# RESDN: A Novel Metric and Method for Energy Efficient Routing in Software Defined Networks

Beakal Gizachew Assefa, Öznur Özkasap, *Senior Member, IEEE* Department of Computer Engineering Koç University, Istanbul, Turkey {bassefa13, oozkasap}@ku.edu.tr

Abstract-Software-defined networking (SDN) paradigm, with the flexible and logically centralized control, enables dynamically minimizing the network energy consumption by redirecting paths of packets. However, the links and switches are designed to accommodate maximum traffic volume and their power consumption is not traffic proportional. Moreover, there exists a trade-off between energy efficiency and network performance that need to be considered together. Addressing these issues, we propose an energy efficiency metric named Ratio for Energy Saving in SDN (RESDN) that quantifies energy efficiency based on link utility intervals. We provide integer programming formulation and method for maximizing the **RESDN** of the network. To the best of our knowledge, **RESDN** approach is novel as it measures how links are profitably utilized in terms of the amount of energy they consume with respect to their utility. We analyze our approach considering various metrics of interest, and different types of SDN enabled switches. Experiments show that maximizing the RESDN value improves energy efficiency while maintaining acceptable network performance. In comparison to state-of-the-art utility-based heuristics, **RESDN** method achieves up to 30% better ratio for energy saving, 14.7 watts per switch power saving, 38% link saving, 2 hops decrease in average path length, 5% improved traffic proportionality.

*Index Terms*—Software-defined networks, SDN, Energy Efficiency, Ratio for Energy Saving, Routing, Heuristics, Traffic Proportionality.

#### I. INTRODUCTION

**S** OFTWARE Defined Networking (SDN) paradigm is based on the concept of forwarding (data) plane and control plane separation. The control and management of the network is done from a logically centralized controller, while the data plane handles the forwarding functionality. SDN provides the powerful feature of network programmability and has been widely adopted by major companies and network equipment vendors. The ease and flexibility of network control has paved the way to several network applications like load balancing, energy efficiency, dynamic routing, advanced security, and traffic engineering [1]–[8].

Global energy consumption has both environmental and economical issues. 10% of the global energy consumption is due to ICT sector out of which 2% is from network components. By 2020 the total electricity cost of cloud data centers is expected to increase by 63% [9], [10]. However, most of the time network resources are under utilized as compared to the service they render. In data center networks, components are utilized 30% to 40% most of the time [11], [12]. The overall power consumption however remains almost the same for varying amount of traffic volume. A practical solution for a traffic proportional energy consumption problem is to sleep/turn off under-utilized components for low traffic volume. Minimizing the number of network components for low traffic, leads to network performance degradation [13]– [18].

1

Energy saving at the cost of network performance degradation, however, is an undesirable result. The trade-off between energy saving and performance is quite a challenging task. Energy efficient solutions mainly focus on minimizing the number of energy consuming devices whereas network performance optimization solutions mainly focus on keeping the devices work at full capacity regardless of low traffic. The two opposing objectives clearly signifies the need for a holistic solution that maintains the trade-off between efficiency and performance.

In SDN, a naive way for measuring the efficiency of a network is to take the ratio of the power used to the maximum power consumption of the network components. A wide range of energy efficiency metrics like Power Usage Efficiency (PUE) have been proposed and used to measure the efficiency of data centers [19], [20]. The Green Grid and Standard Performance Evaluation Corporation (SPEC) are the most widely known efforts among the many. The metrics proposed, however, are either applicable to data centers only or do not consider the utilities of the network resources, and cannot directly be applied to dynamically changing and programmable networks.

Therefore, addressing the need for a metric to measure the energy efficiency of a network with regard to traffic volume and utility of resources, we propose a novel energy efficiency metric, namely RESDN (Ratio for Energy Saving in SDN), that quantifies energy efficiency based on utility intervals defined by minimum and maximum link utility parameters.<sup>1</sup>. The contributions of this work are as follows.

- We propose a novel energy efficiency metric RESDN: the ratio for energy saving in SDN that quantifies energy efficiency based on link utility intervals. To the best of our knowledge, the approach is unique as it measures how links are profitably utilized in terms of the amount of energy they consume with respect to their utility.
- We develop Integer Programming (IP) formulation with the objective of maximizing RESDN and propose heuristics algorithm MaxRESDN for achieving the objective.
- We conduct extensive experiments on Mininet network emulator and POX controller using real network traffic traces and present comparative quantitative analysis of MaxRESDN method. The performance metrics of interest are switch power consumption, RESDN value, percentage of links saved, average path length, throughput, delay, and traffic proportionality.
- We simulate the power consumption of two hardware switches (NEC and Zodiac FX) and one virtual switch (Open vSwitch-OvS) that are OpenFlow enabled, and conduct comprehensive experiments to measure the power consumption of our proposed MaxRESDN heuristic algorithm.
- MaxRESDN heuristic algorithm achieves the highest ratio for energy saving which is up to 30% better than similar state-of-the-art utility-based heuristics algorithms. It also saves up to 38% links and exhibits an average path length closer to the best energy saving heuristics that focuses on performance. MaxRESDN has the highest traffic proportional energy consumption which 3 to 5% better than similar utility-based heuristics. To the best of our knowledge, our approach is the first in maintaining the trade-off between energy efficiency, network performance, and traffic proportionality by optimizing RESDN metric.
- Switch power consumption results show that MaxRESDN exhibits on average up to 14.7 watts, 10 watts, and 3.2 watts less power consumption for NEC, OVS and Zodiac FX switches respectively as compared to other utility-based heuristics for energy efficient routing.
- As the utility parameters directly impact the performance of the MaxRESDN heuristics, we conduct a detailed analysis of the parameters with respect to a range of traffic volumes.

The remainder of the paper is organized as follows. Section II presents related work on energy efficiency in SDN and energy efficiency metrics. Section III presents preliminary work on utility based traffic proportional energy saving in SDN. Section IV describes RESDN metric, provides IP formulation for RESDN maximization, and presents the MaxRESDN heuristics method that maximizes the RESDN of a network. Section V discusses the platform used for our

experiments along with heuristics used for comparison and the characteristics of the hardware switches used. Section VI presents our comprehensive experimental results. Section VII concludes and states future directions.

#### II. RELATED WORK

The control and forwarding plane separation in SDN has led to flexibility for many network services ranging from load balancing, dynamic routing, energy efficient and flexible network control [22]. The primary challenge of energy efficiency in networking is the fact that energy consumption is not proportional to the volume of traffic. The problem is even more difficult in the traditional networking since there is a very limited flexibility. Traffic-aware energy efficient routing techniques in SDN attempt to make energy consumption proportional to the traffic volume. We discuss related work in traffic-aware energy efficient routing in subsection II-A. Measuring the energy efficiency of a network environment including data centers have been investigated and various metrics were also proposed. In subsection II-B, we discuss related work in this area and why we needed a new metrics for SDN.

#### A. Traffic Aware Energy Efficient Routing Techniques

Traffic aware energy efficient routing techniques in SDN can be classified based on the energy saving capabilities they focus on and the topology structure they are designed for. The main energy saving capabilities are the links and the switches. Some of the works [14]–[18] focus on links, where some others focus on forwarding switches [23], [24]. There also exist works [13], [25], [26] that consider the combination of links and switches as the energy saving components. Furthermore, queue engineering based techniques consider the arrival of packets and their waiting times to decide perport power requirement [16]–[18], [24], [27]–[29].

Based on the topology assumption, energy efficient methods can be classified as general or tailored to a specific network topology. Some methods are general in the sense that they are applicable to any kind of network topology [25], [26], [30] and some are only applicable to specific topologies such as Fat-tree, butterfly, and BCube, where Fat-tree and BCube topologies are among the widely used structures used in organizing end systems in data centers [12], [13], [15], [31].

