

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

▶ **To cite this version:**

Nicolas Huin, Myriana Rifai, Frédéric Giroire, Dino Lopez Pacheco, Guillaume Urvoy-Keller, et al.. Bringing Energy Aware Routing closer to Reality with SDN Hybrid Networks. IEEE Transactions on Green Communications and Networking, 2018, pp.1128 - 1139. 10.1109/TGCN.2018.2842123 . hal-01877868

HAL Id: hal-01877868
https://hal.science/hal-01877868

Submitted on 3 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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 Internet Service Provider (ISP) networks. The idea is to adapt routing to the traffic load 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 devices is a delicate task as it can lead to packet losses, SENAtoR also provides 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 us 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 centralized 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 an extended period 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 Open Shortest Path First (OSPF), while SDN routers[1], through their SDN controller, 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 minimize the number of used network devices for the current traffic. Then SENAtoR puts some SDN routers 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 devices over the course of the day.

When SDN routers are put into sleep mode, 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:

A. 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 router 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.

B. 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 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. If it receives no more traffic, the SDN router will effectively turn off the appropriate NICs after a period higher than the link-failure detection period and higher than the convergence of OSPF. We can estimate the convergence time with OSPF timer values. Note that while OSPF has not converged yet, packets can be rerouted through the pre-set tunnels; and since the link and node are still on, packets are not lost during the routing transition.

C. Traffic Spike and link failure mitigation. Energy-aware algorithms exploit network capacity over-provisioning 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 devices, to not impact the failure tolerance

[1]For sake of consistency, we always talk about SDN routers in this paper, even if, formally, an SDN node is able to understand rules mixing fields from layer 2 to layer 4 of the protocol stack.

of networks. We exploit the metrology data received by the controller from SDN routers to detect significant 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: tunnelling, smooth link shutdown, and detection of traffic variations.

- We model and formulate an Integer Linear Program (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 showed 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 savings from 30% to 50% of the network energy consumption. However, as stated earlier, they require on the fly routing changes, which can affect the Quality Of Service (QoS), if not done properly.

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 Ternary Content-Addressable Memory (TCAM). TCAM is a specialized type of high-speed memory that searches its entire contents in a single clock cycle and that is used within SDN devices. 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 routers 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:

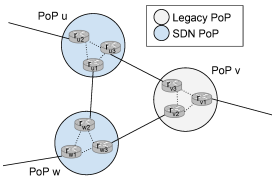


Fig. 1: Hybrid SDN/OSPF network with 3 PoPs.

- **Link Failure Detection and Mitigation.** As in legacy routers, SDN routers 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 legacy routers towards an SDN router, which form backup paths. Later, SDN routers reroute traffic through non-damaged paths. We borrow this idea and propose to use pre-set tunnels, when a node is shut 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 crowds [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. We model the network 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, see Figure 1. Each link (u, v), that represents a connection from one router in u to one router in v, has a maximum capacity C_{uv}.

We consider hybrid networks in which SDN routers are deployed alongside legacy routers.

We consider a scenario in which PoPs do not contain heterogeneous devices, 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 routers 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 can estimate the traffic matrix of its network using (sampled) NetFlow measurements [19] or, in the case of hybrid networks, by combining SDN and OSPF data [10]. Therefore, our solution can monitor traffic and calculate continuously 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 us to consider 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 routers can be put into sleep mode without negative impact on the network. Indeed, all the features needed to smoothly turn off network links without introducing packet losses can only be implemented in SDN routers, as explained in Section III-B. 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 router as one of its end point can be shut down. Since PoPs are interconnected using dedicated routers inside their infrastructure, if a link between two PoPs is shut down, then each router of the link can be shut down, 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 relies on three key mechanisms to prevent traffic loss.

A. Tunneling. Shutting down a link with the SDN controller results in 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 downlink or node to an intermediate node whose shortest path to destination does not use downlinks.

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.

B. Turning off links smoothly. Before putting an SDN PoP router in power save 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[2] 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 router is put in power save mode. This simple strategy prevents any additional packet loss.

C1. 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 sudden traffic spikes. Therefore, we complement SENAtoR with a safeguard mechanism that aims at reactivating inactive SDN PoP routers 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 router 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. We conservatively choose a value of 50% 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.

C2. 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 router. It also directly reroutes the traffic through a different path if possible (including the pre-set tunnels). Nearby SDN routers can detect a link failure in between OSPF nodes due to the traffic variation at their network links. A downstream link, with regard to a failed link, will indeed observe a decrease in 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 router must detect a decrease of 50% of any of its links' load to trigger the link failure mitigation mechanism. Once again, after the OSPF expected convergence period, the controller reuses SENAtoR to obtain a new green architecture eventually.

[2]An HELLO packet is a special message that is sent out periodically from a router to establish and confirm network adjacency relationships.

Summary. When an SDN router has to be put in sleep mode, and links have to be shut down, 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 orders the SDN router 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 router is turned off, and the interface of the legacy router can automatically enter into sleep, using, for instance, IEEE 802.3az Energy-Efficient Ethernet [21].

