Dynamic Management of Forwarding Rules in a T-SDN Architecture with Energy and Bandwidth Constraints

A. Petrosino[1], G. Sciddurlo[1], G. Grieco[1,2], A. Shah[1,2], G. Piro[1,2], L. A. Grieco[1,2], and G. Boggia[1,2]

[1] Dept. of Electrical and Information Engineering (DEI), Politecnico di Bari, Bari, Italy

[2] CNIT, Consorzio Nazionale Interunivesitario per le Telecomunicazioni, Italy

{antonio.petrosino, giancarlo.sciddurlo, giovanni.grieco, awais.shah, giuseppe.piro, alfredo.grieco, gennaro.boggia}@poliba.it

Abstract. Telecom operators recently started to integrate Software-Defined Networking facilities for controlling and managing their optical transport networks. Here, the management of forwarding rules into the resulting Transport Software-Defined Networking (T-SDN) architecture has to be addressed by taking into account the energy and quality of service requirements. While the most of works in the literature studied these aspects separately, the few contributions that simultaneously take care of energy and quality of service requirements present latency, scalability, or control communication issues. Starting from these considerations, this paper formulates a novel methodology for the dynamic and reactive management of forwarding rules in a (potentially large-scale) T-SDN network, based on the knowledge of network topology, the power consumption of optical switches, the expected volume of traffic, and the variability of the actual traffic load. First, the expected volume of traffic and the estimated power consumption of optical switches are exploited to select the minimum number of nodes and transport links to activate, which enable the communication among any source and destination pairs declared within a given traffic matrix. Then, the bandwidth consumption of activated transport links is periodically monitored by a centralized controller and, in case of congestion, a new set of optical switches and transport links are quickly turned on for addressing the growth of the traffic load. The effectiveness of the proposed approach has been investigated through experimental tests and compared against another reference scheme which considers the energy issue only. Obtained results demonstrate its ability to offer higher levels of quality of service to end-users, at the expense of a limited decrease of the registered energy-saving.

Keywords: Software-Defined Networking · Transport Network · Energy Efficiency · Quality of Service · Forwarding Policies.

1 Introduction

Transport networks are rapidly evolving towards flexible and controllable architectures able to dynamically manage the large heterogeneity of data flows [2]. For this reason, telecom operators are revolutionizing their network infrastructures by massively integrating Software-Defined Networking (SDN) facilities (i.e., separation of data and control planes, monitoring and configuration of networking functionalities, and so on) [12]. In this context, the management of forwarding rules in the resulting Transport-SDN (T-SDN) deployment is a very ambitious task to accomplish. A challenging goal, in fact, is to reduce the power consumption of the operating network, while satisfying the requested levels of quality of service (e.g., bandwidth consumption) [19, 18]. At the same time, the high variability of the traffic loads asking for quick, scalable, and easily deployable strategies, makes things worse [3].

Several solutions in the current scientific literature address energy and bandwidth constraints almost separately. From one hand, energy-efficient schemes try to turn off as more optical switches and transport links as possible. Starting from the knowledge of network topology and the expected set of data flows (declared through the so-called traffic matrix), available solutions configure forwarding rules by solving optimization problems [1, 11, 15, 22, 24] or by executing heuristic algorithms [4–7, 13, 16]. With these mechanisms, most of the network traffic is forwarded through a reduced set of links. Therefore, flow dynamics generally bring to network congestion issues. From another hand, the rest of contributions (see [20] and [21] for example) only targets quality of service requirements, while missing the energy constraints.

At the time of this writing, the energy consumption and bandwidth constraints are jointly considered in [10], [23], and [8]. Specifically, [10] presents a multi-objective algorithm that derives the set of links to disable, while fulfilling the expected quality of service constraints. Here, forwarding rules are configured by one of the nodes of the network (acting as a controller) through in-band communications. This, however, increases the latencies of the exchange of control messages, as well as makes the resulting implementation infeasible in large-scale scenarios. In fact, the in-band communication approach is optimal in non-dynamic situations where it is not necessary to update the forwarding rules every few seconds, but not for a dynamic environment because the benefits arising from the presence of a controller interacting with optical switches by means of out-band communications are ignored during the in-band communication mode. The heuristic approach introduced in [23] configures forwarding rules by creating spanning trees of nodes with assigned weights according to their energy consumption. Unfortunately, it does not envisage to monitor the actual traffic load, thus being unable to react to data flow dynamics and congestion episodes. Finally, the work presented in [8] assumes to dynamically configure forwarding rules by taking into account the expected traffic volume and by targeting the shutdown of as many transport links as possible. This solution surely limits the energy consumption, but still lacks in reacting to the variability of the actual traffic load.

