Robust Energy-Aware Routing with Uncertain Traffic Demands

Heng Lin
Tsinghua University
henglin1991@gmail.com

Mingwei Xu
Tsinghua University
xmw@cernet.edu.cn

Yuan Yang
Tsinghua University
yyang@csnet1.cs.tsinghua.edu.cn

***Abstract*—Energy conservation has become a major challenge to the Internet. In existing approaches, a part of line cards are switched into sleep mode for energy conservation, and the routing is configured carefully to balance energy saving and traffic engineering goals, such as the maximum link utilization ratio (MLUR). Typically, traffic demands are used as inputs, and routing is computed accordingly. However, accurate traffic matrices are difficult to obtain and are changing frequently. This makes the approaches difficult to implement. Further, the routing may shift frequently, and is not robust to sudden traffic changes.**

In this paper, we propose a different approach that finds one energy-aware routing robust to a set of traffic matrices, particularly to arbitrary traffic demands. Such a routing without energy consideration is known as the demand-oblivious routing, and is well studied. However, the problem becomes much more challenging when energy conservation is involved. To overcome the challenges, we first define a new metric, namely oblivious performance ratio with energy constraint (OPRE), which reflects the MLUR distance from a routing to the optimal routing when certain energy conservation requirement is satisfied. We model the problem of minimizing the performance ratio, and analyze the lower and the upper bounds. Then, we propose Robust Energy-Aware Routing (REAR) to solve the problem in two phases. REAR select sleeping links based on extended robust link utilization (ERLU) and compute the routing based on a classical demand-oblivious routing algorithm. We evaluate our algorithms on real and synthetic topologies. The simulation results show that REAR can save 19% of line card power while the performance ratio is less than 34%.

I. Introduction

Energy conservation has become a global concern nowadays. The Internet is one of the major energy consumers, and its rapid growth makes the green Internet a hot research topic. In the Internet backbone, energy is mainly drawn by routers and switches. Such devices consume almost full power even if the traffic load is small. Thus, an effective method to save energy in the Internet is to aggregate traffic into part of the routers when the traffic load is small, and switch the under-utilized components (routers or line cards) into off/sleep mode. Such a method is known as the energy efficient routing.

An important issue for energy efficient routing is to avoid network congestion after the traffic is aggregated. Many approaches have been proposed in existing works. Most approaches compute the routing based on real-time traffic matrices or link loads, to achieve a good load balancing and avoid congestion. However, such approaches come at a cost of obtaining real-time traffic data. Furthermore, the routing may shift frequently with the traffic changes, and a sudden traffic change may still induce congestions. To this end, we need to study the robustness of energy efficient routing.

Specifically, we study the energy efficient routing when traffic matrices cannot be obtained or predicted precisely, i.e., with uncertain traffic demands. To achieve robustness, we need to find a routing, which can perform near optimally under a range of traffic matrices. The key technique that makes this possible is the advanced *demand-oblivious routing*. A seminal work [7] find that, the distance between the maximum link utilization ratio (MLUR) of a demand-oblivious routing and the MLUR of the optimal routing is bounded. Thus, no matter how the traffic demand changes, the demand-oblivious routing can guarantee certain performance. We note that this conclusion is in a network without sleeping components, and we need to consider the situation when energy conservation is required.

However, existing demand-oblivious routing algorithms cannot be directly applied to energy efficient routing. There are several challenges. First, we need to define a metric that can effectively measure the distance between a robust energy efficient routing and the optimal routing, because the existing metric for demand-oblivious routing fails in the situation when some components are switched into off/sleep mode. Second, we need to analyze whether the metric can be bounded, just like for demand-oblivious routing. If there exists no bound, then it is not feasible to find a robust energy efficient routing. Third, we need practical algorithms to compute the robust energy efficient routing. Specifically, we need to determine: 1) which routers or line cards should be switched into off/sleep mode, to achieve energy efficiency; and 2) in which path to forward the traffic for robustness, while the path does not traverse the sleeping components.

In this paper, we overcome the aforementioned challenges. First, we define a new metric, namely the oblivious performance ratio with energy constraint (OPRE). The OPRE reflects the MLUR distance from a routing to the optimal one when certain energy conservation requirement is satisfied. We model the problem of minimizing the OPRE. Second, we prove that there exists a robust energy efficient routing with the minimum OPRE, which has an upper bound given a network. Then, we propose Robust Energy-Aware Routing (REAR) scheme, which uses heuristic algorithms to solve the problem. We develop algorithm RLP, which chooses sleeping line cards in a way that the OPRE can be minimized potentially. We then develop algorithm RRA to compute the routing in the remaining topology, by extending existing optimal demand-oblivious routing algorithm. We evaluate our algorithms by simulations on real topologies and synthetic traffic demands with random fluctuations. The results show that REAR can achieve an OPRE of 1.34 while 19% of line card power is saved.

The rest of the paper is organized as follows. Section II shows the related work. Section III presents metric OPRE and formally models the problem. We presents the bounds on OPRE in Section IV, and propose our algorithms in Section V. Section VI shows our simulation setup and results, and Section VII concludes our work.

II. Related Work

Green routing is studied because we can not use as much of a resource as we need [1], such as line cards always work in full mode even with low traffic fraction. So there are many studies to save energy consumption by put some links in sleep mode, for example GreenTE [3] is proposed for maximuming the sleeping links while E^2-MCRA [2] is proposed for minimizing the active links and nodes by a depth-first search approach. Besides these centralized algorithms, EAR [4] and GRiDA [5] are proposed for distributed deployment, both of them are compatible with OSPF sharing additional information between nodes. For addressing the problem of routing convergence when traffic changes frequently, GreenFRR [6] is proposed by leveraging the technique of fast rerouting.

Network may suffer from bad performance because of both frequent traffic demand and routing changes. There are studies to estimate traffic matrices in the context of Internet traffic, such as LP approach [8], Bayesian Infereence techniques [9] and Time-Varying Network Tomography [10]. All the estimatation for traffic matrices result in the changes of routing, [11] proposed an oblivious routing algorithm with a polylog competitive ratio with respect to congestion and [12] give a polynomial time algorithm based on applying the Ellipsoid algorithm to an exponential-size LP model. To be practical for large networks, the oblivious-demand routing based on polynomial-size LP formulation is investigated in [7].

