Green Multi-Stage Upgrade for Bundled-Links SDN/OSPF-ECMP Networks

Lely Hiryanto, Sieteng Soh
Curtin Univ., WA, Australia
{lely.hiryanto, s.soh}@curtin.edu.au

Kwan-Wu Chin
Univ. of Wollongong, NSW, Australia
kwanwu@uow.edu.au

Duc-Son Pham, Mihai Lazarescu
Curtin Univ., WA, Australia
{ducson.pham, m.lazarescu}@curtin.edu.au

***Abstract*—This paper considers the problem of upgrading a *legacy* network into a Software Defined Network (SDN) over multiple stages and maximizing energy saving (ES) in the resulting upgraded network or hybrid SDN. In each stage, an operator needs to select and replace *legacy* switches with SDN switches and seek to switch off as many cables as possible over each link. This paper addresses the said problem where it considers (i) the available budget at each stage, (ii) maximum path delays, (iii) maximum link utilization, (iv) per-stage increase (decrease) in traffic size (upgrade cost), and (v) each non SDN switch must comply with the Open Shortest Path First (OSPF)-Equal Cost Multi-Path (ECMP) protocol. It outlines a Mixed Integer Program (MIP) and a heuristic algorithm called M-GMSU. The results show that (i) MIP and M-GMSU achieve ES of up to 71.93%, (ii) using a larger budget and/or number of stages increases ES, and (iii) the ES of M-GMSU is within 3.55% away from the optimal ES computed by MIP.**

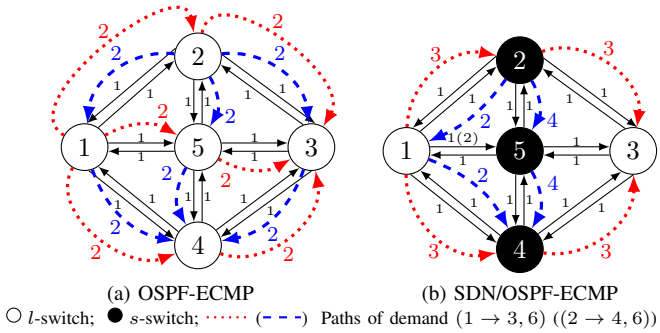
***Index Terms*—Energy Saving, Bundled Links, Hybrid SDNs, Multi-stage Upgrade, Multi-Path Routing, OSPF-ECMP.**

I. Introduction

Software Defined Networks (SDNs) offer operators a new paradigm to manage network demands [1]. An SDN consists of a set of forwarders called SDN-switches or *s*-switches and controllers [1]. These controllers provide a global view of a network, and help operators optimize network performance such as maximizing link utilization (MLU) [2] and/or energy saving (ES) [3]. Consequently, network operators are keen to upgrade their *legacy* networks into SDNs. To do so, they must consider their available budget, maturing SDN technology and cost reduction or depreciation of network equipment over time. Hence, *legacy* switches are likely to be upgraded over multiple stages, creating so called *hybrid*-SDNs, which contain *legacy* or *l*-switches along with *s*-switches.

Recently, energy efficiency has also become a key concern to network operators. It is well-known that current networks are over-provisioned, e.g., link bandwidth, to satisfy traffic demands during peak hours, and they are under-utilized during off-peak periods [4]. To this end, backbone networks are now utilizing IEEE 802.1AX, a *bundled*-link technology comprising of logical links formed from multiple physical cables. The use of IEEE 802.1AX enables network operators to scale the bandwidth or the number of cables in each link as per traffic demands [4]. More importantly, during off-peak hours, unused cables can be switched off to reduce energy cost.

Henceforth, this paper considers a novel network upgrade problem. Specifically, it presents solutions for upgrading a subset of *l*-switches into *s*-switches over multiple stages. In addition, the resulting *hybrid*-SDN must support multi-path routing, and allow each *s*-switch to turn-off the maximum number of *unused* cables. Moreover, (i) active cables must have sufficient capacity to carry all traffic demands, (ii) each path has a delay no larger than a given delay constraint, (iii) there is a maximum budget to upgrade switches per stage, and (iv) each *l*-switch complies with the Open Shortest Path First (OSPF)'s Equal Cost Multi-Path (ECMP) protocol. In addition, the problem must consider traffic volume (switch upgrade cost) that increases (decreases) over stages.



(a) OSPF-ECMP (b) SDN/OSPF-ECMP

Fig. 1: Problem illustration.

To illustrate our problem, consider Fig. 1a. Each link has the indicated cost, has two cables, capacity of five units, and a MLU of 100%. Consider a traffic demand of six units in size from node 1 to 3, denoted as $(1 \rightarrow 3, 6)$, and another demand $(2 \rightarrow 4, 6)$. As shown in Fig. 1a, *l*-switch-1 splits the first demand *equally* into three equal-cost paths $(1, 2, 3)$, $(1, 4, 3)$, and $(1, 5, 3)$, each with a flow of size two and path cost of two; see the dotted lines. The second demand is also split in a similar manner; see the dashed lines. Recall that an unused cable can be off only if at least one of its end nodes is an *s*-switch. Thus, there is no energy saving.

