Bringing Energy Aware Routing closer to Reality with SDN Hybrid Networks

Nicolas Huin, Myriana Rifai, Frédéric Giroire, Dino Lopez Pacheco, Guillaume Urvoy-Keller, Joanna Moulierac

Bringing Energy Aware Routing closer to Reality with SDN Hybrid Networks

N. Huin, M. Rifai, F. Giroire, D. Lopez Pacheco, G. Urvoy-Keller, J. Moulierac
Université Côte d'Azur, Inria, CNRS, I3S, Sophia Antipolis, France

***Abstract*—Energy aware routing aims at reducing the energy consumption of ISP networks. The idea is to adapt routing to the traffic load in order to turn off some hardware. However, it implies to make dynamic changes to routing configurations which is almost impossible with legacy protocols. The Software Defined Network (SDN) paradigm bears the promise of allowing a dynamic optimization with its centralized controller.**

In this work, we propose SENAtoR, an algorithm to enable energy aware routing in a scenario of progressive migration from legacy to SDN hardware. Since in real life, turning off network equipments is a delicate task as it can lead to packet losses, SENAtoR provides also several features to safely enable energy saving services: tunneling for fast rerouting, smooth node disabling and detection of both traffic spikes and link failures.

We validate our solution by extensive simulations and by experimentation. We show that SENAtoR can be progressively deployed in a network using the SDN paradigm. It allows to reduce the energy consumption of ISP networks by 5 to 35% depending on the penetration of SDN hardware, while diminishing the packet loss rate compared to legacy protocols.

I. Introduction

At the core of a large number of energy efficient solutions e.g., energy aware routing, resides a dynamic adaptation of network resources to the network load. However, in legacy networks, operators are reluctant to change network configurations as they are frequently manually set. Energy efficient solutions are thus hard to be put in practice. On the other hand, by placing the control plane in a central controller, the Software Defined Network (SDN) paradigm allows the dynamic control of a network. SDN thus bears the promise of enabling those energy efficient solutions.

Different scenarios may be envisioned for the transition from legacy to SDN networks [1]. One of the most realistic is a progressive migration, where legacy hardware is replaced over a long period of time by SDN hardware. There is thus a coexistence of legacy and SDN, hardware and protocols, in the network. As an example, to route packets inside the network, legacy nodes have to follow legacy protocols, such as OSPF, while SDN nodes may choose the next hops of the packets using an optimization algorithm running in the controller.

In this paper, we consider the problem of *energy aware routing in a hybrid SDN network.* To provide energy optimization in hybrid networks, we introduce SENAtoR- **S**mooth **EN**ergy **A**ware **R**outing. The main idea is that the controller first chooses the set of routes that minimizes the number of used network equipments for the current traffic, and then we put SDN nodes in sleep mode (i.e., power save mode which turns off network interfaces). We consider a typical dynamic traffic of an operator and, hence, our solution adapts the numbers of active and inactive network equipments during the day.

When the SDN nodes are put in sleep and their links are turned off, traffic has to be rerouted, while avoiding packet loss. It is thus impossible to wait for the convergence of the legacy protocols (e.g., OSPF). Moreover, if ISP network traffic usually shows smooth variations of throughput, it also experiences sudden changes which may correspond to (link or node) failures or to flash crowds [2].Thus, to avoid packet loss we propose three mechanisms:

1. Tunneling. This first mechanism is inspired by the solution proposed in [3] to handle single link failure. The goal was to avoid waiting for the convergence of legacy routing protocols by using tunnels from a node with a failing link to an SDN node which can reach an alternative OSPF shortest path in one hop. We reused this idea to reroute using pre-set *tunnels* from any node, with a turned off link, to any other node with a direct path towards the destination which does not include a disabled link.

2. Turning off links smoothly. To prevent OSPF routers from sending packets towards a node which was just put into sleep mode by the energy saving mechanism, we propose to force OSPF re-convergence before the Network Interface Card (NIC) at the SDN is really turned off. The idea is that the SDN controller discards any OSPF packet sent on the node to be disabled to simulate a node failure while any other data packet must be properly processed and forwarded. After a period of time, greater than the link failure detection period and than the convergence of OSPF (which can be estimated with the OSPF timer values), and if no more traffic is received, the SDN router will effectively turn off the appropriate NICs. Note that while OSPF has not converged yet, packets can be rerouted through the pre-set tunnels; and since the link and node is still on, packets are not lost during the routing transition.

3. Traffic Spike and link failure mitigation. Network capacity over-provisioning is exploited by energy aware algorithms to save energy. Indeed, networks are oversized, in particular, to handle traffic variations due e.g., to link failures or flash crowds. It is thus of crucial importance for energy saving mechanisms, which turn off equipments, to not impact the failure tolerance of networks. We exploit the metrology data received by the controller from SDN nodes to detect important traffic variations and react to them.