In this paper, we proposed robust energy-aware routing based on the oblivious-demand [7], which is both green for the energy consumption by pruning some links based on ERLU, and also robust for uncertain traffic demands. It is more practical than existing green routing algorihtm because it almost do not need changes when real traffic varifies.

III. Problem Statement

A. Background of Demand-Oblivious Routing

As mentioned above, demand-oblivious routing aims at finding one routing that performs near optimally under a range of traffic demands. In traffic engineering, a typical metric to evaluate routing performance is the maximum link utilization ratio (MLUR). Clearly, the MLUR is corresponded with a specified traffic matrix (TM). Demand-oblivious routing defines oblivious performance ratio (OPR) to evaluate the routing performance without the knowledge of TM. We briefly present the background.

Given a TM, the distance between a routing to the optimal one is defined as the ratio between their MLURs. Formally, a network is modeled as undirected graph $G(V, E)$, where V is the set of vertices (nodes), and E is the set of edges (links). Let cap_{ij} denote the capacity of the link $(i, j) \in E$. Let d_{ab} denote the traffic demand from origin node a and destination node b, and m denote the TM that contains d_{ab} for all $a, b \in V$. Let $f^r_{ab}(i, j)$ be the fraction of d_{ab} that is routed on link (i, j) $(0 \leq f^r_{ab}(i, j) \leq 1)$, using routing r. Routing r is specified by $f^r_{ab}(i, j)$ for all $a, b \in V$ and $(i, j) \in E$, and we will formally define routing consistency later in Section III.C. Let $U_{r,m,G}$ be the MLUR of routing r under TM m. We have

$$U_{r,m,G} = \max_{(i,j)\in E} \frac{\sum_{a,b} d_{ab} f^r_{ab}(i, j)}{cap_{ij}}. \tag{1}$$

Let $P(\{r\}, \{m\}, G)$ be the *performance ratio* of routing r under TM m, which reflects how far from the routing to the optimal one, and is defined as

$$P(r, \{m\}, G) = \frac{U_{r,m,G}}{\min_{r'} U_{r',m,G}}. \tag{2}$$

The *oblivious performance ratio* (OPR) for routing r is defined by extending TM m to a set of TMs M, where M can be the set of all TMs. We have

$$P(r, M, G) = \max_{m\in M} P(r, \{m\}, G). \tag{3}$$

The target of demand-oblivious routing is to find routing r that minimizes OPR $P(\{r\}, M, G)$. Such a “robust” routing is independent of a specific TM, but can perform near optimally. A seminal work [7] uses linear programming to find the routing that minimizes the OPR, and finds that the optimal solution exists, which means that the OPR is bounded.

B. Oblivious Performance Ratio with Energy Constraint

With energy constraint, a network may have to switch part of line cards into off/sleep mode to save the total energy consumption. This changes the network topology and makes metric OPR fail to evaluate the robustness of a routing. We use an example to show this. Assume that G is a cycle with n unit capacity links. Then, the minimal OPR of G is $2 - 2/n$ [7]. Now we pruning one link from G to save energy, and the topology changes to G^*. Because there is only one routing feasible in G^*, OPR $P(\{r\}, M, G^*)$ equals 1. Since 1 is less than $2 - 2/n$ for $n > 2$, it means that the routing after pruning one link is more robust than before. However, it is false because there are less links and the network is more likely to be congested.

The intrinsic reason for such a “fake robust” is that the topology is not changing when performance ratio is computed in Eq. (2). To address this issue, we extend the definition of OPR. Formally, let G^* be a sub-graph of G that satisfies the energy constraint (We will formally define the energy constraint in Section III.C, i.e., Eq. (9)). Note that in this case, a path cannot traverse a sleeping link, so routing r is limited

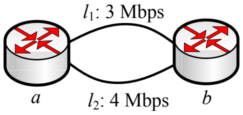


Fig. 1. An example of the OPRE, where the traffic demand is d_{ab}.

by G^* (Eq. (7) in Section III.C). The extended performance ratio is defined as

$$P(r,\{m\},G,G^*)=\frac{U_{r,m,G^*}}{\min_{r'} U_{r',m,G}}. \tag{4}$$

Let $P^*(r,M,G,G^*)$ be the *oblivious performance ratio with energy constraint* (OPRE) for routing r. We define $P^*(r,M,G,G^*)$ by extending m to M. Similar to Eq. (3), we have

$$P^*(r,M,G,G^*)=\max_{m\in M} P(r,\{m\},G,G^*). \tag{5}$$

Note that G^* is used to achieve certain energy conservation target. For an energy conservation target, there may exist a set of different G^*, which result in different OPRE. We will show an example below. When there is no energy constraint, G^* equals G, and Eq. (5) naturally reduces to Eq. (3).

C. Example

We show an example of our new metric OPRE. The network is shown in Fig. 1, where two nodes a,b are connected by two links l_1 and l_2.[1] The capacities of the links are 3 Mbps and 4 Mbps, respectively. To forward a traffic demand of d_{ab}, the optimal routing is to put $\frac{3}{7}d_{ab}$ on link l_1 and $\frac{4}{7}d_{ab}$ on link l_2, which results in the MLUR of $\frac{1}{7}d_{ab}$. This routing is also the optimal demand-oblivious routing, because the routing is optimal no matter how d_{ab} changes. Thus, the OPR equals 1.

Now, assume that links l_1 and l_2 consume the same power, and one link has to be switched off/sleep to save energy. Let us check the OPRE after pruning l_1 and l_2, respectively. If l_1 is pruned, d_{ab} is put on link l_2, and the MLUR is $\frac{1}{4}d_{ab}$, so the OPRE is $\frac{1}{4}d_{ab}/\frac{1}{7}d_{ab}=7/4=1.75$. If l_2 is pruned, d_{ab} traverses link l_1, resulting in the MLUR of $\frac{1}{3}d_{ab}$, so the OPRE is $\frac{1}{3}d_{ab}/\frac{1}{7}d_{ab}=7/3=2.33$. The results tell us that pruning link l_1 is more robust than pruning link l_2. This is consistent with the intuition that link l_2 has a larger capacity and is stronger against congestion.

