

# Green Multi-Stage Upgrade for Bundled-Link SDNs with Budget Constraint

Lely Hiryanto<sup>1</sup>, Sieteng Soh<sup>2</sup>, Kwan-Wu Chin<sup>3</sup> and Mihai Lazarescu<sup>4</sup>

**Abstract**—Upgrading a legacy network into a Software Defined Network (SDN) in stages, and minimizing the energy consumption of a network are now of great interests to operators. To this end, this paper addresses a novel problem: minimize the energy consumption of a network by upgrading switches over multiple stages subject to the available monetary budget at each stage. Our problem considers (i) bundled links that can be powered-off individually only if they are adjacent to a SDN switch, and (ii) decreasing upgrade cost and increasing traffic demands over multiple stages. We formulate the problem as an Integer Linear Program (ILP) and propose a greedy heuristic called Green Multi-stage Switch Upgrade (GMSU). Experiment results show that increasing budget as well as number of stages affect the total energy saved and number of upgraded switches. GMSU produces results that are up to 5.8% off the optimal result. Moreover, on large networks, in which ILP becomes computationally intractable, GMSU uses less than 0.1 second to compute a solution.

**Index Terms**—Hybrid SDNs, Multi-stage Upgrade, Bundled Links, Energy Saving.

## I. INTRODUCTION

Network operators are beginning to adopt a new paradigm: Software Defined Networks (SDNs) [1], [2]. A SDN decouples the *forwarding* and *controlling* role of switches. In particular, each SDN-switch (*s*-switch) is only tasked with packet *forwarding*; this is in contrast to a *conventional/legacy* switch (called *l*-switch) that must also construct a routing table. A SDN also uses *centralized* controllers [3] to: (i) compute the best route or rule for flows, (ii) store rules in the flow table of *s*-switches, and (iii) periodically requests all *s*-switches to send connectivity status [4]. As a result, SDNs allow operators to better manage the resources of their network.

When deploying SDNs, network operators face two main *concerns*. The first is the significant capital investment required to upgrade their *legacy* network into a SDN, and also to retrain personnel to manage the upgraded network. The second concern is that a SDN is an in-progress technology, meaning there is less confidence from users as compared to legacy networks that use proven and mature technologies, especially in terms of their scalability, reliability, robustness and security [5]. As a consequence of these concerns, network operators are more likely to upgrade their *legacy* networks over one or more time periods or stages that may span several years [6].

Author Hiryanto, Soh and Lazarescu are with Computing Discipline of the School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University, Bentley, WA, 6102, Australia (email: lely.hiryanto@postgrad.curtin.edu.au<sup>1</sup>; s.soh@curtin.edu.au<sup>2</sup>; m.lazarescu@curtin.edu.au<sup>4</sup>). Author Chin is with the School of Electrical, Computer, and Telecommunications Engineering, University of Wollongong, Australia (email: kwanwu@uow.edu.au<sup>3</sup>).

Another *concern* is energy efficiency. In particular, the energy consumed by routers/switches has a direct impact on the operating expenditure of a network and the amount of emitted greenhouse gasses [7]. This is further exacerbated by the fact that network operators often over-provision network resources in order to satisfy traffic demands [8]. However, many studies have shown that the average link utilization is only about 30-40% [9]. As a result, researchers have proposed a number of solutions to reduce the energy consumed by a network [7], [10]. One example is IEEE 802.3az [11], where network interfaces are able to adjust their duty-cycle by controlling their active or sleep time according to traffic patterns. Another solution is to use bundled-links or IEEE 802.1AX and IEEE 802.3az, where the capacity of a link can be increased by adding new cables [9], and any unused cables can be placed into *sleep* mode to save energy [12].

This paper addresses a problem that considers the aforementioned concerns. That is, we consider the problem of upgrading a *legacy* network with bundled links into a SDN over multi stages. The goal is to select a set of switches at each stage to maximize the number of powered-off cables to reduce energy consumption subject to (i) the maximum budget at each stage, (ii) depreciation in switch cost, (iii) growing traffic sizes, and (iv) that a cable can only be powered-off if it is connected to at least one *s*-switch. To illustrate our problem, consider the topology and flows depicted in Fig. 1. Assume the total budget is 40 (in \$) and the upgrade is performed over two stages; the budget for each stage is initially set to 20. All switches have equal initial upgrade cost of 15 (in \$) that decreases by 20% in the second stage; the cost in stage-2 reduces to 12. In stage-1, we can upgrade only one switch. The remaining budget,  $20 - 15 = 5$ , is added to the budget of stage-2, which is now equal to  $20 + 5 = 25$ . With that budget, we can upgrade two more switches. First consider the solution depicted by Fig. 1a, where we first upgrade switch-5 in stage-1 followed by switch 3 and 4 in stage-2. The figure shows that only link (2, 5) can be turned off, as indicated by a dotted line, resulting in an energy saving of  $1/12 \times 100\% = 8.33\%$ . Fig. 1b shows an alternative solution. Specifically, we upgrade switch-2 in stage-1 followed by switch 6 and 8 in stage-2. In this case, the SDN controller can switch-off a total of five links. This results in an energy saving of 41.67%. Let us now assume that each link consists of two cables and each unused cable can be powered-off individually. If each link in Fig. 1b requires one cable to carry the flow as shown by the solid lines, then four additional cables can be switched off, in addition to the 10 cables from the five links, see the dotted lines, which have

no flow. Thus, there are  $5 \times 2 + 4 = 14$  powered-off cables, which is equivalent to an energy saving of 58.33%.

