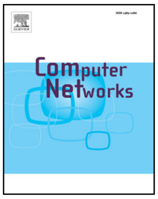
Computer Networks 113 (2017) 188–204

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



Review article

A survey on energy efficiency in software defined networks

Mehmet Fatih Tuysuz[a,*], Zekiye Kubra Ankarali[b], Didem Gözüpek[c]

[a] Department of Computer Engineering, Harran University, Sanliurfa, Turkey
[b] Department of Computer Engineering, Istanbul Şehir University, Istanbul, Turkey
[c] Department of Computer Engineering, Gebze Technical University, Kocaeli, Turkey

ARTICLE INFO

Article history:
Received 6 May 2016
Revised 2 December 2016
Accepted 19 December 2016
Available online 21 December 2016

Keywords:
Software-Defined Networking (SDN)
Energy efficiency
OpenFlow

ABSTRACT

Wide deployment and dense usage of computer networks may cause excessive energy consumption due to the increase in probability of network congestion, frame collisions and packet dropping rates resulting from late-received frames. Besides, based on the dramatic increase in network complexity with wireless, mobile and split tunnel connections, weak visibility into network flows and high cost of some of public and private network services, current networks can also be implied as inefficient in terms of both performance and economy. Software-Defined Networking (SDN) is a novel networking architecture, which provides a directly programmable and (logically) centralized network control, separates network control from forwarding, and enables programmable network components. SDN can have a significant role in reducing the aforementioned excessive energy consumption caused by data centers, network components, and end hosts. In this paper,[1] we examine the principles, benefits, and drawbacks of up-to-date SDN approaches that focus on energy efficiency. We also provide a brief comparison of possible energy gain ratios of existing approaches, discussion on open issues and a guideline for future research.

1. Introduction

In current communication networks, setting up a network requires multiple software and hardware-based networking technologies, such as protocols, switches and routers. Therefore, wired and wireless communication networks, such as personal, local or wide area networks, may face with a fundamental complexity problem in case they consist of a high number of connected devices having different characteristics and requirements. Additionally, devices in traditional data networks need to be configured individually, task updates of devices are time consuming, and the evolution of the hardware functionality is controlled only by the provider. Therefore, implementing a new network policy may require configuring many devices. Furthermore, traditional data networks have devices that are designed to perform specific tasks, such as switches and routers. Hence, they have a static architecture and result in slow-evolving network functionality. However, there is a high amount of increase in the use of mobile devices, server virtualization, and cloud services. All of these activities require dynamic network structure.

In the past decade, network traffic pattern has also changed. Applications now require accessing different servers and databases since users want to reach applications, infrastructure and all other IT resources. Mobile devices are currently used to access the corporate networks; hence, IT staff becomes obliged to protect their data. In addition, data centers are growing. Although intercommunication between these centers has been evolving, the ever-growing communication demand is mostly addressed by increasing the bandwidth and setting up new network cables. As a result, handling with massive datasets is costly with traditional networks. Since network virtualization is now widely used, physical elements can contain more than one virtual network structure. However, the reconfiguration process is not agile, easy, and quick. Besides, users' network functionality is limited; it depends on the vendors and the hardware providers. Consequently, all of the above-mentioned facts constitute novel challenges for network owners.

As the network size increases, its complexity grows exponentially. In this regard, current network infrastructure requires more flexible and dynamic network operations, programmability and easily modified network devices. Therefore, the network must be programmed according to its changing requirements. Currently, the key paradigm to achieve this behavior is Software-Defined Networking (SDN). SDN simply provides a (logically) centralized control point in the network. In an SDN architecture, control plane - the plane that understands the network and decides the flow paths, and data plane - the plane responsible for the transmis-

---

* Corresponding author.

E-mail addresses: ftuysuz@harran.edu.tr (M.F. Tuysuz), zekiyetekin@std.sehir.edu.tr (Z.K. Ankarali), didem.gozupek@gtu.edu.tr (D. Gözüpek).

[1] This work is supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under grant no. 114E245.

http://dx.doi.org/10.1016/j.comnet.2016.12.012
1389-1286/

sion of packets, are separated. Separation of control layer from the data layer enables programmability, increases functionality, 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.

In SDN, network intelligence and state are logically centralized, and the underlying network infrastructure is abstracted from the applications [1]. Network devices merely need to accept instructions from the SDN controller instead of understanding all protocols. Hence, any network element can be changed instantaneously. SDN enables the network to meet instantly emerging needs of business and institutions and help them to customize the network. SDN can also be beneficial in cases like directing devices to safe flows, improving network performance, preparing network bandwidth for scheduled data transfers, keeping network slices apart with controller to keep away from research traffic, transition between multiple data centers and single data center, etc.

In SDN, communication between infrastructure and control layer is provided by the OpenFlow protocol [2], which was released by Stanford University and California University in 2011 and currently managed by the Open Networking Foundation (ONF). Although the terms SDN and OpenFlow are closely related, they are not interchangeable. While SDN is an emerging network architecture that enables networks to be programmable with a high degree of automation, OpenFlow is a protocol that configures network switches in order to provide the communication between the SDN controller and the devices. OpenFlow allows SDN controllers to decide the path of network packets through the network of switches. However, controllers should also include orchestration tools to dynamically and automatically respond to the needs of the network. Moreover, controllers should be able to communicate with other controllers regardless of their own proprietary interfaces and scripting languages.

In a nutshell, SDN technology aims to address the problems of the traditional networks and it is currently supported by the network community. It can provide centralized control of the network even when the network consists of multi-vendor elements. It reduces the complexity through network automation. Moreover, via SDN, IT staff can easily adapt the network to the rapidly changing needs and requirements. In addition, SDN enables more security since it provides a global control point over the network. SDN controller can easily regulate any policy; therefore, IT staff can control even the smallest elements of the network, such as devices, users, and applications.

Another inefficiency of the current networking technology is the high amount of energy it consumes. Current networks are inefficient both environmentally and economically (i.e. CO_2 emission, operational costs, etc.) and hence they should be reconfigured. In the past, research scope of the Information and Communication Technology (ICT) was mainly based on performance and cost. The research community put insufficient effort to the energy consumed by ICTs and their impact on the environment. Current trends, such as increasing electricity costs, reserve limitations, and increasing emissions of carbon dioxide (CO_2) are shifting the focus of ICT towards energy-efficient and well-performed solutions. Even though governments and companies are now aware of the massive carbon emissions and energy requirements, it is obvious that carbon emissions and the amount of energy consumption will continue to increase [3]. As stated by the SMART 2020 study [4], ICT-based CO_2 emissions are rising at a rate of 6% per year. With such a growth ratio, it is expected that CO_2 emissions caused by ICTs will reach 12% of worldwide emissions by 2020. Communication networks designed according to this energy efficiency criteria are called green networks [5]. In this context, SDN procedure can have a significant role in reducing the energy consumption by decoupling network functionalities and utilizing some of the local and network-related parameters. In other words, the global network knowledge and centralized decision-making mechanism of SDN make it a proper environment for realizing green networks.

There have been many reviews focusing on SDN paradigm in the literature, such as the works presented in [6–9]. However, extensive analysis of SDN in terms of energy efficiency has hitherto received little attention. Towards closing this gap, we first present the importance and possible ways of energy saving in SDN, and then examine the principles, benefits and drawbacks of up-to-date SDN approaches that take energy efficiency into account. We also present a brief comparison of possible energy gain ratios of existing approaches, discussion on open issues, and a guideline for future research.

The rest of the paper is organized as follows. Section 2 presents background information on the definition, classification, and procedure of the SDN paradigm. Section 3 presents the importance of energy saving in SDN. Section 4 examines the principles, benefits and drawbacks of up-to-date SDN approaches in terms of energy efficiency. Section 5 compares the existing approaches from the view of energy gain, and finally Section 6 reports the existing issues, challenges and possible future research directions.

2. Background

Although the term SDN has been coined in the last decade, the concept behind the SDN has been evolving since 1990s, driven by the need to offer user-controlled management of forwarding in network nodes [10]. In 1990s, researchers proposed programmable networks as a solution for current issues of networking and new ideas about programmable networks, such as OPENSIG, Active Networking, and DCAN, started to come out. OPENSIG [11] proposed a method that enables the control of the network hardware with a programmable network interface. Active Networking [12,13] suggested a programmable network infrastructure, utilizing user-programmable switches and capsules, which were program fragments (carried in user messages) interpreted and executed by routers. DCAN [14], which is similar to SDN, aimed to decouple control and management layers from network devices and inserted them into other entities.

In 2000s, several methods similar to SDN have also been proposed. In this context, 4D Project [15] aimed to design a network with three layers: a decision plane, a discovery plane (which informs the decision plane about the network to control the data plane) and a control plane. NETCONF [16] is another example for SDN-like approaches. It has a layered structure and a mechanism that can install, change or delete the configurations of the network devices according to the needs of the network, and provide a secure environment while transporting the configuration environment [17]. In addition, Ethane [18] was the immature version of OpenFlow. The project managed policies and security of the network using a centralized controller.

Fig. 1 shows a comparison of the architectures of Traditional and Software-Defined Networks. In traditional networks, control and data planes are united in a single network device. However, to enable programmability, increase functionality, and enable remote management between infrastructures, SDN concentrates on four key features: (i) Decoupling of the control and the data plane, (ii) Detailed observation of the network with a centralized controller, (iii) Open interfaces between devices in control plane (controllers) and those in data plane, and (iv) Programmability of the network by external applications. In addition, as shown in Fig. 2, there are three layers in an SDN; (i) Infrastructure layer (deploys

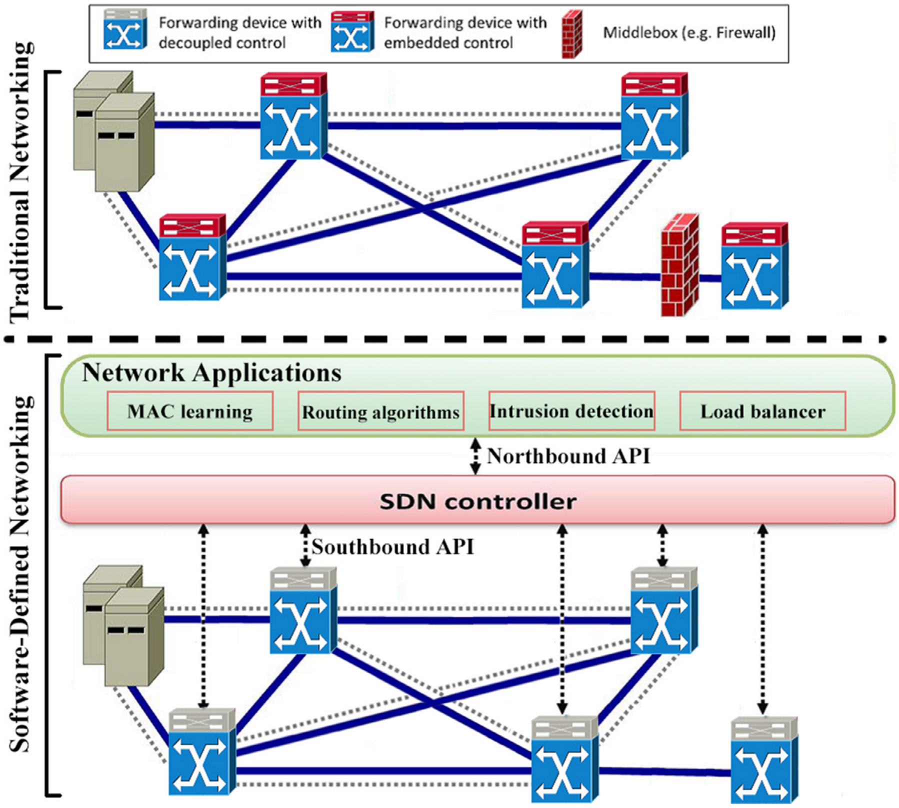


Fig. 1. Traditional Networking versus SDN.

