SDN as a Tool for Energy Saving

Irena Šeremet, Suada Hadžović, Saša Mrdović and Samir Čaušević

***Abstract* — Software Defined Networking (SDN) is considered as a promising solution for optimizing network energy consumption. This paper presents that experimental results show a significant amount of energy saving when using an SDN solution. This conclusion is based on results of energy consumption comparison in a real IP/MPLS network and designed SDN network.**

***Keywords* — Consumption, Efficiency, Energy, Internet, IP/MPLS, Network, Router, SDN, Switch, Telecommunications.**

I. Introduction

TRADITIONALLY, communication networks have been designed to handle peak traffic, thereby being over-provisioned in the low traffic period. Keeping that in mind, many studies have been focused on finding a solution to saving energy by turning off or putting in sleep some parts of the network while maintaining an operational network. This approach is known as Energy Aware Routing and it is inspired by the fact that network devices and links are often underutilized. The main principle is turning on or turning off network components such as ports or modules in devices based on the traffic load. The main challenge in this approach is defining which components to turn on or to turn off without disrupting traffic or aggravating level of Quality of Service.

While traditional networks are complex, hardly manageable and without the necessary flexibility, SDN allows the user to control the allocation of resources at a virtual level through the control plane. SDN does have high development potential for overcoming the traditional network disadvantages and it is expected to have a globally significant increase in numbers of transition from traditional network to SDN.

This article is structured as follows. After this introduction, Section II covers related work, Section III deals with a short IP/MPLS versus SDN comparison, Section IV is about problem statement while Section V presents one approach for energy efficiency improvement. Finally, in Section VI conclusion is drawn.

II. Related Work

Energy Aware Routing has led to numerous research in the network community. In [1], the authors minimized energy consumption under three constraints (delay, packet loss and jitter) using on/off technology in traditional wired networks. Their approach resulted in saving of 40% energy consumption, but at each turning off, a delay variation led to the loss of packets in a heavy traffic case.

The work of the authors in [2], was aimed at increasing the energy efficiency of core networks by providing an intra-domain SDN approach to selectively turn off a subset of links. Authors proposed the framework that dynamically adapts the number of active links to the traffic load. Experimental results showed that the implementation of the proposed algorithms reduced the consumption of 44% links by maintaining a good quality of service.

Contribution of [3] is an end to end validation of the proposed solution designed to reduce the energy consumption of the communication network within a single operator network domain. Validation was performed through simulation and implementation in a real network testbed comprising SDN switches and Open network Operating System SDN controller.

In [4], the authors presented four variants of algorithms that can optimize energy savings in SDNs following approaches of putting the idle devices into sleeping mode, prioritizing the existing active switches over the sleeping ones and increasing the number of idle devices.

SDN technology was used in [5], to suspend links when they are not used, taking into account congestion to maintain the quality of network. Implementation of this approach led to possibility that 33.33% of links and 87.5% of ports could be shut down to save energy.

The authors in [6], surveyed on energy efficiency and most of presented solutions are based on re-routing the flows in the network while minimizing the number of active switches.

The challenges brought by SDN and NFV(Network Function Virtualization) for the deployment of energy aware routing were the focus of [7]. Deployment of SDN devices alongside legacy devices was also studied.

III. IP/MPLS vs SDN

Every router in IP/MPLS network has its control and data plane. The control plane contains all the intelligence of the router and is responsible for maintaining connections and exchanging protocol information with other router or network devices. The data plane or the forwarding plane is responsible for the forwarding packets

Irena Šeremet is with University of Sarajevo, Faculty of Transport and Communications Zmaja od Bosne bb, 71000 Sarajevo, Bosnia and Herzegovina; (phone 387-61-379-771:, e-mail: irena.seremet.1@gmail.com).

Suada Hadžović is with Communications Regulatory Agency, Mehmeda Spahe 1, 71000 Sarajevo, Bosnia and Herzegovina (phone: 387-33-250600; e-mail: shadzovic@rak.ba).

