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

ISRN INRIA/RR--9020--FR+ENG

ISSN 0249-6399

RESEARCH REPORT
N° 9020
January 2017
Project-Team COATI

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[†]

Project-Team COATI

Research Report n° 9020 — January 2017 — 25 pages

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, strikingly, diminishing the packet loss rate compared to legacy protocols.

Key-words: Software Defined Network; energy saving; energy aware routing; hybrid network; packet loss

* This work has been partially supported by ANR program ANR-11-LABX-0031-01.
† Université Côte d'Azur, Inria, CNRS, I3S, France
‡ Université Côte d'Azur, CNRS, I3S, France

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Rapprocher le Routage Efficace en Énergie de la réalité avec les réseaux hybrides SDN

Résumé : Le Routage Efficace en Énergie vise à réduire la consommation énergétique des réseaux de FAI. L'idée est d'adapter le routage en fonction de la charge du traffic pour pouvoir éteindre certains équipements. Toutefois, cela implique d'effectuer des changements dynamiques dans les configurations de routage, ce qui est presque impossible avec les protocoles actuels. Le paradigme SDN (Software Defined Network) apporte la promesse de pouvoir optimiser dynamiquement un réseau grâce à un (ou plusieurs) contrôleur(s) centralisé(s).

Dans ce travail, nous proposons SENAtoR, un algorithme rendant possible le Routage Efficace en Énergie pour un scénario de migration progressive de matériel actuel vers du matériel SDN. Puisqu'éteindre des équipements du réseau est une tâche délicate qui peut entrainer des pertes de paquets, SENAtoR fournit aussi plusieurs méchanismes les éviter : l'utilisation de tunnels pour rerouter rapidement, une désactivation en douceur des noeuds ainsi que la détection de pics de trafic et de défaillances de liens. Nous validons notre solution avec des simulations poussées ainsi que par expérimentation. Nous montrons que SENAtoR peut être progressivement déployé dans un réseau utilisant le paradigme SDN. Il permet de réduire la consommation d'énergie des réseaux de FAI de 5 à 35% selon le taux de pénétration du matériel SDN, tout en réduisant le taux de perte de paquets comparé aux protocoles actuels.

Mots-clés : Software Defined Network; économie d'énergie; réseau hybride; tunnel; perte de paquet

1 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 networks to SDN networks [1]. One of the more realistic one 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 by putting them in power save mode which also 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]. In these cases, the energy efficient solution should be able to react very quickly and switch on previously turned off devices. In this article, we use the terms turn off and sleep mode interchangeably. We thus propose three mechanisms (detailed below): First, we use pre-set *tunnels* as backup routes in case of link failure or turned off devices. Second, we use the SDN controller to suppress any incoming OSPF packet to simulate a link disconnection on the network interface to disable. This forces OSPF nodes to converge to different Shortest Path Trees (SPT). Last, we use SDN monitoring capabilities to detect quickly large unexpected traffic peaks or link failures.

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 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.

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 (*i*) while OSPF has not converged yet, packets can be rerouted through the pre-set tunnels; and (*ii*) since the link and node is still on, packets are not lost during the routing transition.

Detecting Failures or Flash Crowds with SDN. Network capacity overprovisioning 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. In this case, we turn back on all nodes in the network to avoid packet losses.

Our contributions are the following:

- We propose several mechanisms to bring energy aware solutions closer to reality in ISP networks.

- We model the problem of energy aware routing in a hybrid SDN network. We formulate an ILP deciding which network equipements to put into sleep mode, and at the same time, which tunnels to set to reroute the traffic. The formulation presents several difficulties. First, legacy nodes have to route flows through shortest paths following legacy protocols, when 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 equipements are put into sleep mode. Moreover, the set of tunnels, which are used, depends on the level of traffic. The ILP provides optimal solutions for small networks. For larger networks, we propose heuristic algorithms.

- We propose several mechanisms to avoid packet losses when putting network devices into sleep mode: tunneling, smooth shutdown of links, and detection of traffic variations.

- 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. It allows us, first, to use the power model of a real high-end SDN-capable dedicated router - HP5412zl SDN switch, and, second, to exhibit that it is *possible to implement energy saving solutions while reducing packet losses* compared to legacy protocols.