Table I presents the summary of traffic aware energy efficient techniques in SDN. The Topology column shows the kind of topology structure the approach is designed for and tested on. The Utility-Based column indicates if the approach focuses on the utility of the links. The Link column shows if the method considers links as energy saving component. Queue Engineering column shows if the approach applies queue engineering techniques.

Energy efficient techniques use a method of sleeping or turning of unused switches or links when the traffic volume is low and turn them on when traffic volume increases [15], [15], [28], [38]. The objective of energy efficiency is reducing

<sup>&</sup>lt;sup>1</sup>A very preliminary version of this work was presented in [21].

 TABLE I

 CLASSIFICATION OF ENERGY EFFICIENT ROUTING TECHNIQUES IN SDN

Annuagah	Tanalagy	Utility	Link	Queue	
Approach	Approach Topology		LIIK	Engineering	
ElasticTree [13]	Fat-tree	-	$\checkmark$	-	
Carrier Grade [12]	General	-	$\checkmark$	-	
CARPO [14]	Fat-tree	-	$\checkmark$	-	
EnableOpenflow [15]	Fat-tree	-	$\checkmark$	-	
Traffic distribution [32]	General	$\checkmark$	$\checkmark$	-	
GreenSDN [16]	Bcubic & Fat-tree	-	$\checkmark$	$\checkmark$	
Fine-grained [33]	General	$\checkmark$	$\checkmark$	-	
OpenNaas [17]	-	-	$\checkmark$	$\checkmark$	
Orchestrate [18]	-	-	$\checkmark$	$\checkmark$	
Utility-based [26]	General	$\checkmark$	$\checkmark$	-	
Dynamic TA [25]	General	-	$\checkmark$	-	
REsPoNse [23]	-	-	-	-	
NetFPGA QE [24]	-	-	-	$\checkmark$	
GreenRE [27]	General	-	$\checkmark$	-	
Bandwidth-aware [28]	Bcubic & Fat-tree	- 🗸	-	$\checkmark$	
RA-TAH [31]	Fat-tree	-	$\checkmark$	$\checkmark$	
TE based [34]	General	-	$\checkmark$	-	
Re-Routing [35]	General	-	-	$\checkmark$	
Resource-aware [36]	Fat-tree	-	$\checkmark$	-	
SDN /Ethernet [37]	General	-	$\checkmark$	-	

the number of active network components. However, this has an adverse effect on the performance of the network hence degrades the network performance. There is a trade-off between the two opposite objectives which are minimizing energy consumption and increasing performance.

A closely related work considers the remaining bandwidth of links [28]. After formulating the problem using IP, they propose a scheduling algorithm. The heuristics proposed schedules paths for the new flows by considering the paths followed by the preceding flows. Unlike this approach, our focus is not on the remaining bandwidth of the links but the utilities of the links. We measure how much each link is utilized based on the utility interval defined by the minimum and maximum utility parameters.

In contrast to the approach of turning off links for energy saving [14], [26], [27], [39], the work [32] examines if turning off / sleeping switches always result in energy saving. The network is modeled as a graph with the nodes as the switches and the edges as the links. Comprehensive experiments show for cases where the routing power consumption is cubic with respect to traffic volume, distributing the traffic to underutilized links exhibit a better energy saving as compared to turning them off. Since the routing power consumption is polynomial in general, distributing traffic over underutilized links is only applicable to specific cases [32].

The other approach close to our work is [25] which formulates the energy efficiency problem with mixed integer programming (MIP), then proposes heuristic algorithms. There are four variations of the heuristics where two of which namely Shortest Path First (SPF) and Shortest Path Last (SPL) sort the flows in the order of the shortest path. The other two heuristics namely Smallest Demand First (SDF) and Highest Demand First (HDF) sorts the flows according to the rate flow.

In contrast, our approach is flow order independent, it focuses not only on energy efficiency but also in performance, considers the utility of each link, and proposes a link utilitybased energy efficiency metric that simultaneously quantifies efficiency and performance.

#### B. Energy Efficiency Metrics

Data center energy efficiency metrics have been studied widely with companies and standardization organizations. Some of these are the Green Grid, Standard Performance Evaluation Corporation (SPEC), and Transaction Processing Performance Council (TPC) [40], [41]. The Green grid consortium proposes several metrics to measure infrastructure power consumption, carbon dioxide emission, water usage efficiency, and the rate of useful information processed related to resources used. Power Usage Effectiveness (PUE), Data Center Infrastructure Efficiency (DCIE), Carbon Usage Effectiveness (CUE), Water Usage Effectiveness (WUE), and Data Center Productivity (DCP), all of which measure energy efficiency and sustainability.

Power Usage Effectiveness (PUE), the most prevalent metric, is the ratio of the total annual amount of energy that comes into a data center to the energy that is used by IT equipment [42]. IT equipment includes workstations, storage, monitors, and network devices. Total data center energy consists of IT equipment energy consumption plus energy for cooling system components, UPS, switch gear, and data center lightening. The ideal value for PUE is 1 which means all the energy in the data center is used by the IT equipment. The main limitations of PUE metric is that it is only applicable to a single building that supports a data center [39]. The other limitation of PUE is that it does not focus on specifically on links and switches which are the main energy saving capabilities in SDN.

Data Center Infrastructure Efficiency (DCIE), which is the inverse of PUE, is calculated as the ratio of IT equipment power consumption to total facility power consumption. Carbon Usage Effectiveness (CUE) is the product of the amount of carbon dioxide emitted per kilowatt hour (CEF) and the data center's annual PUE [43]. Water Usage Effectiveness (WUE) is the ratio of the annual site water usage in liters to the IT equipment energy usage in kilowatt hours (Kwh) [44]. WUE measures the water consumption in relation to IT equipment energy consumption. However, it is only applicable to a single data center site and does not consider network devices which are the main focus of energy saving in SDN. Data Center Productivity (DCP) measures the quantity of useful information processing completed relative to the amount of some resource consumed in producing the work [19]. DCP treats the data center as a black box where an power enters in to the data center do some useful task and leaves. An important concern in this is how to quantify a useful work.

To the best of our knowledge, there is no metric that is fully focused on network components such as switches and links which are energy saving capabilities in SDN. Using the existing metrics in an SDN environment falls short of capturing the utilities of links which is an important energy saving capability of a network environment. The other drawbacks in the metrics are that they are measured annually and offline. It makes the necessity for new metrics that can measure dynamically changing network environment an indisputable argument. Failing to do so undermines the flexible network management capability that could have been exploited from SDN.

#### **III. PRELIMINARIES**

Next Shortest Path (NSP) and Next Maximum Utility (NMU) heuristics were proposed in our prior work [26] that are generally applicable to any network topology and are flow order independent. NSP and NMU first select those links with utility less than Umin as the candidate links, then aim at redirecting flows passing through the candidate links to the next shortest path or the path in the direction of the link with maximum utility, respectively. The rationale behind selecting links with utilities less than Umin as a candidate is to redirect flows that pass through these links to make them inactive. NSP and NMU attempt to put underutilized links to inactive state. By doing so not only does the number of active links is reduced, but also overall utilities of the links increase. Next, we discuss the NSP algorithm in detail as the preliminary, since it attempts to make every link to be utilized above the minimum utility Umin or to be made inactive.