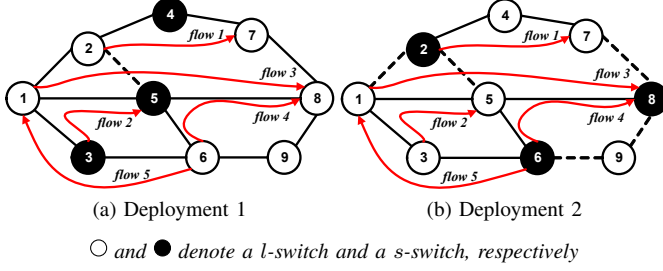


Fig. 1. Two examples of SDN deployment. Dotted lines indicate links that can be switched off.

Our main contributions are as follows. Firstly, we propose a *novel* optimization problem that aims to minimize energy usage when deploying a SDN over  $T \geq 1$  stages subject to a maximum budget at each stage. Our problem is significant as it will be of interest to network operators that (i) have limited budget, and (ii) prefer to upgrade their networks over multiple stages due to minimize the risks that arise when adopting a new technology such as SDNs. Further, our problem considers the depreciation of *s*-switches [6], where one would expect spending a smaller budget when the upgrade is performed over longer periods as long as the network has resources, e.g., bandwidth, to meet traffic demands. Secondly, we consider a SDN with bundled-links. To the best of our knowledge, we are the first to consider switching off cables in a SDN. Thirdly, we formulate our problem as an Integer Linear Programming (ILP) and use it to obtain the optimal solution for small-size networks. Finally, we propose an efficient heuristic algorithm for use in large-size networks. Our experimental results on real network datasets, i.e., Abilene, GEANT and DFN, show that our heuristic produces energy saving, on average, is only 5.8% off from the optimal solution, while using only up to 0.5% of the CPU time required by the ILP.

The rest of this paper is organized as follows. Section II discusses related works followed by Section III, which presents our ILP formulation. In Section IV, we propose our heuristic solution. Section V presents experimental results, and Section VI concludes the paper.

## II. RELATED WORKS

Energy efficiency or reducing the energy consumption of a network has been a key concern of operators and researchers for many years. This issue was first discussed in [8], followed by numerous works such as [9], [13], [14] and [15]. Among these works, only the authors of [9], [14] and [15] address energy savings in large backbone networks that use *bundled* links. However, their work relies on distributed protocols such as OSPF and assume legacy switches that lack programmability and are difficult to manage [16].

In contrast, the SDN paradigm allows network programmability and manageability via centralized controllers. Thus, a SDN is able to better manage the energy consumption of a network. For example, reference [17] considers powering

off links. Further, it guarantees a bounded-delay shortest path among switches and between each switch and its controller. The work assumes the network is a *pure* SDN (*p*-SDN), which consists of only *s*-switches. There are works such as [18]–[20] that consider energy saving in so called *hybrid* SDN or *h*-SDNs that contain both *s*-switches and *l*-switches. These *h*-SDN approaches overcome the financial and trust issues that arise when deploying a *p*-SDN. In particular, solutions that assume a *h*-SDN aim to deploy SDN technologies over multiple time periods or stages.

Caria et al. [21] consider a multi-stage deployment problem. They propose a multi-stage approach to optimize traffic engineering by finding the maximum number of available paths to deliver each traffic demand or flow. A path is available if the *l*-switches that can route the flow through the path are upgraded to *s*-switches. The number of upgraded *l*-switches is constrained by the ratio between the total number of *l*-switches and the given upgrade number of stages. On the other hand, Poularkis et al. [6] use the total budget (in \$) over a maximum number of stages to upgrade a set of *l*-switches. They assume each switch upgrade cost (in \$) decreases with increasing number of stages. Moreover, traffic size (in bytes) also increases over these stages. Their goals are to maximize the number of (i) traffic flows passing through at least one *s*-switch, and (ii) available paths.

The aforementioned works, namely [17]–[20], assume each link contains only one cable. They also consider upgrading *l*-switches over one stage. In contrast, we consider switching off the maximum number of unused cables in a SDN with bundled-links. Moreover, we aim to upgrade *l*-switches into *s*-switches in more than one upgrade period/stage. In addition, to the best of our knowledge, no works have considered the decreasing cost of upgrading switches and increasing traffic demands over multiple stages for SDNs with bundled links. Moreover, we consider maximum link utilization. Lastly, we generalize the work of [6] where we assume each stage has a fixed budget as well as a total budget.

## III. MODEL AND PROBLEM FORMULATION

Let  $G^0(V, E)$  be a *legacy* network that has  $|V|$  nodes or *l*-switches, and  $|E|$  directed links. Each link  $(u, v) \in E$  has  $b_{uv}$  cables, aka *bundle size*. We assume each cable consumes the same amount of energy. Also all cables have the same capacity  $\gamma$ ; this means link  $(u, v)$  has a total capacity of  $c_{uv} = \gamma b_{uv}$ . A network provider aims to upgrade  $G^0(V, E)$  into a SDN over one or more time periods or stages, denoted as  $T \geq 1$ . More specifically, at each stage  $t \in \{1, 2, \dots, T\}$ , the provider selects a set  $V^t \subset V$  of *l*-switches to be replaced with *s*-switches. The duration of each stage  $t$  is determined by the lifetime of a network device; e.g., three to five years [6]. Let  $G^t(V, E)$  be the resulting network after undergoing an upgrade at stage  $t$ . Each link  $(u, v) \in E$  in  $G^t(V, E)$  is a *c*-link if it is adjacent to at least one *s*-switch; otherwise it is a *l*-link. As per [19] and [20], any cable in a *c*-link can be powered-off when it has zero flow rate.