2 Related work

Energy aware routing. Energy aware routing has been studied for several years, see for example [4, 5] for backbone networks, [6, 7] for data center networks, or [8] 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 [9], 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 [10] 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 [11]. 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 [12]. Once the link failure has been detected, OpenFlow already offers a link failure mitigation 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 [13]. 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,

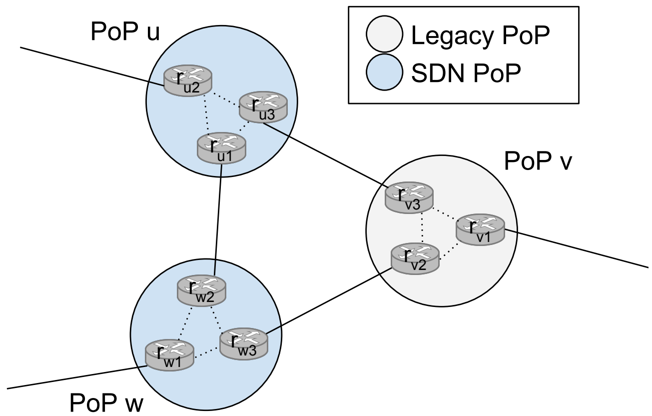


Figure 1: 3 PoPs interconnected in an hybrid network.

which form backup paths. Later, SDN nodes reroute traffic through non damaged paths. We borrow this idea and propose to use pre-set tunnels, which are used 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, 14]. However, abrupt variations happen in case of link failures or flash crowd [15]. Methods have been proposed to detect them in legacy networks, see for example [16, 17]. Netfuse [18] has been proposed in SDN-based data centers to mitigate the effect of traffic variations. In this paper, we propose a method to detect such abrupt variations in a hybrid SDN network.

3 Energy Aware Routing for Hybrid Networks

3.1 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 [19]. Each link $(u, v) \in A$ is connected to a specific router in PoP u and in PoP v, see Figure 1. A link (u, v) has a maximum capacity C_{uv}. We consider hybrid networks in which SDN capable equipments are deployed alongside legacy routers. This kind of networks can be found in the transition from legacy networks to a complete Software Defined Network. We consider a scenario in which PoPs do not contain heterogeneous equipments, i.e, all routers are either SDN capable (in this case, we use the term SDN switch) or legacy. 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.

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, depending of the value of the parameters, to express the different power models (between ON-OFF and energy proportional) found in the literature, see [20] for a discussion. 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 (this is discussed in more details in Section 5). 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. For example, in Figure 1, shutting down the link between PoP A and PoP B will set router R_{A3} to sleep mode, as it is an SDN switch, but R_{B3} will remain active. Shutting down the link between PoP A and PoP C will put R_{A1} and R_{C2} to sleep.

When a 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 (this is discussed in details in Section 5.2). 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 legacy router can automatically enter into sleep, using for instance IEEE 802.3az Energy-Efficient Ethernet [21].

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. We now consider the following problem.

Hybrid Energy Aware Routing (hEAR) with tunnels Problem. *We consider an SDN budget k, i.e., a number of PoPs which can be transitioned to SDN equipments. The hEAR problem is*
- to deploy k SDN PoPs in the network,
- to route a set of demands $\mathcal{D}$, and
- to choose a set of tunnels,
while
- minimizing the total power consumption of the network,
- respecting the link capacities, and
- ensuring that the traffic can be rerouted quickly through tunnels when network equipments are turned off.

With most legacy network mechanisms, tunnels cannot be deployed dynamically during the operation of the network. They have thus to be pre-set statically. To determine which tunnels have to be set, we carry out a preliminary step of tunnel selection. We thus consider in fact two variants of the problems:(i) **hEAR-with-tunnel-selection**, (ii) **hEAR-with-tunnel-preset**.

For the preliminary step of hEAR-with-tunnel-selection, we consider a *minimum baseline level for the traffic*. As a matter of fact, with low traffic, a large number of links may be turned off, leading to the usage of a large number of tunnels. We then find the best set of tunnels for this configuration of traffic. During the operation of the algorithm (hEAR-with-tunnel-preset) and for larger levels of traffic, fewer links are turned off and thus fewer tunnels are used. We may thus only use the tunnels pre-set with the minimum configuration.

In the following, we present an ILP formulation of the problem to find the best placement of SDN nodes and energy aware routing for different levels of traffic. The ILP formulation runs for small networks. For larger networks, we present an efficient heuristic algorithm.

3.2 Integer Linear Program

We propose the following Integer Linear Program to solve the hEAR-with-tunnel-selection problem (in fact, it also solves the simpler hEAR-with-tunnel-preset problem by fixing the variables corresponding to pre-set tunnels to 1). A summary of the notations is found in Table 1. The formulation presents several difficulties. First, legacy nodes have to route flows through shortest paths following legacy protocols, when 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 equipements 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) limits the usage of a backup tunnel. Constraints (8) and (**??**) determine if the next

Table 1: Notation 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

hop of a given demand is active or not. Combining constraints (10) and (11), we ensure that at most one next hop can be selected for a source-destination pair on a router and if it is not an SDN node, the next hop can only be the OPSF next hop. A link can only be shutdown if one of its end points is a SDN node (12) and links between two routers share the same state (13). Finally, a router can only be put to sleep if its link to the other PoP can be shutdown and if it is an SDN node (Constraints (14) & (15))

The ILP can be used to find good solutions for small sized instances, see Section 4. 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 an efficient heuristic algorithm in the following.

