An Efficient Heuristic for Restorable Energy Aware Routing in Networks with Bundled Links

Rui Wang*, Suixiang Gao, Wenguo Yang, Zhipeng Jiang
School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, China
Email: wangrui11@mails.ucas.ac.cn Tel:+86-15501270878; {sxgao, yangwg, jiangzhipeng}@ucas.ac.cn

Abstract— **Energy aware routing via rerouting traffic or traffic aggregating has received wide attention in the last decade. It is important and challenging to take network reliability into consideration to enhance the resilience to failures at the same time. In this paper, we aim to minimize energy expenditure while ensuring network reliability by restoration. As each request has an active path and disjoint backup path for restoration, the backup path can be used if the active path fails. To this end, we first propose the problem called Restorable Energy Aware Routing Problem in networks with bundled links (REAR-BL) when backup sharing is not allowed. We form a linear integer model to minimize the number of powered-on cables of links while putting idle cables to sleep under constraints of restoration and maximum link utilization. Since REAR-BL is Np-hard, we design a fast heuristic to solve it based on Suurballe algorithm and the analysis of a lower bound and an upper bound of the optimal solution. We evaluate our heuristic through extensive simulations on real and synthetic topologies and obtain considerable energy savings with less time when compared to CPLEX solutions and upper and lower bounds.**

Index Terms — **energy aware routing, restoration, link disjoint, linear integer programming, backup path, upper and lower bounds.**

I. INTRODUCTION

Reducing energy consumption has been attracted increasing concern due to the massive amounts (around 10%) of worldwide energy consumption of Information and Communication Technology (ICT) sector [1]. Today's networks are designed with redundant links and over-provisioning link bandwidth to handle peak traffic in the most reliable way without consideration of energy saving. Moreover, the average link utilization in backbone networks is estimated to be around 30~40% [2]. All these provide the potential for saving energy. Therefore, researchers spare no effort to design green networking strategies [3-9] on networking devices like IP (Internet Protocol) routers which consume the largest majority of energy.

Green networking research starts with the seminal work of Gupta and Singh [3] and increases tremendously later [4-6] . Ref. [4] explores the potential power saving in the network design and routing protocols in wire-line networks. Ref. [5, 6] present a model based on MCMF (multi-commodity minimum cost flow problems) to minimize the power consumption of used routers and links. Similarly, Ref. [7] tries to maximize the energy savings of idle links with consideration of load balancing by rerouting traffic and putting idle links into sleep. Ref. [8] considers saving energy where each link is bundled with multiple cables. They formulate an integer linear programming model and propose heuristic algorithms based on linear relaxation to solve it. Ref. [9] considers the same problem in an unsplittable way with a heuristic based on Dijkstra and Yen's k-shortest paths algorithm.

However, for the above green networking works, idle links or routers are put to sleep, which leads to the least number of active links to carry concentrated traffic. Thus the network becomes vulnerable to link failures or sudden traffic bursts. Therefore, it is important to ensure network reliability while saving energy. Taking restoration into account can be a choice as restoration is widely used to ensure resilience to failures [10-13]. For restoration, each source-to-terminal request has an active path and a link disjoint backup path. The traffic is immediately redirected to the backup path only if there is a single link failure on the active path. These works on restoration consider minimizing the total bandwidth usage with only one request. In this paper, we consider saving energy while using restoration with a lot of requests when backup sharing is not allowed.

The combined work of saving energy and restoration is in its infancy. The only related work is shown in [14] where Energy Aware Two Disjoint Paths Routing is proposed. In this problem, each request can be routed via multiple paths besides 2 disjoint paths, i.e., in a splittable way, which is different from restoration. In contrast, we focus on the unsplittable case in wired networks.

The main contributions of this paper are as follows. We address the problem called Restorable Energy Aware Routing in Networks with Bundled Links (REAR-BL) to reduce energy consumption during off-peak hours. We formulate a linear integer programming model for this NP-hard problem to minimize the number of used cables under constraints of restoration without backup sharing and maximum link utilization. Then, we give an upper and lower bound for the optimal solution and design a heuristic to solve REAR-BL, which is based on Suurballe algorithm [15]. Exhaustive simulations confirm the effective of our heuristic. Particularly,

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.

it can reduce energy usage by more than 50% with bundled size larger than 5.

II. Problem Statement and Mathematical Model

In this section, we propose REAR-BL to increase the resilience to link failures. We assume that there is only a single link failure at a time in the network. And requests considered have two disjoint paths.