Our contributions are the following:

- We propose several mechanisms to bring energy aware solutions closer to reality in ISP networks to avoid packet losses when putting network devices into sleep mode: tunneling, smooth shutdown of links, and detection of traffic variations.

- We model and formulate an ILP for the problem of energy aware routing in a hybrid SDN network.

- To validate the solutions, we carried out extensive simulations on several network topologies and show the energy savings for different levels of SDN penetration.

- The mechanisms were implemented and tested on a small SDN platform. The results of the experimentations show that it is *possible to implement energy saving solutions while reducing packet losses* compared to legacy protocols.

II. Related work

Energy aware routing. Energy aware routing has been studied for several years, see for example [4] for backbone networks, [5] for data center networks, [6] for ISP networks, or [7] for wireless networks. The proposed algorithms allow to save from 30% to 50% of the network energy consumption. However, as stated earlier, they imply to do on the fly routing changes.

SDN and Energy aware routing. Multiple works proposed and investigated SDN solutions to implement energy aware routing. For instance, in [8], the authors propose algorithms to minimize the energy consumption of routing by shutting down links while taking into account constraints of SDN hardware such as the size of TCAM memory. Authors in [9] implemented and analyzed ElasticTree, an energy aware routing solution for data center networks. They showed that saving up to 50% can be achieved while still managing traffic spikes. However these solutions require a complete migration of the network to the SDN paradigm.

Hybrid SDN Networks. As the most realistic scenario for the introduction of the SDN paradigm is a progressive migration, we focus on hybrid networks. In these networks, legacy and SDN hardware stand alongside. The difficulty is to make different protocols coexist. Opportunities and research challenges of Hybrid SDN networks are discussed in [1]. Routing efficiently in hybrid networks has been studied in [10]. The authors show how to leverage SDN to improve link utilization, reduce packet losses and delays. We extend this work by considering energy efficiency.

Handling Failures and Flash Crowds. Turning off SDN devices in hybrid IP-SDN networks, can be interpreted as link or node failures by legacy network devices and might decrease the network ability to drain sudden, yet not malicious, traffic surges (due, for instance, to exceptional events such as earthquakes). Consequently, our energy-aware solution implements some features to correctly cope with link failures and flash crowds. The network community has addressed such problems, with the help of SDN, as follows:

- **Link Failure Detection and Mitigation.** As in legacy devices, SDN devices can rely on the legacy BFD algorithm (Bidirectional Forwarding Detection) to detect link failures [11]. Once the link failure has been detected, OpenFlow already offers a link failure mitigation technique through the notion of FAST-FAILOVER group rules, where several rules per flow can be installed. Protection of the link and control channel of OpenFlow requires however more complex solutions, as the one proposed in [12]. To avoid losses in case of link failures in hybrid networks, [3] proposes to introduce pre-set tunnels from a legacy router towards an SDN router, which form backup paths. Later, SDN nodes reroute traffic through non damaged paths. We borrow this idea and propose to use pre-set tunnels, when a node is turned down. This is an adaptation and a generalization of the solution proposed in [3] to handle a link failure. Indeed, we use it for energy efficiency when multiple links are turned off. We also allow tunnels to be set between any (OSPF or SDN) pair of nodes and we carry out practical experimentations to validate the method.

- **Detecting Traffic Variations in SDN Networks.** Traffic variations of backbone networks are usually smooth as the network traffic is an aggregation of multiple flows [2], [13]. However, abrupt variations happen in case of link failures or flash crowd [14]. Methods have been proposed to detect them in legacy networks, see for example [15], [16]. Netfuse [17] has been proposed in SDN-based data centers to mitigate the effect of traffic variations.

III. Energy Aware Routing for Hybrid Networks

A. Model

Routing in a Hybrid Network. A network is modeled as a directed graph $D = (V, A)$ where a node represents a Point of Presence (PoP) and an arc represents a link between two PoPs. A PoP consists of several routers linked together in full mesh [18]. Each link $(u, v) \in A$ is connected to a specific router in PoP u and in PoP v. A link (u, v) has a maximum capacity C_{uv}. We consider hybrid networks in which SDN capable equipments are deployed alongside legacy routers. We consider a scenario in which PoPs do not contain heterogeneous equipments, i.e, all routers are either SDN capable or not. Legacy routers follow a legacy routing protocol, such as OSPF. We denote the next hop to the destination t on a legacy router u by $n^t(u)$. SDN switches are controlled by one or several central controllers and can be configured, dynamically, to route to any of its neighbors.

Traffic estimation. We assume that an ISP is able to estimate the traffic matrix of its network using (sampled) netflow measurements [19] or, in the case of hybrid networks, by combining SDN and OSPF-TE data [10]. Therefore, our solution will monitor traffic and will continuously calculate the set of nodes or links to turn off.