D. Problem Formulation

Now we formally model the problem. Our objective is to minimize OPRE $P^*(r,M,G,G^*)$, where M and G are inputs, and routing r and sub-graph G^* are decision variables. The Min-OPRE problem is as follows.

$$\text{Minimize} \quad P^*(r,M,G,G^*)$$

$$\text{s.t. (1), (4), (5)}$$

$\forall a,b,i\in V$:

$$\sum_{j,s.t.(i,j)\in E} f^{r'}_{ab}(i,j)-\sum_{j,s.t.(j,i)\in E} f^{r'}_{ab}(j,i)=\left\{\begin{array}{rl}1, & i=a\\ -1, & i=b\\ 0, & i\neq a,b\end{array}\right. \tag{6}$$

$\forall a,b,i\in V$:

$$\sum_{j,s.t.(i,j)\in E^*} f^{r}_{ab}(i,j)-\sum_{j,s.t.(j,i)\in E^*} f^{r}_{ab}(j,i)=\left\{\begin{array}{rl}1, & i=a\\ -1, & i=b\\ 0, & i\neq a,b\end{array}\right. \tag{7}$$

$\forall a,b\in V$, $\forall i,j\in E$:

$$0\le f^{r}_{ab}(i,j), f^{r'}_{ab}(i,j)\le 1 \tag{8}$$

$$\sum_{l\in E^*} p(l)/\sum_{l\in E} p(l)\le 1-\theta \tag{9}$$

Constraints (1), (4), and (5) follow our definition of the OPRE. Eq. (6) means that the optimal routing r' in G must be a valid routing, i.e. with routing consistency. Specifically, for the source/destination node, all traffic must be routed/received; and for an intermediate node, the traffic flowing out must equal the traffic flowing in. Similarly, Eq. (7) means that the robust routing r in G^* must be a valid routing. Ineq. (8) means that the range of $f^r_{ab}(i,j)$ and $f^{r'}_{ab}(i,j)$ is from 0 to 1. Ineq. (9) means that the power consumption of links in G^* must be less than a given threshold. In Ineq. (9), θ denotes the power saving ratio we must achieve by using G^*. Here, we use $p(l)$ to denote the power consumption of link l, which is mainly drawn by the line cards at the two ends of the link. Note that $p(l)$ is independent of the traffic volume, and this assumption is based on the fact that in current stage, the power of a line card changes little with the traffic volume [15].

Note that if $G^*=G$ satisfy the energy constraint Ineq. (9) and M is the set of all TMs, the Min-OPRE problem reduces to the demand-oblivious routing problem, and can be transformed into a linear program and solved in polynomial time [7]. However, we must find the optimal G^* to achieve the required power saving ratio in other cases, and Min-OPRE is an MILP and is NP-hard in general.

Also note that we put the power saving in the constraint of our problem, instead of the objective function. In such a way, we can balance the trade off between energy conservation and the OPRE, by setting different values of power saving ratio θ. We can even achieve the maximum power saving ratio by a binary search on θ.

IV. Upper Bounds of the Minimum OPRE

In this section, we analyze the upper bounds of the minimum OPRE. We are interested in two extreme types of networks, i.e., cycles and cliques, because general networks can be seen as intermediate cases between them. We will present the upper bounds with small and large values of θ. We find that, though the minimum OPR of cycles and cliques is similar and bounded by 2 [7], the minimum OPRE is much more different and has a larger upper bound with a larger θ.

[1] When there are parallel links, we can add a virtual node in each parallel link to keep $G(V,E)$ as a simple graph.

These theoretical results tell us that the robust energy efficient routing exists, and tell us how close a robust routing can get to the optimal routing.

Lemma 1. *$\min P^*(r, M, G, G^*) \geq \min P^*(r, M, G)$ if $G^* \subseteq G$, where the equation holds if $G^* = G$.*

Proof. We assume exist a G^*, s.t. min $P^*(r, M, G, G^*) <$ min $P^*(r, M, G)$. Becuase G^* is the subset of G, we directly use the robust routing of G^* to G, then $P^*(r, M, G)$ will equal to $P^*(r, M, G, G^*)$, and arise contradiction. Particularly, when $G^* = G$, G^* is the subset of G, and G is the subset of G^*, so there is both min $P^*(r, M, G, G^*) \geq$ min $P^*(r, M, G)$ and $P^*(r, M, G) \geq P^*(r, M, G, G^*)$, so they are equal to each other when $G^* = G$. This ends our proof. □

Theorem 2. *The minimum OPRE of C_n (the cycle on n vertices with unit capacity links) and K_n (the complete graph on n vertices with unit capacity links) is $2 - 2/n$ if $\theta = 0$.*

Proof. According to Lemma 1, the minimum OPRE of C_n is the minimum OPR, i.e., $\min P^*(r, M, C_n)$, when $\theta = 0$. Similarly, the minimum OPRE of K_n is $\min P^*(r, M, K_n)$ when $\theta = 0$. On the other hand, we know from [7] that the minimum OPR of C_n is $2 - 2/n$, and the minimum OPR of K_n is also $2 - 2/n$. This ends our proof. □

Theorem 2 shows the minimum OPRE when θ equals 0 and all link capacities are the same. In a more general case when there are different link capacities, we have

Theorem 3. *The upper bound of the minimum OPRE for a cycle/complete graph on n nodes is $1 + \frac{\max cap_{ij}}{\min cap_{ij}}$ if $\theta = 0$.*

Proof. See Appendix A. □

Theorem 3 tells us that the minimum OPRE is related to the link capacity difference in the network. The above results are for $\theta = 0$. Now let us see the results for a large θ. Since there is not much difference between the power consumptions for links with different capacities [15], we consider the spanning trees of a graph, which can achieve a near-optimal θ. We have

Theorem 4. *Let G^* be a spanning tree of a cycle on n nodes, and then the upper bound of the OPRE is $1 + \frac{\max cap_{ij}}{\min cap_{ij}}$.*

Proof. See Appendix B. □

Theorem 5. *Let G^* be a spanning tree of a complete graph on n nodes, and then the upper bound of the OPRE is $\frac{n^2}{2}\frac{\max cap_{ij}}{\min cap_{ij}}$.*

Proof. See Appendix C. □

Theorem 4 and Theorem 5 present a large gap between the OPRE upper bounds in the two types of networks, because when G^* is a spanning tree, more links are pruned in the complete graph. This implies that in order to achieve more energy conservation, the OPRE may become larger quickly, and the routing becomes less robust, in the worst case. However, we can still develop effective algorithms to achieve a small OPRE and save energy in general cases, i.e., in real-world networks which have much less link densities than a complete graph.