In order to solve the issues characterizing the current state of the art, this paper proposes a novel methodology for the dynamic management of forwarding rules in T-SDN deployments. This is done by jointly considering the network topology, the power consumption of optical switches, the expected volume of traffic, and the variability of the actual traffic load. In particular, the proposed strategy starts by activating the minimum required nodes and transport links between the source and destination pair predefined within a given traffic matrix, based on the network topology and the estimated power consumption of the optical switches. Then, the bandwidth utilization of the activated transport links is periodically monitored by a centralized controller to recognize the actual traffic load. In case of congestion, new transport links and optical switches are activated to ensure the smooth running of the traffic inside the network. Experimental tests demonstrate the better trade-off between the power consumption and quality of service. The performance of the proposed approach has been experimentally investigated by emulating a T-SDN architecture within a desktop computer. The GÉANT[3] network topology, embracing 40 nodes, 58 bidirectional links, and an OpenDaylight controller, has been implemented within the Mininet environment. The actual traffic load is generated by activating a percentage of requests declared in a traffic matrix describing the data flows between up to 24 host pairs attached to the GÉANT topology. The collected results have been compared with the approach described in [16], which only reduce the energy consumption of the network. The produced results have been compared with the approach presented in [16], since it is a state of the art algorithm that achieves excellent energy savings. Indeed, the approach presented in this paper exhibits the lowest throughput degradation with respect to [16], thus demonstrating its successfully ability to redirect data flows across uncongested paths. Therefore, it is clear that the strategy presented in this paper provides a significant gain in terms of performance, at the expense a limited decrease of the registered energy-saving as compared to [16].

The rest of the paper is organized as follows: Section 2 presents the reference architecture. Section 3 describes the proposed algorithm. The description of the experimental testbed is presented in Section 4 along with the conclusion of the achieved results. Finally, Section 5 draws the conclusions and proposes future research activities.

2 The reference architecture and main assumptions

Figure 1 shows the reference T-SDN network considered in this work. According to the well-known SDN reference model, physical nodes and logical entities are grouped into three layers: infrastructure, control, and application [25]. The infrastructure layer embraces optical switches of the core network and edge routers. Optical switches forward data flows within the core network, according to the configured routing rules. Edge routers, instead, act as sources and destinations of data flows. Furthermore, a centralized controller monitors the infrastructure

[3] https://www.geant.org/Networks (Accessed: 2020-03-15)

layer and dynamically configures forwarding rules based on the outcomes of the routing algorithm working at the application layer.

Both Software-Defined Controller and optical switches implement the OpenFlow stack (southbound interface). The controller, implemented with OpenDaylight framework, periodically queries optical switches for collecting details about the network topology and the amount of bandwidth consumed by each physical port. When needed, it also delivers the new set of forwarding rules across the network. According to OpenFlow specifications, the communication in the southbound interface is managed by means of the REpresentational State Transfer (REST)CONF protocol [2]. The application entity implementing the routing algorithm and the controller interact with each other with RESTful Application Programming Interface (API)s [2]. In this case, the exchanged messages are encoded with the Yet Another Next Generation (YANG) data model (northbound interface) [17].

The design of the novel routing algorithm discussed herein grounds its roots on the following consideration. From one hand, the network operator knows the expected volume of traffic that can be generated between all possible pairs of source and destination edge routers. Such information is stored within the traffic matrix [25] and may vary during the time (e.g., the volume of traffic manageable by the T-SDN network in daily hours may be different from the one available during the night or weekend). On the other hand, the actual traffic load generated within the network may differ from the traffic matrix, spanning from a very limited percentage of the expected volume of traffic to its upper bound. This double level of dynamicity makes challenging the task performed by the routing algorithm. Indeed, the conceived methodology intends to configure the infrastructure layer by jointly considering information stored within the traffic matrix and the traffic load managed by optical switches during the time, periodically monitored by the controller.

