Mbps connecting servers to 10 switches.

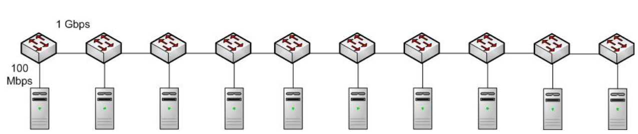

Figure 1: Example of the proposed model of energy consumption

3.1 Flow Mapping Strategies for DCN

The use of bin-packing heuristics is intended to improve the scalability of the consolidation of flows in the Data Center network. Optimum solutions are not guaranteed, but in practice, they can result in good results. For each stream, the algorithms evaluate possible paths and choose what has sufficient capacity. When all flows are assigned (which is not guaranteed), the algorithm return the sets of network assets (switches and links that flows passed) in addition to each used path.

The paths in a network can be easily represented as a graph (G), a set of vertices V and edges E, where each edge connects two nodes $G = (V, E)$, and a weight is assigned to each edge. In graph theory, the calculation of shortest paths between two vertices is a classic problem, among the variations of this problem, we use Dijkstra that calculates the shortest path length from a given source node $s \in V$ for a given destination node $t \in V$ [8]. Based on this, we developed strategies of different weights and with varying factors of oversubscription.

We implemented three flow mapping strategies with the intention of saving energy for DCN. First-fit finds the first path (bin) that attends the flow demand, thus seeking the path that passes through a smaller number of switches to map the flow. Best-fit evaluates all available paths and maps the stream based on the link that provides the least bandwidth capacity, but that capacity must be sufficient to meet the demand of the flow. Worst-fit evaluates all available paths but maps the flow based on the path that has more capacity left over.

The model created for First-fit has the premise that all links have weight equal to 1, so the Dijkstra algorithm will look for the path that passes through a smaller number of switches. The First-fit strategy maps the flows using Shortest paths to the edges with sufficient capacity, defining the parameter weight equal to 1, i.e., equal weights for all links at all ends of the vertices of the graph. In the Best-fit strategy, the model has the premise that the links have the capacity of the link as a weight. Therefore, the Dijkstra algorithm will fetch the path with the lowest capacity used. The mapping of flows with the Best-fit strategy tries to find a link that provides the least bandwidth capacity, but that capacity must be sufficient to meet the demand. In the Worst-fit strategy, the links have the inverse of their capacity links have the inverse of the capacity of the links. So, the Dijkstra algorithm will fetch the remaining leftover path from the links. In this strategy, the weight parameter is used as edge weight by the Shortest path algorithm, the weight assigned to each link is proportional to the inverse of its capacity, $\frac{1}{1000}$, $\frac{1}{100}$ e $\frac{1}{100}$.

The strategies use analogous initial approaches in the process of mapping flows. Initially, after the creation of the flows, all link capacities are increased by the 1:1, 1:5 and 1:20 oversubscription factors. For example, in a 1:5 oversubscription factor the capacity in each link is set to 5 times its real capacity during the mapping phase, i.e., a link can handle 5 times more demand than in a 1:1 factor. Then, for each pair (u, v) of origin and destination, we calculate the set of paths that support the total flow demand (bandwidth). If the link capacity is less than the bandwidth, then the edges are removed from the topology. The path selection process continues until the flow requirements for all sources and targets are met.

3.2 Strategies to Reduce Energy Consumption for DCN

Based on the research related to energy consumption in Data Center networks performed by Mahadevan et al. [15], a set of three main strategies were defined (power on/off links, traffic mapping and link speed adaptation), which were exploited for the development of the algorithm responsible for power saving.

Algorithm 1 presents the steps taken for the application of strategies to reduce energy consumption. Lines 2, 3 and 4 consist on the verification of the links that did not have mapped flows and as a consequence, their disconnection. Between lines 7 and 15, it is checked if the switches have active links, if there are no links in use, switch is turned off. Lines 18, 19 and 20 reduce the speed of the ports with low utilization