V. Robust Energy-Aware Routing Algorithms

In this section, we present our REAR approach to solve the Min-OPRE problem. We first develop the robust link pruning (RLP) algorithm. RLP selects sleeping links that we will prune from the topology, to achieve the required power saving ratio and potentially reduce the OPRE. Then, we develop the robust routing adjustment (RRA) algorithm. RRA adjusts the demand-oblivious routing that has the minimum OPR, to detour the traffic around the sleeping links.

A. Algorithm RLP

Several heuristics have been proposed to select sleeping links without the knowledge of a specific TM, with various considerations [] [] []. However, we need a algorithm that can potentially minimize the OPRE. To this end, we first find the demand-oblivious routing that minimizes the OPR, using the linear programming approach in [7]. Bases on the routing, we define the extended robust link utilization (ERLU), to evaluate the contribution of a link to the OPRE.

Intuitively, a link is less preferred to be pruned if there are more flows traversing the link. Though we are not aware of the TM, the traffic volume is likely to fluctuate largely if there are many flows. This means that if we prune this link, these flows have to be carried by other links in the network, and the OPRE may increase largely. On the other hand, a link is less preferred to be pruned if the link has a less capacity. This is because in the optimal demand-oblivious routing where the OPR is minimized, such a link is not likely to be heavy-loaded, and we should keep the link to maintain the robustness. Formally, let r_0 be the optimal demand-oblivious routing, and u^e_{ij} be the ERLU of link (i, j), we have

$$u^e_{ij}(r_0) = \frac{\sum_{a,b} f^{r_0}_{ab}(i, j)}{cap_{ij}} \tag{10}$$

With the ERLU, we iteratively select the link with the smallest ERLU as the sleeping link. If the topology after pruning the selected link is not connected, we pass the link and continue to search. We stop the process once we have sufficient sleeping links to achieve the required power saving ratio. We develop the RLP algorithm.

The inputs of RLP include graph $G(V, E)$ and required power saving ratio θ, and the output is the sleeping link set S. The ERLU of each link is computed in Steps 1 to 3. S is initialized to be empty (Step 4). In each iteration of the outer loop (Steps 5 to 12), a feasible sleeping link is selected (Steps 7 to 12). If a feasible sleeping link is found and the power saving ratio is not achieved yet (Step 9), the selected link is added into set S (Step 10), binary variable *goon* will be true (Step 11), and the process continues (Step 12). Or else, binary variable *goon* will remain false (Step 6), and the algorithm ends. The computation complexity of the RLP algorithm is $O(LP + |E||V|^2 + (|E| - |V|)|V||E| \log |E|)$ in the worst

```
Algorithm RLP()
  Input: $G(V,E)$, $\theta$;
  Output: the set of sleeping links $S$;
  1: $r_0 \leftarrow$ the optimal demand-oblivious routing [7] on $G$;
  2: for each link $l$ in E
  3:     compute ERLU $u^e_l(r_0)$, using Eq. (10);
  4: $S \leftarrow \emptyset$; $goon \leftarrow$ true;
  5: while $goon$
  6:     $goon \leftarrow$ false;
  7:     for each link $l$ in $E-S$ with increasing order based on $u^e_l(r_0)$
  8:         $G^* \leftarrow (V, E^* = E - S - \{l\})$;
  9:         if $G^*$ is connected and $\sum_{l' \in E^*} p(l')/\sum_{l' \in E} p(l') > 1 - \theta$
  10:            $S \leftarrow S \cup \{l\}$;
  11:            $goon \leftarrow$ true;
  12:            break;
  13: return $S$;
```

case, where $O(LP)$ is for solving the linear programming in [7], $O(|E||V|^2)$ is for computing the ERLU, $O(|E|-|V|)$ is the maximum number of sleeping links, and $O(|V||E|\log|E|)$ is for the connectivity check and the sleeping link search.

B. Algorithm RRA

After pruning the links, we get graph G^* (if exists) that satisfies the energy conservation requirement. Now we should compute the robust routing that minimizes the OPRE. Note that we cannot simply apply the algorithm in [7] on G^*, because the algorithm minimizes OPR but not OPRE. We observe that the optimal demand-oblivious routing r_0 minimizes the OPR in G, and minimizes OPRE when $G^* = G$ according to Theorem 2. Thus, to obtain a robust routing in G^*, we take a heuristic that adjusts routing r_0 to detour the traffic around the sleeping links in G^*.