Now consider a scenario where upgrade is carried out over one stage with a total budget of \$45 and the cost to upgrade each *l*-switch is \$15. First, consider upgrading *l*-switches $\{1, 2, 3\}$ in Fig. 1a to turn off 19 unused cables, e.g., one cable of link $(1, 2)$ and both cables of link $(4, 3)$. Here, we can save ES $= 19/32 \times 100\% = 59.38\%$ of energy. Next, Fig. 1b considers upgrading *l*-switches $\{2, 4, 5\}$ and uses them to increase the number of unused cables that can be powered off. *S*-switch-1 splits demand $(1 \rightarrow 3, 6)$ *equally* onto paths $(1, 2, 3)$ and $(1, 4, 3)$; each with a flow of size three. Here, we adjust the cost of link $(1, 5)$ from one to two such that path

$(1, 5, 3)$ is no longer the shortest path for demand $(1 \rightarrow 3, 6)$; see the link cost in bracket. On the other hand, *s*-switch-2 splits demand $(2 \rightarrow 4, 6)$ onto path $(2, 1, 4)$ and $(2, 5, 4)$ with *unequal* flow sizes of two and four, respectively. *s*-switch-4 and *s*-switch-5 can now turn off six more cables, which yields a higher ES of $(6 + 19)/32 \times 100\% = 78.12\%$.

Our contributions are as follows. Firstly, we propose an optimization problem to maximize energy saving in a *hybrid*-SDN by jointly solving two sub-problems: (i) multi-period *l*-switch upgrade, and (ii) splitting traffic optimally via *s*-switches and setting link cost to ensure that each *l*-switch complies with the OSPF-ECMP protocol. Secondly, we formalize our problem as a Mixed Integer Programming (MIP) to obtain the optimal solution for small-sized networks. Finally, we propose a heuristic algorithm for use in large-scale networks.

II. Related Works

Some works, e.g., [4] and [5], aim to maximize energy saving in backbone networks with *bundled* links. Their goal is to switch off as many *unused cables* as possible. However, these works only consider *legacy* networks and OSPF and MPLS protocols. On the other hand, works such as [6] consider *pure* SDNs, which consist of only *s*-switches. For example, the authors of [6] aim to power off *unused links* and also consider bounding the delay of paths among switches and between each switch and its controller.

Some works consider *hybrid*-SDNs containing both *s*-switches and *l*-switches. The work in [3] splits traffic via *s*-switches to maximize ES but it forces each *l*-switch to use a single shortest path computed by OSPF. Another effort is [7], which partially upgrades a MPLS network into SDN to minimize power usage. The work in [2], [3] and [7] does not consider the problem of selecting which *l*-switch to upgrade. On the other hand, the authors of [8] aim to minimize power usage via selectively upgrading up to m *l*-switches. They assume OSPF routing for all switches. The work in [3], [7], and [8] upgrades networks over one stage.

Poularkis *et al.* [9] address a multi-stage SDN deployment problem. They consider an upgrade cost (in \$) that decreases over time. They assume that traffic size (in bytes) increases over multiple stages. Given a total budget (in \$), they aim to upgrade *l*-switches to maximize the number of (i) traffic flows that passes through at least one *s*-switch, and (ii) paths made available by *s*-switches to route traffic flows.

Recently, the work in [10] addresses a multi-stage SDN deployment problem. Its goal is to maximize ES by shutting down as many *unused cables* in each link as possible. It also considers (i) a maximum budget at each stage, (ii) MLU, (iii) single shortest path routing, and (iv) the upgraded network must be able to support existing flows. As in [9], the authors of [10] considered decreasing switch upgrade cost and increasing traffic volume over time.

This paper addresses a similar problem as in reference [10]. However, this paper considers multi-path routing via OSPF-ECMP. Further, routes in [10] remain unchanged after each upgrade. In contrast, this paper aims to reroute traffic to further maximize ES. Note that considering OSPF-ECMP leads to two *hard* problems that do not exist in [10], i.e., (i) each *s*-switch may need to split traffic *unequally* over multi-paths, and (ii) link costs may need to be adjusted to ensure each *l*-switch complies with OSPF-ECMP. Guo *et al.* [2] consider upgrading networks that support OSPF-ECMP protocol. Thus, they [2] also addressed problems (i) and (ii). However, their goal is to minimize MLU. Further, the upgrade is over one stage and thus does not consider changing traffic sizes as well as upgrade cost over multiple stages.

III. Network and Mathematical Model

Network Model: Let $G^0(V, E)$ be a *legacy* network. It has $|V|$ nodes or *l*-switches and $|E|$ directed links. Each link $(u, v) \in E$ has a *bundle size* of b_{uv} cables, capacity c_{uv} and transmission delay π_{uv} (in seconds). Further, each bundled-link uses IEEE 802.3az cables [11]; each of which is either active or place in *sleep* or low-power state. Each cable has capacity γ, and thus we have $c_{uv} = b_{uv} \times \gamma$.

Let $T \geq 1$ be the planning horizon. The duration of each stage t is determined by the lifetime of network devices; e.g., three to five years [9]. Let $G^t(V, E)$ be the network after undergoing an upgrade at stage t. Let $V^t \subset V$ denote the *l*-switches that have been upgraded to *s*-switches. 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 [7], cables in a *c*-link are powered off when they have a zero flow rate.

Let B be the total budget (in \$) over time T, and $B^t \leq B$ denotes the maximum budget of each period $t \in [1, T]$. Any unused budget in period t, denoted by ΔB^t, can be spent in subsequent stages. Thus, we set $B^t = B/T + \Delta B^{t-1}$. Let p_v^t be the cost of upgrading switch v in period t. The upgrade cost of a switch may vary over time depending on different models and type, e.g., *edge* or whether it is a *core* switches [9]. We use ρ to denote the depreciation rate in the upgrade cost of each switch per stage, where $0 \leq \rho < 1$. Hence, we have $p_v^t = p_v^0 \times (1 - \rho)^{t-1}$, where p_v^0 is the initial cost. The total cost to upgrade *l*-switches in V^t cannot exceed the budget B^t.