Saša Mrdović is with University of Sarajevo, Faculty of Electrical Engineering, Zmaja od Bosne bb, 71000 Sarajevo, Bosnia and Herzegovina (phone: 387-33-250700; e-mail: smrdovic@etf.unsa.ba).

Samir Čaušević is with University of Sarajevo, Faculty of Transport and Communications Zmaja od Bosne 8, 71000 Sarajevo, Bosnia and Herzegovina; (phone:, e-mail: Samir.causevic@gmail.com).

978-1-7281-4790-1/19/$31.00 ©2019 IEEE

from the incoming interface to outgoing interface through the router.

SDN is an architecture, which decouples control plane and data plane leaving only data plane on network devices. Control plane, on the other hand, is placed on the separate device in the network - on the SDN controller. That way, centralized control over network devices is achieved. SDN controllers give the traffic-management instructions to network devices. SDN controllers are programmable and enable automation in the network. Energy optimization that can be applied at various levels of the SDN architecture, or SDN itself that can be used as a means of energy saving.

IV. Problem Statement

A part of the IP/MPLS network topology of a real service provider in Bosnia and Herzegovina is shown in Fig. 1. Topology contains three of Cisco's routers, series C7606. Routers are connected in a ring topology. Every router has a total of six modules. Two of six modules are processor cards, two modules are uplink cards and two modules are line cards for connecting end-devices. Modules of each router with its characteristics and power usage are presented in Table 1. Routers are connected through uplink cards. All ports on the uplink module have a maximum bandwidth of 10 Gb/s. To achieve redundancy on a module level, each router uses a separate uplink module to connect to two other routers.

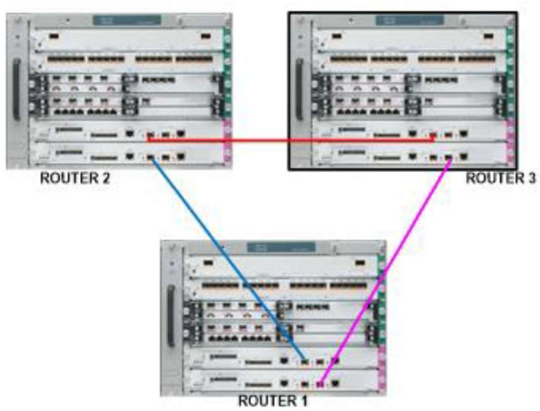

Fig. 1. IP/MPLS topology

Traffic level between routers vary through the day. On monitoring system, we recorded traffic level on Router1's both links in 12 hours period. The highest level of traffic is between 21:00 and 23:00, and the lowest level of traffic is between 03:00 and 07:00. Graph of bandwidth usage on Router 1 is shown on the Fig. 2.

In period from 03:00 to 07:00, both links had the bandwidth usage of only 1 Gb/s. During that time, both uplink modules consumed full energy consumption. The solution of this energy-wasting problem could be better traffic management and putting unused ports and modules into sleep mode. If total traffic of 2 Gb/s went through only one link, the other link and whole other module could be put into sleep mode. Table 1 has shown that one uplink module uses 300W per hour. So total energy saving with this solution would be 1,2 kW per night per device. The average medium-size service provider has about 300 devices in the network with over 1000 links. With this energy-saving practice, the savings could be significant.

