A Traffic and Resource-aware Energy-Saving Mechanism in Software Defined Networks

Mahshid Rahnamay-Naeini*, Sonali Sen Baidya*, Ehsan Siavashi*, and Nasir Ghani†
E-Mail: {mahshid.naeini, sonali.sen-baidya, ehsan.siavashi}@ttu.edu and nghani@usf.edu
*Department of Computer Science, Texas Tech University, Lubbock, TX, USA
†Department of Electrical Engineering, University of South Florida, FL, USA

***Abstract*—Energy-saving mechanisms are crucial in reducing the environmental impact and energy costs of ICT network infrastructures. Integration of renewable energy resources to ICT infrastructures and dynamic energy pricing suggest that not only traffic demands and communication resources but also energy resources should be taken into account in designing energy-saving mechanisms for network infrastructures. With the introduction of Software Defined Networking (SDN) paradigm with a logically centralized controller architecture and programmability of network elements new opportunities have emerged for improving the energy efficiency of the networks. In this paper, by exploiting the SDN features and adopting the idea of powering down unnecessary links to save energy, we formulate an optimization problem for identifying the optimum set of active links that reduces the energy cost of the network while satisfying traffic demands and respecting communication and energy resources. To solve this optimization problem we introduce a computationally efficient heuristic algorithm. Using simulation results, we show that the solutions of the heuristic algorithm reduce the energy cost of the network up to 34% while satisfying the constraints of the problem.**

***Index Terms*—Software Defined Networks, Energy Efficiency, Optimization, Heuristic Algorithm, Traffic Demand, Energy Price**

I. INTRODUCTION

Network infrastructures are major energy consumers of ICT industry due to their large number of devices that are running nonstop in order to ensure availability and responsiveness of the network to dynamic communication demands [1]. To enable reducing the energy cost and the environmental impact of network infrastructures and also helping in reliable operation of the electrical grid, renewable resources and variable pricing signals have been integrated to the ICT systems [2]. However, the tight coupling between control and forwarding planes of traditional networks and complexities in dynamic configuration of components make it difficult to operate the network with energy considerations while capturing the effects of dynamic traffic as well as variable energy prices and renewable resources at various geographical locations.

The recently introduced Software Defined Networking (SDN) paradigm, [3], [4], offers new opportunities to improve various aspects of efficiency and reliability of ICT networks by enabling the programming of network components and reducing the complexity of network management, which are made possible by the separation of the control and forwarding planes. Moreover, the availability of a global view of the network as well as traffic information in a logically centralized controller in SDN allows the development of centralized networking mechanisms that can result in more efficient operating configurations compared to distributed mechanisms that need network-wide interactions and cooperation of components with local information (as in the traditional networks). Although many challenges in SDN such as scalability, security and interoperability are yet to be addressed within the SDN society, many large enterprises and networking organizations are adopting SDN to benefit from the opportunities and features it provides. Particularly, SDN provides opportunities to provision and allocate resources through software and applications in SDN controller, for instance, for energy-saving purposes [5]–[8]. In this paper we exploit the SDN technology capabilities to present a traffic- and resource-aware energy-saving mechanism in SDN-enabled network infrastructures. Here, by *resources* we mean both energy resources and communication resources.

The energy-saving mechanism presented in this paper reduces the energy cost of the network by powering down links, which their activity is not necessary in satisfying traffic demands. Note that, in this paper, the decision on powering off links not only depends on the traffic demand but also on the availability of energy and communication resources. The idea of switching off network elements to save energy is not new and has been studied in the literature [6], [7], [9]–[11]. For instance, energy-aware routing (EAR) is a class of approaches in green networking, which addresses the problem of routing with the minimum number of active elements in the network while satisfying the traffic demand of the network [12], [13]. While the majority of such efforts has been presented for traditional networks [9], [10], [13], energy-saving approaches based on powering off network elements for SDN-enabled network infrastructures have also emerged [6], [7]. While the work presented in the current paper follows a similar idea as that of the above efforts, it presents a novel approach that considers the role of energy resources and energy prices at various locations in addition to the traffic demand and communication resources in identifying the optimum set of active links in the network. The decision on powering links on/off will be dictated by the SDN controller to the switches, e.g., using OpenFlow protocol [14] and adjusting rule tables.