1) Integer Linear Program: We propose an Integer Linear Program (ILP) that decides which network devices to put into sleep mode, and at the same time, which tunnels to set to reroute the traffic[3]. The Integer Linear Program also includes the SDN router placement problem. A summary of the notations is found in Table I. The formulation presents several difficulties. First, legacy nodes have to route flows through shortest paths following legacy protocols, when SDN routers can route a flow freely to any neighbors. Second, tunnels have to be set in such a way that there exists a path for each flow, even when several network devices are put into sleep mode.

The objective function (1) aims at reducing the power consumption of the network with at most k SDN PoPs (2). The flow conservation constraints are given by (3) and the capacity constraints by (4). Constraints (5) to (7) limit the usage of a backup tunnel. Constraints (8) and (9) determine if the next hop of a given demand is active or not. Combining (10) and (11), we ensure that at most one next hop can be selected for a source-destination pair on a router. If it is not an SDN router, the next hop can only be the OPSF next hop. A link can only be shut down if one of its end points is an SDN router (12). Links between two routers share the same state (13). Finally, with (14) and (15), a router can only be put to sleep if all its links to other PoPs can be shut down and if it is an SDN router.

$$\min \sum_{u \in V} P_n(u) \tag{1}$$

$$\sum_{u \in V} s_u \leq k \tag{2}$$

$$\sum_{\{p(u,x) \in \mathcal{P} | x \neq u\}} g_p^{st} - \sum_{\{p(x,u) \in \mathcal{P} | x \neq u\}} g_p^{st} + \sum_{v \in N^+(u)} f_{uv}^{st} - \sum_{v \in N^-(u)} f_{vu}^{st} = \begin{cases} 1 & \text{if } u = s, \\ -1 & \text{if } u = t, \\ 0 & \text{else} \end{cases} \quad \forall (s,t) \in \mathcal{D}, u \in V \tag{3}$$

$$\sum_{(s,t) \in \mathcal{D}} D^{st} \left(f_{uv}^{st} + \sum_{\{p(s,t) | (u,v) \in p\}} g_p^{st} \right) \leq x_{uv} C_{uv} \quad \forall (u,v) \in A \tag{4}$$

$$g_p^{st} \times len(p) \leq \sum_{(u,v) \in p} n_{uv}^{xy}, \quad \forall p(x,y) \in \mathcal{P}, (s,t) \in \mathcal{D} \tag{5}$$

$$g_p^{st} \leq h_{xy}^{t}, \quad \forall p(x,y) \in \mathcal{P}, (s,t) \in \mathcal{D} \tag{6}$$

$$g_p^{st} \leq e_u^{st}, \quad \forall p(u,x) \in \mathcal{P}, (s,t) \in \mathcal{D} \tag{7}$$

$$n_{uv}^{st} - x_{uv} \leq e_u^{st}, \quad \forall (u,v) \in A, (s,t) \in \mathcal{D} \tag{8}$$

$$e_u^{st} \leq 2 - n_{uv}^{st} - x_{uv}, \quad \forall (u,v) \in A, (s,t) \in \mathcal{D} \tag{9}$$

$$\sum_{v \in N^+(u)} n_{uv}^{st} \leq 1, \quad \forall u \in V, (s,t) \in D \tag{10}$$

$$n_{uv}^{st} \leq s_u, \quad \forall (s,t) \in \mathcal{D}, (u,v) \in A \mid v \neq n^t(u) \tag{11}$$

$$x_{uv} \geq 1 - s_u - s_v, \quad \forall (u,v) \in A \tag{12}$$

$$x_{uv} = x_{vu}, \quad \forall (u,v) \in A \tag{13}$$

$$r_{uv} \geq 1 - s_u, \quad \forall (u,v) \in A \tag{14}$$

$$r_{uv} \geq x_{uv}, \quad \forall (u,v) \in A \tag{15}$$

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

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 [22]). 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 equipment 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 router*, 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

[3]It also solves the simpler problem by fixing the variables corresponding to pre-set tunnels to 1.

TABLE I: Notations used for the ILP

Input parameters	
D^{st}	charge of the demand between s and t.
C_{uv}	capacity of link (u, v)
$\mathcal{P}$	set of all paths
$p(s, t)$	set of path between s and t
$\mathcal{T}$	set of all destinations
Decision variables	
$f_{uv}^{st} \in \{0, 1\}$	demand between s and t is forwarded without tunnels between u and v
$g_p^{st} \in \{0, 1\}$	demand between s and t is forwarded on the tunnel p
$h_{ux}^{t} \in \{0, 1\}$	a tunnel to x from u for packet with destination to t is used
$n_{uv}^{st} \in \{0, 1\}$	v is the next hop on u for the demand between s and t
$x_{uv} \in \{0, 1\}$	link (u, v) is on
$s_u \in \{0, 1\}$	u is an SDN PoP
$e_u^{st} \in \{0, 1\}$	next hop on u for demand between s and t is inactive.
$r_{uv} \in \{0, 1\}$	router in PoP u connected to PoP v is on

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 router.