A. Problem Statement

Consider a network that is represented by a directed graph $G(V,A)$, where V is the set of n nodes (i.e., routers) and A is the set of m links. Each link (u,v) consists of $N_{uv}(\geq 1)$ cables. N_{uv} is the bundle size of link. Denote the capacity/bandwidth of each link $(u,v)\in A$ as c_{uv}. The capacity of a single cable can be computed as c_{uv}/N_{uv}. Let $\alpha(0<\alpha\leq 1)$ 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 Bundled Links** (**REAR-BL**) is to minimize the number of used cables 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 α when backup sharing is not allowed. The no sharing case means that a link should reserve capacity larger than total demands of requests which contain it in the active and backup paths. Please see TABLE I for the notation used in the paper.

B. The Linear Integer Programming Model

We formulate a nonlinear programming model P for REAR-BL.

The goal of model P is to minimize the number of used cables. Equation (1) and (2) are flow conservation constraints which ensure an active path and a backup path for each request. Equation (3) demonstrates the link disjoint constraint for each request, i.e., link (u,v) can only be used by either the active path or the backup path. Equation (4) shows maximum link utilization constraint when backup sharing is not allowed. Equation (5) ensures the range of the number of available cables. Equation (6) implies all variables are binaries or integers. We can also add (7~8) to make sure all paths are acyclic. Note that the optimal value of model P is not affected if they are omitted.

Ref. [8] shows that the presented formulation without restoration constraint (3) is NP-hard by reducing the simple two-commodity flow problem in directed graph [16], which is NP-complete. Thus model P is also NP-hard. Exactly solving model P by CPLEX, a high-performance mathematical programming solver, becomes computationally intractable. So we propose heuristic algorithm to solve it. In order to evaluate our algorithm, we give an upper and lower bound of the optimal value of model P.

TABLE I. Notation Used in This Paper

Notation	Meaning
α	threshold of maximum link utilization
c_{uv}	Capacity/bandwidth of link (u,v)
n_{uv}	The number of used cables on link (u,v)
P_1^i	Active path of request i
P_2^i	Backup path of request i
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
$N^+(v)$	Set of outbound links $N^+(v)=\{(v,u)\mid(v,u)\in A\}$
$N^-(v)$	Set of inbound links $N^-(v)=\{(u,v)\mid(u,v)\in A\}$
Paths	Set of all active and backup paths. $Paths=\bigcup_{i=1}^{D}(P_1^i\cup P_2^i)$

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

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

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

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

$$\sum_{i=1}^{D} d_i\left(x_{uv}^i + y_{uv}^i\right) \leq \alpha \frac{n_{uv}}{N_{uv}} c_{uv} \qquad \forall (u,v)\in A \qquad (4)$$

$$0\leq n_{uv}\leq N_{uv} \qquad (5)$$

$$x_{uv}^i, y_{uv}^i \in\{0,1\}, n_{uv}\in\mathbb{N}_0 \qquad (6)$$

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

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

C. Upper and Lower Bound

Let the new model with linear relaxation of variable n_{uv} be model P'. We can get model P' via deleting $n_{uv}\in\mathbb{N}_0$ in (6) of model P. Denote the optimal solution of model P, P' as $n_{uv}(P)$, $\tilde{n}_{uv}(P')$, respectively. Then, we have the following theorem.

Theorem1. For REAR_BL, the optimal value of model P satisfies $\sum_{uv\in A}\tilde{n}_{uv}(P')\leq\sum_{uv\in A}n_{uv}(P)\leq\sum_{uv\in A}\lceil\tilde{n}_{uv}(P')\rceil$, where $\lceil\ \rceil$ denotes 'round up'.

Proof: 1) Model P' is a relaxation of model P, thus the feasible region are larger than that of model P. So $\sum_{uv\in A}\tilde{n}_{uv}(P')\leq\sum_{uv\in A}n_{uv}(P)$.

2) $\lceil\tilde{n}_{uv}(P')\rceil$ satisfies all the constraints of model P, thus is a feasible solution and can be an upper bound. So $\sum_{uv\in A}n_{uv}(P)\leq\sum_{uv\in A}\lceil\tilde{n}_{uv}(P')\rceil$. □

Theorem1 provides the upper and lower bound for model P via solving model P' with less integer variables. The upper bound is also an initial solution of REAR-BL.