NSP (Algorithm 1) identifies under-utilized links and reroutes traffic flows passing through them to the next shortest alternative path. The algorithm takes topology as a graph  $\mathbb{G}$ , set of flows  $\mathbb{F}$ , values Umin, and Umax as inputs. It first initializes the link utilities by taking the current utilities of the links, and then selects the under-utilized links and sets the CandidateList. Line 6 returns the list of paths from  $Z_i$  to  $Z_i$  excluding those links in the CandidateList. The paths stored in  $shortest paths_{ij}$  are sorted according to the path length. The  $shortest paths_{ij}$  is set to  $Path_{ij}$  (line 7). NSP algorithm then picks one of the shortest paths SP to replace the direct link. SP is replaced with the next  $Path_{ii}$ until all the links in  $\forall e_{ab} \in SP, U_{ab} + U_{ij} \leq Umax$  (lines 8 to 10). Replacing a direct link with alternative path needs redirecting the flows passing through it. Basically, the utilities of the hops composing SP are increased and the utility of the direct link are decreased (lines 13 and15). The status of a link is updated if the utility is 0 (all the flows passing are redirected to alternative path SP), hence updates the graph G's status (lines 17-21). Algorithm 1 returns the updated sub graph of  $\mathbb{G}$ .

## IV. RESDN METRIC: RATIO FOR ENERGY SAVING IN SDN

In this section, we propose RESDN energy efficiency metric based on link utility interval defined by the minimum and maximum utility parameters Umin and Umax. Then, we present IP formulation and heuristics to maximize the RESDN value of a network in a dynamic environment.

#### A. Motivation and Definition

Traffic aware energy saving approaches attempt to turn off/sleep network components by observing the traffic volume. Traffic proportional energy consumption is a more

#### Algorithm 1 NSP: NextShortestPath( $\mathbb{G}$ , $\mathbb{F}$ , Umin,Umax)

- 1: **Input:** Graph  $\mathbb{G}$ , set of traffic flow  $\mathbb{F}$ , minimum utility Umin, and maximum utility Umax
- 2: **Output:** Sub graph of  $\mathbb{G}$
- 3:  $\mathbb{U} \leftarrow NetworkStatus(\mathbb{G}, \mathbb{F})$
- 4: CandidateList = ReturnCandidateList(U, Umin)
- 5: for all  $e_{ii} \in CandidateList$  do  $Path_{ij} \leftarrow shortest path_{ij}$ 6:  $SP = Path_{ij}[0] \quad \triangleright$  Pick one of the shortest paths 7: 8: while  $\exists e_{ab} \in SP$  where  $U_{ab} + U_{ij} > Umax$  do  $SP = next(Path_{ij})$ 9: end while 10: for all f passing through  $e_{ij}$  do 11: for all  $e_{ab} \in SP$  do  $U_{ab} = U_{ab} + \frac{\lambda_f}{W_{ab}}$ end for  $U_{ij} = U_{ij} - \frac{\lambda_f}{W_{ij}}$ 12:  $\triangleright$  increment  $U_{ab}$ 13: 14:  $\triangleright$  decrement  $U_{ij}$ 15: end for 16: if  $U_{ij} == 0$  then 17:  $L_{ij} = 0$ ▷ state of link is inactive 18: 19: Turn off  $(e_{ij})$ Update G 20: end if 21: 22: end for

stricter requirement than traffic-awareness. It demands the percentage of power consumption of network components to be proportional to the traffic volume. However, none of the existing energy efficiency metrics discussed in Section II take the utilities of the network components under consideration with regard to the network traffic volume. Addressing this issue, we propose Ratio for Energy Saving in SDN (RESDN) metric that indicates the percentage of links with utilities within the interval defined by *Umin* and *Umax* with respect to the total number of active links.

The RESDN value reflects how the network components are profitably utilized. The current mode of payment for customers in cloud computing is Pay-as-you-go (PAYG) which charges based on the usage of computing resources. Similarly, RESDN enables the network provider to implement Pay-as-you-use (PAYU) in SDN for energy consumption. A simple scenario, in this case, is a network provider aiming to pay the energy cost of a network component only if it is utilized within a utility interval expressed in terms of RESDN. This RESDN policy is expressed by Umin and Umax parameters and are enforced by the controller. The controller decides the links and their corresponding utilities to match the utility interval specified by the provider. The technical solution to this is to redirect the flows passing through underutilized links to other links. Especially in a dynamic environment where the traffic volume increases and decreases, the RESDN value changes, the controller is responsible for keeping the RESDN value at maximum. With this, the energy cost of the network provider would be proportional to the traffic volume being streamed in the network at any time.

The RESDN metric enables to measure the energy efficiency of the network by calculating the ratio of the number of links with utilities between the Umin and Umax to the total number of active links. Thus, it allows the network provider to proactively set the RESDN value.

Table II shows the parameters used in the formulation of the RESDN. The network is modeled as a weighted graph  $\mathbb{G}=(\mathbb{Z},\mathbb{E})$  with  $\mathbb{Z}$  as the set of switches where  $Z_i \in \mathbb{Z}$  represents switch *i* and  $e_{ij} \in \mathbb{E}$  represents that there exists a link between switches  $Z_i$  and  $Z_j$ . The weight  $W_{ij}$  corresponds to the bandwidth of the link  $e_{ij}$ .

Let binary variable  $S_i$  denote the status of switch  $Z_i$  such that

$$S_i = \begin{cases} 1, & \text{if switch } Z_i \text{ is active} \\ 0, & \text{otherwise} \end{cases}$$

Traffic in the network is represented by the set of flows  $\mathbb{F}$ where  $f \in \mathbb{F}$  is defined as  $f=(sr, ds, \lambda_f)$ . sr and  $ds \in \mathbb{Z}$  are the source and destination switches of flow f, and  $\lambda_f$  is the rate of f measured in bits per second.

$$F_{ij} = \begin{cases} 1, & \text{if flow } f \text{ passes through edge } e_{ij} \\ 0, & \text{otherwise} \end{cases}$$

 $\mathbb U$  is the set of the utilities of every edge in the graph  $\mathbb G$  where  $U_{ij}\in\mathbb U$ 

$$U_{ij} = \frac{\sum_{\forall f} F_{ij} x \lambda_f}{W_{ij}} \tag{1}$$

is defined as the ratio of the sum of the rates of the flows passing through the edge  $e_{ij}$  to the link bandwidth  $W_{ij}$ . Utility of a link is between 0 and 1, where 0 means no flow is passing through the link and 1 means the sum of the flow rates passing through the link is equal to the link bandwidth. Let Umin be the minimum utility value to keep a link active and Umax is the maximum utility of a link.

$$X_{ij} = \begin{cases} 1, & Umin \le U_{ij} \le Umax \\ 0, & \text{otherwise} \end{cases}$$

The energy efficiency metric is formally defined as

$$RESDN = \frac{\sum_{\forall e_{ij}} X_{ij}}{\sum_{\forall e_{ij}} L_{ij}}$$
(2)

where  $L_{ij}$  is binary variable that denotes the status of edge  $e_{ij}$ 

$$L_{ij} = \begin{cases} 1, & \text{if edge } e_{ij} \text{ is active} \\ 0, & \text{otherwise} \end{cases}$$

According to Equation 2, RESDN value of 1 means that all links that are turned on are operating between the interval defined by parameters *Umin* and *Umax*. RESDN value of 0 means that none of the links are operating profitably which means that the network is underutilized. The motivation behind the RESDN comes from the idea that network providers want to pay for energy consumption for a given resource only if it is utilized at least to a minimum value of Umin. If the utility of a resource is less than Umin, then it is underutilized. Umax needs to be less than 1 since setting it to 1 would create network congestion which in turn degrades performance. The larger the value of RESDN, the more the percentage of the active links that operate between the Uminand Umax values. Smaller RESDN value shows that a large number of active links operating unprofitably.