3.3 Heuristic Algorithm (SENAtoR)

We propose here SENAtoR (Smooth ENergy Aware Routing) which can be used to solve both variants of the hEAR problem. The problem can be naturally divided into three subproblems: (i) first, *SDN node placement*, to define the subset of SDN nodes in the network (ii) then, *path assignment*, to find a path for every demand in $\mathcal{D}$, (iii) and last, *off link selection*, to select the links we power off and reroute the affected traffic.

3.3.1 SDN node placement

The SDN node placement is multi-criteria. Indeed, SDN nodes have several functionalities. First, they allow a better control of the routing as the next hop of a flow passing through an SDN node can be chosen dynamically by the controller. Thus, it is important to select SDN nodes which are central and

$$\min \sum_{(u,v)\in A} U_{uv}x_{uv} + L_{uv}\mathcal{F}_{uv} + A_u r_{uv} + B_u \tag{1}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

route a large amount of traffic. A way to do this is for example to choose nodes according to their centrality, e.g., *betweenness centrality* or *closeness centrality*. Second, recall that only links adjacent to an SDN node can be turned off. Thus, to be able to reduce efficiently the network energy consumption, we want to cover the maximum number of links with the available budget of k SDN nodes. If we consider this second criterion independently, we can optimize it by solving a MAX k-VERTEX COVER. This problem is known to be NP-hard. However, it is simple enough and can be solved optimally for the topology sizes considered, using for example a simple ILP and CPLEX.

We tested different methods to select the SDN nodes and chose a *simple criterion to express the importance of a node (centrality and covering): the node degree.* The resulting heuristics is: first sort all nodes according to their degree; second, choose the k first nodes. This method gives similar results to the other ones and has the advantages of being simple and to allow a good incremental upgrade to SDN hardware (on the contrary to solving MAX k-VERTEX COVER).

3.3.2 Path Assignment

To assign a path to a demand, we build a weighted residual graph $H_{st} = (V, A')$ and then search for the shortest path between s and t in H_{st}. We build H_{st} the following way.

Nodes in H_{st} are the ones of D and correspond to network routers. For links, we only consider links and tunnels which (i) have enough residual capacities to satisfy the demand D_{st} (ii) can be used by a feasible routing of the demand between s and t. For Condition (ii), we consider each node u and construct its set of out-neighbors 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. We have several cases. If a tunnel is already defined for the destination t, the end of the tunnel is the only neighbor of u. If no tunnel is defined, the next step depends on the variant of the problem. For hEAR-with-tunnel-preset, tunnels are already selected. Thus, u has no neighbor in H_{st}. In the hEAR-with-tunnel-selection variant, we have to set a tunnel in this case. 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. The decision of which tunnel will be really selected is done later when we compute the shortest path in the residual graph.

If u is an SDN node, the routing is done by OpenFlow rules installed by the controller. We have two cases: 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. Otherwise, we only add as neighbor of u in H_{st} the node designed as the next hop by OpenFlow. As for a legacy node, if the link to the next hop given by OpenFlow is inactive, we also consider tunnels in the same way. The decision of installing or not a new OpenFlow rule in u is done later by the algorithm when the shortest path in the residual graph is computed. The same applies for the selected tunnel.

When the residual graph H_{st} is built, we select the shortest path from s to t to route the demand. If this path uses new tunnels or new OpenFlow rules, we add them to the current solution.

3.3.3 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 heuristics stops when all SDN links are either powered off or *non-removable*.