Note that there may be multiple paths in r_0 between a source-destination pair. We first decompose the paths by a standard procedure given in []. Formally, let p^h_{ab} denote the h-th path between nodes a and b, and t^h_{ab} be the fraction of the traffic that is routed on p^h_{ab}. For each p^h_{ab} that traverses a sleeping link, we must find some detour paths to absorb t^h_{ab}. Such detour paths should not increase the OPRE too much. Recall that ERLU $u^e_{ij}(r)$ reflects the link utilization ratio without the knowledge of the TM, under routing r. Thus, we prefer a path whose maximum $u^e_{ij}(r)$ is small. On the other hand, the detour paths should not be too long, in order to limit the path stretch. To this end, we use YEN's algorithm [16] to compute the k-th shortest path in graph G^*. We develop Algorithm RRA.

```
Algorithm RRA()
  Input: $G(V,E)$, sleeping link set $S$,
         optimal demand-oblivious routing $r_0$, $N$;
  Output: robust energy-aware routing $r$;
  1: $G^* \leftarrow G(V, E-S)$; $r \leftarrow r_0$;
  2: for each link $l$ in $S$
  3:     for each $s, d \in V$
  4:         $t \leftarrow f^r_{sd}(l)$;
  5:         $paths \leftarrow$ the extracted paths for $s, d$;
  6:         for each path $p^h_{sd} \in paths$
  7:             if $l \in path$
  8:                 $paths$.remove($p^h_{sd}$);
  9:         $yen\_paths \leftarrow$ yens_algorithm($G^*$, $s$, $d$);
  10:        for $counter = 1$ to $N$
  11:            compute $u^e_{ij}(r)$;
  12:            $p \leftarrow$ the path in $yen\_paths$ with the min. $\max_{ij} u^e_{ij}(r)$;
  13:            $p.fraction = t/N$;
  14:            $paths$.add($p$);
  15:        $r \leftarrow paths$.merge();
  16: return $r$;
```

The inputs of RRA include graph $G(V,E)$, sleeping link set S, optimal demand-oblivious routing r_0 in G, and N which will be used latter. The output is the robust energy efficient routing r. G^* and r are initialized in Step 1. The main loop from Step 2 to Step 14 finds the detour paths for each sleeping link in S, and for each source-destination pair. The traffic fraction that should be detoured is denoted by t (Step 4), and the original paths are extracted and removed from routing r (Steps 5 to 8). Then, the k-th shortest paths are computed in Step 9. The traffic fraction of t is detoured by N times evenly (Steps 10 to 14), and the detour path with the minimum value of the ERLU is selected (Steps 11 and 12). In Step 15, the duplicate paths are merges and new routing r that does not traverse link l is obtained. The computation complexity of the RRA algorithm is $O((|E|-|V|)|V|^2(|R||E| + K|V|(|E| + |V|\log|V|) + N(|E||V|^2 + K|E|) + |R||E|))$ in the worst case, where $O(|E|-|V|)$ is the maximum number of sleeping links, $O(|V|^2)$ is the maximum number of OD pairs, $O(|R||E|)$ is for computing detoured traffic and remove failed paths where $O(|R|)$ is the number of input routings, $O(K|V|(|E| + |V|\log|V|))$ is for solving the yens algorithm where K is the number of shortest paths, $O(N(|E||V|^2 + K|E|))$ is for detouring traffic fraction by N times, and $O(|R||E|)$ is for merging routings.

VI. Performance Evaluation

We evaluate the REAR approach by simulations on three real-world network topologies, including Abilene [17], Geant [18], and CERNET2 [19]. The numbers of nodes and links of the topologies are listed in Table VI. The traffic matrices we used in our simulations are generated as follows. We first generate a baseline traffic matrix using the gravity model [13], where the baseline traffic demand d_{ab} is proportional to their total capacity. Then, the traffic demand is generated randomly in the range from d_{ab}/ω to $d_{ab}\omega$, where $\omega \geq 1$ is a margin factor used to control the fluctuation range of the traffic. We generate 1000 random TMs for each topology with each ω. The power consumption of different line cards is set as constant in the simulations, as shown in Table VI, following [3].

TABLE I
Topologies

Topology	Nodes	Links	Links can be Removed
Abilene	12	15	4
Geant	23	37	15
Cernet2	20	22	3

For comparison, we simulate a recent energy efficient routing approach, namely DMP [14], which also does not use the TMs as inputs. DMP always selects the most power-hungry links to switch into off/sleep mode, and can achieve a large

