Green Traffic Engineering in SDN

Roya Razmnoush
Computer Engineering And Information Technology
AmirKabir University Of Technology
Email: razm.roya1388@aut.ac.ir

Bahador Bakhshi
Computer Engineering And Information Technology
AmirKabir University Of Technology
Email: bbakhshi@aut.ac.ir

Abstract— **Reducing the power consumption of the internet backbone networks is one of the major challenges in computer networks because of high energy cost and environmental issues. SDN is the promising technology that enables fine grain control of network traffic. In this paper, we propose a novel traffic management and routing framework, called SDGTE (Software Defined Green Traffic Engineering), that minimizes the number of turned on links and nodes to reduce the network power consumption. Differently from the approaches that have already been proposed, our solution aims to adapt the network power consumption according to dynamics of network traffic in *online* manner *without prior knowledge* of the future traffic matrices. SDGTE is able to run in real time due to low overhead where the prior exact information of traffic would not be available. Results obtained on realistic case studies show that SDGTE saves enerfy consumption up to 70% while maintaining at most 85% utilization of network links.**

Keywords-Green networks, Online routing, energy-aware network management, energy-aware routing

I. INTRODUCTION

As [1] mentioned, the carbon footprint of Information Communication Technologies (ICTs) is constantly increasing, representing today up to 10% of the global CO_2 emissions. Among the main ICT sectors, 37% of the total emissions are due to telecommunication infrastructures and their equipment, while data centers and personal devices are responsible for the remaining part. It is therefore not surprising that researchers, manufacturers and network providers are spending significant efforts to reduce the power consumption of ICT mechanisms from different angles of green networking [1].

In recent years, most of the researches and works have published in the area of green networking. IP networks can be made greener by working at different levels: by developing novel energy-efficient equipment and architectures which offer support for sleeping primitives or *pipeline IP forwarding*, by defining local procedures to autonomously adjust the state of a single network device according to real-time measurements, and by coordinating the management of the whole network infrastructure to optimize both energy consumption and performance of both routing and device configurations called *energy-aware network traffic management* (EANM) approaches [2].

Practically speaking, the goal of EANM is to adapt the energy consumption to traffic levels. The way EANM is performed related to the following elements: (i) the power profile of network devices, (ii) the routing protocol, according to which traffic engineering is performed, (iii) the Quality of Service (QoS) requirements requested by each traffic flow, (iv) when and how frequently network re-configurations should be performed, i.e., in advance off-line, on-line or real time, (v) where to locate the intelligence of the system (centralized or distributed architecture) and finally, (vi) network survivability in case of failures [2].

While a few previous work considered the EANM problem, conventional online methods had two serious weaknesses. First they are unable to work in absence of traffic matrix; since it is needed as input of the approaches. Second, since they periodically reroute whole traffic flows to adapt routing according to traffic matrix, the approaches are time consuming and apply a great deal overhead. The online approach proposed in [3] performs routing with the knowledge of traffic matrix in various daily periods and in fact it is not a real time approach capable of following traffic variations. The online approach proposed in [4] changes the entire network routing in every time slot, and cause a very large overhead. Differently from the existing works, in this paper we considered online EANM methods which are able to manage the network traffic in real time where prior exact information of traffic would not be available.

Software Defined Networking (SDN) is an emerging networking paradigm that separates the network control plane from the data forwarding plane with the promise to dramatically improve network resource utilization, simplify network management, reduce operating cost, and promote innovation and evolution. Although traffic engineering techniques have been widely exploited in the past and current data networks, such as ATM networks and IP/ MPLS networks, to optimize the performance of communication networks by dynamically analyzing, predicting, and regulating the behavior of the transmitted data, the unique features of SDN require new traffic engineering techniques that exploit the global network view, status, and flow patterns and characteristics available for better traffic control and management. [5, 6]

In this paper, we consider the problem of optimizing the energy consumption in software defined backbone networks and develop a new online and centralized EANM approach called Software Defined Green Traffic Engineering (SDGTE) for routing and traffic management in order to minimize the energy consumption of backbone networks. The approach doesn’t have any knowledge of traffic matrix or traffic variations and it keeps the links utilization between two underutilization and overutilization threshold. For the routing part, we implement a real-time green routing protocol that routes the flows requests one by one in the network without any knowledge of flow rate variations, and for the monitoring part, we propose a green monitoring approach that monitor the traffic network in the specific periods and keep the links utilization below a threshold by rerouting the flows in the network, if flow rate changes over the time and sometimes unpredictably. We have gotten help from Software-Defined Networking (SDN) to create a global view of the network and gathering statistical information from network devices.