III. HEURISTIC ALGORITHM

Model P' is a mixed integer programming which also maybe time consuming for large graph. So how can we design efficient algorithm utilizing the property of the model? Note that $\sum_{uv\in A} n_{uv} = \sum_{uv\in A}\left[\sum_{i=1}^{D} f_{uv}^i / (\alpha \frac{c_{uv}}{N_{uv}})\right]$ where $f_{uv}^i = d_i\left(x_{uv}^i + y_{uv}^i\right)$. Thus

$$\sum_{uv\in A} n_{uv} \in [\sum_{uv\in A}\sum_{i=1}^{D} f_{uv}^i / (\alpha\frac{c_{uv}}{N_{uv}}), \sum_{uv\in A}\left(\sum_{i=1}^{D} f_{uv}^i / (\alpha\frac{c_{uv}}{N_{uv}}) + 1\right)] \quad (7),$$

$$\begin{aligned}\sum_{uv\in A}\sum_{i=1}^{D} f_{uv}^i / (\alpha\frac{c_{uv}}{N_{uv}}) &= \sum_{i=1}^{D}\sum_{uv\in A} f_{uv}^i / (\alpha\frac{c_{uv}}{N_{uv}}) \\ &= \sum_{i=1}^{D}\frac{d_i}{\alpha}\sum_{uv\in A}\left(x_{uv}^i + y_{uv}^i\right)/(\frac{c_{uv}}{N_{uv}}) \approx \sum_{i=1}^{D}\frac{d_i}{\alpha}(w(P_1^i)+w(P_2^i))\end{aligned} \quad (8),$$

where $w_{uv} = \frac{N_{uv}}{c_{uv}}$ and $w(P) = \sum_{(u,v)\in P} w_{uv}$. Thus minimizing the range of $\sum_{uv\in A} n_{uv}$, we can minimize the total weight of the active and backup path of each request. And Suurballe algorithm [17] can be used in our heuristic as it is an algorithm to find a pair of disjoint paths for a source-termination request with minimum total weight.

Based on the analysis above, we propose **Greedy-Suurballe (GS)** algorithm as shown in Fig 1.

Greedy-Suurballe $(G(V,A), LSPs, c, N, \alpha)$

1). Set $w_{uv} = \frac{N_{uv}}{c_{uv}} C$, $c'_{uv} = \alpha c_{uv}$, $n_{uv} = f_{uv} = 0$, $Paths = \phi$;

2). **For** $i = 1, \cdots, D$

Construct $G_f(N_f, A_f, w(\cdot))$ by deleting links in G if $d_i < c'_{uv}$;

Call Suurballe algorithm to generate P_1^i, P_2^i in G_f;

$f_{uv} = f_{uv} + d_i$, $c'_{uv} = c'_{uv} - d_i$, $n_{uv} = f_{uv} / (\alpha * \frac{c_{uv}}{N_{uv}})$, $\forall (u,v) \in P_1^i, P_2^i$

$Paths = Paths \cup \{P_1^i, P_2^i\}$;

End-For

3). Let $A1 = \{(u,v) \mid n_{uv} > 0\}$, $flag_{uv} = 1$;

While $A2 = \{(u,v) \mid (u,v) \in A1, flag_{uv} = 1\} \neq \phi$

$N'_{uv} = n_{uv}$;

Select link $(p,q) = \arg\max_{(u,v)\in A2}(c'_{uv} - f_{uv})$. Set $N'_{pq} = n_{pq} - 1$;

Find affected requests LSP_a whose P_1^i or P_2^i contain link (u,v)

If rearranging LSP_a in $G(V, A1)$ with bundle size N' is feasible

$n_{pq} = n_{pq} - 1$. Update f_{uv}, c'_{uv} and $Paths$

If $n_{pq} = 0$

$flag_{pq} = 0$, $A1 = A1/(p,q)$;

End-If

Else

$flag_{pq} = 0$

End-If

End-While

Output total number of used cables $\sum_{(u,v)\in A1} n_{uv}$.

Fig. 1. Greedy-Suurballe Algorithm

GS algorithm includes three steps. Step 1 sets initial weights where $C = \max_{(u,v)\in A} c_{uv}$. Step 2 finds a feasible solution for REAR-BL via Suurballe algorithm for each request. Step 1 and 2 find the initial solution. We call these two steps as GS1. Step 3 is an iteratively deleting method to improve the solution of Step 2. A link (p,q) with maximum spare capacity is chosen. And we try to put one cable on it to sleep. If it is not feasible, we mark $flag_{pq} = 0$. Then process other links in the same way until all links have $flag = 0$.

