Framework for Traffic Proportional Energy Efficiency in Software Defined Networks

Conference Paper · June 2018

DOI: 10.1109/BlackSeaCom.2018.8433618

2 authors:

Beakal Gizachew Assefa
Addis Ababa Institute of Technology

Oznur Ozkasap
Koc University

Framework for Traffic Proportional Energy Efficiency in Software Defined Networks

Beakal Gizachew Assefa and Oznur Ozkasap
Department of Computer Engineering
Koc University, Istanbul, Turkey
{bassefa13,oozkasap}@ku.edu.tr

***Abstract*—Software Defined Networking (SDN) achieves programmability of network elements by separating the control and the forwarding planes, and provides efficiency through optimized routing and flexibility in network management. As the energy costs contribute largely to the overall costs in networks, energy efficiency is a significant design requirement for modern networking mechanisms. However, designing energy efficient solutions is complicated since there is a trade-off between energy efficiency and network performance. In this paper, we propose traffic proportional energy efficient framework for SDN and heuristics algorithm that maintains the trade-off between efficiency and performance. We also present IP formulation for traffic proportional energy efficiency problem. Comprehensive experiments conducted on Mininet emulator and POX controller using Abilene, Atlanta, and Nobel-Germany real-world topologies and traffic traces show that our approach saves up to 50% energy while achieving a performance closer to the algorithms prioritizing performance.**

I. Introduction

Software Defined Networking (SDN) is a widely researched topic in networking which separates the control and data plane of a network. SDN provides a very powerful feature of network programmability and has been widely adopted by major companies (e.g. Yahoo, Facebook, Google, Cisco, Huawei, and Microsoft) in a relatively short period of time [1], [2]. On the other hand, energy cost constitutes more than 10% of OpEx (Operating Expenses) of an ICT service provider. Energy efficiency has become a significant design requirement for modern networking and is predicted to increase at least by 60% by 2020 [3]. Network components work with full capacity at all times regardless of the traffic volume to provide the highest performance possible.

Traffic proportional energy efficiency is an attempt to make energy cost proportional to the amount of traffic streaming through the network. A practical solution for such problem is to sleep/turn off under-utilized components for low traffic volume. Minimizing the components turned on to accommodate a given traffic, however, degrades performance [4]–[9]. Designing energy efficient solutions is non-trivial since they need to tackle the trade-off between energy efficiency and network performance.

In a dynamically changing network environment, the practicality of solutions in traditional networking is impossible. However, SDN enables the network to adapt the changing environment by recomputing paths, optimizing routes and increasing flexibility. The controller periodically gathers statistical information about the traffic, the status of the switches, the status of the links, the status of the end systems, and the topology of the network.

Related works in energy aware routing are focused on minimizing the number of active network components [4], [5], [10]–[12]. In our previous work [6], we presented two heuristics algorithms that not only minimize energy consumption but also maintain an acceptable performance. The first algorithm attempts to maintain performance where the other one increases the utility of active links. In this work, we propose a three-component framework that makes energy consumption proportional to the traffic volume in a dynamic environment. The contributions are as follows

- We propose a traffic proportional energy efficient framework for SDN. The components in the framework deal with traffic monitoring, topology management, and optimization.
- We develop IP formulation for the traffic proportional energy saving problem with the objective of minimizing the energy cost of switches and links.
- We propose a heuristics that maintains the trade-off between network performance and energy efficiency.
- Experiments are performed using Mininet [13] and real-world topologies (Abiline, Atlanta, and Nobel-Germany) and traffic traces from SNDlib [14]. Results show that the energy cost is proportional to the amount of traffic volume. Energy saving of more than 50% is achieved for a 20% traffic volume and 3 to 5% saving for a 90% traffic volume.

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 describes traffic proportional framework we proposed, provides IP formulation for minimizing the energy cost of network components, and presents example heuristics to solve the IP formulation. Section IV discusses experimental results and analysis. Section V concludes and states future directions.