The reminder of the paper is organized as follows. In Section II we review previous papers on green networking and point out the novelties of our work. In Section III we present the energy management mechanism and describe the system model and mathematical formulation. A set of results obtained on different network topologies are shown and discussed on Section IV. Finally concluding remarks follow in Section V.

II. RELATED WORK

The problem of energy consumption of Internet and the main technical challenges to reduce the energy consumption have presented in the seminal work by Gupta and Singh [7]. In the last years several approaches have proposed to cope with the green problem in wired IP networks.

Proposed Methods for online network management are presented in [1] [3, 4] [8-10]. In [1] a distributed approach with multiple agents that take independent decisions based on local or global data is presented, this approach isn’t able to save energy as much as centralized approaches, because it’s distributed and all the agents must be turned on all the time to take management decisions. In centralized approach [3], the authors propose a MILP-based on-line heuristic to put to sleep line cards and chassis while respecting a limitation on the number of switching-on allowed to each line card along an entire day. The method considers networks operated with MPLS and requires the knowledge of the instant traffic matrices. [4] Considers network operated with a hybrid OSPF+MPLS routing protocol and exploits a restricted path MILP formulation to put to sleep network links. Vasi´c et al. proposes EATe [8] to reduce Internet power consumption through traffic engineering. EATe considers sleeping of links and routers as well as link rate adaption. EATe achieves its routing decisions in a distributed fashion via router coordination and thus requires routers to be able to send announcement and feedback to each other.

To the best of our knowledge no previous work considers online and dynamic approach for EANM that can track the demand variations without any knowledge of past and future traffic matrix, and be able to process the flows one by one and find a green path for them. Some of the previous online EANM works [3, 9] consider a daily traffic matrix or consider a multi-period traffic matrix and solve the routing problem with the knowledge of traffic matrix. Some of the previous online EANM works, get the flow information and it’s QoS constraints as input when the flow arrives to the network [10], one of the previous online EANM approaches, GreenTE [4], solves a MILP problem in specific periods and changes all the routing in every period, GreenTE has a very large overhead and it solves a MILP which is a NP-hard problem.

III. THE SDGTE

In this section we develop our solution, the SDGTE framework, for the EANM problem. First, Section A states our system architecture. Second in Section B, the monitoring functionality is presented with more details. Third, in Section C, we develop an ILP model for rerouting flows in the network. Finally, we have proposed a model for routing new input flows of network in Section D. Table I clarifies the notations used through this paper.

TABLE I. SUMMARY OF NOTATION USED IN THIS PAPER

Notation	**Meaning**
V	Set of nodes in network
E	Set of links in network
H	Set of hosts (traffic sources or destinations) in network
N	Set of ON nodes in network
L	Set of ON links in network
(i,j)	The link between i and j nodes
$R(i,j)$	Remaining link capacity of the link (i,j)
$C(i,j)$	Link capacity of the link (i,j)
α	High Threshold
β	High threshold , $\alpha>\beta$
δ	Low Threshold
s	Demand source
t	Demand destination
d	Traffic demand from s to t
$X(i,j)$	1 if traffic demand from s to t routed through link (i,j), 0 otherwise.
$P(i)$	1 if node i is ON, 0 if the node i is sleep.
$Q(i,j)$	1 if link (i,j) is ON, 0 if the link (i,j) is sleep.

A. System Architecture

The SDGTE framework composed of three main modules: Monitoring module, Routing module and Rerouting module. (Figure 1). As shown in the figure, all these 3 modules place in the SDN controller.

