# Restorable Energy Aware Routing with Backup Sharing in Software Defined Networks

Rui Wang, SuixiangGao, Wenguo Yang, and Zhipeng Jiang
School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, 101408, China
Email: wangrui11@mails.ucas.ac.cn; {sxgao, yangwg, jiangzhipeng}@ucas.ac.cn

***Abstract*** —Energy aware routing with restoration is of much importance for the network to enhance reliability as the relatively concentrated traffic makes the network vulnerable to link failures and sudden traffic bursts after energy saving. For restoration, each request has an active path and a link disjoint backup path. With sufficient path routing knowledge, backup paths can be shared for bandwidth efficiency. This is possible in Software Defined Networks (SDNs) owing to the centralized controller which can collect global information. In this paper, we first combine energy saving with restoration and address the problem called Restorable Energy Aware Routing with Backup Sharing (REAR-BS) in SDNs during off-peak hours when backup sharing is allowed. We formulate a nonlinear integer programming model to minimize the number of used links while putting idle ones to sleep under constraints of maximum link utilization and restoration. We rewrite it as a linear model and give a lower bound for this NP-hard problem. Then, we design a heuristic algorithm called Green Restorable Algorithm (GreRA) to tackle this problem. Extensive simulation results on real and synthetic topologies and traffic demands show the effectiveness of GreRA when compared to CPLEX solutions, shortest path routing with restoration and lower bound.

***Index Terms***—Energy aware routing, restoration, link disjoint, nonlinear integer programming, reliability, backup path.

## I. Introduction

There is a growing concern on minimizing energy consumption in the Information and Communication Technology (ICT) sector as ICT alone is estimated to be responsible for 2 to 10 percent of worldwide energy consumption [1]. And networking devices like IP routers consume the largest majority of energy. This gives rise to some effective energy aware routing strategies [2]-[4].

For energy aware strategies, network components are turned on/off or put to active/sleep mode based on traffic load. Only the minimum number of links and nodes are on/active to support the concentrated traffic for all source-to-terminal requests, while idle ones are off/asleep. If there is a single link failure or sudden traffic bursts, the remaining on/active components with limited resources may not be able to support the demands of all requests immediately. Thus the reliability of network is decreased after energy saving as the network is vulnerable to link failures and sudden traffic bursts. It is important to ensure network reliability while keeping low energy consuming. This can be achieved by taking restorable routing into account while energy saving. Though this combined research is still in its infancy, restorable routing is wide used to ensure resilience to failures [5]-[8]. For restoration, an active path and a link disjoint backup path with sufficient bandwidth are setup for each request simultaneously. The active path is used to transit traffic for the traffic. And the traffic is immediately switched to the backup path only if the active path fails. As the components of the active and backup paths of all requests are powered on, the affected requests can be quickly restorable upon single link failure. We aim to ensure the network be resilient to single link failure while keeping low energy consumption. Hence on the one hand, the number of powered-on components should be minimized from the point of energy saving. On the other hand, disjoint backup paths result in excessive bandwidth usage although they can enhance the resilience to failures. However, backup paths can be shared for bandwidth efficiency. This can be achieved if the nodes know the amount of bandwidth on each link that is currently used by backup paths for providing backup[8] and is possible in Software Defined Networks (SDNs [9], [10]) with simple extensions to link-state routing protocols. SDN employs a new paradigm that decouples the control plan from the data plan and moves the control logic to centralized controller which can ensure compatibility with existing protocols. In SDNs, the flexibility of developing and testing new protocols and the ease of implementation and configuration on a global network view creates favorable conditions for restorable energy aware routing.

The key of restoration is to find 2 disjoint paths [11]. The only work related to restorable energy aware routing is shown in [12]. However, their problem called Energy Aware Two Disjoint Paths Routing is different from restoration in which each request can be routed via multiple paths besides 2 disjoint paths, i.e., in a splittable way. In contrast, we focus on the unsplittable case in wired networks and take restoration into account when backup sharing is allowed.

The main contribution of this paper is as follows. We address the problem called Restorable Energy Aware

Manuscript received March 1, 2015; revised August 18, 2015.
This work was supported by the National 973 Plan project under Grant No. 2011CB706900,the National 863 Plan project under Grant No.2011AA01A102, the NSF of China (11331012, 71171189), the "Strategic Priority Research Program" of the Chinese Academy of Sciences (XDA06010302), and Huawei Technology Co. Ltd.
Corresponding author email: wangrui11@mails.ucas.ac.cn.
doi:10.12720/jcm.10.8.551-561

Routing with Backup Sharing (REAR-BS) to minimize energy expenditure during off-peak hours in SDNs. To the best of our knowledge, this is the first work that combines energy aware routing with restoration. Specially, we compute an active path and a link disjoint backup path for each source-to-terminal request and consider the amount of bandwidth of backup paths that can be shared simultaneously. A nonlinear integer programming model is formulated for this NP-hard problem to minimize the number of used links under constraints of maximum link utilization and restoration with backup sharing. We rewrite this model with quadratic constraint to its linear version and give a lower bound for the optimal solution. Then, we design Green Restorable Algorithm (GreRA) to solve REAR-BS, which is based on the analysis of the amount of bandwidth that can be shared for each link and the idea of incremental minimization. Extensive simulation results show the effectiveness of our algorithm. Particularly, GreRA can reduce energy usage by 15% to 50%, even for maximum link utilization below 50%.

The rest of the paper is organized as follows. Section 2 shows the basic assumptions and definition of REAR-BS. A nonlinear integer programming model and its linear version are formulated. Difficulty analysis and a lower bound are also given. Section 3 proposes heuristic algorithm GreRA with time complexity analysis. Section 4 shows the experimental results. Section 5 presents the conclusion.

## II. Related Work

Green Networking has been widely investigated in the last decade, starting from the seminal work of Gupta and Singh [3] or more recently in [2], [4], [13]-[16]. Gupta and Singh [3] propose uncoordinated sleeping at component level and coordinated sleeping at network level to save energy. Many research focus on reducing energy consumption at network level. Chabarek *et al.* [13] present the measurements of power consumption of network devices and explores the potential power saving in the network design and routing protocols in wire-line networks. Chiaraviglio *et al.* present a similar MCMF (multi-commodity minimum cost flow problems) model to minimize the power consumption of used routers and links under maximum link utilization constraint [4], [14]. Their heuristic sorts routers according to power consumption and iteratively switches off redundant ones. Fisher *et al.* consider saving energy when links are bundled with multiple cables [15]. They formulate an integer linear programming model and propose heuristic algorithms based on linear relaxation to solve it. Zhang *et al.* propose GreenTE [2], a global traffic engineering mechanism which aims to maximize the energy savings of idle links with load balancing consideration by rerouting traffic in a splittable way. CPLEX is used to solve this mixed integer programming problem with a time limit of 300s. Giroire *et al.* [16] propose a new energy aware routing model called GreenRE with the support of data redundancy elimination. They try to switch off possible links and nodes performing redundancy elimination based on the feasible solution of the model via CPLEX. The relatively centralized traffic after energy saving makes the network vulnerable to link/node failure. However, all the aforementioned works do not take enhancing network reliability (for example, using disjoint paths) into account at the same time.