A power model helps to estimate the amount of power consumed by optical switches belonging to the reference T-SDN network. Without loss of generality, this paper considers the model presented in [14], related to NEC PF 5240[4] OpenFlow switches. Here, the total amount of power consumed by an optical switch is given by five contributions:

- the amount of power required to keep the switch on, $P_{\text{base}} = 118.30\,\text{W}$;
- the amount of power needed to configure device settings and active ports, $P_{\text{conf}} = 0.52\,\text{W}$;
- the amount of power needed to install a new OpenFlow rule $P_{\text{flow-mod}} = 20.00\,\mu\text{W}$;
- the amount of power consumed for each control packet $P_{\text{packet}} = 711.00\,\mu\text{W}$;
- the amount of power consumption due to the processing of data flow $P_{\text{flow}} \ll 1\,\mu\text{W}$.

The analysis presented in [14] already demonstrated that the processing of data flows has a very minimal effect on the overall power consumption. Accord-

[4] https://www.necam.com/sdn/Hardware/PF5240Switch/ (Accessed: 2020-04-10)

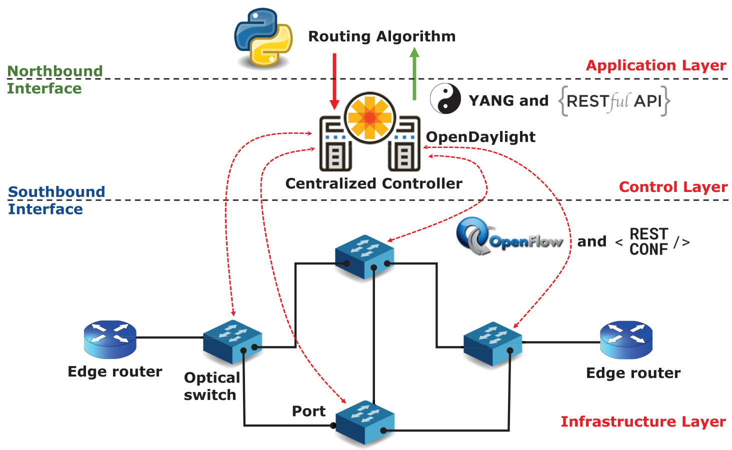


Fig. 1: Reference T-SDN network architecture.

ingly, it is possible to neglect the impact of P_{flow} and develop a strategy based on a traffic independent power model.

3 The conceived approach

The routing algorithm conceived in this paper periodically implements two different tasks. The first one provides an initial configuration of the T-SDN core network, based on the knowledge of the network topology and the expected volume of traffic declared by the traffic matrix. Therefore, it is executed only once, at the beginning of the validity period of the traffic matrix. The second task, instead, is implemented every congestion observation window and provides periodic updates of forwarding rules, based on the actual traffic load passing through the network. In order to effectively react to possible congestion episodes, the duration of the congestion observation window is much smaller than the validity period of the traffic matrix (i.e., tens of seconds instead of hours).

Initial network configuration based on the traffic matrix (Task 1). It intends to reduce the overall power consumption by turning off as many devices and links as possible, while ensuring communication paths for any data flow reported in the traffic matrix. To this end, the network is modeled as an undirected graph G, where nodes represent optical switches and edges represent the transport links connecting optical switches. The set of demands D representing the traffic matrix is described by the pair of source node s and destination node

t with their respective bandwidth demand d^{st}. Nodes belonging to the graph G are sorted according to their power consumption, from the most consuming device to the less consuming one. Links, instead, are randomly ordered. Then, an iteration on nodes is performed. At each iteration, the considered node in the ordered set and all of its links are tentatively turned off. Indeed, it is verified if at least one path exists for each traffic request declared in the traffic matrix. In the affirmative case, that node is removed from G since it is not necessary for the fulfillment of all traffic requests. Otherwise, the considered node and its links are left active into the network. Once the iteration on the nodes is completed, the same procedure is applied to the links. Also, in this case, the goal is to turn off unuseful or redundant links and leave active only a subset of links that guarantees the presence of communication paths for all data flows declared into the traffic matrix. A minimized graph G' is obtained at the end, which represents the network topology guaranteeing the greatest energy savings.