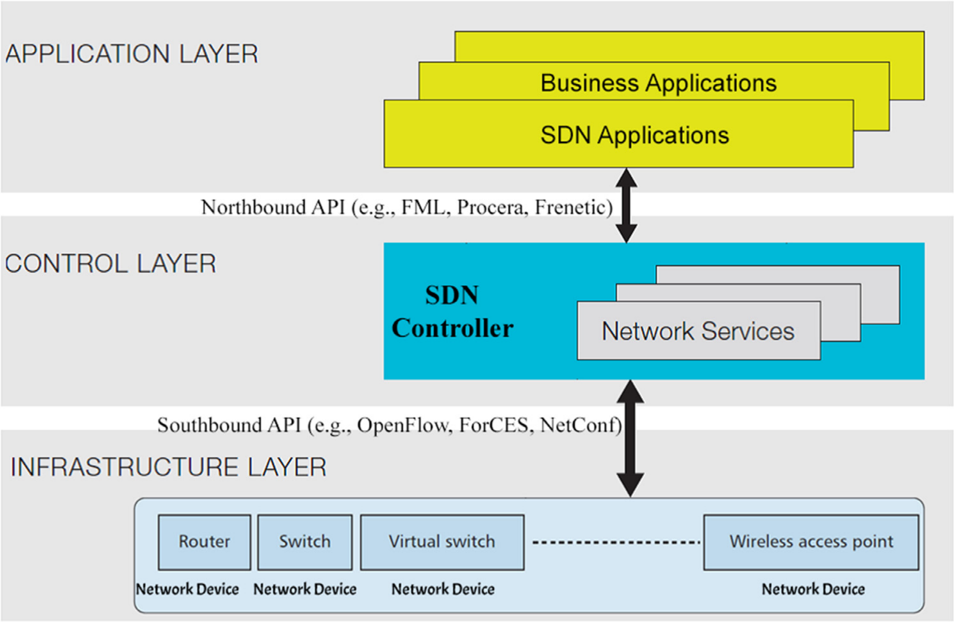


Fig. 2. Simplified view of an SDN architecture.

OpenFlow standards and physical elements), (ii) Control layer (provides control plane over the entire network), and Application layer (transfers instructions or requirements to the controller). Infrastructure layer, also known as the data plane, consists of network devices that support a Southbound API, which allows data switching and forwarding. Southbound API, i.e. OpenFlow protocol, is one of the standardized protocols of ONF and provides the communication between infrastructure and control layers in addition to regulating the controller-switch interaction. Northbound API is the interface that provides communication between control and application layers. Using Northbound API, applications interact with controllers and explore the network state in order to manipulate the services of the network, such as governing the traffic, path computation, and security or any other orchestration operations [7,8]. In control layer, SDN controller(s) centralizes the control functionality of the network. Controllers are placed between forwarding devices and applications, and manage the flow of data through switches. Finally, the application layer is where business and SDN applications, such as network visualization, security applications and routing algorithms, are deployed.

3. Why do we need to save energy

Worldwide energy consumption of ICT equipment exhibits the urgent need for energy efficiency in networking. In [19], authors evaluate the impact of different sectors of ICT on energy consumption and CO_2 emissions, comparing an early report (Gartner Group Report, 2007 [20]) as well as more recent reports (EINS European Project, 2013 [21,22]). Authors state that although the initial Gartner Report has an alarmist perspective, and the potentially explosive growth of energy consumption by ICT has not been substantiated, worldwide ICT energy consumption and CO_2 emissions still continue to grow. Authors also mention that the improving energy efficiency of ICT equipment leads to a slower growth of these metrics than the increase of the worldwide usage of ICT. In this context, ICT can also contribute to a reduction in energy consumption and CO_2 emissions in other sectors, utilizing efficient networking and communication strategies.

Despite many differences between mobile and fixed networks in terms of their design, implementation, maintenance, monitoring, and operation, there is a trade-off between energy consumption and performance in both networks. This trade-off, the level of which varies in mobile and fixed environments, should be taken into account during system design. For instance, energy-efficient operation may necessitate the sending of additional control messages, such as sleep messages from the SDN controller to the switches, and this additional control traffic decreases the bandwidth efficiency. In mobile wireless networks, energy-efficient operation implies reduced power consumption of network devices, which in turn decreases the transmission data rate since data rate and transmission power are related to each other due to Shannon's formula. The sleeping of nodes in both wired and wireless networks may increase the transmission delay not only because of the reduction in data rate but because in some situations the packets need to be queued while waiting for the device (or other devices) to wake up.

In communication networks, energy efficiency can be achieved via two fundamental choices: (i) data rate can be increased, or in other words, packetization delay which is the amount of time required to push the packet's bits into the wired/wireless medium, can be reduced and then terminals can be put into sleep mode for longer time to consume less power within transmissions, or (ii) data rate can be reduced and hence lower power consumption can be achieved at the expense of longer data transmission times [23]. In contrast, in a network managed by SDN, various parts of the SDN structure can be configured dynamically to reduce power consumption. One way is to set the flow according to the traffic of the network and put the unused devices in the network into sleep mode. When there is low traffic load, instead of the whole device, certain ports can be put into sleep mode. Another method is to optimize/reduce the memory size used by forwarding switches as flow tables are stored in Ternary Content Addressable Memory (TCAM), which is expensive and power-hungry. However, reducing TCAM too much may cause frequent flow entry replacements when new flows are to be installed. Hence, more accesses to TCAM may consume more energy. Besides, SDN controller can place new rules about energy efficiency into switches under some constraints, such as number of switches and routing policies. Controller can aim to minimize the energy consumption while reconfiguring new rules according to the system. Virtualization of servers is another way to reduce energy consumption. This way, multiple virtual machines can work on the same physical server. Thus, instead of using many servers inefficiently, constructing virtual machines can save additional energy [24].

4. Energy efficient SDN approaches

Green Networking has been widely studied and numerous solutions have been proposed in the literature. Most of these works can also be adapted to the SDN concept. For instance, authors in [25] present an analytical model that compares the trade-offs between network performance and energy saving. In order to create their adaptive model, they use Adaptive Rate (AR) and Low Power Idle (LPI) transmission techniques and performance and power levels of Advanced Configuration and Power Interface (ACPI) standard. According to their work, packets are transferred in longer times with the AR. However, LPI also results in delay, as it requires the system to sleep or wake-up frequently. LPI also increases burstiness of the traffic due to the same frequent sleep/wake-up initiations. In contrast, AR smooths the traffic. Considering these trade-offs, authors create their analytical model that minimizes the power consumption subject to latency and loss probability constraints. They compare their model with a router that includes AR and LPI capabilities for fixed power and performance levels and show that their optimization model allows energy saving roughly about 16–17% in comparison to the fixed configuration scenario. As mentioned earlier, these types of green networking solutions can also be adapted to the SDN and additional energy gain can be achieved. However, throughout this section, we examine only the works related to energy efficiency in SDN.

SDN, network virtualization (NV), and network functions virtualization (NFV) are three terminologies that are often erroneously used interchangeably. SDN has three distinctive features: (i) Separation of data plane and control plane, (ii) (Logically) centralized control unit, (iii) Network programmability. NV, on the other hand, creates logical segments in an existing network by logically dividing the network at the flow level irrespective of the programmability of the network. It creates a tunnel in the existing network rather than physically connecting two domains, and thereby making the life of network administrators easier by eliminating the burden of changing the physical network. While NV is being widely used for decades, NFV is a relatively new concept referring to putting a service/network function on a tunnel, i.e., virtualizing network functions and migrating them to generic servers. This way, NFV aims to reduce deployment costs for services such as load balancing and firewall setup. The major difference of NV and NFV from SDN is that NV and NFV add virtual tunnels and functions to the physical network, whereas SDN changes the physical network. In other words, SDN is a new concept to provision and manage the network, while NV and NFV are mechanisms to provide features such as resource sharing and new functions without having to change the infrastructure of the network. NV and NFV can exist on existing networks; however, SDN requires the construction of a new network because it is based on a completely different network management mechanism.

With the enormous increase in the number of mobile devices, network operators need to provide new services and increase capabilities in order to meet the ever-growing demands. In this regard, the innovation speed has to increase while the costs decrease. Current network devices are highly specialized and yet it is still hard to catch the speed of change in the environment. The cost of physical infrastructures of mobile networking systems is high as well. Sharing of physical resources may reduce this cost. However, it is not preferred among mobile network operators due to the lack of regulations. For instance, sharing physical resources is forbidden since regulators think it will damage the competition between operators. To increase efficiency (e.g. energy, performance and resource efficiency) and decrease the cost in networking, authors in

[26] propose a solution to the aforementioned issues, by integrating the SDN and network virtualization. Although network virtualization is an effective technology in use since decades, its integration with SDN is a novel concept bringing numerous advantages. In this scope, SDN provides programmable and simpler network devices that reduce the cost and enables more efficient networking. Network virtualization provides different operators to use active sharing, which is concerned with using the same infrastructure but splitting it into slices. This way, it will provide different services and make the competition among operators possible. To see the impact of the integration of SDN and network virtualization on cost and efficiency, authors compare three different network scenarios; (i) classical (where specialized network elements are in use), (ii) SDN-based (where SDN-enabled network components are in use) and (iii) sharing (where network virtualization and network sharing between different operators using FlowVisor controllers are in use). In this context, authors demonstrate that the total number of networking devices, which is related to the expected amount of total power consumption, will decrease in an SDN-based scenario. Besides, the cost of repairing and testing will also reduce since networks will consist of lower number of devices.

Layer-based architecture and the separation of control and data plane in SDN enable dynamic network configuration, which may also lead to numerous energy-aware new strategies. In this context, to evaluate existing energy-aware SDN approaches in detail and discuss their strengths and weaknesses, this section is broken down into subsections based on the approach/metrics used in the studies.

4.1. Flow-controlled energy-efficient SDN approaches

Green Abstraction Layer (GAL) proposed in [27] is one of the traffic-aware energy-efficient SDN approaches. GAL enables internal communication among network devices to exchange power-related data, such as power consumption of a particular switch, a state, or a device. The data is then transferred to the controller and hence, the controller applies real-time control strategies among networking devices. Energy Aware States (EASs) in GAL are defined by logical entities of the network (i.e. ports, switches, actions on traffic flow, etc.), from the point of energy consumption, such as energy consumption at a particular state or at maximum supported data rate. Energy Optimizer module uses information about EASs and calculates the best state for obtaining minimum energy. GAL consists of three phases: discovery, provisioning and monitoring. Discovery phase enables the controller to obtain information about power consumption of each entity. Provisioning phase sets the state of the network nodes according to the optimal energy state of the network. After re-arranging the network state, monitoring phase sends the information about the current state of each entity. Finally, Local Control Policies (LCPs) optimize the choice between energy saving and network performance according to the traffic.