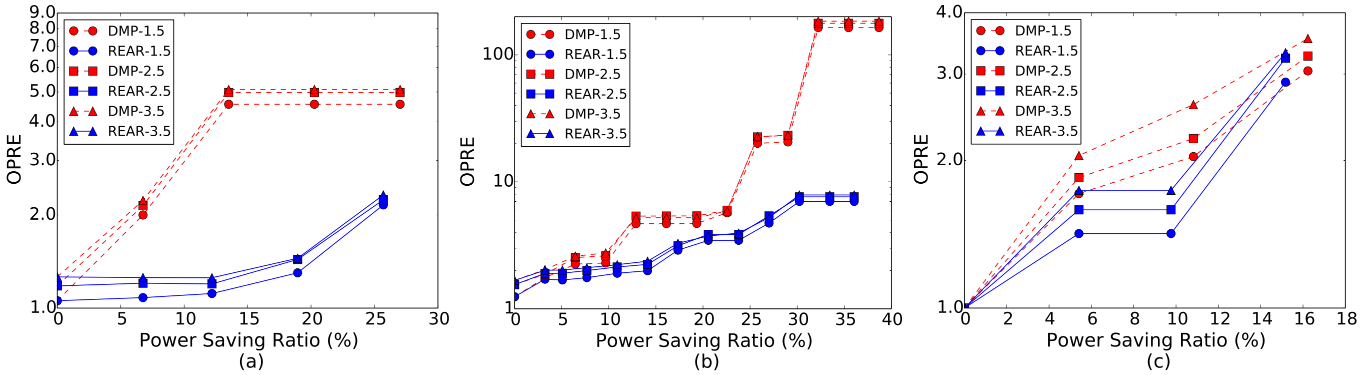


Fig. 2. OPRE versus TMs: (a). Abilene, (b). Geant, (c). Cernet2

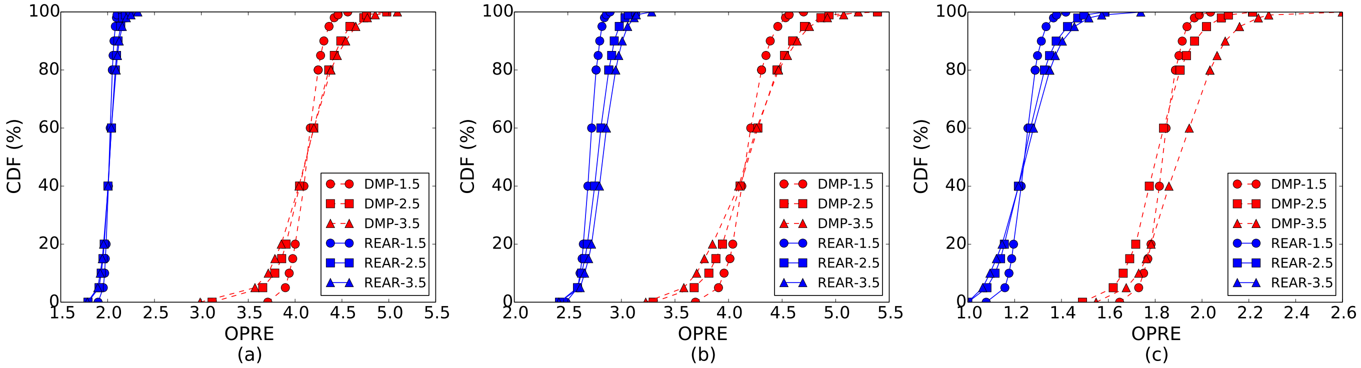


Fig. 3. OPRE versus Power Saving: (a). Abilene with 4 links removed (b). Geant with 6 links removed (c). Cernet2 with 2 links removed

power saving ratio, but does not consider routing robustness. We evaluate the OPRE and its distribution, the power saving ratio, and the path stretch of the approaches.

TABLE II
LINE CARD POWER CONSUMPTION

Line-Card	Speed(Mbps)	Power(Watts)
1-Port OC3	155.52	60
8-Port OC3	1244.16	100
1-Port OC48	2488.32	140
1-Port OC192	9953.28	174

A. OPRE versus Power Saving

Fig. 2 (a) shows the OPRE as a function of power saving ratio θ, in the Abilene topology. We see that the OPRE increases when more power is saved. This is not surprising because the routing becomes less robust when more links are switched into off/sleep mode. Our REAR has an OPRE ranging from 1.06 to 2.32, which is much less than the results of DMP. This is because REAR takes routing robustness into consideration when it prunes links and computes routing. For instance, when $\omega = 3.5$ and 25% power is saved, the OPRE of REAR is 2.32, while the OPRE of DMP is 5.09. This implies that REAR is more robust against traffic uncertainty. We also see that a larger ω results in a larger OPRE, because the traffic volume fluctuates in a wider range. For instance, when ω increases from 1.5 to 3.5, the OPRE of REAR increases about 0.17, while the OPRE of DMP increases about 0.53.

Similar results are showed in Fig. 2 (b) and (c), namely in Geant and Cernet2 topology. In Geant, we see that OPRE sacle with a wider range both in DMP and REAR, it is because Geant is much larger than other topologies, when removing too much links, the network will suffer from more serious congestion. In Cernet2, Although REAR is better than DMP, the difference is marginal when achieved a large θ. It can be explained by the density of links, REAR obtains a better performance in dense topology because it will have more choices.

B. OPRE versus Margin

Fig. 3 (a) shows the amount of TMs as a cumulative distribution function (CDF) of OPRE, when remove 4 links in Abilene topology. We see that the curves of DMP are right of REAR, means REAR always have lower OPRE whatever

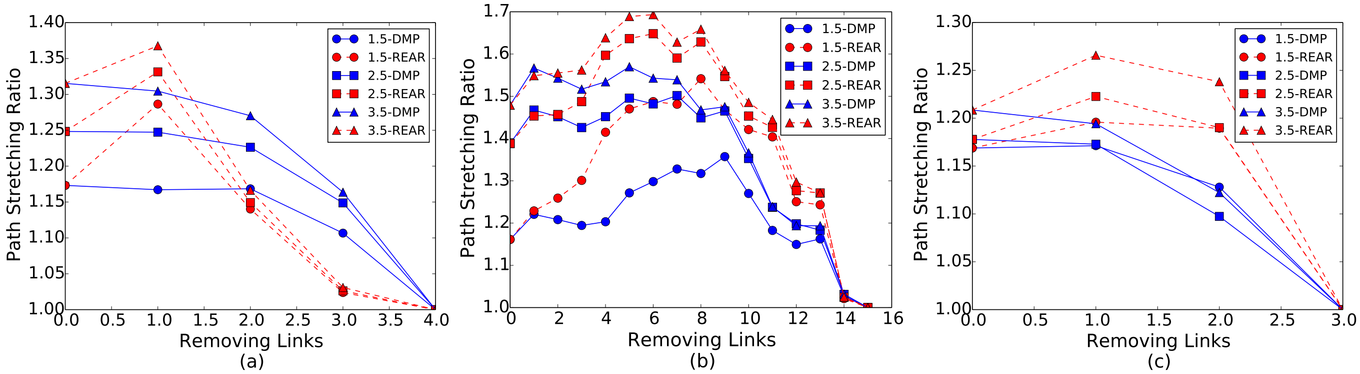


Fig. 4. Path Stretching: (a). Abilene, (b). Geant, (c). Cernet2

TMs is. It is consistent with Fig. 2 (a). We also observe that our REAR has more concentrated curves, means REAR have the almost same performance when TM changes, it is because REAR remove the less 'important' link from topology and the left topology is more robust for TMs than DMP. For instance, REAR ranges from 1.74 to 2.32 while DMP range from 2.97 to 5.09 when $\omega = 3.5$. Also we can see that curve become smooth when ω increases, it is because the TMs range from wider scope, and the OPRE may achieved wider scope too. However we see that whatever ω is, the center of the curve are almost the same postion. It is because we only change the margin ω rather than the base TM, so the medial value will not changes apparently.

Fig. 3 (b) shows the amount of TMs as a CDF of OPRE when remove 6 links in Geant, and Fig. 3 (c) shows the situation when remove 2 links in Cernet2. Similar results are showed in them, however both the centers of REAR curve in Fig. 3 (b) and DMP in Fig. 3 (c) are not at the same postion when ω changes. The reasons are two-folds, firstly the curves is drawn by sampling from origin data so the center of curve may not the real center. On the other hand, we calculate the distance of centers over the scope of OPRE, respectively 0.9% for REAR in Geant and 0.1% for DMP in Cernet2, so it is obvious that from sight of the scope of OPRE, the center of curves are the same 'postion'.

