IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 14, 2021, accepted June 23, 2021, date of publication July 1, 2021, date of current version July 19, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3093899

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

LELY HIRYANTO[1], (Member, IEEE), SIETENG SOH[1], (Member, IEEE), KWAN-WU CHIN[2], DUC-SON PHAM[1], (Senior Member, IEEE), AND MIHAI M. LAZARESCU[1], (Member, IEEE)

[1]School of Electrical Engineering, Computing, and Mathematical Sciences, Curtin University, Perth, WA 6102, Australia
[2]School of Electrical, Computer, and Telecommunications Engineering, University of Wollongong, Wollongong, NSW 2500, Australia

Corresponding author: Lely Hiryanto (lely.hiryanto@postgrad.curtin.edu.au)

This work was supported by the Australian Government through the Department of Foreign Affairs and Trade.

ABSTRACT This paper considers the novel problem of upgrading a *legacy* network into a Software Defined Network (SDN) over multiple stages and saving energy in the upgraded network, or hybrid SDN. That is, in each stage, the problem at hand is to select and replace *legacy* switches with SDN switches and reroute traffic to power off as many unused cables as possible to save energy. Also, the operator must consider: (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) the Open Shortest Path First - Equal Cost Multi-Path protocol. This paper addresses two multi-path routing scenarios: 1) non-link-disjoint and 2) link-disjoint. It outlines a Mixed Integer Program and a heuristic algorithm for each scenario. The experimental results show that: (i) both solutions produce only up to 0.63% higher energy saving in scenario-1 than in scenario-2, (ii) the mixed integer program (heuristic algorithm) for both scenarios give an energy saving up to 71.93% (71.64%), (iii) using a larger budget and/or number of stages can increase the energy saving, and (iv) the saving achieved by the heuristic solution for each scenario is within 4% from the optimal saving.

INDEX TERMS Network planning, IEEE 802.1ax, IEEE 802.3az, multi-stage upgrade, multi-path routing, link-disjoint multi-path routing.

I. INTRODUCTION

ASoftware Defined Network (SDN) offers operators a new network management paradigm [1]. It consists of a set of SDN-switches or *s*-switches and one or more controllers [1]. A controller provides a global view of a network. It helps an operator optimizes network performance such as the maximum link utilization (MLU) [2] and/or energy saving [3]. Consequently, network operators are keen to upgrade their *legacy* networks to SDNs. To do so, they must consider their available budget, advances in SDN equipment and cost reduction or depreciation of network equipment over time. Hence, *legacy* switches or *l*-switches are likely to be upgraded over multiple stages, creating so called *hybrid*-SDNs, which contain *l*-switches along with *s*-switches.

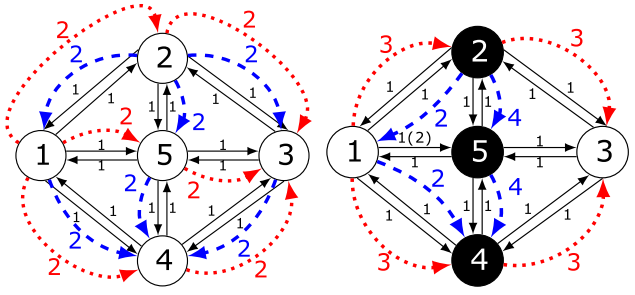
Another recent consideration is energy efficiency. It is well-known that the current networks are overprovisioned, e.g., link bandwidth, which satisfies traffic demands during peak hours but is underutilized during off-peak periods [4]. To this end, backbone networks now utilize IEEE 802.1AX [5], a *bundled*-link technology where logical links consist of multiple physical cables. 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 their energy cost. For example, the work in [4] and [6] aimed to switch off as many cables as possible and reroute traffic flows to the cables from other paths. They considered *multi-path routing* using Multi-Protocol Label Switching (MPLS). On the other hand, the work in [7] considered the Open Shortest Path First - Equal Cost Multi-Path (OSPF-ECMP) to maximize energy saving. Further, multi-paths that do not share a common link, called link-disjoint paths, are used to provide path resiliency against link failures [8]. Reference [9] showed how to save energy in *legacy* networks while maintaining link-disjoint multi-paths. Multi-path routing is ideal for use in SDNs, e.g., [2] and [3] because an SDN controller allows: (i) *s*-switches to

The associate editor coordinating the review of this manuscript and approving it for publication was Peng-Yong Kong.

use non-shortest paths and (ii) each source *s*-switch to split *unequal* amount of traffic onto each path.

Henceforth, this paper considers a novel problem in network upgrade. 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 allows each *s*-switch to turn off the maximum number of *unused* cables. The upgrade maintains the same routing service to users. More specifically, if a traffic demand in a legacy network is routed via link-disjoint paths, the demand must be routed via at least two link-disjoint paths after network upgrade. Otherwise, the demand can be routed via multi-paths that can share common link(s), called *non*-link-disjoint paths, or even a single path. The upgrade is also subjected to the following constraints: (i) active cables must have sufficient capacity to carry 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 OSPF-ECMP. In addition, the solution must consider increasing traffic volume and decreasing switch upgrade cost over multiple stages.

To illustrate our problem, consider Figure 1a. Each link has the indicated cost and two cables; each cable has a capacity of five units of data and a MLU of 100%. Assume the traffic demand from node 1 to 3 is six units, which we denote as $(1 \rightarrow 3, 6)$. There is also another traffic demand $(2 \rightarrow 4, 6)$. As shown in Figure 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. Assume that an unused cable can be switched off only if at least one of its end nodes is a *s*-switch. In this case, there is no energy saving.



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

FIGURE 1. An illustration (a) with equal distribution of traffic flow over equal-cost multiple paths (OSPF-ECMP), and (b) with link cost adjustment and each source *s*-switch can split unequal amount of traffic. Nodes ○ and • represent an *l*-switch and *s*-switch, respectively. The number next to each link denotes its cost. Lines ------ and - - - denote paths for demand $(1 \rightarrow 3, 6)$ and $(2 \rightarrow 4, 6)$, with the number next to each line indicates the traffic volume.

Now consider a scenario where the 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 in the set $\{1, 2, 3\}$ and the same traffic split and routing as in Figure 1a. This allows us to turn off 19 unused cables: one cable in links $\{(1, 2), (2, 1), (2, 3), (2, 5), (1, 5), (5, 3), (1, 4), (3, 4), (4, 3)\}$, and two cables in links $\{(3, 2), (5, 2), (5, 1), (3, 5), (4, 1)\}$. This leads to an energy saving of $19/32 \times 100\% = 59.38\%$. As another example, consider upgrading *l*-switches in $\{2, 4, 5\}$; see Figure 1b. We see *s*-switch-1 splitting 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 so that path $(1, 5, 3)$ is no longer the shortest path for demand $(1 \rightarrow 3, 6)$; see the link cost in a bracket. On the other hand, *s*-switch-2 splits demand $(2 \rightarrow 4, 6)$ onto shortest paths $(2, 1, 4)$ and $(2, 5, 4)$ with *unequal* flow sizes of two and four, respectively. Note that the shortest path $(2, 3, 4)$ is not used so that one cable of link $(3, 4)$ can be switched off. Thus, *s*-switch-4 and *s*-switch-5 can now turn off six more cables, i.e., one additional cable on links $(1, 5)$, $(5, 3)$, $(5, 4)$, $(3, 4)$ and two more cables on link $(4, 5)$, which yield a higher energy saving of $(6 + 19)/32 \times 100\% = 78.12\%$. After the legacy network in Figure 1a is upgraded to a *hybrid* SDN in Figure 1b, both demands are routed via link-disjoint paths.

Given the above research aim, the main *contributions* of this paper are as follows:

- It presents a novel problem to maximize energy saving in a *hybrid*-SDN. It consists of two sub-problems: i) multi-stage *l*-switch upgrade, and (ii) splitting traffic optimally via *s*-switches and setting link cost to ensure that each *l*-switch complies with OSPF-ECMP.
- It contains a novel Mixed Integer Program (MIP) that can be used to compute the optimal solution for small-size networks. The MIP considers two multi-path routing scenarios: 1) non-link-disjoint paths, and 2) link-disjoint paths. The paper also presents an analysis of the complexity of MIP and its NP-Hardness. Note that our solution for routing scenario-1 can be used to upgrade a legacy network where users do not require link-disjoint paths. Further, it produces an upper bound on energy saving for scenario-2.
- It proposes a heuristic algorithm that can be used in large-scale networks for each of the aforementioned routing scenarios. It also outlines the time complexity of the algorithm as well as a proof of correctness.

Next, Section II discusses existing works on minimizing energy expenditure and those that carry out multi-stage upgrade of SDNs. Section III presents our network model, notations and MIP. Section IV describes our proposed heuristic solution. Section V outlines our results. Finally, Section VI concludes the paper and provides future research directions.

II. RELATED WORK

Works on *green routing* aim to reroute traffic for utilizing the minimal number of network components, e.g., line cards or links and switches. The unused components are then powered off [10]. For example, the efforts in [11] and [12] introduced energy-aware routing via single path routing with MLU constraint. References [9] and [7] maximized the energy saving of legacy networks with non-bundled links

by respectively employing MPLS and OSPF-ECMP based multi-path routing that satisfies MLU. The authors of [13] designed an energy-efficient bundled-link with two types of cables. Namely, cables with different energy levels and cables with sleep mode. On the other hand, the work in [14] considered traffic load distribution among IEEE 802.3az cables in a bundled-link to minimize their usage. Other works, such as [4] and [6], aimed to maximize energy saving in backbone networks that support *bundled* links and MPLS. The work in [4] considered the power consumption of all links and *l*-switches is independent of traffic load. On the other hand, the authors of [6] assumed each link and *l*-switch have different power usage. Further, they considered routing over multi-paths, delay tolerance, and MLU.

There are many works on improving the energy efficiency of SDNs. For example, the work in [15] and [16] powered down *unused links* in *pure* SDNs that only have *s*-switches. Both works considered a single communication path bounded by MLU and path delay for each pair of *s*-switches and from each *s*-switch to its associated controller. Many works have considered an incremental upgrade strategy; e.g., the authors of [17] considered *hybrid* SDNs with *s*-switches and *l*-switches. Moreover, these works consider co-existence between an SDN controller that programs *s*-switches and legacy routing protocols, such as OSPF and MPLS. To date, research into *hybrid* SDNs, e.g., [2], [3], [18]–[22] and [23], assumed the SDN controller has access to all required network information, including those from *l*-switches. Our work follows the same assumption, where the placement of multiple SDN controllers is deferred to future work.

A *hybrid* SDN can be formed by incrementally upgrading *l*-switches with *s*-switches [17]. The upgrades are performed over one stage [2], [3], [18]–[20] or multi-stages [21]–[23]. Reference [2] have used a greedy algorithm to upgrade a set of *l*-switches with the highest total traffic load on their outgoing links. The authors considered multi-path routing to minimize MLU. In [3], *s*-switches are randomly and uniformly distributed in a *hybrid* SDN. Each *s*-switch can split traffic to maximize energy saving; however, each *l*-switch uses OSPF to compute a single shortest path. The work in [18] used a given set of partially deployed *s*-switches to minimize the power usage of both *s*-switches and their adjacent links. Another work in [19] considered traffic routing via single path to minimize the power consumption of *s*-switches and links that are adjacent to the *s*-switches. The authors first select a set of *l*-switches based on different criteria, e.g., in decreasing order of their number of *l*-links, before performing traffic routing. On the other hand, the authors of [20] jointly addressed the problems of upgrading up to m *l*-switches and traffic routing to minimize the power usage. They assume OSPF routing for all *l*-switches and single path routing for all *s*-switches. Similar to [20], we jointly optimize the upgraded *l*-switches and traffic routing for maximizing the number of unused cables.

An operator incurs less risk in terms of performance and security degradation if a network is upgraded over multiple stages [22]. To this end, given a total budget (in \$), the work in [21] and [22] aimed to upgrade *l*-switches in order to maximize the number of paths available to *s*-switches over T stages. Moreover, the authors of [21] considered a fixed upgrade cost (in \$) for each *l*-switch. In contrast, Poularkis *et al.* [22] consider an upgrade cost that decreases over time and assume that traffic size (in bytes) increases over multiple stages. In addition, the authors of [22] aimed to maximize traffic controllability, i.e., traffic flows that passes through at least one *s*-switch.

Recently, the work in [23] addressed a multi-stage SDN deployment problem. Its goal is to maximize energy saving by shutting down as many *unused cables* in each link as possible. The authors of [23] considered: (i) decreasing switch upgrade cost and increasing traffic volume over time, (ii) using a maximum budget at each stage, (iii) satisfying MLU, and (iv) ensuring the upgraded network must be able to support existing flows. Their work ensured each flow is routed via a single path with longer delay but does not exceed the given delay constraint. They proposed an Integer Linear Program (ILP) formulation to solve the problem for small networks and a heuristic algorithm called GMSU that can be used for larger networks. In contrast, our recent work in [24] considered two types of multi-path routing for each demand: (i) those that traverse only *l*-switches and (ii) those that traverse at least one *s*-switch. For type (i), the traffic flow of each demand is routed using OSPF-ECMP, i.e., each *l*-switch splits a flow equally over multiple shortest paths. In contrast, for type (ii), the traffic flow of each demand can be split *unequally* over multi-paths that are not necessarily the shortest paths. Moreover, the work addressed two main challenges to maximally switch-off unused cables, i.e., for (i), link costs may need to be adjusted to ensure each *l*-switch *complies* with OSPF-ECMP and, for (ii), each *s*-switch needs to *optimally* split traffic among its selected multi-paths.

We summarize the differences between this paper and our previous work [24] as follows:

- This paper considers two alternative routing scenarios: 1) multi-path routing as in [24], and 2) link-disjoint multi-path routing, where the selected paths for each demand have no common link. In the case where some demands have no link-disjoint paths, the demand is routed as per scenario-1.
- This paper proposes an alternative MIP as well as a heuristic algorithm to implement scenario-2 and their simulation results.
- This paper provides a qualitative analysis of the proposed MIP and heuristic solution.
- This paper discusses the effect of our solutions in terms of traffic controllability [22].

III. PRELIMINARIES

Section III-A first describes the network model. Table 1 summarizes our notations. Section III-B presents a mathematical

TABLE 1. Notations and definitions.