Let $D^t = \{(s_d, \tau_d, \omega_d^t) \mid \forall d \in [1, |D^t|]\}$ denote a set of traffic demands in $G^t(V, E)$. Nodes $s_d \in V$ and $\tau_d \in V$, respectively, represent the source and destination of each demand $d \in [1, |D^t|]$, while $\omega_d^t > 0$ (in bytes) is its traffic volume. Let $D^0 = D^1$ denote the set of traffic demands in $G^0(V, E)$, and ω_d^0 is the initial traffic volume of demand $d \in D^0$. As in [9], we assume the traffic volume for each demand increases with each successive stage with rate $\mu \in [0, 1]$. Thus, we have $\omega_d^t = \omega_d^0 \times (1 + \mu)^{t-1}$. We assume network $G^0(V, E)$ has sufficient capacity to carry all demands at their maximum volume, i.e., ω_d^T for each demand d.

For each demand d, let $P_d^x = \{P_{d,i}^x \mid \forall x \in V, x \neq \tau_d, \forall d \in [1, |D^0|], \forall i \in [1, |P_d^x|]\}$ be a set of paths from node x to node τ_d. Let $y_{d,uv,i}^t$ be a binary variable that is set to 1 (0) if a link (u, v) is included (not included) in any path $P_{d,i}^x$. Thus, each path $P_{d,i}^x \in P_d^x$ is represented as $P_{d,i}^x = \{(u, v) \mid y_{d,uv,i}^t = 1, \forall (u, v) \in E\}$. The delay of each path $P_{d,i}^x$, denoted by $\delta_{d,i}^x$, is computed as the sum of transmission delays over all links

in the path, i.e., $\delta^x_{d,i} = \sum_{(u,v)\in P^x_{d,i}} \pi_{uv}$. Let $\delta^x_{\min,d}$ ($\delta^x_{\max,d}$) be the minimum (maximum) delay among all paths in P^x_d. We consider a delay constraint that allows users to use paths of up to $(\sigma - 1) \times 100\%$ longer than their original delays, i.e., $\delta_{\max,d} = \lceil \sigma \times \delta^{s_d}_{\min,d} \rceil$, for a delay multiplier $\sigma = [1.0, 2.0]$.

Each link (u, v) in stage t has cost $\psi^t_{uv} > 0$ with a value in $[1, \mathrm{I} = 2^{16-1}]$. Let $\psi^t = \{\psi^t_{uv} \mid \forall (u, v) \in E, \forall t \in [1, T]\}$ represent a set of all link costs for stage t. The cost of each path $P^x_{d,i} \in P^x_d$ in stage t, denoted by $\Psi^{x,t}_{d,i}$, is computed as the sum of link costs in ψ^t over all links in the path, i.e., $\Psi^{x,t}_{d,i} = \sum_{(u,v)\in P^x_{d,i}} \psi^t_{uv}$. Let $\Psi^{x,t}_{\min,d}$ denote the minimum cost among the costs of all paths in P^x_d at stage t. A path $P^x_{d,i} \in P^x_d$ is called a *shortest path* if its cost is equal to the minimum cost, i.e., $\Psi^{x,t}_{d,i} = \Psi^{x,t}_{\min,d}$.

Let U_{max} be the MLU threshold, for $0 \le U_{max} \le 1.0$, and $n^t_{uv} \le b_{uv}$ is the number of *powered-on* or *on*-cables. Thus, the maximum capacity of link (u, v) at stage t is $c^t_{uv} = (n^t_{uv}/b_{uv}) \times U_{\max} \times c_{uv}$. Let ε^t be a ratio between the total number of *powered-off* or *off*-cables, and the total number of cables. For each *l*-link (u, v), we set $n^t_{uv} = b_{uv}$ because we assume an *l*-switch cannot turn off an unused cable. Finally, ε_T denote the average energy saving over T stages.

Mathematical Model: Our MIP, see (1a), aims to minimize the number of *on*-cables over T stages. Constraint (1b) and (1c) respectively ensure flow conservation and the traffic volume carried by each selected path i of demand d, denoted by $f^t_{d,i}$, sums to ω^t_d. Further, constraints (1d)-(1f) enforce each selected path i that routes demand d to meet the delay tolerance $\delta_{\max,d}$ and link capacity c^t_{uv} of each link (u, v) on the path. In constraint (1g), variable x^t_u is set to 1 (0) if *l*-switch u is upgraded (not upgraded) at stage t, and each switch is upgraded only once. Constraint (1h) and (1i) respectively ensures the total upgrade cost at each stage is less than or equal to $B^t = B/T + \Delta B^{t-1}$ and all cables in *l*-links are powered on. Let $z^t_{a,uv}$ be an indicator that is set to 1 (0) if link (u, v), at stage t, is (is not) on the shortest path from node u to node a, and $h^t_{u,a}$ denotes a path cost from u to a. Constraints (1j)-(1l) ensure that the traffic volume from *l*-switch u to destination a is split into equal sized segments; each of which has volume $o^t_{u,a}$ and is routed via each shortest path from u to a. Thus, the cost $h^t_{u,a}$ is minimum and ψ^t_{uv} is in the range $[1, \mathrm{I}]$. Finally, constraint (1m) defines the domain of all decision variables.