II. Related Work

Network traffic is defined as the amount of packets transmitted through a network at a given point in time and is measured in bytes. Traffic management involves analyzing, predicting and regulating the behavior of the network devices with regard to traffic in order to optimize the performance and QoS requirements. Examples based on usage of traffic are load balancing, performance optimization, security, access control, bandwidth management, and energy usage [15] [16].

In traffic aware energy saving, the energy saving capabilities are the switches and the links. Approaches in [4], [9], [17]–[19] focus on minimizing the number of active switches. Others focus on putting links into inactive state [5], [20]. Instead of putting the link inactive, Adaptive Link Rate (ALR) based techniques that aim at reducing energy consumption by scaling the link rate are also implemented in SDN [7]. A joint objective of minimizing the energy consumption by links and switches is also widely researched [6], [8], [12], [21], [22]. Most of the energy saving solutions only consider minimizing the number of active components which affects performance.

A variety of IP formulations has been proposed to capture the traffic aware energy consumption problem. However, optimal solutions do not scale to large number nodes and links. Considering this, several heuristics algorithms are proposed. Some algorithms are designed for specific topology structures like fat-tree ([4], [5], [22]) and Bcubic ([7], [23]). There are heuristics that independent of topology structures [6], [9], [12], [24].

ElasticTree [4] is a power management solution based on turning off links and switches according to the amount of traffic load. The three methods proposed are formal solution, greedy bin-packing, and a topology-aware heuristic, where the inputs are network topology, routing constraints, power model, and traffic matrix, and the outputs are the subset of links and per-flow route information. However, they focus on energy efficiency where link utilities and network performance are not considered.

A topology type independent energy-saving algorithm was proposed in [12]. However, it depends on flow arrival order and also do not consider network performance. In our previous work we proposed a utility based heuristics that is topology type independent [6]. It also proposes two heuristics that give priority to either network performance or energy efficiency. On top of prior work, we propose a single algorithm that achieves both efficiency and performance at the same time.

III. Traffic Proportional Framework for Energy Efficiency in SDN

The goal of the traffic proportional framework is to design a controller which makes the energy consumption of the network proportional to the traffic volume. Figure 1 illustrates the framework for traffic proportional energy efficiency. The information of the traffic generated from the applications running on the top of the controller are passed to the Traffic Manager. The Traffic Manager passes this traffic information to the TA Optimizer. The statistics of the network and the topology information is fed to the TA Optimizer from the switches. The Optimizer generates an optimal subgraph based on the traffic volume and the utility of the links. Since the framework is designed to work in a dynamic environment, a low traffic load should result in a subgraph with smaller number of active links and switches, whereas high traffic load should increase the number of active links and switches in the subgraph respectively. In the former, the sleep/turn-off decision for a subset links and switches would be achieved. In the latter, however, a subset of links and switches would be turned on or made active.

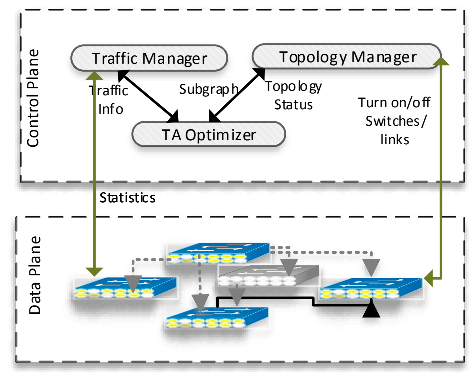

Fig. 1: A three-component framework for traffic proportional energy efficiency in SDN

A. *Traffic Aware Model*

Based on the review of various models used to capture traffic proportional energy consumption, we propose a general optimization model that identifies energy capabilities in SDN from traffic point of view, where the major energy saving components are considered as the links and the switches. The model jointly minimizes the number of switches and the number of links used to accommodate the given traffic. It is tailored to include additional parameters and constraints. For simplicity, the parameters of the optimization model refer to snapshot of the network state. Given the flows which represent the traffic, the problem is defined as allocating links and switches for each flow while minimizing the total number of active links and switches. Inspired by the approaches of [4], [8], [10], [23], we use multi-commodity flow problem formulation. Each flow is treated as a commodity with source, destination, and flow rate. Our model is inspired by the formulation of [4] and the objective of the model is to minimize the power consumption of the links and switches used to handle the flows. The other approaches in this category consider additional parameters like time, bandwidth awareness, and redundant path elimination.

