Paper

Optimized Energy Aware Resource Allocation Algorithm Using Software Defined Network Technology

Ranya Al-Musawi and Obada Al-Khatib

Faculty of Engineering and Information Sciences, University of Wollongong, Dubai, UAE

https://doi.org/10.26636/jtit.2019.129418

Abstract—The number of data centers (DCs) used for storing and processing data has evolved rapidly in recent years. However, the operations held by DCs may relate to a number of disadvantages, primarily presuming in excessive energy and power consumption due to the poor management standards applied. This may lead to a situation in which many devices within the DC operate at full capacity without any tasks assigned for actual execution. A Software Defined Network (SDN) is a network architecture where the control plane is an independent entity from the data plane, yielding to a higher controllability and flexibility over the network. Through the utilization of SDN architecture, a highly functional energy aware network may be established. In this paper, we propose a heuristic algorithm that monitors the current status of an SDN network (in addition to all ingoing and outgoing traffic), in order to dynamically and efficiently allocate network resources by ensuring that only the necessary network devices are active and by turning the idle ones off. The results show that the proposed algorithm reduces energy consumption of the network compared to existing solutions.

Keywords—computer network management, network servers, software defined networking, virtual machines.

1. Introduction

The data center (DC) is considered to be the heart of any organization or company, since it is responsible for all networking operations and for the handling of all ingoing and outgoing data [1], [2]. The hardware that may be found inside a data center includes servers, switches, routers, storage devices, etc., with all of them interconnected via a backbone network to create a comprehensive solution enabling global information exchange [3], [4]. Thus, the term data center often relates to an enormous area or an entire building that houses networking equipment and infrastructure [3], [5]. The exponential increase in Internet traffic calls for the construction of more DCs to ensure that massive storage capacities and fast processing speeds are guaranteed. However, building more DCs will increase the operational costs due to increased energy consumption [6]–[8].

Extensive research has been conducted to ensure energy efficient operation of DCs [7], [9]–[11], taking into consideration various, energy-intensive DC systems, such as the cooling installation, for instance [5], [6]. Researchers have proposed the use of renewable energy sources to power DCs [6], [12], [13]. However, these sources do not offer sufficient reliability levels due to their dependency on weather conditions (sun or wind) [6], [14]. Others have proposed creating a hybrid system that integrates non-renewable and renewable energy sources to balance energy expenses and pollution level [6], [12]–[14].

Excessive energy consumption of DCs may also be caused by poor allocation of DC resources. For example, many devices in a DC may be actively operating at full capacity without being assigned any tasks [5], [6]. Therefore, optimized resource allocation algorithms are proposed in order to handle traffic efficiently and to turn idle devices off [7], [10], [15], [16]. However, switching off some devices might cause degradation in Quality of Service (QoS) and Quality of Experience (QoE), especially during the peak load periods. Thus, such algorithms must consider maintaining acceptable QoS and the QoE levels [8], [10], [11].

The primary aim of this paper is to highlight and resolve the underlying consequences of the failure to manage DC resources in an efficient manner – a phenomenon that leads to high energy consumption, loss of packets, delays, as well as degraded QoS and QoE levels. The solution focuses on optimizing resource allocation and traffic routing procedures within the DC. The proposed network topology is based on Software Defined Networking (SDN) which separates the control plane from the data plane. By doing so, SDN offers good programming flexibility and enables dynamic adjustment based on the current network state and on the network's incoming traffic.

The network will function in the following manner. The main server will be used as a controller unit to manage and supervise all network operations, while other servers will be used to process the traffic and execute the appropriate actions based upon the commands that they receive from the host controller. The controller is responsible for monitoring and analyzing the network's status, in order to take optimized decisions on managing the data flow and alternating the on/off state of the network devices. Physical servers are consolidated and reconfigured into multiple virtual machines (VMs). This is done for the purpose of reducing the number of active devices within the network and, therefore, decreasing the amount of energy and power drawn by the DC. Servers are able to host multiple VMs to make full use of their capacity, where each VM represents a process/task requested by a client. The capacity of each VM is adjusted based on the volume of traffic emerging from the client. Likewise, the controller prioritizes incoming traffic by providing more resources to the client with a higher traffic volume. This approach is used to assure that no overload is encountered for the purpose of minimizing the chances of experiencing packet loss or delays, and to maintain satisfactory QoS and QoE levels. To avoid overload, the controller will not route traffic to a network device if its load exceeds a specified threshold value.

The rest of this paper is organized as follows. Section 2 discusses different research papers addressing the proposed issue. Section 3 presents detailed information about the methodology used while developing the proposed energy saving system. Section 4 focuses on the results retrieved from the emulated network, to prove the successful operation of the proposed solution.

2. Literature Review

The cost of electricity used by DCs is expected to approach 8% of the overall cost of electric power used worldwide by 2020 [5], [7]. Therefore, energy saving techniques have evolved, with main advances achieved in the area of chip-, infrastructure- and system-level energy saving technologies [17].

The chip-level energy saving approach highlights a dynamic adjustment of CPU frequency and voltage supply to decrease overall power consumption [17]. The infrastructure-level energy saving technology consists in installing efficient cooling and heat dissipation systems to avoid high temperatures in DCs [17], [18]. Moving forward, the system-level energy saving method consists in dynamically arranging and distributing workload and tasks between the available devices [17]. However, the underlying question is whether those methods would compromise QoS or QoE offered to end users. The answer would be positive in some cases. However, novel solutions have appeared to address the issue of low QoS and QoE [4].

2.1. Minimizing Cost While Going Green

Going green refers to consuming the minimum amount of energy and reducing carbon dioxide emissions. Experts suggest that renewable energy should be used as source of power for data centers [6]. However, renewable energy systems, e.g. those relying on the sun, require clear skies in order to operate efficiently. Wind is another source of renewable energy, yet its prevalence cannot be guaranteed [6]. To use both renewable and non- renewable energy sources increases the cost of data centers. In addition, the issue of overloads caused by massive traffic waves has not been addressed, resulting in low levels of quality experience by end-users, as well as in packet loss, delays and high latency levels.