Notation	*Definition*
$G^0(V,E)$	A *legacy* SDN with $\|V\|$ nodes and $\|E\|$ links.
$G^t(V,E)$	A *hybrid* SDN with $\|V\|$ nodes and $\|E\|$ links at stage t.
$V^t(V^T)$	The set of upgraded l-switches at stage t (over T stages).
T	The total number of upgrade stages.
B	The maximum available budget over T time stages.
$B^t(\Delta B^t)$	The maximum (unused) budget at stage t.
p_v^t	The cost to upgrade node $v \in V$ at stage t.
$c_{uv}(c_{uv}^t)$	The capacity of link (u,v) in total (at stage t).
$\pi_{uv}(b_{uv})$	The delay (number of cables) of link (u,v).
$\rho\,(\mu_d)$	The decrease (increase) rate of a switch's upgrade cost (demand d's size) at each successive stage.
$D^t\,(D^0)$	A set of demands in $G^t(V,E)$ $(G^0(V,E))$; $\|D^t\| = \|D^0\|$.
(s_d,τ_d,ω_d^t)	A demand from s_d to τ_d with size ω_d^t; $d = [1,\|D^T\|]$.
$P_d^{s_d}$	A set of paths in $G^0(V,E)$ from s_d to τ_d.
$y_{d,uv,i}^t$	An indicator of whether link (u,v) is on path $P_{d,i}^{s_d}$.
$\delta_{\min,d}^{s_d}$	The minimum delay among all paths in $P_d^{s_d}$.
$\delta_{\max,d}$	Delay tolerance; $\delta_{\max,d} = \lceil \sigma \times \delta_{\min,d}^{s_d} \rceil$, $\sigma = [1.0, 2.0]$.
$\mathcal{P}_d^{s_d}$	A set of paths in $P_d^{s_d}$ with delay within $\delta_{\max,d}$.
ψ_{uv}^t	The cost of link (u,v) at stage t.
$R_d^{s_d,0}$	A set of shortest paths in $\mathcal{P}_d^{s_d}$.
$U_{\max}$	The threshold of maximum link utilization.
$f_{d,i}^t$	The traffic size carried by $\mathcal{P}_{d,i}^{s_d} \in \mathcal{P}_d^{s_d}$ at stage t.
n_{uv}^t	The number of powered-on cables in link (u,v) at stage t.
x_v^t	An indicator of whether switch v is upgraded at stage t.
$\varepsilon^t\,(\varepsilon_T)$	Energy saving at stage t (average energy saving over T stages).
$o_{u,a}^t$	The traffic size carried by a shortest path from l-switch u to destination a.
$z_{a,uv}^t$	An indicator of whether link (u,v) is on any shortest path to destination a at stage t.
$h_{v,a}^t$	The path cost from switch v to switch a at stage t.
I	A non-negative integer number not exceeding $2^{16}-1$.

model of the optimization problem. Finally, Section III-C analyzes the problem complexity.

A. NETWORK MODEL

Let $G^0(V,E)$ be a *legacy* network with $|V|$ nodes or l-switches and $|E|$ directed links. Each link $(u,v) \in E$ has a *bundle size* with b_{uv} cables and a propagation delay of π_{uv} (in seconds). The capacity of each cable is γ (in bytes). Thus, the link capacity is $c_{uv} = b_{uv} \times \gamma$ (in bytes).

Let $T \geq 1$ be the given planning horizon. The duration of each stage $t \leq T$ is determined by the lifetime of network devices, e.g., three to five years [22]. 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 s-switch is a hybrid switch; an example is the OpenFlow-hybrid switch in [25], which supports both OpenFlow and normal Ethernet switching operation. 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 [18], [19], and [20], only cables in a c-link are powered off when they have no traffic. Also, every cable of each c-link runs IEEE 802.3az [26], meaning it can be placed in either active or *sleep* state. Without loss of generality, this paper assumes each l-switch does not turn off unused cables, i.e., the switch does not comply with the IEEE 802.3az [26] standard.

Let B be the total budget (in \$) over time T and $B^t \leq B$ denotes the maximum budget at stage t. The total cost to upgrade l-switches in V^t cannot exceed the budget B^t. Any unused budget in stage 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 (in \$) be the cost of upgrading switch v in stage t. The upgrade cost of a switch may vary over time depending on its model and type, e.g., *edge* or *core* switch [22]. We use ρ to denote the depreciation rate in switch upgrade cost, 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.

Let $D^t = \{(s_d, \tau_d, \omega_d^t) \mid \forall d \in [1, |D^t|]\}$ denote a set of traffic demands in $G^t(V,E)$. Node $s_d \in V$ and $\tau_d \in V$, respectively, represent the source and destination of each demand $d \in [1, |D^t|]$. Demand d has a traffic volume $\omega_d^t > 0$ (in bytes). 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 $d \in [1, |D^0|]$. The traffic volume for each demand d increases with each successive stage with rate $\mu_d \in [0,1]$. Thus, we have $\omega_d^t = \omega_d^0 \times (1+\mu_d)^{t-1}$. 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 $d = [1, |D^0|]$, let $P_d^x = \{P_{d,i}^x \mid \forall x \in V,\ x \neq \tau_d, \forall i \in [1, |P_d^x|]\}$ 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 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 $\delta_{d,i}^x$ (in seconds), is computed as the sum of propagation delays over all links in the path, i.e., $\delta_{d,i}^x = \sum_{(u,v)\in P_{d,i}^x} \pi_{uv}$. We assume that the propagation delay π_{uv} of link (u,v) is proportional to the distance between node u and v [27]. Let $\delta_{\min,d}^x$ ($\delta_{\max,d}^x$) be the minimum (maximum) delay among all paths in P_d^x. We allow users to use paths that are up to $(\sigma-1)\times 100\%$ longer than their original delays, i.e., $\delta_{\max,d} = \lceil \sigma \times \delta_{\min,d}^{s_d} \rceil$, for a delay multiplier $\sigma = [1.0, 2.0]$. Let $\mathcal{P}_d^{s_d} \subset P_d^{s_d}$ denote a set of paths in $P_d^{s_d}$ that satisfy delay constraint $\delta_{\max,d}$. One can use Yen's algorithm [28] to generate set $\mathcal{P}_d^{s_d}$ for each demand d.

Let $\mathrm{I} = 2^{16-1}$ represent the maximum OSPF cost that can be assigned to each link (u,v) [29]. Here, each link (u,v) in stage t has cost $\psi_{uv}^t \in [1, \mathrm{I}]$. Let $\psi^t = \{\psi_{uv}^t \mid \forall (u,v) \in E, \forall t \in [1,T]\}$ denote a set of link costs for stage t. Thus, ψ^0 denotes the set of initial link costs. The cost of each path $P_{d,i}^x \in P_d^x$ in stage t, denoted by $\Psi_{d,i}^{x,t}$, is computed as the sum of link cost in ψ^t over all links on the path from node x to destination τ_d, i.e., $\Psi_{d,i}^{x,t} = \sum_{(u,v)\in P_{d,i}^x} \psi_{uv}^t$. Let $\Psi_{\min,d}^{x,t}$ denote the minimum cost of all paths in P_d^x at stage t. A path $P_{d,i}^x \in P_d^x$ is called the *shortest path* if its cost is equal to the minimum cost, i.e., $\Psi_{d,i}^{x,t} = \Psi_{\min,d}^{x,t}$. Let R^0 denote the set of shortest path(s) for all demands in $G^0(V,E)$ and $R_d^{s_d,0} \in R^0$ be a set of shortest paths of demand d. Note that the shortest paths in $G^0(V,E)$ have the shortest delay, and thus we have $R_d^{s_d,0} \subseteq \mathcal{P}_d^{s_d}$.

Let $f_{d,uv}^t \leq \omega_d^t$ denote the flow of demand d along link (u,v) at stage t. We have $f_{d,uv}^t > 0$ and $f_{d,vu}^t = 0$ ($f_{d,uv}^t = 0$ and $f_{d,vu}^t > 0$) if demand d flows from nodes u to v (v to u). Let $f_{d,i}^t = f_{d,uv}^t$ denote the flow size or volume of demand

d carried by path $P^x_{d,i} \in P^x_d$ with $(u, v) \in P^x_{d,i}$ and $x = s_d$ for every stage $t \in [1, T]$. We use f^t_{uv} to denote the traffic flow of link (u, v) at stage t, i.e., $f^t_{uv} = \sum_{d\in[1,|D^t|]} f^t_{d,uv}$. 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* cables or *on*-cables to carry traffic f^t_{uv}. Thus, the maximum capacity of link (u, v) at stage t is $c^t_{uv} = (n^t_{uv}/b_{uv}) \times U_{\max} \times c_{uv}$.

Finally, unused or idle cables are switched off by powering off their line card to save energy. Specifically, a cable can be powered-off, called *off*-cable, if it is connected to at least one s-switch, i.e., a cable of a c-link. Note that line cards consume a significant fraction of a router's energy consumption [4]. Thus, without loss of generality, we assume a cable's energy consumption is equivalent to that of its line card. Also, similar to the energy saving model of [4], each cable with traffic consumes the same amount of energy. For example, an *on*-cable with 1% load and another with 100% load consume the same amount of energy. Note that in practice, each port of energy-efficient switches continues to consume the maximum power even with 10% traffic load [30]. Let ε^t be the energy saving in stage t. Formally, it is computed as

$$\varepsilon^t = \frac{\sum_{(u,v)\in E} (b_{uv} - n^t_{uv})}{\sum_{(u,v)\in E} b_{uv}}. \tag{1}$$

In words, the energy saving ε^t is a ratio between the total number of *off*-cables and the total number of cables in the network. 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 denotes the average energy saving over T stages, i.e., $\varepsilon_T = \frac{1}{T}\sum_{t=1}^{T} \varepsilon^t$. Table 1 summarizes the notations used in this paper.

B. MATHEMATICAL MODEL

We formulate our problem as a Mixed Integer Program (MIP). We consider two routing scenarios: 1) multi-path routing: the traffic of a demand d is split onto multi-paths and these paths can have common link(s), and 2) link-disjoint path routing: the selected paths of a demand d must not have any common link(s). First, we outline the MIP for scenario-1, see (2b), before outlining MIP (2b) for scenario-2.

1) SCENARIO-1: MULTI-PATH ROUTING

Our MIP, see (2a), aims to minimize the number of *on*-cables over T stages. Constraint (2c) conserves flows and ensures there is at least one path connecting source s_d to destination τ_d. Constraint (2d) ensures the traffic volume $f^t_{d,i}$ of each selected path $i \in [1, |\mathcal{P}^{s_d}_d|]$ of demand d sums to ω^t_d. Constraints (2e) and (2f) respectively 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. Constraint (2g) limits the number of *on*-cables to the bundle size of each link.

In constraint (2h), variable x^t_u is an indicator of whether l-switch u is upgraded at stage t. This constraint ensures each switch is upgraded only once. Constraint (2i) ensures the total upgrade cost at each stage is less than or equal to $B^t = B/T + \Delta B^{t-1}$, while constraint (2j) enforces all cables of l-links are powered on. Note that only cables in c-links can be turned off.

Let $z^t_{a,uv}$ indicate whether link (u, v) at stage t is on the shortest path from node u to a. Further, let $h^t_{u,a}$ denote the path cost from u to a. Constraints (2k)-(2m) 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 (2m) defines the domain of all decision variables.

$$\min \sum_{t=1}^{T} \sum_{(u,v)\in E} n^t_{uv} \tag{2a}$$

$$\text{s.t.} \sum_{(u,v)\in E} y^t_{d,uv,i} - \sum_{(v,u)\in E} y^t_{d,vu,i} = \begin{cases} 1, & u = s_d \\ -1, & u = \tau_d \\ 0, & u \ne s_d, \tau_d \end{cases}, \tag{2b}$$

$$\sum_{i=1}^{|\mathcal{P}^{s_d}_d|} f^t_{d,i} = \omega^t_d, \tag{2c}$$

$$\sum_{(u,v)\in E} (y^t_{d,uv,i} \times \pi_{uv}) \le \delta_{\max,d}, \tag{2d}$$

$$\sum_{d=1}^{|D^t|} \sum_{i=1}^{|\mathcal{P}^{s_d}_d|} (y^t_{d,uv,i} \times f^t_{d,i}) \le (n^t_{uv}/b_{uv}) \times U_{\max} \times c_{uv}, \tag{2e}$$

$$0 \le n^t_{uv} \le b_{uv}, \tag{2f}$$

$$\sum_{t=1}^{T} x^t_u \le 1, \tag{2g}$$

$$\sum_{v\in V} (p^t_v \times x^t_v) \le \sum_{k=1}^{t} B^k - \sum_{k=1}^{t-1} \sum_{v\in V} (p^k_v \times x^k_v), \tag{2h}$$

$$n^t_{uv} = \max\left\{ n^t_{uv}, b_{uv} \times \left(1 - \sum_{k=1}^{t} x^k_u - \sum_{k=1}^{t} x^k_v\right)\right\}, \tag{2i}$$

$$\sum_{d=1,\tau_d=a}^{|D^t|} \sum_{i=1}^{|\mathcal{P}^{s_d}_d|} y^t_{d,uv,i} \times f^t_{d,i} \le z^t_{a,uv} \times \sum_{d=1,\tau_d=a}^{|D^t|} \omega^t_d, \tag{2j}$$

$$0 \le o^t_{u,a} - \sum_{d=1,\tau_d=a}^{|D^t|} \sum_{i=1}^{|\mathcal{P}^{s_d}_d|} y^t_{d,uv,i} \times f^t_{d,i} \le (1 - z^t_{a,uv}) \times \sum_{d=1,\tau_d=a}^{|D^t|} \omega^t_d, \tag{2k}$$

$$(1 - z^t_{a,uv}) \le h^t_{v,a} + \psi^t_{uv} - h^t_{u,a} \le (1 - z^t_{a,uv}) \times \mathrm{I}, \tag{2l}$$

$$y^t_{d,uv,i}, x^t_u, z^t_{a,uv} \in \{0, 1\};\ f^t_{d,i}, o^t_{u,a}, h^t_{u,a} \ge 0. \tag{2m}$$

Except (2h), all constraints in MIP (2b) are for each stage $t \in [1, T]$. Constraint (2c) is for each node $u \in V$, traffic demand $d \in [1, |D^t|]$, and path $i \in [1, |\mathcal{P}^{s_d}_d|]$. Constraint (2d)

applies to each demand $d \in [1, |D^t|]$, while (2e) considers all demands and $|\mathcal{P}_d^{s_d}|$ paths for each demand. Constraint (2f), (2g) and (2j) exist for all links $(u, v) \in E$, while constraint (2h) applies to each $u \in V$. Finally, constraints (2k) - (2m) are evaluated for every destination $a \in V$ and each link $(u, v) \in E$, with a starting node $u \in V$ is a l-switch, i.e., $x_u^t = 0$.

2) SCENARIO-2: LINK-DISJOINT PATH ROUTING

We show how to revise MIP (2b) to support link-disjoint path routing; we call the revised MIP as DP-MIP. More specifically, if traffic demand d in legacy network $G^0(V, E)$ is routed via two or more link-disjoint shortest paths, i.e., $|R_d^{s_d,0}| > 1$, DP-MIP must route the demand via at least two link-disjoint paths in $\mathcal{P}_d^{s_d}$. Otherwise, DP-MIP can route the demand via any one or more paths in $\mathcal{P}_d^{s_d}$, which is a set of paths from s_d to τ_d each of which has delay within $\delta_{\max,d}$.