TABLE 1: ENERGY USAGE

Device	Moduls	Numer of ports	Energy Consumtion
7606 I	76-ES+T-40G	40	418 W
	76-ES+T-2TG	2	300 W
	76-ES+T-2TG	2	300 W
	7600-ES+20G3C	20	276 W
	RSP720-3C-GE	2	310 W
	RSP720-3C-GE	2	310 W
7606 II	76-ES+T-40G	40	418 W
	76-ES+T-2TG	2	300 W
	76-ES+T-2TG	2	300 W
	7600-ES+20G3C	20	276 W
	RSP720-3C-GE	2	310 W
	RSP720-3C-GE	2	310 W
7606 III	76-ES+XC-40G3C	40	400 W
	76-ES+T-2TG	2	300 W
	76-ES+T-2TG	2	300 W
	7600-ES+20G3C	20	276 W
	RSP720-3C-GE	2	310 W
	RSP720-3C-GE	2	310 W

To apply the most efficient energy-saving solutions in networks, view of the whole network should be global. That way, devices in the network would calculate the states on all links, find the best paths and put in the sleep mode other links. Also, since traffic level varies through the day, network devices should be programmed to constantly measure states in the links and take appropriate actions. Ideally, the whole process should be automated.

When managing traffic, today's IP/MPLS is using the traffic-engineering feature. Traffic engineering (TE) [8] is a process of optimizing network performances by monitoring, analyzing, calculating and defining data behaviour over the network. MPLS TE uses components and protocols such as routing protocols (OSPF i.e. Open Shortest Path First or IS-IS i.e. Intermediate System to Intermediate System), reservation protocols (RSVP i.e. resource Reservation Protocol [9]), and path calculation algorithms (Constrained Shortest Path First - CSPF). But MPLS TE is limited when it comes to automation, programmability or global view of the network. In the next section, we will consider potential energy savings that SDN can bring.

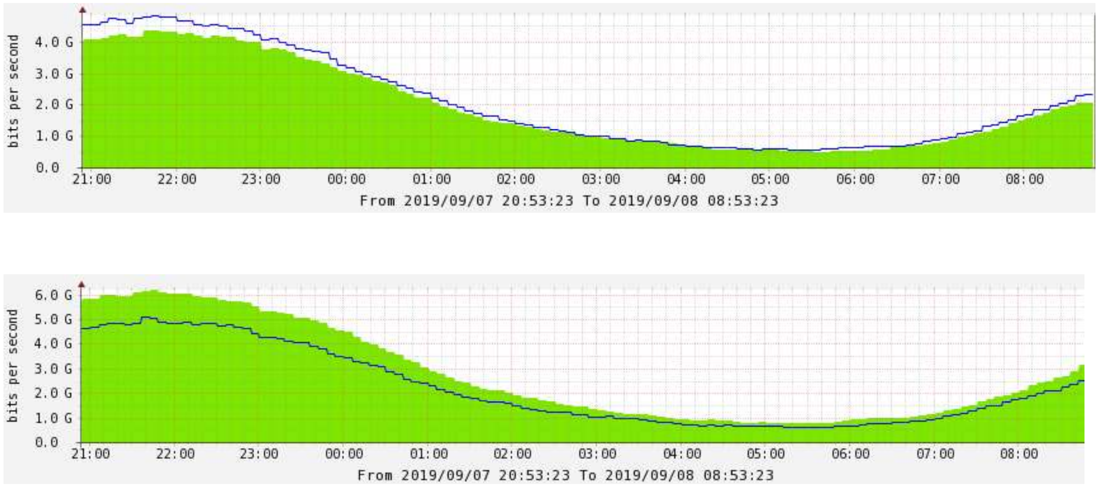

Fig. 2. Bandwidth usage on both links of Router 1

V. Proposed Energy Efficiency Improvement

We believe that SDN could enable energy saving solution by switching off underutilized links. SDN controller has a global view of whole network and insight into the link utilization. Controller can determine primary path in the network and turn off other underutilized links. If utilization on primary path ever become too high and congestion occurs, SDN controller can again turn on other paths. Turning on/off underutilized links should not negatively affect packet loss and delay.

To confirm this belief, we have simulated the same topology in Mininet simulator. The main difference is that we have replaced routers with OpenFlow switches. The simulated network contains three OpenFlow switches (S1, S2 and S3) connected in a ring topology, two hosts (H1 and H2) connected to S1 and S2, and RYU controller (C0). Topology is presented in Fig. 3. The controller C0 measures bandwidth on the links. If the measured link usage is less than the defined value, the controller will shut down underutilized link. If bandwidth ever increases and became greater than defined value, controller will turn on that link.