Except (1g), all constraints in (1) are for each stage $t \in [1, T]$. Constraint (1b) is for each node $u \in V$, traffic demand $d \in [1, |D^t|]$, and path $i \in [1, K]$. Constraint (1c) applies to each demand $d \in [1, |D^t|]$, while (1d) considers all demands and K paths. Constraint (1e), (1f) and (1i) exist for all links $(u, v) \in E$, while constraint (1g) applies to each $u \in V$. Finally, constraints (1j) - (1l) are evaluated for every destination $a \in V$ and each link $(u, v) \in E$, with a starting node $u \in V$ as a *l*-switch, i.e., $x^t_u = 0$.

Note that our problem is a generalized version of the NP-hard problem in [10] that considers single shortest path routing. Thus, our problem is at least as hard as the problem in [10]. The next section describes a heuristic solution for use in large-scale networks.

$$\min_{y^t_{d,uv,i}, f^t_{d,i}, z^t_{a,uv}, o^t_{u,a}, h^t_{u,a}, x^t_i, n^t_{uv}} \quad \sum_{t=1}^{T} \sum_{(u,v)\in E} n^t_{uv} \tag{1a}$$

s.t.

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

$$\sum_{i=1}^{K} f^t_{d,i} = \omega^t_d \tag{1c}$$

$$\sum_{(u,v)\in E} (y^t_{d,uv,i} \times \pi_{uv}) \le \delta_{\max,d} \tag{1d}$$

$$\sum_{i=1}^{K} \sum_{d=1}^{|D^t|} (y^t_{d,uv,i} \times f^t_{d,i}) \le (n^t_{uv}/b_{uv}) \times U_{\max} \times c_{uv} \tag{1e}$$

$$0 \le n^t_{uv} \le b_{uv} \tag{1f}$$

$$\sum_{t=1}^{T} x^t_u \le 1 \tag{1g}$$

$$\sum_{v\in V} (p^t_v \times x^t_v) \le \sum_{k=1}^{t} B^k - \sum_{k=1}^{t-1} \sum_{v\in V} (p^k_v \times x^k_v) \tag{1h}$$

$$n^t_{uv} = \max \left\{ n^t_{uv}, b_{uv} \times \left(1 - \sum_{k=1}^{t} x^k_u - \sum_{k=1}^{t} x^k_v\right) \right\} \tag{1i}$$

$$\sum_{i=1}^{K} \sum_{d=1,\tau_d=a}^{|D^t|} y^t_{d,uv,i} \times f^t_{d,i} \le z^t_{a,uv} \times \sum_{d=1,\tau_d=a}^{|D^t|} \omega^t_d \tag{1j}$$

$$0 \le o^t_{u,a} - \sum_{i=1}^{K} \sum_{d=1,\tau_d=a}^{|D^t|} y^t_{d,uv,i} \times f^t_{d,i} \le (1 - z^t_{a,uv}) \times \sum_{d=1,\tau_d=a}^{|D^t|} \omega^t_d \tag{1k}$$

$$(1 - z^t_{a,uv}) \le h^t_{v,a} + \psi^t_{uv} - h^t_{u,a} \le (1 - z^t_{a,uv}) \times \mathrm{I} \tag{1l}$$

$$y^t_{d,uv,i}, x^t_u, z^t_{a,uv} \in \{0, 1\};\ f^t_{d,i}, o^t_{u,a}, h^t_{u,a} \ge 0 \tag{1m}$$

IV. Heuristic Solution

This section outlines *Multi-Paths Green Multi Stage Upgrade* (M-GMSU). As per Algorithm 1, it consists of three phases. Given a legacy network $G^0(V, E)$, Phase 1 initially routes each traffic demand according to OSPF-ECMP. Specifically, for each link $(u, v) \in E$, Line 1 of M-GMSU sets the initial link cost, denoted as ψ^0_{uv}, to the link delay π_{uv}. For each demand $d \in [1, |D^T|]$, Line 3 uses Yen's algorithm [12] to generate a set $P^{s_d}_d$ containing up to K paths from s_d to τ_d in order of increasing delay. Thus, $P^{s_d}_d$ initially contains the first K shortest paths. Lines 4-8 distribute the traffic volume ω^T_d equally over all shortest paths in $R^{s_d,0}_d \subseteq P^{s_d}_d$ and compute the total volume f^T_{uv} over each link (u, v). Line 10 calculates the number of *on*-cables n^T_{uv} for each link (u, v). Line 11 uses n^T_{uv} to get the total number of *off*-cables in each link (u, v) of node v, denoted by w_v at stage T. Line 12 concludes Phase 1 by initializing X with all *l*-switches in V.

For each stage t, Phase 2 calls function **Selection**(), shown as Algorithm 2, in Line 14. It aims to upgrade switches starting from the largest ratio w_v/p^t_v to maximize the number of *off*-cables. More specifically, given a set X of candidate *l*-switches to upgrade and budget B^t, **Selection**() returns a set $V^t \subset V$ of

upgraded l-switches, remaining l-switches X, set L that stores each c-link (u, v) with non-zero traffic flow, and remaining budget ΔB^t. Line 15 of M-GMSU then adds the remaining budget ΔB^t to the budget for stage $t+1$. Phase 3 uses **MGTE**() or Algorithm 3 in Line 16 to reroute the traffic flow of demand d via a set $R_d^{s_d,t}$ of *routable* paths; see Definition 1. The goal is to turn-off more cables, when possible.