The Monitoring module monitors the network in predefined time slots (e.g., 5 minutes monitoring slot). This module collects link states of the network besides using the counters which are maintained per-flow in SDN switches. This module enforces specific policies to identify a list of flows which must be rerouted in the network. Afterwards the identified list is given to the Rerouting module. When a flow is given to this module to be rerouted, the Rerouting module finds a new route for the flow by solving an ILP (Integer Linear Programming) problem. The problem is elucidated in section B with more details.

When a new flow request enters the network, the controller gives the request to the Routing module. Since essential information about rate and bandwidth of this flow is unknown, the Routing module performs routing with estimation of flow bandwidth. This module also solves an ILP problem to find a green route for the flow. This problem is described in section C.

SDGTE is completely centralized, it uses a controller which gathers the flows' rate variations from the routers and uses this information in *Under_Utilized_Check* and *Over_Utilized_Check* functions to determine flows list for rerouting. We use OpenFlow protocol for gathering flows' bandwidth, update routing tables .The controller implements the Routing and Monitoring modules.

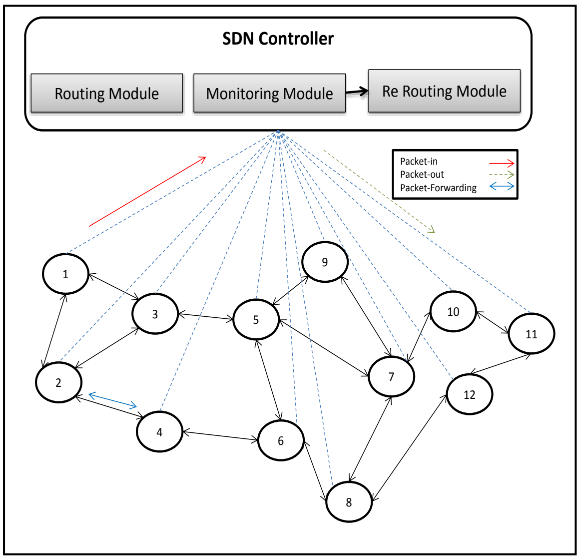

Figure 1. SDGTE Architecture

B. Monitoring Module

Figure 2 shows pseudo-code of Monitoring module. The monitoring module not only evaluates the network but also implements two functions, *Under-Utilized-Check* function (lines 1-8) and *Over-Utilized-Check* function (lines 10-17).

The module uses the SDN capabilities to obtain the flows information. The module set a counter for each flow, when the flow arrives to the network. This counter enables the Monitoring Module to obtain the flow required bandwidth in every moment.

The *Over-Utilized-Check* function checks whether link utilization exceeds a predefined value α and if so flags the link as over-utilized then some flows on the link are rerouted by the Rerouting modules in order to reduce the link utilization.

The *Under-Utilized-Check* function checks whether link utilization is less than a predefined value δ and if so flags the link as under-utilized, then all flows on the link are rerouted and then the link is switched off).

C. ILP model for flow Rerouting

For Rerouting a flow in the network, we have developed an ILP model, which is given the network topology, flow request, and current network devices status, then finds an energy efficient path for the flow.

We have modeled the network as a directed graph $G = (V, E)$, where V is the set of nodes (i.e., routers) and E is the set of links. A link can be put to sleep mode, if there is no traffic on it and a router can be put to sleep mode if all the links connected to it are asleep. A router composed of a chassis and a set of line cards that enable it to connect to other devices.

The inputs of the ILP problem are: network topology, remaining links capacity, network status and the flow request information (s: flow source, t: flow destination, d: traffic demand from s to t). Note that in rerouting phase, the traffic demand is estimated using the pre-flow counters in SDN switches. The Rerouting module get the flow bandwidth (d) from the Monitoring module, which obtain this bandwidth using OpenFlow .The objective of this ILP problem is to find a path for the flow in the network that minimizes the power consumption of network and puts as much links and nodes in sleep mode as possible, besides keeping the links utilization under the threshold α. The ILP model is as follows where the $X(i,j)$, $P(i,j)$ and $Q(i,j)$ are the binary variables of the ILP problem.