In traffic aware model, the network is represented as a bidirectional weighted graph $\mathbb{G}$= ($\mathbb{Z}$,$\mathbb{E}$) where $\mathbb{Z}$ is the set of switches and $Z_i \in \mathbb{Z}$ represents switch i and $e_{ij} \in \mathbb{E}$ represents that there is link between switches Z_i and Z_j. The weight W_{ij} corresponds to the bandwidth of the link connecting switches Z_i and Z_j. Let binary variables S_i and L_{ij} denote the status of switch Z_i and link e_{ij} such that

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

and

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

CS_i and C_{ij} are power consumption of switch Z_i and the link e_{ij} measured in watt.

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

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

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

Where $Umin$ and $Umax$ are the minimum and maximum utilities of links to keep the trade-off between performance and efficiency which we call energy profit margins.

The multi-objective function (equation 1) minimizes the sum of the energy consumption of the switches and the links. The first item in the objective function, the sum of $L_{ij} * C_{ij}$, refers to the total energy consumption incurred by all flows using edge e_{ij}. The second item is the total power consumption of all active switches in the network. The objective function jointly minimizes the sums subject to the constraints described next.

$$\text{minimize } (\sum_{\forall e_{ij}} L_{ij} * C_{ij} + \sum_{\forall Z_i} S_i * CS_i) \tag{1}$$

$$\text{subject to } \sum_{\forall f} f_{ij} * \lambda_f \leq W_{ij} \ , \forall e_{ij} \tag{2}$$

$$\sum_{\forall f} f_{ij} = \sum_{\forall f} f_{ji} \ , Z_i \& Z_j \neq sr, Z_i \& Z_j \neq ds \tag{3}$$

$$f_{mj} = f_{in} \ , \ Z_m = sr \ , \ Z_n = ds \ , \ \forall e_{mj} \ , \ \exists e_{in} \tag{4}$$

$$f_{ij} \leq S_j \ and \ f_{ji} \leq S_j \ , \forall Z_j \in \mathbb{Z} \tag{5}$$

$$f_{ji} \leq S_j \ \forall Z_j \in \mathbb{Z} \tag{6}$$

$$S_i \leq \sum_{\forall f} [f_{ij} + f_{ji}] \ , \ \forall Z_i \in \mathbb{Z} \tag{7}$$

$$L_{ij} \leq S_i \ , \ L_{ij} \leq S_j \ \forall Z_i \tag{8}$$

Equation 2 states that the total rate of flows between two switches should not exceed the link capacity. Equations 3 and 4 assert conservation of flow which states that a flow is not created or lost in the network. Constraint in Equation 3 states that the number of flows entering to and leaving from switches which are neither destination nor sources of a flow should be equal. Constraint on equation 4 assures a flow entering from source switch should leave the network at the destination switch. The constraints on equations 5, 6, and 7 maintain the correlation between switches and links using the switch state variable and flow-link variables. While constraints 5 and 6 state that no flow should use a link connected to an inactive switch, constraint 7 states that if no flow is passing through the links connected to a given switch, then the switch is switched off. Constraint in equation 8 states that a link that is connected to an inactive switch should be inactive. Given the formulation and the parameters of the model, the output of the optimizer is list of links and active switches for each flow $f \in \mathbb{F}$.

B. Heuristics Algorithm

We have proposed NextShortestPath (NSP) and NextMaximumUtility (NMU) algorithms that maintain the trade-off between performance and energy efficiency in our prior work [6]. The objective of NSP 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. Whereas NSP gives priority to performance, NMU focuses on maximizing the utility of active links. In this work, we present a heuristics that achieve both simultaneously. We define the Energy Profit Threshold (EPT) as follows.

$$EPT \leftarrow 100 * \frac{\sum_{\forall e_{ij}} X_{ij}}{\sum_{\forall e_{ij}} L_{ij}}$$

EPT value of 100 means that all links that are turned on are operating between the interval defined by parameters $Umin$ and $Umax$. EPT value of 0 means that none of the links are operating profitably which means that the network is under utilized.