Algorithm 1 Algorithm for the application of energy reduction strategies

```
1:  procedure EnergyReductionMechanisms(topology)
2:      for all link ∈ topology.links do
3:          if link.maxcapacity = link.capacity then
4:              link.status ← off
5:          end if
6:      end for

7:      for all switch ∈ topology.switches do
8:          usedlinks ← 0
9:          for all link ∈ switch.links do
10:             if link.status = on then
11:                 usedlinks ← usedlinks +1
12:             end if
13:         end for
14:         if usedlinks = 0 then
15:             switch.status ← off
16:         end if
17:     end for
18:     linkcapacities ← [10, 100, 1000]
19:     for all link ∈ topology.links do
20:         usedcapacity ← link.maxcapacity - link.capacity
21:         for all lcap ∈ linkcapacities do
22:             if usedcapacity <= lcap then
23:                 link.maxcapacity ← lcap
24:                 BREAK
25:             end if
26:         end for
27:     end for
28: end procedure
```

Algorithm 2 presents the steps to calculate the energy consumed by the topology. On lines 2, 3, 4 and 5 it is verified whether the switches are connected. If true, consumption is based on previously defined values. The same happens with the ports of the switches in lines 6, 7 and 8, that after having their status checked, has the consumption values for each type of port assigned. Finally, the total consumption of the topology on line 13 is returned.

Algorithm 2 Algorithm for calculating energy consumption

```
1:  procedure EnergyConsumptionCalculation(topology)
2:      energycost ← 0
3:      for all switch ∈ topology.switches do
4:          if switch.status = on then
5:              energycost ← energycost + switch.cost
6:              for all link ∈ switch.links do
7:                  if link.status = on then
8:                      energycost ← energycost + link.cost
9:                  end if
10:             end for
11:         end if
12:     end for
13:     return energycost
14: end procedure
```

4 EVALUATIONS AND DISCUSSION

For the construction of the fat-tree topology, a simulator was developed in Python to automatically generate the Data Center scenarios used in the experiments, implementing the algorithms of the flow mapping strategies and performing the energy consumption calculation based on the proposed consumption model. We used the Fast Network Simulation Setup (FNSS) [20], which is a set of network tools that allows researchers to simplify the process of creating scenarios for network experiments. It enables the analysis of different topologies, based on a dataset or generator, and configuration of the characteristics of the links such as capacity, weights, delay, and sizes of workflows. FNSS uses internally the NetworkX library [21] to construct the graph, which in turn performs the creation, manipulation and study of the dynamic structure and function of complex networks.

The tests were performed on a Fat-tree topology with different strategies, numbers of PODs, port speed, network loads, and number of flows.

The proposed strategies were submitted to the traffic patterns *Random*, $Stride(1)$ and $Stride(i)$ [3]:

- *Random*: A host sends to any other host on the network with uniform probability. The source and destination are randomly selected. This traffic pattern replicates a normal process in which applications are placed randomly in the Data Center network.
- $Stride(1)$: The destination of a stream from the host x is the host $[(x + i) \ mod \ (numberofhosts)]$, where the hosts are numbered from the left to the right as 0, 1, ..., hosts-1.
- $Stride(i)$: It is characterized by inter-pod communication, thus forcing the use of the core layer. Where k is the quantity of pods used in experiments ($k = (4, 8, 12)$).

Besides, we have implemented the network configurations - (1Gb, 100Mb, 100Mb) - where the communication speeds respectively represent the speed of links that connect the Core to the aggregation layer, attaching the Aggregation to the Edge layer, and attaching the Edge layer to the hosts. Three network loads of 20%, 50%, and 80% were tested, with an average of 10 flows per host. Table 1 shows the different parameters used in the experiments.

Table 1: Configuration of experiments

Experiments			
POD Size	k=4	k=8	k=12
Network Configuration	1000,100,100		
Workloads	*Random*	$Stride(1)$	$Stride((k/2)\hat{}2)$
Oversubscription Factors	1:1	1:5	1:20
Network Load	20%	50%	80%
Strategies	First-fit	Best-fit	Worst-fit

4.1 Experiments and Results

In this section, we compare the energy savings of the proposed strategies with respect to traditional DCN energy consumption. Due to the large number of results obtained from the combination of algorithms and variation of the network configurations, aggregated data are presented that reflect the detailed results. We focused on the experimental results that are presented in the Figures 2, 3 and 4, where the energy consumption in kW/h is shown on the Y axis. In the Figures 2(a), 3(a) and 4(a) the results obtained using the *Random* traffic pattern are displayed, where it is possible to check the allocation of flows dynamically by the standard of traffic, where several streams may have passed through the same link, and since there was no change in the traffic capacity of the network, some streams were not mapped. The occurrence of 2 unmapped flows was verified, which is equivalent to 1.25% of the total in First-fit, Best-fit, and Worst-fit with the 1:1 oversubscription factor as can be observed in Figure 2(a), differently of factors 1:5 and 1:20 that had all the mapped flows and generated an energy saving of 46.22%, where as with the 1:1 factor the energy saving was 14.22%.