Power Model and Energy Aware Mechanism. To model the power consumption of a link, we use a hybrid model comprised of a baseline cost, representing the power used when the link is active, and a linear cost depending on its throughput. This allows to express the different power models (between ON-OFF and energy proportional) found in the literature, see [20]. The power usage of a link is expressed as follows

$$P_l(u, v) = x_{uv} U_{uv} + \mathcal{F}_{uv} L_{uv}$$

where x_{uv} represents the state of the link (ON or OFF), U_{uv} is the baseline power consumption of an active link, $\mathcal{F}_{uv}$ the total amount of bandwidth on the link, and L_{uv} the power coefficient of the link. Routers have two power states: active or sleep, and their total consumption $P_n(u)$ is given by

$$P_n(u) = B_u + A_u + \sum_{v \in N^+(u)} P_l(u, v)$$

where B_u is the sleep state power usage and A_u the additional power used when the equipment is active.

To save energy, links must be powered down and routers put to sleep. Only SDN switches can be put into sleep mode without negative impact on the network. As it should be done dynamically according to the network traffic, the decision is taken by the SDN controller. Thus, only links with an SDN switch as one of its end point can be shutdown. Since PoPs are interconnected using dedicated routers inside their infrastructure, if a link between two PoPs is shutdown, then each router of the link can be shutdown, if it is SDN capable.

B. Our proposition: SENAtoR

SENAtoR turns off nodes and links based on the traffic load on the PoP links. It can be summarized with these three main propositions that prevent traffic loss.

1. Tunneling. Shutting down a link with the SDN controller results in a failure detection by OSPF and a convergence period. To avoid losing packets during the re-convergence phase, we use pre-set tunnel backup paths to redirect traffic that would otherwise be lost. The idea is to reroute the traffic that would use this down link or node to an intermediate node whose shortest path to destination does not use down links.

With most legacy network mechanisms, tunnels cannot be deployed dynamically during the operation of the network. They have thus to be pre-set statically. We thus consider two variants of the problem: (i) with tunnel selection, (ii) with a pre-configured set of tunnels.

2. Turning off links smoothly. Before putting an SDN PoP switch in powersave mode, the SDN controller stops sending any OSPF packet to its neighbors. This allows neighboring OSPF routers to converge to a network view excluding this node. Indeed, at the expiration of a *dead interval* timer, if no Hello packet is received from a direct neighbor router, an OSPF router declares such a neighbor as dead and stops forwarding traffic to it. The *dead interval* is usually set to 3 $\times$ *hello interval*. The *hello interval* indicates how frequently an OSPF router must send Hello packets. However, while the *dead interval* timer does not expire, the link is considered to be active and traffic flows over this link. This is why in SENAtoR, after the dead interval plus a safety margin of 10 additional seconds, and if no traffic is received through its links (that we define as the OSPF expected convergence period), the SDN PoP switch is put in powersave mode. This simple strategy prevents any additional packet loss.

3a. Traffic spikes mitigation. Sudden traffic spikes are relatively rare due to the high statistical multiplexing in the backbone of ISPs. However, exceptional events (such as earthquakes) can lead to flash crowds. Therefore, we complement SENAtoR with a safeguard mechanism that aims at reactivating inactive SDN PoP switches in case of a sudden traffic spike. The latter event is defined on a per link basis as follows: the controller is collecting the traffic load on each interface of every SDN active switch at a small time scale (in our experiments, once per minute). We then compare the real traffic level received at interface i, $E_i(t)$, to the estimated rate, $E_i^{ES}(t)$, at the last epoch where SENAtoR took its decision of turning off some links. In case $E_i(t) \geq 1.5 \times E_i^{ES}(t)$, for any interface i, all inactive SDN routers are re-enabled. The value of 50% was chosen in a conservative manner, since, in general, ISP networks are over-provisioned. After the OSPF expected convergence period, the controller reruns SENAtoR to obtain a new green architecture if possible.

3b. Link failure mitigation. We employ a mechanism similar to the traffic spike mitigation mechanism in case of link failures. When a link connected to an SDN active router or in between OSPF nodes fails, SENAtoR turns on again any inactive SDN node. It also directly reroute the traffic through a different path if possible (including the pre-set tunnels). A link failure in between OSPF nodes can be detected by nearby SDN nodes due to the traffic variation at their network links. A downstream link, with regard to a failed link, will indeed observe a decrease of the rate of one interface as compared to what the traffic matrix predicts. We benefit from the fact that in typical ISP networks, traffic is all-to-all, i.e., from one PoP (Point of Presence) to any other PoP. Hence, any SDN router in the network is likely to detect the link loss, as a fraction of the traffic it handles is affected by the failure. Again we use a conservative threshold of 50%, i.e., an SDN switch must detect a decrease of 50% of any of its links' load to trigger the link failure mitigation mechanism. Once again, after the OSPF convergence expected period, the controller reuses SENAtoR to obtain eventually a new green architecture.