Algorithm 1 MEPT($\mathbb{G}$, $\mathbb{F}$,$\mathbb{U}$,$Umin$,$Umax$)

1: **Input:** Graph $\mathbb{G}$, set of traffic flow $\mathbb{F}$, initial utility $\mathbb{U}$ minimum utility $Umin$, and maximum utility $Umax$
2: **Output:** Modified utility of links $\mathbb{U}$ and graph $\mathbb{G}$
3: **for all** $f = (sr, ds, \lambda_f) \in F$ **do**
4: $\quad Pall_{sr,ds} \leftarrow \ \texttt{AllPath}(sr, ds)$
5: $\quad MaxEPT \leftarrow 0$
6: $\quad$ **for all** $P \in Pall_{sr,ds}$ **do**
7: $\quad\quad$ **if** $MaxEPT \leq EPT(P)$ **then**
8: $\quad\quad\quad MaxEPT \leftarrow EPT(P)$
9: $\quad\quad\quad path_f \leftarrow P$
10: $\quad\quad$ **end if**
11: $\quad$ **end for**
12: $\quad$ **for all** $e_{ab} \in path_f$ **do**
13: $\quad\quad U_{ab} \leftarrow U_{ab} + \dfrac{\lambda_f}{W_{ij}}$
14: $\quad$ **end for**
15: **end for**
16:
17: **for all** $e_{ab} \in \mathbb{G}$ **do**
18: $\quad$ **if** $U_{ab} == 0$ **then**
19: $\quad\quad e_{ab} \leftarrow inactive$
20: $\quad$ **end if**
21: **end for**
22: **return** $\mathbb{G}$

Algorithm 1 aims to maximize the EPT value of the network as it assigns a path to each flow. The inputs for the MEPT are the network represented as graph $\mathbb{G}$, the 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 rate of flow $_f$ which correspond to source address, destination address, and rate of flow f measured in bits per second (line 3). Line 13 increments the utilities of all the links on the path assigned to flow f by the $\frac{\lambda_f}{W_{ij}}$ factor. Line 6 to 11 assigns each flow to the path that maximizes the EPT among alternatives. The $pathmaximizingEPT$ method assigns paths per each flow in an incremental manner. Lines 17 to 21 makes link e_{ab} inactive if its corresponding utility U_{ab} is 0.

IV. Experimental Setup and Results

The experimental setup consists of POX controller and Mininet [13] network emulator installed on Ubuntu 16.04 64-bit. The topologies are created on Mininet, and the heuristics are implemented on the POX controller. The experiments are conducted using real traces from SNDlib [14], in particular the Abilene, Atlanta, Nobel-Germany topologies.

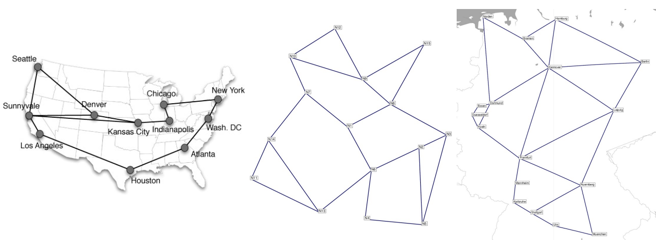

Fig. 2: Images of the network topologies used a)Abilene, b)Atlanta, and c) Nobel-Germany

Figure 2 shows the images of the Abilene, Atlanta, and Nobel-Germany topologies used in this experiment. Abilene network created by the Internet2 community and connects regional network among universities, corporations, and institutions in the US. The Atlanta network represents an ATM network in Los Angeles. Nobel-Germany is provided by a European project called Nobel and connects cities in Germany.

TABLE I: Topologies and Traces

Topology	Nodes	Edges	Min Deg.	Max Deg.	Avg Deg.
Abilene	12	15	1	4	2.5
Atlanta	15	22	2	4	2.95
Nobel-Germany	17	26	2	6	3.06

Table I presents the number of nodes, edges, the minimum degree of a node, the maximum degree of a node and the average degree of a node of the topologies used in this experiment.