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 34% for different levels of SDN hardware installation.*

For the parameters of the power model, we considered the cases of two different hardwares: our HP5412zl SDN router and an *ideal energy efficient* SDN router as discussed in [23]. 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 [24], 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 [25], TCAM represents 30% of the consumption of a high-end router, and considering results from [23], we can safely assume that an *ideal energy efficient* switch could save up to 60% of energy in sleep mode. We compare the two on a small topology, `atlanta` composed of 15 nodes and 33 links and then evaluate the performance of the heuristic on larger networks such as `germany50` (50 nodes and 88 links), `zib54` (54 nodes and 81 links) and `ta2` (65 nodes and 108 links).

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

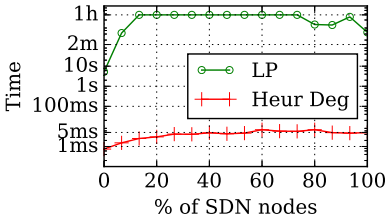


Fig. 2: Computation time of the ILP and of the heuristic on `atlanta`.

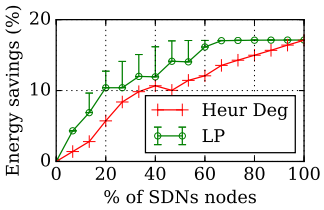


Fig. 3: Energy Savings for the ILP and for the heuristic on `atlanta`.

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 thus: first, sort all nodes according to their degree; second, choose the k first nodes. This method has the advantages of being simple and allowing small changes in the configuration when new SDN equipements are deployed.

A. ILP vs. Heuristic

We use the `atlanta` network (composed of 15 nodes and 22 links) and the traffic matrices provided by SNDLib[4] to compute the energy savings for different fractions of SDN nodes. We solve the ILP with CPLEX (which is a High-performance mathematical programming solver for linear programming) and set a time limit of one hour (as the ILP is has a large number of variables and constraints). The results presented correspond to the best solution found by the solver within the time limit. Note that for fractions 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 ms to find a solution in all settings.

[4]SNDlib [26] is a library of test instances for Survivable fixed telecommunication Network Design.

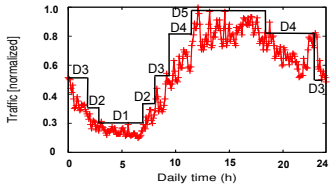


Fig. 4: Daily traffic in multi-period.

Computation Time. In Figure 2, we show the computation time of the two solutions. We see that the ILPs are complex to solve as the limit of 1 hour of computation time is reached for a large number of points (percentage of SDN nodes between 13 and 73%). Points which are solved quickly correspond to easier settings: when $k = 0$, no SDN node has to be placed thus no energy saving is possible. When k is large ($\geq 73\%$), we are close to a full SDN network. Indeed, almost all links are covered by SDN nodes and thus can be turned off. Moreover, the large density of SDN nodes allows to reroute almost all traffic. These settings thus are easier to solve. On the contrary, the heuristic is always fast: it takes at most $5\,\mathrm{ms}$ to find a solution in all settings.

Energy Savings. We see in Figure 3 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 a pure SDN one and when all links can be turned off and nodes put to sleep. We can thus suppose that the heuristic provides solutions with gains not too far from the optimal solution. We thus use it to study larger networks for which the ILP cannot even find a feasible solution in a reasonable amount of time.

B. Simulations on larger networks

We further looked at the performance of the heuristic on `atlanta` (15 nodes and 22 links), `germany50` (50 nodes and 88 links), `zib54` (54 nodes and 81 links) and `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 are enough to cover a whole day period. Consequently, a relatively small number of routing reconfigurations is needed for the 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 4), five traffic matrices (labeled D1 to D5) are sufficient. These matrices are normalized and adapted to the size of each studied topology.