The introduced energy-saving problem is casted in a nonlinear optimization problem. To solve this problem, we present a computationally efficient heuristic algorithm that finds a sub-optimal solution to the problem. We will show through our simulation studies that the presented algorithm results in up to 34% savings in the energy cost of the network while satisfying the constraints of the problem and responding to energy prices.

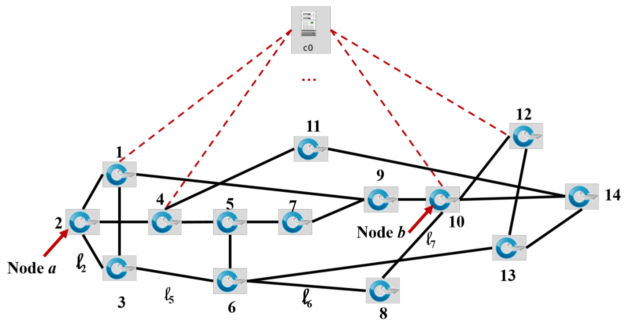


Fig. 1. An example of SDN-enable network with one controller and the set of switches connected based on the NSF Network topology.

II. System Model

We model the network of switches in an SDN-enabled network with an undirected graph, denoted by $G{=}{<}V,E{>}$, where V is the set of nodes in the network and E is the set of links represented by the unordered pairs of nodes with cardinality denoted by L. We assume that an SDN controller is responsible for all the switches in the network. The schematic of this system with the SDN switches and the SDN controller is depicted in Fig. 1. Also, a summary of the notations used in this paper is presented in Table I. Note that we use the terms node and switch interchangeably.

We associate a unique index to each link of the network from the set $\mathcal{L} = \{1, 2, ..., L\}$, i.e., links are denoted by ℓ_k for $k \in \mathcal{L}$. We assume that each link, say ℓ_k, has a capacity (Gbps) denoted by B_k, for which traffics larger than this amount will result in performance degradation and packet loss.

A path in the network is an ordered sequence of links, which we denote by a sequence of link indices. For instance, based on the network in Fig. 1, a path from node a to node b is $(2, 5, 6, 7)$ according to the link indices marked in the figure. For every pair of nodes in the network, say nodes $x, y \in V$, we consider a set of pre-calculated shortest paths from node x to node y denoted by $\mathbf{R}_{xy} = \{\mathbf{P}^{(1)}_{xy}, \mathbf{P}^{(2)}_{xy}, ..., \mathbf{P}^{(c)}_{xy}\}$, where the paths are sorted based on their length (e.g., based on number of hops, delay or physical distance metrics) from shortest to longest. For instance, $\mathbf{P}^{(1)}_{xy}$ is the shortest path and $\mathbf{P}^{(i)}_{xy}$ is the i-th shortest path from node x to node y. We calculate up to K shortest paths between every pair of nodes, i.e., we assume $1 \leq |\mathbf{R}_{xy}| \leq K$, where $|.|$ denotes the cardinality of the set. We assume that at each time only one path is used for routing messages from node x to node y. We also assume that node y can use a different path, from the one that node x selects, to communicate with node x. To refer to the j-th link in the i-th shortest path from node x to node y, we simply use $\mathbf{P}^{(i)}_{xy}(j)$ notation. We also denote the number of links in the path by $D(\mathbf{P}^{(i)}_{xy})$. Moreover, function $g(\mathbf{P}^{(i)}_{xy}, k)$ defined as $g(\mathbf{P}^{(i)}_{xy}, k) := \sum_{j=1}^{D(\mathbf{P}^{(i)}_{xy})} \delta(\mathbf{P}^{(i)}_{xy}(j), k)$, where δ is the Kronecker delta function, specifies if the link index k belongs to path $\mathbf{P}^{(i)}_{xy}$ or not. The function g returns 1 if the link k belongs to $\mathbf{P}^{(i)}_{xy}$ and 0 otherwise.