#### B. IP formulation for RESDN Optimization

9

The objective function of our IP model is to maximize the RESDN value of the network.

maximize 
$$RESDN$$
 (3)

subject to 
$$\sum_{\forall f} F_{ij} x \lambda_f \le W_{ij}$$
,  $\forall e_{ij}$  (4)

$$\sum_{\forall f} F_{ai} = \sum_{\forall f} F_{ib} , \ Z_i \neq f^{sr}, Z_i \neq f^{ds}$$
(5)

$$F_{mj} = F_{in}$$
,  $\forall f \ Z_m = f^{sr}$ ,  $Z_n = f^{ds}$ ,  $\forall e_{mj}$ ,  $\exists e_{in}$  (6)

$$F_{ij} \leq S_j \text{ and } F_{ji} \leq S_j \text{ , } \forall Z_j \in \mathbb{Z}$$
 (7)

$$S_i \le \sum_{\forall f} [F_{ij} + F_{ji}], \ \forall Z_i \in \mathbb{Z}$$
 (8)

$$L_{ij} \le S_i \text{ and } L_{ij} \le S_j \ \forall Z_i, Z_j \in \mathbb{Z}$$
 (9)

The constraint in Equation 4 states that the sum of the rates of flows between two switches should not exceed the link capacity. The constraint in equation 5 states that the number of flows entering and leaving switches which are neither destination nor sources of flow should be equal. The constraint in equation 6 assures a flow entering from source switch should reach the destination switch. The constraint in 7 asserts that a flow should not be assigned to a link that is connected to an inactive switch. Constraint in equation 8 models the relationship between the flows passing through a link and a switch. It asserts that if no flow is passing through all the links connected to switch  $Z_i$ , then change the binary variable  $S_i = 0$ . The constraint in Equation 9 asserts that a link should be put in active state if and only if both of the switches it is connecting are active, otherwise, it is inactive.

#### C. MaxRESDN Heuristics Method

The energy efficient routing is an NP-hard problem [13], [14], [37]. Formal solutions can best fit for a small number of network switches but fail to scale up when the number is in the order of hundreds. However, in cloud data centers where the number of physical machines is in the order of thousand, the number of switches is in the order of hundreds. That triggers the need for heuristics algorithms that can run in reasonable polynomial time to provide sub-optimal solutions.

We propose the heuristics algorithm named MaxRESDN (Maximize RESDN). Algorithm 2 aims to maximize the RESDN value of the network as it assigns a path to each flow.

TABLE II TABLE OF SYMBOLS

Parameters	Description
G	A graph that consists of $\mathbb{Z}$ and $\mathbb{E}$
$\mathbb{Z}$	Set of switches $Z_i \in \mathbb{Z}$ represents switch $i$
$S_i$	Binary variable of switch $Z_i$
$\mathbb{E}$	Set of links $e_{ij} \in \mathbb{E}$ is a link between switches $Z_i$ and $Z_j$
$L_{ij}$	Binary variable of link $e_{ij}$ (active or inactive)
$W_{ij}$	Bandwidth of the link $e_{ij}$
$\mathbb{F}$	Set of flows $f \in \mathbb{F} = (sr, ds, \lambda_f)$
f	A flow $f=(sr, ds, \lambda_f)$ with source , destination and flow rate
$F_{ij}$	Binary variable of flow f passing through link $e_{ij}$
U	Set of utilities of all links
$U_{ij}$	Utility of link $e_{ij}$
$X_{ij}$	Binary variable of link $e_{ij} \ Umin \le U_{ij} \le Umax$

Algorithm 2 MaxRESDN (G, F,U,Umin,Umax)	Algorithm 3 PathMaxRESDN(sr,ds, $\lambda_f$ )
1: Input: Graph G, set of traffic flow F, utility of links U, minimum utility $Umin$ , and maximum utility $Umax$ 2: Output: Modified utility of links U and graph G 3: for all $f = (sr, ds, \lambda_f) \in F$ do 4: $path_f \leftarrow PathMaxRESDN (sr, ds, \lambda_f)$ 5: for all $e_{ij} \in path_f$ do 6: $U_{ij} \leftarrow U_{ij} + \frac{\lambda_f}{W_{ij}}$ 7: end for 8: end for 9: for all $e_{ij} \in \mathbb{E}$ do 10: if $U_{ij} == 0$ then 11: $L_{ij} \leftarrow 0$ 12: end if 13: end for	1: Input: Graph G,utility of links U, minimum utility Umin, maximum utility Umax, source node sr, destination node ds, and flow rate $\lambda_f$ 2: Output: path <sub>f</sub> for flow f(sr, ds, $\lambda_f$ ) that maximizes RESDN         3: AllPath $\leftarrow$ all paths between sr and ds         4: for all Path P $\in$ AllPath do         5: for all links $e_{ab} \in$ P do         6: if $U_{ab} + \lambda_f > UMax$ then         7: AllPath.remove(P)         8: end if         9: end for         10: end for         11: Path <sub>f</sub> $\leftarrow$ MaxRESDN(Allpath)

The inputs for the MaxRESDN are the network represented as graph  $\mathbb{G}$ , the current utility of links  $\mathbb{U}$ , set of flows  $\mathbb{F}$ , the minimum utility Umin and maximum utility Umax. Each flow  $f = (sr, ds, \lambda_f) \in \mathbb{F}$  is expressed as source sr, destination ds, and flow rate  $\lambda_f$  which correspond to source address, destination address, and rate of f measured in bits per second (line 3). Line 6 increments the utilities of all the links on the path assigned to flow f by the  $\frac{\lambda_f}{W_{ij}}$  factor. Line 4 assigns each flow to the path that maximizes the RESDN among alternatives using the PathMaxRESDN(sr,ds, $\lambda_f$ ) method. Lines 9 to 13 makes link  $e_{ij}$  inactive if its corresponding utility  $U_{ij}$  is 0.

Algorithm 3 (PathMaxRESDN) returns the path that maximizes RESDN for a given flow f with source sr, destination ds, and flow rate  $\lambda_f$ . Line 3 initializes the *Allpath* list with all paths from sr to ds of flow f. Lines from 5 to 9, ensure the stability of the network by making the utilities of all links in each path plus the rate of flow of f not to exceed the *Umax* value. While line 11 selects the path with maximum RESDN value and assign it to Path<sub>f</sub>.

#### V. EXPERIMENTAL PLATFORM

This section presents the platform used to conduct our experiments. First, the simulation setup and the performance metrics are discussed, then utility based heuristics algorithms used for comparison are presented. Finally, the characteristics of the three OpenFlow protocol enabled switches we used in the experiments for measuring power consumption are presented.

#### A. Setup and Performance Metrics

The experimental platform is constructed using POX controller and Mininet [45] network emulator installed on Ubuntu 16.04 64-bit. The network topology is created on Mininet, and the heuristics are implemented on POX controller. Our experiments are conducted using real traces from SNDlib [46], in particular the GEANT dynamic traffic trace of the European research network.

Figure 1 shows the GEANT topology used in our experiments [46]. The number of nodes and bidirectional links are 22 and 36, respectively. Traffic demand matrices are



Fig. 1. GEANT network topology a) Map view, b) Graph view

aggregated for 15 minutes from a 4-month duration trace. There exists total 11460 traffic demand matrices. The number of flows in each traffic demand matrix ranges from 82 to 462. To show traffic awareness in our experiments, different traffic volumes ranging from 20% to 90% with respect to the network capacity are used.

TABLE III TABLE OF METRICS AND MEASUREMENT UNITS

Metric	Unit / Description		
Average Power Consumption of Switches	Watt		
RESDN	metric as percentage		
Links saved	percentage of links saved		
Average path length	#hops		
Throughput	Mbits/sec		
Delay	milliseconds		
Traffic Proportionality	[0,1]		

Table III presents the performance metrics of interest measured in our experiments, namely average power consumption of switches, RESDN value, percentage of links saved, average path length, throughput, delay, and traffic proportionality. The network performance is analyzed in terms of average path length, throughput and delay.

$$P_{switch} = P_{base} + P_{config} + P_{control} \tag{10}$$

Equation 10 shows the total power consumption of a switch ( $P_{switch}$ ) as the sum of  $P_{base}$ ,  $P_{config}$  and  $P_{control}$  [47].  $P_{base}$  is the power consumption for keeping the switch on without any active ports. The configuration power consumption  $P_{config}$  is calculated as

$$P_{config} = \sum_{i}^{N_{activePorts}} c_i . P_{port}$$
(11)

where  $c_i$  is the percentage of the maximum line speed of the port and  $P_{port}$  is the power consumption of a port at full capacity measured in watt.