To provide energy efficiency and manage the available resources in SDN, authors in [28] propose a network management system with the help of Green Abstraction Layer (GAL) [27]. The system takes topology, resources, and traffic demand as input and calculates flows and energy states of each element that will take part in flows. All possible paths for a given flow are calculated and a list of nodes is prepared together with nodes' capacities and their number of occurrences. To show the effectiveness of the proposed scheme, authors implement an emulation environment and investigate the total network level consumption of a day-night traffic profile; composed by six macro-flows, where 24 hours of a day have been divided into 300 time slots. Results show a good accuracy level, as the traffic decreases, the links enter lower power states and the traffic processing and forwarding continue regularly.

The work in [29] also aims to optimize energy consumption in SDN. The authors first provide a Mixed Integer Linear Programming (MILP) formulation and then propose a Strategic Greedy Heuristic algorithm. In their proposed algorithm, access nodes of the traffic flow inform the rest of the nodes about their traffic load and then the controller sets an optimized traffic flow and turns off as many devices as possible. Throughout the paper, authors set three constraints; (i) Capacity constraint makes sure that links are efficiently used and there are no underused or overloaded links, (ii) Traffic demand constraint ensures that all traffic demands are allocated to related paths with enough capacities, and (iii) Node-link relation constraint shuts down the node when all of the links connected to it are turned off. In order to evaluate the efficiency of the proposed algorithm, authors constitute two networks, a campus network and a mesh network, using three levels of traffic demands: low (night), mid (average daytime), and high (yearly peak). The results show that the proposed method can save high amount of energy (up to 45%), especially at nighttime.

In order to optimize energy efficiency in SDN, authors in [30] present the GreenSDN approach, which integrates three different protocols that operate at different layers of the network: Adaptive Link Rate (ALR) [31], which is a chip-level protocol, Synchronized Coalescing (SC) [32], which is active at node-level, and Sustainability-oriented Network Management System (SustNMS) [33], which operates at network level. ALR works on links and changes data rates according to the traffic load of the network, whereas SC protocol works as LPI. However, while LPI works on individual parts of the network devices, SC can put a whole device into the idle mode. SustNMS controls the network and balances the trade-off between QoS and energy efficiency to quickly respond to changing traffic patterns. The work makes use of Mininet as the emulation tool and POX as the controller. In addition, since it is important to control the QoS, check the efficiency of the traffic engineering, and compute the expected amount of energy consumption, authors also exploit OpenFlow protocol tools for network monitoring.

The implementation of GreenSDN architecture follows mainly three steps. First, the monitoring part is implemented. This part utilizes the information about the traffic beginning from the edge-nodes and continuing through used paths. Second, power states are implemented. GreenSDN defines two types of power profiles: a load proportional profile and a linear profile. While the former consumes energy proportional with the workload, the latter works with a constant energy. Finally, green energy efficiency protocols are implemented. ALR protocol increases the link rate when it is turned on. SC compares the total number of packets per second with a previously determined buffer. If it is higher than the buffer, SC is turned off to meet the workload; otherwise, packets are put together to provide maximum energy saving. As mentioned before, SustNMS is responsible from the whole network. It tries to provide energy efficient network traffic by taking into account all the traffic flows, used paths, and also the switches that will be put into sleep mode.

In [34] authors propose Exclusive Routing (EXR) to improve fair-sharing routing (FSR), which is a common routing method for fair allocation. FSR selects a subset of links and uniformly spreads flows across the links without any delay. However, this behavior indicates that all links work with less than full capacity. Throughout the paper, authors demonstrate that FSR approach uses most of the switches with utilization lower than 55% of its full capacity. In this context, authors claim that efficient use of already activated links, i.e. full use of their capacity, and turning off more switches will decrease energy usage even more. Thus, the main idea is to eliminate low utilization of links and switches. EXR has two kinds of flows: *active* and *suspended*. The controller manages the whole process, notifies the switches about letting or blocking a flow. When a new flow arrives, if there are any idle paths without any flow on it, the flow is set as active and it is sent. If there is no idle path but there is a path with suspended flows with lower priorities than the current flow, then the flow is set as active and it is sent via this preemptive path. If this is not the case, the flow is set as suspended. If more than one option is available, the path with the minimum number of idle switches is selected. When a process is finished, controller sets flows once again. In order to show the effectiveness of the proposed scheme, authors implement the EXR and FSR, using an OpenFlow protocol. Throughout the simulations, authors compare effective utilization ratios of the EXR and FSR. The results show that EXR uses links almost with the full capacity; however, FSR only reaches about 50% utilization of its full capacity. As a conclusion, irrespective of whether the network is idle or busy, EXR enables energy saving by creating efficiently utilized link structure or full use of capacity. Assigning priorities to the flows increases the chance to obtain more flexible management of the network. Delays can be prevented using the Earliest Deadline Strategy, or throughput can be improved using shortest job first strategy. Authors observe the average utilization ratio of all the active links through a partial Fat-Tree test bed that contains two pods and 8 servers. Results show that while the effective link utilization ratio under EXR almost reaches 100%, it only hovers at around 50% most of the time when using FSR. This result validates energy-centric efficiency of the EXR. However, it should be noted that network resiliency implies backup paths and fast switching times that mandates idle resources. Therefore, the proposed scheme may not be able to maintain an acceptable level of service in the face of faults and challenges to normal operation.

Authors in [35] focus on re-routing the traffic according to the network status. They take advantages of SDN to create their model by collecting real-time information about network traffic and information related to users' requests before joining the network. The work mainly focuses on integrated chassis[2] and line-cards, as they consume energy more than any other element in a router. The goal of the study is to minimize the link usage in such a way that the total link utilization will be under 50%. In this context, authors symbolize routers as star graphs, where the center represents the integrated chassis and the leaves represent the line-cards. If there is no request, line-cards are set to the sleep mode and if all line-cards are asleep, then the integrated chassis is set to the sleep mode as well. To validate the efficiency of the proposed scheme, authors run simulations on three synthetic networks that have different sizes and compare their scheme with the Cplex method. The results show that the proposed scheme saves an important amount of energy, outperforming Cplex. However, as depicted by the authors, proposed method results in highly-loaded network environment, which makes the network vulnerable to link failures and sudden traffic bursts.

Authors in [36] present a different approach for re-routing the traffic. They demonstrate that existing routing protocols, such as Open Shortest Path First (OSPF) or IS-IS [37] spread the load among multiple paths and achieve QoS. Nevertheless, these protocols do not save enough energy, since they increase the total activity of links (turning them on and off). Therefore, authors propose a new method called Routing for Minimization of Active Devices (RMAD). What RMAD essentially does is to find a number of shortest paths and among them, to choose the path that has the maximum number of active links. The aim is to avoid activating more links and to increase the sleep time of the links. Afterwards, sleep state links are activated along the chosen path. Authors also propose RMAD+, which provides a hop count constraint on paths once shortest paths are found and chooses the path with the least number of active nodes. Since nodes consume more energy than links, RMAD+ is estimated to save more energy than RMAD due to having less active nodes. In order to evaluate the efficiencies of the proposed two methods, authors utilize multi-rooted fat-tree topology for their simulation and observe the node sleep ratio and average transmission time. The results illustrate that node sleep ratios of RMAD and RMAD+ are higher than the ratios of OSPF (RMAD+ is the highest). RMAD+ also has the highest average transmission time since it chooses the path with the lowest number of active nodes, which is not necessarily the shortest path.

Flow-controlled SDN solutions mainly provide an internal communication among network devices to exchange power-related data and optimize the choice between energy saving and network performance according to the traffic load. Since stations are aware of the traffic load of the network/rest-of-the-nodes; controllers set an optimized traffic flow by turning off as many devices as possible, collecting real-time information about network traffic and information related to users' requests before joining the network. In this context, existing flow-controlled SDN approaches can balance QoS and energy efficiency with a qualitative network management system [28,30], provide multiple options for changing traffic load [29], enable fair-share routing [34], support an optimized energy management during low activity periods [35], and decrease the number of active devices to obtain power saving while having qualitative communication [36]. Nevertheless, these solutions may also have some drawbacks. For instance, execution of OpenFlow with GAL may result in delay for stations in response time [28], controller's response time to traffic changes may not be that effective as suggested in fast changing channel condition [29], implementation of all three level operations may cause additional overhead [30], transition between active and suspended flows may cause additional delay as all links work with very high capacity [34], energy saving can be achieved only when the network is relatively idle [35], or an increase in energy efficiency may even cause a dramatic decrease in network throughput performance [36].

4.2. Energy-aware SDN approaches for data center networks

Data Center Networks (DCNs) are another application area of green networking. The rise of IT technologies, such as cloud computing, led to an enormous increase in the number of DCNs. To offer full time qualitative service experience, data centers are 7/24 available and this situation results in an excessive use of energy. In 2009, DCNs were accounted for 2% of worldwide energy consumption with an economic impact of US $30 billion [38]. This rate is expected to grow apace every year since data center hardware expenditures are growing dramatically [39]. Some of energy-aware DCN-related surveys can be found in [40,41]. A typical DCN is the home to thousands of hosts; it stores hundreds of networking elements, many processors, power suppliers, cooling systems, etc. In order for the networking part to work properly, all of these parts should be turned on [42].

To decrease the power consumed by DCNs, first we need to optimize power consumption of servers and other networking elements since they are the fundamental sources of power consumption [43]. For instance, storage represents 20 to 40 percent of total power consumption in a typical data center. It consumes an important amount of energy even if the device is in the idle mode. Therefore, optimizing storage problems can improve energy saving. Resource sharing between servers, such as memory, storage, and fans, can be another way of tackling the energy efficiency problem. Another approach is to optimize components that do not consume power proportionally with their utilization. The average utilization of a server is mostly about 10–50% of its maximum utilization [44] and more than 80% of its links have very low traf-

---

[2] A router consists of an integrated chassis (includes power supply, heat dissipation system, cables etc.), several line-cards and multiple ports [24].

fic [45]. Thus, re-arranging traffic using optimal paths and eliminating idle resources by putting them into low-energy states or turning them off enable saving high amount of energy [46]. In the literature, there are numerous studies that aim at reducing the total energy consumption caused by DCNs. However, throughout this paper, we will only investigate energy-efficient DCN solutions that are based on SDN.

ElasticTree [47] is one of the most popular works proposed to save energy in DCNs using SDN. It dynamically optimizes network energy by ensuring qualitative traffic flow, meeting performance constraints, and turning off as many devices as possible. It is comprised of three parts: *optimizer, routing*, and *power control*. Optimizer part outputs a minimum-power subset of the network for a given load using the inputs of fault tolerance, power model, traffic matrix, and topology. Then, the results are sent to the routing and power control parts. Routing part chooses the paths for flows and applies the routing set. Finally, the power control part arranges power states and turns the device on or off if necessary.