Software Defined Networking has been attracting a growing concern in recent years [10]. Assefa and Ozkasap [17] address the importance of energy efficiency mechanism in SDN and summarize these strategies for SDN [18], [19]. For example, Heller *et al.* present ElasticTree [18] to save energy through switching off unnecessary links and switches of Data Center Networks, which is implemented using OpenFlow and focusing on tree-based topologies. Giroire *et al.* [19] aim to minimize energy consumption in SDNs under constraints of capacity and finite rule space on routers. They present an integer programming model as well as an efficient heuristic. Wang *et al.*[20] consider minimizing the power of integrated chassis and line-cards in SDNs based on an expand topology constructed according to routers' connection. Link utilization and delay constraints are considered in their integer programming model. Two heuristics are present. One tries to adjust as few requests as possible, the other recomputes the path with minimum energy increase for each request. Markiewicz *et al.* [21] try to switch on a minimum amount of routers and links to carry the traffic. They form a mixed integer programming model and propose heuristics that iteratively select a path for each request among precomputed shortest pathsin different processing order of requests. However, all these works do not take reliability into account as well.

Restoration has been proposed to enhance reliability [5]-[8]. The key of restoration is to find 2 link/node disjoint paths for a source-terminal pair. Finding these 2 paths with minimum total length is called shortest disjoint pair problem [11]. Suurballe and Tarjan propose Suurballe algorithm to solve this problem accurately in $O(m\log_{(1+m/n)} n)$ time ($n$, $m$: the number of nodes and links, respectively) [11]. Kodialam and Lakshman present algorithms based on Suurballe algorithm and linear programming duality for dynamic routing of restorable bandwidth guaranteed paths [5], [7]. They analyze the case when backup sharing is not allowed or allowed and formulate relative models. But they only process request one at a time and aim to minimize the bandwidth consumed by one request. Kar *et al.* considers dynamic routing with restorable routing when backup sharing is not allowed [8]. They focus on improved path selection and propose algorithms based on Suurballe algorithm and minimum interference criteria. Bejerano *et al.* [6] combine computing Qos paths for one request with restoration under constraints such as bandwidth and delay.

They focus on finding a minimum restoration topology of a set of bridges, with each bridge protecting a portion of an active path. Guo *et al.* proposes the Np-complete problem MCLPP [22] to find a pair of link disjoint paths with minimal total length under multiple link metric constraints. However, all these works focus on one request and do not take energy consumption into account.

The only work that combines energy saving with 2 disjoint paths is shown in [12]. Lin *et al.* address Energy Aware Two Disjoint Paths Routing (EAR-2DP) to maximally switch off redundant links while using 2 disjoint paths routing under constraint of maximum link utilization. They show the problem is NP-complete. However, they assume that the flow is splittable. Namely, in addition to a pair of disjoint paths, a request can be routed via multiple intersecting paths. They propose 2DP-SP algorithm to solve EAR-2DP. 2DP-SP pre-computes candidate 2DPs for each request based on Yen's k shortest paths algorithm [23] and select the pair with minimum total length as much as possible. Then redundant links are switched off iteratively. Their work is extended to switch off both nodes and links [24]. They give priority to switch nodes. However, their heuristics are time-consuming and they do not find the intrinsic relationship between link disjoint version and node disjoint version.

Different from works above, we first address REAR-BS which combines energy saving with restoration when backup sharing is allowed. And our heuristic which focuses on link disjoint version can be easily adapted to find node disjoint paths via node splitting [25].

## III. Problem Statement and Mathematical Model

In this section, we give the definition of REAR-BS and formulate a nonlinear mathematical model based on following assumptions. First of all, we assume that there is at most one single link failure at one time. Secondly, each link consumes the same power and can be asleep independently. A link can be put to sleep only if there is no traffic in both inbound and outbound directions. Moreover, we assume that each request considered here always has two disjoint paths during off-peak hours. Finally, we focus on the unsplittable case. Traffic transmits only on the active path. Backup path should have sufficient bandwidth to route the request in case of failure.

### A. Problem Statement

Consider a network that is represented by a directed graph $G(N,A)$, where $N$ is the set of $n$ nodes (i.e., routers) and $A$ is the set of $m$ links. Denote the capacity/bandwidth of each link $(u,v)\in A$ as $c_{uv}$. Let $\alpha(0<\alpha\le 1)$ be the maximum link utilization threshold. There are $D$ requests $LSPs=\{(s_i,t_i,d_i), i=1,\cdots,D\}$ in the network, where $s_i$, $t_i$, $d_i$ represent the source, terminal, traffic demand of request $i$, respectively. The goal of Restorable Energy Aware Routing with Backup Sharing (REAR-BS) is to minimize the number of used links $L$ by rerouting traffic in an unsplittable way so that each request $i$ has an active path $P_1^i$ and a link disjoint backup path $P_2^i$ and the maximum link utilization is no more than $\alpha$ when backup sharing is allowed. Both of $P_1^i$ and $P_2^i$ should have sufficient capacity (larger than $d_i$) to route request $i$. Please see Table Ifor the notation used in the paper.

TABLE I: NOTATION USED IN THIS PAPER

| Notation | Meaning |
|---|---|
| $\alpha$ | threshold of maximum link utilization |
| $c_{uv}$ | Capacity/bandwidth of link $(u,v)$ |
| $c'_{uv}$ | Available capacity/bandwidth of link $(u,v)$ |
| $P_1^i$ | Active path of request $i$ |
| $P_2^i$ | Backup path of request $i$ |
| $Paths$ | Set of all active and backup paths. $Paths=\bigcup_{i=1}^{D}(P_1^i \cup P_2^i)$ |
| $R_{uv}$ | Reserved bandwidth required of link $(u,v)$, $R_{uv}=R_{uv}^a+R_{uv}^b$ |
| $R_{uv}^a$ | One part of $R_{uv}$ ,corresponding to active paths |
| $R_{uv}^b$ | One part of $R_{uv}$ ,corresponding to backup paths |
| $R_{pquv}^b$ | A possible value of $R_{uv}^b$, corresponding to the case if link $(p,q)$ fails |
| $L$ | Total number of used links |
| $\eta_{uv}$ | 1 if link $(u,v)$ is in use,0 otherwise |
| $x_{uv}^i$ | 1 if link $(u,v)$ is used by $P_1^i$ of request $i$ ,0 otherwise |
| $y_{uv}^i$ | 1 if link $(u,v)$ is used by $P_2^i$ of request $i$ ,0 otherwise |
| $z_{uvpq}^i$ | 1 if $x_{pq}^i=y_{uv}^i=1$, 0 otherwise |
| $N^+(v)$ | Set of outbound links $N^+(v)=\{(v,u)\mid(v,u)\in A\}$ |
| $N^-(v)$ | Set of inbound links $N^-(v)=\{(u,v)\mid(u,v)\in A\}$ |
| $w_{uv}$ | Weight of link $(u,v)$. 1 if link $(u,v)$ is not in use,0 otherwise |
| $w(P)$ | Weight of path $P$. $w(P)=\sum_{(u,v)\in P} w_{uv}$ |
| $LB$ | A lower bound of REAR-BS. Optimal value of model $P"$. |

Backup sharing case can always achieve more energy savings than no-sharing case. As for the no-sharing case, each link should reserve bandwidth larger than the total demands of requests that use it on the active path or the backup path. For the sharing case, each link should reserve bandwidth larger than the sum of the demands of requests that use it on the active path and the maximum demand of requests that use it on the backup path. Take Fig. 1 for example. This is a network with 6 nodes and 11 links. Assume that each link has capacity of 10M except link $(s_1,t_1)$ which has 20M. Let $\alpha=50\%$ .Consider REAR-BS for two requests: $LSPs=\{(s_1,t_1,5),\ (s_2,t_2,5)\}$.