Summary. When an SDN switch has to be put in sleep mode and links have to be shutdown, the mechanism is the following: the SDN controller first reroutes the traffic so that no flows are passing through this node or link. Then, the SDN controller sends the order to the SDN switch to enter into sleep mode or to disable the interface corresponding to the link. Since no more data packets are using the link, the interface of the SDN node is turned off and the interface of legacy router can automatically enter into sleep, using for instance IEEE 802.3az Energy-Efficient Ethernet [21].

C. How to route and select off-link with SENAtoR

We propose an Integer Linear Program that decides which network equipments to put into sleep mode, and at the same time, which tunnels to set to reroute the traffic. The ILP includes also SDN node placement problem. Due to lack of space, we do not present here the equations of the linear program, it can be found in [22]. The formulation presents several difficulties. First, legacy nodes have to route flows through shortest paths following legacy protocols, while SDN nodes can route a flow freely to any neighbors. Second, tunnels have to be set in a way there exists a path for each flow, even when several network equipments are put into sleep mode.

The ILP can be used to find good solutions for small sized instances, see Section IV. The computation time is however prohibitive to find optimal solution as the problem is NP-complete (indeed, it comprises as subproblem the EAR problem which is NP-complete [23]). For larger instances, it is even impossible to find feasible solutions using the ILP. We thus propose in the following an efficient heuristic algorithm

SENAtoR to solve the problem of Energy Aware Routing for Hybrid Networks. This heuristic has two steps: first, it assigns routes to the flows using eventually tunnels, then it selects the equipments to turn-off. Note that two possibilities for the configuration of the tunnels are considered, (i) with dynamic tunnel selection, (ii) with a pre-configured set of tunnels.

1) Path Assignment: To assign a path to a demand, we build a weighted residual graph $H_{st} = (V, A' \subseteq A)$ and then search for the shortest path between s and t in H_{st}. Nodes in H_{st} are the ones of D and correspond to network routers. We only consider links and tunnels which have enough residual capacities to satisfy the demand D_{st}. For each node u, its set of out-neighbors is constructed as follows:

If u is a legacy node, the routing is done by the legacy routing protocol towards next hop $n^t(u)$ if the link to $n^t(u)$ is active. In this case, the only neighbor of u in H_{st} is $n^t(u)$. Otherwise, if the link to $n^t(u)$ is inactive, the routing is done through a tunnel. (1) If a tunnel from u is already defined for the destination t, the neighbor of u in H_{st} is set as the tunnel endpoint. (2) If no tunnel is defined, the next step depends on the variant of the problem. $(2i)$ In the tunnel selection variant, we have to set a tunnel. We thus add all the potential tunnels by adding any node that can reach the destination t, using direct forwarding (OSPF or OpenFlow) or existing tunnels. $(2ii)$ With pre-configured set of tunnels, u has no neighbor in H_{st}.

If u is an SDN node, the routing is done by OpenFlow rules installed by the controller. We have two cases: (1) if no OpenFlow rule is set for the demand in node u, any neighbor can be the next hop. The neighbors of u in H_{st} are the same as in the original digraph D. (2), we only add as neighbor of u in H_{st} the node designed as the next hop by OpenFlow. Similar to legacy node, if the link to the next hop given by OpenFlow is inactive, we consider tunnels in the same way described above.

A weighted shortest path from s to t will then be computed in the residual graph H_{st} leading to the decision of which tunnel will be selected and whether we need to install or not a new OpenFlow rule for the SDN node.

2) Off Link Selection: Once all demands have been assigned a path, we try to power off links to save energy. We consider SDN links one by one, i.e., links with at least one SDN endpoint. We select the active link with the smallest amount of traffic on both arcs. We then try to reroute all the demands flowing through that link. If no valid routing can be found, the link is set as *non-removable* and the previous routing is restored. If a valid routing is found, the link is set as *inactive* and powered off. We then consider the remaining active links. The heuristic stops when all SDN links are either powered off or *non-removable*.

IV. Numerical evaluation

In this section, we evaluate the solutions proposed on different ISP topologies. We first compare the performances of the ILP and of the heuristic algorithm on a small topology. We then use SENAtoR on larger networks of SNDLib. We show that *SENAtoR obtains energy savings that range from 5% up to 35% for different levels of SDN hardware installation.*

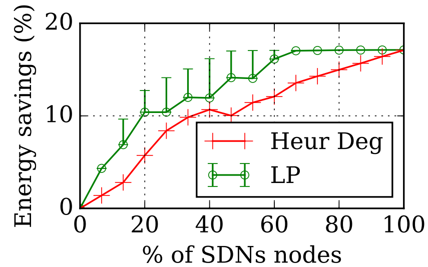


Fig. 1: Energy Savings for the ILP and the heuristic on `atlanta`.