Applying the network configuration $k = 4$, we had the occurrence of 32 unmapped flows, which is equivalent to 2.5% of the total with the use of the 1:1 oversubscription factor, and the 1:5 and 1:20 factors without the incidence of unmapped flows. In Figure 3(a) it was possible to verify that the use of 1:5 and 1:20 oversubscription factors provided to the decrease in the number of unmapped flows and also, energy savings compared to the 1:1 factor with 80% load. The oversubscription factor 1:20 obtained the energy savings rate of 69.74% using the First-fit and Best-fit strategies. In comparison with the 1:1 and 1:5 factors that obtained energy savings rates of 25.54% and 60.67% lower than the traditional topology respectively, the energy savings achieved the best factor using 1:20 factor. It is due to the high number of hosts (432) with the randomly generated $k = 12$ topology and the 80% load. Regarding the unmapped flows, it should be noted that the use of the 1:1 factor in combination with the First-fit, Best-fit, and Worst-fit strategies presented 146, 274 and 146 unmapped flows, which is equivalent to 3.38%, 6.34%, and 3.38% of total flows respectively.

In Figure 3(c), it is possible to verify a variation of the energy consumption reduction rate for the high network load using the 1:1 oversubscription factor. This variation occurred with the First-fit and Best-fit strategies, which had energy consumption values with a high network load of 21.80%. When we applied the oversubscription factors 1:5 and 1:20, the costs were unchanged, with a rate of 64.21% lower than traditional energy consumption using First-fit and Best-fit flow mapping strategies. However, the Worst-fit strategy remained at the same energy-saving rate for the three oversubscription factors, keeping the rate at 3.25% lower than traditional energy consumption. This variation is evident in Figure 3(c), because as the network load between the 128 hosts of the fat-tree topology with $k = 8$ increases to 50% and 80%, the 1:1 factor keeps a greater number of equipment and links active to meet the flow generated. Even with the reduction of the economy rate with the high load, all the flows were mapped by the strategies, regardless of the used oversubscription factor.

In Figure 4(c) it is possible to observe a variation of the energy consumption rate for the network load of 80% using the 1:1 oversubscription factor. This variation occurred with the First-fit and Best-fit strategies, which presented lower energy consumption than the traditional topology of 28.23% when used the network load of 80%. Besides the use of factors 1:5 and 1:20, the values presented variation only in 1:5 oversubscription factor with a load of 80%. When we used 1:20 oversubscription factor, it did not undergo modifications regardless of the network load applied, with a rate around 70% lower than traditional energy consumption with First-fit and Best-fit flow mapping strategies.

The Worst-fit strategy, however, remained at the same energy-saving rate for the three oversubscription factors, keeping the rate of 4.67% lower than traditional energy consumption. The variation of the energy consumption using the 1:1 oversubscription factor is evident in Figure 4(c), as the network load among the 432 hosts of the fat-tree topology with $k = 12$ increases 20% to 50% and 50% to 80%, because the 1:1 oversubscription factor maintains a higher number of devices and active links to meet the generated flow.

4.2 Discussion

The impacts of fat-tree topology sizes on energy savings for the proposed strategies were compared. A fat-tree topology with a pod k consists of $k^2/4$ core switches, $k^2/2$ switches of aggregation, $k^2/2$ border switches and $k^3/4$ hosts. For the traffic patterns, the number of hosts was changed by changing the k parameter. Then, the source and destination of the flows for a set of hosts were randomly assigned. When comparing energy savings to the fat-tree topologies of size $k = 4$, $k = 8$ and $k = 12$, it was possible to verify that energy consumption increases in relation to traditional consumption. The decrease of the energy consumption from the use of Best-Fit and First-Fit strategies was boosted with the variation of the size of the topology. It happened because, with the increase of the k-value in the fat-tree topology, the strategies could choose among more paths and to map the flow efficiently.

Because the pattern of traffic in a DCN varies widely, the use of different oversubscription factors may be an opportunity to ensure the correct mapping of flows with a relevant energy efficiency index. Based on the results obtained, it was possible to verify that the use of different factors of oversubscription, especially in large environments, aggregates availability to the links. It occurs because network oversubscription provides sufficient bandwidth capacity to match the service guarantees of compromised bandwidth networks, which ended up happening at various times with high network load and regular traffic of *Random* over the experiments.