Definition 1. A set of paths $R_d^{s_d,t} \subseteq P_d^{s_d}$ at stage t from source node s_d to destination node τ_d are *routable* if (i) each link $(u, v) \in R_d^{s_d,t}$ has sufficient capacity to carry flow of demand d, and (ii) each l-switch $x \in R_d^{s_d,t}$ complies with the OSPF-ECMP protocol.

Note that Definition 1 considers the largest traffic volume, i.e., the flow size f_{uv}^T at the last stage T, to ensure each routable path can carry traffic at any stage $t \leq T$.

Algorithm 1 : **M-GMSU**

Input: $G^0(V, E)$, T, B, D^T, p_v^0, U_{max}, μ, ρ
Output: P^t, V^t, f_{uv}^t, n_{uv}^t, ψ_{uv}^t, ε^t

▷ *Phase 1: initialize traffic routing*
1: Set $\psi_{uv}^0 = \pi_{uv}$ for each link $(u, v) \in E$
2: **for** $(d \in [1, |D^T|])$ **do**
3: Generate a set $P_d^{s_d}$ of K alternative paths
4: Put each path $P_{d,i}^{s_d} \in P_d^{s_d}$ with the shortest delay in $R_d^{s_d,0}$
5: Route flow of size $\omega_d^T / |R_d^{s_d,0}|$ via each path $R_{d,i}^{s_d,0} \in R_d^{s_d,0}$
6: **for** (each $R_{d,i}^{s_d,0} \in R_d^{s_d,0}$ and $(u, v) \in R_{d,i}^{s_d,0}$) **do**
7: $f_{uv}^T = f_{uv}^T + \omega_d^T / |R_d^{s_d,0}|$
8: **end for**
9: **end for**
10: $n_{uv}^T = \lceil f_{uv}^T / (\gamma \times U_{max}) \rceil$ for each $(u, v) \in E$
11: Compute w_v for each $v \in V$
12: $X = V$
13: **for** $(t \in \{1, 2, \ldots, T\})$ **do**
▷ *Phase 2: upgrade switches*
14: $\{V^t, \Delta B^t, L\}$ = **Selection**(X, B^t)
15: $B^{t+1} = B^{t+1} + \Delta B^t$
▷ *Phase 3: reroute traffic and set link cost*
16: $\{R^t, \psi^t\}$ = **MGTE**(R^{t-1}, L, X, t)
17: Compute ε^t
18: **end for**

Algorithm 2 : **Selection**()

Input: X, B^t
Output: V^t, ΔB^t, L

1: **for** (each $v \in X$ that has $p_v^t \leq B^t$ **and** $w_v > 0$) **do**
2: Find the v that has **max**$\{w_v / p_v^t\}$
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: **if** $(n_{uv}^T > 0)$ **then**
9: $L = L \cup (u, v)$
10: **end if**
11: **end for**
12: **end for**
13: $\Delta B^t = B^t$

Let $R^t = \{R_d^{s_d,t} \mid \forall d \in [1, |D^t|], \forall t \in [1, T]\}$ contain all routable paths for all demands in D^t at each stage t. Further, let $R_{d,i}^{s_d,t}$ denote the i^{th} *routable path* in $R_d^{s_d,t}$. Line 1 of **MGTE**() initializes set R^t (ψ^t) with paths (link costs) from the previous stage $t-1$, and a set $\mathcal{L}$ with all c-links in set L. We use $R_{d,i}^{x,t} \subseteq R_{d,i}^{s_d,t}$ to denote a *routable sub-path* from an l-switch $x \in R_{d,i}^{s_d,t}$ to node τ_d, where $x \neq \tau_d$ is the closest l-switch to source s_d. Line 2 enumerates each set $R_d^{x,t}$ for every set $R_d^{s_d,t} \in R^t$. Let $\tilde{P}_d^{s_d,t} = \{P_d^{s_d} - R_d^{s_d,t}\}$ denote a set of paths in $P_d^{s_d}$ that are *not selected* at stage t to route demand d. For each set $\tilde{P}_d^{s_d,t}$, Line 2 enumerates a set of sub-paths in P_d^x that are *not selected* at stage t, denoted by $\tilde{P}_d^{x,t} = \{P_d^x - R_d^{x,t}\}$.

Algorithm 3 : **MGTE**()

Input: R^{t-1}, L, X, t
Output: R^t, ψ^t

1: $R^t = R^{t-1}$, $\psi^t = \psi^{t-1}$ and $\mathcal{L} = L$
2: Generate $R_d^{x,t}$ and $\tilde{P}_d^{x,t} = \{P_d^x - R_d^{x,t}\}$
3: **while** $(\mathcal{L} \neq \{\})$ **do**
4: Find $(u, v) \in \mathcal{L}$ with the smallest r_{uv}
5: Put all paths that pass (u, v) in Q_{uv}
6: $n_{uv}^T = n_{uv}^T - 1$
7: **for** (each path $R_{d,i}^{s_d,t} \in Q_{uv}$ and $r_{uv} > 0$) **do**
8: **if** (**Reroute**$(R_{d,i}^{s_d,t})$ == true) **then**
9: $r_{uv} = r_{uv} - f_{d,i}^T$
10: Update $R_d^{s_d,t}$ and $R_d^{x,t}$
11: **end if**
12: **end for**
13: **if** $(r_{uv} > 0)$ **then** $success$ = false
14: **else**
15: $\{\psi^t, success\}$= **LinkCost**(R^t, X)
16: **end if**
17: **if** ($success$ == false) **then**
18: Revert back each changed set $R_d^{s_d,t}$ to its previous paths
19: $n_{uv}^T = n_{uv}^T + 1$
20: $\mathcal{L} = \mathcal{L} - (u, v)$
21: **else if** $(n_{uv}^T == 0)$ **then**
22: $\mathcal{L} = \mathcal{L} - (u, v)$
23: $L = L - (u, v)$
24: **end if**
25: **end while**
26: Compute w_x for each $x \in X$