For the parameters of the power model, we considered the cases of two different hardware: our HP5412zl SDN switch and an *ideal energy efficient* SDN switch as discussed in [24]. In the first case, we measured the power consumption using a wattmeter: the switch uses 95W when in sleep mode and 150W if it is active ($B_u = 95, A_u = 55$). According to Cisco specifications [25], links are using 30W as a baseline and go up to 40W when at full capacity ($U_{uv} = 30, L_{uv} = 10$). In order to have a fast recovery from sleep mode, the TCAM must be kept under power to preserve the forwarding rules. According to [26], TCAM represents 30% of the consumption of a high end router, and considering results from [24], we can safely assume that an *ideal energy efficient* switch could save up to 60% of energy in sleep mode.

For the choice of SDN nodes in the networks, we tested and evaluated different methods such as node degree, centrality, and covering (*betweenness centrality*, *closeness centrality* and Max k-Vertex Cover). Finally, we chose the simplest one in terms of computation and that gives similar results: *the node degree*. The resulting selection is: first sort all nodes according to their degree; second, choose the k first nodes. This method has the advantages of being simple and to allow a good incremental upgrade to SDN hardware.

A. ILP vs. Heuristic

We use the `atlanta` network (composed of 15 nodes and 22 links) and the traffic matrices provided by SNDLib to compute the energy savings for different number of SDN nodes. We solve the ILP with CPLEX and set a time limit of one hour (as the ILP is complex). The results presented correspond to the best solution found by the solver within the time limit. Note that for percentage of SDN nodes below 13% and greater than 73%, the ILP solves the problem optimally in less than one hour. The heuristic takes at most $5\,\mathrm{ms}$ to find a solution in all settings.

We see in Figure 1 the possible energy savings. The error bars for the ILP represent the relative gap of the solution provided by CPLEX when the time limit is reached. The solutions provided by the heuristic save at most 5% less than the ones found with the ILP. The energy savings range from 0 when no links can be turned off to 17% when the network is pure SDN and when all links can be turned off and nodes put to sleep. We can thus suppose that the heuristic provides good solutions. We thus use it to study larger networks for which the ILP cannot even find a feasible solution in a reasonable amount of time.

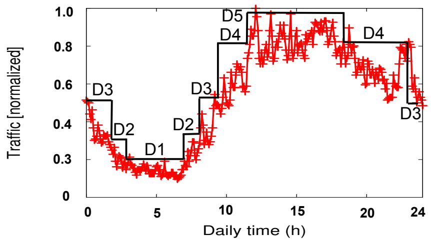


Fig. 2: Daily traffic in multi-period.

B. *Simulations on larger networks*

We further look at the performance of the heuristic on `atlanta` and on a larger network, `ta2` (65 nodes and 108 links).

a) Traffic Model: Since ISP traffic is roughly stable over time with clear daily patterns, a few traffic matrices is enough to cover a whole day period. Consequently, a relatively small number of routing reconfigurations allows operators to obtain most of the energy savings [20] and avoid making frequent reconfigurations. Indeed, as exemplified by the daily variations for a typical link in the Orange ISP network (Figure 2), five traffic matrices (labeled D1 to D5) are sufficient. These matrices are normalized and adapted to the size of each studied topology.

Then, we compute the best hybrid energy aware routing for each matrix and adapt the routing when the traffic changed.

b) Daily savings: In Figure 3, we compare the energy savings during the day for the two topologies. The top figures represent the savings with HP switches and the bottom ones the savings with *ideal energy efficient* switches. We look at 4 different levels of SDN deployment: 10%, 25%, 50% and 100% of upgraded nodes in the network. For each period, we compare the energy used to the one of a legacy network at the same period. On a full SDN network, the difference between night and day energy savings is between 4% and 7% (4 and 9% with *ideal switches*). With HP switches, we can save up to 19% on `atlanta` and 21% on `ta2` with a full SDN networks. With *ideal* switches, we obtain higher savings, between 25% and 31%.

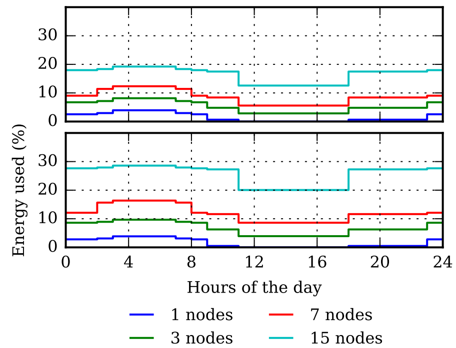
Due to lack of space, we do not present additional plots for other networks (`germany50` (50 nodes and 88 links) and `zib54` (54 nodes and 81 links)). These results together with simulations on number of tunnels, delay and additional stretch can be found in [22]. We show in this report that our energy-aware solution has a limited impact on these parameters. Concerning the median delay, it rarely goes above 10 ms for all four networks.