Let $y_{d,i}^t$ be an indicator of whether the path $\mathcal{P}_{d,i}^{s_d} \in \mathcal{P}_d^{s_d}$ is selected to route demand d at stage t. Let l_d be another indicator which is set to 1 if the shortest paths in $R_d^{s_d,0}$ are link-disjoint and $|R_d^{s_d,0}| > 1$. On the other hand, if $R_d^{s_d,0}$ contains either one path or non link-disjoint multi-paths, l_d is set to zero. DP-MIP uses all constraints of MIP (2b) and includes the following three constraints:

$$y_{d,i}^t \le f_{d,i}^t \le y_{d,i}^t \times \omega_d^t, \quad \text{(2n)}$$

$$\sum_{i=1}^{|\mathcal{P}_d^{s_d}|} y_{d,i}^t \ge l_d + 1, \quad \text{(2p)}$$

$$y_{d,i}^t \times y_{d,uv,i}^t + y_{d,j}^t \times y_{d,uv,j}^t \le (1 - l_d) + 1. \quad \text{(2q)}$$

Constraint (2n) sets $y_{d,i}^t = 1$ if path $\mathcal{P}_{d,i}^{s_d}$ is able to carry the traffic of demand d, i.e., it has $f_{d,i}^t > 0$. Otherwise, both constraints set $y_{d,i}^t = f_{d,i}^t = 0$, which indicates that path $\mathcal{P}_{d,i}^{s_d}$ does not carry traffic. For every $l_d = 1$, constraint (2p) guarantees at least two paths in $\mathcal{P}_d^{s_d}$ are selected to route demand d. Then, constraint (2q) ensures that every pair of selected paths are link-disjoint. In this case, constraint (2q) evaluates every link $(u, v) \in E$ to ensure that link (u, v) is not simultaneously used by both paths $\mathcal{P}_{d,i}^{s_d}$ and $\mathcal{P}_{d,j}^{s_d}$, i.e., both $y_{d,uv,i}^t$ and $y_{d,uv,j}^t$ cannot be equal to one. For each $l_d = 0$, constraints (2p) and (2q) ensure that there is at least one selected path to route demand d. For this case, if there is more than one selected path, they are not necessarily link-disjoint. Note that constraints (2n) to (2q) apply to every demand $d \in [1, |D^t|]$ at each stage $t \in [1, T]$.

C. PROBLEM COMPLEXITY

Our problem is related to two NP-hard problems: (i) OSPF cost setting problem [29]: given a network $G(V, E)$, maximum link utilization for each link $(u, v) \in E$, and a set of traffic demands, assign an integer cost for each link to optimize a given network performance metric, e.g., network delay; and (ii) 0-1 Multiple Knapsack Problem (MKP) [31]: given m items, each of which has a profit and weight, and T knapsacks, each of which has a maximum weight capacity, select T-disjoint subsets of items that maximize the total profit, subject to each subset having a total weight no more than its knapsack's capacity.

With respect to problem (i), the network performance of interest is the minimum number of *on*-cables that have sufficient capacity to carry traffic demands. The cost assigned to each link is used to calculate the shortest path from each l-switch to any destination τ_d in D^t. These shortest paths define the total traffic volume on each link, which then determine the number of *on*-cables. Thus, our problem is at least as hard as the problem in (i).

Our problem can be reduced to MKP when (a) there is no depreciation in switch upgrade cost, and (b) the number of *on*-cables per link $(u, v) \in E$ is known, i.e., the traffic splits and shortest paths used to carry traffic flows of demand $d \in [1, |D^t|]$ are fixed at each stage t. Note that the profit and weight of each item in MKP are respectively equivalent to the number of *off*-cables for each switch $v \in V$ and the switch upgrade cost p_v^t. Further, the maximum budget at each stage B^t is the same as a knapsack's capacity in MKP. Further, our problem aims to upgrade T disjoint subsets of l-switches that minimize the total number of *on*-cables over multiple stages T, i.e., maximize the total number of *off*-cables instead of the total profit in the MKP. Thus, our problem is also as hard as MKP. The following section describes our heuristic solution for the optimization problem.

IV. SOLUTION

This section outlines our greedy heuristic solution called **Multi-Paths Green Multi-Stage Upgrade** (M-GMSU). Section IV-A first describes M-GMSU, where it routes each traffic demand via multi-paths that may have common link(s). Then, Section IV-B presents our approach called DP-GMSU, which uses M-GMSU but adopts link-disjoint path routing. Section IV-C gives an example. Section IV-D analyzes the correctness of M-GMSU and DP-GMSU as well as their time complexity.

A. DETAILS OF M-GMSU

One can run M-GMSU offline in a centralized server that may also act as the SDN controller. As per Algorithm 1, it consists of three phases: (1) initialize traffic routing, (2) upgrade switches, and (3) reroute traffic and set link cost. Phase 1 is used only in stage $t = 1$, while Phase 2 is at the beginning of each stage (in years). On the other hand, rerouting in Phase 3, in addition to being computed at the beginning of each stage, can be used whenever a significant change occurs in network traffic within the stage, e.g., every week. At each upgrade stage t, M-GMSU produces: (i) a set of upgraded switches V^t, (ii) a set of paths $R_d^{s_d,t}$ to route each demand d, (iii) the number of *on*-cables n_{uv}^t on each link (u, v), and (iv) energy saving ε^t.

1) PHASE 1: INITIAL ROUTING

Given a legacy network $G^0(V, E)$, Phase 1 initially routes each traffic demand according to OSPF-ECMP. For each

Algorithm 1 : M-GMSU

Input: $G^0(V, E)$, T, B, D^T, p_v^0, $U_{\max}$, μ, ρ
Output: R^t, V^t, n_{uv}^t, ε^t, ψ^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 set $\mathcal{P}_d^{s_d}$
4: Put each path $\mathcal{P}_{d,i}^{s_d} \in \mathcal{P}_d^{s_d}$ 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_u for each $u \in V$ using (3)
12: $X = V$
13: **for** ($t \in \{1, 2, \ldots, T\}$) **do**
▷ *Phase 2: upgrade switches*
14: $\{V^t, X, L, \Delta B^t\} =$ **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 using (1)
18: **end for**

link $(u, v) \in E$, Line 1 of M-GMSU sets the initial link cost, denoted as ψ_{uv}^0, to the link delay π_{uv}. For each demand $d \in [1, |D^T|]$, Line 2 uses Yen's algorithm [28] to generate set $\mathcal{P}_d^{s_d}$, which contains all paths from s_d to τ_d in order of increasing delay and within $\delta_{max,d}$. Lines 4–8 distribute the traffic volume ω_d^T equally over all shortest paths in $R_d^{s_d,0}$ and compute the total volume f_{uv}^T over each link (u, v). Line 10 calculates the number of *on*-cables n_{uv}^T for each link (u, v). At line 11, the total number of unused cables incident at node u is computed as

$$w_u = \sum_{(u,v)\in E} (b_{uv} - n_{uv}^T), \quad u \in V. \tag{3}$$

The term $(b_{uv} - n_{uv}^T)$ in (3) denotes the number of *off*-cables in each link (u, v) at the *last* stage T. Equation (3) uses $(b_{uv} - n_{uv}^T)$ to compute w_u because we observe that the largest flow for each demand occurs at stage T. Recall that the size of each traffic demand d grows at a rate of $\mu_d \geq 0$ per stage. Thus, if a link (u, v) that has n_{uv}^T number of *on*-cables can carry traffic demand at stage T, its *on*-cables can also carry traffic demands at any stage $t < T$. It implies that we have $n_{uv}^t \leq n_{uv}^{t+1}$ and $(b_{uv} - n_{uv}^t) \geq (b_{uv} - n_{uv}^{t+1})$ for each link (u, v). In this case, the $(b_{uv} - n_{uv}^t)$ number of unused cables at stage t include the $(b_{uv} - n_{uv}^T)$ number of unused cables which can remain off at the next stage $t + 1$. Thus, upgrading a set of l-switches with the highest total number of unused cables at the earliest possible stage can maximize the overall energy saving. Line 12 concludes Phase 1 by initializing X with all l-switches in V. In summary, Phase 1 produces (i) the set of alternative paths $\mathcal{P}_d^{s_d}$ and initial shortest paths $R_d^{s_d,0}$ for each demand d, (ii) total *on*-cables n_{uv}^T of each link $(u, v) \in E$ at stage T, and (iii) weight w_v for each node $v \in V$. This set of information will be used in Phase 2 and Phase 3.

2) PHASE 2: SWITCH UPGRADES

For each stage t, Phase 2 calls **Selection**(), shown as Algorithm 2, in Line 14. It generates a set V^t that contains *upgradable* l-switches, which is defined as follows.

Definition 1: A set V^t is upgradeable if (i) each switch $v \in V^t$ has a non-zero weight $w_v > 0$, and (ii) the total cost to upgrade all switches in V^t is at most B^t.

Phase 2 uses the ratio w_v/p_v^t to upgrade a switch with the maximum *off*-cables per cost unit. It starts from the largest ratio w_v/p_v^t in order to maximize the number of *off*-cables, and hence, energy saving, over T stages.

Algorithm 2 : Selection()

Input: X, B^t
Output: V^t, X, L, ΔB^t
1: **for** (each $v \in X$ that has $p_v^t \leq B^t$ **and** $w_v > 0$) **do**
2: Find switch 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: $\Delta B^t = B^t$

The details of **Selection**() are as follows. Line 1 considers only each candidate switch $v \in X$ that has (i) an upgrade cost p_v^t within budget B^t, i.e., $p_v^t \leq B^t$, and (ii) weight $w_v > 0$, i.e., switch v has cables to switch off. Among all nodes that satisfy the two criteria, Line 2 selects a node, say v, that has the largest ratio w_v/p_v^t. Line 3 removes node v from X. Line 4 includes v into the set of upgradeable nodes V^t and Line 5 computes the remaining budget B^t. For each l-switch neighbor, denoted as u, of the upgraded l-switch v, Line 7 reduces its weight w_u by the total cables to be switched off by node v. Lines 8–10 place each c-link (u, v) into the set L if some traffic demand passes the link, i.e., $n_{uv}^T > 0$. Lines 1–12 are repeated until the remaining budget B^t is not sufficient to upgrade any remaining l-switch in X, or each switch $v \in X$ has no unused cable to turn off, i.e., $w_v = 0$. Finally, Line 13 records the remaining budget B^t as ΔB^t. Line 15 of M-GMSU then adds the remaining budget ΔB^t to the budget for stage $t + 1$. In summary, function **Selection**() returns a set $V^t \subset V$ of *upgraded* l-switches, the remaining l-switches X, set L that stores each c-link (u, v) with non-zero traffic flow, and the remaining budget ΔB^t. The upgraded switches V^t are used in Phase 3 to increase the number of *off*-cables on every c-link, when possible.

3) PHASE 3: TRAFFIC REROUTING AND LINK COST SETTING

Phase 3 uses function **MGTE**() or Algorithm 3 in Line 16. The function adapts the greedy approach proposed in [4] and [6]. Specifically, **MGTE**() switches off as many c-link's cables as possible and reroutes traffic flows over these cables to other paths. It starts from the cable that has the smallest

used capacity. The rationale for this greedy approach is that such a cable has the smallest amount of traffic to be rerouted, and thus, more likely to be switched off. However, switching off a cable is feasible only if each traffic flow of demand d that passes through the cable can be rerouted via a set $R_d^{s_d,t}$ of *routable* paths defined as follows.

Definition 2: A set of paths $R_d^{s_d,t} \subseteq \mathcal{P}_d^{s_d}$ at stage t from source node s_d to destination node τ_d is routable if (i) each link $(u, v) \in R_d^{s_d,t}$ has sufficient capacity to carry the flow of demand d, and (ii) each l-switch $x \in R_d^{s_d,t}$ equally distributes each incoming traffic flow d over $m \geq 1$ shortest paths from x to destination τ_d.

Note that Definition 2 considers the largest traffic volume, i.e., the flow size ω_d^T of demand d to ensure each *routable* path can carry traffic at any stage $t \leq T$. All paths in the set $R_d^{s_d,0}$ are routable because each path is the shortest path and can carry $\omega_d^T/|R_d^{s_d,0}|$ amount of traffic. Further, the set $R_d^{s_d,t}$ can contain paths with different delays. However, the cost of all selected paths for each demand d from any l-switch in the paths must be equal. In this case, Phase 3 adjusts the cost of all links to satisfy OSPF-ECMP for each l-switch.

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** // or **RerouteDP**(.)
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_u for each $u \in X$ using (3)

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 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 subpath* 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. Let $R_d^{x,t}$ be a set of routable subpaths from l-switch x to destination τ_d, i.e., $R_d^{x,t} = \{R_{d,i}^{x,t} \subseteq R_{d,i}^{s_d,t} \mid x \in R_{d,i}^{s_d,t}, i \in [1, |R_d^{s_d,t}|]\}$. We have $|R_d^{x,t}| \leq |R_d^{s_d,t}|$ because two routable paths for a demand d, e.g., $R_{d,i}^{s_d,t}$ and $R_{d,j}^{s_d,t}$, can have the same subpath, i.e., $R_{d,i}^{x,t} = R_{d,j}^{x,t}$. For example, Figure 2 shows six paths from source $s_d = 1$ to destination $\tau_d = 11$. Assume only the following five paths are routable: (1, 2, 5, 8, 11), (1, 2, 5, 9, 11), (1, 3, 7, 9, 11), (1, 3, 6, 9, 11), and (1, 4, 6, 9, 11). Nodes 5, 3, 4 are the closest l-switches to node 1. Nodes 5 and 3 have two routable subpaths, i.e., $R_d^{3,t} = \{(3, 7, 9, 11), (3, 6, 9, 11)\}$ and $R_d^{5,t} = \{(5, 8, 11), (5, 9, 11)\}$, while node 4 has only one subpath, i.e., $R_d^{4,t} = \{(4, 6, 9, 11)\}$. Line 2 enumerates each set $R_d^{x,t}$ for every set $R_d^{s_d,t} \in R^t$. Let $\tilde{P}_d^{s_d,t} = \{\mathcal{P}_d^{s_d} - R_d^{s_d,t}\}$ 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 also enumerates a set of subpaths in $\mathcal{P}_d^x$ that are *not selected* at stage t, denoted by $\tilde{P}_d^{x,t} = \{\mathcal{P}_d^x - R_d^{x,t}\}$. For example, traffic from node 1 to 11 in Figure 2 has one path $\tilde{P}_d^{1,t} = \{(1, 4, 6, 10, 11)\}$ that is not used to route the traffic. Thus, we have one non-selected subpath, $\tilde{P}_d^{4,t} = \{(4, 6, 10, 11)\}$.

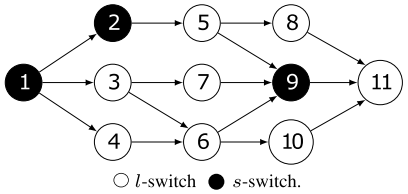


FIGURE 2. An example to generate each set of routable subpaths and a set of non-selected subpaths from source $s_d = 1$ to destination $\tau_d = 11$. Assume path (1, 4, 6, 10, 11) is not selected to route traffic from nodes 1 to 11. Nodes 3, 4 and 5 are the closest l-switches to node 1. There are three sets of routable subpaths $R_d^{3,t} = (3, 7, 9, 11), (3, 6, 9, 11)$.