Lines 4-6 select a c-link $(u, v) \in \mathcal{L}$ that contains a cable with the least used capacity $r_{uv} = (f_{uv}^T - \gamma \times U_{\max} \times \lfloor f_{uv}^T / \gamma \times U_{\max} \rfloor)$, record each path in every set $R_d^{s_d,t} \in R^t$ that passes link (u, v) in a set Q_{uv}, and turn off one cable in link (u, v). To satisfy criterion (ii) of Definition 1, each subpath $R_{d,i}^{x,t} \in R_d^{x,t}$ must carry the same size of traffic. Line 8 uses function **Reroute**() to find $m \geq 1$ paths, each of which can be used to carry traffic volume $f_{d,i}^T$ of path $R_{d,i}^{s_d,t}$. In order, **Reroute**() does one of the following options: (i) use all $m \geq 1$ paths in $\{R_d^{s_d,t} - R_{d,i}^{s_d,t}\}$ such that each path carries an additional volume of $f_{d,i}^T / m$; (ii) if s_d is a s-switch, finds one path in set $\{R_d^{s_d,t} - R_{d,i}^{s_d,t}\}$, which has no common node with any other path in the set, that can carry the extra traffic volume $f_{d,i}^T$. If there is no single path that can carry the extra traffic, find m paths in set of non-selected paths $\tilde{P}_d^{s_d,t}$ that can carry the traffic; or (iii) if s_d is a l-switch, find one path in set $\tilde{P}_d^{s_d,t}$ that can carry traffic volume $f_{d,i}^T$. However, if set $R_d^{s_d,t}$ contains only $R_{d,i}^{s_d,t}$, find $m \geq 1$ paths in $\tilde{P}_d^{s_d,t}$. If Line 8 can reroute path $R_{d,i}^{s_d,t}$, i.e., **Reroute**() returns true, Line 9 reduces the used capacity r_{uv} by $f_{d,i}^T$. Further, Line 10 removes path $R_{d,i}^{s_d,t}$ (subpath $R_{d,i}^{x,t}$) and includes the found m paths (each subpath $R_{d,j}^{x,t}$ of the found m paths) into set $R_d^{s_d,t}$ ($R_d^{x,t}$).

If Lines 7 - 12 fail to reroute all paths in Q_{uv}, i.e., $r_{uv} > 0$, Line 13 sets *success* to false. Otherwise, Line 15 calls function **LinkCost**() to set link costs in ψ^t such that all sub-paths in

$R_d^{x,t}$ become the only shortest sub-paths from node x to τ_d. This function solves the following Linear Program (LP) (2) to adjust the link costs in ψ^t. It extends the LP in [13] to ensure any sub-path in each $\tilde{P}_d^{x,t}$ is not a shortest path. As in [13], LP (2) uses variable $e_{d,i}^{x,t}$ called *excess cost* for each routable sub-path $R_{d,i}^{x,t} \in R_d^{x,t}$ to approximate the optimal link cost. Variable $e_{d,i}^{x,t}$ is zero if *only* all paths in $R_d^{x,t}$ have minimum cost. Formally, **LinkCost**() is defined as

$$\min_{e_{d,i}^{x,t},\, \psi_{uv}^t} \quad \sum_{d=1}^{|D^t|} \sum_{x \in X} \sum_{R_{d,i}^{x,t} \in R_d^{x,t}} e_{d,i}^{x,t} \tag{2a}$$

s.t.

$$\sum_{(u,v) \in R_{d,i}^{x,t}} \psi_{uv}^t - e_{d,i}^{x,t} = \sum_{(u,v) \in R_{d,i+1}^{x,t}} \psi_{uv}^t - e_{d,i+1}^{x,t} \tag{2b}$$

$$\sum_{(u,v) \in \tilde{P}_{d,j}^x} \psi_{uv}^t - \sum_{(u,v) \in R_{d,1}^{x,t}} \psi_{uv}^t - e_{d,1}^{x,t} \geq 1 \tag{2c}$$