TABLE I
Table of Notations

V	$\triangleq$	Set of nodes (SDN switches) in the network.
N	$\triangleq$	Total number of nodes in the network.
E	$\triangleq$	Set of links in the network.
L	$\triangleq$	Total number of links in the network.
$\mathcal{L}$	$\triangleq$	Set of link indices in the network, i.e., $\mathcal{L} = \{1, 2, ..., L\}$.
ℓ_k	$\triangleq$	Link with index k, where $k \in \mathcal{L}$.
B_k	$\triangleq$	Capacity of link ℓ_k in Gbps.
$\mathbf{R}_{xy}$	$\triangleq$	A sorted set of shortest path between switches x and y.
$\mathbf{P}^{(i)}_{xy}$	$\triangleq$	The i-th shortest path from node x to node y.
$\mathbf{P}^{(i)}_{xy}(j)$	$\triangleq$	The index of j-th link in $\mathbf{P}^{(i)}_{xy}$.
$D(\mathbf{P}^{(i)}_{xy})$	$\triangleq$	Number of links in the path $\mathbf{P}^{(i)}_{xy}$.
$g(\mathbf{P}^{(i)}_{xy}, k)$	$\triangleq$	A function which specifies if link ℓ_k belongs to the path $\mathbf{P}^{(i)}_{xy}$ or not.
$\mathbf{T}$	$\triangleq$	Traffic matrix of the network.
T_{xy}	$\triangleq$	Traffic in Gbps from node x to node y.
δ	$\triangleq$	A small value representing variations in traffic between nodes.
X_i	$\triangleq$	A binary variable indicating if link ℓ_i is active ($X_i = 1$) or inactive ($X_i = 0$).
$Q^k_{xy}(\mathbf{X})$	$\triangleq$	A function indicating if the k-th shortest path from node x to y is active ($Q^k_{xy} = 1$) or inactive ($Q^k_{xy} = 0$).
e^x_i	$\triangleq$	Energy price of active link ℓ_i at switch x.
e_i	$\triangleq$	Total energy cost of active link ℓ_i, i.e., $e_i = e^x_i + e^y_i$, where x and y are the nodes at the two ends of ℓ_i.
$W_i(\mathbf{X})$	$\triangleq$	Amount of traffic going through link ℓ_i.
$U_a(\mathbf{X})$	$\triangleq$	Utility function of switch a.
d	$\triangleq$	Maximum degree of nodes in the network.

We consider a traffic matrix $\mathbf{T(n)}$, where $T_{xy}(n)$ entry in $\mathbf{T(n)}$ denotes the amount of traffic in Gbps from node x to node y at discrete time n. We assume that the traffic matrix of the network is variable in time; however, for short time intervals (between each execution of our proposed algorithm), we assume the changes are not significant and their effects can be taken into account by considering small deviation values ($\pm\delta$) based on traffic prediction models. Discussions on such models are out of the scope of this paper but their effects can easily be incorporated to the model.

In order to keep track of the energy cost due to the set of active links in the network, we define the energy price of active link, say ℓ_i, connected to switch x, at discrete time n by $e^x_i(n)$. Then, the total energy cost of active link, ℓ_i is defined to be $e_i(n) = e^x_i(n) + e^y_i(n)$, where x and y are the two nodes at the two ends of the link. Note that variable $e^x_i(n)$ allows us to incorporate the effects of various energy-pricing signals at switches located at different geographical regions as well as the variable availability of renewable resources that change the price of energy at switches. Similarly to the traffic variability, we assume that variations in the values of $e_i(n)$ in short time intervals between execution of our algorithm are not significant; however, to capture the effect of changes in energy resources and prices we apply our proposed algorithm repeatedly over fix intervals (or when a significant change in the energy resources is reported to the SDN controller). Hereafter, to simplify the notation we drop the time label n from these variables and focus on a single time interval.

The goal here is to find which path to keep active for the communication among nodes in the network to minimize the energy cost while satisfying the traffic demand and respecting the capacity of the links and energy prices. We assume that even if the traffic entry in the traffic matrix from node x to node y is zero at a point of time, i.e, $T_{xy} = 0$, we still need to keep at least one path from node x to node y for possible future communications (i.e., small variations in traffic, i.e., δ).

We use the binary variable X_i to denote if link ℓ_i is active (when $X_i = 1$) or inactive (when $X_i = 0$). In summary, the goal is to identify vector $\mathbf{X} = (X_1, X_2, ..., X_L)$, representing the link statues in the SDN-enabled network. In order to keep track of which path is active for communications from node x to node y, we define function Q^k_{xy} that returns 1 if the k-th shortest path from node x to node y is active and returns 0 otherwise. The definition of $Q^k_{xy}(\mathbf{X})$ is given by

$$Q^k_{xy}(\mathbf{X}) := \prod_{s=0}^{k-1}(1 - Q^s_{xy}(\mathbf{X})) \prod_{i=1}^{D(\mathbf{P}^{(k)}_{xy})} X_{\mathbf{P}^{(k)}_{xy}(i)}, \qquad (1)$$

where we assume that $Q^0_{xy}(\mathbf{X}) = 0$. This definition is based on the idea that all the shorter paths than k-th shortest path should be inactive before the k-th shortest path can be active. This is because at each time only one path is active and we prefer the shorter paths over the longer ones.