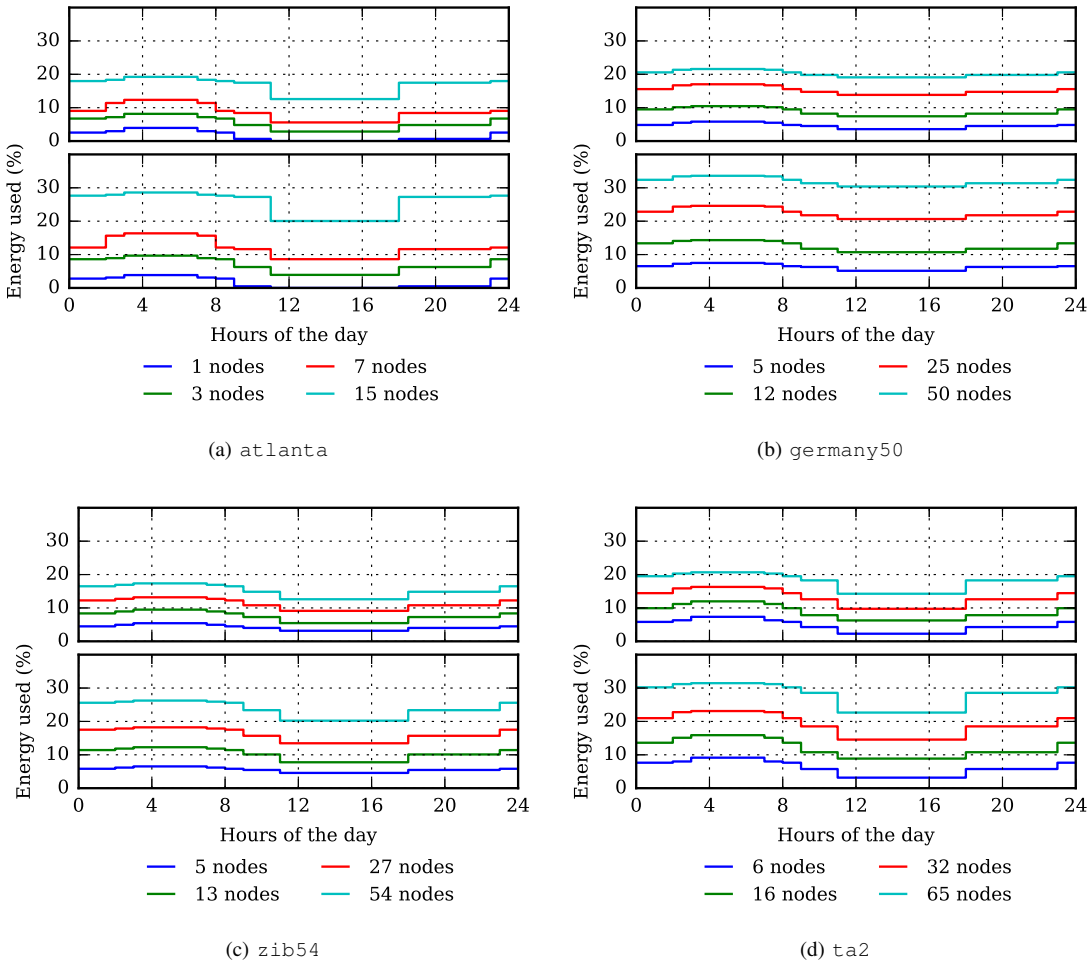
Then, we compute the best hybrid energy-aware routing for each matrix and adapt the routing when the traffic changes.

b) Daily savings: In Figure 5, we compare the energy savings during the day for the four topologies. The top figures represent the savings with HP switches and the bottom ones the savings with *ideal energy efficient* switches. We look at four 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`, 22% on `germany50`, 17% on `zib54` and 21% on `ta2` with a full SDN network. With *ideal* switches, we obtain higher savings, at least 20% for all the topologies and for all the matrices, and up to 34% (reached for `germany50`). We observe that the energy saved on the network increases with the number of SDN routers. With only 10% of SDN nodes in the network and for HP switches, we can save up to 4% on `atlanta`, 7% on `germany50` and `zib54`, and 9% on `ta2`. Deploying SDN nodes from 10% to 25% of the network increases, on average, the savings by 4% on all networks and increasing the number of nodes to 50% save an additional 4% to 6% on average. Finally, the energy savings of a full SDN network are between 13 and 19% on `atlanta`, 19 and 22% on `germany50`, 13 and 17% on `zib54` and 14 and 21% on `ta2`. On the `atlanta` network, with a 10% deployment we can only save power during the periods with the lowest amount of traffic. On larger networks, the energy savings for partial deployment is more significant. With only a 10% deployment, we can already shut down between 5% and 9% of the links in the network. With 25% of SDN nodes, we can save between 7% and 16% of energy for all three networks. At most, we can save 34% on `germany50`, 26% on `zib54` and 31% on `ta2`.

c) Number of tunnels: The number of tunnels used is presented in Figure 6. For small SDN budgets (up to 30% of the network for `atlanta`, 20% for larger networks), the average number of tunnels greatly increases with the number of SDN nodes. The reason is that more network links may be turned off, and thus, more backup tunnels are needed. The number of tunnels then levels off and decreases. Indeed, with a large penetration of SDN in the network, SDN nodes can dynamically forward the traffic regardless of OSPF, and the traffic can be rerouted before arriving at the turned-off link. Thus, fewer backup tunnels are needed. The maximum average number of tunnels needed per node is proportional to the size of the network (3 for `atlanta`, 8 for `germany50`, 9 for `zib54` and 15 for `ta2`). Finally, while the number of tunnels needed may seem high, we see in the next section that the impact of this overhead on the network performance (packet loss or delay) is not noticeable.

d) Stretch and delay: By nature, Energy-Aware Routing has an impact on the length of the route in the network. As we turn off links, we remove shortest paths. Moreover, tunnels can also increase the path length. We use the stretch ratio metrics to measure this increase. The stretch ratio of a path between s and t is the ratio of the path length divided by the length of a shortest path between s and t. In Figure 7, we show the stretch ratio of the paths for four levels of SDN deployment. We only show the stretch for the period with the lowest amount of traffic, as it is the period with the largest number of turned off links and thus the one with the largest stretch.

Most of the demands are barely affected by SENAtoR. The median stays around a ratio of 1 with a maximum of 1.25 for `atlanta` at 100% deployment, 1.25 for `germany50` at



(a) `atlanta` (b) `germany50`

(c) `zib54` (d) `ta2`

Fig. 5: Daily energy savings over the day on (a) `atlanta`, (b) `germany50`, (c) `zib54` and (d) `ta2`, 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 router.