Equation 12 shows the power consumption of control  $P_{control}$  where  $r_{PacketIn}$  and  $E_{PacketIn}$  are the rate and energy consumption of PacketIn.  $r_{FlowMod}$  and  $E_{FlowMod}$  are the rate and energy consumption of the FlowMod operations.

$$P_{control} = r_{PacketIn} x E_{PacketIn} + r_{FlowMod} x E_{FlowMod}$$
(12)

The ratio of energy saving metric for SDN is calculated as

$$RESDN = \frac{\sum_{\forall e_{ij}} X_{ij}}{\sum_{\forall e_{ij}} L_{ij}}$$

where  $L_{ij}$  is binary variable that denotes the status of edge  $e_{ij}$  and  $X_{ij}$  is a binary variable that tells if the utility of the link is between Umin and Umax. Although, our primary goal is energy saving, RESDN also captures performance by the Umax value.

The percentage of links saved is calculated as

Links saved = 
$$100(1 - \frac{\sum_{\forall e_{ij}} L_{ij}}{|\mathbb{E}|})$$
 (13)

The average path length in terms of average number of hops is calculated as

Average path length = 
$$\frac{\sum_{\forall f} \sum_{\forall e_{ij}} (F_{ij})}{|\mathbb{F}|}$$
 (14)

Throughput is calculated as the amount of data transferred per unit of time. In Mininet, we use Iperf command to measure the throughput of source and destination pairs. In the experiments, we measure the average throughput of all source and destination pairs.

Delay is calculated as the amount of time needed for data packets to be transferred from their source to destination. In our experiments, network delay is measured as the time it takes for the packets of a flow to start at the source and reach at the destination node. The delay is measured both on the Mininet side and POX controller.

Traffic proportionality is defined as the ratio of the traffic volume percentage to the network power consumption and is computed as

$$\left(\frac{\frac{\%\text{Traffic Volume}}{M\frac{\sum_{\forall Z_i} SP_i}{|\mathbb{Z}|} + N\frac{\sum_{\forall e_{ij}} L_{ij}}{|\mathbb{E}|}}\right)(M+N) \tag{15}$$

where M:N is the ratio of the power consumption of the switches to the links. In these experiments, the ratio M:N is used as 3:1 [26].

#### B. Algorithms Used for Comparison

The heuristics algorithms we used to compare with MaxRESDN heuristics are listed in Table IV. The methodology in [25] formulates the problem with MIP and proposes four heuristic algorithms, namely Shortest Path First (SPF), Shortest Path Last (SPL), Smallest Demand First (SDF), and Highest Demand First (HDF). The first flow is assigned its corresponding shortest path, the succeeding flows are assigned paths where the change in energy consumption is minimized. Each heuristics algorithm uses a different criteria to sort the flows and processes them in the corresponding order. The four variations of the algorithms first sort the flows according to the shortest path first, shortest path last, smallest demand first, and highest demand first.

As explained in Section III, the objective of NSP algorithm is to re-route flows passing through the under-utilized links to the next shortest path. NMU, on the other hand, chooses the path that has the link with maximum utility. While NSP gives priority to performance, NMU focuses on maximizing the utility of active links. Both NSP and NMU are not only ordering independent but can also be applied on top of other algorithm outputs to improve efficiency. The B heuristics is the application of NSP or NMU algorithms on top of the results of SPF, SPL, SDF, or HDF algorithms. It refers to the best outcome (B) in terms of energy efficiency.

TABLE IV TABLE OF HEURISTICS

Abbreviation	Description
SPF [25]	Shortest Path First
SPL [25]	Shortest Path Last
SDF [25]	Smallest Demand First
HDF [25]	Highest Demand First
NSP [26]	Next Shortest Path
NMU [26]	Next Maximum Utility
B [26]	Best Combination of NSP (NMU) with others
MaxRESDN	Maximize RESDN

Table V shows the values of *Umin* and *Umax* parameters used in the experiments for different percentage of traffic volumes. The values that maximize the number of links saved are determined by a greedy-based approach. A detailed analysis of these parameters is presented in Section VI-D.

TABLE V PARAMETERS OF Umin and Umax Used

Traffic volume percentage	20	30	40	50	60	70	80	90
Umin	31	28	30	25	20	19	15	12
Umax	82	85	90	90	92	95	95	95

#### C. Types of Switches

Table VI shows the parameters used in the power calculation of two hardware switches (NEC PF 5240 and Zodiac FX) and one virtual switch Open vSwitch (OvS) which are OpenFlow enabled. NEC PF 5240 is a hybrid switch technology that adds OpenFlow protocol to the traditional network functionality [47], [48]. Open vSwitch (OvS) is a multilayer virtual switch designed for enabling network automation by supporting various protocols including but not limited to OpenFlow [47], [49]. Zodiac FX is an OpenFlow switch designed for small scale uses [50].

 TABLE VI

 Switch Power Consumption parameters

Parameters	NEC PF 5240 [47], [48]	OvS [47]	Zodiac FX [50]
Base[W]	118.33	48.7397	15
$P_{port}[W]$	0.52	Na	0.15
$\vec{E}_{PacketIn}$ [ $\mu$ W/packet]	711.30	775.53	775.53
$E_{FlowMod}$ [ $\mu$ W/packet]	29.25	356.743	1455.13

#### VI. EXPERIMENTAL RESULTS

This section presents our extensive experimental results in terms of switch power consumption, RESDN and energy efficiency, network performance, and analysis of the link utility interval parameters.

#### A. Switch Power Consumption Results

Figure 2 shows the average power consumption of the switches for the NEC PF 5240 OpenFlow-based switch versus traffic volume. As compared to the best combination (B) of NSP or NMU with other algorithms (Table IV), MaxRESDN power consumption is on average 6 to 9 watts less than the B algorithm. As compared to NSP and NMU [26], MaxRESDN has shown up to 14.7 and 10.7 watts less in power consumption, respectively. This indicates that through maximizing the RESDN parameter, improvements in energy savings are achieved for all traffic volumes.



Fig. 2. Power Profile for NEC PF 5240 switch in watt vs a range of traffic volume from 20% to 90%

Figure 3 shows the power consumption of Zodiac FX switch for NSP, NMU, B, and MaxRESDN algorithms versus traffic volume. Similar to the previous experiments, the average power consumption of switches is proportional to the traffic volume. However, power consumption of MaxRESDN is on average 3.2, 2.3, and 2 watts less than NSP, NMU and B algorithms, respectively.



Fig. 3. Power Profile for Zodiac FX switch in watt vs a range of traffic volume from 20% to 90%

Figure 4 shows the power consumption of OvS switches versus traffic volume. Results show that as the traffic volume increases, the average power consumption of switches also increases. The MaxRESDN algorithm exhibits 5.8, 10, and 8.9 watts less energy consuming as compared to B, NSP, and NMU respectively.

Overall, Figures 2, 3, and 4 show that MaxRESDN is the most energy efficient as compared to the other algorithms (Table IV). Our findings indicate that maximizing the RESDN value helps minimizing the energy consumption. This is mainly because enforcing links to operate within the minimum and maximum utility parameters minimizes the number of active ports.



Fig. 4. Power Profile for OvS switch in watt vs a range of traffic volume from 20% to 90%

### B. RESDN and Energy Efficiency Results

Figure 5 shows the RESDN values of all heuristic algorithms as the traffic volume increases. The MaxRESDN algorithm achieves the highest RESDN value which is 22% better for low traffic and 10% better than others for high traffic volume. This is because unlike the other algorithms whose objective is to minimize the number of links used in the network, MaxRESDN algorithm aims at maximizing the RESDN value, which increases the utility. In particular, for low traffic, it is more likely for flows that pass through overutilized links to be redirected to underutilized links.