III. Optimization Formulation

We cast the interactions among all the parameters introduced in Section II in an optimization framework as follows. The objective of this optimization is to minimize the energy cost of the SDN-enabled network based on the set of active links while considering various energy prices at switches, traffic demands, link capacity limitations, and the connectivity constraints.

$$\underset{\mathbf{X}}{\text{minimize}} \quad \sum_{i \in \mathcal{L}} X_i e_i$$

$$\text{subject to: } (1) \sum_{s=1}^{|R_{xy}|} Q^s_{xy}(\mathbf{X}) = 1, \text{ for all } x, y \in V \text{ and } x \neq y$$

$$(2) \sum_{x,y \in V} (T_{xy} + \delta) \Big(\sum_{s=1}^{|R_{xy}|} Q^s_{xy}(\mathbf{X}) g(\mathbf{P}^{(s)}_{xy}, k) \Big) \leq X_k B_k, \text{ for all } k \in \mathcal{L}$$

$$(3) X_i \in \{0, 1\}, \text{ for all } i \in \mathcal{L}.$$

This optimization is over vector $\mathbf{X}$, which specifies the link statuses in the network. Constraint (1) is to ensure that there will be at least one path among all the nodes in the network and that the network of switches are connected. Constraint (2) limits the solution to the space for which the capacity constraints of all the links in the network are respected according to the traffic matrix of the system and the set of active links. Based on Constraint (2), we define function $W_i(\mathbf{X}) = X_i \Big(\sum_{x,y \in V} (T_{xy} + \delta) \big(\sum_{s=1}^{|R_{xy}|} Q^s_{xy}(\mathbf{X}) g(\mathbf{P}^{(s)}_{xy}, k) \big) \Big)$, which specifies the amount of traffic that goes through link ℓ_i. This problem is a nonlinear programming problem due to nonlinearity of Q^k_{xy} in terms of X_is. Although the problem is solvable by nonlinear programming solvers, it is not computationally efficient for large networks; specifically that due to dynamic traffic demands and energy prices it needs to be solved repeatedly. To solve this computationally complex problem, we propose a heuristic algorithm that can provide a sub-optimal solution to this problem more efficiently.

Algorithm 1 Energy- and Traffic-aware Active Link Selection

```
X = (1, 1, ..., 1).        ▷ Initially all the links are active.
Sort switches based on their energy price (e.g., electricity
  price and availability of renewable resources at switches)
  and save them in vector Z.
WhileFlag=true;        ▷ A flag to test if the while
  loop should be executed or not depending on the changes
  in X. We set it to be true initially to enter the loop.
while WhileFlag do
    WhileFlag=False;
    for i:=1 to N do
        c:= the i-th node in Z; ▷ c is the current switch to be
          decided about its link states.
        Calculate U_c presented in (2) based on current X.
        for j:=1 to # of active links connected to c do
            Calculate U_c for current X except that j-th active link
              of switch c is powered off.
        end for
        Find the best utility value among the calculated U_c
          values and power off at most one link connected
          to switch c based on the scenario corresponding to
          the best U_c value and update X accordingly. If "no
          change" in the status of links is the best U_c value do
          not change X.
    end for
    if There is a change in X then
        WhileFlag=True.
    end if
end while
```

IV. Heuristic Algorithm

In this section, we present a heuristic algorithm that identifies a vector $\mathbf{X}$ that satisfies the constraints of our optimization problem defined in Section III while reducing the energy cost of the network. This heuristic algorithm is described with the pseudocode presented in Algorithm 1.