$$min \sum_{i \in V} P(i) + \sum_{(i,j) \in E} Q(i,j) \qquad (1)$$

$$\text{s.t. } \sum_{(i,j) \in E} X(i,j) - \sum_{(j,i) \in E} X(j,i) =$$

$$\begin{cases} 1 & i = s \\ 0 & i \neq s,t \\ -1 & i = t \end{cases} \quad \forall i \in V \qquad (2)$$

$$P(i) \geq Q(i,j) \quad \forall (i,j) \in E, \forall i \in V \qquad (3)$$

$$P(i) \geq Q(j,i) \quad \forall (j,i) \in E, \forall i \in V \qquad (4)$$

$$P(i) = 1 \quad \forall i \in H \qquad (5)$$

$$P(i) = 1 \quad \forall i \in N \qquad (6)$$

$$Q(i,j) = 1 \quad \forall (i,j) \in E \qquad (7)$$

$$R(i,j) - X(i,j) * d \geq (1 - \propto) * C(i,j) \; \forall (i,j) \in E \qquad (8)$$

Equation 1 is the objective function that computes the total power consumption in network. Equation 2 states the flow conservation constraints. Equations 3 and 4 ensure that a node is put to sleep only when it's entire connected links are asleep. Equation 5 ensures that a host node (traffic generator) is on all the time. Equation 6 ensures that the turned on nodes must remain turned on after routing the given flow because they are passing traffic or are the traffic hosts. Equation 7 ensures that turned on links must remain turned on after routing the given flow because they there are traffic on them. Equation 8 states that MLU must be no greater than a configured threshold α.

D. *ILP model for routing new flows*

When a new flow enters the network, we have no information about its required bandwidth, so the Routing module estimates the flow information for routing process. In order to find a green route, the Routing module solves an ILP problem similar to the ILP problem in section B; the only difference is that constraint (8) is substituted with (9).

$$R(i,j) \geq (1-\beta)*C(i,j) \quad \forall (i,j) \in E \tag{9}$$

This constraint delineates the fact that link utilization before routing this flow on it, must be under β).

The module estimation represents the fact that if a link utilization is no bigger than β, then this link utilization after rerouting would not exceed the threshold value α. This estimation is closely related to size of entered network flows.

```
Monitoring:

Function UnderUtilized_Check()
  FOR EACH (i,j) in E
    IF $R(i,j) \geq (1-\delta)*C(i,j)$ THEN
      WhILE THERE IS FLOW ON THE LINK
        Rereoute the flow(s,t,d)
      END WHILE
  END FOR
END Function

Function OverUtilized_Check()
  FOR EACH (i,j) in E
    WHILE $R(i,j) \leq (1-\alpha)*C(i,j)$ THEN
      Select flow(s,t,d) from link
      Rereoute the flow(s,t,d)
    END WHILE
  END FOR
END Function
```

Figure 2. Pseudo-code of Monitoring

IV. EVALUATION

In this section, we evaluate the SDGTE framework. First in section A we explain our experiment environment, assumptions and our traffic model, besides the topology and energy model. Second in section B, we show our power savings results, third in section C, the obtained MLU is discussed, finally the complexity and overhead issues are discussed in section D.

A. *Expriment setup*

In order to implement our algorithm, we developed an event-based simulator with java programming language. As well as, we use *gurobi* solver [11] for solving the ILP problem. We compared our algorithm results with GreenTE [4], furthermore we also consider OSPF protocol as the common routing protocol in networks for evaluation. In this paper we use 20%, 85% and 95% as the default values respectively for δ, β and α, thresholds.

1) *Topology*

Two different network topologies were chosen, respectively with 12 nodes (see Figure 3) and 25 nodes (France Topology). The larger topology (France Topology) belongs to the largely used SND-Library [12].

These topologies vary in size as listed in table II. The 12-nodes topology will be used for a comprehensive understanding of the results, while the other topology confirms the findings for larger and more realistic networks. There are two kinds of functionality for nodes, some of them are just router and the others could have traffic generator and router functionality.