V. Experimentations

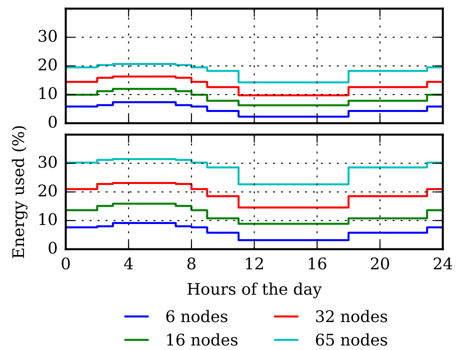
In this section, we present results obtained on a Mininet testbed with the SENAtoR solution. Our objective is to demonstrate that SENAtoR can indeed turn off links and put SDN switches in power save mode without losing packets thanks to our smooth integration with OSPF to anticipate link shutdown.

A. *Testbed*

We built a hybrid SDN testbed using Mininet and a remote Floodlight controller. The Mininet network topology is based



(a) `atlanta`



(b) `ta2`

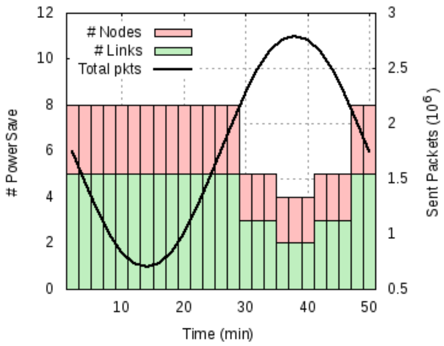
Fig. 3: Daily energy savings over the day for the (a) `atlanta`, and (b) `ta2` networks. with 10, 25, 50 and 100% SDN nodes deployment. Top plots: power model of the HP switch. Bottom plots: power model of an ideal energy efficient SDN switch.

on `atlanta` with 50% SDN deployment. OSPF routers are materialized as host nodes in Mininet and run the Quagga software while Open vSwitches (OvS) act as SDN switches. Our Floodlight controller is able to parse and respond to OSPF hello packets received and forwarded by the SDN OvS switches (through adequate Openflow rules installed in the SDN switches) ; hence ensuring the correct functioning of the adjacent OSPF routers. Tunnels are implemented as simple GRE tunnels and the interplay between the tunnel interface and the regular interfaces is controlled by tuning the administrative distance so that regular interfaces have a higher priority. When SENAtoR notifies an SDN PoP switch to got into sleep mode, we turn off all of its interfaces and disconnect it from the rest of the network. During this powersave mode, the memory keeps the set of rules previously installed by the controller in order to perform a quick recovery back to normal active mode.

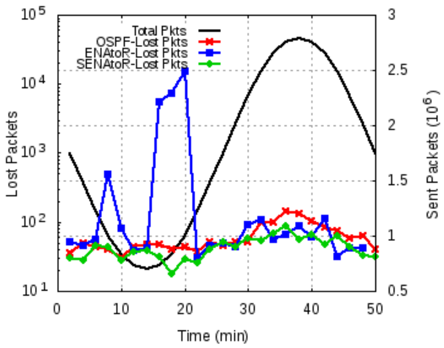
B. *Lossless link turn-off.*

In Figure 4a we vary the traffic over time in order to simulate smooth variations on the average rate. This is achieved by taking one traffic matrix and scaling it using the same sinusoidal function as in Figure 2.

The energy saving results in Figure 4a are in line with the ones of Section IV Figure 3, i.e., same number of links and nodes turned off in all cases, which is reassuring as we use the same code at the controller. The added value of



(a) Number of turned off links



(b) Packet loss

Fig. 4: `atlanta` topology

the experiment is to assess if the interplay between SDN and OSPF is effective, i.e., that our smooth link shutdown approach effectively avoids data losses. Figure 4b portrays the time series of packet loss with pure OSPF (OSPF operates the complete network and no link is turned off in this case), SENAtoR and ENAtoR (SENAtoR without the smooth link shutdown). The figure shows the importance of anticipating the link shutdown (resulting from putting SDN switches in sleep mode) as is done in SENAtoR as losses explode to 10^4 packets when this feature is disabled (ENAtoR). In this case, the high loss rate of ENATOR is proportional to the amount of times it takes for OSPF to declare the link down multiplied by the traffic intensity. In contrast, SENAtoR manages to maintain the same packet loss as a full OSPF network without any links shutdown, with negligible loss rates ($10^{-4}\%$), even though it is using less links and nodes in the network.

C. *Traffic spikes*

To illustrate the traffic spike mitigation mechanism, we consider a fixed traffic matrix (no scaling) and we induce a traffic spike either at an OSPF node directly connected to an SDN switch (Figure 5a) or between OSPF nodes (Figure 5b). We report the cumulative distribution function (CDF) of loss rates of all connections. Clearly, the spike detection algorithm of SENAtoR allows it to outperform OSPF. One of the reasons of such a phenomenon is that regular OSPF nodes have no mechanisms to automatically load balance packets in case of traffic spikes.