2.2. Traffic

Statistics reveal that the amounts of power drawn by data centers during peak traffic loads are staggering [19]. However, since traffic intensity fluctuates between day and night, some of the network devices may be turned off to save power during light traffic conditions [20]. On the other hand, if some of the devices are to be turned off, this may cause serious issues and complications affecting data processing. For example, in the case of a traffic strike, the data center will be unable to handle the surge due to some of its devices being inactive [1], [20]. To resolve this matter, the controller should continuously monitor traffic loads and make decisions to switch devices on or off accordingly [10], [20], [21]. This process should be dynamic, in order not to affect ongoing operation of the center [20].

2.3. Routing Algorithms

It is important to maintain an approach that unravels how each task should be transmitted through the ideal path, in the sense of going through the minimum number of devices to get to the final destination that possesses the appropriate capacity to execute the task at hand. Ideal capacity refers to CPU and memory demands that are to be preserved according to the amount of traffic. Such an approach allows to use the resources rapidly to avoid packet loss and delays [4]. To achieve this, a mechanism should be available in the network to enable it to calculate the CPU power and other requirements of a given task, and to prioritize it accordingly [4]. A high priority task is provided with relevant resources and is given priority over other tasks, so that it may be processed faster. In addition, the algorithm that searches for the shortest path to direct traffic to the appropriate server for execution, should make sure that the server is unoccupied [21], [22]. For instance, when a server is overloaded, the SDN controller uses another node to migrate the traffic to an unoccupied server for processing.

2.4. QoS and QoE

QoS and QoE are two important factors when considering end user satisfaction and when rating DC capacity. QoS is determined by diminishing packet loss, delay and high latency levels [8], [23]. When subjected to any of those factors, the service offered to the end user subject to deterioration, due to the distorted or damaged nature of information received [23]. Delay, especially when receiving any type of visual information, may occur easily if the transmission time is not synchronized between the source and the client. High latency occurs mainly when the data center suffers from traffic overload and is unable to process data in an efficient and rapid manner. This causes late responses experienced by the end user. QoE depends on all three factors listed pertaining to QoS. However, delay and packet loss play a huge role in affecting QoE [8]. Two experiments in [8] were conducted to prove how packet loss and delay negatively affect the quality of a video stream.

2.5. Network Function Virtualization

In network function virtualization (NFV), hardware components are replaced by a software algorithm [8]. Such an approach results in less hardware being used in data centers. Hence, the reduced amount of energy consumed. Cost is also cut down, as DCs are no longer required to purchase expensive hardware, and providing maintenance is no more a consideration [8], [10]. Unlike traditional data centers, NFV technology enables easy control and surveillance of the execution of processes and facilitates the routing of traffic, especially when integrated with SDN technology [8], [10].

2.6. Software Defined Network

Software Defined Network (SDN) offers infrastructure providers (InPs) better control over the network. It enables easy modification of all components present within the network using a software algorithm [8], [10], [24]–[26].

The control plane offers full control over all components within a given network. It is able to monitor traffic, analyze work loads, migrate traffic, turn off idle servers and prioritize tasks in a dynamic manner using SDN technology [4], [27]. Statistical data about traffic load and analyses of the work processes are updated periodically in order to make dynamic decisions and to identify the necessary network changes accordingly [4], [7], [27], [28]. The control plane calculates CPU and bandwidth requirements of the tasks/requests in order to route them to the appropriate server that has the required capacity to process them rapidly. The requirements of various tasks are sorted based on priority. The higher the CPU and bandwidth requirement, the higher the priority granted to the task in order to process it quicker [4], [7], [27], [28]. This helps decrease latency within the data center, and also facilitates dealing with workloads in a dynamic manner. Interaction between the control plane and the data plane and the sending of control signals are based on the standard SDN protocol known as Open Flow, where cooperation with the forwarding plane is established to manage the data as per the main controller's instructions [8], [10], [24].

What distinguishes SDN technology from other existing network architectures is the flexible programmable interface. It may be effortlessly upgraded by updating the algorithm to suit the changing needs of the data center. This is in contrast with existing networks, where numerous changes need to be introduced on hardware and software levels if the data centre acquires an upgrade [7], [8], [10], [24]–[26].

SDN and NFV are two technologies that are highly similar. The main difference lies in the scope of responsibility of each technology. SDN is liable for global control, resource allocation, routing decision making and work flow process within a network, while NFV is an algorithm that executes those processes instead of having a hardware device to perform the task [8], [10], [26].

Implementing both technologies together creates a hybrid data center with flexible control over a network that is programmable [7], [27], [29], and conserves energy.

2.7. CPU Frequency

In [30] extensive research and tests have been conducted to construct a dynamic mechanism that changes the router's CPU frequency on single-core and multi-core processors, based on incoming traffic intensity. With low traffic intensity, CPU frequency is lowered to reduce power consumption. When incoming traffic intensity increases, CPU frequency is increased dynamically to meet the network's demands. This mechanism allows to save power while ensuring efficient traffic handling and management [30].

CPU frequency may be manipulated via hardware- or software-based methods [12]. In hardware, the logical processor's request to achieve a specific frequency rate is stored in a register, and then the processor with the highest required frequency will be chosen. The main drawback of the hardware method is the frequency difference between the processors, where one processor may experience a high load while running on a low CPU frequency, because the decision of other lightly loaded processors to maintain the low CPU frequency has been complied with. The solution is to implement software that utilizes the hardware-provided information about the load and provides the appropriate CPU frequency to each processor independently. Alternatively, it may balance the running load of the applications in a manner that evenly distributes them amongst the processors to provide an equivalent dynamic frequency rate for all processors [12].