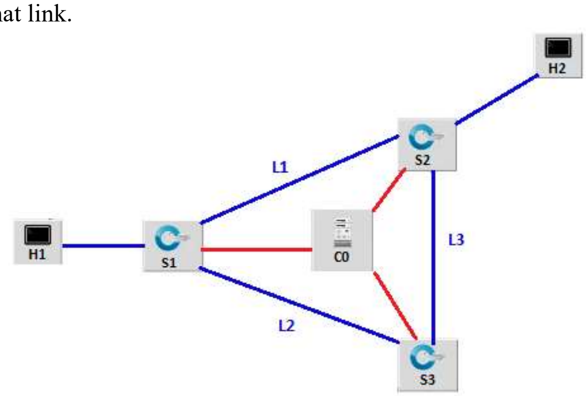

Fig. 3. Mininet SDN topology

Bandwidth on links L1 and L2 is configured to be 100 Mb/s. To see which link is used for data transfer, we implemented the first script (Script1) on the controller that monitors and counts how many packages go through each port on the switches. After that, we started a ping from host H1 to H2 and the script. Results are presented in Fig. 4. After we analyzed the number of tx packets (outgoing direction) on both ports on S1 and number of rx packets (inbound direction) on S2 and S3, we concluded that by default S1 uses both links to send packets from H1 to H2.

```
EVENT ofp_event->SimpleMonitor13 EventOFPPortStatsReply
datapath         in-port  eth-dst           out-port packets  bytes
---------------- -------- ----------------- -------- -------- --------
datapath         port     rx-pkts  rx-bytes rx-error tx-pkts  tx-bytes tx-error
---------------- -------- -------- -------- -------- -------- -------- --------
0000000000000001        1       79     9060        0       79     9060        0
0000000000000001        2       33     4489        0       37     4707        0
datapath         port     rx-pkts  rx-bytes rx-error tx-pkts  tx-bytes tx-error
---------------- -------- ----------------- -------- -------- --------
0000000000000002        1       79     9060        0       79     9060        0
0000000000000002        2       26     3741        0       40     5050        0
datapath         port     rx-pkts  rx-bytes rx-error tx-pkts  tx-bytes tx-error
---------------- -------- -------- -------- -------- -------- -------- --------
0000000000000003        1       37     4707        0       33     4489        0
0000000000000003        2       40     5050        0       26     3741        0
```

Fig. 4. Link usage statistics

Next step was implementing another script (Script2) on the controller that keeps track of used bandwidth on the links. If bandwidth usage on the link is lower than 30% and the summary value of bandwidth usage on both links (L1+L2) is lower than 90%, the controller will shut down the link. That means, even if bandwidth usage on link L2 is lower than 30%, but summary bandwidth usage of L1+L2 is higher than 90% L2 link will still be in use. This is the protection for over utilizing link L1.

Using iperf tool, we send traffic from host H1 to H2 with the bandwidth of 80Mb/s and run Script2. Results are presented in Fig. 5.

```
EVENT ofp_event->SimpleMonitor13 EventOFPPortStatsReply
datapath         in-port  eth-dst           out-port packets  bytes
---------------- -------- ----------------- -------- -------- --------
datapath         port     rx-pkts  rx-bytes rx-error tx-pkts  tx-bytes tx-error
---------------- -------- -------- -------- -------- -------- -------- --------
0000000000000001        1       43     5413        0       43     5413        0
0000000000000001        2       33     4489        0       37     4707        0
datapath         port     rx-pkts  rx-bytes rx-error tx-pkts  tx-bytes tx-error
---------------- -------- ----------------- -------- -------- --------
0000000000000002        1       43     5413        0       43     5413        0
0000000000000002        2       26     3741        0       40     5050        0
datapath         port     rx-pkts  rx-bytes rx-error tx-pkts  tx-bytes tx-error
---------------- -------- -------- -------- -------- -------- -------- --------
0000000000000003        1       37     4707        0       33     4489        0
0000000000000003        2       40     5050        0       26     3741        0
```