At the beginning all the links in the network are assumed to be active. We assume that the initial network is connected and is able to satisfy the traffic demand based on the capacities of the links. In this algorithm, the SDN controller selects the switches one by one based on their availability of renewable resources and energy prices. More specifically, switches with less available renewable resources and higher energy price are selected earlier. For each switch the SDN controller decides about powering off at most one link of the switch. We name the latter sequence of decision steps for all the switches the *link selection process*. The algorithm repeats the link selection process until there is no change in the state of the links for the entire switches in the network. To make the decision about powering off a link at each step of the link selection process, the SDN controller defines a utility function for each switch, say switch a, based on the state of the links in the network as following:

$$U_a(\mathbf{X}) = \sum_{i \in \mathcal{L}} Z(X_i)(X_i B_i - W_i(\mathbf{X})) \\ -\alpha \prod_{x,y \in V, x \neq y} (\sum_{s=1}^{|R_{xy}|} Q^s_{xy}(\mathbf{X})) - \beta \sum_{i \in \mathcal{L} \text{ s.t. } \ell_i \text{ is connected to node } a} X_i e^a_i, \tag{2}$$

where function $Z(X_i)$ returns integer γ if $(X_i B_i - W(X_i)) \geq 0$ and integer μ otherwise. In the above utility function, the first term addresses the traffic load in the links with respect to their capacities. The less loaded the links are the larger is the utility (i.e., the amount of empty capacity will be multiplied by γ); however, if a link is overloaded then $(X_i B_i - W(X_i))$ will be negative and there will be a large penalty (i.e., μ) and the utility function will be reduced. The second term in (2) ensure that the links' states satisfy the connectivity requirements of the network (at least one path from R_{xy} is active between every nodes x and y). If there is any violation in this condition then there will be a large penalty (i.e., α), which is larger than the penalty associated with violating the links' capacity limits (i.e., $\mu < \alpha$). Furthermore, based on the third term in (2), the energy cost of active links at a switch also reduces its utility function according to the coefficient β. In summary, parameters α, β, γ, and μ represent large, positive integer coefficients with the following relationship: $0 \leq \gamma < \beta \ll \mu \ll \alpha$. Note that the first term in (2) is not contradicting the idea of packing traffic to small number of links because there is a trade-off between this term and the cost of active links presented in the third term, which allows balancing the distribution of traffic among links while powering off as many links as possible. The effect of coefficients and their relations as explained above is very important in obtaining the desirable distribution. At each step of the selection process, the SDN controller selects one of the active links of a switch to power it off if such decision maximizes the utility of the switch. If "no change" in the state of the links results in higher utility than powering off one link, then the controller will not power off any of the links for the specific switch.

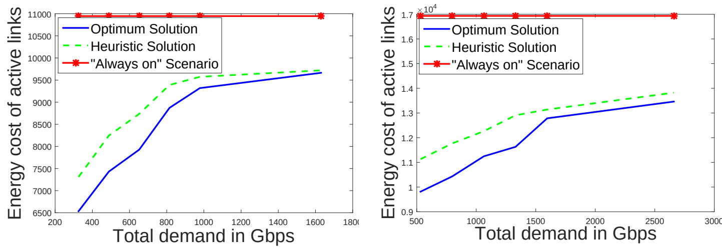
If we denote the maximum degree of nodes in the network by d, then the computational complexity of this algorithm is $O(NlogN + (L-N)N.d)$. The $NlogN$ term is for sorting the nodes based on their energy prices and available renewable resources at the beginning of each execution. The second term in the above complexity will be in the order of $O(N)$ for a sparse network (i.e., L being in the order of number of nodes in the network) and since most of the real-world networks are sparse this algorithm is computationally efficient for solving the proposed optimization problem. The SDN controller calculates $\mathbf{X}$ periodically to address the dynamics of traffic demands as well as the energy prices and renewable resources at various locations. The desirable properties of the heuristic algorithm, including its convergence properties and its constraint-satisfying solutions, can be verified by mapping the problem to a complete information, ordinal potential game model [15], [16], which the details of such mapping and the proof of its features are out of the scope of this paper. In this paper, we rely on our simulation results to show the effectiveness of the proposed heuristic. An important point to notice is that the success of this heuristic depends on the availability of global information at the controller (an SDN feature).