It is easy to find the optimal solution. The paths are $P_1^1$: $s_1$->$t_1$, $P_2^1$: $s_1$-> $u$ -> $v$ ->$t_1$, $P_1^2$: $s_2$->$t_2$ and $P_2^2$: $s_2$-> $u$ -> $v$ -> $t_2$. 7 links are used with 5M bandwidth consumed respectively. Though link $(u,v)$ is on both backup paths of two requests, it only needs to reserve $\sum\{d_i \mid (u,v)\in P_1^i\}+\max\{d_i \mid (u,v)\in P_2^i\}=\max\{5,5\}=5\text{M}$. However, it needs to reserve $\sum\{d_i \mid (u,v)\in P_1^i \; or \; (u,v)\in P_2^i\}=5+5=10\text{M}$ in the no-sharing case while its available bandwidth is only 10*50%=5M. For the no-sharing case to be possible, one can adjust $P_2^2$ to $s_2$ -> $p$ -> $s_1$ -> $t_1$ -> $q$ -> $t_2$ to make all constraints satisfied. But this uses 2 more links than the sharing case.

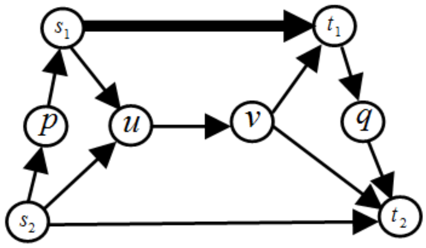



Fig. 1. Example.

Example above considers the situation when active paths are disjoint. It becomes difficult when active paths intersect, i.e., they share common links. If we want to compute $R_{uv}$, the reserved bandwidth required of link $(u,v)$, we should analyze every potential link $(p,q)$ that may be failed. Denote the set of requests whose active path contains $(u,v)$ as $S_{uv}=\{i \mid (u,v)\in P_1^i, i=1,\cdots,D\}$. And denote the set of requests whose active path uses $(p,q)$ and backup path uses $(u,v)$ as $S_{pquv}=\{i \mid (p,q)\in P_1^i, (u,v)\in P_2^i, i=1,\cdots,D\}$. $R_{uv}$ should consist of two parts of reserved bandwidth: $R_{uv}^a$, $R_{uv}^b$. $R_{uv}^a$ corresponds to active paths and $R_{uv}^b$ corresponds to backup paths. Since active paths cannot share bandwidth, $R_{uv}^a$ equals the demands of requests in $S_{uv}$, i.e., $R_{uv}^a=\sum_{i\in S_{uv}} d_i$. For $R_{uv}^b$, consider single link failure case. If link $(p,q)$ is failed, all traffic of requests whose active path contains $(p,q)$ will be switched to backup paths. Thus $R_{uv}^b$ should be larger than bandwidth of requests in $S_{pquv}$. Consider all possible single link failure cases, $R_{uv}^b$ should large than $\max_{(p,q)\in A} R_{pquv}^b=\max_{(p,q)\in A}\sum_{i\in S_{pquv}} d_i$. Hence, the minimal value of $R_{uv}$ is $\sum_{i\in S_{uv}} d_i+\max_{(p,q)\in A}\sum_{i\in S_{pquv}} d_i$.

### *B. The Nonlinear Integer Programming Model*

We formulate a nonlinear programming model $P$ for REAR-BS.

$$\min \sum_{(u,v)\in A} \eta_{uv}=L \qquad P$$

$$s.t. \sum_{v\in N^+(u)} x_{uv}^i-\sum_{v\in N^-(u)} x_{vu}^i =\begin{cases}1, & u=s_i\\ -1, & u=t_i, \forall u\in V, i=1,\cdots,D\\ 0, & \text{else}\end{cases} \qquad (1)$$

$$\sum_{v\in N^+(u)} y_{uv}^i-\sum_{v\in N^-(u)} y_{vu}^i =\begin{cases}1, & u=s_i\\ -1, & u=t_i, \forall u\in V, i=1,\cdots,D\\ 0, & \text{else}\end{cases} \qquad (2)$$

$$x_{uv}^i+y_{uv}^i\le 1, \quad \forall (u,v)\in A, i=1,\cdots,D \qquad (3)$$

$$\sum_{i=1}^{D} x_{uv}^i d_i+\max_{pq\in A}\left\{\sum_{i=1}^{D} x_{pq}^i y_{uv}^i d_i\right\}\le \alpha c_{uv}\eta_{uv}, \forall (u,v)\in A \qquad (4)$$

$$\sum_{v\in N^+(u)} x_{uv}^i\le 1, \; \forall u\in N\setminus\{t_i\}, i=1,\cdots,D \qquad (5)$$

$$\sum_{u\in N^-(t_i)} x_{ut_i}^i\le 1, \; \forall t_i, i=1,\cdots,D \qquad (6)$$

$$\sum_{v\in N^+(u)} y_{uv}^i\le 1, \quad \forall u\in N\setminus\{t_i\}, i=1,\cdots,D \qquad (7)$$

$$\sum_{u\in N^-(t_i)} y_{ut_i}^i\le 1, \qquad \forall t_i, i=1,\cdots,D \qquad (8)$$

$$x_{uv}^i, y_{uv}^i, \eta_{uv}\in\{0,1\} \qquad (9)$$

The goal of model $P$ is to minimize the number of used links $L$. Equation (1) and (2) are flow conservation constraints which represent an active path and a backup path for each request. Equation (3) demonstrates the link disjoint constraint for restoration as only one of $x_{uv}^i$ and $y_{uv}^i$ can equal 1. In other words, link $(u,v)$ can only be used by the active path or the backup path. Equation (4) represents maximum link utilization constraint and also implies backup sharing is allowed. It ensures the reserved bandwidth $R_{uv}$ cannot exceed the available bandwidth $\alpha c_{uv}$. From the analysis of $R_{uv}$ in the above section, we can easily find that $R_{uv}^a=\sum_{i\in S_{uv}} d_i=\sum_{i=1}^{D} x_{uv}^i d_i$ and $R_{uv}^b=\max_{(p,q)\in A}\sum_{i\in S_{pquv}} d_i=\max_{pq\in A}\left\{\sum_{i=1}^{D} x_{pq}^i y_{uv}^i d_i\right\}$. If reserved bandwidths is needed, link $(u,v)$ is in use and $\eta_{uv}=1$. Equations (5~8) ensure that the active path and backup path are acyclic.

This model has a quadratic constraint (4) which can be linearized. Consider the linear version of (4):

$$\sum_{i=1}^{D} x_{uv}^i d_i+\sum_{i=1}^{D}(x_{pq}^i+y_{uv}^i-1)d_i\le \alpha c_{uv}\eta_{uv}.\forall (u,v),(p,q)\in A \qquad (10)$$

It is not equivalent to (4) when $x^i_{pq} = y^i_{uv} = 0$. We introduce 0-1 integer variable $z^i_{pquv}$ to get equivalent form of (4):

$$\begin{cases} \sum_{i=1}^{D} x^i_{uv} d_i + \sum_{i=1}^{D} z^i_{pquv} d_i \le \alpha c_{uv} \eta_{uv}, \forall (u,v),(p,q) \in A \\ z^i_{uvpq} \ge x^i_{pq} + y^i_{uv} - 1, \forall (u,v),(p,q) \in A, i = 1,\cdots,D \end{cases} \quad (11)$$

Thus equations (1~3), (5~11) and $z^i_{pquv} \in \{0,1\}$ (12) constitute a new equivalent model $P'$ with $m + 2Dm + Dm^2$ variables and $4Dn + Dm + Dm^2 + m^2$ constraints.