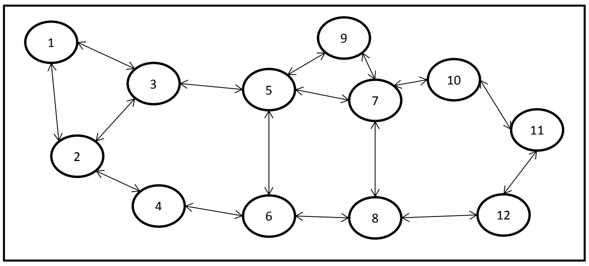

Figure 3. Network with 12 nodes

2) *Traffic demands*

We were interested in evaluating the possible power saving that could be obtained in a realistic scenario. For this purpose, we considered the realistic traffic matrix in the simulations which varied according to a day-night patterns (see Figure 4).

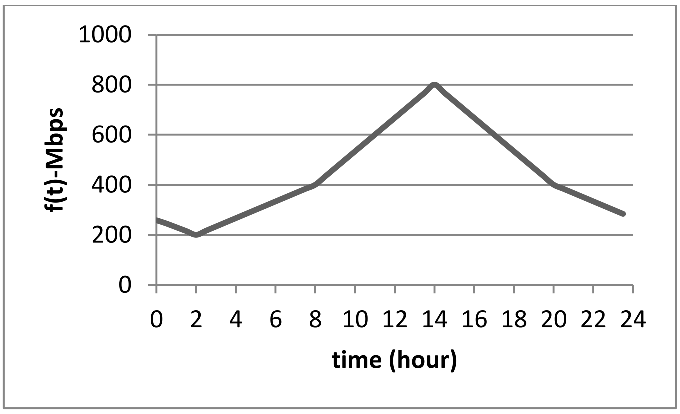

Figure 4. Actual profile of traffic in backbone networks

3) *Energy model & Network capacity*

We use the energy model presented in [3] in our simulations. This energy model assumes that a router is composed of a chassis and a set of line cards. A chassis can be put to sleep if all of its line cards are asleep. The chassis and cards capacity and consumption are provided in Table III.

B. *Power saving*

We explore the power saving under SDGTE using two different network topologies and traffic matrixes. We compute the power consumption of the network as the total power that the links and nodes consume when they are on.

The power consumption results for small topology (12 nodes) are shown in Figure 5. We compared the total percentage of power consumption of SDGTE with GreenTE.

According to our results, in average, SDGTE saves about 60% power, in addition SDGTE outperforms better than GreenTE all the time.

The power consumption results for France topology are shown in Figure 6. We compared the total percentage of power consumption of SDGTE with GreenTE. According to our results, in average, SDGTE saves about 75% power, in addition SDGTE outperforms better than GreenTE all the time.

TABLE II. NETWORK TOPOLOGIES USED IN THIS EVALUATION

Network	Nodes	Links (Line Cards)	Hosts
Small	12	32	6
France	25	90	14

TABLE III. ROUTER CHASIS AND CARDS

	Capacity	hourly Power Cons.
Chasis	16 Gbps	86.4
Cards	1 Gbps	7.3

C. *Link utilization*

We explore the maximum link utilization (MLU) under SDGTE using two different network topologies and traffic matrixes. We compute the MLU of the network as the maximum occupied links capacity at the moment.

The MLU results for small topology (12 nodes) are shown in Figure 7. We compared the MLU of SDGTE with GreenTE and OSPF. According to our results, in average point of view, SDGTE MUL is under the 95% of links capacity and it's near to GreenTE MLU.

The MLU results for France topology are shown in Figure 8. We compared the MLU of SDGTE with GreenTE and OSPF. According to our results, in average point of view, SDGTE MUL is under the 85% of links capacity and it's near to GreenTE MLU.

D. *Complexity & overhead*

We explore the SDGTE rerouting overhead using the total number of flows that reroute in any 30 minutes. More flow reroute cause more overhead in message passing and packet loss. Our results show that GreenTE reroutes 2 times flows more than SDGTE in every 30 min for small (12 nodes) topology. And GreenTE reroutes 10 times flows more than SDGTE in every min.