C. Path Stretching

Fig. 4 (a) shows the path stretching ratio (PSR) as a function of the number of removed links in Abilene topology. We see that PSR will equal to 1 when maximum number of links are removed. It is because when the topology reduces to a spanning tree, the routing between two vertices become unique, namely each path is the shortest path. Our REAR has an worst PSR value of 1.30, better than 1.36 in DMP, when remove 1 link and $\omega = 3.5$. We observe that when remove more than 1 link, the DMP is even better than REAR sometimes. This is because both REAR and DMP implement our RRP algorihtm for adjusting routing, and in RRP we always choose the routing path from k-shortest paths for the traffic on the removed link, which make the PSR little as possible. We also see that a larger ω results in a larger PSR, and reduces to 1 as explained above, because the demand-oblivious routing is computed based on ω, when ω increases the routing will find much more paths for avoiding sudden traffic demand, which makes the PSR increases. For example, when ω increases from 1.5 to 3.5, the PSR of REAR increases about 0.08, while the PSR of DMP increasess about 0.14.

Fig. 4 (b) and Fig. 4 (c) show the similar results in Geant and Cernet2, while our REAR is always better than DMP. For instance, REAR have the worst PSR of 1.35 and DMP is 1.54 when $\omega = 1.5$ in Geant, REAR have the worst PSR of 1.20 and DMP is 1.27 when $\omega = 3.5$

VII. Conclusion

The conclusion goes here.

References

[1] J. Carter and K. Rajamani, "Designing Energy-Efficient Servers and Data Centers," Computer, 2010.

[2] S. Avallone, G. Ventre. "Energy efficient online routing of flows with additive constraints." Computer Networks, 2012.

[3] M. Zhang, C. Yi, B. Liu, et al. "GreenTE: Power-Aware Traffic Engineering," in Proc. of IEEE ICNP, October 2010.

[4] A. Cianfrani, V. Eramo, M. Listanti, and M. Polverini, "An OSPF enhancement for energy saving in IP networks," in IEEE INFOCOM Workshop on Green Communications and Networking, April 2011.

[5] A. P. Bianzino, L. Chiaraviglio, M. Mellia, et al. "GRiDA: GReen Distributed Algorithm for energy-efficient IP backbone networks," Computer Networks, 2012.

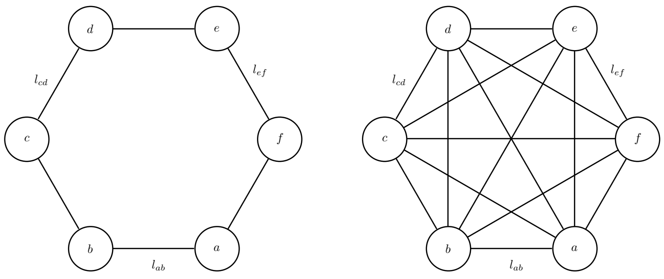
[6] Y. Yang, M. Xu, Q. Li. "Towards fast rerouting-based energy efficient routing," Computer Networks, 2014.

[7] D. Applegate and E. Cohen, "Making Intra-Domain Routing Robust to Changing and Uncertain Traffic Demands: Understanding Fundamental Tradeoffs," in Proceedings of ACM SIGCOMM 03, Karlsruhe, Germany, August 2003.

[8] O. Goldschmidt. "ISP Backbone Traffic Inference Methods to Support Traffic Engineering," In Internet Statistics and Metrics Analysis (ISMA) Workshop, San Diego, CA, December 2000.

[9] C. Tebaldi and M. West. "Bayesian Inference of Network Traffic Using Link Count Data," J. of the American Statistical Association, June 1998.

[10] J. Cao, D. Davis, S. Vander Weil, and B. Yu. "Time-Varying Network Tomography," J. of the American Statistical Association, 2000.



(a) Circle (b) Clique

Fig. 5. Circle and Clique topology of 6 nodes

[11] Harald Räcke. "Minimizing congestion in general networks," In Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science (FOCS), 2002. Co-Winner of Best Paper Award.

[12] Y. Azar, E. Cohen, A. Fiat, H. Kaplan, and H. R acke. "Optimal oblivious routing in polynomial time," In Proceedings of the 35th ACM Symposium on the Theory of Computing, 2003.

[13] M.Roughan, A.Greenberg, C.Kalmanek, M.Rumsewicz, J.Yates, and Y.Zhang. "Experience in measuring backbone traffic variability: models, metrics, measurements, and meaning," InProceedings of the 2nd Internet Measurement Workshop. ACM, 2002

[14] A.P. Bianzino, L. Chiaraviglio, M. Mellia, "Distributed algorithms for green IP networks," in: IEEE INFOCOM Workshop on Green Networking and Smart Grid, 2012

[15] C. Hou, F. Zhang, A. F. Anta, L. Wang, and Z. Liu. "A Hopby-hop Energy Efficient Distributed Routing Scheme," in Proc. of Greenmetrics Workshop, 2013.

[16] Yen, Jin Y. "Finding the k Shortest Loopless Paths in a Network," Management Science 17 (11), 1971

[17] "Abilene Core Topology," [Online], Available: https://itservices. stanford.edu/service/network/internet2/abilene

[18] "Network Topology," [Online], Available: http://geant3.archive.geant.net/Network/NetworkTopology/pages/home.aspx

[19] "Cernet2 Topology," [Online], Available: http://www.edu.cn/20060111/3170220.shtml

APPENDIX A
PROOF OF THEOREM 3

Proof. Fig. 5 shows the 6-nodes cycle and clique topologies. We label three links l_{ab}, l_{cd} and l_{ef} for our proof, let cap_{ab}, cap_{cd} and cap_{ef} be their link capacities. Now a arbitrary traffic comes, we compute the optimal routing and forward the demand on the topologies, let dem_{ab}, dem_{cd} and dem_{ef} be the traffic on each link. Without loss of generality, we say the link l_{ab} is the bottleneck, namely the link with MLUR.