### C. Problem Analysis

**Difficulty Analysis.** Note that if there is only one request, REAR-BS is equivalent to shortest path pair problem [11] and can be solved exactly in polynomial time by Suurballe algorithm. However, it becomes significantly difficult if the number of requests grows. Assume that there are two requests with equal demands and link capacity can only afford for one request. REAR-BS involves two-path problem [26] as a subproblem in this case. Two-path problem is a Np-complete problem which aims to find a pair of link disjoint paths for two source-terminal pairs composed of four distinct nodes. Thus REAR-BS is Np-hard.

Due to the difficulty of REAR-BS, solving model $P'$ for optimal solutions via CPLEX becomes impractical even for small networks. Thus we propose heuristic algorithm to tackle this problem.

**Lower Bound.** Denote the optimal solution of model $P'$ as $L_{opt}(P')$ .We give a lower bound of $L_{opt}(P')$ for comparison in case that CPLEX is impractical. There are 3 potential ways to get the lower bound.

1. Relax all the integrality constraints of model $P'$. This is an intuitive lower bound of model $P'$ via linear relaxation. However, relaxing constraint $(4''')$ results in a very small value of $z^i_{pquv}$ and $\eta_{uv}$. Thus the lower bound is too small which far away from the optimal $L_{opt}(P')$.

2. Relax a part of the integrality constraints of model $P'$. For example, let $z^i_{pquv} \in [0,1]$. However, the new model is a mixed integer programming model (MIP). Generally speaking, MIP is NP-hard. Thus the new model may be impractical for CPLEX.

3. Relax the capacity constraint. Assume that the link capacity is infinite. The new model $P''$ for REAR-BS can be written as follows. Though this is an integer model, we prove that the coefficient matrix is unimodular [27].

$$\min \sum_{(u,v)\in A} \eta_{uv} = L \qquad P''$$

$$\begin{aligned} s.t.\quad & (1), (2), (9) \\ & x^i_{uv} + y^i_{uv} \le \eta_{uv}, \quad \forall (u,v) \in A, i = 1,\cdots,D \end{aligned} \qquad (13)$$

**Theorem 1:** The coefficient matrix of model $P''$ is unimodular.

*Proof:* Let $X = (x^1_{uv},\cdots,x^2_{uv},\cdots,x^D_{uv},\cdots,y^1_{uv},\cdots,y^D_{uv},\cdots,\eta_{uv},\cdots)$. The model $P''$ can be written as $\min c^T X \ s.t. AX \ge b, X \ge 0$, where $A$ is the coefficient matrix and $b,c$ are integer column vectors.

$$A = \begin{pmatrix} A_1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & A_D & & & & & & \\ & & & A_1 & & & & & \\ & & & & \ddots & & & & \\ & & & & & A_D & & & \\ -E_1 & & & -E_1 & & & E_1 & & \\ & \ddots & & & \ddots & & & \ddots & \\ & & -E_D & & & -E_D & & & E_D \end{pmatrix},$$

where $A_i$ is an incidence matrix and $E_i$ is an identity matrix, $\forall i = 1,\cdots,D$. Note that $A_i$ is unimodular [27], thus the determination of every basis matrix of $A_i$ has value 1 or -1. Denote $A1_i$ as a basis matrix of $A_i$. For $A$, $A$ can be transformed to $A'$ via elementary column transformation and to $A'$ via elementary row and column transformation as follows.

$$A \to A' = \begin{pmatrix} A_1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & A_D & & & & & & \\ & & & A_1 & & & & & \\ & & & & \ddots & & & & \\ & & & & & A_D & & & \\ 0 & & & 0 & & & E_1 & & \\ & \ddots & & & \ddots & & & \ddots & \\ & & 0 & & & 0 & & & E_D \end{pmatrix},$$

$$\to A'' = \begin{pmatrix} A1_1 & & & & & & & \\ & \ddots & & & & & & \\ & & A1_D & & & & & \\ & & & E_1 & & & & \\ & & & & \ddots & & & \\ & & & & & E_D & & \\ & & & & & & 0 & \\ & & & & & & & \ddots & \\ & & & & & & & & 0 \end{pmatrix}$$

Thus every basic matrix $B$ of $A$ can transformed to a matrix shaped like a diagonal matrix which is composed by $A1_i$ and $E_i$. And $|\det(B)| = |\prod_{i=1}^{D} |\det(A1_i)| \prod_{i=1}^{D} |\det(E_i)||$ $= 1$. So the determination of every basis matrix of $A$ has value 1 or -1. Thus $A$ is unimodular.

Therefore, the linear relaxation of model $P''$ has integer optimal solution for integer vector $b$ [27]. So we can get lower bound of $P'$ by solving the linear relaxation of $P''$. We denote this lower bound as $LB$ and choose it as a comparison in Section 5. Noting that $LB$ equals the optimal value of REAR-BS if the link capacity is large enough.

**GreRA** $(G(N,A), LSPs, C, \alpha)$
**%Part 1: Initialization**
Sort requests according to priority order;
Denote the new order as $1,\cdots,D$;
Set $L=0$, $Paths=\phi$, $w_{uv}=1$, $c_{uv}^{'}=\alpha c_{uv}$,
$R_{uv}^{a}=R_{uv}^{b}=R_{pquv}^{b}=0$, $\forall(u,v),(p,q)\in A$;
**% Part 2: Find active path** $P_1^i$ **for request** $i$, $i=1,\cdots,D$
**For** $i=1,\cdots,D$
Construct $G_a(N_a,A_a,w(\cdot))$ by deleting links in $G$ if $d_i<c_{uv}^{'}$;
Call Dijkstra algorithm to generate $P_1^i$ with weight $w_{uv}$;
$R_{pq}^{a}=R_{pq}^{a}+d_i$, $\forall(p,q)\in P_1^i$
Call ***Update Process*** $(L,w,P_1^i)$
**End For**
**% Part 3: Find active path** $P_2^i$ **for request** $i$, $i=1,\cdots,D$
**For** $i=1,\cdots,D$
$R_{pquv}^{b'}=R_{pquv}^{b}+d_i$, $\forall(p,q)\in P_1^i,(u,v)\in A$; (10)
$R_{uv}^{b'}=\max_{(p,q)\in A}R_{pquv}^{b'}=\max\left(R_{uv}^{b},\max_{(p,q)\in P_1^i}R_{pquv}^{b'}\right)$, $\forall(u,v)\in A$; (11)
Construct $G_b(N_b,A_b,w(\cdot))$ by deleting links in $P_1^i$ and links in $G$ if $R_{uv}^{a}+R_{uv}^{b'}>\alpha c_{uv}$;
Call Dijkstra algorithm to generate $P_2^i$ with weight $w_{uv}$;
$R_{pquv}^{b}=R_{pquv}^{b'}$, $R_{uv}^{b}=R_{uv}^{b'}$, $\forall(p,q)\in P_1^i,(u,v)\in P_2^i$;
Call ***Update Process*** $(L,w,P_2^i)$
$P^i=\{P_1^i,P_2^i\}$;
**EndFor**

Fig. 2. Algorithm GreRA.

## IV. HEURISTIC ALGORITHM

Note that Suurballe algorithm can solve REAR-BS exactly if there is only one request. When there are many requests, Suurballe algorithm is not applicable to REAR-BS because of backup sharing even if these requests are processed one by one. Thus, we propose **Green Restorable Algorithm (GreRA)** to solve REAR-BS without considering Suurballe algorithm.