Let  $B$  be the total budget (in \$) over time  $T$ , and  $B^t \leq B$  denotes the maximum available budget at each period  $t$ . We set  $B^t = B/T$ , and any unused budget in period  $t$  can be spent in subsequent stages. We use  $p_v^t$  to denote the cost of upgrading switch  $v$  in period  $t$ , and  $\rho$  is the depreciation rate in upgrade cost of each switch per stage, where  $0 \leq \rho < 1$ . Hence, the upgrade cost of a switch  $v$  at stage  $t$  is given as  $p_v^t = p_v^0 \times (1-\rho)^{t-1}$ , where  $p_v^0$  is the initial cost. Let  $D^t$  denote a set of traffic demands in  $G^t(V, E)$ . For  $d = 1, 2, \dots, |D^t|$ , each demand  $d = (s_d, \tau_d, \omega_d^t)$  is from node  $s_d \in V$  to node  $\tau_d \in V$  with traffic size  $\omega_d^t > 0$  (in bytes). We set  $\omega_d^1 = \omega_d^0$ , where  $\omega_d^0$  is the flow size of demand  $d$  in  $G^0(V, E)$ . Following [6], we assume the size of each demand  $d$  increases with an equal rate of  $\mu \geq 0$  per period. Thus,  $\omega_d^t = \omega_d^0 \times (1 + \mu)^{t-1}$ . Let  $U_{max}$  be the maximum link utilization threshold, where  $0 \leq U_{max} \leq 1.0$ , and  $n_{uv}^t$  is the number of powered-on cables, aka *on-cables*, for link  $(u, v)$  in period  $t$ , where  $0 \leq n_{uv}^t \leq b_{uv}$ . Thus, the maximum utilization of link  $(u, v)$  is given as  $(n_{uv}^t/b_{uv}) \times c_{uv} \times U_{max}$ . Let  $\varepsilon^t$  be the energy saving in time period  $t$ , computed as the ratio between the total number of powered-off cables and the total cables in  $G^t(V, E)$ . We use  $\varepsilon_T = \frac{1}{T} \sum_{t=1}^T \varepsilon^t$  to represent the average energy saving over  $T$  stages.

We formulate our optimization problem as an Integer Linear Program (ILP); see (1). Its objective, i.e., (1a), is to minimize the total number of *on-cables* over  $T$  periods; this ensures the minimal energy is used by the network. Constraint (1b) is the standard flow conservation constraint that ensures a single path between source  $s_d$  and destination  $\tau_d$ . In particular, it ensures all links on the path are active. Let  $y_{d,uv}^t$  be the binary variable that is set to 1 (0) if the path for flow  $d$  is routed (not routed) via link  $(u, v)$  at stage  $t$ . Constraint (1c) ensures that every path must have a path length that is no longer than the length of its shortest path  $\delta_d^0$ . The load of each link is bounded as per constraint (1d), meaning the load must not exceed the capacity of *on-cables* for a given link utilization  $U_{max}$  threshold. Constraint (1e) bounds the number of *on-cables* to the bundle size of each link. Constraints (1f) to (1h) relate to our energy-aware or green multi-stage upgrade model. Specifically, constraint (1f) ensures that at each stage the total upgrade cost does not exceed the budget for that stage. In constraint (1g), the binary variable  $x_v^t$  is set to 1 (0) if node  $v$  is upgraded (not upgraded) at stage  $t$ . The constraint also ensures each switch  $v$  is only upgraded once. Constraint (1h) enforces all cables in each  $l$ -link to be powered-on because only cables in a  $c$ -link can be turned-off. Finally, constraint (1i) defines the two decision variables to be binary.

Note that the aforementioned constraints, except (1g), are repeated for each stage  $t$ , where  $t \in \{1, 2, \dots, T\}$ . Constraints (1b) and (1c) are for each traffic demand  $d \in D^t$ , while constraints (1d), (1e) and (1h) exist for all links  $(u, v) \in E$ . Further, constraint (1g) applies to each  $v \in V$ .

Lastly, if our ILP only consists of constraints (1b) to (1e), then it is equivalent to the NP-complete Two-Commodity Integral Flow (TCIF) problem [22]. Further, if there is no decrease in the switch upgrade cost  $p_v^t$  at each stage  $t$ ,

constraints (1f) to (1g) of the ILP can be reduced to the NP-complete 0/1 multiple knapsack problem [23]. The next section describes a heuristic solution for our problem.

$$\min_{n_{uv}^t, x_v^t, y_{d,uv}^t} \sum_{t=1}^T \sum_{(u,v) \in E} n_{uv}^t \quad (1a)$$

s.t.