In ElasticTree, the optimizer has three options to enhance the traffic: *Formal Model, Greedy Bin-Packing*, and *Topology-aware Heuristic*. Formal Model formulates the problem as *Multi-Commodity Flow* problem and tries to minimize network power while satisfying all traffic constraints (e.g. link capacity, flow conservation and demand satisfaction) of MCF problem. Formal model forces traffic to flow through active links and switches. However, its scalability is low, and it takes time to calculate the optimal set. Therefore, it is useful for networks with less than 1000 nodes. The second option, which is Greedy Bin-Packing, does not distribute the flow randomly; instead, it chooses the left-most path. If a flow cannot be transferred using any path, it will be sent to the path with the lowest load. Finally, Topology-aware Heuristic uses advantages of fat-tree topology to manage the traffic. It assumes perfectly divisible flows; it splits the data according to the link capacities and activates the required links in line with the demand of above or below layer. Perfectly divisible data is not always practical and may lead low quality solutions; yet the proposed heuristic method is faster than the others and it requires less information to work.

ECODANE [48] is another project that aims to obtain energy efficiency in DCNs. It has five modules: *Data center network, Optimizer, Power control, Forwarding*, and *Traffic generator*. Authors use the ElasticTree [47] data center network, which is designed for Energy-optimized Data Centers. The Optimizer module computes minimum energy subset using the Topology-aware heuristic [47]. The Power control module forces switches to turn on/off or to change the state to an appropriate power mode. The Forwarding module uses a hierarchical load balancing routing module, which works together with the optimizer. Throughout the paper, authors perform extensive tests to observe the efficiency of their proposed method and obtain 10% - 35% energy saving depending on the traffic locality, whether it is highly localized (within a rack), mid-localized (within a POD), or non-localized (across the data center).

A hybrid data center testbed, using the idea behind the ECODANE project, is proposed in [49]. The aim of the work is to obtain a hybrid testbed using real network elements and virtual emulation tools, to have an OpenFlow controller supporting energy-aware functions, and to better understand which parts of the networking elements consume how much energy. Hybrid testbed helps to obtain scalable and realistic network experience since it combines physical and virtual tools. In this work, network architecture consists of a data center network, an OpenFlow controller, and a traffic generator. NetFPGA [50] routers, which allow measurements of power consumption and regulations of power scaling, are used as the hardware part of the network. Additionally, Mininet [51], a simple and low-priced network testbed to develop OpenFlow applications, is used for the emulation.

Authors in [52] propose ECDC, which is an energy-efficient data center approach. It is an OpenFlow-based data center scenario having different types of customers, such as private home users or business users. ECDC separates the traffic according to users' demands. The network is composed of OpenFlow controller, switches, servers, and a management station that is connected to all other devices in the network. Management station orchestrates the network, generates policies, allocates virtual machines, and turns off unused devices by checking predefined threshold values or timeouts of each services. As long as monitoring information on CPU, power, and memory-load is available and provided to the management station, ECDC enables saving an important amount of energy.

In [53], authors propose a Correlation-Aware Power Optimization Algorithm (CARPO), which consolidates traffic flows by eliminating unnecessary links to decrease energy consumption. CARPO focuses on saving the energy of DCN rather than lowering servers' energy consumption. In this work, authors implement the algorithm using a central power manager and OpenFlow switches. Since the switches consume high percentage of total energy of the network even when they are in idle mode, turning them off is the best way to lower the energy consumption. Using the constraints of maximum link capacity and the equality of incoming and outgoing data rate, consolidation procedure tries to minimize energy consumption by turning off switches as much as possible.

The difference of CARPO from the existing approaches is that CARPO rearranges the traffic flow according to the correlations between flows. In [53], authors validate two main observations. First, different traffic flows are not tightly correlated or they are sometimes even negatively correlated, so they do not peak at the same time. Second, 90% of the link utilizations is much less than the peak values of the flow, so the consolidation process must be based on the average link utilization other than peak values to save more energy. In this context, CARPO first analyzes correlations between flows, and then consolidates the traffic accordingly. Finally, the algorithm sets link rates according to the demand. The authors solve the problem using mathematical tools at first and obtain a near-optimal solution, which requires perfect knowledge of future demand. To obtain a real-life solution for the problem, the authors also introduce an algorithm based on greedy-bin packing. This algorithm assigns links to the flows, while satisfying link capacity and correlation requirements between concurrent flows. In order to show the effectiveness of the proposed scheme, authors also compare the energy consumption and the performance of CARPO with the ElasticTree [47], GoogleP [44] and CARPO-C, which is a version of CARPO without link rate adaptation. In this context, authors implement a hardware test-bed that is composed of 10 virtual switches configured with a production 48-port OpenFlow switch and 8 servers. The empirical results show that CARPO leads to high amount of energy saving (up to 46%) and a limited delay increase for a DCN.

Consequently, utilizing SDN in data center networks may lead to several energy-centric advantageous, such as obtaining qualitative performance, energy efficiency and robustness at the same time [47], adapting the set of active network components dynamically according to the total traffic [48], balancing the QoS and efficient use of resources [52], and focusing on saving network's energy other than server's deeply analyzed traffic [53]. Yet, the existing proposals may also have some drawbacks. For instance, in [47], finding an optimal set of active links and traffic flow may take longer than predicted and any adaptation attempt may fail or may cause a performance loss in fast changing channels. In [48], overall throughput performance may even decrease since there is no adjustment for switches' link rates. In [49], DC testbed architecture may cause sensitive data to be transferred late. In [50], using outdated information may result in dramatic decrease on performance and energy saving since monitoring information on CPU, power, and memory-load may not be always available. Besides, in [53], highly loaded links may cause delay-sensitive multimedia flows suffer as the procedure increases delay.

4.3. Rule-placement and TCAM-based energy-aware SDN approaches

Authors in [54] propose a method to eliminate the rule space problem of existing energy-aware routing (EAR) approaches. EAR approaches mainly assume that there is an infinite amount of rule space and hence an OpenFlow switch can hold an infinite number of rules. However, this assumption is not realistic in practice, as flow tables are implemented with Ternary Content Addressable Memory (TCAM), which is expensive and power-hungry. To solve the problem, authors come up with the idea of *default rule*. According to this work, in a routing process, if there is not any predefined rule for a packet, the default rule is applied. To avoid communication delays between routers and the centralized controller, packets that have the default rule are forwarded only to a default port (each switch has only one default port) without contacting the controller. In this context, authors first formulate the problem using Integer Linear Program (ILP) and present a heuristic algorithm to minimize energy consumption. The algorithm consists of two steps. First, it finds feasible routing solution for each request, providing capacity and rule space constraints. The flow table is filled until it gets full. Subsequently, the port that carries most of the flow is assigned as the default port so that there will be more space available for new rules. Second, it reduces the number of active links; the links with low traffic load are turned off and their traffic is transported to other active links. An important portion of energy can be saved using the proposed method as it reduces the amount of rule space, communication delay between routers and the controller, and the total number of active links.

Another method that aims to decrease the power consumed by TCAM is the Compact TCAM method proposed in [55]. Since lookup process consumes high amount of energy while searching for 356 bits[3] flow entries in TCAM flow table, the authors introduce the Flow-ID concept in lieu of standard flow entries in order to decrease the power consumption and cost. In the new TCAM lookup process, for a flow to be forwarded, a complete match in the flow table is not necessary; a smaller match is also enough. Thus, flow entries can be compacted into smaller IDs and switches can apply all processes with this new ID. The Flow-ID assignment process starts once a new flow enters the network. Ingress switch forwards a packet to the controller if the flow is not found in the flow table. The controller forms a Flow-ID, stores it in its local reference table, and then re-sends to the switch containing some information with a COMPACT flag. If the flag is clear, the rules on the packet are processed. Otherwise, the actions based on Flow-ID are performed. At the final step, the controller gives a command to remove the Flow-ID and then the original packet is returned and forwarded. Authors build the proposed Compact TCAM framework on an OpenFlow enabled NetFGPA platform and demonstrate that switch fabric power gain can be about 2.5 times of a layer_2 switch and high amount of energy saving (up to 80%) can be achieved compared to SDN switches.

It has hitherto been clear that the centralized nature of the SDN controller and its ability to follow every single flow in the network is beneficial for network management. Authors in [56] propose DevoFlow, which is a modification of the OpenFlow controller. It simply breaks the coupling between centralized control and centralized visibility; in other words, it is a protocol that disburdens the controller by assigning some of its work to switches and obtain cost-effective and energy efficient management of the network. The underlying reason for this modification is the interference of the controller in flows transfers high amount of workload to the control plane. Besides, collecting statistical (periodical) information all around the network burdens the controller and lessens its ability to manage big packets. To eliminate these deficiencies, authors develop some mechanisms to increase the control of switches and the efficiency of statistics collection. Rule Cloning is the action where if a flow matches with another flow, the switch clones the rule locally to prevent unnecessary use of TCAM and to minimize communication between controller and switch. In addition, with the proposed Local Actions mechanism, switches can manage some set of flows on their own without invoking the controller. Furthermore, to improve the efficiency of OpenFlow statistics collection, authors offer two different approaches; (i) Use sFlow and (ii) Triggers and Reports. Use sFlow, which is implemented in a switch, collects information that contains headers of randomly chosen packets via sampling and transmits it to the controller. Triggers and Reports sends statistical information with a push-based mechanism only when a triggering condition is met. Accordingly, authors' custom-made simulation results show that DevoFlow can increase throughput over ECMP routing by 16% and 24% for Clos and HyperX networks, respectively.

Authors in [57] also discuss TCAM used in OpenFlow switches. To lower the power consumption and latency, they propose a new OpenFlow switch. In this work, when a packet is received, it is directed to the input memory. While receiving it, a module called Packet Parser extracts its important fields and creates a flow-key, which identifies the packet. Flow-keys are stored and searched in flow tables that can be implemented in software-based SRAM or hardware-based TCAM. In addition, to lower the time used in TCAM, a packet prediction circuitry is also implemented. The circuitry tries to predict the flow-key and prevents the flow-key to enter the TCAM lookup process. The paper presents two prediction methods. One is Direct Map, which extracts flow-keys from predefined location. This method selects the most varying fields between subsequent flows. The second one is called Sub-Field Hash. It intelligently extracts flow-keys from precisely defined fields and uses DJB Hash function to form a packet signature. Prediction circuitry achieves both energy saving and low latency, bypassing TCAM. Energy saving of switches with prediction circuitry will be high enough and energy waste will be insignificant if correct prediction rates are relatively high.

Consequently, Rule-placement and TCAM-based SDN approaches may enable an energy-aware routing method while optimizing rule spaces of switches [54], reduce the size of flow entries and manage large sized SDN flows [55], ease the controller's work by transferring some of its works to switches [56], and lessen the effect of TCAM on power consumption [57]. Nevertheless, fast-changing dynamic channels may also cause switching the default port too frequently [54], forming a Flow-ID, storing and re-sending it to the switch may also increase latency [55], local actions taken by switches complicate the controller's global view and management [56], and fast-changing channels may reduce the correct prediction rates [57].