50% deployment, 1.33 for `zib54` at 10%, and 1.25 for `ta2` at 25%. 90% of the paths have at most a ratio less than or equal to 3. Below a 50% deployment, we need an increased number of tunnels to forward the traffic, and thus, we also increase the length of the paths. On a full SDN network, we only see the stretch due to powered off links.

Even though some paths reach a stretch ratio of 14 on `germany50` and 9 on `zib54`, we can see in Figure 8 that the network delay stays relatively low. Indeed, the paths with a big stretch are mostly one-hop paths that used to be on currently inactive links. To compute the delays, as the delay is proportional to the distance in an optical network [27], we use the distances given by the geographical coordinates in SNDlib for the `germany50` network. We got an average value of 1.8 ms per link. Since the coordinates are not given for the other topologies, we used the same average value for `atlanta`, `zib54`, and `ta2`. The median delay rarely goes above 10 ms for all four networks. The `zib54` network experiences the worst delay with a half SDN deployment, with almost 35ms of delay. *The bottom line is that SENAtoR allows us to stay below a delay of* 50 ms. This is important, as this value is often chosen in Service Level Agreements (SLAs) as the maximum allowed delay for a route in a network [28]. Thus, even if new routes computed by our algorithms may sometimes experience a high value of stretch, this will not be a problem for network operators.

V. Experimentations

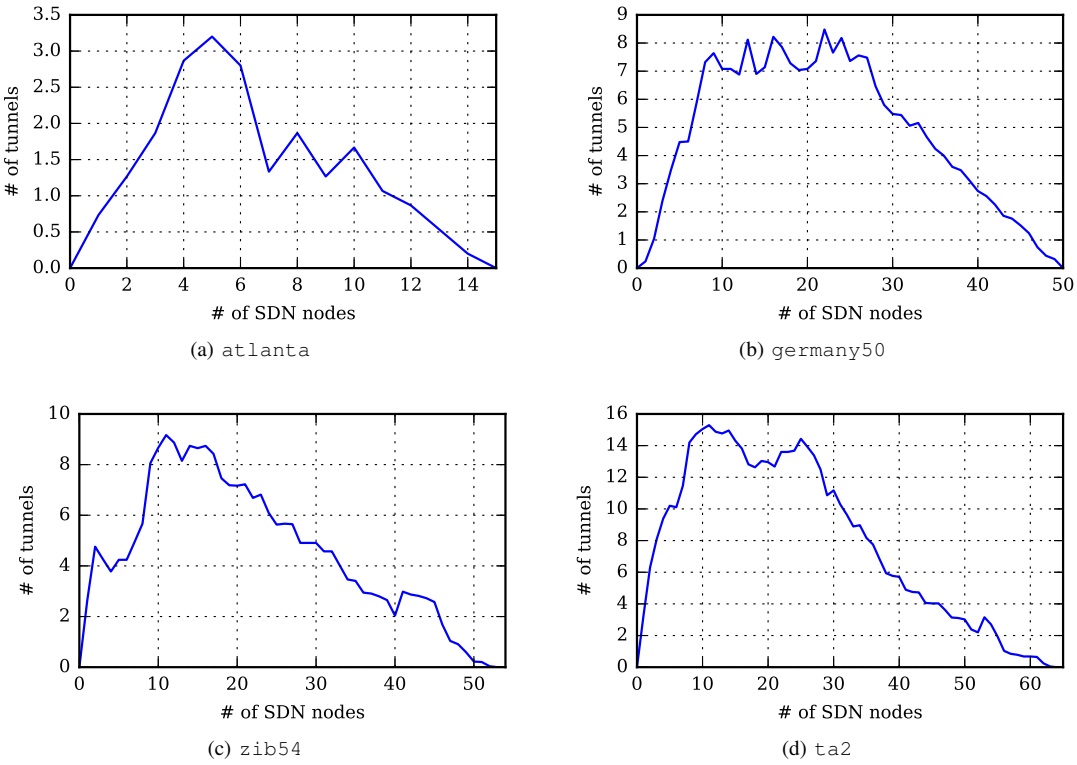
In this section, we present results obtained on a Mininet testbed with SENAtoR. Our objective is to demonstrate that SENAtoR can indeed turn off links and put SDN routers in power save mode without losing packets thanks to our smooth integration with OSPF to anticipate link shutdown.