$$\sum_{(u,v) \in E} y_{d,uv}^t - \sum_{(v,u) \in E} y_{d,vu}^t = \begin{cases} 1, & u = s_d \\ -1, & u = \tau_d \\ 0, & u \neq s_d, \tau_d \end{cases}, \quad (1b)$$

$$\sum_{(u,v) \in E} y_{d,uv}^t \leq \delta_d^0, \quad (1c)$$

$$\sum_{d \in D^t} (\omega_d^t \times y_{d,uv}^t) \leq (n_{uv}^t/b_{uv}) \times U_{max} \times c_{uv}, \quad (1d)$$

$$0 \leq n_{uv}^t \leq b_{uv}, \quad (1e)$$

$$\sum_{v \in V} (p_v^t \times x_v^t) \leq \sum_{k=1}^t B^t - \sum_{k=1}^{t-1} \sum_{v \in V} (p_v^k \times x_v^k), \quad (1f)$$

$$\sum_{t=1}^T x_v^t \leq 1, \quad (1g)$$

$$n_{uv}^t = \max(n_{uv}^t, b_{uv} \times (1 - \sum_{k=1}^t x_u^k - \sum_{k=1}^t x_v^k)), \quad (1h)$$

$$y_{d,uv}^t, x_v^t \in \{0, 1\}. \quad (1i)$$

#### IV. PROPOSED HEURISTIC-BASED SOLUTION

Our greedy approach **green multi-stage upgrade (GMSU)** is shown in Algorithm 1. For each stage  $t \in \{1, 2, \dots, T\}$ , **GMSU** selects the  $l$ -switch  $v$  with the largest  $w_v$  and an upgrade cost  $p_v^t$  that is no more than the available budget  $B^t$ . Here,  $w_v$  corresponds to the total number of unused cables incident at node  $v$  that can be switched off. Each  $w_v$  is computed as follows,

$$w_v = \sum_{(u,v) \in E} (b_{uv} - n_{uv}^T), \quad \forall v \in V \quad (2)$$

Note that  $n_{uv}^T$  is the required number of *on-cables* in each link  $(u, v)$  such that the network can carry the traffic demand of stage  $T$ . This means for each link  $(u, v)$  at stage  $T$ , the maximum number of cables that can be switched-off or *off-cables* is  $(b_{uv} - n_{uv}^T)$ . We use  $n_{uv}^T$  in (2) such that the *off-cables* at stage  $T$  are also off at each stage  $t < T$ . This ensures we always upgrade the  $l$ -switches with the largest unused cables at the earliest possible stage in order to increase the overall energy saving. Note that the size of each traffic demand  $d$  increases at a rate  $\mu \geq 0$  per stage; the traffic size at each stage  $t$  increases to  $\omega_d^t = \omega_d^0 \times (1 + \mu)^{t-1}$ . Thus, the largest flow for each demand  $d$  occurs at stage  $T$ , and at each stage  $t < T$ , more cables can be turned-off.

Line 2 of Algorithm 1 generates the shortest path  $P_d^0$  for each demand  $d$  in  $G^0(V, E)$ . Lines 3-5 compute the traffic volume  $f_{uv}^T$  for each link  $(u, v) \in E$  at stage  $T$ , for  $\omega_d^T =$

---

**Algorithm 1 : Green Multi-Stage Upgrade (GMSU)**

---

**Input:**  $G^0(V, E)$ ,  $T$ ,  $B$ ,  $D$ ,  $p_v^0$ ,  $U_{max}$ ,  $\mu$ ,  $\rho$ **Output:**  $P_d^0$ ,  $V^t$ ,  $f_{uv}^t$ ,  $n_{uv}^t$ 

```
1: for ( $d \in D$ ) do
2:   Generate a shortest path  $P_d^0$  and store it in  $P^0$ 
3:   for ( $(u, v) \in P_d^0$ ) do
4:      $f_{uv}^T = f_{uv}^T + \omega_d^T$ 
5:   end for
6: end for
7: for ( $(u, v) \in E$ ) do
8:    $n_{uv}^T = \lceil f_{uv}^T / (\gamma \times U_{max}) \rceil$ 
9: end for
10: Compute  $w_v$  for each  $v \in V$  using (2)
11:  $X = V$ 
12: for ( $t \in \{1, \dots, T\}$ ) do
13:    $\{V^t, B^t\} = \text{Selection}(X, t)$ 
14:    $B^{t+1} = B^t + B^t$ 
15:   Compute  $\varepsilon^t$ 
16: end for
```

---

$\omega_d^1 \times (1 + \mu)^{T-1}$ . Thus,  $f_{uv}^T$  gives the maximum flow size at link  $(u, v)$ . Lines 7-9 calculate the number of *on*-cables  $n_{uv}^T$  of link  $(u, v)$  at stage  $T$  that can carry flow of size  $f_{uv}^T$ . Line 10 then uses (2) to compute  $w_v$  for each node  $v \in V$ . Line 11 initializes a set of *candidate* nodes  $X$  with all  $l$ -switches in  $V$ . For each stage  $t$ , Line 13 uses the function **Selection()**, shown in Algorithm 2, to obtain  $V^t$  from  $X$  and compute any remaining budget  $B^t \geq 0$ . At Line 1, **Selection()** considers for each node  $v$ , (i) upgrade cost is within budget, and (ii) has weight  $w_v > 0$ , i.e., has cables to switch off. Among all nodes that satisfy the two criteria, Line 2 selects node  $v$  with the largest value  $w_v$ . Line 3 removes the selected node from  $X$ , Line 4 includes  $v$  into the set  $V^t$ , and Line 5 computes the remaining budget. Finally, Lines 6-8 reduce the weight  $w_u$  for each neighbor node  $u$  of the selected node  $v$  by the total cables switched off by node  $v$ . Lines 1-9 are repeated until either criterion (i) or (ii) fails. In Line 14, the remaining budget at stage  $t$  is added to the budget for stage  $t + 1$ , i.e.,  $B^{t+1}$ . Finally, Line 15 calculates the energy saving ratio  $\varepsilon^t$ , which is the total number of *off*-cables ( $b_{uv} - n_{uv}^t$ ) over the total cables  $b_{uv}$ , for all links  $(u, v) \in E$ . Note that  $n_{uv}^t$  is the number of *on*-cables required to carry the total traffic volume  $f_{uv}^t = f_{uv}^T / (1 + \mu)^{T-t}$  at each period  $t$ . If  $u$  and  $v$  are both  $l$ -switches, we set  $n_{uv}^t = b_{uv}$ .