Time Complexity: Suurballe algorithm uses $O(m\log_{(1+m/n)} n)$ time [15]. Thus, Step 2 requires $O(Dm\log_{(1+m/n)} n)$. In the worst case, Step 3 needs m times of that in Step 2. Thus part 3 requires $O(Dm^2\log_{(1+m/n)} n)$ time. Therefore, the time complexity of GS algorithm is $O(Dm^2\log_{(1+m/n)} n)$ while GS1 needs $O(Dm\log_{(1+m/n)} n)$.

IV. EVALUATION

In this section, we show the effectiveness of GS on energy saving when compared with CPLEX [18] and the upper and lower bound.

A. Experimental Setup

We have run extensive simulations on 4 core networks with traffic matrices obtained from the Simple Network Description Library [19]. Their statistics are shown in TABLE II. Last column shows the range of demand value.

TABLE II. NETWORK TOPOLOGIES

Networks	#Nodes	#Links	#Requests	c_{uv}	d
Newyork	16	98	240	1000	2~42
Norway	27	102	702	1000	1~14
India	35	160	595	600	1~10
Germany	44	166	462	10000	2~76

We use $PS = \sum_{(u,v)} n_{uv} / \sum_{(u,v)} N_{uv} * 100$ to calculate the power saving rate. As far as we know, this is the first work on REAR-BL. So we use CPLEX to solve model P as a fair comparison. As CPLEX cannot ensure a feasible solution in limited time (even 24h), we set a limit of 300s (excludes preprocessing time). We also compare GS with the min and max savings obtained by the upper and lower bound in theorem 1 to find 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.

B. Energy Saving with Different Bundle Size

We evaluate GS when bundle size N_{uv} changes from 1 to10. As shown in Fig 2, for the four networks, the PS of GS grows sharply when N_{uv} increases from 1 to 3 and grows gently close to max savings. There is only less than 10% difference on PS between GS and max saving. GS can save more than 50% of energy with N_{uv} above 5. The average running time of different solutions when $N_{uv} = 5$ are shown in TABLE III. GS1 has nearly the same PS values as the min savings with running time 10% less than that of the latter. GS performs better than CPLEX, especially on Germany with much less time. One can use GS1 to get initial solution instead of solving model P' for time consideration.

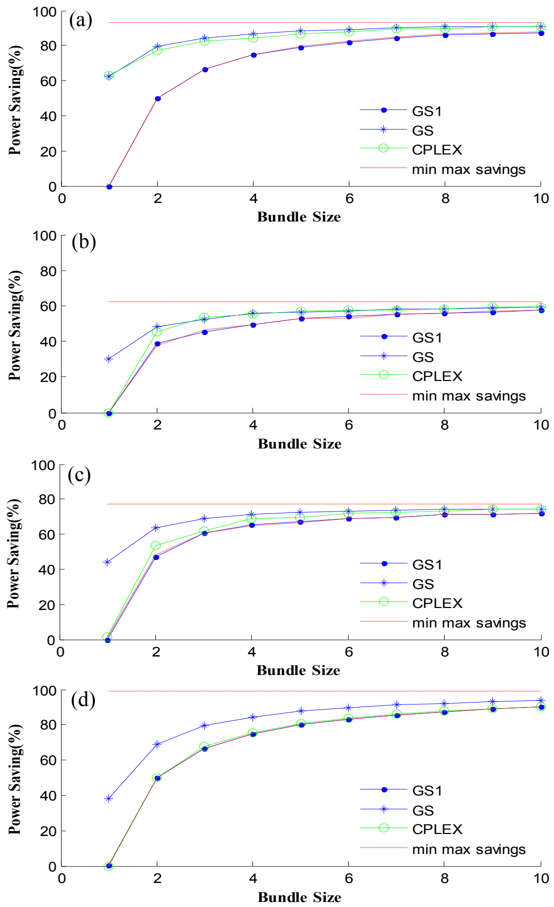

Fig. 2. Energy saving on different networks when $\alpha = 1$. (a) Newyork, (b) Norway, (c) India, (d) Germany