Fig. 5. RESDN Values NSP,NMU, SPF, SPL, SDF, HDF, B and MaxRESDN heuristic algorithms versus a traffic volume ranging from 20% to 90%

Figure 6 shows the energy saving in terms of the percentage of links put to sleep. The B heuristics which is the best combination of the NSP or NMU algorithm exhibits the highest energy saving, while NSP and NMU have the lowest energy saving. MaxRESDN heuristics saves 38% for 20% traffic volume and 6.5% for 90% traffic volume.



Fig. 6. The percentage of links saved versus traffic volume ranging from 20% to 90%

Figure 7 presents the power consumption of all algorithms for the Zodiac switch. The results show that MaxRESDN switch power consumption is 16 to 22 watts which is on average 4 watts less than the SDF algorithm. MaxRESDN also demonstrated a power consumption on average 2 watts less than the B algorithm. These results show that by maximizing the RESDN value which is based on link utilities, by doing it frees ports on switches, hence it manages to reduce the power consumption of switches.



Fig. 7. The average power consumption of the Zodiac switch measured in watts versus traffic volume ranging from 20% to 90%

#### C. Network Performance Results

Figure 8 shows the average path length (in terms of the number of hops) versus the traffic volume, and is calculated as given in equation 14. The NSP heuristics achieves the best path length since its objective is redirecting flows to the next alternative shortest path. The decisions are made periodically by selecting the underutilized links as candidates then redirecting the flows. B heuristics which is the combination of NSP and the best of SPF, SPL, SDF, and HDF in terms of energy saving shows improvement in the average path length. This indicates that our previous heuristics still can be applied on top of other algorithms and improve both performance and energy saving. The MaxRESDN heuristics performance is closer to NMU that has the objective of maximizing the utilities of the links, by directing the flows passing through underutilized links to more utilized links.

The experimental results show that NSP and NMU have the lowest average path length but least energy saving, since they first consider performance rather than energy efficiency. The other algorithms in [25] exhibit better energy saving than NSP and NMU but worst average path length. This is because their objective is energy saving primarily. However, applying NSP and NMU on top of them increases energy saving. NSP and NMU balance the trade-off between performance and energy saving. The MaxRESDN heuristics, as compared to the other heuristics, has the maximum RESDN value. By maximizing a single RESDN value, the MaxRESDN has shown closer results to the best combination algorithm in terms of both energy saving and average path length.



Fig. 8. Average path length in terms of number of hops versus traffic volume ranging from 20% to 90%

Assuming that the optimal value for *Umin* and *Umax* parameters is found for a given traffic, the MaxRESDN heuristics gives the maximum RESDN that achieves traffic proportional energy consumption and keeps the trade-off between energy efficiency and performance.



Fig. 9. Throughput of heuristic algorithms in Mbps versus a range of traffic volume from 20% to 90%

Figure 9 shows the throughput of the algorithms in Mbps with respect to the volume of traffic that ranges from 20% to 90%. The throughput measurement is done on Mininet using the iperf command. The average bandwidth of links is set to 100 Mbps and the average flow rate is 7.79 Mbps. Results show that the NSP algorithm exhibits the highest throughput for all the traffic range and is followed by NMU and MaxRESDN. This is because, for the three heuristics, the algorithms are initialized with the shortest paths. The maximum RESDN value by MaxRESDN as shown in figure 5 for all the traffic volume exhibits the least power consumption and also throughput close to that of NSP and NMU. For

all the algorithms, the throughput decreases slightly as the traffic size increases after 30%. Under normal conditions where energy saving is not applied, throughput is meant to increase until the traffic volume is near 100% then decreases because of congestion. However, in energy saving routing algorithms, the attempt is to minimize the number of active links and switches. Therefore, the average throughput would decrease even when the traffic volume is near 50%. The trend of throughput in energy saving algorithms is different from non-energy saving performance focused heuristics.



Fig. 10. Delay of heuristic algorithms in milliseconds versus a range of traffic volume from 20% to 90%

Figure 10 shows the delay of the heuristics algorithms measured in milliseconds with respect to traffic volume ranging from 20% to 90%. The NSP algorithm exhibits the least delay followed by NMU. The reason for this is because both NSP and NMU are initialized by the shortest paths. The SDF algorithm is 3 to 5 milliseconds worse in delay than the NSP algorithm. The MaxRESDN algorithm has a 1 millisecond to 3 milliseconds close to the NSP algorithm. It can be observed that for all of the energy saving heuristics, the delay tends to slightly increase with the increasing traffic volume. This is because all the energy saving algorithms attempt to minimize the number of links and switches used.

Figure 11 shows the traffic proportionality of heuristics for the range of traffic volume from 20% to 90%. The MaxRESDN heuristics have demonstrated the maximum traffic proportionality value. The B heuristics, which is an attempt of keeping the trade-off between performance and energy saving, has a closer traffic proportionality to our approach.

Figure 12 shows average traffic proportionality of all traffic volumes from 20% to 90% for MaxRESDN algorithm compared with the other algorithms (in Table IV). The MaxRESDN heuristics exhibits the largest traffic proportionality in terms of link energy consumption. NSP algorithm has the lowest traffic proportionality. As compared to those heuristics which give priority to network performance such as NSP and SPF, our approach is 4 to 5% better in traffic pro-



Fig. 11. Traffic proportionality for the range of traffic volume from 20% to 90%



Fig. 12. Average traffic proportionality of MaxRESDN as compared to other heuristics

portionality, and 3 to 4% better in traffic proportionality than heuristics that give priority to energy saving and maximizing utility such as NSP and HDF.

#### D. Analysis of Utility Parameters

A challenge with the MaxRESDN algorithm is its dependence on the link utility interval parameters. The trends of link utility interval parameters Umin and Umax versus traffic volume and percentage of links saved are analyzed in this subsection. For the fixed Umax value to 95%, we study the effect of the Umin parameter on the percentage of links saved for traffic volume ranging from 20% to 90% Umin. Likewise, for the fixed Umin value to 30%, we study the effect of Umax value on the percentage of links saved. Understanding the relations and the trend for values of Umin and Umax versus the traffic volume would be significant in predicting the utility parameters for future use.



Fig. 13. Effect of utility parameter Umin values on the percentage of links saved for different traffic volumes ranging from 20% to 90% by fixing Umax to 95%

Figure 13 shows the effect of the Umin parameter on the percentage of links saved for traffic volumes 20% to 90%. Each line represents a traffic volume. Fixing Umaxto 95%, and ranging Umin value from 10% to 90%, the trend shows that the value of the Umin that maximizes the links saved increases till it reaches a peak then drops. As the traffic volume increases the peak Umin value increases. The analysis exhibits similar trend for different traffic volumes.



Fig. 14. Effect of utility parameter Umax values on the percentage of links saved for different traffic volumes ranging from 20% to 90% by fixing Umin to 30%

Figure 14 shows the effect of the Umax value on the percentage of links saved for traffic volumes from 20% to 90%. Each line shows the traffic volume the experiment is conducted. The Umin parameter is set to 30% and the Umax value ranges from 40% to 100%. For every flow size, the Umax value that maximizes the number of links saved increases till it reaches its peak then it mainly remains constant except for the 20% traffic volume where there is a slight decrease.



Fig. 15. The effect of the traffic volume ranging from 20% to 90% on the value of Umin and Umax that maximizes the percentage of links saved fixing Umax values to 85%, 90% and 95%

Figure 15 shows the relationship between the traffic volume and the Umin value that maximizes the energy saving. It shows that as the traffic volume increases the Umin that maximizes energy saving increases. The reason for this is because the utilities of the links increase with traffic volume. For example when the percentage of traffic is 90%, the link with minimum utility is above 40%. Which means that if we pick Umin value less than 40%, the candidate list would be empty. Similarly as the traffic volume decrease to 20%, the link utilities starts from 12% and we will have more links in the candidate list. According to results on figures 13 and 15, the problem that need to be solved hence is to find or estimate the peak Umin value for a given traffic volume.