When we applied factors 1:5 and 1:20, the energy saving rate in the network reached 70.02% on a fat-tree topology of size $k = 12$ using 432 hosts, and 64.82% on a fat-tree topology of $k = 8$ using

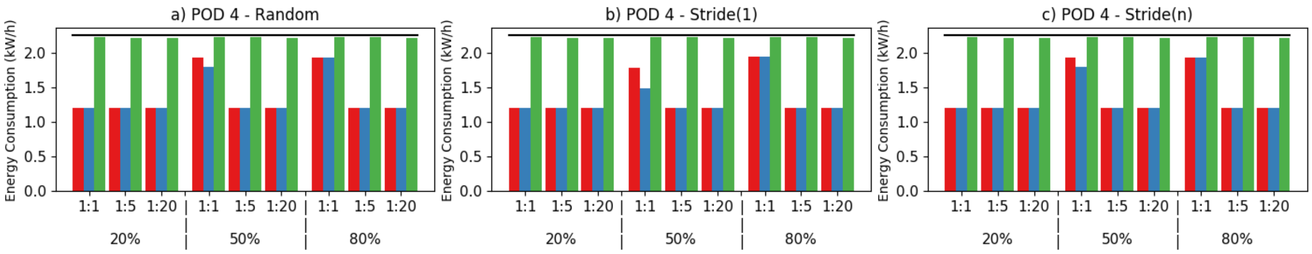

Figure 2: Simulation results with POD = 4

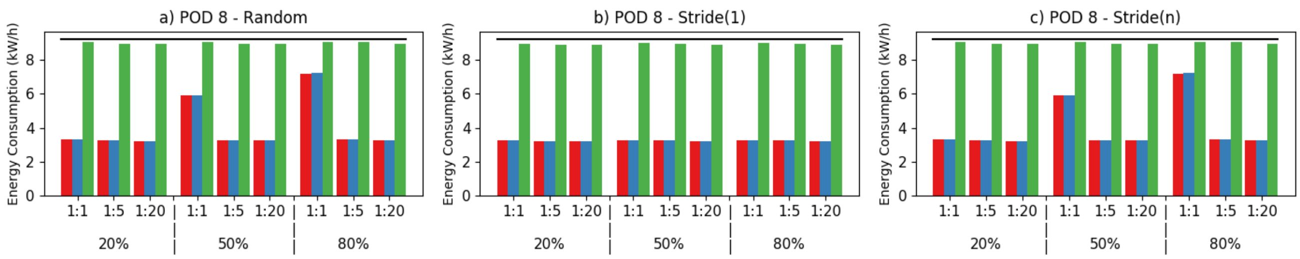

Figure 3: Simulation results with POD = 8

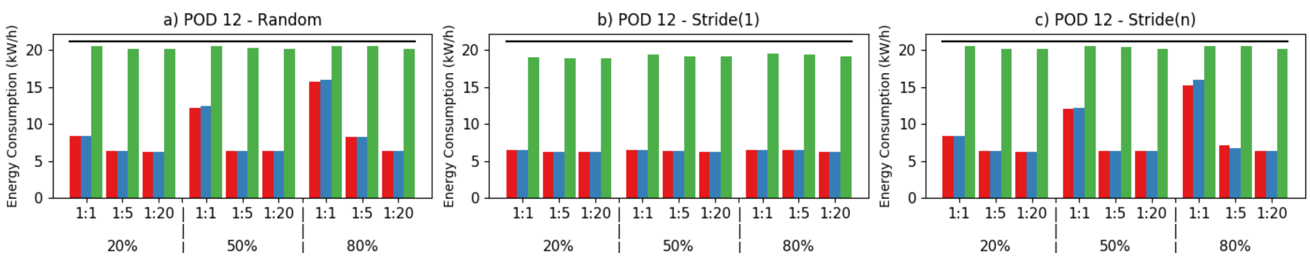

Figure 4: Simulation results with POD = 12

128 hosts. The energy savings concerning the 1:1 oversubscription factor for the same topologies remained the same reached for $k = 12$, varying to $k = 8$. It is mainly since when using the 1:1 oversubscription factor, it was necessary to keep more equipment and ports connected, thus influencing the final energy consumption of the topology.