TABLE III. AVERAGE RUNNING TIME ON ALL NETWORKS WHEN $\alpha = 1$

	Networks			
Time(s)	Newyork	Norway	India	Newyork
GS1	0.5	4.6	4.5	9.7
GS	6.6	41.6	45.1	63.2
Min max savings	7.7	151.0	63.2	222.2

C. *Energy Saving with Different* α

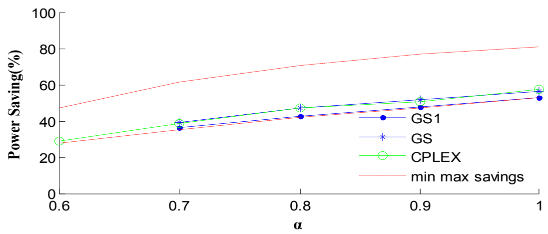

Fig. 3. Energy saving on Norway when $N_{uv} = 5$

We evaluate GS when α changes from 0.1 to 1 on Norway when $N_{uv} = 5$. Results on other networks are similar which thus are omitted. As shown in Fig. 3, the PS increases as α gets larger. GS cannot save energy with α under 0.5 while CPLEX with 300s cannot save energy with α under 0.6. GS1 always achieves similar PS as min savings. And GS performs better than CPLEX. All these simulation results confirm the efficiency of GS1 and the effectiveness of GS.

V. CONCLUSION

In this paper, we consider the problem of minimizing the total number of used cables while taking restoration into account under maximum link utilization constraint. We formulate a linear integer model as well as an upper and lower bound for this NP-hard problem. Then, we propose a heuristic GS which performs very close to optimal and significantly better than CPLEX as the energy savings of GS is 10% less than the max savings. It can save up to 50% with bundled size above 5. We plan to give a tighter lower bound and design more efficient heuristic for REAR-BL as a future work.

REFERENCES

[1]M. Pickavet, W. Vereecken, S. Demeyer, P. Audenaert, B. Vermeulen, C. Develder, D. Colle, B. Dhoedt, and P. Demeester, "Worldwide Energy Needs for ICT: The Rise of Power-Aware Networking," in *IEEE ANTS*, Mumbai, India, 2008, pp. 1-3.
[2]J. Guichard, F. Le Faucheur and J. Vasseur, *Definitive MPLS Network Designs*: Cisco Press, 2005.
[3]M. Gupta and S. Singh, "Greening of the Internet," in *ACM SIGCOMM*, 2003, pp. 19-26.
[4]J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power Awareness in Network Design and Routing,": IEEE INFOCOM, 2008, pp. 457 - 465.
[5]L. Chiaraviglio, M. Mellia and F. Neri, "Reducing Power Consumption in Backbone Networks," in *IEEE ICC*, 2009, pp. 1 - 6.
[6]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, 2012.
[7]M. Zhang, C. Yi, B. Liu, and B. Zhang, "GreenTE: Power-aware Traffic Engineering," in *IEEE ICNP*, Kyoto, Japan, 2010, pp. 21-30.
[8]W. Fisher, M. Suchara and J. Rexford, "Greening Backbone Networks: Reducing Energy Consumption by Shutting Off Cables in Bundled Links," in *ACM SIGCOMM workshop on Green networking*, New Delhi, India, 2010, pp. 29-34.
[9]G. Lin, S. Soh, K. Chin, and M. Lazarescu, "Efficient heuristics for energy-aware routing in networks with bundled links," *Computer Networks*, vol. 57, no. 8, pp. 1774-1788, 2013.
[10]M. Kodialam and T. V. Lakshman, "Dynamic Routing of Bandwidth Guaranteed Tunnels with Restoration," in *IEEE INFOCOM* Tel-Aviv, Israel, 2000, pp. 902 - 911.
[11]Y. Bejerano, Y. Breitbart, A. Orda, R. Rastogi, and A. Sprintson, "Algorithms for Computing QoS Paths With Restoration," *IEEE/ACM Transactions on Networking*, vol. 13, no. 3, pp. 648-661, 2005.
[12]M. Kodialam and T. V. Lakshman, "Restorable Dynamic Quality of Service Routing," *IEEE Communications Magazine*, vol. 40, no. 6, pp. 72 - 81, 2002.
[13]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, 2003.
[14]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, 2014.
[15]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.
[16]"On thecomplexity of time table and multi-commodityow problems,",.
[17]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.
[18]CPLEX, http://www.ilog.com/products/cplex/
[19]Simple Network Description Library (SNDlib), http://sndlib.zib.de.