Given the minimized graph G', the shortest communication path for each data flow of the traffic matrix is identified according to the Dijkstra algorithm [9]. The calculated shortest paths are converted to forwarding rules and pushed on OpenFlow switches by the controller.

Redefinition of forwarding rules based on congestion episodes (Task 2). In a dynamic environment where the actual traffic load changes, this task further adapts forwarding rules based on user demands and link capacity. To this end, the controller periodically sends OpenFlow messages to the switches, requesting information about the bandwidth consumption of their enabled ports. This helps to identify the activation of new flows that may congest transport links and provoke service degradation. This monitoring procedure allows to detect link congestion when the total bandwidth of the considered link is at least 90% occupied. Once detected the overloaded links, the data flows triggering that event are put within the congestion list. The recursive algorithm discussed before is implemented again over the network topology that excludes congested links. As a consequence, the algorithm will turn on transport links or optical switches that were turned off at the beginning. Then, a new shortest path is defined for each data flow in the congestion list, converted to forwarding rules, and pushed on OpenFlow switches. At the end of the congestion observation window, the network is configured as indicated by the first task. Therefore, congestion episodes are periodically managed, starting from a baseline network configuration.

To provide further insight, the pseudo-code describing the main functionalities of the conceived approach has been reported in Algorithm 1.

Algorithm 1 Pseudo code of the proposed methodology

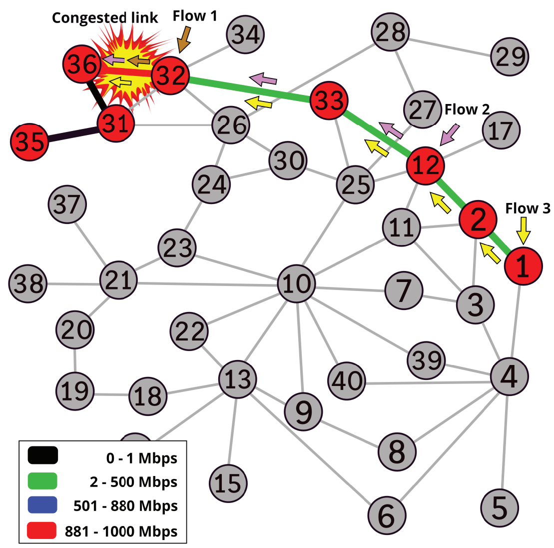
```
Input: Graph G(nodes, links), set D of demand with traffic requirement d^st ∀(s,t) ∈ D
Output: Updated flow tables, Final graph
TASK 1 (G, D):
G′ ← G
Nodes Optimization on G′                  ▷ Nodes are sorted in a most power order.
for i ← 1 to |nodes| do
    turn_off(nodes[i])
    for all (s,t) ∈ D do
        if !path_exists(s,t) then
            turn_on(nodes[i])
        end if
    end for
end for
Links Optimization on G′                  ▷ Links are selected in random order.
for i ← 1 to |links| do
    turn_off(links[i])
    for all (s,t) ∈ D do
        if !path_exists(s,t) then
            turn_on(links[i])
        end if
    end for
end for
Push Forwarding Rules
for all (s,t) ∈ D do
    path(s,t) ← Dijkstra algorithm
    push_flow_rules()
end for
# Controller monitors links bandwidth consumption.#
TASK 2 (G, D):
if congestion_occurs then
    Nodes Optimization
    Links Optimization
    for all (s,t) ∈ D do
        path(s,t) ← Dijkstra algorithm
        for i ← 1 to |link_in_path| do
            if remaining_link_capacity < d^st and link_overloaded then
                Congestion_list ← (s,t)
            end if
        end for
        if !Congestion_list.contains(s,t) or
any path without overloaded links exists to satisfy (s,t) then
            update(remaining_link_capacity)
            push_flow_rules()
        end if
        G′′ ← remove_overloaded_links(G)
        if !Congestion_list.empty() then
            Run TASK 2(G′′, Congestion_list)
        end if
    end for
end if
```