4.4. VM placement-based energy-aware SDN approaches

Energy Efficient and QoS aware Virtual Machine (VM) Placement (EQVMP) [58] is a method that aims to cure lack of energy saving mechanisms of VM placement. Energy-efficient algorithms in VM placement policies sacrifice from network performance. After aggressively cutting down the energy usage, traditional routing algorithms fail to find optimized routes for flows. SDN can overcome this issue by implementing effective routing algorithms and appropriate VM allocations in order to achieve low

---

[3] In SDN, flow entries can potentially support large number of flows, such as 15 field tuple - that requires 356 bits to define a flow.

latency and high throughput, i.e. a qualitative network. To maintain a desired network performance, authors propose EQVMP algorithm, which consists of three steps: VMs resource demand, VM traffic, and topology matrix. In EQVMP, Hop Reduction part gets active at first. VMs and the traffic load between VMs are represented as a graph. To reduce the number of hops, the graph-partitioning problem is modeled. In this context, the graph is partitioned into smaller groups with fewer connections. This way, Hop Reduction reduces the cost of relation between groups and increases the balance of each group. After reducing the traffic load between groups and lowering the delay with shorter connections (Hop Reduction), Energy Saving part minimizes the resource utilization. After sorting VMs according to the resource demands in decreasing order, the algorithm seeks a VM with minimum resources available for a recently activated server. Finally, the last step is the Load Balancing. This module prevents congestion with the help of SDN structure, and re-routes flows using the controller. If link utilization reaches a pre-determined threshold value, flows are headed to another link that has less utilization. In order to validate the efficiency of the proposed scheme, authors build an Ns-2 based simulation environment that has a 3-tier fat-tree datacenter network, consisting of 16 core-level, 32 aggregate-level and 32 edge-level switches. Each edge-level switch can connect 8 servers, and each server can host 4 VMs. EQVMP is compared to well-known VM allocation methods, such as First Come First Serve (FCFS), Largest Task First (LTF), and Round Robin (RR). The results show that EQVMP outperforms other allocation methods and enhances system throughput by almost 25%.

Authors in [59] focus on traffic-aware placement of VMs to obtain better utilization of resources. They present two algorithms: *server-driven* and *network-driven*. These algorithms aim to efficiently allocate resources of intra-DCN. The server-driven algorithm first chooses a server for a VM and then determines the switches that will provide the flow of traffic. The network-driven algorithm first selects the switches and then an appropriate server for VM. To select servers and switches, there are three options: First Fit (FF), Best Fit (BF), and Worst Fit (WF). FF ranks servers and switches in ascending order according to their computational power and overall traffic load, respectively, and then chooses the first one in the list. WF chooses the most unloaded servers or switches. Finally, BF chooses the most loaded servers or switches. In order to show the effectiveness of the proposed method, authors evaluate the results obtained in simulation in terms of link utilization. The results show that network-driven algorithm with FF option leads to higher power saving than the others.

Authors in [60] combine VM placement and routing optimization in DCNs to achieve energy saving and propose a joint host-network algorithm by using depth-first search with best-fit option and selecting an element with the smallest and sufficient remaining capacity. Authors allocate particular nodes for each VM and another dummy node for each connection between the VM and server to indicate the memory constraint of the server. They also introduce a parallel processing method that divides DCNs into clusters and finds optimal paths in a parallel way. The method deals with connections in between subnets of servers and VMs. Authors' custom-made simulation results demonstrate that the proposed scheme outperforms existing *network-only* and *host-only* optimization solutions in terms of energy efficiency and throughput.

In a nutshell, VM-placement based SDN approaches mainly combine two effective methods to save energy; (i) VM placement and (ii) routing optimization [60]. These approaches may result in not only energy saving, but also achieve high throughput and low latency [58], provide cloud-fluent traffic engineering and increase in link utilization [59]. Nevertheless, these approaches may also have some drawbacks; re-computation time of new VM-placement might be higher than predicted [58], proposed scheme(s) may be too late to react in fast-changing channels [59], and operation of VM migration itself may consume high energy [60].

4.5. Different types of energy-aware SDN approaches

In the literature, most of the proposed energy-aware networking solutions make use of Layer 3. However, SDN can also provide efficiency in lower layers. As an example, authors in [61] propose a controller that makes use of Layer 2 protocols to reduce the energy consumption. Main goal of the proposed scheme is to prevent connections that cause loops between OpenFlow-enabled elements. The controller in this work is called GreenMST. It uses minimum spanning tree protocol (STP) to prevent excessive energy usage. The control software was built upon the Beacon controller [62], which is an open-source, multithreading-supported, and Java-based OpenFlow controller. GreenMST stores instantaneous states of the network using Topology Manager (TM). TM uses Link Layer Discovery Protocol to figure out whether there are any changes in the connectivity of switches. In case of a change, new minimum spanning tree of the network is calculated using Kruskal algorithm. Afterwards, the proposed scheme creates a Change Set, which informs the controller about the links that will be turned on or off. In order to examine whether the system is working properly, authors mainly study three different periods during the tests. First, they examine the period starting from the activation of GreenMST until reaching to a stable condition so that GreenMST rules are ensured (no loops, no partition, etc.). Second, they examine the period when a new device enters the network until the controller updates the overlay of the network. Finally, the third study covers the period of calculating MST and checking the consistency of the network with GreenMST settings again. GreenMST can save an important amount of energy as it provides loop-free communication among devices. However, the proposed system does not detect network failures, such as a link-down. Besides, it may not work well in large networks since the time measurements reveal that the time spent does not grow linearly with the increasing amount of links between devices.

Authors in [63] introduce a Hyper-Cellular network protocol called HyCell, which is an example of green software-defined radio access networks (SDRAN) with energy efficient base station (BS) operations. HyCell decouples control and traffic coverage, and manages them using BSs, Control BSs (CBSs), and Traffic BSs (TBSs). CBSs provide central control over TBSs and mobile users. CBSs are responsible for user equipment's (UE) network access. In this context, CBSs assign UEs to the nearest and available TBSs. HyCell also decouples software and hardware of BSs in order to provide easier updates for BSs. Active TBSs send their traffic load information to CBSs, which then determine the least loaded TBS for a waiting user. Since CBSs have the global view of TBSs, they can put TBSs into sleep mode if they have a low traffic load to maximize energy efficiency. Authors also point out that the wake up time of PCs are high. Therefore, the proposed method may not be suitable for scenarios requiring low latency.

Authors in [64] implement an algorithm to reduce energy consumption in Passive Optical Networks (PON), which are fiber optic access networks. According to the statistics, access networks are responsible for almost 70% of the total energy consumption of the Internet [65]. PON systems also consume high amount of energy in the Internet today. In a PON, there are mainly two modules: Optical Line Terminal (OLT) and Optical Network Unit (ONU). Authors create an EMM-UDS algorithm to reduce energy consumption by defining three energy states for both OLT and ONU: *awake, port sleep*, and *board sleep* for the OLT, and *awake, sleep*, and *critical* for the ONU. The proposed algorithm is initiated when the maximum time required to complete a data transmission is smaller than the pre-determined threshold value. Once the transmission is completed, ONU sends queue length information to the OLT, and then OLT sends this information to the OpenFlow controller. The controller decides on the sleep conditions of ONUs. If the controller decides to put ONU into sleep mode, the corresponding OLT port is also set to port sleep mode. After a certain period, ONU enters the critical mode and OLT port activates the awake mode to warn the controller about the buffer queue length. If the controller wakes up the ONU, the power state of ONU is also changed to awake mode.

PayLess proposed in [66] is a monitoring framework that decreases the cost of collecting statistical information. In SDN, applications collect statistics via controller with different periods, i.e. per flow, once in five second, etc. Collecting this information increases the burden on the controller and causes network overhead. PayLess provides a separate layer to control the collection of statistics. It consists of four components; *Request Interpreter, Scheduler, Switch Selector*, and *Aggregator & Data Store*. Request Interpreter provides communication between layers. Scheduler manages time for collecting statistics. Switch Selector chooses the switches from which the information will be extracted. Finally, data is collected and stored by the Aggregator & Data Store component. Applications request statistics using RESTful API of PayLess. Users then create a Monitoring-Request object, specifying Type, Metrics, Entity, Aggregation Level, Priority, Monitor, and Logging information about the requested monitoring object. PayLess responds with an access-id, and then application uses this id to extract requested information. Although energy efficiency is not the main goal of PayLess, statistical information obtained through it can also help to achieve an energy-efficient networking environment.

Authors in [67] propose an extension for OpenFlow switches to make them work in different power saving modes. In this work, Power Control module, referred to as OpenFlow Switch Controller (OSC), sends the power changing commands. Controller employs OSC to manage power states of switches for turning on or off a switch, disabling or enabling links or ports, or changing energy consumption level. They introduce three types of new messages (OFPT_PORT_MOD, OFPT_LINECARD_MOD and OFPT_SWITCH_MOD) to change the energy states of ports, line cards, and switches. Accordingly, authors build a hardware test-bed including a NOX Controller, an OSC and a NetFPGA based OpenFlow switch. Experimental results show that the proposed extension can send commands from the NOX to the OSC to turn on/off the switch and to turn on/off the ports of the NetFPGA card. In addition, total power consumption (1900 mW) cost of the extension is negligible, when the potential of high amount of power saving is considered.

Authors in [68] emphasize that currently SDNs usually coexist with traditional networks, and hence they are partially deployed at present. In this context, authors investigate how to save energy in partially deployed SDNs. First, they formulate the problem as a multi-commodity flow problem using constraints on capacity, flow, etc. to minimize the total number of SDN devices. Since the formulation is not practical due to its complexity, a heuristic algorithm follows their formulation. The algorithm first obtains the smallest closed sets; nodes in this set only have traffic with the nodes of this set. The algorithm gives weights to the links and nodes in order to calculate the energy consumption of each possible path. According to the traffic demand, the algorithm then finds spanning trees for the smallest closed sets to figure out the path with minimum energy. The purpose here is to minimize the distance. If a traffic flow between a set of nodes is chosen, the algorithm first chooses an edge between these nodes with the smallest weight and continues this process until it reaches the destination. If there is an overloaded link, then the algorithm removes it and repeats the same steps using the second highest option.

Accordingly, the proposed SDN controller approach [61] that mostly makes use of Layer 2 protocols may assist avoiding loops in-between connections. However, in the proposed scheme, computation time increases faster than the number of links between devices, which can be a problem in large networks. Energy efficiency can also be increased in Hyper-Cellular networks with the help of SDN, yet the proposed method [63] may not be suitable for scenarios requiring low latency. Additionally, PayLess proposed in [66] collects statistics and eases the burden on the controller, yet it is not compatible with distributed controller platforms. Finally, the proposal for the extension of OpenFlow switches [67] provides detailed control over line cards' and ports' energy consumption. In this work, energy efficiency can be further improved with a dynamic link rate adaptation.

5. Evaluation of energy saving techniques and proposed approaches ın SDN

Energy saving in SDN can be addressed through hardware or software-based enhancements; in other words, energy efficiency can be achieved in chip, node or network levels. While hardware-based solutions are mainly applied on forwarding switches, software-based solutions are applied on the controller or on access nodes. Link rate adaptation, load balancing among links, re-routing the traffic flow, turning a device (or only a part of it) on or off, rule placement, minimizing the TCAM, network virtualization, etc. are fundamental techniques that address energy saving in SDN.