V. Evaluation

To evaluate the performance of the proposed model, we consider a set of simulation scenarios. In these simulations, we consider two network topologies; the NSF network shown in Fig. 1 and the Abilene network as presented in [17]. For each of these networks, we pre-calculate 10 shortest paths for each pair of switches. To do so, we assign weights to the links of the network based on their physical length (which, for example, can affect the delay as well as the energy cost due to the energy consumption of the communication repeaters for long distances). As these networks represent backbone topologies, we assume that the links of the network bundle multiple physical links and their capacities are one of the values from $\{100\text{Gbps}, 200\text{Gbps}, 400\text{Gbps}\}$; however, all the bundled links are considered as a single link in our system model. We synthesize the traffic matrix of the networks similar to the data sets available in SNDlib [17]. In particular, we randomly select traffic flows from the SNDlib data sets and assign them to random pairs of nodes in our networks. We call this traffic matrix the *base traffic matrix* with the total demand of 532Gbps and 326Gbps for NSF and Abilene networks, respectively. The base traffic matrix models a light load on the network. To represents various traffic loads, we scale the base traffic matrix by factors 1.5, 2, 2.5, 3, and 5. We assign capacities to links in a way that even in the peak traffic the most loaded links in the network are at most $\%90$ loaded.

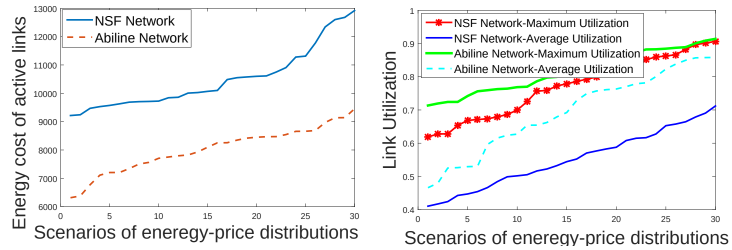
In order to model the energy cost of active links, we generate two energy price values for each link of the network (for its two end points), i.e., we generate an energy price vector $\mathbf{E}$ with length $2L$. The energy price of the links at switch x are generated based on $a^x + b^x_i$, where a^x is a random number representing the common energy price of the links at node x and b^x_i is a random number for each link ℓ_i representing the variations in the energy price of different links at the switch x due to, for example, their length. We assume that $a^x \in [100, 400]$ and $b^x_i \in [1, 50]$ and they correspond to the energy consumption of line cards in Watt (W). Finally, in our simulations we set the value of the parameters in (2) to be $\gamma = 5$, $\beta = 50$, $\mu = 500$, and $\alpha = 50,000$.

In Fig. 2, we present our results on the energy cost of active links in the two networks for the case of optimal solution (derived by APMonitor with MATLAB interface [18]), heuristic solution, and "always on" solution (the case that all the links are active). Clearly, the traffic load does not affect the energy cost of the "always on" scenario. In addition, we observer that as the traffic increases so does the energy cost of the network, i.e., it is more difficult to save energy as the network load increases. Moreover, the results based on the proposed heuristic algorithm lead to energy costs, which in the best case



(a) (b)

Fig. 2. Total energy cost of active links in the network for (a) Abilene network and (b) NSF network.



(a) (b)

Fig. 3. (a) Total energy cost of active links, and (b) bandwidth utilization of active links, for various scenarios of energy-price distribution for switches using the heuristic algorithm. Scenarios are sorted based on the energy cost.

(for the least traffic load in both networks) are about 34.27% and 33.22% less than the cost of the "always on" scenario for NSF and Abilene networks, respectively. Also, in the worst case, i.e., for the least loaded cases, the energy cost of heuristic solution is 13.4% and 12.1% more than the optimal solution for NSF and Abilene networks, respectively. As the traffic increases the resulted energy cost of the heuristic algorithm approaches the optimal solution. Besides the decrease in the energy cost, it is important to mention that all the resulted solutions from the heuristic algorithm in our experiments have met the connectivity and capacity constraints of the network.

In the next study, we use the same setting as before except that we generate multiple random $\mathbf{E}$ vectors representing different energy-price distributions at various locations for switches. Based on the heuristic results in Fig. 3-a, we observe that the energy-price distribution affects the energy efficiency of the solution. Note that the results in Fig. 3 are sorted based on the resulted energy cost for different scenarios. Moreover, the results presented in Fig. 3-b show that the energy-price distribution at switches also affect the traffic distribution in the links and the bandwidth utilization. Therefore, considering the energy resources and prices at switches largely affect the optimum set of active links, the energy cost, and the bandwidth utilization of the network.