As an example, we show how to upgrade the topology shown in Fig. 1a over  $T = 2$  stages using a budget of  $B = 40$ , meaning we have  $B^1 = B^2 = 20$ . Each switch  $v$  has an upgrade cost of  $p_v^1 = \$15$  with a decrease rate of  $\rho = 0.2$  per stage, and thus,  $p_v^2 = \$12$ . Each link  $(u, v)$  has  $b_{uv} = 2$  cables; each cable has a capacity of  $\gamma = 5$ , and a maximum utilization limit of  $U_{max} = 0.8$ . Further assume there are five flows in Fig. 1a, i.e.,  $D^1 = \{(2,7,2), (3,5,1), (1,8,1), (6,8,1), (6,1,0.5)\}$ , with an increase rate of  $\mu = 0.2$  per stage. Thus,  $D^2 = \{(2,7,2.4), (3,5,1.2), (1,8,1.2), (6,8,1.2),$

---

**Algorithm 2 : Selection()**

---

**Input:**  $X$ ,  $t$ **Output:**  $V^t$ ,  $B^t$ 

```
1: for ( $v \in X$  that has  $p_v^t \leq B^t$  and  $w_v > 0$ ) do
2:   Find  $v$  that has  $\max\{w_v\}$ 
3:    $X = X - v$ 
4:    $V^t = V^t \cup v$ 
5:    $B^t = B^t - p_v^t$ 
6:   for ( $u \in X$  and  $(u, v) \in E$ ) do
7:      $w_u = w_u - (b_{uv} - n_{uv}^T)$ 
8:   end for
9: end for
```

---

$(6,1,0.6)\}$ . Line 2 generates the shortest path for each demand, e.g.,  $P_1^0 = (2, 4, 7)$  and  $P_5^0 = (6, 3, 1)$ . Lines 3-5 compute  $f_{uv}^T$  e.g.,  $f_{2,4}^2 = 2.4$  and  $f_{1,2}^2 = 0$ . Lines 7-9 obtain each  $n_{uv}^T$  for link  $(u, v)$ , e.g.,  $n_{2,4}^2 = 1$  and  $n_{1,2}^2 = 0$ . Line 10 computes  $w_v$  of each node  $v$ , e.g.,  $w_1 = 4$  and  $w_2 = 5$ . Among all switches in  $X$ , only switch  $v = 2$  can be upgraded at  $t = 1$ . Thus, Line 13 returns  $V^1 = \{2\}$  and  $B^1 = 20 - 15 = 5$ . Line 7 of **Selection()** then decreases weight  $w_u$  if node  $u$  is a neighbour of node  $v = 2$ , e.g.,  $w_1 = w_1 - (b_{1,2} - n_{1,2}^2) = 2$ . So, in Lines 14-15, we obtain  $B^2 = 20 + B^1 = 25$ . As shown in Fig. 1b, upgrading switch-2 at  $t = 1$  can switch-off five cables; line 17 of **GMSU** outputs one *off*-cable in link  $(2,4)$  and two *off*-cables each in link  $(1,2)$  and  $(2,5)$ . Thus,  $\varepsilon^1 = 5/24 \times 100 = 20.83\%$ . For the last stage  $t = 2$  with  $B^2 = 25$ , two more switches are selected for upgrade, i.e.,  $V^2 = \{8, 6\}$ . So, for  $t = 2$ , we get a total of 14 *off*-cables: two cables from each c-link, which are denoted as dashed lines, and one cable from each of the c-links  $(2,4)$ ,  $(3,6)$ ,  $(5,6)$  and  $(5,8)$ ; hence,  $\varepsilon^2 = 58.33\%$ . Thus,  $\varepsilon_2 = 39.58\%$ .

Line 2 of **GMSU**, assuming Floyd-Warshall's algorithm [24], takes  $O(|V|^3)$ . The calculation of  $f_{uv}^T$  for all  $|D|$  demands in Lines 3-5 require  $O(|D||E|)$ . Note that all values of  $\omega_d^T$  can be computed in  $O(|V|^2)$ . Lines 7-9 require  $O(|E|)$ . Line 10 requires  $O(|E|)$  and Line 11 takes  $O(|V|)$  to copy all nodes in  $V$  to  $X$ . The function **Selection** in Line 13 takes  $O(|V|^2 + |V| + |V||E|) = O(|V||E|)$  because (i) Line 2 takes  $O(|V|)$ , (ii) Lines 3-5 each takes  $O(1)$ , (iii) Lines 6-8 requires  $O(|E|)$ , and these lines are repeated at most  $|V|$  times. Line 14 takes  $O(1)$ , while Line 15 takes  $O(|E|)$ . Note that Lines 13-15 are repeated  $T$  times, for a constant value of  $T$ , e.g., we use  $T = 5$  at maximum. Thus, Lines 12-16 requires  $O(|V||E|)$ . Overall, the time complexity of **GMSU** is  $O(|V|^2|E|)$  since we consider  $|E| \geq |V|$  and  $|D| \leq |V|^2$ .