3. Methodology

The factors that cause excessive energy consumption may be divided into two categories: internal and external. External causes include poor placement of network components inside the data center, where many active devices are purchased and used to accommodate the incoming traffic and to avoid overload. Internal causes related to inefficient resource allocation inside the network, leading to overloads. This may cause glitches, where some network devices stop responding, yielding to delays, high latencies and packet loss. Here, we focus on internal resource allocation, with its purpose being to evade overload and to ensure maximized use of all network device resources. This ensures that no energy is dissipated without being used. The number of physical active devices inside the DC network is minimized as well.

3.1. Proposed Energy Saving System

The proposed energy saving approach consists in employing an optimized dynamic resource allocation heuristic algorithm that controls the flow of traffic and manages it in a dynamic manner. Dynamic management refers to creating a system that has the ability to monitor traffic and analyze the workload inside the DC based on information collected and processed, with traffic being then routed to the appropriate destination.

The network system will be designed and created using C++, relying both on SDN and NFV technologies. The system will have the form of a small network for demonstration purposes; however, the heuristic may be incorporated into large scale networks as well. The aim is to attain results that reveal a huge reduction in energy consumption, while accomplishing a network resource utilization rate. The results may be verified by monitoring statistical data available after the programs were executed multiple times, as the process of the users' arrival to the system is random. The success of the algorithm heuristic will also be confirmed through the comparison of the amount of energy consumed in the proposed approach and in other solutions found in traditional DCs. The network has a main server, which will be considered as the controller shown in Fig. 1. The other servers in the network are responsible for executing and processing data based upon the command signals that they receive from the controller. The controller has the privilege of supervising all of the network devices, such as switches, hosts and virtual machines. The network has two switches, each connected to two servers or hosts, where each host has a number of virtual machines (Fig. 1).

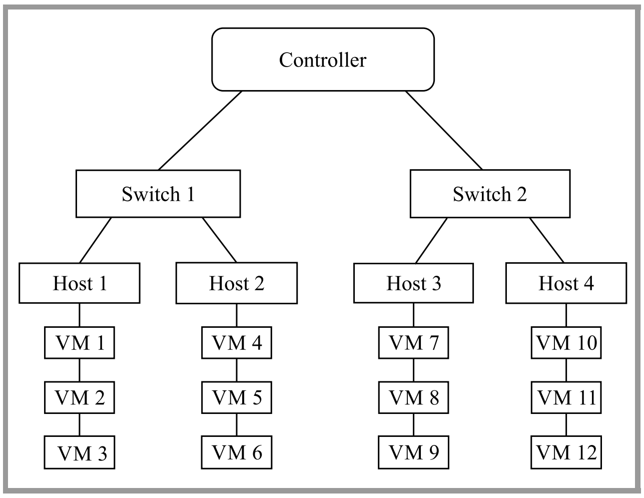


Fig. 1. The proposed network architecture.

The switches are of the Cisco Catalyst 4948 WS-C4948-S variety. When in the idle state, the power drawn by the switch is 176 W, which equals approximately 58.6% of the full load power of 300 W. To avoid unnecessary energy dissipation, idle switches are dynamically turned off when they are not in use. However, when a huge wave of traffic enters the network, they are turned back on. To avoid overload on the switches and to reduce the power consumed by them, a maximum threshold for power and capacity is set at 95% of the original power and capacity, translating to 285 W at 253 MHz. The controller will take into consideration the threshold as the highest permitted capacity and power consumption value which should not be exceeded when calculating and routing traffic towards the designated switch.

The servers (hosts) are based on Intel Xeon Quad Core processors, and operate at 2.27 GHz. The power consumed by idle servers amounts to 100 W a piece, which equals 71.4% of the full load power of 140 W. To avoid energy dissipation, idle servers or servers with no traffic are dynamically turned off. Similarly, when a huge wave of traffic enters the network, those servers are turned back on to accommodate the workload. The maximum power and capacity threshold for the servers is set at 95% of the original power and capacity level, translating to 133 W at 2.16 GHz. This allows to avoid overload and surges in energy and power consumption.

3.2. Algorithm Insights

The controller maintains the maximum threshold values for the switches/servers and ensures that they are not exceeded by continuously monitoring the available capacity and the incoming workload, and by calculating the power of the server/switch k as:

$$P_k = r P_{\max} + (1 - r) P_{\max} U_k(t) b_k \ , \qquad (1)$$

where $0 \leq r \leq 1$ is the percentage of time in which the server is idle, $P_{\max}$ is the maximum power of the server/switch at full load, $U_k(t)$ denotes the utilization of server/switch k, and b_k is a Boolean number that is zero if server/switch k does not have any assigned tasks and it will be set to one if switch/server k has been assigned a task for execution. The utilization, $U_k(t)$, is calculated as:

$$U_k(t) = \frac{C b_k}{\text{Original capacity}} \ , \qquad (2)$$

where C represents the utilized capacity of switch/server k.

The number of the incoming clients is preset. However, the amount of traffic that each client conveys is random.

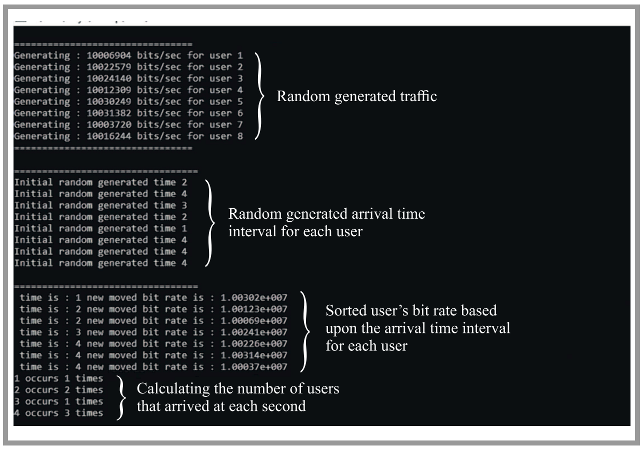


Fig. 2. Users with random bitrates initially enter at random times and are then sorted in an ascending order.