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). Line 8 uses the function **Reroute**() to reroute traffic carried by path $R_{d,i}^{s_d,t} \in Q_{uv}$ via $m \geq 1$ alternative paths in set $\mathcal{P}_d^{s_d}$. Recall that criterion (ii) of Definition 2 requires each subpath $R_{d,i}^{x,t} \in R_d^{x,t}$ to carry the same traffic size. To satisfy criterion (ii), each of the m paths must carry an additional $f_{d,i}^T/m$ amount of traffic. **Reroute**() considers the following two possible cases in order to find m paths: 1) the set $R_d^{s_d,t}$ contains only $R_{d,i}^{s_d,t}$ and 2) the set $R_d^{s_d,t}$ contains $R_{d,i}^{s_d,t}$ and $m \geq 1$ paths. For case 1), **Reroute**() finds $m \geq 1$ paths in the set of non-selected paths $\tilde{P}_d^{s_d,t}$; each of which can carry an additional traffic volume of $f_{d,i}^T/m$. For case 2), the function carries out the following three steps:

(a) Use all m paths if each of the m paths is able to carry an additional traffic of size $f_{d,i}^T/m$.
(b) If step (a) fails and s_d is an s-switch, find one of the m paths which has no common node with any of the other

$m-1$ paths, i.e., a node-disjoint path that can carry traffic volume $f^T_{d,i}$. If such a path does not exist, find $k \geq 1$ path(s) in the set $\tilde{P}^{s_d,t}_d$. Here, each path must be able to carry an additional traffic of size $f^T_{d,i}/k$.

(c) If step (a) fails and s_d is an l-switch, find one path in the set $\tilde{P}^{s_d,t}_d$ to carry an additional traffic volume of $f^T_{d,i}$.

Step (b) uses a node-disjoint path to ensure that its subpath from any l-switch x to destination τ_d is the only path that carries the additional traffic of size $f^T_{d,i}$. In step (c), path $R^{s_d,t}_{d,i}$ must be rerouted to only one non-selected path $R^{s_d,t}_{d,j} \in \tilde{P}^{s_d,t}_d$ to ensure that each path in $\{R^{s_d,t}_d - R^{s_d,t}_{d,i}\}$ and path $R^{s_d,t}_{d,j}$ carry the same volume of traffic demand d. The function **Reroute**() returns false when one of the two cases fails to find m paths.

If Line 8 is able to reroute path $R^{s_d,t}_{d,i}$, i.e., **Reroute**() returns true, Line 9 reduces the used capacity r_{uv} by $f^T_{d,i}$. Further, Line 10 removes path $R^{s_d,t}_{d,i}$ and subpath $R^{x,t}_{d,i}$. It then includes the found m paths and each subpath $R^{x,t}_{d,j}$ of these m paths into the set $R^{s_d,t}_d$ and $R^{x,t}_d$, respectively. Note that updating $R^{x,t}_d$ includes adding or removing subpaths in $\tilde{P}^{x,t}_d$. If Lines 8–11 fail to reroute all paths in Q_{uv}, i.e., $r_{uv} > 0$, Line 13 sets *success* to false. Otherwise, Line 15 calls the function **LinkCost**().

LinkCost() solves the Linear Program (LP) in (4b) to adjust the link costs in ψ^t such that all subpaths in $R^{x,t}_d$ become the only shortest subpaths from node x to τ_d. It is based on the LP in [32], which aims to minimize the difference in path cost or *excess cost* for every pair of shortest subpaths in each set $R^{x,t}_d$. In this way, the total number of the shortest subpaths in every $R^{x,t}_d$ can be maximized. Briefly, the approach in [32] allocated cost $\psi^t_{uv} > 0$ to each link $(u, v) \in E$ by considering two constraints: (i) all routable subpaths in $R^{x,t}_d$ must have the same minimum cost, i.e., $\Psi^{x,t}_{d,i} = \Psi^{x,t}_{d,j}$ for all $R^{x,t}_{d,i}, R^{x,t}_{d,j} \in R^{x,t}_d$, and (ii) the cost of each routable subpath $R^{x,t}_{d,i} \in R^{x,t}_d$ is less than the cost of each non-selected subpath $\tilde{P}^{x,t}_{d,j} \in \tilde{P}^{x,t}_d$, i.e., $\Psi^{x,t}_{d,i} < \Psi^{x,t}_{d,j}$. The LP in [32] used variable $e^{x,t}_{d,i}$ to denote the *excess cost* for each routable subpath $R^{x,t}_{d,i} \in R^{x,t}_d$ to approximate the optimal link cost. Here the equality in constraint (i) becomes $\Psi^{x,t}_{d,i} - e^{x,t}_{d,i} = \Psi^{x,t}_{d,j} - e^{x,t}_{d,j}$, whilst the inequality in constraint (ii) is converted to $\Psi^{x,t}_{d,i} - e^{x,t}_{d,i} \leq \Psi^{x,t}_{d,j}$.

The LP in [32], however, cannot be applied directly to our problem for two reasons. First, constraint (ii) may have $\Psi^{x,t}_{d,i} = \Psi^{x,t}_{d,j}$ and produce $e^{x,t}_{d,i} = 0$, which makes a non-selected path $\tilde{P}^{x,t}_{d,j}$ become a shortest subpath. In contrast, our link cost setting ensures that any non-selected subpath in $\tilde{P}^{x,t}_d$ cannot be a shortest subpath. Thus, we modify constraint (ii) in [32] to $\Psi^{x,t}_{d,j} - (\Psi^{x,t}_{d,i} - e^{x,t}_{d,i}) \geq 1$ such that we have $e^{x,t}_{d,i} > 0$ when the link cost setting produces $\Psi^{x,t}_{d,i} = \Psi^{x,t}_{d,j}$. Second, the LP in [32] does not consider the maximum delay constraint for each shortest subpath. On the other hand, our proposed LP (4b) requires each shortest subpath to have delay within a given maximum delay.

To this end, **LinkCost**() is formally defined as:

$$\min \sum_{d=1}^{|D^t|} \sum_{x \in X} \sum_{R^{x,t}_{d,i} \in R^{x,t}_d} e^{x,t}_{d,i} \tag{4a}$$

$$\text{s.t.} \sum_{(u,v) \in R^{x,t}_{d,i}} \psi^t_{uv} - e^{x,t}_{d,i} = \sum_{(u,v) \in R^{x,t}_{d,i+1}} \psi^t_{uv} - e^{x,t}_{d,i+1}, \tag{4b}$$

$$\sum_{(u,v) \in \tilde{P}^x_{d,j}} \psi^t_{uv} - \sum_{(u,v) \in R^{x,t}_{d,1}} \psi^t_{uv} - e^{x,t}_{d,1} \geq 1, \tag{4c}$$

$$\psi^0_{uv} \leq \psi^t_{uv} \leq \mathrm{I}, \tag{4d}$$

$$\sum_{(u,v) \in R^{x,t}_{d,1}} \psi^t_{uv} \leq \Psi^{x,0}_{\max,d}, \tag{4e}$$

$$e^{x,t}_{d,i} \geq 0. \tag{4f}$$

Objective (4a) minimizes the total excess cost to maximize the number of paths $R^{x,t}_{d,i} \in R^{x,t}_d$ that have an excess cost $e^{x,t}_{d,i}$ of zero. For each set $R^{x,t}_d$, constraint (4c) requires each consecutive pair of subpaths, i.e., $R^{x,t}_{d,i}, R^{x,t}_{d,i+1} \in R^{x,t}_d$, to have the same minimum cost. As (4c) enforces all subpaths in $R^{x,t}_d$ to have the same cost, (4d) only needs one subpath in the set $R^{x,t}_d$, i.e., $R^{x,t}_{d,1}$, to ensure each non-selected subpath $\tilde{P}^{x,t}_{d,j}$ in $\tilde{P}^{x,t}_d$ has a larger cost than every subpath in $R^{x,t}_d$. Constraint (4e) ensures that the cost ψ^t_{uv} of each link $(u, v) \in E$ within $[\psi^0_{uv}, \mathrm{I} = 2^{16-1}]$. Let $\Psi^{x,0}_{\max,d}$ be the maximum cost among all subpaths in P^x_d at stage zero. Recall that in Phase 1, the cost of each link (u, v) is initialized as the link's delay π_{uv}. Thus, we have $\delta^x_{\max,d} \leq \Psi^{x,0}_{\max,d}$, and each subpath in P^x_d with cost no higher than $\Psi^{x,0}_{\max,d}$ has delay no longer than delay constraint $\delta^x_{\max,d}$. Constraint (4f) uses $R^{x,t}_{d,1} \in R^{x,t}_d$ to ensure that the cost of each subpath in $R^{x,t}_d$ is not higher than the maximum cost $\Psi^{x,0}_{\max,d}$. Thus, each subpath in $R^{x,t}_d$ satisfies the maximum delay constraint $\delta^x_{\max,d}$. Both (4e) and (4f) guarantee that any non-selected subpath $P^x_{d,k} \notin P^x_d$ does not have the minimum cost. The last constraint (4f) requires each $e^{x,t}_{d,i}$ to be positive.

If the total excess cost, i.e., (4a), is not zero, **LinkCost**() sets *success* to false and returns ψ^t without updating 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**() successfully updates set ψ^t with new link costs, it sets *success* to true. If link (u, v) has no *on*-cable, i.e., $n^T_{uv} = 0$, Line 22 (Line 23) removes the link from sets $\mathcal{L}$ (L). This allows all cables in the c-link (u, v) to remain *off* in subsequent stages. Line 26 of **MGTE**() then updates the weight w_x of each l-switch $x \in X$ because the new routing produced by Lines 3 - 25 is able to increase the number of *off*-cables. Finally, Line 17 of M-GMSU computes ε^t. Overall, Phase 3 produces a set R^t that contains all routable paths for all demands in D^t at each stage $t \in [1, T]$ and a set of link costs ψ^t.

B. DETAILS OF DP-GMSU

This section presents our approach to enable link-disjoint path routing in M-GMSU; we call this approach DP-GMSU. In DP-GMSU, we replace the function **Reroute**() in Line 8 of function **MGTE**() with the function **RerouteDP**(). As in **Reroute**(), the function **RerouteDP**() aims to reroute traffic carried by path $R_{d,i}^{s_d,t} \in Q_{uv}$ via $m \geq 1$ alternative paths in set $\mathcal{P}_d^{s_d}$. For each demand d, the function considers two possible cases: 1) the demand is initially routed via non link-disjoint paths, or 2) the demand is initially routed via link-disjoint paths. For case 1), function **RerouteDP**() uses the function **Reroute**() to reroute demand d using not necessarily link-disjoint paths. For case 2), the function **RerouteDP**() aims to reroute demand d via at least two link-disjoint paths to route demand d. The function carries out the following three steps:

(a) If set $R_d^{s_d,t}$ contains $R_{d,i}^{s_d,t}$ and other $m \geq 2$ paths, where each path can carry an additional traffic of size $f_{d,i}^t/m$, use all of the m paths.
(b) If step (a) fails and s_d is an s-switch, find a node-disjoint path among the $m \geq 2$ paths that can carry an additional traffic of size $f_{d,i}^t/m$. If such path does not exist, find $k \geq 1$ link-disjoint paths in the set $\tilde{P}_d^{s_d,t}$. Here, each path must be able to carry an additional traffic of size $f_{d,i}^T/k$ and are link-disjoint with the m paths.
(c) If step (a) fails and s_d is an l-switch, find one path in the set $\tilde{P}_d^{s_d,t}$ that can carry an additional traffic of size $f_{d,i}^T$ and is link-disjoint with the m paths.

The function **RerouteDP**() returns false when it fails to find the m paths from either of the two cases.

C. AN EXAMPLE

This example illustrates how to use the three phases of M-GMSU and DP-GMSU to upgrade the legacy network $G^0(V, E)$ in Figure 1a, where each link has a delay of one second. The plan is to upgrade the network in $T = 2$ stages using a total budget $B = \$45$, i.e., $B^1 = B^2 = \$22.5$. Each switch v has an initial upgrade cost of $p_v^0 = p_v^1 = \$15$, which is reduced to $p_v^2 = \$12$ at the second stage. The first demand $d = 1$ which is $(s_1 = 1, \tau_1 = 3, \omega_1^0 = 5)$, and the second demand $d = 2$, i.e., $(s_2 = 2, \tau_2 = 4, \omega_2^0 = 5)$, have their traffic size increases by $\mu = 0.2$ per stage, i.e., from $\omega_1^1 = \omega_2^1 = 5$ to $\omega_1^2 = \omega_2^2 = 6$. This example considers a delay tolerance $\sigma = 1.1$, e.g., demand $d = 1$ that is routed via path $(1, 2, 3)$ has $\delta_{\min,1}^1 = 2$ and maximum path delay of $\delta_{\max,1} = \lceil 1.1 \times 2 \rceil = 3$ seconds.

In Phase 1, M-GMSU equally distributes demands $d = 1$ ($d = 2$) via paths with shortest delays in set P_1^1 (P_2^1). For example, initially demand $d = 1$ has routable paths $R_1^{1,0} = P_1^1 = \{(1, 2, 3), (1, 5, 3), (1, 4, 3)\}$. Traffic volume $\omega_1^2 = 6$ is then split equally into $\omega_1^2/|R_1^{1,0}| = 2$ units each. Figure 1a shows the traffic distribution for both demands. From the distribution, we get the total traffic volume on each link, e.g., $f_{(2,3)}^2 = 2 + 2 = 4$, and the required number of *on*-cables on each link, e.g., $n_{(2,3)}^2 = \lceil f_{(2,3)}^2/\gamma \times U_{max} \rceil = \lceil 2/5 \times 0.8 \rceil = 1$ *on*-cable; thus there are $(b_{(2,3)} - n_{(2,3)}^2) = (2 - 1) = 1$ unused cables for link $(2, 3)$. Thus, there are $w_1 = 8$, $w_2 = w_3 = w_4 = 8$ and $w_5 = 12$ *off*-cables for the respective l-switches $X = \{1, 2, 3, 4, 5\}$.