Fig. 16. The effect of the traffic volume ranging from 20% to 90% on the value of Umax that maximizes the percentage of links saved fixing Umin values to 20%, 30%, and 40%

Figure 16 illustrates the trend of Umax value in relation to traffic volume by fixing the value of Umin the at 30%. It demonstrates that as the volume of traffic increases, the Umax maximizing the energy saving exhibits an increasing trend. It also shows that for a traffic volume which is more than 72%, the Umax value keep a constant value near 95%. According to the results depicted on Figure 16 and reffig:tanalysisofumin, picking the right value of Umin and Umax has a direct impact on the performance of the RESDN algorithm.

#### VII. CONCLUSIONS

The flexibility of network control by SDN can be used for energy efficient routing. However, minimizing energy consuming devices without compromising performance is a challenging task. We proposed energy efficiency metric named RESDN (Ratio for Energy Saving in SDN) that puts the utility of links under consideration. Unlike other energy efficiency metrics, RESDN is based on the link utility interval parameters, and captures the dynamic network changes in SDN. We also provided an IP formulation with the objective of increasing the RESDN of a network environment and a heuristics named MaxRESDN that maximizes energy saving. The experiments are conducted on Mininet network emulator and POX controller using the GEANT network topology and dynamic traffic traces. We also simulated the power consumption of OvSwitch, NEC PF5240, and Zodiac FX OpenFlow switches.

Experimental results show that maximizing the RESDN value improves energy efficiency while maintaining acceptable network performance. MaxRESDN is up to 30% better in the number of links and achieves up to 14.7 watts, 10 watts, and 3.2 watts less power consumption for NEC, OVS and Zodiac FX switches respectively as compared to other utilitybased heuristics for energy efficient routing. The maximum RESDN value is achieved by our proposed MaxRESDN method, which also performs close to the solutions that give priority to network performance in terms of average path length, throughput, and delay. Our approach not only does have 3 to 5% higher traffic proportionality but also maintains the trade-off between network performance and energy efficiency. Since the performance of the MaxRESDN heuristics greatly depends on the value of utility interval parameters, we performed a detailed analysis of the parameters with regard to traffic volume. It is demonstrated that the values of utility interval parameters directly affect the efficiency of the algorithm.

As future work, we aim to utilize supervised and reinforcement machine learning techniques to predict the link utility interval parameters in an efficient manner. We also plan to apply the idea behind RESDN metric to incorporate end systems in a data center platform, and improve the scalability through parallelizing the MaxRESDN algorithm to handle multiple flow arrivals concurrently.

#### **ACKNOWLEDGMENTS**

This work was partially supported by the COST (European Cooperation in Science and Technology) framework, under Action IC0804 (Energy Efficiency in Large Scale Distributed Systems), and by TUBITAK (The Scientific and Technical Research Council of Turkey) under Grant 109M761.

#### REFERENCES

- B. G. Assefa and O. Ozkasap, "A survey of energy efficiency in sdn: Software-based methods and optimization models," *Journal of Network* and Computer Applications, 2019.
- [2] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan 2015.
- [3] J. Metzler and A. Metzler, "The 2015 guide to sdn and nfv," January, available at: https://www. a10networks. com/sites/default/files/resource-files/2015Ebook-A10-all. pdf [accessed: 9 Jan 2018], 2015.
- [4] "Flexible sdn solutions for the agile enterprise," http://e.huawei.com/ en/solutions/technical/sdn, accessed: 2017-12-30.
- [5] W. Zhang, X. Zhang, H. Shi, and L. Zhou, "An efficient latency monitoring scheme in software defined networks," *Future Generation Computer Systems, Elsevier*, vol. 83, pp. 303 – 309, 2018.
- [6] M. Cokic and I. Seskar, "Software defined network management for dynamic smart grid traffic," *Future Generation Computer Systems*, *Elsevier*, vol. 96, pp. 270 – 282, 2019.
- [7] C. Qiu, S. Cui, H. Yao, F. Xu, F. R. Yu, and C. Zhao, "A novel qosenabled load scheduling algorithm based on reinforcement learning in software-defined energy internet," *Future Generation Computer Systems, Elsevier*, vol. 92, pp. 43 – 51, 2019.
- [8] F. Shang, L. Mao, and W. Gong, "Service-aware adaptive link load balancing mechanism for software-defined networking," *Future Generation Computer Systems*, vol. 81, pp. 452 – 464, 2018.
- [9] GreenPeace, "Clicking clean: How the Companies are creating the green internet," 2014. [Online]. Available: http://www.greenpeace.org/ international/en/
- [10] R. Maaloul, L. Chaari, and B. Cousin, "Energy saving in carrier-grade networks: A survey," *Computer Standards & Interfaces*, vol. 55, pp. 8–26, 2018.
- [11] L. Zhou, L. N. Bhuyan, and K. K. Ramakrishnan, "Dream: Distributed energy-aware traffic management for data center networks," in *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, ser. e-Energy '19. New York, NY, USA: ACM, 2019, pp. 273–284. [Online]. Available: http://doi.acm.org/10.1145/ 3307772.3328291
- [12] D. Staessens, S. Sharma, D. Colle, M. Pickavet, and P. Demeester, "Software defined networking: Meeting carrier grade requirements," in *Local Metropolitan Area Networks (LANMAN)*, 2011 18th IEEE Workshop on, Oct 2011, pp. 1–6.
- [13] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, "ElasticTree: Saving Energy in Data Center Networks," in *NSDI*, vol. 10, 2010, pp. 249–264.
- [14] X. Wang, Y. Yao, X. Wang, K. Lu, and Q. Cao, "Carpo: Correlationaware power optimization in data center networks," in 2012 Proceedings IEEE INFOCOM. IEEE, 2012, pp. 1125–1133.
- [15] N. H. Thanh, P. N. Nam, T.-H. Truong, N. T. Hung, L. K. Doanh, and R. Pries, "Enabling experiments for energy-efficient data center networks on openflow-based platform," in 2012 Fourth International Conference on Communications and Electronics (ICCE). IEEE, 2012, pp. 239–244.
- [16] B. B. Rodrigues, A. C. Riekstin, G. C. Januário, V. T. Nascimento, T. C. Carvalho, and C. Meirosu, "Greensdn: Bringing energy efficiency to an sdn emulation environment," in 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM). IEEE, 2015, pp. 948–953.
- [17] H. Zhu, X. Liao, C. de Laat, and P. Grosso, "Joint flow routingscheduling for energy efficient software defined data center networks: A prototype of energy-aware network management platform," *Journal* of Network and Computer Applications, vol. 63, pp. 110 – 124, 2016.
- [18] R. Ana C, J. Guilherme C., R. Bruno B, N. Viviane T, C. Tereza C.M.B., and M. Catalin, "Orchestration of energy efficiency capabilities in networks," *Journal of Network and Computer Applications*, vol. 59, pp. 74 – 87, 2016.
- [19] "The Green Grid, url = https://www.thegreengrid.org, note = Accessed: 2017-12-30."
- [20] "Standard Performance Evaluation Corporation, url = https://www. spec.org, note = Accessed: 2019-01-20."
- [21] B. G. Assefa and O. Ozkasap, "A novel utility based metric and routing for energy efficiency in software defined networking," in 2019 International Symposium on Networks, Computers and Communications (ISNCC), June 2019, pp. 1–4.