GreRA is based on the analysis of the amount of bandwidth that can be shared for each link and the idea of incremental minimization. We use a weight variable $w_{uv}$ to characterize the contribution of link $(u,v)$ to the increment of total used links. The value of $w_{uv}$ changes from initial value 1 to 0 once link $(u,v)$ is being used. Namely, using the link again does not cause any increase of the goal value $L$. This is very helpful in the computation of incremental minimization. The algorithm is shown in Fig. 2.

**Update Process** $(L,w,P)$
$L=L+w(P)$;
$Paths=Paths\cup P$;
$w_{uv}=0, c_{uv}^{'}=\alpha c_{uv}-RB_{uv}^{a}-RB_{uv}^{b}$, $\forall(u,v)\in P$;

Fig. 3. Functionupdate process.

GreRA consists of three parts. Part 1 is an initialization process which starts by considering all requests unarranged. Hence for initial values $w_{uv}=1$, $R_{uv}^{a}=R_{uv}^{b}=R_{pquv}^{b}=0$, $\forall(u,v),(p,q)$. The initial value of $w_{uv}$ means a link to be newly used will make goal value $L$ increase 1. Part 2 is a process that finds the active path for each request which is very similar to Shortest Path routing algorithm except for the variable weight $w_{uv}$ in using Dijkstra algorithm. In the **Update Process** (Fig. 3), $w_{uv}$ can be changed to 0 once the link is used. This ensures a link being used does not increase the goal value any more. Through Dijkstra algorithm, we can select a path with minimal increment in weight. Namely, the chosen active path reuses used links as many as possible. For the active path, $R_{uv}^{a}$ is changed thus $c_{uv}^{'}$ needs to update. Finally, Part 3 finds the backup path for each request one by one. This part is more difficult than part 2 as backup sharing is allowed here. We use $R_{pquv}^{b'}$ and $R_{uv}^{b'}$ to represent the temporary values of $R_{pquv}^{b}$ and $R_{uv}^{b}$ if link $(p,q)$ which is on active paths fails, respectively. When computing the backup path $P_2^i$ for request $i$, we need to find all candidate links for $P_2^i$ which satisfy constraint $R_{uv}^{a}+R_{uv}^{b'}\le\alpha c_{uv}$.Thus we need to get correct value of $R_{uv}^{b'}$ if link $(u,v)$ is chosen to be on $P_2^i$. Note that $R_{uv}^{b}=\max_{(p,q)\in A}R_{pquv}^{b}=\max_{(p,q)\in A}\sum_{i\in S_{pquv}}d_i$ and $S_{pquv}$ is now increased by a new element, request $i$. Hence only the failures on the active paths $P_i^1$ should be considered. That is, $R_{pquv}^{b'}=R_{pquv}^{b}+d_i$, $\forall(p,q)\in P_1^i,(u,v)\in A$. And $R_{uv}^{b'}$ can be got by intelligent comparison instead of all possible values of $R_{pquv}^{b'}$ because only the values on $P_1^i$ are changed, i.e., $R_{uv}^{b'}=\max_{(p,q)\in A}R_{pquv}^{b'}=\max\left(R_{uv}^{b},\max_{(p,q)\in P_1^i}R_{pquv}^{b'}\right)$. When all candidate links are found, we can compute $P_2^i$ using Dijkstra. Update $R_{pquv}^{b}$ and $R_{uv}^{b}$ if $P_2^i$ exists. Then call function Update Process as in Part 2. Repeat the same process until all requests are done. Finally, unused links $\{(u,v)\,|\,(u,v)\notin Paths\}$ are put to sleep for energy saving.

For the implementation of GreRA of REAR-BS in SDNs, the controller first collects network information, traffic matrixand users' requests. Then the controller runs GreRA to find the minimum subset of powered on links for REAR-BS using the collected information. It disseminates a control message of a sleeping list of idle links to routers after all active and backup paths are set up. Thus idle links will be put to sleep.

We name the algorithm substituting the underline sentences to 'equal link weight 1' in Part 2 and Part 3 **Shortest Path routing with Backup Sharing(SP-BS)** for comparison purpose.

***Time Complexity:*** Dijkstra algorithm can be solved in $O(n^2)$ time. Thus, Part 2 requires $O(Dn^2)$. In the worst case, the length of $P_1^i$ is $O(n)$. Equation (10) and (11) needs $O(nm)$ time. Thus Part 3 requires $O(Dnm)$ time. Therefore, the time complexity of GreRAis $O(Dnm)$. So it is for SP-BS.

Though the number of nodes is doubled and the number of links increases by $n$ via node splitting, the time complexity of GreRA still holds for the node disjoint version of REAR-BSbecause $O(D2n(n+m))=O(Dnm)$.

## V. Evaluation

In this section, we show the effectiveness of GreRA on energy saving when compared to SP-BS, CPLEX [28]and $LB$.

### A. Experimental Setup

We have run extensive simulations on 3 real networks with real and synthetic traffic matrices to explore the energy savings of GreRA. These networks include Internet2 [29],GÉANT [30]and Sprint [31].

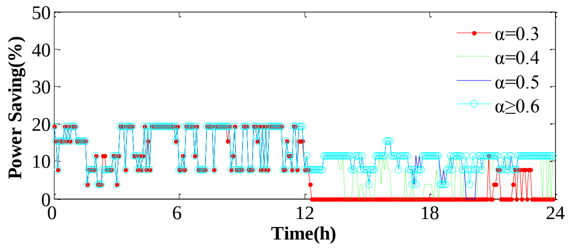
TABLE II: Network Topologies

| Network | Nodes | Links | Access Nodes | Transit Nodes | Requests Number |
|---|---|---|---|---|---|
| Internet2 | 9 | 26 | 9 | 0 | 66 |
| GÉANT_S | 23 | 74 | 11 | 12 | 110 |
| GÉANT | 23 | 74 | 23 | 0 | 506 |
| Sprint | 44 | 166 | 22 | 22 | 462 |

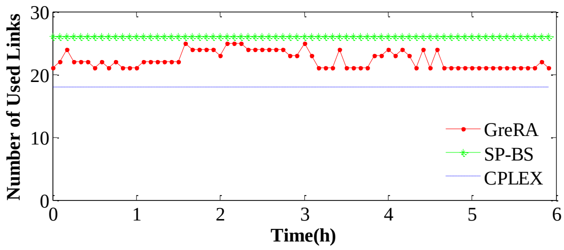
We obtained the topology of Internet2 and 288 traffic matrices measured on August 21st, 2008 at an interval of 5 minutes from the author of [2]. For GÉANT, the topology and 96 traffic matrices measured on May 5th, 2005 every 15 minutes are from [30]. The first 72 traffic matrices of Internet2 and first 24 traffic matrices of GÉANT corresponding to 0:00~6:00 are used for the off-peak hours. The PoP-level topology of Sprint is derived from Rocketfuel [31]. We set link capacity using the method of [32] and generate 10 traffic matrices using gravity model [33], [34] such that the maximum link utilization is 10%~100% , with an increment of 10% when SP routing are used to route traffic. Denote these traffic matrices as TM_1~TM_10.

Two kinds of nodes are considered here: access nodes and transit nodes. Requests considered always have 2 link disjoint paths. This can be easily achieved through running Suurballe Algorithm for every node pair. Only node pairs which have 2 link disjoint paths can be access nodes, i.e., sources or terminals of requests. Since CPLEX may be impractical to solve model $P'$ for REAR-BS, we random select some nodes as access nodes for GÉANT and Sprint to reduce the scale of requests.Table II shows these details where GÉANT_S is a copy of GÉANT with different access nodes. We assume that each link has equal power consumption and use $PS=L/m$ to calculate the power saving rate.