4 Numerical evaluation

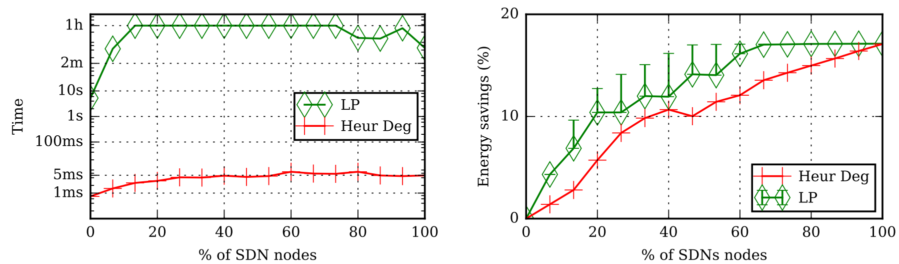
In this section, we evaluate the solutions proposed on different ISP topologies. We first compare the performances of the ILP and of the heuristic algorithm on a small topology. We then solve the hEAR problem on larger networks of SNDLib. We show that *energy savings up to 35% can be obtained for different levels of SDN hardware installation.*

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 [23]. In the first case, we use reading for the HP switch and values from Cisco WAN switch [24]. The switch uses 95W when in sleep mode and 150W if it is active ($B_u = 95, A_u = 55$). According to Cisco specifications, links are using 30W as a baseline and go up to 40W when at full capacity ($U_{uv} = 30, L_{uv} = 10$). In the second case, we consider an *ideal energy efficient* switch. In order to have a fast recovery from sleep mode, the TCAM must be kept under power to preserve the forwarding rule. According to [25], TCAM represent 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.

4.1 ILP vs. Heuristic

We use the `atlanta` network (composed of 15 nodes and 22 links) to compare the ILP and the heuristics presented in the last section. We consider the traffic matrices provided by SNDLib and we compute the energy savings for different number of SDN nodes. We solve the ILP with CPLEX. As the ILP is complex, we set a time limit of one hour. The results presented correspond to the best solution found by the solver within the time limit.

Computation Time. In Figure 2(a), we show the computation time of the two solutions. We see that the ILPs are complex as the limit of 1 hour of



(a) Time

(b) Energy savings

Figure 2: Comparison between the ILP and the heuristics on the atlanta network (15 nodes and 22 links).

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. The settings thus are easier to solve. On the contrary, the heuristics is always very fast. It takes at most 5 ms to find a solution in all settings.

Energy Savings. We see in Figure 2(b) the possible energy savings in the network. 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 heuristics 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 heuristics 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.

4.2 Simulations on larger networks

We further look at the performance of the heuristics on `atlanta` and on larger networks such as `germany50` (50 nodes and 88 links), `zib54` (54 nodes and 81 links) and `ta2` (65 nodes and 108 links).

Traffic Model In this work, we assume that an ISP is able to estimate the traffic matrix of its network using (sampled) netflow measurements [26] or, in the case of hybrid networks, by combining SDN and OSPF-TE data [11]. Estimation errors can be handled by our traffic variation detection algorithms (Section 5.2). Since ISP traffic is roughly stable over time with clear daily patterns, a few traffic matrices would be enough to cover a whole day period. Consequently, a

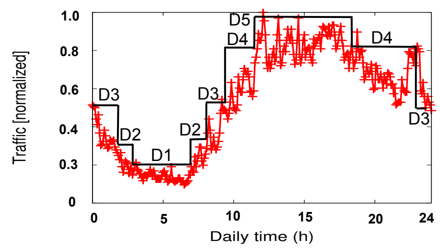


Figure 3: Daily traffic in multi-period

relatively small number of routing reconfigurations allows operators to obtain most of the energy savings [20] and avoid making frequent reconfigurations.

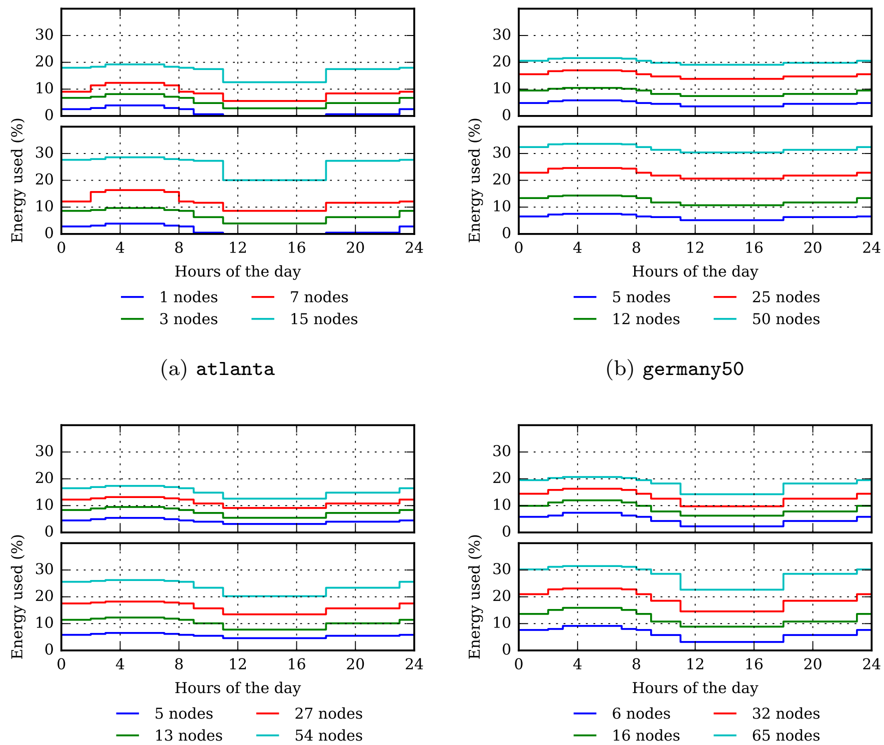
Indeed, as exemplified by the daily variations for a typical link in the Orange ISP network, see Figure 3, five traffic matrices (labeled D1 – D5) are enough to represent the daily variations. Inspired from this observation, we select these 5 different traffic matrices as baseline for our simulations and experiments.