Traffic-aware solutions, such as link rate adaptation, load balancing among links, and re-routing the traffic flow, are inspired by the fact that network components are often underutilized. Therefore, turning on or off network devices (or only a part of it, such as a forwarding switch, a CPU, and a port) dynamically according to network conditions and traffic load enables saving an important amount of energy. Rule placement is another way to save energy in SDN. It mainly focuses on how to place the rules in forwarding switches, and also on how many rules (a forwarding switch can hold a finite number of rules) must be defined to ensure that similar flows follow similar paths. In addition, forwarding switches in SDN use TCAM. Since TCAM is very expensive and power hungry, TCAM approaches proposed in the literature aim to reduce the memory need of information stored in the forwarding switches. Last but not the least, in a virtualized network, physical elements can contain more than one virtual network structure. Therefore, it may reduce both the cost of constructing new hardware for the new network and power consumed by this hardware. Network virtualization also provides dynamic network abilities; in case there is an immediate need for a change in a network, it enables fast transmission from the old system to a new one [9]. This capability of network virtualization also results in performance increase and energy saving.

All of the energy-efficient SDN approaches examined in Section IV evaluate the effectiveness of their approaches by implementing their own simulation environment. The parameters, topologies, network condition, and the simulation software used by these approaches are completely different from each other and mainly no comparison is made between existing approaches. Authors simply compare the efficiency of their algorithms with the traditional networking scenario. Therefore, it is extremely hard to compare their efficiencies in terms of energy saving without implementing these solutions one by one in the same platform.

Throughout this section, we evaluate the expected energy efficiency levels of each approach individually. The evaluation criteria are carried out by taking into consideration the specific metrics of each approach, such as the target environment (fixed, mobile or data center), maximum achievable throughput, coverage, number of active links/devices, impact of each parameters used, the topology used in simulations, achieved results, static/dynamic structure of the implemented network, advantageous/disadvantageous of the proposed scheme, issues that may arise, overhead of each issue, impact of the operation (chip-level, node-level, network-level), additional protocol supports, advantageous/disadvantageous of the additional protocol support, etc. Weight batching of the aforementioned metrics, which are investigated in Section 4 in detail, we believe that it will also be possible to reach to a general opinion about these proposed works. In this context, in order to summarize the features, differences and the expected amount of energy gains of the proposed energy-efficient SDN solutions, we make a brief comparison and clarify the issues that may arise in Table 1.

As shown in Table 1, expected energy gain is classified into three levels; (i) *High*, (ii) *Medium*, and (iii) *Low. High* implies that the proposed scheme is expected to save high amount of energy together with a throughput increase or at least without a throughput loss. *Medium* implies that the proposed scheme is expected to achieve high or considerable amount of energy at the expense of limited throughput decrease. Finally, *Low* implies that the proposed scheme is expected to save only a limited amount of energy while having decrease on throughput as well.

Apart from the energy efficiency levels, proposed SDN solutions are mostly based on a single energy saving technique as shown in Table 1. However, these techniques are mostly independent from each other and can be applied all together to further increase the energy efficiency.

6. Issues, challenges and future research directions

Although SDN architecture has been introduced as a troubleshooter, it also brings some new issues that need to be regulated. For instance, the risk of attacks to the network may increase with the SDN due to decoupling of the control and data plane. Recently introduced elements, such as controllers-software, control-data and control-application communications, can face threats. In addition, there is a lack of SDN security applications, which should provide a secure transmission between control and application layer through Northbound API.

Since there are multiple vendors and operators in the network industry, coexistence of network components is another issue. To create an SDN structure, independent components need to be composed; hence, network elements should be inter-operable. In this kind of multi-vendor industry, there should be a standard for the inter-communication of network devices. SDN has achieved the creation of a dynamic network structure by facilitating users to control the network via applications; nevertheless, the number of applications to manage and regulate the network is insufficient. Therefore, additional effort must be performed in this direction [7].

As mentioned through Section 4, turning a network (or a part of it) on/off dynamically based on channel utilization may result in energy saving [29,47,52-54,67]. Yet, it is challenging, an NP-hard problem [69], as there is a trade-off among energy efficiency, delay and throughput performance. Energy-efficient SDN paradigm has been examined very little from the theoretical perspective in the literature. As future research, optimization problems that focus on these trade-offs can be studied. These problems may then be analyzed from the perspective of approximation algorithms or parameterized complexity. Dynamic networks constitute an even more challenging environment for optimization problems and hence, applying heuristic solutions with low complexity while also addressing these trade-offs has paramount importance in dynamic networks.

Another issue is the optimization of the memory size used by forwarding switches as flow tables are stored in TCAM [54-57]. However, TCAM already provides a limited memory. If it is reduced even further, it will cause frequent flow entry replacements when new flows are to be installed. Frequent flow entry replacement means more accesses to TCAM and hence, more energy consumption. Optimization problems addressing these tradeoffs can be a fruitful avenue for research. For instance, one problem might be to optimize the memory size used by forwarding switches in order to minimize the energy consumption subject to a certain upper limit for flow entry replacements.

Another issue in SDN is the rule placement [54,56,62]. It has a direct impact on routing, and hence on network performance and energy efficiency. SDN controller has to place energy-aware adaptive rules into switches under some constraints, such as number of switches and routing policies. This issue has not been explored thoroughly in the literature. Novel traffic-aware and repetition-based rule placement methods that address the tradeoff between routing performance and energy efficiency in the SDN controller need to be designed.

Flexibility, which is the capability of the system to adapt to dynamic network settings, and scalability, which is the capability of a network to handle a growing amount of work, are also two essential research issues. Traffic-aware SDN approaches are required to be flexible [34] and scalable [49]. Although there have been many energy-efficient SDN related proposals in the literature, they do not mainly focus on scalability and flexibility issues stemming from dense network deployment, channel fluctuations and different nested topologies. Existing works can be explored in terms of horizontal (adding/removing more nodes to/from a network) and vertical (adding/removing resources to/from a single node in the network) scalability. Additionally, flexibility ratios of different nested topologies can be compared and more scalable and flexible novel solutions that can be change adaptively according to dynamic channel conditions can be proposed.

Fairness is another issue that should be taken into account in communication networks especially in resource allocation. Fair allocation of resources is usually desired; yet, this requirement may conflict with other objectives, such as total data rate and energy consumption. Fairness is vital in many scenarios such as bandwidth allocation, traffic engineering, Quality of Service (QoS), and energy consumption. Since fairness is a global measure as it is related to the resource allocation of all entities, global network knowledge and centralized decision-making mechanism of SDN make it convenient to provide fairness [34]. There are also certain limitations of SDN that lead to new fairness measures. For instance, SDN has limited flow table sizes; therefore, resource allocation mechanisms need to ensure fairness to the network entities in terms of flow evictions. A possible research direction is to design resource allocation methods providing fairness in terms of flow evictions and flow table sizes in SDN. In addition, fairness in energy efficiency is largely unexplored in SDN literature. When there are multiple SDN domains, fairness in the energy consumption of these domains should be taken into account while making resource allocation decisions related to routing, traffic engineering etc.

Network virtualization is also an effective way to overcome the ossification of Internet by enabling multiple virtual networks to coexist on a shared infrastructure [58–60]. Virtual network embedding/assignment is a resource allocation problem concerned with the assignment of physical resources to the virtual networks. In this context, virtualization of servers can be another way to reduce energy consumption in SDN. As an example, authors in [70] suggest using SDN for virtual network embedding due to its global topology knowledge. This way, multiple virtual machines can work on the same physical server. Thus, instead of using many servers inefficiently, constructing virtual machines enables additional energy saving. A possible research issue is to implement SDN-based routing algorithms with appropriate VM allocations that reduce the total number of hops, minimize resource utilization, balance the traffic load and achieve high throughput, low latency and energy efficiency.

Table 1
A brief comparison of energy-efficient SDN approaches proposed in the literature.

Ref. No	Protocol name	Utilized Controller	Emulation/ Simulation	Operation	Parameters Used	Technic/method used	Additional protocol Support	Energy Efficiency Level	Capabilities	Implementation difficulty	Issues that may arise	Target Environment
[28]	Green OpenFlow Extension	Any controller	Numerical analysis	Node-level based, Network-level based	Energy aware states, green abstraction layer, power management	Traffic-aware	GAL	High	Balances QoS and energy efficiency with a qualitative network management system	Hard to implement	Execution of OpenFlow with GAL may result in delay for stations in response time	Fixed Network
[29]	Strategic Greedy Heuristic	Any controller	CPLEX	Node-level based	Traffic load	Traffic-aware	–	Medium	Provides multiple options for changing traffic load	Easy to implement	Controller's response time to traffic changes needs improvement	Fixed Network
[30]	GreenSDN	Pox	Mininet	Chip, node and network level based	Traffic statistics, Power profiles, monitoring	Traffic-aware	SustNMS, SC and ALR	High	Balance between QoS and energy efficiency	Hard to implement	Implementation of all three level operations may cause additional overhead	Fixed Network
[34]	Exclusive Routing	Any controller	Custom-made	Network-level based	Active and suspended flows	Traffic-aware	–	Medium	Enables fair-share routing	Easy to implement	May cause delay as all links work with very high capacity	Fixed Network
[35]	Heuristic Algorithms	Any controller	CERNET	Network-level based	Link utilization, packet delay, network topology	Traffic-aware	–	Low	Optimized energy management during low activity periods	Easy to implement	Achieves energy saving only when the network is relatively idle.	Fixed Network
[36]	RMAD	Any controller	NS2	Node-level based, Network-level based	Sleep ratio, low number of active nodes	Traffic-aware	–	Medium	Decrease the number of active devices and obtain power saving while having qualitative communication	Easy to implement	Increase in energy efficiency causes a decrease in performance	Fixed Network
[47]	ElasticTree	Nox	Custom-made	Network-level based	Link capacity, flow conservation, demand satisfaction, fault tolerance	Data center-based	–	High	Obtaining qualitative performance, energy efficiency and robustness at the same time	Hard to implement	Adaptation to dynamic channels: Finding an optimal set of active links and traffic flow may take long time	Data center networks
[48]	ECODANE	Nox	Mininet	Network-level based	Power modes, traffic locality, link utilization	Data center-based	ElasticTree	High	Dynamically adapts the set of active network components according to the total traffic	Hard to implement	There is no adjustment for switches' link rates	Data center networks

(*continued on next page*)

Table 1 (*continued*)