The metrics we used to test our framework are energy saving, and average path length. Energy saving is calculated as

$$Energy\ Saving \leftarrow 100(1 - \frac{\sum_{\forall e_{ij}} L_{ij}}{|\mathbb{E}|})$$

$$Average\ Path\ Length \leftarrow \frac{\sum_{\forall f} \sum_{\forall e_{ij}} (f_{ij})}{|\mathbb{F}|}$$

The heuristics algorithms we used to test our framework are Next Shortest Path (NSP), Next Maximum Utility (NMU), and Maximize Energy Profit Threshold (MEPT). The three heuristics are topology independent, keep the trade-off between performance and energy efficiency and can easily applicable on the top of other algorithms. They are also different in the parameters they optimize. The rationale behind the NSP and NMU is to redirect the paths of flows passing through underutilized links to next shorter path and in the direction of the link with maximum utility respectively [6]. If all the flows passing in such a link are redirected successfully, that link will be put into the inactive state. The MEPT heuristics, on the other hand, jointly maximizes link utilities at the same time minimizes the number of active links through maximizing the EPT value of the network.

Figure 3 shows the energy saving and average path length results for the NSP, NMU, and MEPT heuristics for traffic volumes ranging from 20% to 90% in the Abilene topology. In terms of energy saving, the MEPT exhibits the maximum energy saving, whereas NMU and NSP exhibit the lowest energy saving. NSP is the lowest in terms of average path length. This is because NSP gives priority to performance first then energy saving.

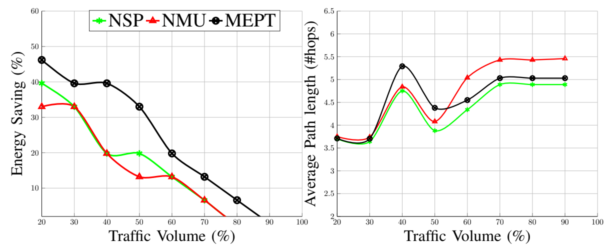

Fig. 3: Comparison of algorithms for different traffic volumes a) Energy Efficiency b) Average Path Length for Abilene topology and traffic trace

Figure 4 shows the energy saving and average path length analysis of NSP, NMU, and MEPT algorithms on the Atlanta topology and traffic volume ranging from 20% to 90%. Energy saving of more than 50% is achieved for 20% traffic volume. Similar to the case of the Abilene topology, the maximum energy is achieved by MEPT heuristics. In terms of average path length, three algorithms show similar results.

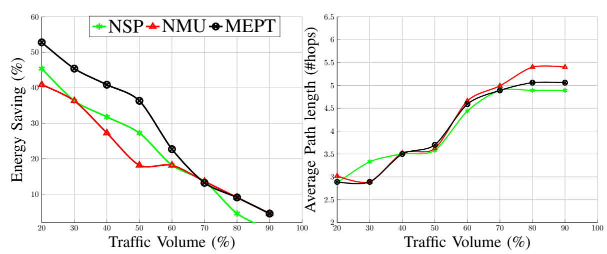

Fig. 4: Comparison of algorithms for different traffic volumes a) Energy Efficiency b) Average Path Length for Atlanta topology and traffic traces

Figure 5 shows the energy saving and average path length results of the NSP, NMU, and MEPT heuristics tested on Nobel-Germany topology for traffic volumes ranging from 20% to 90%. Similar to the Abilene and Atlanta topologies, the MEPT shows the maximum energy saving. The NSP is still the best in terms of average path length.