## V. PERFORMANCE EVALUATION

We have implemented **GMSU** in C++ and used IBM ILOG CPLEX Optimizer to solve the ILP. Our experiments are conducted on a 64-bits Windows machine with an Intel-core-i7 CPU @3.60 GHz and 16 GB of memory. We use five actual network topologies, listed in Table I. For Abilene and GÉANT, we use their actual traffic matrices. Since we could not find the *real* traffic matrices of DFN, Deltacom and TATA, we

use the gravity model [25] to generate their traffic flows. In Section V-A, we first evaluate the scalability of **GMSU** and ILP in terms of their running time (in CPU seconds). Then, in Section V-B and Section V-C, we study increasing budget and number of stages, respectively, on energy saving and the total number of upgraded  $l$ -switches. We set the capacity of each cable to  $\gamma = 2.5$  Gbps. Each link consists of four cables, meaning all links has a capacity of 10 Gbps. For Abilene, however, it has a link with only one cable. The initial cost of each switch is  $p_v^0 = \$100\text{K}$ . The depreciation rate of upgrading a switch is  $\rho = 40\%$ , and traffic increases at a rate of  $\mu = 22\%$ . The link utilization  $U_{max}$  is 80%.

TABLE I  
RUNNING TIME (IN CPU SECONDS).

Name	$ V $	$ E $	$ D $	Running Time	
				GMSU	ILP
Abilene [26]	12	30	132	0.0008	0.16
GÉANT [27]	23	74	466	0.0013	2.99
DFN [28]	58	174	3306	0.0086	108.1
Deltacom [28]	113	161	12656	0.054	N/A
TATA [28]	145	372	20880	0.092	N/A

#### A. Running Time Performance

We set the total budget to  $B = \$1\text{M}$ ; this is the highest budget used by the authors in [6]. We also set the number of stages to  $T = 3$ . As shown in Table I, **GMSU** requires only less than 0.1 second to generate the result for each network. By contrast, ILP fails to produce any result for Deltacom and TATA after running for more than five hours. Thus, from here on, we consider only Abilene, GÉANT and DFN when comparing the performance of **GMSU** and ILP.

#### B. Effect of Increasing Budget

We consider the following budget  $B = \{\$200\text{K}, \$400\text{K}, \$600\text{K}, \$800\text{K}, \$1\text{M}\}$  to show the impact of increasing budget on the average energy saving  $\varepsilon_T$ , and we set  $T = 3$ . Fig. 2a shows that  $\varepsilon_T$  of **GMSU** is slightly less as compared to ILP. More specifically, on average **GMSU** produces  $\varepsilon_T$  that is only 0.22%, 3.52% and 1.38% off from the optimal for Abilene, GÉANT and DFN, respectively. Fig. 2a also shows that **GMSU** produces  $\varepsilon_T$  of only up to 22.98% and 20.03% for Deltacom and TATA respectively. These percentages are significantly lower as compared to the other three networks, i.e., up to 61.4%, 55.41% and 48.75% for Abilene, GÉANT, and DFN, respectively. Deltacom and TATA have larger number of  $l$ -switches to upgrade than the other three networks. Thus, each allocated budget for the two networks can upgrade significantly smaller percentage of  $l$ -switches. As an example, with a budget of  $\$1\text{M}$ , **GMSU** can upgrade seven  $l$ -switches of Abilene, meaning **GMSU** upgraded 58.33% of the total number of  $l$ -switches. Using the same budget for TATA, **GMSU** can upgrade only 12.41% of the total  $l$ -switches. To

validate our explanation, for TATA, we rerun the experiment with a budget of  $B = \$5\text{M}$ , which is sufficient to upgrade 51.72% of the network's  $l$ -switches. We find that **GMSU** produces  $\varepsilon_T = 54.97\%$ .

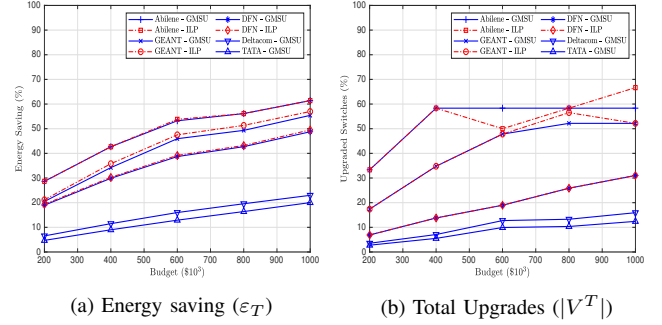


Fig. 2.  $\varepsilon_T$  and  $|V^T|$  for various  $B$ .