We use SP-BS as a fair comparison in our evaluation as there is no research considering backup sharing with energy saving. Though CPLEX is a high-performance mathematical programming solver, it cannot ensure a feasible solution in limited time (even 24h). So we set a limit of 300s (excludes preprocessing time) when using CPLEX. We also compare GreRA with $LB$ to indicate the difference from optimal solution in case CPLEX fails. Simulations are performed on a Lenovo PC with 2.5GHz CPU and 2GB RAM using Matlab.




(a) GreRA




(b) $\alpha=0.4$

Fig. 4. Energy saving on Internet2. (a) $PS$ underGreRA (b) The number of used links under GreRA, SP-BS and CPLEX when $\alpha=0.4$

### B. Energy Saving

Fig. 4(a) shows the energy savings for Internet2 under GreRA when maximum link utilization threshold $\alpha$ changes from 0.1 to 1. All source-terminal requests are considered. GreRA is not able to put any link to sleep when $\alpha\le0.2$ due to a small amount of available bandwidth. When $\alpha=0.3$, GreRA can save 14.32% energy on average during 0h~12h. It can hardly put links to sleep during 12h~24h because of heavy traffic. The larger $\alpha$ is, the more available bandwidth becomes. Thus, the value of $PS$ grows with the increase of $\alpha$. When $\alpha=0.4$, GreRA behaves better than that of $\alpha=0.3$ during 12:00~24:00 due to larger available bandwidth. The case of $\alpha=0.5$ is similar with the case of $\alpha>=0.6$. They are only different at some $PS$ values around 18h. For $\alpha>=0.6$, GreRA generate the same set of active and backup paths and performs the same on $PS$. So does GreRA for $\alpha\ge0.4$ during 0h~12h, particularly. It can save up to 19.23% energy of links.

TABLE III: AVERAGE PERFORMANCE ON ALL NETWORKS FOR $\alpha = 1$

| | *PS* (%) | | | | Time(s) | | | |
|---|---|---|---|---|---|---|---|---|
| Network | GreRA | SP-BS | CPLEX | *LB* | GreRA | SP-BS | CPLEX | *LB* |
| Internet2 | 14.05 | 0 | 30.77* | 30.77* | 0.08 | 0.07 | 74.35 | 0.14 |
| GÉANT_S | 47.47 | 16.61 | 0.96 | 58.33 | 0.30 | 0.29 | 588.94 | 3.37 |
| GÉANT | 14.58 | 0 | -- | 32.43 | 1.49 | 1.45 | -- | 3.46 |
| Sprint | 49.34 | 20.48 | -- | 65.06 | 4.93 | 4.89 | -- | 8.54 |

Note: '*': optimal; '--': out of memory.

We compare the number of used links under GreRA, SP-BS and CPLEX during 0h~6h when $\alpha = 0.4$ to further investigate the performance of GreRA during off-peak hours. Note that GreRA performs the same for $\alpha \geq 0.4$ during this off-peak period. So do SP-BS and CPLEX. Thus we only show the result for $\alpha = 0.4$. This low value for maximum link utilization threshold implies the great benefit of restoration for reliability and energy saving. Though two link disjoint paths needs more links, backup sharing contributes to that resource consumption is not that much.As shown in Fig. 4(b), CPLEX is the best performer as it computes the optimal value of $L$, 18 links. SP-BS performs the worst because all traffic requests are considered. And Internet2 is a small topology that leaves less choice for routing paths. SP-BS prefers the shortest paths and it uses all links up. SP-BS does not apply to the case of all source-terminal pairs. GreRA is in the middle of the performance. It uses 21 links at best and uses 22 links on average. The average number of $L$ ofGreRA is 4 links more than optimal value. Though CPLEX solutions are optimal, it consumes much more time than GreRA. The running time of GreRA is less than 0.1s while that of CPLEX is 74.35s on average, as shown in Table III.

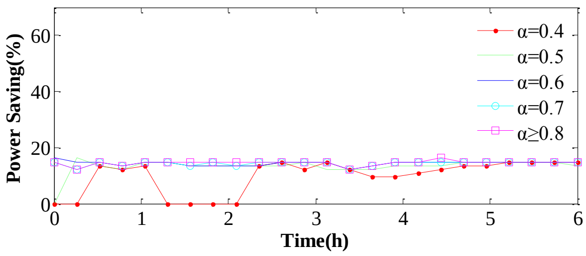



Fig. 5. Energy saving on GÉANT under GreRA.

Fig. 5 shows the performance of GreRA on GÉANT with all traffic demands when $\alpha$ changes from 0.1 to 1. Though we choose the off-peak period during 1h~6h, GreRA cannot put any link to sleep for $\alpha \leq 0.3$ due to insufficient available capacity. When $\alpha = 0.4$, it is still lack of available bandwidth for all traffic demands during 1h~2h. The $PS$ value shows an increasing trend as $\alpha$ becomes larger. GreRA shows little difference in $PS$ for $0.5 \leq \alpha \leq 1$. The $PS$ value is around 15%. In particular, GreRA produces the same energy saving rate for $\alpha \geq 0.8$. And its energy saving rate is 14.58% on average. GreRA only uses about 17% additional links than $LB$ as shown in Table III.

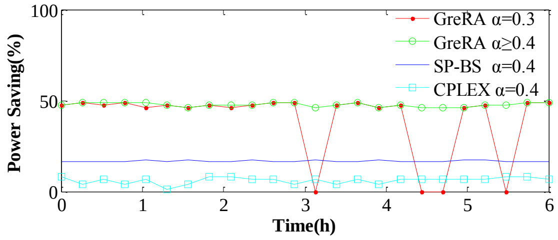
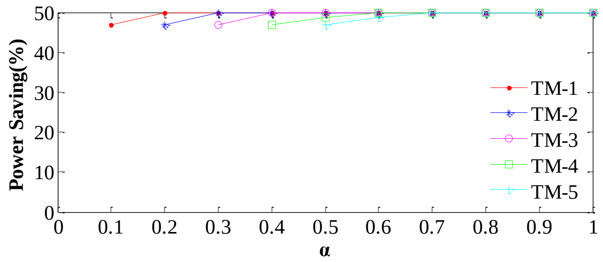



Fig. 6. Energy saving on GÉANT_S under GreRA , SP-BS and CPLEX

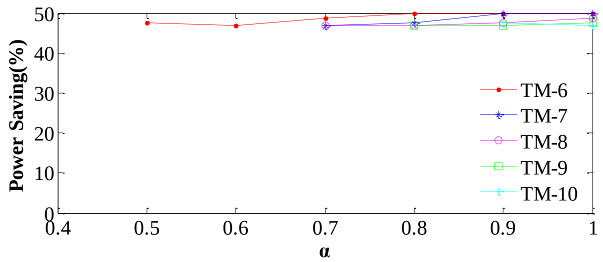
Because all traffic demands are considered, SP-BS uses up all 74 links for $\alpha \geq 0.4$ and achieve no energy saving. So we omit its performance here. Traffic demands from 23 nodes makes a large scale for CPLEX, thus it becomes impractical for CPLEX. CPLEX cannot afford to give a feasible solution in 24h for this case. In order to give a fair comparison, we evaluate the performance of GreRA, SP-BS and CPLEX for GÉANT_S with fewer traffic demands, as shown inFig. 6. Note that the number of requests of GÉANT_S is only approximately a quarter of that of GÉANT. Unlike for GÉANT, GreRA can put about 50% links to sleep for most cases for GÉANT_S when $\alpha = 0.3$. And GreRA achieve the same $PS$ for $\alpha \geq 0.4$. We compare the performance of three kinds of solutions when $\alpha \geq 0.4$. As shown in Fig. 6, they have the same curve for $\alpha \geq 0.4$, respectively. CPLEX with a limited solving time 300s performs the worst with $PS$ below 10%. This illustrates that CPLEX is not practical for NP-hard problems even at a medium scale. SP-BS is in the middle with $PS$ around 16%. GreRA is the best performer with about 47% energy saving rate which is 3 times as large as that of SP-BS. What's more, GreRA consumes less than 0.3s for GÉANT_S with $PS$ 8 times larger than that of CPLEX. GreRA only uses 10.86% additional links than $LB$. As the network provides more sufficient bandwidth with less requests, $LB$ becomes more close to the optimal value. Thus GreRA is more closeto optimal in GÉANT_S than in GÉANT.