We next compute the best hybrid energy aware routing for them. We then switch between hEAR routings when we detect that the amount of traffic has changed. We discuss and test how to do that in practice without packet losses in Section 5.

Daily savings In Figure 4, 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 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 2% and 7% (3.5 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 networks. With *ideal* switches, we obtain higher savings, between 25% and 35%.

Number of tunnels We look at the number of tunnels used in Figure 5. 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 to the turned off link. Thus, less 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



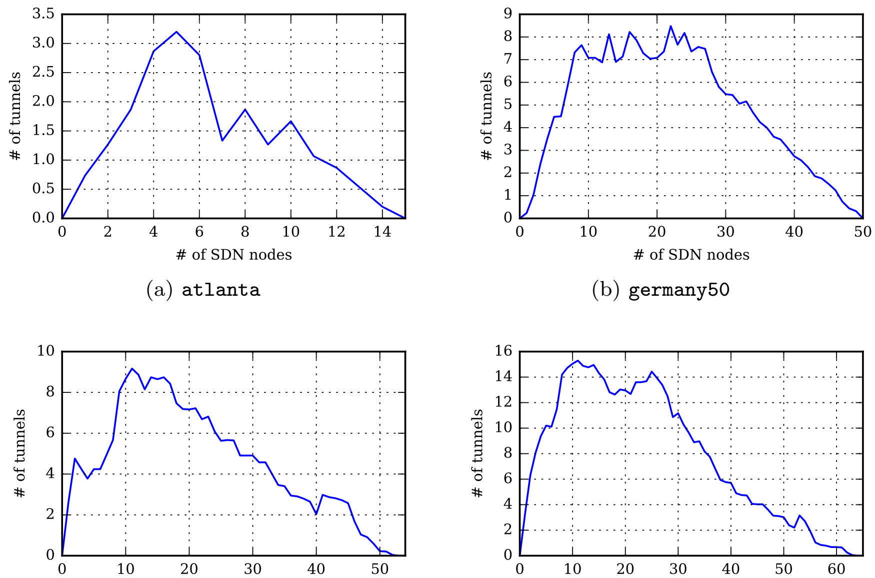
(a) `atlanta` (b) `germany50` (c) `zib54` (d) `ta2`

Figure 4: Daily energy savings over the day for the (a) `atlanta`, (b) `germany50`, (c) `zib54` and (d) `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.

`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.

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. In Figure 6, 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,

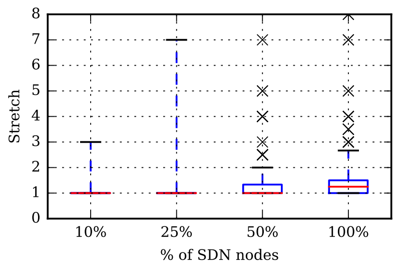


(a) `atlanta` (b) `germany50` (c) `zib54` (d) `ta2`

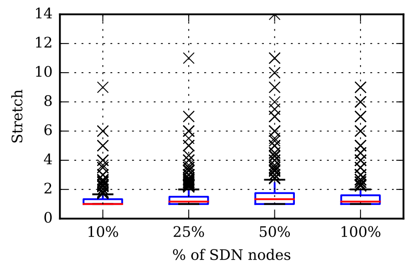
Figure 5: Number of average tunnels installed per node on the (a) `atlanta`, (b) `germany50`, (c) `zib54` and (d) `ta2` networks. as a function of the percentage of SDN nodes.

1.25 for `germany50` at 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 3. The stretch of the paths follows the same behavior as the number of tunnels needed for a valid hEAR. 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.