In Phase 2 and stage $t = 1$, **Selection**() upgrades only l-switch $v = 5$ that has the highest ratio $w_5/p_5^1 = 12/15 = 0.8$. Thus, the function returns $X = \{1, 2, 4, 5, 6\}$, $V^1 = \{5\}$, remaining budget $\Delta B^1 = \$7.5$, weight $w_1 = 8 - 3 = 6$ and $w_2 = w_4 = w_3 = 6$, and four c-links with traffic flows, i.e., $L = \{(1, 5), (5, 3), (2, 5), (5, 4)\}$. Function **MGTE**() initializes $R^1 = \{R_1^{1,0} = \{(1, 2, 3), (1, 5, 3), (1, 4, 3)\}, R_2^{2,0} = \{(2, 1, 4), (2, 5, 4), (2, 3, 4)\}\}$, $\mathcal{L} = L$, $\psi_{uv}^1 = \psi_{uv}^0 = 1$ for each link $(u, v) \in E$. The function enumerates two sets of routable subpaths $R_1^{1,1}$ and $R_2^{2,1}$ which are the same as their routable paths in R^1 because the source of both demands are legacy. Thus, we have $\tilde{P}_1^{1,1} = \tilde{P}_2^{2,1} = \{\}$. Lines 5 - 6 of **MGTE**() only turn off one cable in c-link $(1, 5)$. Both **Reroute**() or **RerouteDP**(), in Line 8, use their second case with step (a) to reroute path $(1, 5, 3) \in Q_{(1,5)}$ to paths $\{(1, 2, 3), (1, 4, 3)\}$. Each of the two paths is able to carry an additional traffic of size $2/2 = 1$ unit; see Figure 1b. **LinkCost**() solves the LP in (4b) and returns a zero excess cost for each selected path in sets $R_1^{1,1}$ and $R_2^{2,1}$. It increases the cost of link $(1, 5)$ by one such that the non-selected path $(1, 5, 3)$ has cost $\psi_{(1,5)}^1 + \psi_{(5,3)}^1 = 2 + 1 = 3$, which is higher than the selected paths $R_1^{1,1} = \{(1, 2, 3), (1, 2, 3)\}$, each with a cost of two; see Figure 1b. By using f_{uv}^1 for each link, e.g., $f_{(2,5)}^1 = f_{(2,5)}^2/(1 + 0.2) = 2/1.2 = 1.67$, there are 14 unused cables: one cable each on link $(2, 5)$ and $(5, 4)$ and two cables each on link $(1, 5)$, $(5, 1)$, $(5, 2)$, $(5, 3)$, $(3, 5)$ and $(4, 5)$. Thus, we can save $\varepsilon^1 = 14/32 = 43.75\%$ of energy.

In stage $t = 2$, with budget $B^2 = B^2 + \Delta B^1 = 22.5 + 7.5 = \$30$, **Selection**() returns $X = \{1, 3\}$, $V^2 = \{2, 4\}$, $L = \{(2, 5), (5, 4), (1, 2), (2, 1), (1, 4), (2, 3), (3, 4), (4, 3)\}$, and $\Delta B^2 = 6$. Further, **MGTE**() reroutes only path $(2, 3, 4)$, which passes c-link $(3, 4)$ to path $(2, 5, 4)$ to turn off the only cable in the link. As shown in Figure 1b, two routable paths carry different traffic volume of demand $d = 2$. This is allowable as the source of the path, i.e., $v = 2$, is now an s-switch. For this last stage, another 11 unused cables can be off, i.e., one cable each on links $(1, 2)$, $(2, 1)$, $(2, 3)$, $(1, 4)$, and $(4, 3)$, and two cables each on links $(3, 2)$, $(3, 4)$ and $(4, 1)$. Thus, M-GMSU obtains energy saving $\varepsilon^2 = (11 + 14)/32 = 78.12\%$ and average energy saving over $T = 2$ of $\varepsilon_2 = (43.75 + 78.12)/2 = 60.94\%$.

D. ALGORITHM ANALYSIS

The following two propositions analyze M-GMSU in terms of the algorithm's compliance to all constraints in MIP (2b) and time complexity, respectively.

Proposition 1: Given a legacy network $G^0(V, E)$, at each stage $t \in [1, T]$, M-GMSU produces (a) a set of s-switches V^t with the total upgrade cost within the maximum available budget B^t, and (b) a routing set R^t and a set of link costs ψ^t that satisfy the following constraints: (i) the maximum link

utilization $U_{max} \times c_{uv}$, (ii) the maximum path delay $\delta_{max,d}$ for each demand $d \in [1, D^t]$, and (iii) OSPF-ECMP for each l-switch $x \notin \{V^k \mid k \in [1, t]\}$.

Proof: For result (a), at each stage $t \in [1, T]$, Phase 2 via function **Selection**() ensures that a switch can be upgraded only if its cost is no more than the remaining budget; see Line 1 of function **Selection**(). For result $b(i)$, M-GMSU and DP-GMSU respectively use function **Reroute**() and **RerouteDP**() in Line 8 of function **MGTE**() to find $m \geq 1$ paths, each of which can carry an equal extra traffic volume of size $f^t_{d,i}/m$. This indicates that the total traffic volume f^t_{uv} for each link $(u, v) \in E$ that belongs to every of m paths is within the maximum link utilization, i.e., $f^t_{uv} \leq U_{\max} \times c_{uv}$. Further, to address requirement $b(ii)$, each of the m paths is found from either $R^{s_d,t}_d \in \mathcal{P}^{s_d}_d$ or $\tilde{P}^{s_d,t}_d \in \mathcal{P}^{s_d}_d$ that has delay no longer than $\delta_{\max,d}$. This proves that Line 8 of function **MGTE**() produces a set of paths R^t that satisfy the maximum delay constraint. Function **MGTE**() uses either function **Reroute**() or **RerouteDP**() and **LinkCost**() to satisfy $b(iii)$. Functions **Reroute** and **RerouteDP**() always *equally* distribute extra traffic volume $f^t_{d,i}$ to $m \geq 1$ paths. Therefore, each subpath from l-switch x to destination τ_d, i.e., $R^{x,t}_{d,i}$ that resides in any of the m paths, carries the same size of traffic demand d. Afterwards, function **LinkCost**() in Line 15 of function **MGTE**() ensures each subpath $R^{x,t}_{d,i}$ is a shortest path. It solves LP (4b) and *only* updates the link costs in ψ^t if each subpath $R^{x,t}_{d,i}$ in every set $R^{x,t}_d$ has a minimum cost. If all calls to functions **Reroute**() (or **RerouteDP**()) and **LinkCost**() return false at every stage $t \in [1, T]$, M-GMSU uses the initial link costs $\psi^t = \psi^0$ and routing $R^t = R^0$. Recall that set R^0 contains all shortest paths within delay $\delta_{\max,d}$. Each path in every $R^{s_d,t}_d \in R^0$ carries an equal traffic size of demand d; see Line 4 of M-GMSU. Consequently, each path from l-switch x to τ_d carries an equal size of traffic demand d. □

Proposition 2: The time complexity of M-GMSU and DP-GMSU is $O(|V|^2|E|^2 + \alpha|E|)$.

Proof: Let us first compute the time complexity of functions **Selection()** and **MGTE()** before analyzing the complexity for M-GMSU and DP-GMSU. Function **Selection**() takes $O(|V|^2 + |V| + |V||E|) = O(|V||E|)$ because (i) Line 2 has a run-time of $O(|V|)$, (ii) Lines 3–5 each takes $O(1)$, (iii) Lines 6–11 requires $O(|E|)$, and these lines are repeated at most $|V|$ times. Finally, Line 13 gives a constant time of $O(1)$.

The time complexity of **MGTE**() is computed as follows. Note that $|D| = |D^t|$ for every $t \in [1, T]$. Line 1 takes $O(K|D| + 2|E|)$. Line 2 has the worst case of time complexity of $O(K|D||V|)$. Lines 3–25 are repeated $O(|E|)$ times because, in the worst case, the number of c-links in $\mathcal{L}$ is the as same the number of links in E. For each repetition, Line 4 and Line 5 respectively need $O(|E|)$ and $O(|D||E|)$, while Line 6 takes a constant time. Let K be the maximum among the number of paths $|\mathcal{P}^{s_d}_d|$ for each demand d. Line 8 uses either function **Reroute**() or function **RerouteDP**(). Function **Reroute**() falls in either case 1) that takes $O(K|E|)$, or case 2) that consists of three steps. Specifically, Steps (a) and (c) require $O(K|E|)$, while step (b) takes $O(K^2|E|)$. Note that each step must check the residual capacity of each link in each of the K paths. Thus, the worst case of time complexity of **Reroute**() is $O(K^2|E|)$. Function **Reroute**() is used by **RerouteDP**() for its first case. On the other hand, the second case of **RerouteDP**() executes steps (a), (b), and (c). The worst case is in step (b) that also takes $O(K^2|E|)$. Thus, **RerouteDP**() also requires $O(K^2|E|)$. Line 9 takes a constant time, while Line 10 needs up to $O(K)$. Lines 7–12 are repeated in the worst case $O(|D|)$ times, and thus, they take $O(K^2|D||E|)$ time. Function **LinkCost**(), called in Line 15, solves LP (4b) in $O(\alpha)$, where α is the worst case run-time to solve the LP. Line 18 can revert up to $K|D|$ paths and update traffic volume on each link. Thus, its complexity is $O(K|D||E|)$. Lines 19–20 and Lines 22–23 take $O(1)$, while Line 26 takes $O(|E|)$. Thus, the time complexity of **MGTE**() is $O(|D| + |E| + K|D||V| + |E|(|E| + |D||E| + K^2|D||E| + \alpha + K|D||E|) + |E|) = O(|E|(K^2|D||E| + \alpha))$.

We are now ready to show the time complexity of M-GMSU. Line 1 needs $O(E)$. Yen's algorithm [28], used in Line 3, takes $O(K|D||V|(|E| + |V|log|V|))$ to generate up to K alternative paths for each demand in D. Line 4 needs $O(K|D|)$ in worst case and Lines 5–8 take $O(K|D||E|)$. Note that traffic volume ω^T_d for each demand $d \in D$ can be computed in $O(1)$. Lines 10 and 11 require $O(|E|)$, while Line 12 takes $O(|V|)$. Thus, Phase 1 takes in total $O(K|D||V|(|E| + |V|log|V|) + \alpha + K|D| + K|D||E| + K|D||E| + |E| + |V|) = O(K|D||V|(|E| + |V|log|V|))$. As previously described, **Selection**() called in Line 14 takes $O(|V||E|)$. Line 15 takes $O(1)$. As previously explained, **MGTE**() called in Line 16 takes $O(|E|(K^2|D||E| + \alpha))$. Note that Lines 14–16 are repeated T times. Thus, Phase 2 and Phase 3 have a time complexity of $O(T(|V||E| + |E|(K^2|D||E| + \alpha))) = O(T|E|(K^2|D||E| + \alpha))$. Finally, Line 17 needs $O(T|E|)$. Thus, the time complexity of M-GMSU is $O(K|D||V|(|E| + |V|log|V|) + T|E|(K^2|D||E| + \alpha)) = O(T|E|(K^2|D||E| + \alpha))$. Since in general we have $|E| \leq |V|^2$, $|D| \leq |V|^2$, and $T = 5$ and $K \leq 20$ are constants, the time complexity of M-GMSU is $O(|V|^2|E|^2 + \alpha|E|)$. The time complexity of DP-GMSU is the same as M-GMSU because their only difference is on the use of respectively **Reroute**() and **RerouteDP**(), which have the same time complexity. □

V. EVALUATION

We have implemented M-GMSU in C++ and Gurobi [33] 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, which are also used in [23]; see Table 2. For Abilene and GÉANT, we use their actual traffic matrices. For DFN, Deltacom and TATA, we use the gravity model [34] 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 [22], we set $\rho = 40\%$ and $\mu = 22\%$. We assign an initial upgrade cost p^0_v of \$50K, \$100K or \$150K by drawing a random number from $\mathcal{N}(2, 0.5)$ for each node v. We then

TABLE 2. Running time (in CPU seconds).

Name	$\|V\|$	$\|E\|$	$\|D\|$	Running Time			
				M-GMSU	MIP	DP-GMSU	DP-MIP
Abilene	12	30	132	0.13	3.08	0.22	2.24
GÉANT	23	74	466	1.57	71942.81	1.24	95599.2
DFN	58	174	3306	24.72	N/A	20.47	N/A
Deltacom	113	322	12656	313.42	N/A	277.27	N/A
TATA	145	372	20880	498.92	N/A	434.65	N/A

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 delay multiplier $\sigma = 1.1$.

This section is organized as follows. First, Section V-A evaluates the scalability of MIP, DP-MIP, M-GMSU and DP-GMSU in terms of their running time in CPU seconds. Then, Sections V-B and V-C analyze the effect of increasing budgets and stages on energy savings, respectively. Next, Section V-D and Section V-E study the effect of using single path routing and link-disjoint multi-path routing, respectively, on energy saving. Further, Section V-F reports the energy saving performance of MIP and M-GMSU against prior techniques in [22] and [19]. Finally, Section V-G provides additional findings.

A. RUNNING TIME

We set the budget to $B = \$1.2M$ and consider $T = 3$ stages to compare the run-time performance (in CPU seconds) of MIP, DP-MIP, M-GMSU and DP-GMSU. From Table 2, we see that the run time of all solutions increases with network size and traffic demands. The table shows that the run time of M-GMSU is far less than that of MIP, e.g., 1.57 versus 71942.81 seconds for GÉANT. Similarly, DP-GMSU runs significantly faster than DP-MIP, e.g., 1.24 versus 95599.2 seconds for the network, i.e., GÉANT. Further, MIP and DP-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 and DP-MIP versus DP-GMSU using only Abilene and GÉANT.

B. EFFECT OF INCREASING BUDGETS

Here, we consider $B = \{\$200K, \$400K, \$600K, \$800K, \$1M, \$1.2M\}$ and $T = 3$. Referring to Figure 3, M-GMSU and MIP have a higher ε_T value when the budget is large. For Abilene with budget $B = \$200K$, M-GMSU and MIP produce $\varepsilon_T = 35.67\%$ and $\varepsilon_T = 38.01\%$, respectively. Increasing the budget to $B = \$1.2M$, M-GMSU and MIP achieve a higher saving of $\varepsilon_T = 71.64\%$ and $\varepsilon_T = 71.93\%$, respectively. For GÉANT, MIP fails to compute ε_T for $B = \{\$200K, \$400K\}$ after running for one week. Running M-GMSU on Abilene and GÉANT results in energy saving that is on average only 1.32% and 3.57%, respectively, off from the optimal ε_T value obtained from solving MIP. M-GMSU produces ε_T of only up to 32.22% and 23.97% for Deltacom and TATA, respectively. The reason is because

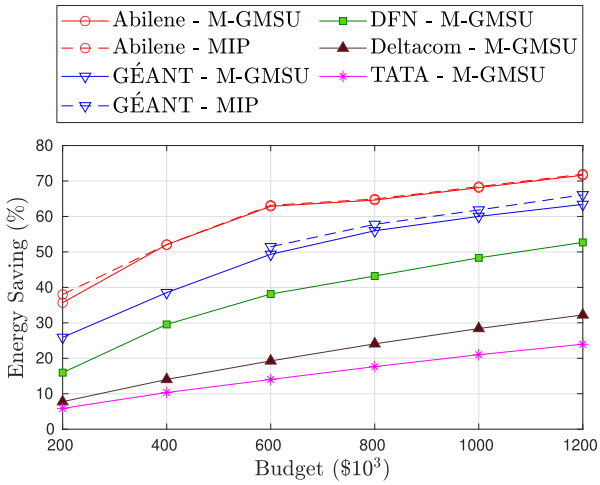


FIGURE 3. Energy saving ε_T of M-GMSU and MIP for various budget B.

Deltacom and TATA have a larger number of l-switches to upgrade than the other three networks. It means an allocated budget can only upgrade a significantly smaller percentage of l-switches. As energy saving ε_T is the result of turning off the unused cables in c-links, more s-switches can potentially lead to more switched off cables.