We show that the SDGTE complexity is very smaller than GreenTE, since GreenTE change all the network routing in every 5 min using a NP-hard MLIP which takes time to solve, but SDGTE solve the ILP problem for just one flow and just reroute some of the flows in monitoring periods and not all the traffic matrix.

Since, our ILP problem is solved in a very smaller time than GreenTE MILP, so our algorithm can applied to real topologies with very small overhead and packet loss.

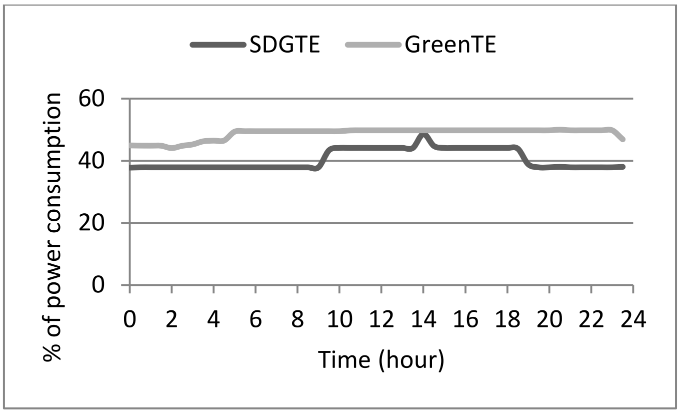

Figure 5. Power consumption percentage of 12 nodes topology (Wat)

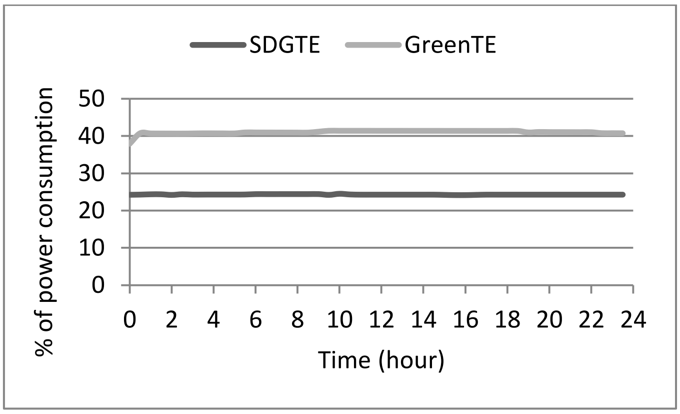

Figure 6. Power consumption percentage of France topology (Wat)

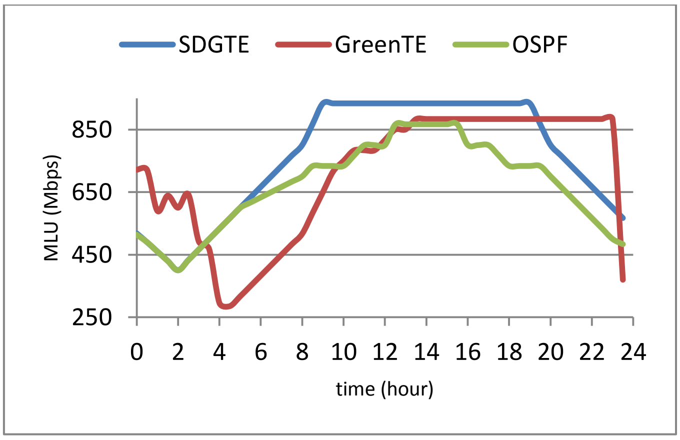

Figure 7. MLU of 12 nodes topology (Mbps)