Fig. 7 presents the energy saving on Sprint under GreRA and SP-BS with 10 traffic matrices TM_1~TM_10. Fig. 7 (a) and (b) show the $PS$ under GreRA with $\alpha$ changing from 0.1 to 1. The smaller $\alpha$ and lower traffic demands result in insufficient bandwidth. Under this circumstance, GreRA may fail to switch any links. For example, GreRA cannot put links to sleep for TM_2 when $\alpha = 0.1$. So does for TM_6 when $\alpha \leq 0.4$. GreRA shows an increasing trend as $\alpha$ becomes larger.

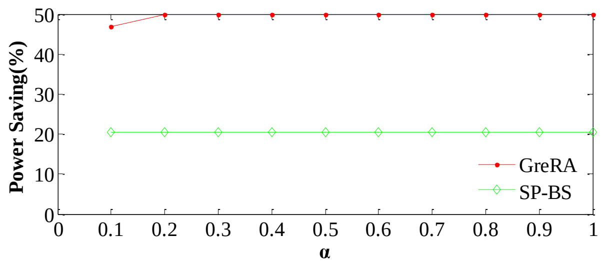
When there are sufficient capacity, GreRA can save up to 50% energy. The topology of Sprint is too large for CPLEX to solve, so we use SP-BS as the only comparison. Fig. 7 (c) shows the comparison of GreRA and SP-BS on TM_1. The results on other traffic matrices are similar to that of TM_1, so we omit it. SP-BS can only save 20.5% energy while GreRA achieves about 50% for TM_1 with similar running time 4.9s. For Sprint, GreRA uses about 15.7% additional links than *LB* as shown in Table III.




(a) GreRA




(b) GreRA



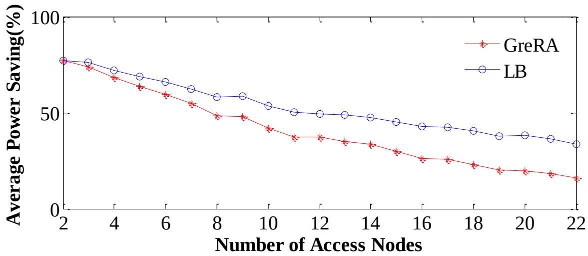

(c) TM_1

Fig. 7. Energy saving on Sprint under GreRA and SP-BS with different $\alpha$. (a) *PS* underGreRA for TM_1~TM_5 (b) *PS* under GreRA for TM_6~TM_10 (c) *PS* under GreRA and SP-BS for TM_1.

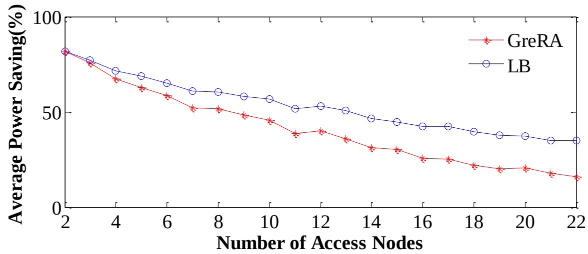
### C. *GreRA vs Lower Bound*

As CPLEX becomes impractical to solve the optimal value of GÉANT and Sprint, we compare GreRA with the lower bound *LB* with different number of access nodes when $\alpha = 1$ to further investigate the difference on the performance of *PS* between GreRA and optimal solution. We use the traffic matrix at 0h for GÉANT and TM_1 for Sprint with the number of random access nodes changing from 2 to 22. We random generate 20 instances for each case and average the results over these instances.

As shown in Fig. 8(a), the difference between GreRA and *LB* on *PS* gets larger as the number of access nodes increases for GÉANT. Fig. 8(b) shows the same trend for Sprint. The fewer the access nodes are, the more sufficient the link capacity will be. Note that *LB* equals the optimal solution of REAR-BS when the bandwidth is large enough. So *LB* is more close to optimal solution with fewer access nodes. The *PS* of GreRA is very near that of *LB* with the number of access nodes below 10. Particularly, GreRA is optimal with 2 access nodes and almost optimal with 3~5 access nodes. Moreover, GreRA uses about 15% additional links than *LB* when the number of access nodes is larger than 10. Though both GreRA and *LB* runs very fast, the running time of GreRA is only about half of *LB*'s. Please see 0for reference. On average, GreRA uses 10.68% additional links for GÉANT and 11.65% additional links for Sprint when compared to *LB*. Thus, the difference between GreRA and the optimal solution will be much smaller. This demonstrates the effectiveness of GreRA.




(a) GÉANT




(b) Sprint

Fig. 8. Energy saving under GreRA and *LB* with different access nodes when $\alpha = 1$. (a) *PS* forGÉANT on traffic matrix at 0h (b) *PS* for Sprint on TM_1.

From the simulations above, we can draw a conclusion that GreRA is a very efficient and effective algorithm for REAR-BS. It performs perfectly better than SP-BS on energy saving rate and takes much less time than CPLEX. More importantly, the comparison with lower bound shows that GreRA is near optimal.

## VI. Conclusion

In this paper, we first take restoration into account while energy saving. The logically centralized feature of SDN lays the foundation for restoration with backup sharing. To this end, we consider the problem called REAR-BS in SDN during off-peak hours to minimize the number of used links subject to the following constraints: maximum link utilization, a pair of disjoint paths for each request and backup sharing. We formulate a nonlinear integer model and its linearized version and give a lower bound for the optimal solution of this NP-hard problem. Then, we propose a heuristic GreRA to handle this problem which performs significantly better than Shortest Path routing with backup sharing and consumes less time than CPLEX. The energy savings can be up to 50% during off-peak hours. And GreRA uses no more than 15% additional links as compared to the lower bound of optimal solution. We plan to give a tighter lower bound for REAR-BS and further consider reliable energy aware routing with delay constraint as a future work.

## REFERENCES


[1] M. Pickavet, W. Vereecken, S. Demeyer, P. Audenaert, B. Vermeulen, *et al.*, "Worldwide energy needs for ICT: The rise of power-aware networking," in *Proc. IEEE International Symposium on Advanced Networks & Telecommunication Systems*, Mumbai, India, 2008, pp. 1-3.

[2] M. Zhang, C. Yi, B. Liu, and B. Zhang, "GreenTE: Power-aware traffic engineering," in *IEEE ICNP*, Kyoto, Japan, 2010, pp. 21-30.

[3] M. Gupta and S. Singh, "Greening of the internet," in *Proc. ACM SIGCOMM*, Karlsruhe, Germany, 2003, pp. 19-26.