Note that in the last upgrade stage T, both MIP and M-GMSU route the majority of traffic demands via single paths. For example, when the budget B is \$1.2M and $T = 3$ stages, MIP routes only 1.26% and 6.44% of traffic demands via multi-paths for Abilene and GÉANT, respectively. Similarly, M-GMSU routes 37.77%, 17.24% and 3.19% of traffic demands via multi-paths for GÉANT, DFN, and Deltacom, respectively. For Abilene, M-GMSU routes each of its traffic demands via a single path. Similarly, M-GMSU routes only two demands of TATA via multi-paths. Note that there are 18.18%, 76.61%, 61.83%, 67.35% and 73.9% of traffic demands with multi-paths within delay tolerance for Abilene, GÉANT, DFN, Deltacom and TATA, respectively.

C. EFFECT OF INCREASING STAGES

Next, we investigate how the number of stages, namely $T = \{1, 2, 3, 4, 5\}$ impact energy saving ε_T. The budget B is \$1.2M. As shown in Figure 4, the energy saving ε_T for Abilene, GÉANT, and DFN decreases as T increases. For example, the energy saving ε_T for M-GMSU when it runs over Abilene (GÉANT) decreases from 74.56% to 66.67% (75% to 61.15%) when T increases from one to five. Notice that for Abilene and GÉANT, M-GMSU produces ε_T value

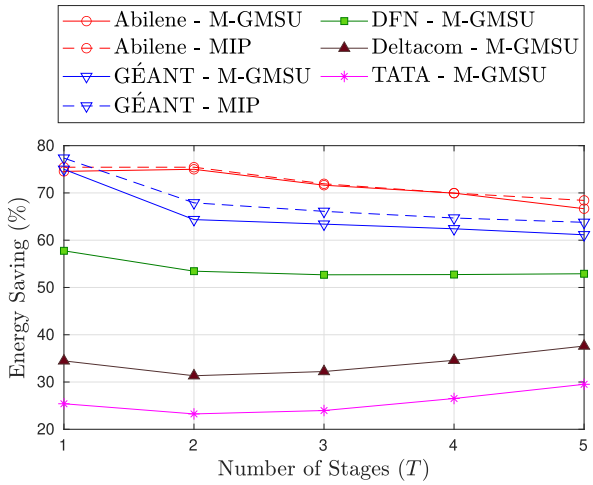


FIGURE 4. Energy saving ε_T of M-GMSU and MIP for various T stages.

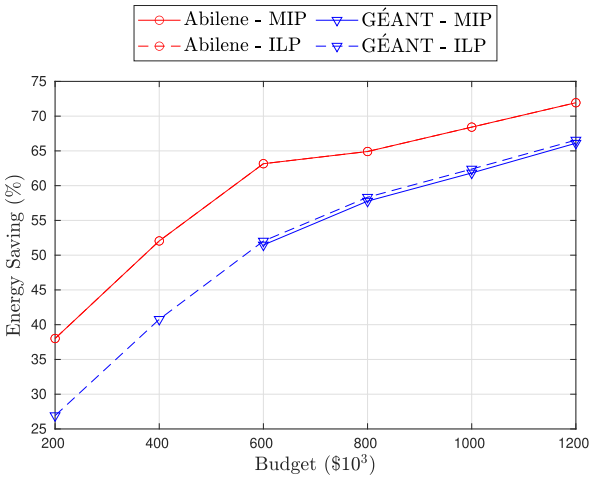


FIGURE 5. Energy saving ε_T of MIP and ILP [23].

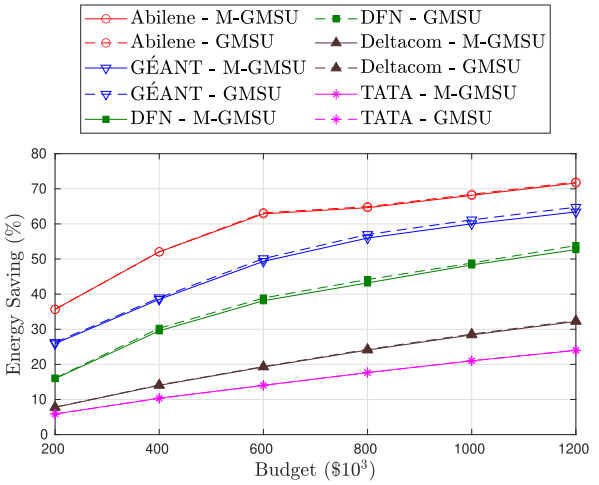


FIGURE 6. Energy saving ε_T of M-GMSU and GMSU [23].

that is on average only 0.94% and 4%, respectively, off from the optimal energy saving, which is produced 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 which results in larger ε_T values. In contrast, for smaller networks such as Abilene, a budget of $B = \$1.2\text{M}$ can be used to upgrade a larger percentage of switches in earlier stages. As a result, it reduces the number of switches to be upgraded in later stages, and thus fewer unused cables can be turned off. In addition, as the later stages have a higher traffic volume, it is unlikely that these remaining switches have idle or off cables. In other words, upgrading these switches does not significantly increase ε_T.

D. MULTI-PATH VERSUS SINGLE PATH ROUTING

In this section, we aim to compare the energy saving ε_T calculated by MIP and M-GMSU against that computed by ILP and GMSU [23], respectively. Briefly, ILP and GMSU use a single path that satisfies a given delay tolerance to route each traffic demand. ILP is the optimal approach that provides the optimal energy saving ε_T, while GMSU is the heuristic approach that produces a sub-optimal ε_T value. Further, similar to MIP and M-GMSU, ILP and GMSU perform rerouting at each upgrade stage. Here, we consider budget $B = \{\$200\text{K}, \$400\text{K}, \$600\text{K}, \$800\text{K}, \$1\text{M}, \$1.2\text{M}\}$ and $T = 3$ upgrade stages.

As shown in Figure 5, the energy saving of MIP is very close to that of ILP for each budget. Similarly, Figure 6 shows that M-GMSU and GMSU result in similar ε_T value. For Abilene, MIP and ILP produce the same saving. On average, for GÉANT, MIP produces 0.91% lower ε_T value as compared to ILP. Similarly, M-GMSU produces 0.29% and 1.62% less energy saving than GMSU for Abilene and GÉANT, respectively. Further, the ε_T value of M-GMSU is 1.77%, 0.72%, and 0.06% lower than that of GMSU for DFN, Deltacom and TATA, respectively. GMSU is more likely to have successful traffic rerouting because M-GMSU requires each l-switch x to distribute traffic over $k \geq 1$ shortest paths from x to the flow's destination. Note that traffic rerouting in GMSU is subjected only to path delay tolerance and MLU threshold. Note that ILP and GMSU are computationally faster than MIP and M-GMSU, respectively. For example, ILP respectively runs in 0.06 and 30.31 seconds for Abilene and GÉANT, while GMSU requires less than 2 seconds for each network. The reason is because both ILP and GMSU do not include link-cost setting.

E. EFFECT OF ROUTING VIA LINK-DISJOINT PATHS

This simulation aims to show the impact of routing traffic demands via link-disjoint paths. It uses $B = \{\$200\text{K}, \$400\text{K}, \$600\text{K}, \$800\text{K}, \$1\text{M}, \$1.2\text{M}\}$ and $T = 3$ stages. As shown in Figure 7, the energy saving of DP-GMSU is only 1.32% and 3.64% less than the savings of DP-MIP for Abilene and GÉANT, respectively. As an example, for budget $B = \$1.2\text{M}$, DP-MIP (DP-GMSU) produces $\varepsilon_T = 71.93\%$ (71.64%) and $\varepsilon_T = 65.77\%$ (63.29%) for Abilene and GÉANT, respectively. For GÉANT, DP-MIP fails to produce results for $B = \{\$200\text{K}, \$400\text{K}\}$ after running for one week. Similarly, DP-MIP fails to obtain results for DFN, Deltacom and TATA.

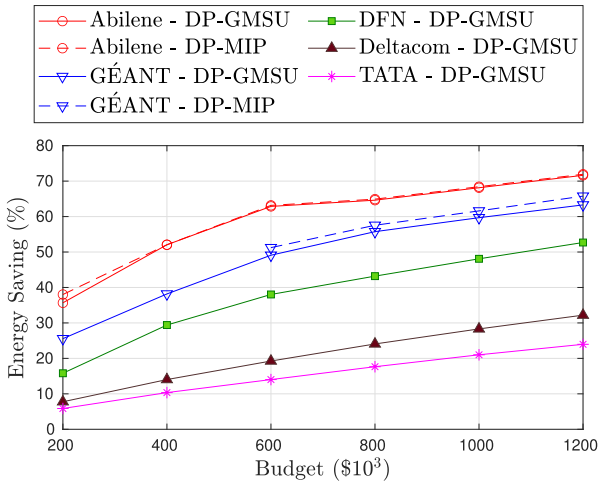


FIGURE 7. Energy saving ε_T of DP-MIP and DP-GMSU.

Overall, as shown in Figure 3 and Figure 7, DP-MIP and DP-GMSU produce energy savings that are close to those of MIP and M-GMSU, respectively. For Abilene, all solutions, i.e., DP-MIP, DP-GMSU, MIP and M-GMSU, produce the same ε_T. For GÉANT, the average saving ε_T of DP-MIP is only 0.32% less than that of MIP. Similarly, the saving of DP-GMSU is only 0.63% less than that of M-GMSU. Similarly, for DFN and Deltacom, the energy saving obtained by DP-GMSU is only 0.36% and 0.06% off, respectively, from the saving of M-GMSU. Moreover, DP-GMSU and M-GMSU produce the same saving for TATA. The reason is because DP-MIP and DP-GMSU route the majority of traffic demands via single paths. More specifically, for Abilene with budget $B = \$1.2$M, DP-MIP cannot route any traffic demand via link-disjoint paths. It routes only 2.27% of demands via non link-disjoint multi-paths. For GÉANT and budget $B = \$1.2$M, DP-MIP routes 10.3% and 7.58% of traffic demands via link-disjoint and non-link-disjoint paths, respectively. Similarly, DP-GMSU routes all demands of Abilene via single path routing, while for GEANT, it uses link-disjoint and non-link-disjoint paths to route only 10.3% and 29.4% of traffic demands, respectively. Note that the percentage of traffic demands routed over link-disjoint paths that also satisfies a given delay tolerance for Abilene, GÉANT, DFN, Deltacom, and TATA is 6.06%, 55.15%, 9.83%, 2.96%, and 3.86%, respectively.

F. M-GMSU VERSUS TWO EXISTING SOLUTIONS

In this section, we compare the performance of M-GMSU against two existing solutions, i.e., Local Search (LS) [22], and Energy-Efficient Genetic Algorithm for hybrid SDNs (EEGAH-MNL) [19], in terms of traffic controllability and energy saving. For brevity, in this paper we call EEGAH-MNL as GA. Briefly, LS aims to upgrade l-switches over multi-stages subject to a given total budget B. However, the goal is to maximize the total traffic controllability over $T \geq 1$ stages, denoted by TC. Moreover, LS is allowed to use its entire budget in one stage. On the other hand, GA aims to minimize the power consumption of links that are adjacent to an s-switch (c-links) and s-switches in a single upgrade stage, i.e., $T = 1$. Both LS and GA consider single path routing. Note that GA generates each shortest path using only the powered-on links, and thus, producing paths with long delays. Both LS and GA consider non-bundled links where they only have one cable.

We compare the performance of M-GMSU, LS, and GA using the following scenarios: (i) single path routing with 10% delay tolerance, (ii) initial upgrade cost of $p_v^0 = \$100$K for each switch v with decrease rate of $\rho = 40\%$; all switches have the same upgrade cost, (iii) a set of budget $B = \{\$200$K, \$400K, \$600K, \$800K, \$1M, \$1.2M$\}$, (iv) $T = 3$ upgrade stages, (v) MLU threshold of 80%, (vi) traffic size of each demand d increases with rate $\mu_d = 22\%$, (vii) each link contains $b_{uv} = 4$ cables, and (viii) only s-switches can turn off unused cables.

Next, we provide additional settings for our simulations:

1) We consider the following link models: (i) each link contains only one cable, i.e., $b_{uv} = 1$, and (ii) each link contains $b_{uv} = 4$ cables. For model (ii), we calculate the energy saving for LS and GA from the traffic volume on each link. Specifically, each link with traffic volume ω uses an equivalent of $\lceil \omega/\gamma \rceil$ cables, where γ is the capacity of each cable.
2) To simulate multi-stage upgrades for GA, we run the algorithm $T = 3$ times. At each stage $t \in \{1, 2, 3\}$, the number of l-switches that can be replaced by s-switches in GA is equal to the number of upgraded l-switches in M-GMSU. Note that we assume all switches have the same upgrade cost so that GA can upgrade the same number of l-switches as M-GMSU in decreasing order of the number of l-links.
3) The fitness function of GA is changed to the sum of the total number of powered-on links, assuming that the power rate of all links is the same.

Note that LS fails to produce results for DFN, Deltacom and TATA after running for three days. Thus, we use only Abilene and GÉANT to compare the TC and ε_T performance of M-GMSU, LS and GA.

1) PERFORMANCE ON TC

The TC values for non-bundled and bundled link models are exactly the same. Thus, the TC results in Figure 8 apply to both link models. Figure 8 shows that LS consistently produces, on average, higher TC than M-GMSU and GA for Abilene and GÉANT. The results are expected as the goal of LS is to maximize TC. As an example, for budget $B = \$200$K, LS produces 38.35% and 49.75% higher TC than M-GMSU, and 43.1% and 48% higher TC than GA for Abilene and GÉANT, respectively. However, as the budget increases to $B = \$1.2$M, the difference between TC of LS and M-GMSU (LS and GA) reduces to only 5.95% (9.01%) and 4.69% (5.7%) for the two respective networks. The reason is because the budget at each stage becomes larger with increasing budget B. Thus, M-GMSU and GA upgrade most of the l-switches at earlier stages and hence, produces TC with

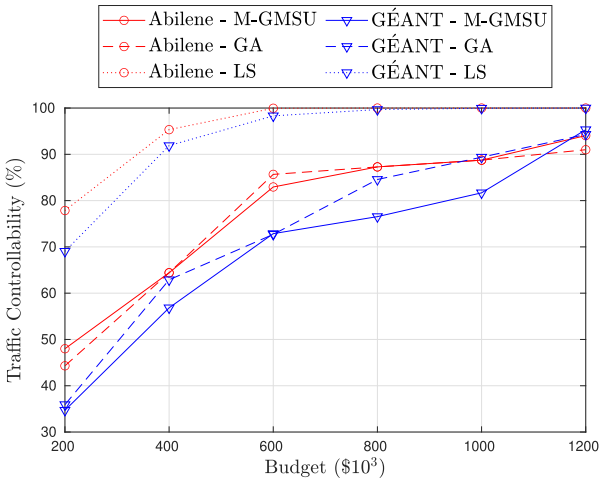


FIGURE 8. Traffic Controllability of M-GMSU, LS [22], and GA [19].