$$\psi_{uv}^0 \leq \psi_{uv}^t \leq \mathrm{I} \tag{2d}$$

$$\sum_{(u,v) \in R_{d,1}^{x,t}} \psi_{uv}^t \leq \Psi_{\max,d}^{x,0} \tag{2e}$$

$$e_{d,i}^{x,t} \geq 0 \tag{2f}$$

Objective (2a) minimizes the total excess costs to maximize the number of paths $R_{d,i}^{x,t} \in R_d^{x,t}$ that have an excess cost $e_{d,i}^{x,t}$ of zero. For each set $R_d^{x,t}$, constraint (2b) requires each consecutive pair of sub-paths, i.e., $R_{d,i}^{x,t}, R_{d,i+1}^{x,t} \in R_d^{x,t}$, to have the same minimum cost. As (2b) enforces all sub-paths in $R_d^{x,t}$ to have the same cost, (2c) only needs one sub-path in the set $R_d^{x,t}$, i.e., $R_{d,1}^{x,t}$, to ensure each non-selected sub-path $\tilde{P}_{d,j}^{x,t}$ in $\tilde{P}_d^{x,t}$ has a larger cost than every sub-path in $R_d^{x,t}$. Constraint (2d) bounds the cost ψ_{uv}^t of each link $(u,v) \in E$ within $[\psi_{uv}^0, \mathrm{I} = 2^{16-1}]$. Let $\Psi_{\max,d}^{x,0}$ be the maximum cost among all sub-paths in P_d^x at the initial stage 0. Constraint (2e) uses $R_{d,1}^{x,t} \in R_d^{x,t}$ to ensure that the cost of each sub-path in $R_d^{x,t}$ is no larger than the maximum cost $\Psi_{\max,d}^{x,0}$. Thus, each sub-path in $R_d^{x,t}$ satisfies the maximum delay constraint $\delta_{\max,d}^x$. Both (2d) and (2e) guarantee that any non-selected subpath $P_{d,k}^x \notin P_d^x$ does not have the minimum cost. The last constraint (2f) requires each $e_{d,i}^{x,t}$ to be positive.

If the total excess costs obtained in (2a) is not zero, **LinkCost**() sets $success$ = false, and returns ψ^t with non-updated link costs. Further, Lines 18-20 revert the routable paths R^t to their previous paths, set the cable(s) in link (u,v) back to *on*, and remove link (u,v) from the set $\mathcal{L}$. If **LinkCost**() can successfully update set ψ^t with new link costs, it sets $success$ = true. If link (u,v) has no *on*-cable, i.e., $n_{uv}^T = 0$, Line 22 (Line 23) removes the link from sets $\mathcal{L}$ (L). This allows all cables in the *c*-link (u,v) remain *off* in subsequent stages. Line 26 of **MGTE**() then updates the value w_x of each *l*-switch $x \in X$ because the new routing produced by Lines 3 - 25 can increase the number of *off*-cables incident to the switch. Finally, Line 17 of M-GMSU computes ε^t.

Lines 1-12 of M-GMSU takes $O(K|D||V|(|E| + |V|log|V|))$. **Selection**() takes $O(T|V||E|)$, while **MGTE**() requires $O(T|E|(K^2|D||E|+\alpha))$, assuming LP (2) has a time complexity of $O(\alpha)$, and the LP is called $O(E)$ times per stage. Line 15 (17) needs $O(T)(O(T|E|))$. Since in general we have $|E| \leq |V|^2$, $|D| \leq |V|^2$, and $T = 5$ and $K = 10$ are constants, the overall time complexity of M-GMSU is $O(|V|^2|E|^2 + \alpha|E|)$.

V. Evaluation

We have implemented M-GMSU in C++ and used Gurobi Optimizer to solve our MIP. Our experiments are conducted on a 64-bit Linux machine with an Intel-core-i7 CPU @3.60 GHz and 16 GB of memory. We use five actual network topologies; see Table I, which are also used in [10]. For Abilene and GÉANT, we use their actual traffic matrices. For DFN, Deltacom and TATA, we use the gravity model [14] to generate traffic flows as there are no public traffic matrices. We set $\gamma = 2.5$ Gbps, $b_{uv} = 4$ cables, and U_{max} is set to 80%. As per [9], we set $\rho = 40\%$ and $\mu = 22\%$. We assign an initial upgrade cost p_v^0 of \$50K, \$100K or \$150K by drawing a random number from $\mathcal{N}(2, 0.5)$ for each node v. We then round it to the nearest integer, where a value of one maps to 50K, two to \$100K, and three to \$150K. Each experiment uses M-GMSU and MIP with a delay multiplier $\sigma = 1.1$ and the maximum number of alternative paths $K = 10$.

A. Running Time

To compare the run-time performance (in CPU seconds) of M-GMSU and MIP, we set the budget to B = \$1.2M and consider $T = 3$ stages. From Table I, we see that the run time of M-GMSU and MIP increases with network size and traffic demands. The table shows that the time of M-GMSU is far less than MIP, e.g., 1.07 versus 92100.16 seconds for GÉANT. Further, MIP failed to produce results for DFN, Deltacom and TATA because the optimizer ran out of memory. Thus, for the remaining simulations, we compare the performance of M-GMSU against MIP only for Abilene and GÉANT.

B. Effect of Increasing Budgets

The available budget affects the average energy saving ε_T. Here, we consider B = {\$200K, \$400K, \$600K, \$800K, \$1M, \$1.2M} and $T = 3$. Fig. 2 shows that M-GMSU and MIP produce higher ε_T for a larger budget. For Abilene with budget B = \$200K (B = \$1.2M), M-GMSU and MIP produce $\varepsilon_T = 32.16$ ($\varepsilon_T = 71.64$) and $\varepsilon_T = 34.5$ ($\varepsilon_T = 71.93$), respectively. For GÉANT, MIP fails to produce ε_T for B = {\$200K, \$400K, \$600K} after running for five days. Running M-GMSU on Abilene (GÉANT) results in an energy saving that is only 2.03% (3.53%) off from the optimal ε_T obtained by MIP. From Fig. 2, M-GMSU produces ε_T of only up to 32.22% and 23.97% for Deltacom and TATA, respectively. This is because Deltacom and TATA have a larger number of *l*-switches to upgrade than the other three networks. Thus, each allocated budget for the two larger networks can upgrade significantly smaller percentage of *l*-switches. As energy saving ε_T stems from turning off unused cables in *c*-links, more *s*-switches can potentially switch off more cables.