D. *Link failure*

We consider again a fixed traffic matrix (no scaling) and we induce a link failure either between an SDN switch and an OSPF router or in between two OSPF routers and report the corresponding loss rates on Figures 6a and 6b. We compare three cases: (i) the legacy OSPF scenario, in which the link failure is handled with a long convergence time, (ii) SENAtoR using OSPF Link State (LS) Updates only to detect network changes; and (iii) SENAtoR with its *Link failure* detection and mitigation mechanism.

We first observe that even in case (ii), *SENAtoR does not experience higher loss rates than case (i)* (and significantly lower loss rates when failure on OSPF-OSPF link). This happens *even though some of the switches and links were down at the time of the failure*, and had to be switched on. Indeed, SDN switches do not need to wait for the OSPF convergence before rerouting traffic through the pre-established set of tunnels. The link failure mitigation mechanism further improves the situation.

We then observe a counter intuitive result: the loss rates using SENAtoR are smaller when the failure occurs on an OSPF-OSPF link rather than on an SDN-OSPF link. Two factors contribute to this result. First, SDN nodes are placed at key locations in the network such that they convey more traffic. Hence, a failure at these nodes induces higher loss rates. Second, as soon as a downstream SDN node detects a link failure in an OSPF-OSPF link, SENAtoR limits the traffic flowing on this link by instructing upstream SDN nodes to reroute their traffic.

VI. Conclusion

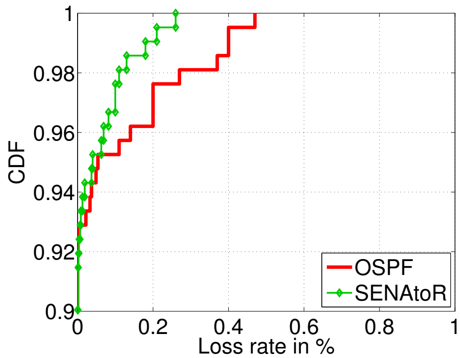
In this paper, we presented SENAtoR, an energy aware routing solution that preserves failure tolerance and traffic overload management of the network. SENAtoR was enriched with lossless link/node turn-off, spikes, and traffic failure detection services. SENAtoR implementation and experimentation with emulated devices running full OSPF agents shows that we can deal with unexpected network events correctly. More strikingly, our experiments show that even when green services are enabled and traffic spikes occur in a non SDN capable node, SENAtoR provides loss rates lower than the all-OSPF case, since the SDN controller can provide most appropriate routes. As a conclusion, SENAtoR provides energy savings while being compatible with current network infrastructures. As a future work, SENAtoR can be enriched with a deeper study about the traffic network variations in order to provide the most adapted thresholds for the spikes and link failure detections.

References

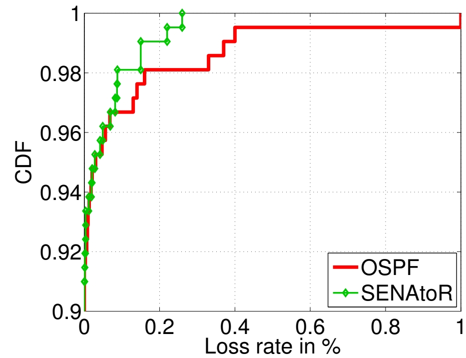
[1] S. Vissicchio, L. Vanbever, and O. Bonaventure, “Opportunities and research challenges of hybrid software defined networks,” *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 2, 2014.

[2] M. Roughan, A. Greenberg, C. Kalmanek, M. Rumsewicz, J. Yates, and Y. Zhang, “Experience in measuring backbone traffic variability: Models, metrics, measurements and meaning,” in *ACM IMW*, 2002.

[3] C.-Y. Chu, K. Xi, M. Luo, and H. J. Chao, “Congestion-aware single link failure recovery in hybrid sdn networks,” in *IEEE INFOCOM*, 2015.

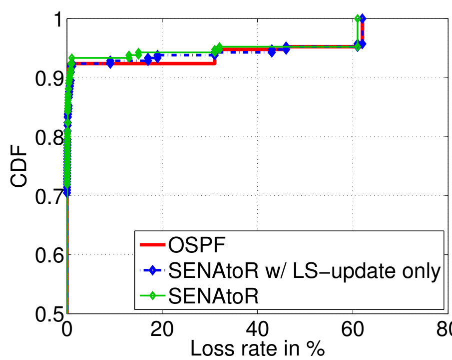


(a) In an SDN-OSPF link

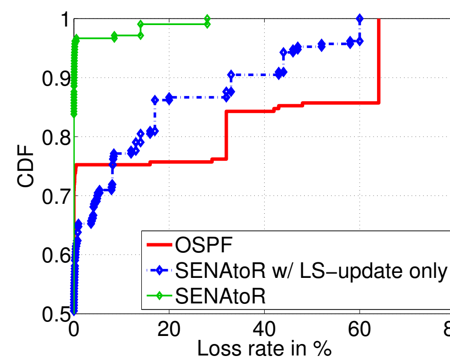


(b) In an OSPF-OSPF link

Fig. 5: Traffic spike experiment with the `atlanta` topology



(a) On an SDN-OSPF link



(b) On an OSPF-OSPF link

Fig. 6: Link failure experiment with the `atlanta` topology

[4] E. L. Rouzic, E. Bonetto, L. Chiaraviglio, F. Giroire, F. Idzikowski, F. Jimenez, C. Lange, J. Montalvo, F. Musumeci, I. Tahiri, A. Valenti, W. V. Heddeghem, Y. Ye, A. Bianco, and A. Pattavina, "Trend towards more energy-efficient optical networks," in *ONDM*, Brest, France, 2013, pp. 211–216.