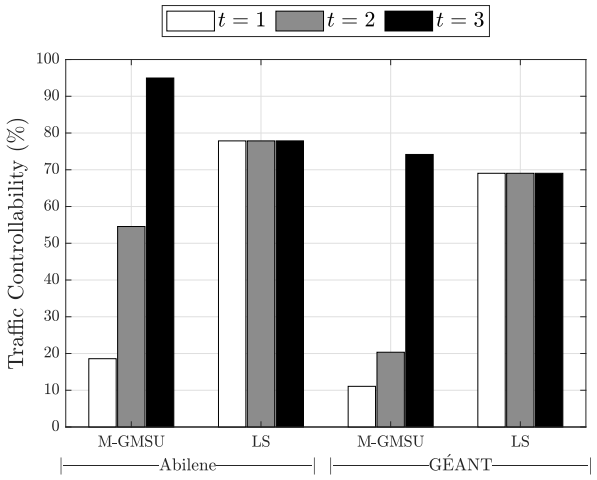


FIGURE 9. Traffic Controllability for $T = 3$ and $B = \$200K$.

values closer to LS. As shown in Figure 8, M-GMSU and GA produce comparable TC values. At maximum, M-GMSU results in 3.17% and 9.49% lower TC than GA for Abilene and GÉANT respectively. The reason is because GA upgrades *l*-switches with the highest total number of *l*-links or *node degree*. On the other hand, M-GMSU selects *l*-switches which do not necessarily have the highest total number of node degrees. Note that switches with the highest node degree are likely to be traversed by more end-to-end paths [35]. To further analyze TC performance, we show the value of TC at each stage in Figure 9. Note that M-GMSU and GA produce a similar trend, and thus, the figure only compares the results of M-GMSU and LS.

Figure 9 shows the TC produced by M-GMSU and LS at stage 1 to 3 using a budget of $B = \$200K$. M-GMSU consistently produces higher TC at the last stage, whilst TC of LS remains the same over the three stages. For Abilene, the TC produced by M-GMSU increases drastically from 18.58% to 94.97%, while LS yields the same TC of 77.86% from stage $t = 1$ to $t = 3$. Similarly for GÉANT, the TC achieved by M-GMSU escalates from 11.07% to 74.17%, whilst LS produces the same TC of 69.04% for each stage t. The reason is because LS spends its entire budget upgrading *l*-switches in the first stage. In contrast, M-GMSU has a maximum budget to spend at each stage. Further, M-GMSU aims to maximize ε_T, while LS aims to maximize TC. Thus, on average, LS results in a higher TC.

2) ENERGY SAVING PERFORMANCE

This section first evaluates the energy saving ε_T produced by M-GMSU, LS and GA for non-bundled and bundled link models. Then, it analyzes the saving ε^t of M-GMSU and LS at each stage $t \in \{1, 2, 3\}$. Lastly, it shows the advantage of saving more energy at later stages on energy cost.

a: NON-BUNDLED LINKS MODEL

Figure 10 shows the energy saving ε_T for the non-bundled link model. We see that LS uses all links to route a set of end-to-end traffic demands via shortest paths, and hence,

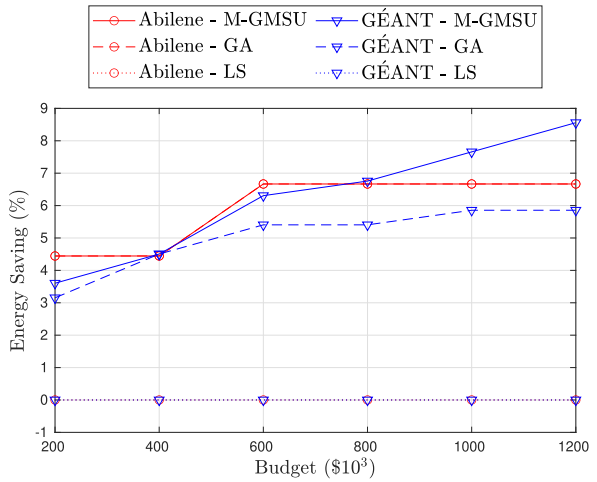


FIGURE 10. Energy saving ε_T of M-GMSU, LS [22], and GA [19] for link model with single cable.

no energy saving. In contrast, M-GMSU and GA can save energy because both solutions turn off as many links as possible and route traffic demands using the remaining active links. For Abilene, both solutions produce the same energy saving. As an example, for budget $B = \$200K$, M-GMSU and GA produce the same $\varepsilon_T = 4.44\%$ which increases to $\varepsilon_T = 6.67\%$ for larger budget $B = \$1.2M$. On the other hand, for GÉANT and budget $B = \$200K$, GA results in 12.5% less ε_T than M-GMSU. As the budget increases to $B = \$1.2M$, M-GMSU significantly overcomes GA with 26.32% higher saving. Note that the energy savings of M-GMSU outperforms those of GA for the other networks, i.e., DFN, Deltacom and TATA.

b: BUNDLED LINK MODEL

We evaluate the energy saving performance of LS and GA for the bundled-links model. Figure 11 shows that LS can save energy. It produces less ε_T value than M-GMSU and GA with budget up to $B = \$200K$ for Abilene and GÉANT. For budget $B = \$200K$, LS gives $\varepsilon_T = 21.05\%$ and $\varepsilon_T = 21.96\%$ for Abilene and GÉANT, respectively. On the other

hand, M-GMSU respectively produces higher ε_T of 29.24% and 22.3% for Abilene and GÉANT. Similarly for Abilene, GA produces the same $\varepsilon_T = 29.24\%$ which is higher than LS. However, GA produces ε_T of 21.28% which is slightly less than LS for GÉANT. However, as the budget increases, LS produces higher energy saving than M-GMSU and GA. For Abilene and GÉANT with budget $B = \$800$K, LS obtains respectively 16.67% (16.67%) and 14% (16%) higher ε_T value than M-GMSU (GA). We use Figure 12 to explain the reasons for the higher ε_T values that are produced by LS when the budget increases. We consider only M-GMSU in Figure 12 to analyze the energy saving performance against LS at each stage. The reason is because the energy savings produced by GA for all budgets, as shown in Figure 11, are the same for Abilene and only 0.05% off from the savings resulted by M-GMSU for GÉANT.

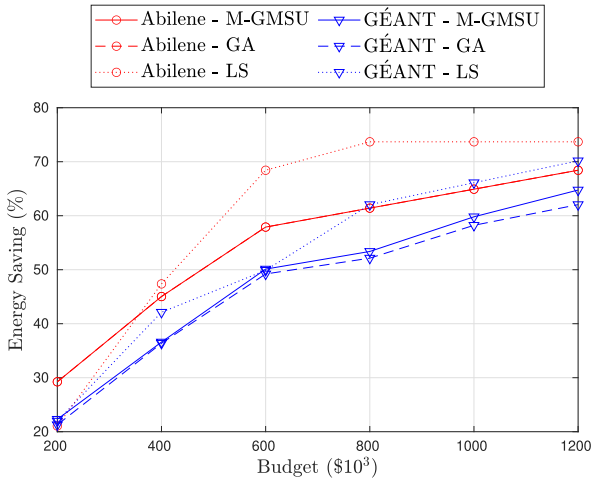


FIGURE 11. Energy saving ε_T of M-GMSU, LS [22], and GA [19] for bundled link model.

c: ENERGY SAVING PER STAGE

Figure 12 shows a comparison of the ε_T produced by M-GMSU and LS at each stage $t \in \{1, 2, 3\}$ for Abilene and GÉANT using a budget of $B = \$200$K and bundled link model. As shown in Figure 12, ε_T increases at each stage for M-GMSU, while ε_T of LS decreases slightly at later stages, especially for GÉANT. The reason is because LS uses its entire budget at stage $t = 1$ and thus, the number of s-switches upgraded by LS remains the same from stage 1 to 3. Recall that the traffic size increases at a rate of $\mu = 22\%$ per stage, and thus some cables need to be switched on, which decrease the energy saving of LS over $T = 3$ stages. Moreover, budget $B = \$400$K and $B = \$200$K are not sufficiently large for LS to upgrade all l-switches of Abilene and GÉANT in only one stage, respectively. On the other hand, M-GMSU constrains the maximum budget that can be spent at each stage. Thus, M-GMSU is able to upgrade more switches at the later stages, which increases energy savings. It is important to note that, in general, M-GMSU would upgrade a larger number of switches than LS since the upgrade cost decreases over time/stages.

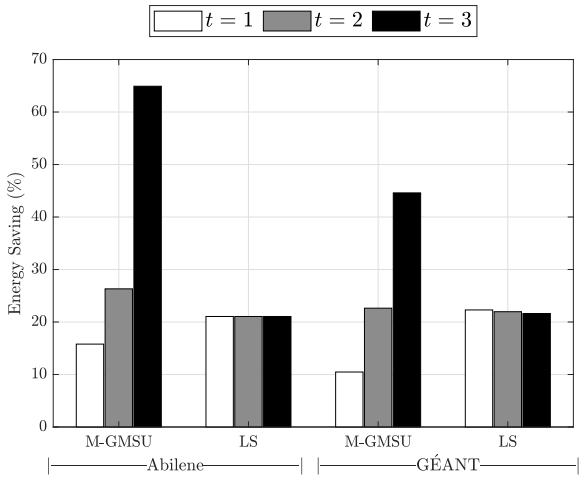


FIGURE 12. Energy saving ε_T for $T = 3$ and $B = \$200$K.

d: BENEFIT OF MORE ENERGY SAVING AT LATER STAGE

The following case study shows the benefit of saving more energy in later stages. Note that, in general, electricity cost increases in later years. For example, in the United States, reference [36] projects an annual increase in energy prices of 3.29% from 2020 to 2025. Assume Abilene and GÉANT carries out an upgrade every two-year using a total budget of $B = \$200$K for $T = 3$ upgrade stages. For Abilene, M-GMSU and LS produce $\{15.79\%, 26.32\%, 64.91\%\}$ and $\{21.05\%, 21.05\%, 21.05\%\}$ of energy saving, respectively, at each stage. Assuming an initial energy cost of \$1 per *on*-cable, M-GMSU will be able to save $\$(0.1579 + 0.2632 \times 1.0329^2 + 0.6491 \times 1.0329^4) = \$1.775$, while LS saves only \$0.6747. For GÉANT, M-GMSU saves \$0.8538 which is slightly higher than LS, which only saves \$0.7033.

G. ADDITIONAL RESULTS

This section reports additional findings in terms of increased path delays and link utilization when using our approach. Further, it analyzes the benefit of using l-switches that can turn off unused cables, e.g., those that support the IEEE 802.3az standard. Let us call this *green* l-switch as *gl*-switch. In addition, it discusses the energy saving performance of our approach against existing techniques in non-SDNs and pure SDNs. Section V-G1, V-G2, and V-G3 use the total budget of $B = \$1.2$M and the total number of upgrade stages is $T = 3$. On the other hand, Section V-G4 and V-G5 use a different total budget B over the same total number of upgrade stages, i.e., $T = 3$.

1) PATH DELAY

Figure 13 shows a small increase in path delays for all networks when $B = \$1.2$M is used. More specifically, path delays produced by MIP (M-GMSU) for Abilene and GÉANT, on average, are increased by 0.43% (1%) and 2.84% (0.01%), respectively. Further, only 4.3% (12.12%) and 28.39% (0.25%) of the paths in the respective networks have 10% longer delays. Note that all simulations allow 10%

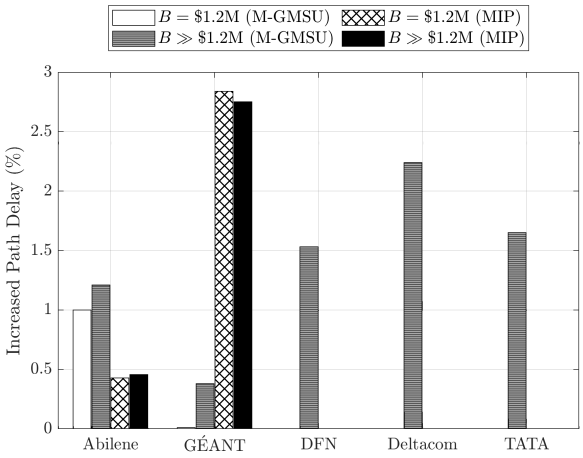


FIGURE 13. Increase in path delay produced by M-GMSU and MIP.

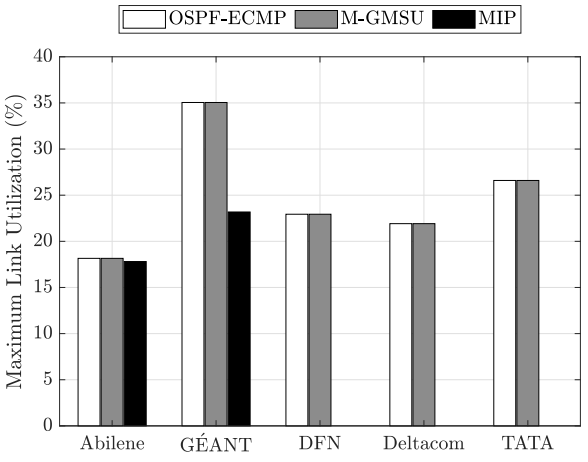


FIGURE 14. Maximum link utilization produced by OSPF-ECMP, M-GMSU, and MIP.

delay tolerance and each path originally uses the shortest path. Thus, each path cannot have a lower delay or more than 10% increase in delay. For DFN, Deltacom and TATA, there is no increase in path delay for a budget of $B = \$1.2\text{M}$. This is because the said budget can only upgrade 37.93%, 25.66%, and 19.31% of switches in the respective networks. However, when we increase the budget such that M-GMSU can upgrade more switches and all links are *c*-links, M-GMSU is able to route some demands via longer paths to maximize energy saving; see the results in Figure 13 for $B \gg \$1.2\text{M}$. For example, there are respectively 22.39% and 16.51% of traffic demands that use longer paths for Deltacom and TATA. In this case, the average path delay of these networks is increased by 2.24% and 1.65%, respectively.

2) LINK UTILIZATION

To see the effect of our MIP and M-GMSU on link utilization, we first measure the initial utilization of all links of each network, i.e., before upgrading the network. Recall that the initial routing of each demand follows the OSPF-ECMP protocol. As shown in Figure 14, we find that the maximum link utilization in the five networks ranges between 18% and 36% when all cables are turned on. More specifically, for Abilene and GÉANT, the maximum link utilization is 18.16% and 35.05%, respectively. Then, we measure link utilization of each network after upgrading the network using MIP or M-GMSU. Using MIP, the maximum link utilization in Abilene and GÉANT decreases to 17.81% and 23.18%, respectively. On the other hand, M-GMSU does not change the maximum link utilization for all networks. The reason is because M-GMSU limits the number of *on*-cables on each link at each stage according to the number of *on*-cables used at the last upgrade stage T when performing traffic rerouting. Moreover, M-GMSU reroutes any traffic demand at each stage by using its largest volume at stage T. Thus, the maximum link utilization is less likely to increase significantly.