The general findings of the experiments presented in Figures 3 to 5, is that energy saving decreases as the amount

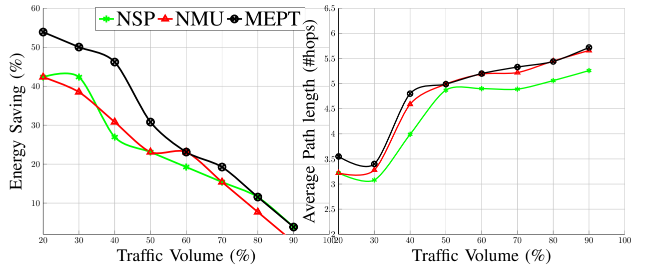

Fig. 5: Comparison of algorithms for different traffic volumes a) Energy Efficiency b) Average Path Length for the Nobel-Germany topology and traffic trace

of traffic volume increases. The maximum energy saving which is slightly above 50% is achieved when the traffic volume is 20% and the lowest energy saving (3%) is achieved when the traffic volume is 90% for the tree topologies. The average path length increases with the traffic volume. The NSP algorithm energy saving is on average 10% less than MEPT for low traffic. NSP is best for energy saving scenarios where performance is also vital. MEPT not only is the best in terms of energy saving but also gives a performance close to NSP. NSP and NMU can be applied on top of other results, since the inputs are snapshots of the network status. MEPT on the other hand can be applied in a dynamic environment where live re-directing of paths is required.

V. Conclusions

Energy efficiency has become a critical concern for IT equipment designs. The main problem in networks however is energy consumption is constant regardless of the volume of traffic. The flexible control provided by SDN paved the way to optimize the efficiency of network environment dynamically. In this work, we presented traffic proportional energy efficiency framework for SDN, IP formulation for the traffic proportional energy efficiency problem, and a heuristics algorithm. The framework is implemented on top of POX controller. Experiments conducted on a real topologies and traffic traces show that our approach not only saves up to 50% of the energy cost at low traffic volume, but also achieves performance closer to the next shortest path algorithm. As future work, we aim to extend the framework to data center networks where end-systems would also be considered in the formulation.

References

[1] 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.

[2] D. Kreutz, F. M. V. Ramos, P. E. Verssimo, 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] GreenPeace. (2014) Clicking clean: How the Companies are creating the green internet. [Online]. Available: http://www.greenpeace.org/international/en/

[4] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, "ElasticTree: Saving Energy in Data Center Networks," in *NSDI*, 2010.

[5] X. Wang, Y. Yao, X. Wang, K. Lu, and Q. Cao, "CARPO: Correlation-AwaRe power optimization in data center networks," in *IEEE INFOCOM, 2012*, pp. 1125–1133.

[6] 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)*.

[7] 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 *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, May, pp. 948–953.

[8] H. Zhu, X. Liao, C. de Laat, and P. Grosso, "Joint flow routing-scheduling 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.

[9] 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.

[10] F. Giroire, J. Moulierac, T. K. Phan, and F. Roudaut, "Minimization of network power consumption with redundancy elimination," vol. 59. Elsevier, 2015, pp. 98 – 105.

[11] 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, pp. 1–6.

[12] 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.

[13] 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.

[14] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, "SNDlib 1.0–Survivable Network Design Library," in *Proceedings of the 3rd International Network Optimization Conference (INOC 2007)*.

[15] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, "A roadmap for traffic engineering in SDN-OpenFlow networks," *Computer Networks*, vol. 71, pp. 1–30, 2014.

[16] S. A. Pistirica, O. Poncea, and M. C. Caraman, "QCN Based Dynamically Load Balancing: QCN Weighted Flow Queue Ranking," in *Control Systems and Computer Science (CSCS), 2015 20th International Conference on*. IEEE, pp. 197–204.

[17] M. Canini, D. Venzano, P. Perešíni, D. Kostić, and J. Rexford, "Identifying and Using Energy-critical Paths," in *Seventh Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT, 2011. ACM.

[18] 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.

[19] 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.

[20] T. Nguyen *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, 2013.

[21] G. Xu, B. Dai, B. Huang, and J. Yang, "Restorable energy aware routing with backup sharing in software defined networks," *Journal of Communications*, vol. 10, 2015.

[22] 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, pp. 1–5.

[23] W. Rui, SuixiangGao, Y. Wenguo, and J. Zhipeng, "Bandwidth-aware energy efficient routing with sdn in data center networks," in *Embedded Software and Systems (ICESS), 2015 IEEE 17th International Conference on*, pp. 766–771.

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