4 Performance Evaluation

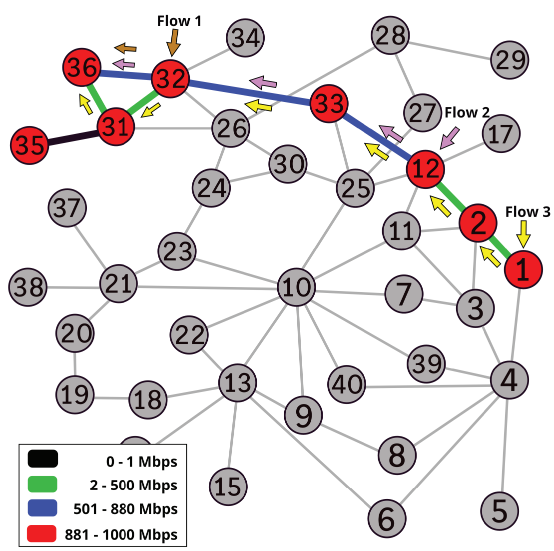
The performance of the proposed approach is experimentally investigated by emulating a T-SDN architecture within a desktop computer Intel Core i7-7700, RAM 16GB, with Ubuntu 18.04 64-bit. Specifically, the GÉANT topology with 40 nodes and 58 bidirectional links is implemented with Mininet, since it allows to virtualize a network of OpenFlow switches with Open vSwitch kernel. The OpenDaylight framework is used as the network controller. The routing algorithm has been developed in Python. Without loss of generality, the conducted analysis considers transport links supporting 1 Gbps of bandwidth. A traffic matrix is arbitrarily created in order to describe the data flows expected between up to 24 host pairs randomly attached to the GÉANT topology. Each data flow in the traffic matrix presents a request rate of 400 Mbps. The actual traffic load is generated by activating a percentage of requests declared in the traffic matrix. To provide further insights, the performance of the proposed approach has been compared with respect to the algorithm presented in [16], which only tries to minimize energy consumption.

Figure 2 depicts a simplified example showing the ability of the conceived solution to successfully react to congestion episodes. The example considers three data flows, asking for 400 Mbps of bandwidth each, directed to the same destination. Figure 2a represents the network topology configured according to the algorithm presented in [16]. It is possible to observe that the link connecting node 32 to node 36, which only offers 1 Gbps of bandwidth, is congested. This means that the strategy presented in [16] is not able to fulfill the quality of service levels requested by the considered data flow. Note that the initial network configuration provided by Task 1 of the algorithm presented in Section 3 coincides with the one obtained through [16]. Differently, from [16], however, Task 2 implemented by the approach described in this paper adapts forwarding rules in reaction to congestion episodes. Figure 2b clearly shows how the path followed by Flow 3 is updated. Accordingly, the link between node 32 and node 36 in the example is not overloaded and the quality of service requested by all the three flows is achieved. Quantitative key performance indicators discussed below include the total power consumption of the T-SDN network, the percentage of deactivated links, and the percentage of throughput degradation registered by active data flows.

The total amount of power consumed by an operating T-SDN network is evaluated by considering 6 to 24 active data flows, generating 100% of the data rate declared in the traffic matrix (that is equal to 400 Mbps). In case, the network has all the optical switches and transport links turned on, the total power consumption is equal to 4792.32 W. This is reported in Figure 3 as the peak value achievable in the absence of any energy-aware routing strategy. The other two curves reported in Figure 3 shows the amount of power consumed by the considered T-SDN network as a function of the number of active data flows, when forwarding rules are set according to the algorithm presented in [16] and the solution conceived in this paper. As expected, results show that the increment of the number of active data flows always requires higher number of



(a) Network configuration based on [16].



(b) Network configuration achieved with the proposed approach.

Fig. 2: Example showing the ability of the proposed approach to achieve energy and quality of service constraints.