[5] Y. Shang, D. Li, and M. Xu, "Energy-aware routing in data center network," in *ACM workshop on Green networking*, 2010.

[6] R. Modrzejewski, L. Chiaraviglio, I. Tahiri, F. Giroire, E. Le Rouzic, E. Bonetto, F. Musumeci, R. Gonzalez, and C. Guerrero, "Energy Efficient Content Distribution in an ISP Network," in *IEEE Global Communications Conference (GLOBECOM 2013)*, Atlanta, United States, Dec. 2013.

[7] F. De Rango, F. Guerriero, and P. Fazio, "Link-stability and energy aware routing protocol in distributed wireless networks," *IEEE Transactions on Parallel and Distributed systems*, vol. 23, no. 4, 2012.

[8] F. Giroire, J. Moulierac, and T. K. Phan, "Optimizing rule placement in software-defined networks for energy-aware routing," in *IEEE GLOBECOM*, 2014.

[9] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, "Elastictree: Saving energy in data center networks." in *NSDI*, 2010.

[10] S. Agarwal, M. Kodialam, and T. Lakshman, "Traffic engineering in software defined networks," in *IEEE INFOCOM*, 2013.

[11] D. Katz and D. Ward, "Bidirectional Forwarding Detection (BFD)," RFC 5880, Oct. 2015.

[12] S. Sharma, D. Staessens, D. Colle, M. Pickavet, and P. Demeester, "In-band control, queuing, and failure recovery functionalities for openflow," *IEEE Network*, vol. 30, no. 1, 2016.

[13] G. Iannaccone, C. Diot, I. Graham, and N. McKeown, "Monitoring very high speed links," in *ACM IMW*, 2001.

[14] A. Lakhina, M. Crovella, and C. Diot, "Characterization of network-wide anomalies in traffic flows," in *ACM IMC*, 2004.

[15] ——, "Mining anomalies using traffic feature distributions," in *ACM Computer Communication Review*, vol. 35, no. 4, 2005.

[16] I. Ari, B. Hong, E. L. Miller, S. A. Brandt, and D. D. Long, "Managing flash crowds on the internet," in *IEEE/ACM MASCOTS 2003*, 2003.

[17] Y. Wang, Y. Zhang, V. Singh, C. Lumezanu, and G. Jiang, "Netfuse: Short-circuiting traffic surges in the cloud," in *IEEE ICC*, 2013.

[18] F. Giroire, A. Nucci, N. Taft, and C. Diot, "Increasing the robustness of ip backbones in the absence of optical level protection," in *IEEE INFOCOM*, 2003.

[19] B. Claise, "RFC 3954 - Cisco Systems NetFlow Services Export Version 9," 2004.

[20] F. Idzikowski, L. Chiaraviglio, A. Cianfrani, J. L. Vizcaíno, M. Polverini, and Y. Ye, "A survey on energy-aware design and operation of core networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, 2016.

[21] K. Christensen, P. Reviriego, B. Nordman, M. Bennett, M. Mostowfi, and J. A. Maestro, "Ieee 802.3 az: the road to energy efficient ethernet," *IEEE Communications Magazine*, vol. 48, no. 11, 2010.

[22] N. Huin, M. Rifai, F. Giroire, D. L. Pacheco, G. Urvoy-Keller, and J. Moulierac, "Bringing Energy Aware Routing closer to Reality with SDN Hybrid Networks," INRIA Sophia Antipolis - I3S, Research Report RR-9020, Jan. 2017.

[23] F. Giroire, D. Mazauric, J. Moulierac, and B. Onfroy, "Minimizing routing energy consumption: from theoretical to practical results," in *IEEE/ACM GreenCom*, 2010.

[24] T. Vu, V. Luc, N. Quan, T. Thanh, N. Thanh, and P. Nam, "Sleep mode and wakeup method for openflow switches," *Journal of Low Power Electronics*, vol. 10, no. 3, pp. 347–353, 2014.

[25] "Cisco 1941 series integrated services routers data sheet."

[26] P. T. Congdon, P. Mohapatra, M. Farrens, and V. Akella, "Simultaneously reducing latency and power consumption in openflow switches," *IEEE/ACM Transactions on Networking (TON)*, 2014.