Ref. No	Protocol name	Utilized Controller	Emulation/ Simulation	Operation	Parameters Used	Technic/method used	Additional protocol Support	Energy Efficiency Level	Capabilities	Implementation difficulty	Issues that may arise	Target Environment
[49]	ECODANE	Pox	Mininet	Network-level based	Power modes, traffic locality, link utilization	Data center-based	ElasticTree	–	Provides a testbed architecture that combines real network devices with emulation	Hard to implement	DC testbed architecture may cause sensitive data to be transferred late	Data center networks
[52]	ECDC	Any controller	open-source cloud management software	Network-level based	Traffic load	Data center-based	–	Medium	Balances the QoS and efficient use of resources	Hard to implement	Monitoring information on CPU, power, and memory-load, may not be always available	Data center networks
[53]	CARPO	Any controller	OPNET	Network-level based	Correlation, consolidating the traffic, link rate adaptation	Data center based	–	High	Focuses on saving network's energy other than server's, deeply analyzed traffic	Easy to implement	It may not be good for delay-sensitive flows as it increases the delay	Data center networks
[54]	Heuristic rule-space optimization	Any controller	CPLEX	Node-level based, Network-level based	Rule placement, TCAM	Rule space-based	–	High	Obtain an energy-aware routing method while optimizing rule spaces of switches	Easy to implement	Fast-changing dynamic channels may cause switching the default port too frequently	Data center networks, fixed networks
[55]	Compact TCAM	Any controller	Discrete event simulator in JAVA	Node-level based	Flow entry size, Flow-ID, latency, lookup process	TCAM-based	–	Medium	reduces the size of the flow entries and manages large sized SDN flows	Easy to implement	Forming a Flow-ID, storing and re-sending it to the switch will increase latency	Data center networks, fixed networks
[56]	DevoFlow	DevoFlow	Custom-made	Node-level based	Rule cloning, local actions	TCAM-based	–	Medium	Ease the controller's work, transferring some works to switches	Easy to implement	Local actions taken by switches complicates the controller's global view and management	Data center networks, fixed networks
[57]	TCAM Bypass	Any controller	Custom-made	Chip-level based	TCAM, flow-keys, flow tables, latency	TCAM-based	–	High	Lessens the effect of TCAM on power consumption	Easy to implement	Correct prediction rates may reduce in fast-changing channels	Data center networks, fixed networks
[58]	EQVMP	Any controller	NS2	Network-level based	VM resource demand, VM traffic, hop reduction, traffic load, delay, throughput	Network virtualization	–	Medium	An energy saving method that achieves high throughput and low latency	Hard to implement	Re-computation time of new VM placement is high	Data center networks

(*continued on next page*)

Table 1 (*continued*)

Ref. No	Protocol name	Utilized Controller	Emulation/ Simulation	Operation	Parameters Used	Technic/method used	Additional protocol Support	Energy Efficiency Level	Capabilities	Implementation difficulty	Issues that may arise	Target Environment
[59]	NV-aware traffic eng.	Any controller	Custom-made Java simulator	Node-level based, Network-level based	Link utilization	Network Virtualization	–	Low	Cloud-fluent traffic engineering and increase in link utilization	Hard to implement	Proposed scheme may be too late to react in fast-changing channels	Data center networks
[60]	Joint Host-Network Optimization	Beacon	Custom-made	Node-level based, Network-level based	VM Placement, Parallel processing	Network Virtualization	–	High	Combines 2 effective methods to save energy; VM placement and routing optimization	Hard to implement	the operation of VM migration itself may consume high energy	Data center networks
[61]	GreenMST	GreenMST	Custom-made	Node-level based	Loop-free communication, minimal interactions	Loop-free communication	LLDP	Low	Avoids loops in-between connections	Easy to implement	Computation time increases faster than #of links between devices, which can be a problem in large networks	Fixed networks
[63]	HyCell	Any controller	Custom-made	Network-level based	Base station, central control, load information	Hyper-Cellular	–	High	Presents an energy-efficient protocol for SDRAN	Hard to implement	This method may not be suitable for scenarios requiring low latency	Mobile networks
[64]	Poster	Any controller	Custom-made	Network-level based	Passive optical network, fiber optic, power states	Buffer queue length	IPACT	Low	Decrease in awake time of PON modules	Hard to implement	It is assumed that EPON system has only one OLT and two ONUs	Optical access networks
[66]	PayLess	Any controller	Mininet	Node-level based	Statistic collection, monitoring	Monitoring framework	RESTful API	Low	Collects statistics and eases the burden on the controller	Hard to implement	Not compatible with distributed controller platforms	Fixed networks
[67]	OSC	Nox	Hardware testbed	Chip-level based	Power saving modes, OpenFlow messages	OpenFlow switch extension	ECODANE	High	Provides detailed control over line cards' and ports' energy consumption	Hard to implement	Energy efficiency can be further improved with a dynamic link rate adaptation	Data center networks
[68]	Heuristic algorithm	Any controller	Mininet	Node-level based	Network topology, Weights on links,	Distance minimization	–	Medium	Designed for hybrid networks, reduction in computational time	Easy to implement	Proposed method does not perform well in highly-loaded networks	Data center networks

Most works about energy efficient techniques for SDN in the literature focus on a single SDN domain. However, SDN domains consist of different content providers and ISP networks, each managed by their own controller according to different policies. Algorithms developed for energy efficiency in single domain SDN environment might be inefficient in multi-domain SDN networks. As future research, novel techniques for energy efficiency in multi-domain SDN networks may be designed and comparatively evaluated by single-domain solutions.

In a nutshell, existing energy-aware SDN approaches mainly focus on a single issue (such as link rate adaptation, load balancing among links, re-routing the traffic flow, turning a device on or off, rule placement, minimizing the TCAM or network virtualization). Although it is not possible to apply load balancing among links and turning resources on and off simultaneously to gain energy, most of the other aforementioned techniques are independent from each other and can be applied simultaneously. Therefore, through gathering most of these techniques under a single/few roof(s), energy optimization can be achieved with a far-reaching solution proposal.

7. Conclusion

This paper focuses on evaluating the energy efficiency levels of the existing energy-centric SDN approaches, taking into consideration of specific metrics, such as the target environment, achievable throughput, number of active links/devices, the topology used in simulations and the achieved results. Link rate adaptation, load balancing among links, re-routing the traffic flow, turning a device on/off, rule placement, minimizing the TCAM, and network virtualization are fundamental techniques that address energy saving in SDN. Throughout the evaluation process, we show that proposed energy-centric SDN schemes may result in numerous benefits, such as balancing QoS and energy efficiency with a qualitative network management system, providing multiple options for changing traffic load, enabling fair-share routing, supporting energy management during low activity periods, and decreasing the number of active devices to obtain power saving while having communication with good quality.

Although SDN architecture has been introduced as a troubleshooter, it also brings some new issues that need to be regulated. For instance, execution of SDN-based energy-centric approaches may result in an increase in the response time of the station, finding an optimal set of active links and traffic flow may take longer than predicted and any adaptation attempt may fail or may have a performance loss in fast changing channels, implementation of all three level operations may cause additional overhead, operation of VM migration itself may consume high energy, and an increase in energy efficiency may even cause a dramatic decrease in network throughput performance. Furthermore, coexistence of network components is another issue. To create an SDN structure, independent components need to be composed; hence, network elements should be inter-operable. SDN has achieved the creation of a dynamic network structure by facilitating users to control the network via applications. Nevertheless, the number of applications to manage and regulate the network is insufficient. Therefore, additional effort must be performed in this direction.

In a nutshell, proposed energy-centric SDN approaches mainly use a single technique to save energy. However, throughout this paper, we show that most of these techniques are independent from each other and can be applied simultaneously to further increase the energy efficiency. In this context, we believe that this paper can also pave the way for integrated energy-saving mechanisms in SDN as it summarizes and compares features, differences, expected amount of energy gains of related methods and addresses the issues that may arise.

References

[1] O. N. Foundation, "Software-defined networking: the new norm for networks (white paper)," April 13, 2012.

[2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, et al., OpenFlow: enabling innovation in campus networks, ACM SIGCOMM Comput. Commun. Rev. 38 (2008) 69–74.

[3] S. Zeadally, S. Khan, N. Chilamkurti, Energy-efficient networking: past, present, and future, J. Supercomput. 62 (2012) 1093–1118 2012/12/01.

[4] T.C. Group, (2010). Smart 2020 - enabling the low carbon economy in the information age. Available: http://www.smart2020.org/.

[5] A.P. Bianzino, C. Chaudet, D. Rossi, J.-L. Rougier, A survey of green networking research, Commun. Surv. Tut. IEEE 14 (2012) 3–20.

[6] D. Kreutz, F.M. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, S. Uhlig, Software-defined networking: a comprehensive survey, in: Proceedings of the IEEE, 103, 2015, pp. 14–76.

[7] Y. Jarraya, T. Madi, M. Debbabi, A survey and a layered taxonomy of software-defined networking, Commun. Surv. Tut. IEEE 16 (2014) 1955–1980.

[8] B. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, T. Turletti, A survey of software-defined networking: past, present, and future of programmable networks, Commun. Surv. Tut. IEEE 16 (2014) 1617–1634.

[9] M. Jammal, T. Singh, A. Shami, R. Asal, Y. Li, Software defined networking: state of the art and research challenges, Comput. Netw. 72 (2014) 74–98.

[10] S. Sezer, S. Scott-Hayward, P.-K. Chouhan, B. Fraser, D. Lake, J. Finnegan, et al., Are we ready for SDN? Implementation challenges for software-defined networks, Commun. Mag. IEEE 51 (2013) 36–43.

[11] A.T. Campbell, I. Katzela, K. Miki, J. Vicente, Open signaling for ATM, internet and mobile networks (OPENSIG'98), ACM SIGCOMM Comput. Commun. Rev. 29 (1999) 97–108.

[12] D.L. Tennenhouse, J.M. Smith, W.D. Sincoskie, D.J. Wetherall, G.J. Minden, A survey of active network research, Commun. Mag. IEEE 35 (1997) 80–86.

[13] D.L. Tennenhouse, D.J. Wetherall, Towards an active network architecture, in: DARPA Active NEtworks Conference and Exposition, 2002. Proceedings, 2002, pp. 2–15.

[14] I. Leslie, S. Crosby, S. Rooney, (2015). DCAN: devolved control of ATM networks, November. Available: http://www.cl.cam.ac.uk/research/srg/netos/projects/archive/dcan/#pub.

[15] A. Greenberg, G. Hjalmtysson, D.A. Maltz, A. Myers, J. Rexford, G. Xie, et al., A clean slate 4D approach to network control and management, ACM SIGCOMM Comput. Commun. Rev. 35 (2005) 41–54.

[16] R. Enns, M. Bjorklund, J. Schoenwaelder, NETCONF configuration protocol, Network (2011).

[17] Netconf, Configuration Management in High Assurance Environments (White Paper), 2009 January.

[18] M. Casado, M.J. Freedman, J. Pettit, J. Luo, N. McKeown, S. Shenker, Ethane: taking control of the enterprise, in: ACM SIGCOMM Computer Communication Review, 2007, pp. 1–12.

[19] E. Gelenbe, Y. Caseau, The impact of information technology on energy consumption and carbon emissions, Ubiquity 2015 (2015) 1.

[20] I. Gartner Estimates, Industry accounts for 2 percent of global CO2 emissions, press release, 2007.

[21] B. Lannoo, S. Lambert, W. Van Heddeghem, M. Pickavet, F. Kuipers, G. Koutitas, et al., D8. 1. Overview of ICT energy consumption, Network Excell. Internet Sci. FP7 8 (2013) http://www.internet-science.eu/sites/eins/files/biblio/EINS_D.

[22] W. Van Heddeghem, S. Lambert, B. Lannoo, D. Colle, M. Pickavet, P. Demeester, Trends in worldwide ICT electricity consumption from 2007 to 2012, Comput. Commun. 50 (2014) 64–76.