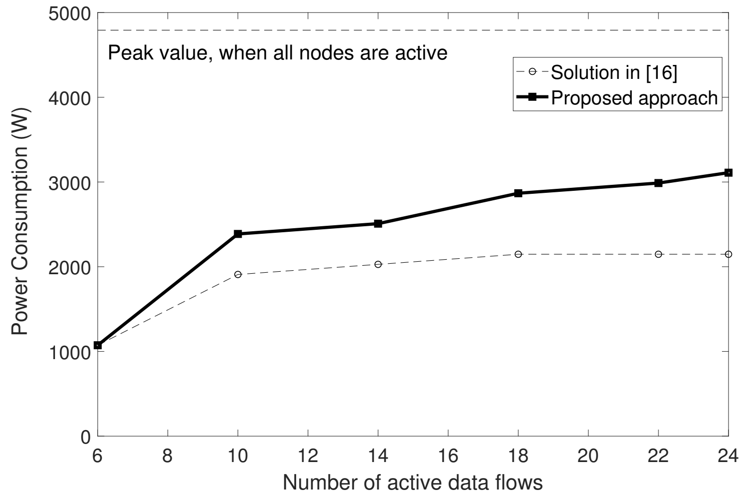


Fig. 3: Power consumption.

optical switches and links to activate in the network. This inevitably brings to an increment of the overall power consumption. It is also evident that the algorithm presented in [16] ensures the highest power saving, thanks to its ability to shut down as many optical switches and transport links as possible, without taking care of the quality of service level offered to end-users. On the contrary, the methodology presented in the paper registers a slight increment of the power consumption due to the activation of more optical switches and links, triggered in answer to congestion events.

Figure 4, showing the percentage of deactivated transport links, fully confirms the aforementioned discussion: the proposed solution forwards data flows through a higher number of uncongested paths. It is also possible to observe that the difference between the two investigated approaches becomes more evident when the number of active data flows increases. In this case, in fact, the higher the bandwidth requirement, the higher the number of paths to activate for avoiding network congestion.

The real effectiveness of the conceived solution is highlighted in Figure 5, which reports the degradation of the throughput registered by active data flows due to bandwidth constraints, measured as a function of the traffic load (expressed as a percentage of the bandwidth requirement declared in the traffic matrix). Since [16] does not apply any re-routing strategy after the congestion, a large traffic load seriously degrades network performance. The approach presented in this paper exhibits the lowest throughput with respect to [16], thus demonstrating its successful ability to redirect data flows across uncongested paths. From these considerations, it is evident that the strategy presented in

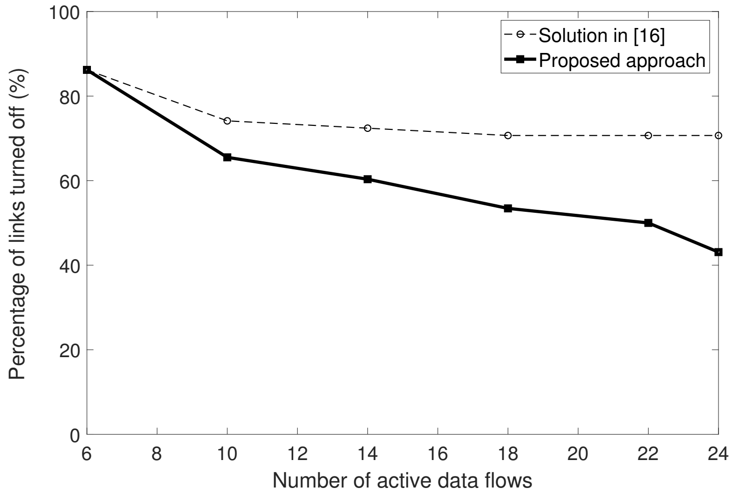


Fig. 4: Percentage of links turned off.