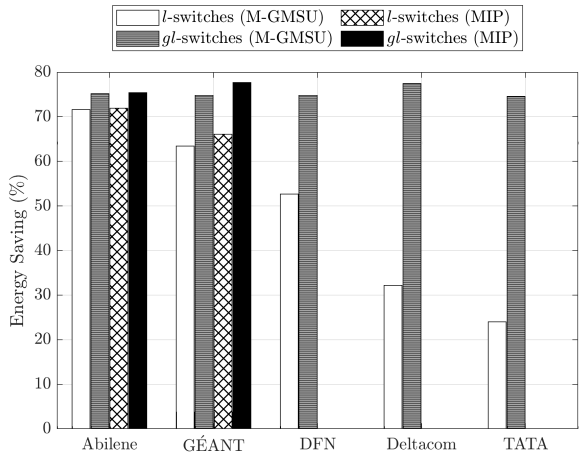


FIGURE 15. Energy saving performance with *l*-switches and *gl*-switches.

3) ENERGY SAVINGS IN NETWORKS WITH GREEN LEGACY-SWITCHES

This section examines the effect of using *gl*-switches, i.e., *l*-switches that support energy efficient technology, e.g., IEEE 802.3az, to turn-off unused cables in each *l*-link. Recall that the reported energy savings in all previous sections consider non *gl*-switches, and thus unused cables in each *l*-link are still *on*. For this examination, we modify Equation 1 to include unused cables in both *c*-links and *l*-links. Figure 15 shows that MIP increases the energy saving of Abilene and GÉANT from 71.93% to 75.44% and 66.1% to 77.7%, respectively. Similarly, M-GMSU improves the energy saving of Abilene and GÉANT from 71.64% to 75.15% and 63.4% to 74.78%, respectively. Further, For DFN, Deltacom and TATA, M-GMSU increases their saving from 52.68% to 74.81%, 32.22% to 77.43%, and 23.97% to 74.55%, respectively. The additional saving accumulates because the unused cables in each *l*-link can now be powered off by *gl*-switches, and hence saving more energy.

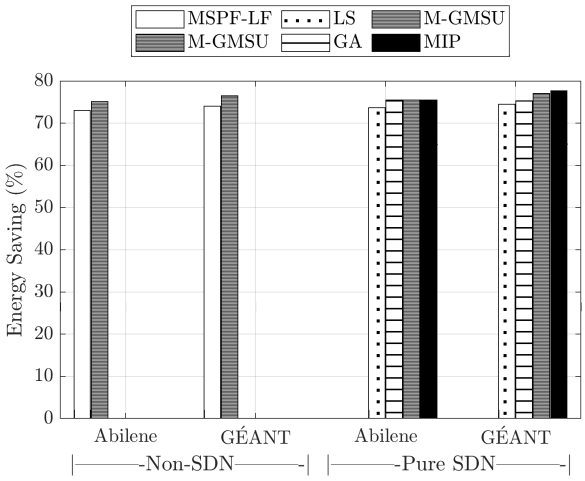


FIGURE 16. Energy saving performance in non-SDN and pure SDN.

4) ENERGY SAVINGS IN NON-SDNs

This section evaluates the energy savings in legacy networks or *non-SDNs*. More specifically, we use the greedy-based heuristic solution, called MSPF-LS, and its simulation results in [6] to represent the energy saving in non-SDNs. Similar to M-GMSU, the MSPF-LS approach in [6] considered multi-path routing, delay constraint, maximum link utilization threshold, and bundled links. Further, the results reported in [6] use the same network topologies as ours, i.e., Abilene and GÉANT. In addition, MSPF-LS used *gl*-switches that can also perform traffic rerouting. Thus, for M-GMSU, we use a total budget B that is sufficiently large to upgrade l-switches that can control all traffic flows and turn off all unused cables at the beginning of each upgrade stage. As reported in [6], MSPF-LF produced 73% and 74% energy saving for Abilene and GÉANT respectively; see Figure 16. On the other hand, the energy saving produced by M-GMSU is 75.14% and 76.46% for the respective networks. Thus, our results are better than those reported for MSPF-LF.

5) ENERGY SAVINGS IN PURE SDNs

This section presents the energy saving in *pure* SDNs. To represent the energy savings, we use GA [19] and LS [22]. In this case, except for the total budget B, we use the same scenarios and settings as in Section V-F for M-GMSU, LS and GA. We use a sufficiently large budget to upgrade all l-switches at the first stage and calculate energy saving ε_T over $T = 3$ stages. Figure 16 shows that MIP obtains the optimal energy saving of 75.44% and 77.7 for Abilene and GÉANT, respectively. M-GMSU and GA produce the same energy saving of 75.44% for Abilene and 77.03% and 75.34%, respectively, for GÉANT. For LS, the energy saving for both networks is 73.68% and 74.44%, respectively. The results show that our solutions, i.e., MIP and M-GMSU, outperform both GA and LS for *pure* SDNs. Further, we observe that the energy saving achieved in *pure* SDNs is higher as compared to those in non-SDNs and hybrid SDNs; viz. Section V-F.

VI. CONCLUSION

This paper considers the problem of upgrading a legacy network that supports OSPF-ECMP into an 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 OSPF-ECMP. This paper considers two routing scenarios: 1) multi-path and 2) link-disjoint. We have formulated an MIP for scenario-1 and its extension, called DP-MIP, for scenario-2. In addition, we have proposed two heuristic solutions: M-GMSU for scenario-1 and DP-GMSU for scenario-2. Our simulations have shown that M-GMSU and DP-GMSU require significantly less CPU time than MIP and DP-MIP, respectively. Further, M-GMSU and DP-GMSU obtain energy saving that is only up to 4% off from the optimal saving obtained by MIP and DP-MIP, respectively. The energy saving of DP-MIP and DP-GMSU when considering link-disjoint paths is only 0.63% off from the saving attained by MIP and M-GMSU. Moreover, M-GMSU produces up to 1.77% less energy saving than GMSU, which uses single path routing. We find that increasing budget and number of stages result in larger energy savings. Further, M-GMSU produces higher energy saving at later stages than an existing technique, called LS, that tends to spend its entire budget at the first stage. As the energy price (in $) is expected to increase every year, M-GMSU is expected to perform better than LS in terms of reducing the OPEX of networks. As a future work, we plan to consider multi-controllers and their placement in an upgraded hybrid SDN.

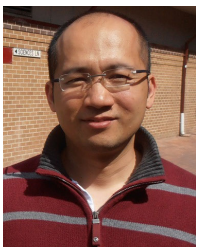
REFERENCES

[1] S. Saraswat, V. Agarwal, H. P. Gupta, R. Mishra, A. Gupta, and T. Dutta, "Challenges and solutions in software defined networking: A survey," *J. Netw. Comput. Appl.*, vol. 141, pp. 23–58, Sep. 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," *Comput. Netw.*, vol. 154, pp. 60–72, May 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] *IEEE Standard for Local and Metropolitan Area Networks–Link Aggregation*, Standard IEEE 802.1AX-2014, 2014. [Online]. Available: https://standards.ieee.org/standard/802_1AX-2014.html
[6] G. Lin, S. Soh, M. Lazarescu, and K.-W. Chin, "Power-aware routing in networks with delay and link utilization constraints," in *Proc. 37th Annu. IEEE Conf. Local Comput. Netw.*, Tallahassee, FL, USA, Oct. 2012, pp. 272–275.
[7] J. Moulierac and T. K. Phan, "Optimizing IGP link weights for energy-efficiency in multi-period traffic matrices," *Comput. Commun.*, vol. 61, pp. 79–89, May 2015.
[8] Y. Guo, F. Kuipers, and P. Van Mieghem, "Link-disjoint paths for reliable QoS routing," *Int. J. Commun. Syst.*, vol. 16, no. 9, pp. 779–798, 2003.
[9] G. Lin, S. Soh, K.-W. Chin, and M. Lazarescu, "Energy aware two disjoint paths routing," *J. Netw. Comput. Appl.*, vol. 43, pp. 27–41, Aug. 2014.
[10] M. Zhang, C. Yi, B. Liu, and B. Zhang, "GreenTE: Power-aware traffic engineering," in *Proc. 18th IEEE Int. Conf. Netw. Protocols*, Kyoto, Japan, Oct. 2010, pp. 21–30.
[11] 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.

[12] A. Cianfrani, V. Eramo, M. Listanti, M. Polverini, and A. V. Vasilakos, "An OSPF-integrated routing strategy for QoS-aware energy saving in IP backbone networks," *IEEE Trans. Netw. Service Manage.*, vol. 9, no. 3, pp. 254–267, Sep. 2012.
[13] J. Galán-Jiménez and A. Gazo-Cervero, "Designing energy-efficient link aggregation groups," *Ad Hoc Netw.*, vol. 25, pp. 595–605, Feb. 2015.
[14] 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.
[15] A. Ruiz-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.
[16] A. Fernandez-Fernandez, C. Cervello-Pastor, and L. Ochoa-Aday, "Achieving energy efficiency: An energy-aware approach in SDN," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington DC, USA, Dec. 2016, pp. 1–7.
[17] R. Amin, M. Reisslein, and N. Shah, "Hybrid SDN networks: A survey of existing approaches," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3259–3306, May 2018.
[18] 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.
[19] J. Galán-Jiménez, "Legacy IP-upgraded SDN nodes tradeoff in energy-efficient hybrid IP/SDN networks," *Comput. Commun.*, vol. 114, pp. 106–123, Dec. 2017.
[20] N. Huin, M. Rifai, F. Giroire, D. Lopez 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.
[21] T. Das, M. Caria, A. Jukan, and M. Hoffmann, "A techno-economic analysis of network migration to software-defined networking," 2013, *arXiv:1310.0216*. [Online]. Available: http://arxiv.org/abs/1310.0216
[22] K. Poularakis, G. Iosifidis, G. Smaragdakis, and L. Tassiulas, "One step at a time: Optimizing SDN upgrades in ISP networks," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, Atlanta, GA, USA, May 2017, pp. 1–9.
[23] L. Hiryanto, S. Soh, K.-W. Chin, and M. Lazarescu, "Green multi-stage upgrade for bundled-link SDNs with budget and delay constraints," *IEEE Trans. Green Commun. Netw.*, early access, May 21, 2021, doi: 10.1109/TGCN.2021.3082617.
[24] L. Hiryanto, S. Soh, K.-W. Chin, S. Pham, and M. Lazarescu, "Green multi-stage upgrade for bundled-links SDN/OSPF-ECMP networks," in *Proc. IEEE ICC*, Montreal, QC, Canada, Jun. 2021, pp. 1–7.
[25] ONF, "OpenFlow switch specification: Version 1.3.3 (wire protocol 0 × 04)," ONF, Menlo Park, CA, USA, Tech. Rep. ONF TS-015, Sep. 2013. [Online]. Available: https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.3.3.pdf
[26] 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.
[27] G. Wang, Y. Zhao, J. Huang, and W. Wang, "The controller placement problem in software defined networking: A survey," *IEEE Netw.*, vol. 31, no. 5, pp. 21–27, Sep./Oct. 2017.
[28] J. Y. Yen, "Finding the *K* shortest loopless paths in a network," *Manage. Sci.*, vol. 17, no. 11, pp. 712–716, Jul. 1971.
[29] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS weights in a changing world," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 4, pp. 756–767, May 2002.
[30] V. Sivaraman, P. Reviriego, Z. Zhao, A. Sánchez-Macián, A. Vishwanath, J. A. Maestro, and C. Russell, "An experimental power profile of energy efficient Ethernet switches," *Comput. Commun.*, vol. 50, pp. 110–118, Sep. 2014.
[31] P. Toth and S. Martello, *Knapsack Problems: Algorithms and Computer Implementations*. Hoboken, NJ, USA: Wiley, 1990.
[32] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, "Inferring link weights using end-to-end measurements," in *Proc. 2nd ACM SIGCOMM Workshop Internet Measurment (IMW)*, 2002, pp. 231–236.
[33] L. Gurobi Optimization. (2021). *Gurobi Optimizer Reference Manual*. [Online]. Available: http://www.gurobi.com
[34] M. Roughan, "Simplifying the synthesis of Internet traffic matrices," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 5, pp. 93–96, Oct. 2005.
[35] D. K. Hong, Y. Ma, S. Banerjee, and Z. M. Mao, "Incremental deployment of SDN in hybrid enterprise and ISP networks," in *Proc. Symp. SDN Res.*, Santa Clara, CA, USA, Mar. 2016, pp. 1–7.
[36] U. S. Energy Information Administration. *Annual Energy Outlook 2019*. Accessed: Jan. 8, 2020. [Online]. Available: https://www.eia.gov/outlooks/aeo/data/browser/#/?id=3-AEO2019&cases=ref2019&sourcekey=0

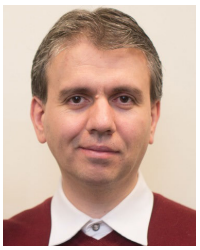
LELY HIRYANTO (Member, IEEE) received the bachelor's degree in computer science from Tarumanagara University, Indonesia, in 2001, and the Postgraduate Diploma and Master of Science degrees in computer science from Curtin University, in 2015 and 2016, respectively, where she is currently pursuing the Ph.D. degree. Her research interests include software defined networking, energy-aware traffic engineering, and network optimization.

SIETENG SOH (Member, IEEE) received the B.S. degree in electrical engineering from the University of Wisconsin–Madison, in 1987, and the M.S. and Ph.D. degrees in electrical engineering from Louisiana State University, Baton Rouge, in 1989 and 1993, respectively. From 1993 to 2000, he was with Tarumanagara University, Indonesia. He is currently a Senior Lecturer with the School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University, Perth, Australia. He has published over 100 international conference and journal articles. His current research interests include algorithm design, network optimization, and network reliability.



KWAN-WU CHIN received the Bachelor of Science and the Ph.D. degrees (Hons.) from Curtin University, Australia, in 1997 and 2000, respectively. From 2000 to 2003, he was a Senior Research Engineer with Motorola. In 2004, he joined as a Senior Lecturer with the University of Wollongong. He is currently an Associate Professor. To date, he holds four United States of America (USA) patents and has published more than 160 conference Pand journal articles. His research interests include medium access control protocols for wireless networks, and resource allocation algorithms/policies for communications networks.

DUC-SON PHAM (Senior Member, IEEE) received the Ph.D. degree from the Curtin University of Technology, in 2005. He is currently a Senior Lecturer with the Discipline of Computing, Curtin University, Perth, Western Australia. His current research interests include sparse learning theory, large-scale data mining, convex optimization, and advanced deep learning with applications to computer vision and image processing. He was a recipient of the Young Author Best Paper Award 2010 for a publication in IEEE TRANSACTIONS ON SIGNAL PROCESSING.



MIHAI M. LAZARESCU (Member, IEEE) received the B.S. (Hons.) and Ph.D. degrees in computer science from Curtin University, Perth, Australia, in 1996 and 2000, respectively. He has been a Senior Member of the IMPCA Research Institute for ten years. He is currently the Computing Discipline Lead and an Associate Professor with Curtin University. He has published over 80 articles in refereed international journals and conference proceedings in the areas of artificial intelligence, machine vision, data mining, and network reliability.