Now let's consider the robust routing. When the routing changes, the traffic on each link also changes. We image a simple situation, just forward the traffic from one link to another path base on the optimal routing, and the link utilization of links on the path will increase, and accordingly the link utilization of former link will reduce to 0. Obviously, the routing must have a worse performance than the optimal robust routing, so its upper bound also is the bound of minimum OPRE.

For Case I.I, the adjusted link may be l_{cd}, the new bottleneck still be l_{ab}. The traffic on l_{ab} is increased by the origin traffic on link l_{cd}, So we compute OPRE by:

$$\frac{\frac{dem_{ab}+dem_{cd}}{cap_{ab}}}{\frac{dem_{ab}}{cap_{ab}}} = 1 + \frac{dem_{cd}}{dem_{ab}} \tag{11}$$

Because the old bottleneck is l_{ab}, so we have inequation:

$$\frac{dem_{cd}}{cap_{cd}} < \frac{dem_{ab}}{cap_{ab}} \tag{12}$$

From Ineq. (12) we get:

$$\frac{dem_{cd}}{dem_{ab}} < \frac{cap_{cd}}{cap_{ab}} \tag{13}$$

So we have OPRE in Case I.I combined by Eq. (11) and Ineq. (13):

$$1 + \frac{dem_{cd}}{dem_{ab}} < 1 + \frac{cap_{cd}}{cap_{ab}} \tag{14}$$

For Case I.II, the adjusted link also be l_{cd}, but the new bottleneck may be l_{ef}. Computing the OPRE similar as above:

$$\frac{\frac{dem_{ef}+dem_{cd}}{cap_{ef}}}{\frac{dem_{ab}}{cap_{ab}}} = \frac{dem_{ef} * cap_{ab}}{dem_{ab} * cap_{ef}} + \frac{dem_{cd} * cap_{ab}}{dem_{ab} * cap_{ef}} \tag{15}$$

Because the old bottleneck is l_{ab}, so we have Ineq. (12) and :

$$\frac{dem_{ef}}{cap_{ef}} < \frac{dem_{ab}}{cap_{ab}} \tag{16}$$

So we get inequation from Ineq. (16):

$$\frac{dem_{ef} * cap_{ab}}{dem_{ab} * cap_{ef}} < 1 \tag{17}$$

Also from Ineq. (12) we have :

$$\frac{dem_{cd} * cap_{ab}}{dem_{ab}} < cap_{cd} \tag{18}$$

Combine Eq. (15), Ineq. (17) and Ineq. (18), we have OPRE in Case I.II:

$$\frac{dem_{ef} * cap_{ab}}{dem_{ab} * cap_{ef}} + \frac{dem_{cd} * cap_{ab}}{dem_{ab} * cap_{ef}} < 1 + \frac{cap_{cd}}{cap_{ef}} \tag{19}$$

For Case II, the adjusted link may be l_{ab}, namely the old bottleneck link, the new bottleneck maybe l_{ef}. Computing the OPRE similar as above:

$$\frac{\frac{dem_{ef}+dem_{ab}}{cap_{ef}}}{\frac{dem_{ab}}{cap_{ab}}} = \frac{dem_{ef} * cap_{ab}}{dem_{ab} * cap_{ef}} + \frac{cap_{ab}}{cap_{ef}} \tag{20}$$

Becuase the old bottleneck has the maximum link utilization, so combine Ineq. (17) and Eq. (20) we obtain OPRE in Case II:

$$\frac{dem_{ef} * cap_{ab}}{dem_{ab} * cap_{ef}} + \frac{cap_{ab}}{cap_{ef}} < 1 + \frac{cap_{ab}}{cap_{ef}} \tag{21}$$

To sum up, we have inequation from Ineq. (14), Ineq. (19) and Ineq. (21):

$$1 + \max\{\frac{cap_{cd}}{cap_{ab}}, \frac{cap_{cd}}{cap_{ef}}, \frac{cap_{ab}}{cap_{ef}}\} < 1 + \frac{\max_{l\in E} cap_l}{\min_{l\in E} cap_l} \tag{22}$$

□

APPENDIX B
PROOF OF THEOREM 4

Proof. The spanning tree of circle is the topology which lose one link. In Appendix. A, when we consider the robust routing, we adjust the traffic on one link to another path, it is similar that we switch off the link and forward the traffic to some other path. The complete proof is almost the same with Appendix. A. □

APPENDIX C
PROOF OF THEOREM 5

Proof. Every link in the spanning tree will split the graph to two components, the flow between them must need trace this link. we say the bottleneck link is l_{ab} for optimal routing on the origin topology, and let l_{ef} be the bottleneck link for the robust routing on the spanning tree. Other symbols are consistent with Appendix. A. So we have OPRE:

$$\frac{\frac{\sum_{(i,j)\in E} dem_{ij}}{cap_{ef}}}{\frac{dem_{ab}}{cap_{ab}}} = \frac{cap_{ab}}{cap_{ef}} \sum_{(i,j)\in E} \frac{dem_{ij}}{dem_{ab}} \tag{23}$$

Because the l_{ab} is the bottleneck link, we have:

$$\frac{dem_{ij}}{cap_{ij}} < \frac{dem_{ab}}{cap_{ab}}, \forall (i,j) \in E \tag{24}$$

$$\frac{dem_{ij} * cap_{ab}}{dem_{ab}} < cap_{ij}, \forall (i,j) \in E \tag{25}$$

So we have OPRE:

$$\frac{cap_{ab}}{cap_{ef}} \sum_{(i,j)\in E} \frac{dem_{ij}}{dem_{ab}} < \sum_{(i,j)\in E} \frac{cap_{ij}}{cap_{ef}} \tag{26}$$

And we know that, the amount of l_{ij} between two parts must less than $\frac{n^2}{2}$, so OPRE must lower than:

$$\sum_{(i,j)\in E} \frac{cap_{ij}}{cap_{ef}} < \frac{n^2}{2} \frac{\max_{l\in E} cap_l}{\min_{l\in E} cap_l} \tag{27}$$

□