- [22] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 3, pp. 1617–1634, 2014.
- [23] N. Vasić, P. Bhurat, D. Novaković, M. Canini, S. Shekhar, and D. Kostić, "Identifying and using energy-critical paths," in *Proceedings* of the Seventh Conference on emerging Networking Experiments and Technologies. ACM, 2011, p. 18.
- [24] T. H. Vu, V. C. Luc, N. T. Quan, N. H. Thanh, and P. N. Nam, "Energy saving for OpenFlow switch on the NetFPGA platform based on queue engineering," *SpringerPlus*, vol. 4, no. 1, p. 64, 2015.
- [25] A. Markiewicz, P. N. Tran, and A. Timm-Giel, "Energy consumption optimization for software defined networks considering dynamic traffic," in *IEEE 3rd International Conference on Cloud Networking* (*CloudNet*), Oct 2014, pp. 155–160.
- [26] B. G. Assefa and O. Ozkasap, "Link utility and traffic aware energy saving in software defined networks," 2017 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), 2017.
- [27] F. Giroire, J. Moulierac, T. K. Phan, and F. Roudaut, "Minimization of network power consumption with redundancy elimination," in *Computer communications*, vol. 59. Elsevier, 2015, pp. 98–105.
- [28] G. Xu, B. Dai, B. Huang, J. Yang, and S. Wen, "Bandwidth-aware energy efficient flow scheduling with sdn in data center networks," in *Future Generation Computer Systems*, vol. 68. Elsevier, 2017, pp. 163–174.
- [29] M. M. Tajiki, S. Salsano, L. Chiaraviglio, M. Shojafar, and B. Akbari, "Joint energy efficient and qos-aware path allocation and vnf placement for service function chaining," *IEEE Transactions on Network and Service Management*, vol. 16, no. 1, pp. 374–388, 2018.
- [30] R. Wang, W. Y. SuixiangGao, and Z. Jiang, "Restorable energy aware routing with backup sharing in software defined networks," *Journal of Communications*, 2015, vol. 10, no. 8, 2015.
- [31] T. N. Huu, N. P. Ngoc, H. T. Thu, T. T. Ngoc, D. N. Minh, H. N. Tai, T. N. Quynh, D. Hock, C. Schwartz *et al.*, "Modeling and experimenting combined smart sleep and power scaling algorithms in energy-aware data center networks," *Simulation Modelling Practice and Theory*, vol. 39, pp. 20–40, 2013.
- [32] R. Garroppo, G. Nencioni, L. Tavanti, and M. G. Scutella, "Does traffic consolidation always lead to network energy savings?" *IEEE Communications Letters*, vol. 17, no. 9, pp. 1852–1855, 2013.
- [33] R. Bolla, R. Bruschi, F. Davoli, and C. Lombardo, "Fine-grained energy-efficient consolidation in sdn networks and devices," *IEEE Transactions on Network and Service Management*, vol. 12, no. 2, pp. 132–145, 2015.
- [34] T. H. Vu, V. C. Luc, N. T. Quan, T. M. Nam, N. H. Thanh, and P. N. Nam, "The new method to save energy for Openflow Switch based on traffic engineering," in *Electronic Design (ICED), 2014 2nd International Conference on*, Aug 2014, pp. 309–314.
- [35] R. Wang, Z. Jiang, S. Gao, W. Yang, Y. Xia, and M. Zhu, "Energyaware routing algorithms in software-defined networks," in World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on a, June 2014, pp. 1–6.
- [36] M. Rahnamay-Naeini, S. S. Baidya, E. Siavashi, and N. Ghani, "A traffic and resource-aware energy-saving mechanism in software defined networks," in 2016 International Conference on Computing, Networking and Communications (ICNC). IEEE, 2016, pp. 1–5.
- [37] R. Maaloul, R. Taktak, L. Chaari, and B. Cousin, "Energy-aware routing in carrier-grade ethernet using sdn approach," *IEEE Transactions* on Green Communications and Networking, 2018, vol. 2, no. 3, pp. 844–858, 2018.
- [38] D. Tuncer, M. Charalambides, S. Clayman, and G. Pavlou, "Adaptive resource management and control in software defined networks," *IEEE Transactions on Network and Service Management*, vol. 12, no. 1, pp. 18–33, 2015.
- [39] M. Levy and D. Raviv, "A novel framework for data center metrics using a multidimensional approach," in 15th LACCEI International Multi-Conference for Engineering, Education, and Technology: Global Partnerships for Development and Engineering Education, July 2017, pp. –.
- [40] D. Vemula, B. Setz, G. S. V. Rao, G. Gangadharan, and M. Aiello, "Metrics for sustainable data centers," *IEEE Transactions on Sustainable Computing*, 2017, vol. 2, no. 3, pp. 290–303, 2017.
- [41] M. Levy and D. Raviv, "An overview of data center metrics and a novel approach for a new family of metrics," Advances in Science,

Technology and Engineering Systems J, vol. 3, no. 2, pp. 238–251, 2018.

- [42] E. Jaureguialzo, "Pue: The green grid metric for evaluating the energy efficiency in dc (data center). measurement method using the power demand," in *Telecommunications Energy Conference (INTELEC), 2011 IEEE 33rd International.* IEEE, 2011, pp. 1–8.
- [43] D. Azevedo and M. Patterson, "Carbon usage effectiveness (cue): a green grid data center sustainability metric, 2010," *The Green Grid*, 2010.
- [44] D. Azevedo, S. C. Belady, and J. Pouchet, "Water usage effectiveness (wue<sup>TM</sup>): A green grid data center sustainability metric,2011," *The Green Grid*, 2011.
- [45] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: rapid prototyping for software-defined networks," in *Proceedings of the 9th* ACM SIGCOMM Workshop on Hot Topics in Networks, 2010, p. 19.
- [46] S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski, "Sndlib 1.0—survivable network design library," in *Networks: An International Journal*, vol. 55. Wiley Online Library, 2010, pp. 276–286.
- [47] F. Kaup, S. Melnikowitsch, and D. Hausheer, "Measuring and modeling the power consumption of openflow switches," in *Network and Service Management (CNSM), 2014 10th International Conference on*. IEEE, 2014, pp. 181–186.
- [48] "NEC ProgrammableFlow PF5240 Switch, url = https://www.necam. com/pflow/doc.cfm?t=pflowpf5240switch, note = Accessed: 2018-12-26."
- [49] "Production quality, multilayer open virtual switch," accessed: 2018-12-26. [Online]. Available: http://www.openvswitch.org/
- [50] "Zodiac FX OpenFlow Switch Hardware-Northbound Networks, url = https://northboundnetworks.com/products/zodiac-fx, note = Accessed: 2018-12-26."



**Beakal G. Assefa** is a Ph.D. Candidate, teaching and research assistant in the Department of Computer Engineering at Koç University. He received his M.S. in Computer Engineering from Izmir Institute of Technology in 2012. He received his B.S. degree in Computer Science and Information Technology from Haramaya University, Ethiopia. His research interests include databases, softwaredefined networks, energy efficiency, machine learning, and distributed systems



Öznur Özkasap received the Ph.D. degree in Computer Engineering from Ege University in 2000. From 1997 to 1999, she was a Graduate Research Assistant with the Department of Computer Science, Cornell University, where she completed her Ph.D. dissertation. She is currently a Professor with the Department of Computer Engineering, Koç University, which she joined in 2000. Her research interests include distributed systems, multicast protocols, peer-to-peer systems, bio-inspired distributed algorithms, mobile and vehicular ad

hoc networks, energy efficiency, cloud computing, and computer networks. Prof. Özkasap is IEEE Senior Member, serves as an Area Editor of the Future Generation Computer Systems journal, Elsevier Science, and National Representative of ACM-W Europe. She is a recipient of the Turk Telekom Collaborative Research Awards, the Career Award of TUBITAK (The Scientific and Technological Research Council of Turkey), and TUBITAK/NATO A2 Ph.D. Research Scholarship Abroad, and she was awarded Teaching Innovation Grants by Koç University.