Fig. 5. Link usage statistics after Script2

After that, we analyzed again the output of Script1 and

978-1-7281-4790-1/19/$31.00 ©2019 IEEE

situation was the same as before – both links were in use. Then, we decreased the generated traffic in iperf tool. Now, H1 was sending traffic to H2 with the intensity of 20Mb/s. After we run Script2, the link between S1 and S3 was down, and only link L1 was used to transfer 20Mb/s. By analyzing the output of Script1, we noticed that rx packet values on S2 were constantly growing, and rx packet values on S3 remained unchanged. Further on, even when link L2 went down, any packet loss in this communication was not registered. The only change we noticed was a bit higher latency in H1-H2 ping, as shown in Fig. 6, the most common latency during ping was 0,08 ms, but when the change in the network occurred latency was at some point 1,85 ms and then again 0,08 ms.

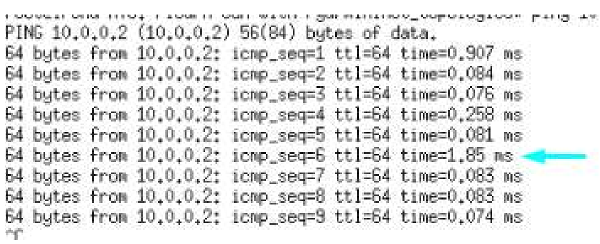

Fig. 6. RTT (Round-Trip Time) change

VI. Conclusion

SDN is an emerging architecture that is adaptable, manageable, cost-effective, and dynamic, making it ideal for nowadays applications.

Combining SDN with the existing methods of energy efficiency used in traditional networks could be a win-win solution. SDN controllers could be made energy aware. This awareness could enable them to use energy more efficiently by shutting down on needed interfaces and cards in SDN switches.

References

[1] Gelenbe, E. and Silvestri, S. Optimisation of Power Consumption in Wired Packet Networks. In: Bartolini, N., Nikoletseas, S., Sinha, P., Cardellini, V. and Mahanti, A., Eds., Telecommunications Engineering, Springer Berlin Heidelberg, 2009, 717-729.

[2] R. Carpa, O. Gluck and L. Lefevre, "Segment routing based traffic engineering for energy efficient backbone networks," in 2014 IEEE International Conference on Advanced Networks and Telecommuncations Systems (ANTS), 2014, pp. 1-6.

[3] Radu Carpa. Energy Efficient Traffic Engineering in Software Defined Networks. Networking and Internet Architecture, Université de Lyon, 2017.

[4] F. A. Moghaddam and P. Grosso, "Linear programming approaches for power savings in software-defined networks," in 2016 IEEE NetSoft Conference and Workshops (NetSoft), 2016, pp. 83-87.

[5] Kra, L., Gondo, Y., Goore, B.T. and Asseu, O. Contribution to the Optimization of the Energy Consumption in SDN Networks. Journal of Sensor Technology, 2018, 8, 59-67, https://doi.org/10.4236/jst.2018.83005

[6] Tuysuz, M.F.; Ankarali, Z.K.; Gözüpek, D. A Survey on Energy Efficiency in Software Defined Networks. Comput. Netw. 2017, 113, 188–204.

[7] Nicolas Huin. Energy efficient software defined networks. Networking and Internet Architecture [cs.NI]. Université Côte d'Azur, 2017.

[8] "IETF standard: Requirements for Traffic Engineering over MPLS," IETF. September 1999, [Online]. Available: https://tools.ietf.org/html/rfc2702.

[9] "IETF standard: RSVP-TE Extensions to RSVP fot LSP Tunnels." [Online]. December 2001, Available: https://tools.ietf.org/html/rfc2702.