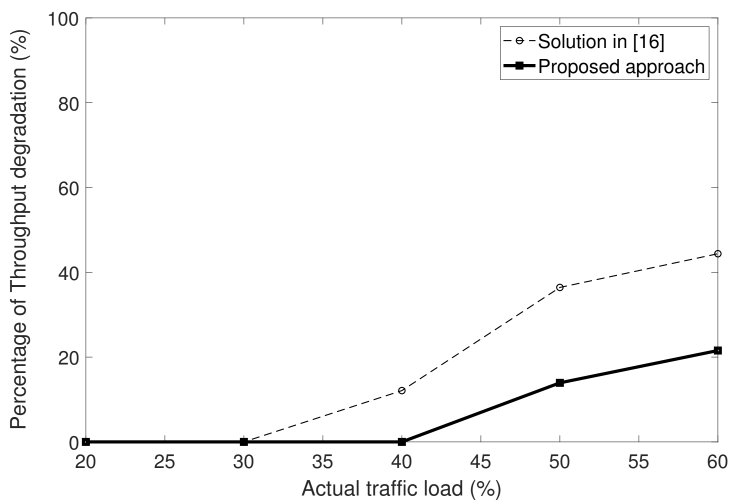


Fig. 5: Throughput degradation registered by active data flows

this paper provides a significant gain in terms of performance at the expense of a limited decrease of registered energy-saving as compared to [16].

5 Conclusions

This paper focuses on T-SDN networks and formulated a novel methodology for the dynamic management of forwarding rules in the presence of energy and bandwidth constraints. The conceived approach configures communications paths and decides optical switches and transport links to be activated by jointly considering the network topology, the power consumption of optical switches, the expected volume of traffic, and variability of the actual traffic load. Experimental tests demonstrate its ability to achieve the best compromise between the power consumption of the overall network and the quality of service offered to end-users. Future research activities will analyze the complexity and investigate the behavior of the considered solution in complex and large scale network while considering realistic traffic matrix and flow generation statistics. Moreover, it will also investigate its adoption in hierarchical T-SDN deployments, based on two layers of controllers introduced to improve scalability and provide a comprehensive comparison with the other available state of the art approaches.

Acknowledgment

This work was mainly supported by the Apulia Region (Italy) Research project INTENTO (36A49H6). It was also partially supported by the PRIN project no. 2017NS9FEY entitled “Realtime Control of 5G Wireless Networks: Taming the Complexity of Future Transmission and Computation Challenges” funded by the Italian MIUR and by the Italian MIUR PON projects Pico&Pro (ARS01 01061), AGREED (ARS01 00254), FURTHER (ARS01 01283) and RAFAEL (ARS01 00305).

References

1. Al Mhdawi, A.K., Al-Raweshidy, H.S.: iPRDR: Intelligent Power Reduction Decision Routing Protocol for Big Traffic Flood in Hybrid-SDN Architecture. IEEE Access **6**, 10944–10955 (2018)
2. Alvizu, R., Maier, G., Kukreja, N., Pattavina, A., Morro, R., Capello, A., Cavazzoni, C.: Comprehensive Survey on T-SDN: Software-Defined Networking for Transport Networks. IEEE Communications Surveys Tutorials **19**(4), 2232–2283 (2017)
3. Arif, M., Wang, G., Geman, O., Balas, V.E., Tao, P., Brezulianu, A., Chen, J.: Sdn-based vanets, security attacks, applications, and challenges. Applied Sciences **10**(9), 3217 (2020)
4. Assefa, B.G., Ozkasap, O.: Link utility and traffic aware energy saving in software defined networks. In: Proc. of IEEE International Black Sea Conference on Communications and Networking. pp. 1–5 (2017)
5. Assefa, B.G., Ozkasap, O.: A Novel Utility Based Metric and Routing for Energy Efficiency in Software Defined Networking. In: Proc. of International Symposium on Networks, Computers and Communications. pp. 1–4 (2019)