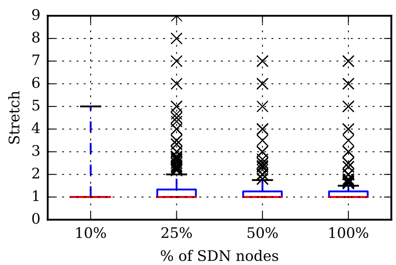
Even though some paths reach a stretch ratio of 14 on `germany50` and 9 on `zib54`, we can see in Figure 7 that the delay on the network stays relatively low. Indeed, the paths with a big stretch are mostly one-hop paths that used to be on currently inactive links. To compute the delays, as the delay 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 two 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 using SENAtoR, we stay below a delay of 50ms.* This is important, as this value is often chosen by Service Level Agreements (SLAs) as the maximum allowed delay for a route in a network [28]



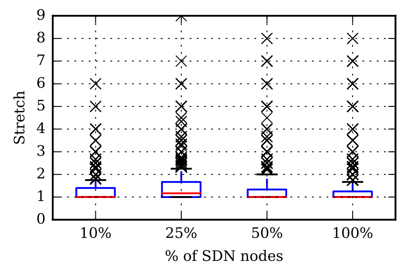
(a) `atlanta`



(b) `germany50`



(c) `zib54`



(d) `ta2`

Figure 6: Stretch ratio for four different levels of SDN deployment on (a) `atlanta`, (b) `germany50`, (c) `zib54` and (d) `ta2` networks. The box represents the first and third quartiles and the whiskers the first and ninth deciles.

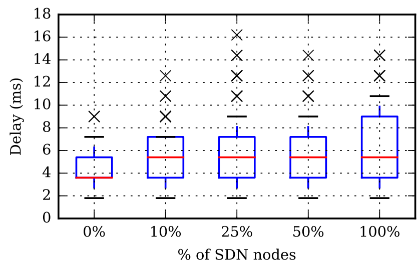
Thus, even if new routes computed by our algorithms may experience sometimes a high value of stretch, this will not be a problem for network operators.

5 Experimentations

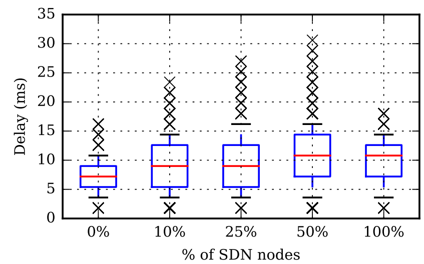
In this section, we present results obtained on a Mininet testbed with the SENAtoR solution. Our objective in this section is twofold. First, we aim at demonstrating that SENAtoR can indeed turned off links and put SDN switches in power save mode without loosing packets thanks to a smooth integration with OSPF to anticipate link shutdown. Second, we introduce and evaluate a link failure and a traffic spike detection algorithm that enables SENAtoR to cope with those small time scale events, as compared to energy saving, which is performed at a larger time scale.

5.1 Testbed

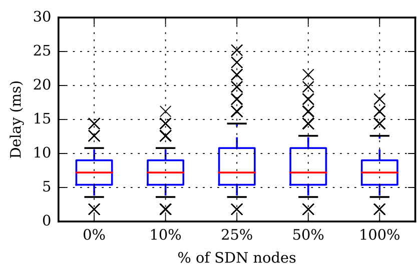
We have built a hybrid SDN testbed using Mininet [29] and a Floodlight [30] controller. OSPF routers are materialized as host nodes in Mininet and run the



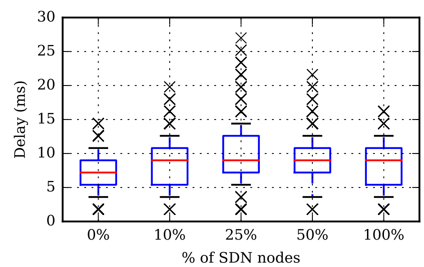
(a) `atlanta`



(b) `germany50`



(c) `zib54`



(d) `ta2`

Figure 7: Delays for the demands in the (a) `atlanta`, (b) `germany50`, (c) `zib54` and (d) `ta2` networks.

Quagga software [31] while Open vSwitches (OvS) [32]) 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. The very same code implementing the heuristics proposed and evaluated in Section 3 and 4 respectively, is used by the Floodlight controller. 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 to put into sleep mode an SDN PoP switch, we turn of all of its interfaces and disconnect it from the rest of the network. We believe this is a fair simulation of powersaving mode, as our tests effected on our HP5412zl switch depicted that putting an SDN switch in powersave mode (inactive SDN switch) is equivalent to shutting down all of the network interface modules and shutting down all of background modules, that are not used anymore, and to decrease the energy consumption of the fan, while keeping the set of rules previously installed by the controller. *Keeping the previously installed rules in memory enables a quick recovery from powersave mode to normal active mode.*

We consider in this section the Atlanta topology (15 PoP nodes, 22 links between PoPs) of SNDlib with 50% SDN deployment. As stated in Section 3, a PoP is composed of several routers, each router connects the PoP X to another PoP Y, and the routers in a PoP are connected together using a central node such that they form a star topology.