In addition, the arrival time of each client is also random. After completion of the network analysis process, the controller monitors the incoming traffic and sorts the users according to their arrival times, as shown in Fig. 2, and then calculates the remaining available capacity on each switch/server through Eqs. (1) and (2) to assure that the capacity is sufficient for all users arriving at the same time. If the bitrate of all users arriving at the same time is greater than the available capacities of all the switches and servers, the controller will allow the entrance of only some of the users, which the network components can accommodate without getting overloaded for the purpose of processing them, while other users will be terminated/blocked to avoid overload. Users who have been blocked/terminated may establish a new connection request later.

The aim of sorting the users' arrival times in an ascending order is to apply the FIFO management technique, where users who arrive first get served first, which is a crucial step towards assuring that the clients will not experience significant delays. Eliminating the overload problem eradicates session timeouts that lead to delays and loss of packets by the network. Figure 2 is a simple representation that deliberates the processes going on inside the network, assuming that eight random users with random bitrates ranging from 10 to 99 Mbps are generated with random arrival times ranging from 1 to 4 s. The algorithm will primarily sort the users in an ascending order based on their arrival time and will then determine the number of users entering at the same instant. It then starts with the second one and observes how many users arrived at that time (in Fig. 2, it is only one user). It compares the bitrates of those users against the available capacity on the switch. If the switch has enough capacity, the user is allowed to enter and is served. The user processing time will be calculated and when the traffic is processed inside the switch, the users exit and are be transported to the appropriate host destination, down to a virtual machine inside the host, where appropriate capacity is provided based upon the user's demands.

3.3. Power and Energy Saving

To reduce the number of active devices and energy consumption, servers are consolidated into VMs. This ensures maximum utilization, while keeping the rest of the servers/hosts off. When traffic comes from multiple clients, the controller will observe which client has the most vol-

Stage 1	1. Incoming traffic gets sorted in an ascending order based upon the user's arrival time. 2. The heuristic examines the number of user entering at each instance of arrival time.
Stage 2	1. The heuristic compares the capacity of the network components against the incoming traffic at each arrival time instance. 2. Users having a capacity that exceeds the network capacity threshold or may cause an overload, are terminated/blocked immediately.
Stage 3	1. The amount of process time for each user on the switch/host is calculated preliminarily. 2. User's traffic (data rate) is guided to one of the switches based upon previous stages analysis. 3. Available capacity left after the users enter the switch is displayed. 4. Idle switches/hosts are turned off.
Stage 4	1. Traffic exits the switch and is forwarded to one of the hosts based upon the available capacity. 2. Traffic is assigned to a virtual machine inside the host and is offered with the appropriate CPU capacity to accommodate the workload. 3. Available capacity left after the users enter the host is displayed. 4. Idle hosts are turned off.
Stage 5	1. Traffic exits the hosts. 2. The number of terminated/blocked (if any) users is displayed. 3. The amount of power consumed at each time interval on each network component is displayed

Fig. 3. Network stages of the proposed algorithm.

ume of traffic to move the data to the appropriate VM, granting it a higher CPU rate and providing priority over others to assure that no packet loss or delays occur. If the network faces a low traffic wave, it will direct the traffic through one of the switches while keeping the other switch off, and the same applies to the hosts. Subsequently, if the network components are finished with processing traffic and become idle, the controller will dynamically turn them off to decrease energy and power consumption. However, they will get turned back on if a huge amount of traffic enters the network. A feasible representation of the individual stages of the algorithm applied to incoming users is shown in Fig. 3.

3.4. Testing Procedure

The assessment of the proposed heuristic algorithm is an essential measure for revealing the reliability and the success of the new algorithm. Results may be observed as the program is executed, presenting a clear map of all processes going on inside the network at each stage. Starting from the moment the users enter the network at each time instant, until the time they exit. Statistics are also provided throughout the execution process to reveal how much capacity was used by each user on each switch and host. Furthermore, the algorithm displays the capacity left after each user has entered and exited the network device concerned. Likewise, at the end, the algorithm indicates the number of users that were terminated for the purpose of avoiding overloading the network. It also previews how much power has been consumed in each second and on each switch/host, as the users enter at different times.

4. Results and Findings

In this section, we disclose the results and findings based on the examination of the algorithm concerned. The emulated proposal employs random inputs, such as traffic and arrival times for each of the users. To validate functionality of the heuristic algorithm, a mapping representation may be drawn up showing user's progress through all routing stages inside the network, as shown in Fig. 3.

4.1. Resource Allocation and Mapping

In stage one, when the incoming traffic is processed and enters stages 2–4, the outcome may be clearly seen in Fig. 4. The program output is based on a sample of 108 different users who enter at random times ranging from 1 to 4 s. The observation that can be made is concerned with how the users are served the moment they enter the network and how the switches are turned on and off based upon the number of users and the amount of their traffic.

Primarily, the number of users and their traffic amount entering at each time instant are noted and then the available capacity on the network devices is computed. As a result, each user's traffic is routed to the device which has sufficient capacity to serve the user and the total time required to serve the user is calculated. For example, as depicted in Fig. 4, the users entering in second one are served by a switch and then they are directed out of the switch before the new users arrive at second two. The algorithm goes on until all users have been served and have exited the switch towards the hosts.