After analyzing the results, it is possible to infer that energy efficiency can be achieved using two fundamental choices: (i) increase the data rate, or in other words reduce the amount of time required for the data to travel through the physical environment, thus reducing the time that network equipment needs to be connected, in order to consume less energy within the transmissions, or (ii) the data rate may be reduced and therefore less energy consumption can be achieved at the expense of transmission times of longer data. In contrast, in an SDN-managed network, various parts of the SDN structure can be dynamically configured to reduce energy consumption. One way is to set the flow according to network traffic and put unused devices on the network in sleep mode. When there is low traffic load, instead of the entire device, certain ports may be put into sleep mode. In addition, the SDN controller can put new rules on energy efficiency in switches under some restrictions, such as number of switches and traffic engineering policies. The SDN controller may seek to minimize energy consumption while reconfiguring new rules according to DCN requirements.

5 CONCLUSIONS AND FUTURE WORK

Data Centers consume a tremendous amount of energy. Researchers have dedicated their efforts to improve the efficiency of server consumption. However, insufficient attention has been given to the effectiveness of DCN. This work focused on energy efficiency for DCN, using a combination of different strategies to reduce energy consumption such as switching off switches and ports, reducing low-speed port speeds, and consolidating traffic based on efficient mapping of the flows.

In our experiments, we found a reduction in the energy consumption up to 70.02% using a fat-tree topology of size $k = 12$ (oversubscription factors of 1:5 and 1:20 - where there are no energy savings due to lack of any shutdown opportunities). The use of different oversubscription factors in DCNs may be an opportunity to ensure the correct mapping of flows with a relevant index of energy efficiency.

Although the SDN paradigm has been presented as an ally in solving the energy consumption problem in DCN, it may also bring some new problems that need to be verified. For example, implementing energy-centered approaches based on SDN may result in an increase in response time as well as finding an optimal set of active links and the flow of traffic may take longer than expected and any attempt adaptation may fail or lose performance on rapidly changing channels. Therefore, additional effort must be made in this direction.

As a future work, it is suggested to implement the algorithm in an OpenFlow controller, to execute both the allocation of the network flows and the establishment of the connections, as well as the maintenance of the links, exploring the centralization functionalities of the SDN controllers. The approach of this work was concentrated in a Data Center on an individual basis. As the growth in demand for cloud computing applications, a future research point is how to create distributed and energy efficient Data Centers. Reducing energy consumption in geographically distributed Data Center architectures is more complex as many factors have to be taken into account, such as: server localization, site load balancing, and optimization of the topology overlay virtual network.

REFERENCES

[1] Dennis Abts, Michael R Marty, Philip M Wells, Peter Klausler, and Hong Liu. 2010. Energy proportional datacenter networks. In *ACM SIGARCH Computer Architecture News*, Vol. 38. ACM, 338–347.

[2] Muhammad Abdullah Adnan and Rajesh Gupta. 2013. Path consolidation for dynamic right-sizing of data center networks. In *IEEE 6th International Conference on Cloud Computing (CLOUD)*. IEEE, 581–588.

[3] Mohammad Al-Fares, Alexander Loukissas, and Amin Vahdat. 2008. A scalable, commodity data center network architecture. In *ACM SIGCOMM Computer Communication Review*, Vol. 38. ACM, 63–74.

[4] M Faizul Bari, Raouf Boutaba, Rafael Esteves, Lisandro Z Granville, Maxim Podlesny, Md Golam Rabbani, Qi Zhang, and Mohamed Faten Zhani. 2013. Data center network virtualization: A survey. *IEEE Communications Surveys & Tutorials* 15, 2 (2013), 909–928.

[5] Theophilus Benson, Ashok Anand, Aditya Akella, and Ming Zhang. 2010. Understanding data center traffic characteristics. *ACM SIGCOMM Computer Communication Review* 40, 1 (2010), 92–99.

[6] Manu Budhiraja and Shubhra Saggar. 2013. Green IT: Harvesting Heat using TPV (Thermophotovoltaic). *International Journal of Managment, IT and Engineering* 3, 9 (2013), 323–333.

[7] Rodrigo N Calheiros, Rajiv Ranjan, Anton Beloglazov, César AF De Rose, and Rajkumar Buyya. 2011. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience* 41, 1 (2011), 23–50.

[8] Nicos Christofides and Graph Theory. 1975, 415p. *An algorithmic approach.* New York: Academic Press Inc.