[4] L. Chiaraviglio, M. Mellia, and F. Neri, "Reducing power consumption in backbone networks," in *Proc. IEEE ICC*, Dresden, Germany, 2009, pp. 2298-2303.

[5] M. Kodialam and T. V. Lakshman, "Dynamic routing of bandwidth guaranteed tunnels with restoration," in *Proc. IEEE INFOCOM*, Tel-Aviv, Israel, 2000, pp. 902-911.

[6] Y. Bejerano, Y. Breitbart, A. Orda, R. Rastogi, and A. Sprintson, "Algorithms for computing QoSpaths with restoration," *IEEE/ACM Transactions on Networking*, vol. 13, no. 3, pp. 648-661, June 2005.

[7] M. Kodialam and T. V. Lakshman, "Restorable dynamic quality of service routing," *IEEE Communications Magazine*, vol. 40, no. 6, pp. 72 - 81, June 2002.

[8] K. Kar, M. Kodialam, and T. V. Lakshman, "Routing restorable bandwidth guaranteed connections using maximum 2-route flows," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 772-781, Oct. 2003.

[9] D. Levin, A. Wundsam, B. Heller, N. Handigol, and A. Feldmann, "Logically centralized? State distribution trade-offs in software defined networks," in *Proc. First Workshop on Hot SDN, ACM SIGCOMM*, 2012, pp. 1-6.

[10] B. A. A. Nunes, M. Mendonca, X. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1617-1634, Aug. 2014.

[11] J. W. Suurballe and R. E. Tarjan, "A quick method for finding shortest pairs of disjoint paths," *Networks*, vol. 14, no. 2, pp. 325-336, 1984.

[12] G. Lin, S. Soh, K. Chin, and M. Lazarescu, "Energy aware two disjoint paths routing," *Journal of Network and Computer Applications*, vol. 43, pp. 27-41, Apr. 2014.

[13] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power awareness in network design and routing," in *Proc. 27th Conference on Computer Communications*, Phoenix, AZ, 2008, pp. 1130-1138.

[14] L. Chiaraviglio, M. Mellia and F. Neri, "Minimizing ISP network energy cost: Formulation and solutions," *IEEE/ACM Transactions on Networking*, vol. 20, no. 2, pp. 463-476, Apr. 2012.

[15] W. Fisher, M. Suchara, and J. Rexford, "Greening backbone networks: Reducing energy consumption by shutting off cables in bundled links," in *Proc. ACM SIGCOMM Workshop on Green Networking*, New Delhi, India, 2010, pp. 29-34.

[16] F. Giroire, J. Moulierac, T. K. Phan, and F. Roudaut, "Minimization of network power consumption with redundancy elimination," *Computer Communications*, vol. 59, pp. 98-105, Mar. 2015.

[17] B. G. Assefa and O. Ozkasap, "State-of-the-art energy efficiency approaches in software defined networking," presented at The Fourteenth International Conference on Networks, Barcelona, Spain, Apr. 19-24, 2015.

[18] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, *et al.*, "ElasticTree: Saving energy in data center networks," in *Proc. USENIX NSDI*, San Jose, CA, 2010, pp. 249-264.

[19] F. Giroire, J. Moulierac, and T. K. Phan, "Optimizing rule placement in software-defined networks for energy-aware routing," in *Proc. IEEE GLOBECOM*, Austin, TX, 2014, pp. 2523-2529.

[20] R. Wang, Z. Jiang, S. Gao, W. Yang, Y. Xia, and M. Zhu, "Energy-aware routing algorithms in Software-Defined Networks," presented at IEEE WoWMoM, Sydney, NSW, 2014, 2014, pp. 1-6.

[21] A. Markiewicz, P. N. Tran, and A. Timm-Giel, "Energy consumption optimization for software defined networks considering dynamic traffic," in *Proc. IEEE 3rd International Conference on Cloud Networking*, Luxembourg, 2014, pp. 155-160.

[22] Y. Guo, F. Kuipers, and P. Van Mieghem, "Link-disjoint paths for reliable QoS routing," *International Journal of Communication Systems*, vol. 16, no. 9, pp. 779-798, June 2003.

[23] J. Y. Yen, "Finding the K shortest loopless paths in a network," *Manag Sci.*, vol. 17, no. 11, pp. 712-716, 1971.

[24] G. Lin, S. Soh, M. Lazarescu, and K. Chin, "Reliable green routing using two disjoint paths," in *Proc. IEEE ICC*, Sydney, NSW, 2014, pp. 3727-3733.

[25] Y. Perl and Y. Shiloach, "Finding two disjoint paths between two pairs of vertices in a graph," *Journal of the Association for Computing Machinery*, vol. 25, no. 1, pp. 1-9, Jan. 1978.

[26] S. Fortune, J. Hopcroft, and J. Wyllie, "The directed subgraph homeomorphism problem," *Theoretical Computer Science*, vol. 10, no. 2, pp. 111-121, 1980.

[27] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, USA: Prentice Hall, 1993, ch.11, pp. 447-449.

[28] CPLEX. [Online]. Available: http://www.ilog.com/products/cplex/

[29] Internet2. [Online]. Available: http://www.internet2.edu/

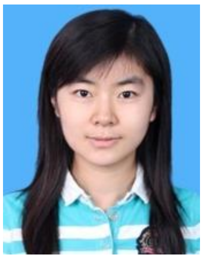
[30] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon, "Providing public intradomain traffic matrices to the research community," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 83-86, Jan. 2006.

[31] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, "Measuring ISP topologies with rocketfuel,"*IEEE/ACM Transactions on Networking*, vol. 12, no. 1, Feb. 2004.

[32] S. Kandula, D. Katabi, B. Davie, and A. Charny, "Walking the tightrope: Responsive yet stable traffic engineering," in *Proc. Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Philadelphia, USA, 2005, pp. 253-264.

[33] M. Roughan, A. Greenberg, C. Kalmanek, M. Rumsewicz, J. Yates, and Y. Zhang, "Experience in measuring internet backbone traffic variability: Models metrics, measurements and meaning," in *Proc. ACM SIGCOMM Internet Measurement Workshop*, Marseille, France, 2002, pp. 91-92.

[34] D. Applegate and E. Cohen, "Making intra-domain routing robust to changing and uncertain traffic demands: Understanding fundamental tradeoffs," in *Proc. Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Karlsruhe, Germany, 2003, pp. 313-324.



**Rui Wang** is currently working toward the Ph.D. degree in operational research and control theory in University of Chinese Academy of Sciences, Beijing, China. Her research interests are communication network optimization and energy aware routing in Software Defined Networks.

**Suixiang Gao** received the Ph.D. degree in Mathematics from Institute of Applied Mathematics of Chinese Academy of Sciences in 1998. Now, he is a Professor of School of Mathematics in University of Chinese Academy of Sciences, Beijing, China. His research interests are optimization theory and algorithms, communication networks optimization, graph theory and network flows.

**Wenguo Yang** was born in 1974. He received the M.A's. degree in operation research and control theory from Beijing Jiaotong University in 2003 and the Ph.D. degree from University of the Chinese Academic of Sciences in 2006, respectively. Now, he is an Associate Professor in University of Chinese Academy of Sciences. His research interests include robust optimization, traffic network flow analysis, optimization methods, wireless sensor network and emergency management.

**Zhipeng Jiang** received his Ph.D. degree in operational research and control theoryfrom University of Chinese Academy of Sciences in 2011. He is now a lecture in University of Chinese Academy of Sciences.His research interests aremobile communication network optimization, smart grid and Software Defined Networking