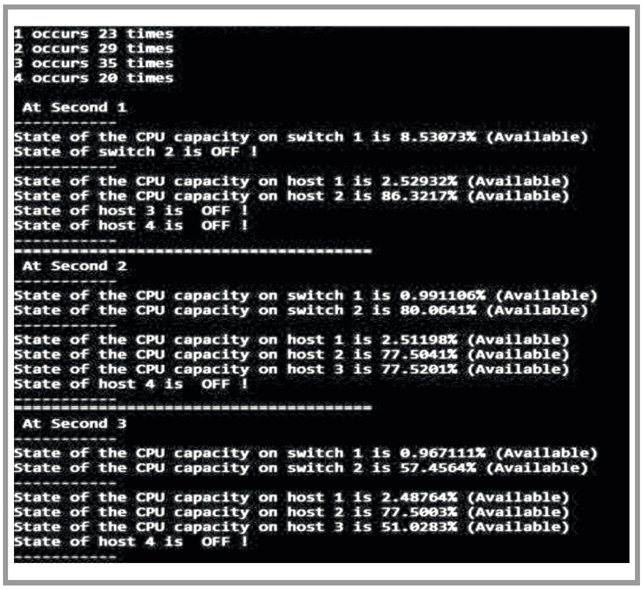


Fig. 4. Traffic handling and routing inside the network.

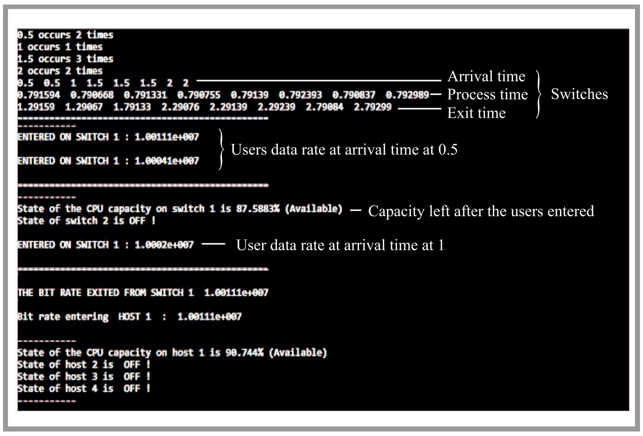


Fig. 5. Illustration of traffic with overlapping intervals between exit time and new arrival time are guided.

Another example is concerned with a situation in which the exit time of the current user(s) overlaps with the arrival time of new incoming user(s), as illustrated in Fig. 5. The controller took the users arriving at 0.5 s and guided them to switch one, after it has realized that their exit time is beyond second one, which is the arrival time of the next incoming user(s). The controller granted access to switch one to the user(s) arriving at second one and it has also turned off switch two because it state is idle. Notice that the controller keeps track of the available capacity of each switch on a continuous basis, after each arrival or exit of a new user. another observation may be made in Fig. 5, showing that as one user exits switch one at 0.5 s, it immediately enters host one, while keeping idle hosts 2–4 off. After the user has entered host one, it will be assigned to a virtual machine, where appropriate CPU capacity will be dynamically provided to accommodate the workload and to process it efficiently. This process continues until the traffic of all users has been processed and guided out of the network.

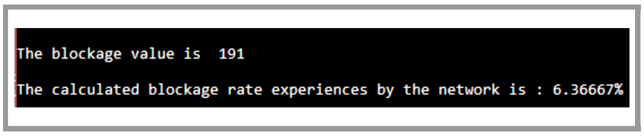


Fig. 6. Blockage rate.

The number of blocked users who were not served inside the network due to network capacity constraints and the threshold set to avoid overload, are displayed at the end, as shown in Fig. 6. QoS is maintained through the illustration on how the users are served. As shown in Fig. 5, all users who have been served by the network have not experienced any delays. That is, they have exited the network as per the expected exit time calculated initially when they entered the network. For the considered simulation parameters, the network has also not experienced any overload, and the blockage constraint in the proposed algorithm is incorporated for the purpose of avoiding bottlenecks, overload and delays. As these issues are evaded, it can be confirmed that the proposed algorithm does not compromise QoS.

4.2. Confirming Energy and Power Conservation

A test was conducted on the designed network with 204 random users and random incoming bit rates. Figure 7 unveils energy consumption, in J/s, after the users have passed through all network stages and have exited the network. Since switch two was turned off (energy is zero) throughout the entire test, due to the fact that no incoming traffic has entered it as switch one was able to accommodate all incoming traffic, it may be clearly noted how the system is able to save energy in an efficient manner thanks to the proposed algorithm.

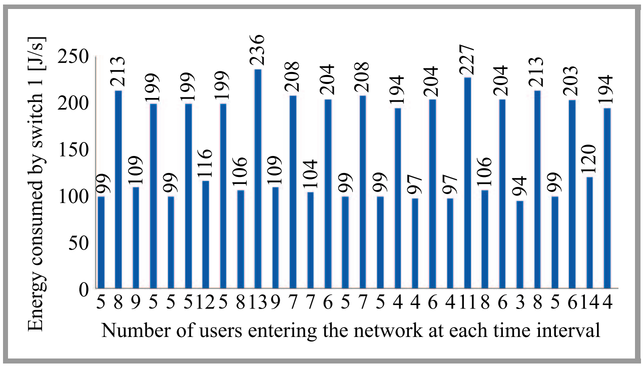


Fig. 7. Amount of energy consumed by the users entering the network after being sorted.