A. Testbed

We built a hybrid SDN testbed using Mininet and a single remote centralized Floodlight controller. [5]

The Mininet network topology is based on `atlanta` with 50% SDN deployment. OSPF routers are materialized as host

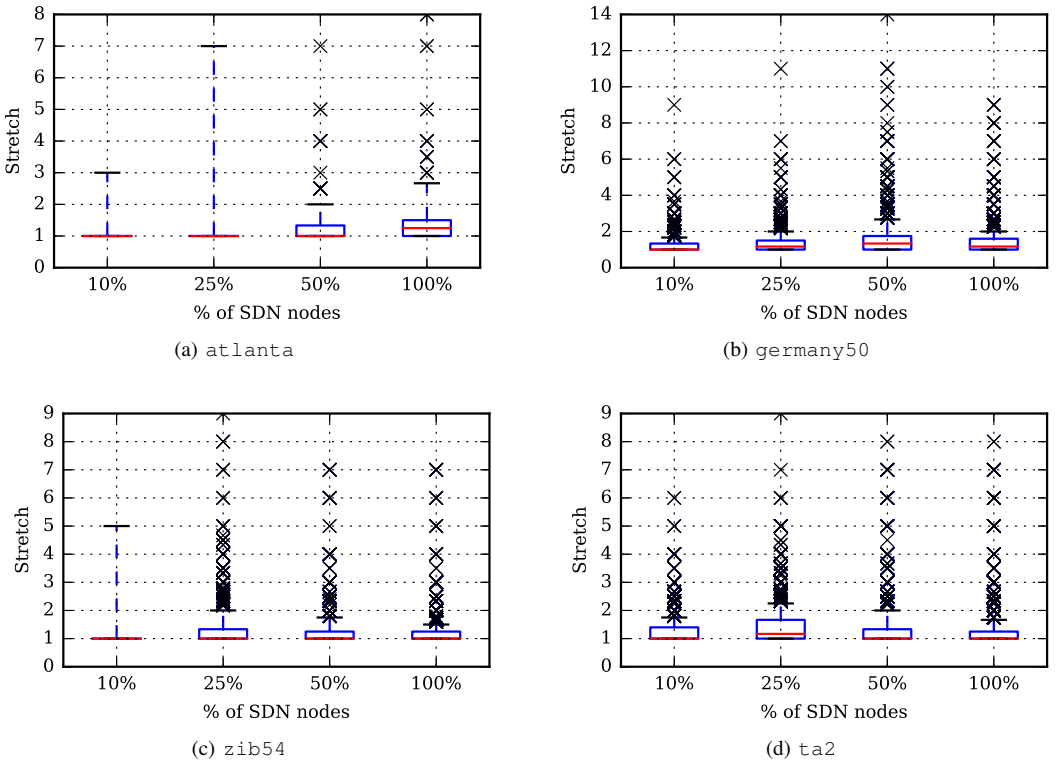
[5]Mininet [29] is a tool that can create a realistic virtual network, running real kernel, switch and application code, on a single machine (VM, cloud or native) and Floodlight [30] is the world's leading open source software-defined networking (SDN) community.



(a) atlanta (b) germany50

(c) zib54 (d) ta2

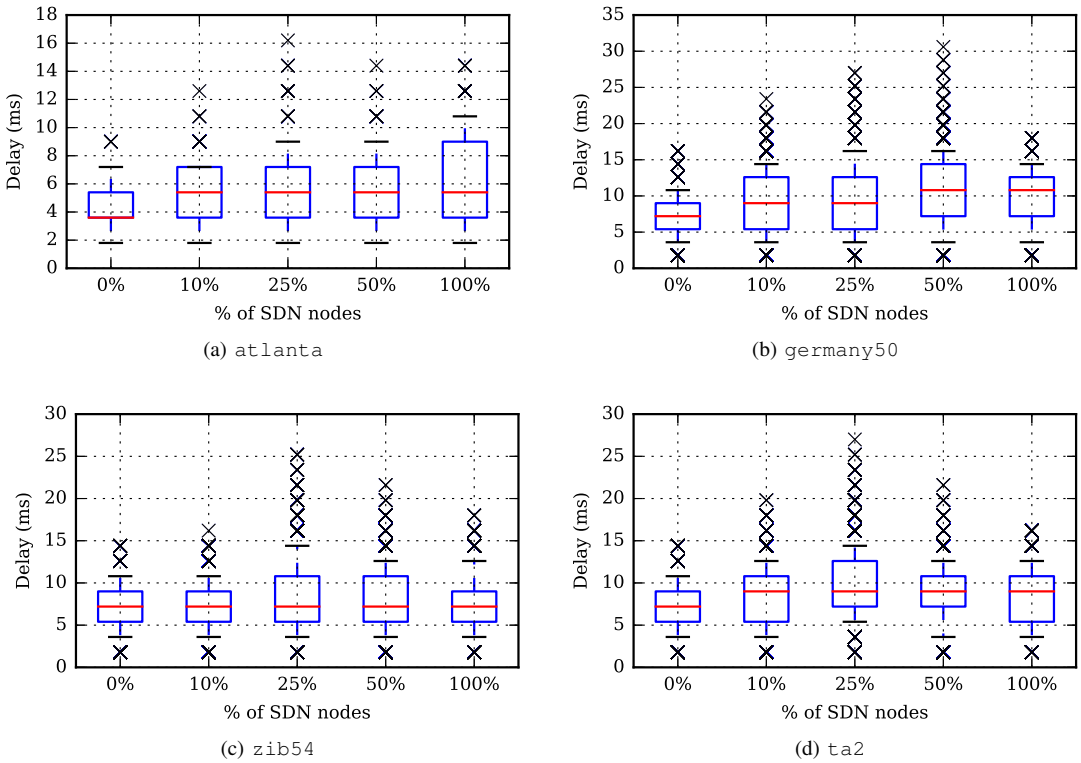
Fig. 6: Number of average tunnels installed per node on (a) atlanta, (b) germany50, (c) zib54 and (d) ta2.