For a given traffic matrix, each source-destination pair corresponds to one constant bit rate (CBR) UDP connection in our experiments.

5.2 OSPF-SDN interaction and traffic spikes/link failures

We have implemented three key features that complement the SENAtoR algorithm of Section 3.

Lossless link turn-off. Before putting an SDN PoP switch in powersave mode which turns off its interfaces, Floodlight stops sending any OSPF packet to its neighbors. This allows neighboring OSPF routers to converge to a network view excluding this node. Indeed, after the default *dead interval* of $3 \times$ *hello interval* without receiving any Hello packet, an OSPF router declares its neighbor as dead and stops using the link. However, until the end of the dead interval, the link is considered to be active and traffic flows over this link. 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. To avoid any losses when a node should be put in powersave mode, SENAtoR thus stops sending Hello packets during the expected convergence time, before actually putting the node in powersave mode.

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

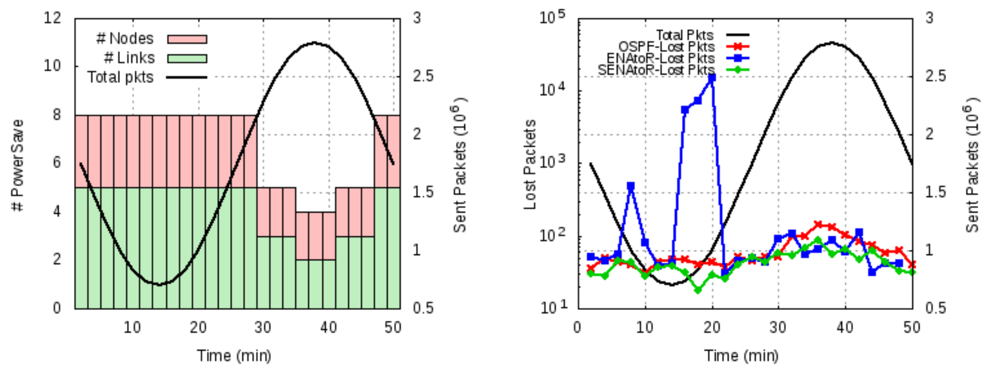
SENAtoR with a safeguard mechanism that aims at reactivating inactive SDN PoP switches in case of a sudden traffic spike. The latter event is defined on a per link basis as follows: the controller is collecting the traffic load on each interface of every SDN active switch at a small time scale (in our 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. There is thus little need taking actions unless the rate fluctuation is severe. After the OSPF expected convergence period, the controller reruns the SENAtoR solution to obtain a new green architecture if possible.

Link failure mitigation. We employ a mechanism similar to the traffic spike mitigation mechanism in case of link failures. There are two cases: either the link that fails is connected to an SDN active router or it is not. In the first case, the SDN switch takes an immediate action: it undoes any previous action taken by the SENAtoR solution, i.e., it turns on again any inactive SDN node, and packets are rerouted through a different path if possible (including the pre-set tunnels). In the second case – a link in between two OSPF routers fails – we try to mitigate this event by turning on previously inactive SDN nodes. 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 the SENAtoR solution to obtain a new green architecture if possible.

5.3 Lossless link turn-off.

In Figure 8a we vary the traffic over time (continuous black curve), so that we have a factor of 6 in between the minimal and the maximal total traffic in the network. This is achieved by taking one traffic matrix and scaling it using a sinusoidal function to impose smooth variations on the average rate, similarly to what is expected in a typical ISP network. 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 the SENAtoR algorithm.