4.3. Comparison with Traditional Data Center Networks

Traditional data centers, such as [8], are mainly concerned with processing data in an efficient manner to guarantee good QoS and QoE levels experienced by the end user, while neglecting the extensive amounts of power and energy being dissipated. Another proposal that suggests a dynamic resource allocation approach but neglects QoS and QoE rates is [7]. However, the algorithm proposed in this paper aims to assure that both problems arising in [7] and [8] are resolved at once. QoS and QoE are dependent upon packet loss rate, delays and high latencies. The proposed algorithm proves that quality may be maintained (unlike in [7]) through avoiding overload and dynamically allocating traffic while granting it with the resources required to accommodate the workload. The problem with [8] is that the network devices remain active at all times. To prove the distinct nature of the proposed algorithm, the following values retrieved from Fig. 8 are compared against a sys-

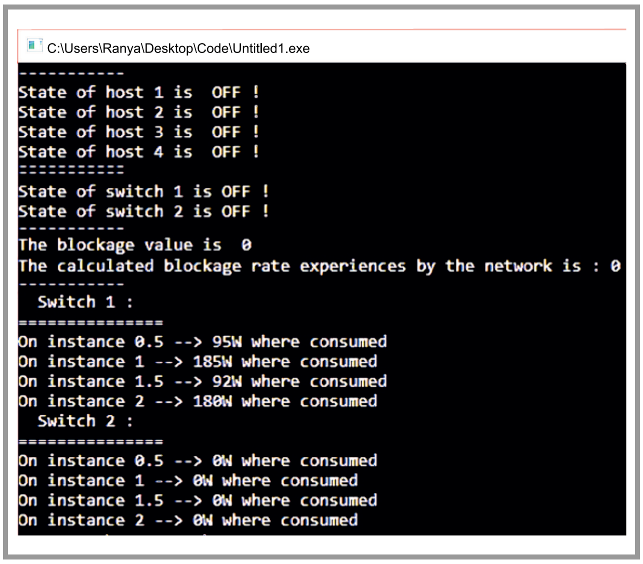


Fig. 8. Statistics concerning the power consumed at each instance after all clients have been served and directed out of the network switches (network components are turned back off).

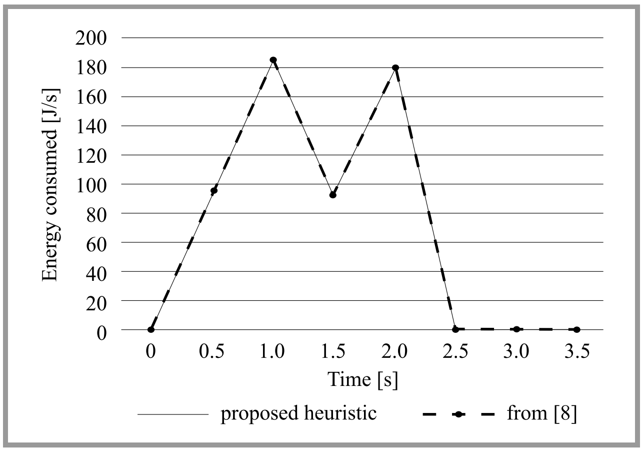


Fig. 9. Energy consumed by switch one by the proposed algorithm and the algorithm described in [8].

tem that relies on principles similar to those of [8], meaning that it employs an optimized resource allocation plan to avoid overload but fails to conserve energy. Since the switches are turned off when they are not in use, switch two dissipates zero watts, while switch one handles all of the incoming traffic. A simple graph representation may be seen in Figs. 9 and 10, where a comparison between [8] and the proposed algorithm has been shown.

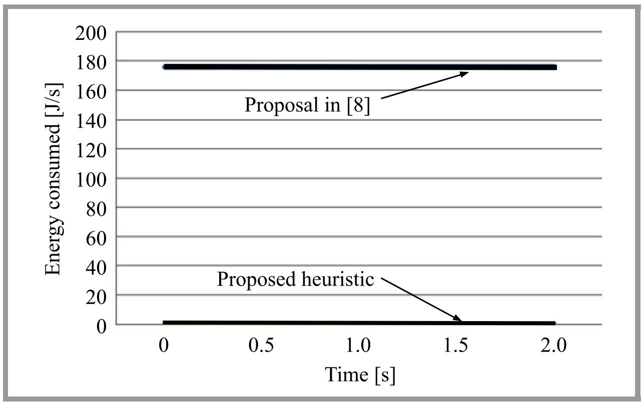


Fig. 10. Difference in energy consumption between the proposed algorithm and the solution proposed in [8], for switch two.

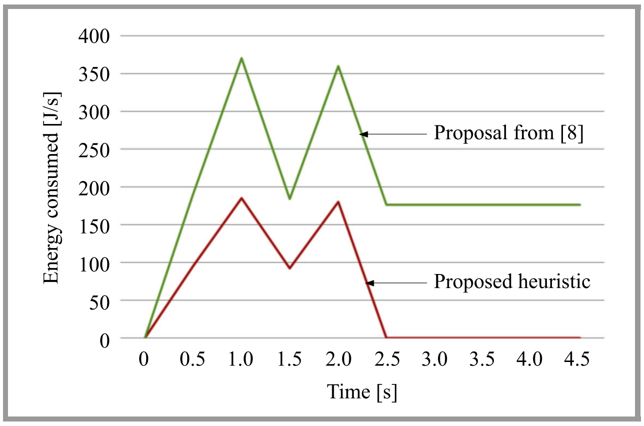


Fig. 11. Total energy consumed by the proposed algorithm and the algorithm described in [8].

Assuming that both solutions proposed in this paper and in [8] are tested on the same network devices chosen for simulation purposes, the following results may be observed. Both proposals are concerned with avoiding overload and routing traffic within the network in a studied manner. As shown in Fig. 9, energy consumption is hypothetically similar in both scenarios. However, in Fig. 10, with observations focused on switch 2, [8] consumes more power, as illustrated in Fig. 11, since it is always on.

5. Conclusion and Future Work

The proposed heuristic algorithm has been proved to attain the expected outcomes discussed in the earlier sections, with energy consumption confirmed to decrease drastically and with quality maintained due to the fact that no packet loss or delays are experienced within the network. Future work may be concerned with incorporation of a mechanism that is able to save the data and reallocate them in the case in which a physical host from the data center suffers a failure.

References

[1] C. Mastroianni, M. Meo, and G. Papuzzo, "Analysis of a self-organizing algorithm for energy saving in data centers", in *Proc. 2013 IEEE Int. Symp. on Parallel Distrib. Process., Worksh. and Phd Forum*, Cambridge, MA, USA, 2013, pp. 907–914 (doi: 10.1109/IPDPSW.2013.184).