(a) atlanta (b) germany50

(c) zib54 (d) ta2

Fig. 7: Stretch ratio for four different levels of SDN deployment on (a) atlanta, (b) germany50, (c) zib54 and (d) ta2. The box represents the first and third quartiles and whiskers the first and ninth deciles.



(a) atlanta (b) germany50

(c) zib54 (d) ta2

Fig. 8: Delays for the demands on (a) atlanta, (b) germany50, (c) zib54 and (d) ta2.

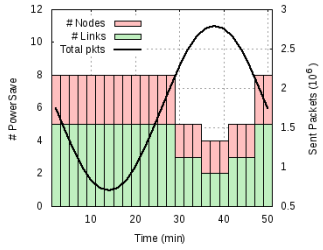


Fig. 9: Number of turned off links experiment on atlanta.

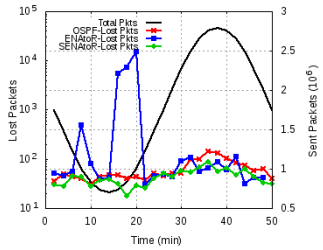


Fig. 10: Packet loss experiment on atlanta.

nodes in Mininet and run the Quagga software[6] while Open vSwitches (OvS)[32]) act as SDN routers. Our Floodlight controller can parse and answer to OSPF hello packets received and forwarded by the SDN OvS switches (through adequate OpenFlow rules installed in the SDN routers); hence ensuring the proper operation of the adjacent OSPF routers. Tunnels are implemented as simple GRE tunnels[7], 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 router to enter into sleep mode, we turn off all of its interfaces and disconnect it from the rest of the network. In power saving mode, the memory keeps the set of rules previously installed by the controller to perform a quick recovery back to normal active mode when requested by the controller.

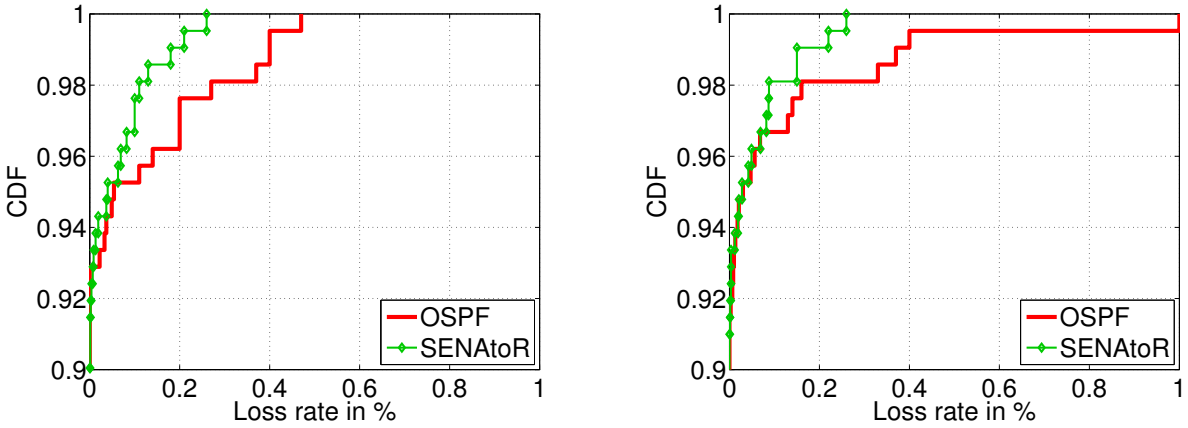
B. Lossless link turn-off.

In Figure 9 we vary the traffic over time to simulate smooth variations of the average rate. This is achieved by taking one traffic matrix and scaling it using the same sinusoidal function as in Figure 4. The bars in the figure correspond to the number of links that are turned off and the number of nodes that are put in powersave mode by SENAtoR.

The energy saving results in Figure 9 are in line with the ones of Section IV Figure 5, i.e., the same number of links and nodes turned off in all cases, which was expected as we use the same code at the controller during simulations and experiments. The added value of 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 10 portrays the time series of packet loss with a pure OSPF set-up (i.e. OSPF operates the complete network,