6. Assefa, B.G., Ozkasap, O.: RESDN: A Novel Metric and Method for Energy Efficient Routing in Software Defined Networks. IEEE Transactions on Network and Service Management pp. 1–1 (2020)
7. Awad, M.K., Rafique, Y., Alhadlaq, S., Hassoun, D., Alabdulhadi, A., Thani, S.: A Greedy Power-Aware Routing Algorithm for Software-Defined Networks. In: Proc. of IEEE International Symposium on Signal Processing and Information Technology (ISSPIT). pp. 268–273 (2016)
8. Ba, J., Wang, Y., Zhong, X., Feng, S., Qiu, X., Guo, S.: An SDN energy saving method based on topology switch and rerouting. In: NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium. pp. 1–5 (2018)
9. Deng, Y., Chen, Y., Zhang, Y., Mahadevan, S.: Fuzzy dijkstra algorithm for shortest path problem under uncertain environment. Applied Soft Computing **12**(3), 1231–1237 (2012)
10. Fernández-Fernández, A., Cervelló-Pastor, C., Ochoa-Aday, L.: A Multi-Objective Routing Strategy for QoS and Energy Awareness in Software-Defined Networks. IEEE Communications Letters **21**(11), 2416–2419 (2017)
11. Giroire, F., Moulierac, J., Phan, T.K.: Optimizing Rule Placement in Software-Defined Networks for Energy-Aware Routing. In: Proc. of IEEE Global Communications Conference. pp. 2523–2529 (2014)
12. Goransson, P., Black, C., Culver, T.: Software Defined Networks: A Comprehensive Approach. Morgan Kaufmann (2016)
13. Heller, B. and Seetharaman, Srini and Mahadevan, Priya and Yiakoumis, Y. and Sharma, P. and Banerjee, S. and McKeown, Nick: Elastictree: Saving Energy in Data Center Networks. 7th USENIX NSDI pp. 17–17 (2010)
14. Kaup, F., Melnikowitsch, S., Hausheer, D.: Measuring and Modeling the Power Consumption of OpenFlow Switches. In: Proc. of 10th International Conference on Network and Service Management (CNSM) and Workshop. pp. 181–186 (2014)
15. Li, H., Jiang, G., Chai, R.: Energy Consumption Optimization Based Joint Routing and Flow Allocation Algorithm for Software Defined Networking. In: Proc. of 19th International Symposium on Wireless Personal Multimedia Communications (WPMC). pp. 311–316 (2016)
16. Maaloul, R., Taktak, R., Chaari, L., Cousin, B.: Energy-Aware Routing in Carrier-Grade Ethernet Using SDN Approach. IEEE Transactions on Green Communications and Networking **2**(3), 844–858 (2018)
17. Parladori, G., Gasparini, G., Ruggi, F., Broi, A.D., Simone, V., Nicassio, F.: YANG Modelling of Optical Nodes. In: Proc. of 20th Italian National Conference on Photonic Technologies (Fotonica 2018). pp. 1–4 (2018)
18. Polese, M., Chiariotti, F., Bonetto, E., Rigotto, F., Zanella, A., Zorzi, M.: A survey on recent advances in transport layer protocols. IEEE Communications Surveys & Tutorials **21**(4), 3584–3608 (2019)
19. Rehmani, M.H., Davy, A., Jennings, B., Assi, C.: Software defined networks-based smart grid communication: A comprehensive survey. IEEE Communications Surveys & Tutorials **21**(3), 2637–2670 (2019)
20. Sathyanarayana, S., Moh, M.: Joint Route-Server Load Balancing in Software Defined Networks Using Ant Colony Optimization. In: Proc. of International Conference on High Performance Computing Simulation (HPCS). pp. 156–163 (2016)
21. Stefano, A.D., Cammarata, G., Morana, G., Zito, D.: A4SDN - Adaptive Alienated Ant Algorithm for Software-Defined Networking. In: Proc. of 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC). pp. 344–350 (2015)

22. Vasić, N., Bhurat, P., Novaković, D., Canini, M., Shekhar, S., Kostić, D.: Identifying and Using Energy-Critical Paths. In: Proc. of the Seventh COnference on emerging Networking EXperiments and Technologies. pp. 1–12 (2011)
23. Wang, H., Li, Y., Jin, D., Hui, P., Wu, J.: Saving Energy in Partially Deployed Software Defined Networks. IEEE Transactions on Computers **65**(5), 1578–1592 (2016)
24. Wu, Z., Ji, X., Wang, Y., Chen, X., Cai, Y.: An Energy-Aware Routing for Optimizing Control and Data Traffic in SDN. In: Proc. of 26th International Conference on Systems Engineering (ICSEng). pp. 1–4 (2018)
25. Xia, W., Wen, Y., Foh, C.H., Niyato, D., Xie, H.: A Survey on Software-Defined Networking. IEEE Communications Surveys Tutorials **17**(1), 27–51 (2015)