VI. Conclusions

Software Defined Networking (SDN) paradigm offers new opportunities to improve various aspects of efficiency and reliability of ICT networks. In this paper, we specifically exploited the availability of the global view of the network at the logically centralized SDN controller and the programmability of SDN switches to propose an energy-saving mechanism for infrastructure networks. To do so, we formulated an optimization framework for identifying the optimum set of links to be powered off to reduce the energy cost of the network while satisfying the traffic demand and respecting the communication and energy resources of the network. We particularly discussed that the price of energy as well as the availability of energy resources varies for switches in different geographical locations and it is important to be considered in designing energy-saving mechanisms. Finally, we proposed a heuristic algorithm to solve the optimization problem in a computationally efficient way and presented the results using the proposed algorithm. In the future work, we will prove convergence and desirable properties of the heuristic algorithm using a complete information, ordinal potential games and implement it as an SDN application in Mininet (an SDN emulator) [19] to evaluate the model in more realistic settings.

References

[1] S. Lambert and et al., *Worldwide electricity consumption of communication networks*, Optics express, vol. 20, no. 26, pp. B513B524, 2012.

[2] M. Erol-Kantarci and H.T. Mouftah, *Energy-Efficient Information and Communication Infrastructures in the Smart Grid: A Survey on Interactions and Open Issues*, IEEE Communications Surveys Tutorials, vol. 17, no. 1, pp. 179-197, 2015.

[3] B. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, *A survey of software-defined networking: Past, present, and future of programmable networks,* IEEE Communications Surveys Tutorials, vol. 16, no. 3, pp. 16171634, 2014.

[4] H. Kim and N. Feamster, *Improving network management with software defined networking*, IEEE Communications Magazine, pp. 114119, 2013.

[5] B. G. Assefa and O. Ozkasap, *State-of-the-art Energy Efficiency Approaches in Software Defined Networking*, ICN: The Fourteenth International Conference on Networks, 2015.

[6] F. Giroire, J. Moulierac, and T. Khoa Phan, *Optimizing Rule Placement in Software-Defined Networks for Energy-aware Routing,* Globecom, Symposium on Selected Areas in Communications, 2014.

[7] A.Markiewicz, N. T. Phuong, and A. Timm-Giel, *Energy consumption optimization for software defined networks considering dynamic traffic,* IEEE 3rd International Conference on Cloud Networking (CloudNet), pp.155,160, 2014.

[8] D. Staessens, S. Sharma, D. Colle, M. Pickavet, and P. Demeester, *Software defined networking: Meeting carrier grade requirements,* in Local Metropolitan Area Networks (LANMAN), 18th IEEE Workshop on, Oct, pp. 16, 2011.

[9] L. Chiaraviglio, M. Mellia, and F. Neri, *Minimizing ISP Network Energy Cost: Formulation and Solutions,* In IEEE/ACM Transaction in Networking, pp. 463 476, 2011.

[10] L. Chiaraviglio, M. Mellia, and F. Neri, *Reducing Power Consumption in Backbone Networks*, IEEE International Conference on Communications, 2009, ICC '09, 14-18 June 2009,pp 1 - 6.

[11] F. Idzikowski, S. Orlowski, C. Raack, H. Woesner, A. Wolisz, *Dynamic routing at different layers in IP-over-WDM networks Maximizing energy savings*, Optical Switching and Networking, Special Issue on Green Communications and Networking, vol. 8, no. 3, pp. 181-200, 2011.

[12] Y. Shang, D. Li, and M. Xu, *Energy-aware routing in data center network,* In 1st ACM SIGCOMM Workshop on Green Networking, 2010.

[13] F. Giroire, J. Moulierac, T. K. Phan, and F. Roudaut, *Minimization of Network Power Consumption with Redundancy Elimination*, In IFIP Networking, 2012.

[14] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, *Openflow: Enabling Innovation in Campus Networks,* In ACM CCR, pages 69 74, 2008.

[15] J.D. Monderer and L. Shapley, *Potential games. Games and Economic Behavior*, vol. 14,no.1,pp. 124-143,1996.

[16] M. Rahnamay-Naeini and M. Sabaei, *A Combinational Perspective in Stimulating Cooperation in Mobile Ad Hoc Networks,* Journal of Computer Science and Technology, V26(2): 256-268, 2011.

[17] S. Orlowski, M. Pióro, A. Tomaszewski and R. Wessäly, *SNDlib 1.0–Survivable Network Design Library,* Networks Journal, vol. 55, no. 3, 2010.

[18] J. D. Hedengren, APMonitor optimization suite, [Online] available at: http://apmonitor.com, 2014.

[19] B. Lantz, B. Heller, and N. McKeown, *A network in a laptop: Rapid prototyping for software-defined networks,* in HotNets, Oct 2010.