[23] F. Davoli, Green networking and network programmability: a paradigm for the future internet? presented at the SSEEGN 2013, Christchurch, New Zeland, November 20-22, 2013.

[24] B.G. Assefa, O. Ozkasap, State-of-the-art energy efficiency approaches in software defined networking, in: ICN 2015, 2015, p. 268.

[25] R. Bolla, R. Bruschi, A. Carrega, F. Davoli, Green network technologies and the art of trading-off, in: Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on, 2011, pp. 301–306.

[26] B. Naudts, M. Kind, F.-J. Westphal, S. Verbrugge, D. Colle, M. Pickavet, Techno-economic analysis of software defined networking as architecture for the virtualization of a mobile network, in: Software Defined Networking (EWSDN), 2012 European Workshop on, 2012, pp. 67–72.

[27] R. Bolla, T. Szemethy, R. Bruschi, F. Davoli, L. Di Gregorio, P. Donadio, et al., The green abstraction layer: a standard power-management interface for next-generation network devices, IEEE Internet Comput. (2013) 82–86.

[28] R. Bolla, R. Bruschi, F. Davoli, C. Lombardo, Fine-grained energy-efficient consolidation in SDN networks and devices, IEEE Trans. Network Service Manage. (2015).

[29] A. Markiewicz, P.N. Tran, A. Timm-Giel, Energy consumption optimization for software defined networks considering dynamic traffic, in: Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on, 2014, pp. 155–160.

[30] B.B. Rodrigues, A.C. Riekstin, G.C. Januario, V.T. Nascimento, T.C. Carvalho, C. Meirosu, GreenSDN: bringing energy efficiency to an SDN emulation environment, in: Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on, 2015, pp. 948–953.

[31] C. Gunaratne, K. Christensen, B. Nordman, S. Suen, Reducing the energy consumption of ethernet with adaptive link rate (alr), Comput. IEEE Trans. 57 (2008) 448–461.

[32] M. Mostowfi, K. Christensen, Saving energy in LAN switches: new methods of packet coalescing for energy efficient ethernet, in: Green Computing Conference and Workshops (IGCC), 2011 International, 2011, pp. 1–8.

[33] C.H. Costa, M.C. Amaral, G.C. Januário, T.C. Carvalho, C. Meirosu, SustNMS: Towards service oriented policy-based network management for energy-efficiency, in: Sustainable Internet and ICT for Sustainability (SustainIT), 2012, 2012, pp. 1–5.

[34] D. Li, Y. Shang, C. Chen, Software defined green data center network with exclusive routing, in: INFOCOM, 2014 Proceedings IEEE, 2014, pp. 1743–1751.

[35] R. Wang, Z. Jiang, S. Gao, W. Yang, Y. Xia, M. Zhu, Energy-aware routing algorithms in software-defined networks, in: A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on, 2014, pp. 1–6.

[36] D.N. Syouta Oda, Yutaka Fukuda, Takeshi Ikenaga, Flow-based routing schemes for minimizing network energy consumption using OpenFlow, presented at the ENERGY 2014, Chamonix, France, April 20, 2014.

[37] R.W. Callon, Use of OSI IS-IS for routing in TCP/IP and dual environments, 1990.

[38] J. Shuja, K. Bilal, S.A. Madani, M. Othman, R. Ranjan, P. Balaji, et al., Survey of techniques and architectures for designing energy-efficient data centers, IEEE Syst. J. (2014).

[39] Gartner Group. (2015). Forecast: data centers, worldwide, November , 2010–2015.

[40] K. Bilal, S.U.R. Malik, O. Khalid, A. Hameed, E. Alvarez, V. Wijaysekara, et al., A taxonomy and survey on green data center networks, Future Gen. Comput. Syst. 36 (2014) 189–208.

[41] K. Bilal, S.U. Khan, A.Y. Zomaya, Green data center networks: challenges and opportunities, in: Frontiers of Information Technology (FIT), 2013 11th International Conference on, 2013, pp. 229–234.

[42] D. Abts, B. Felderman, A guided tour through data-center networking, Queue 10 (2012) 10.

[43] S. Saha, J.S. Deogun, L. Xu, Energy models driven green routing for data centers, in: Global Communications Conference (GLOBECOM), 2012 IEEE, 2012, pp. 2529–2534.

[44] D. Abts, M.R. Marty, P.M. Wells, P. Klausler, H. Liu, Energy proportional datacenter networks, in: ACM SIGARCH Computer Architecture News, 2010, pp. 338–347.

[45] T. Benson, A. Akella, D.A. Maltz, Network traffic characteristics of data centers in the wild, in: Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, 2010, pp. 267–280.

[46] J. Carter, K. Rajamani, Designing energy-efficient servers and data centers, Computer 43 (2010) 76–78.

[47] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, et al., ElasticTree: saving energy in data center networks, in: NSDI, 2010, pp. 249–264.

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

[49] N.H. Thanh, B.D. Cuong, T.D. Thien, P.N. Nam, N.Q. Thu, T.T. Huong, et al., Ecodane: a customizable hybrid testbed for green data center networks, 2013 International Conference on Advanced Technologies for Communications (ATC 2013), 2013.

[50] J. Naous, G. Gibb, S. Bolouki, N. McKeown, NetFPGA: reusable router architecture for experimental research, in: Proceedings of the ACM Workshop on Programmable Routers for Extensible Services of Tomorrow, 2008, pp. 1–7.

[51] Available: http://yuba.stanford.edu/foswiki/bin/view/OpenFlow/Mininet.

[52] M. Jarschel, R. Pries, An OpenFlow-based energy-efficient data center approach, in: Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, 2012, pp. 87–88.

[53] X. Wang, Y. Yao, X. Wang, K. Lu, Q. Cao, Carpo: correlation-aware power optimization in data center networks, in: INFOCOM, 2012 Proceedings IEEE, 2012, pp. 1125–1133.

[54] F. Giroire, J. Moulierac, T.K. Phan, Optimizing rule placement in software-defined networks for energy-aware routing, in: Global Communications Conference (GLOBECOM), 2014 IEEE, 2014, pp. 2523–2529.

[55] K. Kannan, S. Banerjee, Compact TCAM: flow entry compaction in TCAM for power aware SDN, in: Distributed Computing and Networking, Springer, 2013, pp. 439–444.

[56] J.C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, A.R. Curtis, S. Banerjee, Devoflow: cost-effective flow management for high performance enterprise networks, in: Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks, 2010, p. 1.

[57] P.T. Congdon, P. Mohapatra, M. Farrens, V. Akella, Simultaneously reducing latency and power consumption in openflow switches, IEEE/ACM Trans. Netw. (TON) 22 (2014) 1007–1020.

[58] S.-H. Wang, P.P.-W. Huang, C.H.-P. Wen, L.-C. Wang, EQVMP: energy-efficient and QoS-aware virtual machine placement for software defined datacenter networks, in: Information Networking (ICOIN), 2014 International Conference on, 2014, pp. 220–225.

[59] M. Gharbaoui, B. Martini, D. Adami, G. Antichi, S. Giordano, P. Castoldi, On virtualization-aware traffic engineering in OpenFlow data centers networks, in: Network Operations and Management Symposium (NOMS), 2014 IEEE, 2014, pp. 1–8.

[60] H. Jin, T. Cheocherngngarn, D. Levy, A. Smith, D. Pan, J. Liu, et al., Joint host-network optimization for energy-efficient data center networking, in: Parallel & Distributed Processing (IPDPS), 2013 IEEE 27th International Symposium on, 2013, pp. 623–634.

[61] L. Prete, F. Farina, M. Campanella, A. Biancini, Energy efficient minimum spanning tree in OpenFlow networks, in: Software Defined Networking (EWSDN), 2012 European Workshop on, 2012, pp. 36–41.

[62] D. Erickson, The beacon openflow controller, in: Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, 2013, pp. 13–18.

[63] T. Zhao, L. Wang, X. Zheng, S. Zhou, and Z. Niu, HyCell: enabling GREEN base station operations in software-defined radio access networks, arXiv preprint arXiv:1503.06264, 2015.

[64] B. Yan, J. Zhou, J. Wu, Y. Zhao, Poster: SDN based energy management system for optical access network, in: Communications and Networking in China (CHINACOM), 2014 9th International Conference on, 2014, pp. 658–659.

[65] Y. Zhang, P. Chowdhury, M. Tornatore, B. Mukherjee, Energy efficiency in telecom optical networks, Commun. Surv. Tut. IEEE 12 (2010) 441–458.

[66] S.R. Chowdhury, M.F. Bari, R. Ahmed, R. Boutaba, Payless: a low cost network monitoring framework for software defined networks, in: Network Operations and Management Symposium (NOMS), 2014 IEEE, 2014, pp. 1–9.

[67] T.H. Vu, P.N. Nam, T. Thanh, N.D. Linh, T.D. Thien, N.H. Thanh, Power aware OpenFlow switch extension for energy saving in data centers, in: Advanced Technologies for Communications (ATC), 2012 International Conference on, 2012, pp. 309–313.

[68] H. Wang, Y. Li, D. Jin, P. Hui, J. Wu, Saving energy in partially deployed software defined networks, IEEE Trans. Comput. (2015).

[69] M. Zhang, H. Yu, Power-aware networking, in: Handbook of Research on Redesigning the Future of Internet Architectures, 2015, p. 96.

[70] R. Guerzoni, R. Trivisonno, I. Vaishnavi, Z. Despotovic, A. Hecker, S. Beker, et al., A novel approach to virtual networks embedding for SDN management and orchestration, in: Network Operations and Management Symposium (NOMS), 2014 IEEE, 2014, pp. 1–7.

Mehmet Fatih Tuysuz holds the B.*Sc*. degrees from İnönü University, department of Electric and Electronic Engineering and Anatolian University, department of Business Administration. He holds the M.*Sc*. degree from Dokuz Eylül University, department of Electric and Electronic Engineering. During his M.*Sc*. thesis, he worked on the "Quality of Service Enhancement of VoIP applications over wireless networks" and published several papers in this area. He joined Gebze Institute of Technology, department of Computer Engineering in 2008 as a Ph.D. student and graduated his Ph.D in 2013. Currently, he has been working at Harran University, Computer Engineering Department as an assistant professor. His interest includes VoIP, Wireless QoS, energy-aware communications and energy optimization in wireless networks.

Didem Gözüpek is an Associate Professor with the Computer Engineering Department, Gebze Technical University, Kocaeli, Turkey. She received the B.S. degree (high honors) in telecommunications engineering from Sabancı University, Istanbul, Turkey, in 2004, the M.S. degree in electrical engineering from the New Jersey Institute of Technology (NJIT), Newark, NJ, USA, in 2005, and the Ph.D. degree in computer engineering from Bogazici University, Istanbul, Turkey, in 2012. From 2005 to 2008, she worked as an R&D Engineer in a telecommunications company in Istanbul. Her main research interests are structural and algorithmic graph theory, approximation algorithms, and optimization problems in communication networks. Dr. Gözüpek received the CAREER Award from the Scientific and Technological Research Council of Turkey (TUBITAK) in 2014, the Dr. Serhat Özyar Young Scientist of the Year Honorary Award in 2013, and the Bogazici University Ph.D. Thesis Award in 2012. She was a finalist for the Google Anita Borg Memorial Scholarship in 2009.