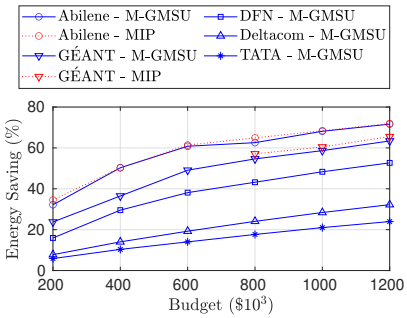


Fig. 2: ε_T for various B

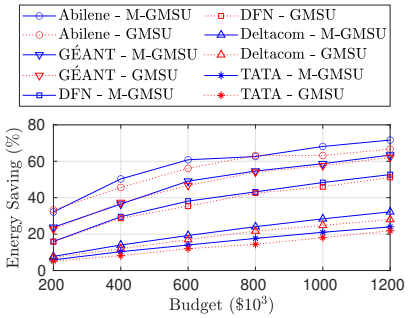


Fig. 3: M-GMSU vs GMSU [10]

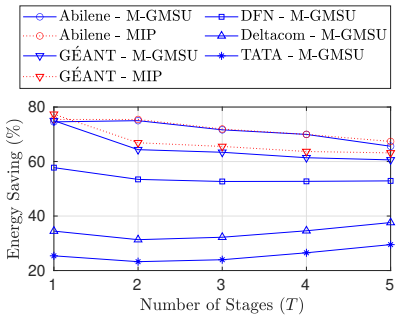


Fig. 4: ε_T for various T

TABLE I: Running time (in CPU seconds)

Name	$\|V\|$	$\|E\|$	$\|D\|$	Running Time	
				M-GMSU	MIP
Abilene	12	30	132	0.14	0.6
GÉANT	23	74	466	1.07	92100.16
DFN	58	174	3306	33.01	N/A
Deltacom	113	322	12656	398.75	N/A
TATA	145	372	20880	818.777	N/A

Fig. 3 compares ε_T produced by M-GMSU against GMSU [10] that uses single path routing. Fig. 3 shows that M-GMSU produces, on average, 4.44%, 1.92%, 3.38%, 11.71%, and 14.85% higher ε_T than GMSU for Abilene, GÉANT, DFN, Deltacom and TATA, respectively. The results show the benefit of using multiple path routing to maximize ε_T.

C. Effect of Increasing Stages

Next, we investigate how the number of stages T impact energy saving. We consider one to five stages and its impact on ε_T. The budget B is \$1.2M. As shown in Fig. 4, the energy saving ε_T for Abilene, GÉANT, and DFN decreases as T increases. For example, the energy saving ε_T produced by M-GMSU for Abilene (GÉANT) decreases from 74.56% to 65.61% (75% to 60.61%) when T increases from one to five. Notice that for Abilene (GÉANT), M-GMSU produces ε_T that is only 0.95% (3.55%) off from the optimal ε_T obtained by MIP. In contrast, energy saving ε_T for Deltacom (TATA) increases from 34.47% to 37.61% (25.4% to 29.52%) when T increases from one to five. For these two larger networks, there are more switches to upgrade in later stages. This results in larger ε_T values. In contrast, for smaller networks, e.g., Abilene, budget B = \$1.2M can upgrade a larger percentage of switches in earlier stages. Thus, there are fewer number of switches in later stages with unused cables that can be turned off. Further, the growing traffic volume can result in remaining switches having smaller number of *off*-cables and thus, upgrading them does not significantly increase ε_T.

VI. Conclusion

This paper considers the problem of upgrading a legacy network that supports OSPF-ECMP protocol into a SDN over multiple stages. A key aim is that an upgraded network must maximize energy saving. To do so, we consider the maximum available budget at each stage, MLU, maximum path delay, and each l-switch must comply with the OSPF-ECMP protocol. We formulated an MIP and proposed a heuristic solution called M-GMSU. Our simulations show that M-GMSU requires significantly less CPU time than MIP. Further, it obtains ES that is only up to 3.55% off from the optimal ES obtained by MIP. We find that increasing budget and number of stages result in larger ES. M-GMSU results in up to 14.85% higher ES than an existing technique, called GMSU, that uses single path routing.

References

[1] S. Saraswat, V. Agarwal, H. P. Gupta, R. Mishra, A. Gupta, and T. Dutta, "Challenges and solutions in software defined networking: A survey," *Journal of Netw. and Comp. Appl.*, vol. 141, pp. 23–58, 2019.

[2] Y. Guo, Z. Wang, Z. Liu, X. Yin, X. Shi, J. Wu, Y. Xu, and H. J. Chao, "SOTE: Traffic engineering in hybrid software defined networks," *Computer Networks*, vol. 154, pp. 60–72, 2019.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] N. Huin, M. Rifai, F. Giroire, D. L. Pacheco, G. Urvoy-Keller, and J. Moulierac, "Bringing energy aware routing closer to reality with SDN hybrid networks," *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 4, pp. 1128 – 1139, Dec. 2018.

[9] 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.

[10] L. Hiryanto, S. Soh, K.-W. Chin, and M. Lazarescu, "Green multi-stage upgrade for bundled-link sdns with budget constraint," in *Proc. 29th ITNAC*, Auckland, New Zealand, Nov. 2019, pp. 1–7.

[11] 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.

[12] J. Y. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, Jul. 1971.

[13] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, "Inferring link weights using end-to-end measurements," in *2nd ACM SIGCOMM Workshop on Internet measurement*, 2002, pp. 231–236.

[14] M. Roughan, "Simplifying the synthesis of internet traffic matrices," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 5, pp. 93–96, Oct. 2005.