[2] D. A. Alboaneen, B. Pranggono, and H. Tianfield, "Energy-aware virtual machine consolidation for cloud data centers", in *Proc. 2014 IEEE/ACM 7th Int. Conf. on Util. and Cloud Comput.*, London, UK., 2014, pp. 1010–1015 (doi: 10.1109/UCC.2014.166).

[3] I. Widjaja, A. Walid, Y. Luo, Y. Xu, and H. J. Chao, "Small versus large: Switch sizing in topology design of energy-efficient data centers", in *Proc. 2013 IEEE/ACM 21st Int. Symp. on Qual. of Serv. IWQoS 2013*, Montreal, QC, Canada, 2013, pp. 51–56 (doi: 10.1109/IWQoS.2013.6550264).

[4] J. Perel *et al.*, "All-optical packet/circuit switching-based data center network for enhanced scalability, latency, and throughput", *IEEE Network*, vol. 27, no. 6, pp. 14–22, 2013 (doi: 10.1109/MNET.2013.6678922).

[5] M. Seymour, "Is energy efficiency enough? Filling the engineering gap in data center design and operation", in *Proc. 15th IEEE Intersoc. Conf. on Thermal and Thermomech. Phenom. in Electron. Syst. ITherm 2016*, Las Vegas, NV, USA, 2016, pp. 702–709 (doi: 10.1109/ITHERM.2016.7517616).

[6] A. Amokrane, M. F. Zhani, R. Langar, R. Boutaba, and G. Pujolle, "Greenhead: Virtual data center embedding across distributed infrastructures", *IEEE Trans. on Cloud Comput.*, vol. 1, no. 1, pp. 36–49, 2013 (doi: 10.1109/TCC.2013.5).

[7] B. Yu, Y. Han, X. Wen, X. Chen, and Z. Xu, "An energy-aware algorithm for optimizing resource allocation in software defined network", in *Proc. IEEE Global Commun. Conf. GLOBECOM 2016*, Washington, DC, USA, 2016, pp. 1–7 (doi: 10.1109/GLOCOM.2016.7841589).

[8] E. Grigoriou, A. A. Barakabitze, L. Atzori, L. Sun, and V. Pilloni, "An SDN-approach for QoE management of multimedia services using resource allocation", in *Proc. IEEE Int. Conf. on Commun. ICC 2017*, Paris, France, 2017, pp. 1–7 (doi:10.1109/ICC.2017.7997261).

[9] B. Pavithra and R. Ranjana, "Energy efficient resource provisioning with dynamic VM placement using energy aware load balancer in cloud", in *Proc. Int. Conf. on Inform. Commun. and Embedded Syst. ICICES 2016*, Chennai, India, 2016, pp. 1–6 (doi: 10.1109/ICICES.2016.7518919).

[10] S. Subbiah and V. Perumal, "Energy-aware network resource allocation in SDN", in *Proc. Int. Conf. on Wirel. Commun., Sig. Process. and Netw. WiSPNET 2016*, Chennai, India, 2016, pp. 2071–2075 (doi: 10.1109/WiSPNET.2016.7566506).

[11] X. Wen, Y. Han, H. Yuan, X. Zhou, and Z. Xu, "An efficient resource embedding algorithm in software defined virtualized data center", in *Proc. IEEE Global Commun. Conf. GLOBECOM 2015*, San Diego, CA, USA, 2015, pp. 1–7 (doi: 10.1109/GLOCOM.2015.7417556).

[12] Q. Zhang, Q. Zhu, M. F. Zhani, and R. Boutaba, "Dynamic service placement in geographically distributed clouds", in *Proc. IEEE 32nd Int. Conf. on Distrib. Comput. Syst.*, Macau, China, 2012, pp. 526–535 (doi: 10.1109/(ICDCS.2012.74).

[13] S. Padma, K. Vijayalakshmi, and G. Sangameshwaran, "Power generation using hybrid renewable energy resources for domestic applications", in *Proc. Int. Conf. on Wirel. Commun., Sig. Process. and Netw. WiSPNET 2016*, Chennai, India, 2016, pp. 1993–1998 (doi: 10.1109/WiSPNET.2016.7566491).

[14] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for internet-scale systems", *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 123–134, 2009 (doi: 10.1145/1594977.1592584).

[15] M. Rahnamay-Naeini, S. S. Baidya, E. Siavashi, and N. Ghani, "A traffic and resource-aware energy-saving mechanism in software defined networks", in *Proc. Int. Conf. on Comput., Netw. and Commun. ICNC 2016*, Kauai, HI, USA, 2016, pp. 1–5 (doi: 10.1109/ICCNC.2016.7440553).

[16] T. Yang, Y. C. Lee, and A. Y. Zomaya, "Energy-efficient data center networks planning with virtual machine placement and traffic configuration", in *Proc. IEEE 6th Int. Conf. on Cloud Comput. Technol. and Sci.*, Singapore, 2014, pp. 284–291 (doi: 10.1109/CloudCom.2014.135).

[17] K. Zhang, T. Wu, S. Chen, L. Cai, and C. Peng, "A new energy efficient VM scheduling algorithm for cloud computing based on dynamic programming", in *Proc. IEEE 4th Int. Conf. on Cyber Secur. and Cloud Comput. CSCloud 2017*, New York, NY, USA, 2017, pp. 249–254 (doi: 10.1109/CSCloud.2017.46).

[18] M. K. Patterson, "The effect of data center temperature on energy efficiency", in *Proc. 11th Intersoc. Conf. on Thermal and Thermomech. Phenom. in Electron. Syst.*, Orlando, FL, USA, 2008, pp. 1167–1174 (doi: 10.1109/ITHERM.2008.4544393).