As expected, Fig. 2a shows that either **GMSU** or ILP results in more  $\varepsilon_T$  when the allocated budget is larger. The larger budget allows both solutions to upgrade more switches. Recall that  $\varepsilon_T$  is computed only from switched-off  $c$ -links. Thus, more  $s$ -switches can potentially power off more cables. However, as shown in Fig. 2b, ILP does not always translate into more upgraded  $l$ -switches when budget gets larger. For example, ILP upgrades 58.33% of  $l$ -switches in Abilene using a budget of  $B = \$400\text{K}$ , while only 50% of  $l$ -switches with a larger budget of  $B = \$600\text{K}$ . The reason is because for each stage, ILP aims to upgrade  $l$ -switches to minimize the number of *on*-cables, i.e., not to maximize the number of upgraded switches. On the other hand, Fig. 2b shows that **GMSU** consistently upgrades more  $l$ -switches with larger budgets.

#### C. Effect of Increasing Number of Stages

We use  $B = \$1\text{M}$  and vary the value of  $T$  from 1 to 5 to see the effect of  $T$  on  $\varepsilon_T$ . Fig. 3a shows that, for each stage  $T$ , **GMSU** is able to obtain  $\varepsilon_T$  that is equal (close) to that of the ILP for Abilene (GÉANT and DFN). For example, **GMSU** produces the optimal  $\varepsilon_T$  at 57.19% for Abilene, and only 5.8% (1.23%) off from the optimal result for GÉANT (DFN) for  $T = 5$ .

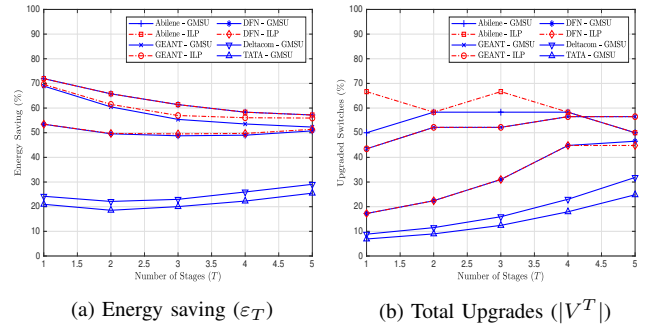


Fig. 3.  $\varepsilon_T$  and  $|V^T|$  for various  $T$ .

One can observe that the  $\varepsilon_T$  for Abilene, GÉANT, and DFN decreases as  $T$  increases. More specifically, for Abilene (DFN), the  $\varepsilon_T$  generated by **GMSU** decreases from 71.93%

to 57.19% (53.45% to 51.38%) when  $T$  increases from 1 to 5. In contrast, the  $\varepsilon_T$  for Deltacom (TATA) increases from 24.22% to 29.07% (20.97% to 25.48%) when  $T$  increases from one to five. The reason is because the allocated budget  $B = \$1\text{M}$  can upgrade a larger percentage of switches in smaller networks in earlier stages as compared to larger networks. Thus, for smaller networks, there are fewer number of switches in later stages with cables that can be turn off, i.e.,  $w_v > 0$ . Further, with increasing traffic sizes, the remaining switches has smaller number of *off*-cables and thus, upgrading them does not significantly increase  $\varepsilon_T$ . On the other hand, for larger networks, due to cheaper upgrade cost, more switches can be upgraded in later stages, resulting in larger  $\varepsilon_T$ . Note that for these larger networks, increasing traffic sizes also decreases the number of cables that can be switched-off. However, the effect is less prominent because there are more switches with unused cables to select in larger networks. Nevertheless, as shown Fig. 3b, in all networks, small or large, one can upgrade more switches with a larger  $T$  value. Note that **GMSU** initially sets each stage to have an equal budget. Any unused budget at an earlier stage  $t$  is added to the budget of period  $t + 1$ . As the upgrade cost at  $t + 1$  is less expensive than that at  $t$ , **GMSU** is able to upgrade more  $l$ -switches with the budget at  $t + 1$  than with the budget at  $t$ . Thus, a larger number of stages  $T$  can result in more upgraded  $l$ -switches with unused cables to turn-off at the later periods.

## VI. CONCLUSION

This paper considers the problem of upgrading a legacy network into a SDN over multiple stages. Our aim is to minimize energy expenditure subject to the total available budget as well as the budget in each stage, increasing traffic volume and decreasing upgrade cost over multiple stages. We have formulated the problem as an ILP and proposed a greedy solution called **GMSU**. Our experiments using five real networks show that increasing budget and stages increase energy savings. The simulation results show, on average, **GMSU** is able to produce energy saving no more than 5.8% off from the optimal solution. Further, while ILP fails to generate results on networks with a high number of edges, **GMSU** requires only less than one second of CPU time on the such networks. As a future work, we aim to optimize the number of stages to minimize the total budget.