The energy saving results in Figure 8a are in line with the ones of Section 4 Figure 4(a), i.e., same number of links and nodes turned in all cases, which is not surprising as we use the same code at the controller. 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 8b portrays the time series of packet loss with pure OSPF (OSPF operates



(a) Number of turned off links

(b) Packet loss

Figure 8: `atlanta` topology

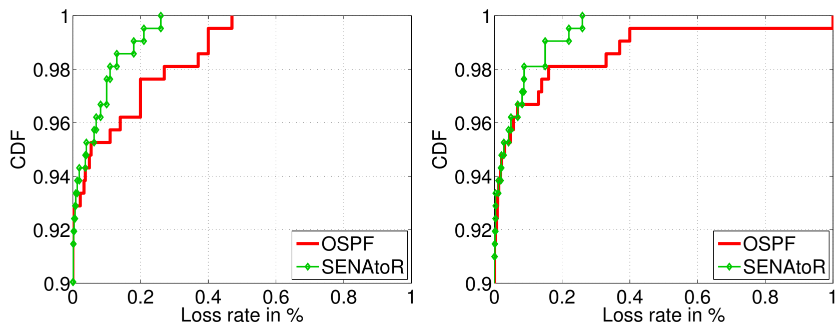
the complete network and no link is turned off in this case), SENAtoR and ENAtoR (SENAtoR without the smooth link shutdown). The figure shows the importance of anticipating the link shutdown (resulting from putting SDN switch in sleep mode) as is done in SENAtoR as losses explode to 10^4 packets when 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.

5.4 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 9a) or between OSPF nodes (Figure 9b). We report the cumulative distribution function (CDF) of loss rates of all connections (source-destination pairs). Clearly, the spike detection algorithm of SENAtoR allows it to outperform OSPF, even though it is using less active links and nodes. 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.

5.5 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. We present in Figures 10a and 10b the loss rates for the former and latter case respectively. We compare here three protocols: (i) the legacy OSPF



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

Figure 9: Traffic spike experiment with the `atlanta` topology

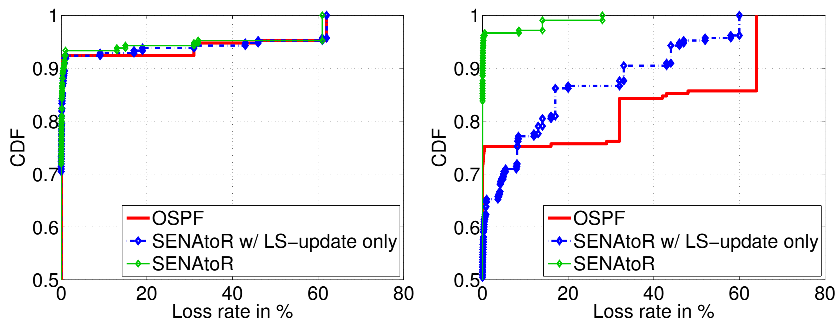
scenario, in which the link failure is handled by the OSPF protocol (with a long convergence time), (*ii*) the SENAtoR solution using OSPF Link State (LS) Updates only to detect network changes; and (*iii*) the SENAtoR solution with its *Link failure* detection and mitigation mechanism.

We first observe that even without link failure mitigation, *SENAtoR does not experience higher loss rates than the legacy OSPF protocol* (and significantly lower loss rates in the OSPF-OSPF case), *even though some of the switches and links were down at the time of the failure*, and had to be switched on. The explanation is that 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 further observe a counter intuitive result, which is that the loss rates using SENAtoR are smaller when the failure occurs on an OSPF-OSPF link rather than 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, which it can do by instructing upstream SDN nodes to reroute their traffic. So, while OSPF convergence time is slow, this is mitigated by the fact that less traffic is sent over the lossy link as soon as SENAtoR detects the failure.

6 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 devices. In this article, we presented an energy aware routing solution to improve the energy efficiency of ISP networks, by means of an algorithm with small computational



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

Figure 10: Link failure experiment with the `atlanta` topology

times for a network with several tens of nodes and links.

As our numerical results show, in an all-to-all traffic matrix, an incremental deployment of SDN can provide between 5% to 35% of energy savings.

To preserve failure tolerance and traffic overload management of the network, SENAtoR was enriched with lossless link/node turn-off, spikes, and traffic failure detection services. Our 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.

[4] L. Chiaraviglio, M. Mellia, and F. Neri, "Energy-aware backbone networks: a case study," in *IEEE ICC*, 2009.

[5] E. Le Rouzic, E. Bonetto, L. Chiaraviglio, F. Giroire, F. Idzikowski, F. Jiménez, C. Lange, J. Montalvo, F. Musumeci, I. Tahiri *et al.*, "Trend towards more energy-efficient optical networks," in *Optical Network Design and Modeling (ONDM), 2013 17th International Conference on*. IEEE, 2013, pp. 211–216.

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

[7] J. Araujo, F. Giroire, J. Moulierac, Y. Liu, and R. Modrzejewski, "Energy efficient content distribution," *The Computer Journal*, p. bxv095, 2015.

[8] 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.

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

[10] 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.

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

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

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

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

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

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

[17] 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.

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

[19] 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.

[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."

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

[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] "Opendaylight," https://www.opendaylight.org/.

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

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

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399