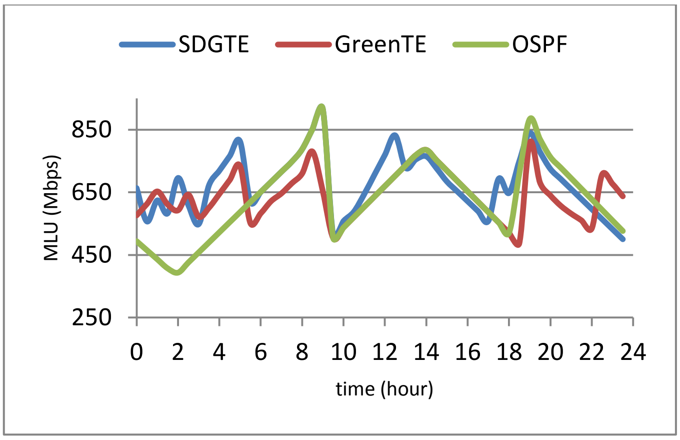

Figure 8. MLU of France topology (Mbps)

V. CONCLUSION AND FUTURE WORK

In this paper, we have presented a new online and centralized EANM approach called Software Defined Green Traffic Engineering (SDGTE) for routing and traffic management in order to minimize the energy consumption of backbone networks. The approach doesn't have any knowledge of traffic matrix or traffic variations and it keeps the links utilization between two underutilization and overutilization threshold. For the routing part, we implemented an ILP problem that finds a green path for flows. real-time green routing protocol that routes the flows requests one by one in the network without any knowledge of flow rate variations, and for the monitoring part, we proposed a green monitoring approach that monitor the traffic network in the specific periods and keep the links utilization below a threshold by rerouting the flows in the network, if flow rate changes over the time and sometimes unpredictably. We have gotten help from Software-Defined Networking (SDN) to create a global view of the network and gathering statistical information from network devices.

The evaluation results showed the proposed method to reduce 75% power consumption in comparison with traditional routing protocols (OSPF), and to reduce 15% power consumption in comparison with the online GreenTE mechanism. Also the studies showed that SDGTE achieves MLU fewer than 85%.

In the future works, we will introduce a new heuristic algorithm which is capable of solving the problem needless of optimization solver's support.

REFERENCES

[1] A.P. Bianzino, L. Chiaraviglio, M. Mellia, and J. Rougier. “Grida: Green distributed algorithm for energy-efficient ip backbone networks. ” Computer Networks, 56(14):3219–3232, June 2012

[2] Addis, B., et al. (2015). “Energy management in communication networks: a journey through modelling and optimization glasses.” arXiv preprint arXiv:1507.02636.

[3] B. Addis, A. Capone, G. Carello, L.G. Gianoli, and B. Sans´o. “Energy management through optimized routing and device powering for greener communication networks “. IEEE/ACM Transactions on Networking, 22(1):313–325, February 2014.

[4] M. Zhang, C. Yi, B. Liu, and B. Zhang. “Greente: Power-aware traffic engineering “. In The 18th IEEE International Conference on Network Protocols, pages 21–30. IEEE, October 2010.

[5] Openflow switch specification v1.0–v1.4 <https://www.opennetworking.org/sdn-resources/onf-specifications>.

[6] S. Agarwal, M. Kodialam, T. Lakshman, “Traffic engineering in software defined networks, ” in: Proceedings of the 32nd IEEE International Conference on Computer Communications, INFOCOM’13, April 2013, pp. 2211–2219.

[7] M. Gupta and S. Singh, “Greening the internet,” in Proc. ACM SIGCOMM, 2003, pp. 19–26.

[8] N. Vasi´c and D. Kosti´c. “Energy-aware traffic engineering “. In Proc. of the 1st International Conference on Energy-Efficient Computing and Networking, pages 169–178. ACM, 2010.

[9] A. Capone, C. Cascone, L.G. Gianoli, and B. Sans`o. “Ospf optimization via dynamic network management for green ip networks “. In 2013 Sustainable Internet and ICT for Sustainability (SustainIT), pages 1–9. IEEE,October 2013.

[10] S. Avallone and G. Ventre. “Energy efficient online routing of flows with additive constraints “. Computer Networks, 56(10):2368 – 2382, 2012.

[11] (2015). "Gurobi Optimization - The Best Mathematical Programming Solver." from http://www.gurobi.com.

[12] “SNDlib: Library of test instances for survivable fixed telecommunication network design,” 2006 [Online]. Available: http://sndlib.zib.de/home.action.