[19] N. T. Hieu, M. D. Francesco, and A. Yl-Jski, "Virtual machine consolidation with usage prediction for energy-efficient cloud data centers", in *Proc. IEEE 8th Int. Conf. on Cloud Comput.*, New York, NY, USA, 2015, pp. 750–757 (doi: 10.1109/CLOUD.2015.104).

[20] A. Markiewicz, P. N. Tran, and A. Timm-Giel, "Energy consumption optimization for software defined networks considering dynamic traffic", in *Proc. IEEE 3rd Int. Conf. on Cloud Netw. CloudNet 2014*, Luxembourg, 2014, pp. 155–160 (doi: 10.1109/CloudNet.2014.6968985).

[21] D. Henni, Y. Hadjaj-Aoul, and A. Ghomari, "Probe-SDN: A smart monitoring framework for SDN-based networks", in *Proc. Global Inform. Infrastruc. and Netw. Symp. GIIS 2016*, Porto, Portugal, 2016, pp. 1–6 (doi: 10.1109/GIIS.2016.7814940).

[22] C. Zhang, X. Huang, G. Ma, and X. Han, "A dynamic scheduling algorithm for bandwidth reservation requests in software-defined networks", in *Proc. 10th Int. Conf. on Informa., Commun. and Sig. Process. ICICS 2015*, Singapore, 2015, pp. 1–5 (doi: 10.1109/ICICS.2015.7459856).

[23] K. Liu, Y. Cao, Y. Liu, G. Xie, and C. Wu, "A novel min-cost QoS routing algorithm for SDN-based wireless mesh network", in *Proc. 2nd IEEE Int. Conf. on Comp. and Commun. ICCC 2016*, Chengdu, China, 2016, pp. 1998–2003 (doi: 10.1109/CompComm.2016.7925051).

[24] S. Tomovic, I. Radusinovic, and N. Prasad, "Performance comparison of QoS routing algorithms applicable to large-scale SDN networks", in *Proc. IEEE EUROCON 2015 – Int. Conf. on Comp. as a Tool EUROCON 2015*, Salamanca, Spain, 2015, pp. 1–6 (doi: 10.1109/EUROCON.2015.7313698).

[25] Y. Guo, Z. Wang, X. Yin, X. Shi, and J. Wu, "Optimize routing in hybrid SDN network with changing traffic", in *Proc. 26th Int. Conf. on Comp. Commun. and Netw. ICCCN 2017*, Vancouver, BC, Canada, 2017, pp. 1–8 (doi: 10.1109/ICCCN.2017.8038397).

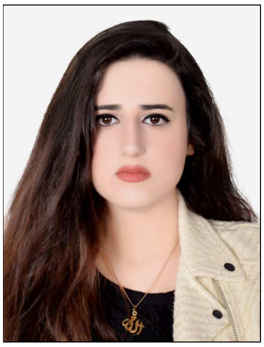
[26] Y. Han, J. Li, J.-Y. Chung, J.-H. Yoo, and J. W. Hong, "SAVE: Energy-aware virtual data center embedding and traffic engineering using SDN", in *Proc. of the 2015 1st IEEE Conf. on Netw. Softwariz. NetSoft 2015*, London, UK, 2015, pp. 1–9 (doi: 10.1109/NETSOFT.2015.7116142).

[27] A. Ishimori, F. Farias, E. Cerqueira, and A. Abelm, "Control of multiple packet schedulers for improving QoS on OpenFlow/SDN networking", in *Proc. 2nd Eur. Worksh. on Software Defined Netw.*, Berlin, Germany, 2013, pp. 81–86 (doi: 10.1109/EWSDN.2013.20).

[28] J. Pang, G. Xu, and X. Fu, "SDN-based data center networking with collaboration of multipath TCP and segment routing", *IEEE Access*, vol. 5, pp. 9764–9773, 2017 (doi: 10.1109/ACCESS.2017.2700867).

[29] A. Bentaleb, A. C. Begen, R. Zimmermann, and S. Harous, "SDNHAS: An SDN-enabled architecture to optimize QoE in HTTP adaptive streaming", *IEEE Trans. on Multimed.*, vol. 19, no. 10, pp. 2136–2151, 2017 (doi: 10.1109/TMM.2017.2733344).

[30] P. Desmond, "5 contributors to data center energy inefficiency", 2011 [Online]. Available: https://blog.schneider-electric.com/datacenter/2011/09/06/5-contributors-to-data-center-energy-inefficiency/



Ranya Al-Musawi is currently a student at the University of Wollongong (Australia), perusing her final year to receive a B.Sc. degree in Computer Engineering. She was a representative of engineering students at the representative council for the year of 2017 at Wollongong. Her main interests are in automation, embedded systems and telecommunication networks.
E-mail: rmmam836@uowmail.edu.au
Faculty of Engineering and Information Sciences
University of Wollongong Dubai
Dubai Knowledge Park
P.O. Box 20183, Dubai, UAE

Obada Al-Khatib received his B.Sc. degree in Electrical Engineering (Honors) from Qatar University, Doha, Qatar, in 2006, M.Eng. degree in Communication and Computer Technology (Honors) from the National University of Malaysia, Bangi, Malaysia, in 2010, and Ph.D. degree in electrical and information engineering from the University of Sydney, Sydney, Australia, in 2015. From 2006 to 2009, he was an Electrical Engineer with the Consolidated Contractors International Company, Qatar. In 2015, he joined the Centre for IoT and Telecommunications at the University of Sydney, as a Research Associate. Since 2016, he has been with the Faculty of Engineering and Information Sciences, University of Wollongong in Dubai, UAE, as an Assistant Professor. His current research interests are in the areas of smart grid communication, wireless resource allocation and management, cooperative communications and wireless network virtualization.
https://orcid.org/0000-0001-9473-2365
E-mail: obadaalkhatib@uowdubai.ac.ae
Faculty of Engineering and Information Sciences
University of Wollongong Dubai
Dubai Knowledge Park
P.O. Box 20183, Dubai, UAE