[9] Edward G Coffman Jr, Michael R Garey, and David S Johnson. 1996. Approximation algorithms for bin packing: a survey. (1996), 46–93.

[10] Brandon Heller, Srinivasan Seetharaman, Priya Mahadevan, Yiannis Yiakoumis, Puneet Sharma, Sujata Banerjee, and Nick McKeown. 2010. ElasticTree: Saving Energy in Data Center Networks. In *7th USENIX Conference on Networked Systems Design and Implementation*, Vol. 3. 19–21.

[11] T Huong, Daniel Schlosser, P Nam, Michael Jarschel, N Thanh, and Rastin Pries. 2011. ECODANE—reducing energy consumption in data center networks based on traffic engineering. In *11th Würzburg Workshop on IP: Joint ITG and Euro-NF Workshop Visions of Future Generation Networks (EuroView2011)*.

[12] Jonathan Koomey. 2011. Growth in data center electricity use 2005 to 2010. *Oakland, CA: Analytics Press. August* 1 (2011), 2010.

[13] Teemu Koponen. 2012. Software is the Future of Networking. In *Proceedings of the Eighth ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS '12)*. ACM, New York, NY, USA, 135–136.

[14] Bob Lantz, Brandon Heller, and Nick McKeown. 2010. A Network in a Laptop: Rapid Prototyping for Software-defined Networks. In *9th ACM SIGCOMM Workshop on Hot Topics in Networks (Hotnets-IX)*. ACM, New York, NY, USA, Article 19, 6 pages.

[15] Priya Mahadevan, Sujata Banerjee, and Puneet Sharma. 2010. Energy Proportionality of an Enterprise Network. In *First ACM SIGCOMM Workshop on Green Networking (Green Networking '10)*. ACM, New York, NY, USA, 53–60.

[16] Priya Mahadevan, Puneet Sharma, Sujata Banerjee, and Parthasarathy Ranganathan. 2009. A Power Benchmarking Framework for Network Devices. In *8th International IFIP-TC 6 Networking Conference (NETWORKING '09)*. Springer-Verlag, Berlin, Heidelberg, 795–808.

[17] Jennifer Mankoff, Robin Kravets, and Eli Blevis. 2008. Some computer science issues in creating a sustainable world. *Computer* 41, 8 (2008), 102–105.

[18] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008. OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review* 38, 2 (2008), 69–74.

[19] Tran Manh Nam, Nguyen Huu Thanh, Ngo Quynh Thu, Hoang Trung Hieu, and Stefan Covaci. 2015. Energy-aware routing based on power profile of devices in data center networks using SDN. In *12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. IEEE, 1–6.

[20] Lorenzo Saino, Cosmin Cocora, and George Pavlou. 2013. A toolchain for simplifying network simulation setup. In *Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 82–91.

[21] Daniel A Schult and P Swart. 2008. Exploring network structure, dynamics, and function using NetworkX. In *7th Python in Science Conferences (SciPy 2008)*, Vol. 2008. 11–16.

[22] Mehmet Fatih Tuysuz, Zekiye Kubra Ankarali, and Didem Gözüpek. 2017. A survey on energy efficiency in software defined networks. *Computer Networks* 113 (2017), 188–204.

[23] Sergio Roberto Villarreal, María Elena Villarreal, Carlos Becker Westphall, and Carla Merkle Westphall. 2014. Legacy Network Infrastructure Management Model for Green Cloud Validated Through Simulations. *International Journal on Advances in Intelligent Systems* 7 (2014), 124–135.

[24] Tran Hoang Vu, Pham Ngoc Nam, Tran Thanh, Nguyen Duy Linh, To Duc Thien, Nguyen Huu Thanh, et al. 2012. Power aware OpenFlow switch extension for energy saving in data centers. In *International Conference on Advanced Technologies for Communications (ATC)*. IEEE, 309–313.

[25] Xiaodong Wang, Yanjun Yao, Xiaorui Wang, Kefa Lu, and Qing Cao. 2012. CARPO: Correlation-aware power optimization in data center networks. In *IEEE 31st Annual International Conference on Computer Communications (INFOCOM)*. IEEE, 1125–1133.

[26] Y. Zhang and N. Ansari. 2012. HERO: Hierarchical energy optimization for data center networks. In *2012 IEEE International Conference on Communications (ICC)*. 2924–2928.