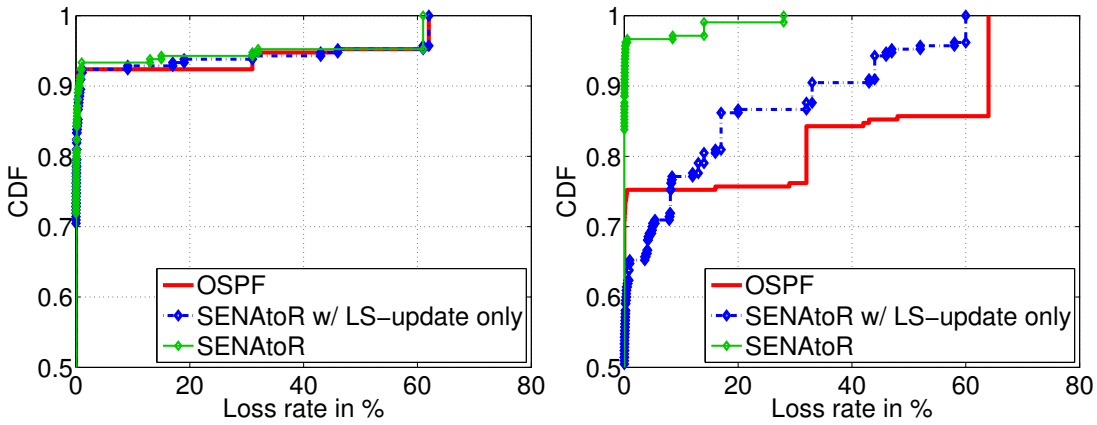
[6]Quagga [31] is a routing software suite, providing implementations of OSPFv2, OSPFv3 for Unix platforms.

[7]Generic Routing Encapsulation (GRE) tunnels are used to establish a direct, point-to-point connection between network nodes.



(a) In an SDN-OSPF link (b) In an OSPF-OSPF link

Fig. 11: Traffic spike experiment on `atlanta`.



(a) On an SDN-OSPF link (b) On an OSPF-OSPF link

Fig. 12: Link failure experiment on `atlanta`.

and no link is turned off in this case), SENAtoR and ENAtoR (SENAtoR without the smooth link shutdown functionality). The figure shows the importance of anticipating the link shutdown (resulting from putting SDN routers in sleep mode) as done by SENAtoR, as losses reach 10^4 packets when this feature is disabled (ENAtoR). The high loss rate of ENATOR is proportional to the amount of time 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 pure OSPF network without any links shutdown, with negligible loss rates ($10^{-4}\%$), even though it is using fewer links and nodes in the network.

C. Traffic spikes

To illustrate the traffic spike mitigation mechanism, we consider a fixed traffic matrix, and we induce a traffic spike either at an OSPF node directly connected to an SDN router (Figure 11a) or between OSPF nodes (Figure 11b). We report the cumulative distribution function (CDF) of loss rates of all connections. The spike detection algorithm of SENAtoR outperforms OSPF in terms of loss rate. Indeed, in an SDN-OSPF scenario, SENAtoR shows a maximum loss rate of around 0.25% while OSPF shows a maximum loss rate higher than 0.4%. In the OSPF-OSPF scenario, SENAtoR still shows a maximum loss rate of around 0.25%, while the loss rate experienced by OSPF reaches 1%.

One of the key reasons behind this observation is that regular OSPF nodes have no mechanisms to load balance packets in case of traffic spikes automatically.

D. Link failure

In this experiment, we again consider a fixed traffic matrix, and we induce a link failure either between an SDN router and an OSPF router or in between two OSPF routers and report the corresponding loss rates on Figures 12a and 12b.

We compare three cases: (i) the pure 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 the failure occurs on an OSPF-OSPF link). This is a key observation as we have an absence of

losses *even though some of the routers and links were down at the time of the failure*, and had to be switched on. This is because SDN routers 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 counterintuitive 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

Providing energy saving services in current networks is a challenging task as care must be taken to avoid traffic disruption and preserve failure tolerance capabilities even when the network operates with a reduced number of links/devices. In this paper, we presented SENAtoR, an energy-aware routing solution that preserves failure tolerance and traffic overload management of the network. SENAtoR is enriched with lossless link/node turn-off, spikes, and traffic failure detection services. SENAtoR's implementation and experimentation with emulated devices running full OSPF agents show that we can deal with unexpected network events correctly. More strikingly, our experiments show that even when our energy reduction mechanism is enabled and traffic spikes occur, SENAtoR features 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 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.

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

[23] 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.

[24] "Cisco 1941 series integrated services routers data sheet," August 2014. [Online]. Available: http://www.cisco.com/c/en/us/products/collateral/routers/1900-series-integrated-services-routers-isr/data_sheet_c78_556319.html

[25] 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.

[26] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, "SNDlib 1.0–Survivable Network Design Library," in *Proceedings of the 3rd International Network Optimization Conference (INOC 2007), Spa, Belgium*, April 2007, http://sndlib.zib.de, extended version accepted in Networks, 2009. [Online]. Available: http://www.zib.de/orlowski/Paper/OrlowskiPioroTomaszewskiWessaely2007-SNDlib-INOC.pdf.gz

[27] B.-Y. Choi, S. Moon, Z.-L. Zhang, K. Papagiannaki, and C. Diot, "Analysis of point-to-point packet delay in an operational network," *Computer networks*, vol. 51, no. 13, 2007.

[28] W. Fawaz, B. Daheb, O. Audouin, M. Du-Pond, and G. Pujolle, "Service level agreement and provisioning in optical networks," *IEEE Communications Magazine*, vol. 42, no. 1, 2004.

[29] "Mininet," http://mininet.org/.

[30] "Floodlight," http://www.projectfloodlight.org/.

[31] "Quagga," http://www.nongnu.org/quagga/.

[32] "Open vswitch," http://openvswitch.org/.