## REFERENCES

- [1] S. Jain *et al.*, "B4: experience with a globally-deployed software defined WAN," in *ACM SIGCOMM*, Hong Kong, China, Aug. 2013, pp. 3 – 14.
- [2] C.-Y. Hong *et al.*, "Achieving high utilization with software-driven WAN," in *Proc. ACM SIGCOMM*, Hong Kong, China, Aug 2013, pp. 15–26.
- [3] Y. E. Oktian, S. Lee, H. Lee, and J. Lam, "Distributed SDN controller system: A survey on design choice," *Comput. Netw.*, vol. 121, pp. 100–111, Jul. 2017.
- [4] D. K. Hong, Y. Ma, S. Banerjee, and Z. M. Mao, "Incremental deployment of SDN in hybrid enterprise and ISP networks," in *Proc. Symp. on SDN Research*, Santa Clara, CA, USA, Dec. 2016, pp. 1–7.
- [5] Sandhya, Y. Sinha, and K. Haribabu, "A survey: Hybrid SDN," *J. Netw. Comput. Appl.*, vol. 100, pp. 35–55, Dec. 2017.
- [6] K. Poularakis, G. Iosifidis, G. Smaragdakis, and L. Tassiulas, "One step at a time: Optimizing SDN upgrades in ISP networks," in *Proc. IEEE INFOCOM*, Atlanta, GA, USA, May 2017, pp. 1–9.
- [7] R. Maaloul, L. Chaari, and B. Cousin, "Energy saving in carrier-grade networks: A survey," *Computer Standards & Interfaces*, vol. 55, pp. 8–26, Jan. 2018.
- [8] M. Gupta and S. Singh, "Greening of the internet," in *Proc. ACM SIGCOMM*, Karlsruhe, Germany, Oct. 2003, pp. 19–26.
- [9] W. Fisher, M. Suchara, and J. Rexford, "Greening backbone networks: reducing energy consumption by shutting off cables in bundled links," in *Proc. ACM SIGCOMM*, New Delhi, India, Aug. 2010, pp. 29–34.
- [10] M. F. Tuysuz, Z. K. Ankarali, and D. Gzpek, "A survey on energy efficiency in software defined networks," *Comput. Netw.*, vol. 113, pp. 188–204, Feb. 2017.
- [11] K. Christensen, P. Reviriego, B. Nordman, M. Bennett, M. Mostowfi, and J. Maestro, "IEEE 802.3az: the road to energy efficient ethernet," *IEEE Commun. Mag.*, vol. 48, no. 11, pp. 50–56, Nov. 2010.
- [12] M. Rodriguez-Perez, M. Fernandez-Veiga, S. Herreria-Alonso, M. Hmila, and C. Lopez-Garcia, "Optimum traffic allocation in bundled energy-efficient ethernet links," *IEEE Syst. J.*, vol. 12, no. 1, pp. 593–603, Mar. 2018.
- [13] M. Zhang, C. Yi, B. Liu, and B. Zhang, "GreenTE: Power-aware traffic engineering," in *Proc. 18th IEEE ICNP*, Kyoto, Japan, Oct. 2010, pp. 21–30.
- [14] G. Lin, S. Soh, K.-W. Chin, and M. Lazarescu, "Efficient heuristics for energy-aware routing in networks with bundled links," *Comput. Netw.*, vol. 57, no. 8, pp. 1774–1788, Jun. 2013.
- [15] Y. Yang, M. Xu, D. Wang, and S. Li, "A hop-by-hop routing mechanism for green internet," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 1, pp. 2–16, Jan. 2016.
- [16] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, "A roadmap for traffic engineering in SDN-OpenFlow networks," *Comput. Netw.*, vol. 71, pp. 1–30, Oct 2014.
- [17] A. R. Rivera, K.-W. Chin, and S. Soh, "GreCo: An energy aware controller association algorithm for software defined networks," *IEEE Commun. Lett.*, vol. 19, no. 4, pp. 541–544, Apr. 2015.
- [18] Y. Wei, X. Zhang, L. Xie, and S. Leng, "Energy-aware traffic engineering in hybrid SDN/IP backbone networks," *J. Commun. Netw.*, vol. 18, no. 4, pp. 559–566, Aug. 2016.
- [19] H. Wang, Y. Li, D. Jin, P. Hui, and J. Wu, "Saving energy in partially deployed software defined networks," *IEEE Trans. Comput.*, vol. 65, no. 5, pp. 1578–1592, May 2016.
- [20] N. Huin, M. Rifai, F. Giroire, D. L. Pacheco, G. Urvoy-Keller, and J. Moulherac, "Bringing energy aware routing closer to reality with SDN hybrid networks," *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 4, pp. 1128 – 1139, Dec. 2018.
- [21] M. Caria, A. Jukan, and M. Hoffmann, "A performance study of network migration to SDN-enabled traffic engineering," in *Proc. IEEE GLOBECOM*, Atlanta, GA, USA, Dec. 2013, pp. 1391 – 1396.
- [22] S. Even, A. Itai, and A. Shamir, "On the complexity of time table and multi-commodity flow problems," in *Proc. 16th Annual Symposium on Foundations of Computer Science (SFCS)*, Oct. 1975, pp. 184–193.
- [23] D. Pisinger, "An exact algorithm for large multiple knapsack problems," *European Journal of Operational Research*, vol. 114, no. 3, pp. 528–541, May 1999.
- [24] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Sten, *Introduction to Algorithms: Third Edition*. London, England: MIT Press, 2009.
- [25] M. Roughan, "Simplifying the synthesis of internet traffic matrices," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 5, pp. 93–96, Oct. 2005.
- [26] Yin Zhang's Abilene Traffic Matrix. [Online]. Available: <http://www.cs.utexas.edu/~yzhang/research/AbileneTM/>
- [27] M. Roughan. (2017) Internet Traffic Matrices. [Online]. Available: [http://www.maths.adelaide.edu.au/matthew.roughan/project/traffic\\_matrix/](http://www.maths.adelaide.edu.au/matthew.roughan/project/traffic_matrix/)
- [28] The Internet Topology Zoo. [Online]. Available: <http://www.topology-zoo.org/dataset.html>