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 (OPR) with energy constraint, 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) or algebraic connectivity, 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 [**?**] 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 off/sleep 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 node; and 2) in which path to forward the traffic.

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 XXX, which chooses off/sleep line cards in a way that the OPRE can be minimized potentially. We then develop algorithm XXX 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

related work

III. Model

The network is modeled as a undirected graph $G = (V, E)$, where V is the set of vertices (i.e., routers, switches and end hosts), and E is the set of links. In graph G, two vertices u and v are called connected if G contains a path between u and v. Then We say graph G is connected, if and only if arbitrary pair of vertices in the graph is connected.

Let $\theta(G) = \{(V, E - \{e\})|e \in E\}$ denote the network set after removing the link e from G. Then, choosing the connected graph from $\theta(G)$ to consist of a new set, denoted by $\Theta(G) = \{G|G \in \theta(G)\ \&\&\ G\ is\ connected\}$. We call $\Theta(G)$ as successor of G.

A traffic matrix (abbreviation as TM below) is the set of traffic demands between each Origin-Destination(OD) pair in network G, and the $routing$ specifies how traffic of each OD pair is routed across the network. Usually, there are multiple paths for each OD pair and each path routes a fraction of the traffic. TM can be represented by a set of trinary group like (a, b, d_{ab}), where a and b is the origin and destination node respectively, d_{ab} is the traffic demand between the OD pair.

Let r denote the $routing$ mentioned above, which is specified by a set of values $f_{ab}(i, j)$ that specifies the fraction of demand from a to b that is routed on the link (i, j). So an OD pair contribute to the traffic of link (i, j) is $d_{ab}f_{ab}(i, j)$, and all the traffic across link (i, j) can be calculated as :

$$\sum_{(a,b,d_{ab})\in TM} d_{ab}f_{ab}(i, j) \tag{1}$$

Futhermore, we define the utilization of link as traffic acorss the link divide the capacity of link, as ;

$$u_{ij} = \frac{\sum_{a,b} d_{ab}f_{ab}(i, j)}{cap_{ij}} \tag{2}$$

where cap_{ij} is the capacity of the link (i, j).

A common metric of the performance for a given routing respect to a certain TM is the maximum link utilization ratio (MLUR). This is the maximum one of all links, Formally, the maximum link utilization ratio of a routing r on TM m in network $G(V, E)$ is

$$U_{r,m,G} = \max_{(i,j)\in E} u_{ij} \tag{3}$$

The optimal routing in all possible routes R for network G is a routing which minimize the MLUR, the minimum maximum link utilization ratio is called optimal utilization, can be represented by :

$$OptU_{m,G} = \min_{r\in R} U_{r,m,G} \tag{4}$$

The performance ratio of a given routing r on a given TM m and a given network G meaures how far from the routing to the optimal one, it is defined as the MLUR divided by optimal utilization, as following :

$$P(\{r\}, \{m\}, G) = \frac{U_{r,m,G}}{OptU_{m,G}} \tag{5}$$

We now extend the definition of performance ratio of a routing, from the specific TM m to TM set M.

$$P(\{r\}, M, G) = \max_{m\in M} P(\{r\}, \{m\}, G) \tag{6}$$

Obviously, the optimal routing in routing set R is the routing which minimize the extended performance ratio, such as :

$$P(R, M, G) = \min_{r\in R} P(\{r\}, M, G) \tag{7}$$

I.E. the routing r which arrive at the value of $P(R, M, G)$ is the most "robust" routing for the TM set M in the network G, and if the M range enough, we say that the "robust" routing is independent of specific TM.

But definition of "robust" above will not work well for next situation. Let us take a cycle network topology C as an simple example, in which we should choose one link to close. Before link cutting, the $P(R, M, C)$ is approximate to 2, but no matter which link is chosen to close, the cycle network will change to a line network L. Obviously, the $P(R, M, L)$ will always equal to 1. It means that there is no difference from removing which link, But the contradiction here is that, the choice for which link should be removed is really different because the links are not always the same with each other, such as their capacity.

The reason for the "fake robust" is that, the routing in the successor graph (i.e. L in above example) become unique, the current routing always be the optimal routing. More generally, we should make a little modification for the performance ratio when the network self changes.

Let G be the origin network, and the $G^* \in \Theta(G)$ be the successor network from G after removing some links, we define extended performance ratio between different graphs as the MLUR of successor graph divide the optimal utilization of origin graph, and get the minimum value of all routings, like :

$$P(R^*, \{m\}, G, G^*) = \min_{r\in R^*} \frac{U_{r,m,G^*}}{OptU_{m,G}} \tag{8}$$

where R^* is the routing set on network G^*.

And question is that how to meaure a successor network topology is "robust" enough for the TM set M when pruning is proceeding. We consider the worst situation, namely the

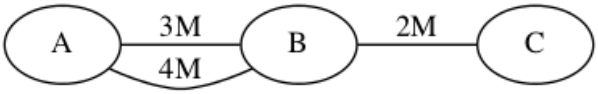


Fig. 1. Model Example

successor network topology has its maximum performance ratio when the TM is $m \in M$, described as following :

$$P(R^*, M, G, G^*) = \max_{m \in M} P(R^*, \{m\}, G, G^*) \tag{9}$$

Now we can say that, if a successor network arrive the minimum performance ratio, it is the "robust" successor network graph. Formally, we define the performance ratio as optimal successor performance ratio :

$$P^*(M, G) = \min_{G^* \in \Theta(G)} P(R^*, M, G, G^*) \tag{10}$$

where M is the TM set, and R^* is routing set.

If the scope of M is large engouh, the optimal successor network graph is also independent from specific TM.

A. Model Example

We will take an example to explain how to choose the link to close in our algorithm. Three hostes include : A, B, C, three links are with repectively capacity of 3M, 4M and 2M. For simpleness, we suppose there are two TM : $(A, B, 2M), (A, C, 1M)$ and $(A, B, 1M), (A, C, 1M)$. For each traffic matrix, the optimal route is obvious, we will trace 2M from A to B across the lower link and trace the 1M from A to C across the upper one for the first traffic matrix, whose maximum link utilization is 0.5. we will trace all the traffic acorss the lower link for the second traffic matrix, whose maximum link utilization is 0.5 as well.

Now for some reason, we will choose one link to shut down for power saving without lose connection of the network. There are two choice, remove either the upper link or the lower link. Let us take a little calculation: when remove the upper one, we should change all the traffic acorss the lower link, as a result, in the first TM the maximum link utilization is 0.75 and the second is 0.5; when close the lower link, we should trace all the traffic acorss the upper link, in the first TM the link utilization is 1 and the other is 0.667. So according to our theory, the $P^*(M, G)$ should be 1.5 and the optimal successor network topology will be the one which 3M link is closed.

IV. Algorithm

In our paper, Robust Energy-Aware Routing (REAR) algorithm works in two phases. Firstly, REAR select links should be put to sleep from the origin topology based on extended robust link utilization or algebraic connectivity, then compute the robust routing by demand-oblivious routing algorithm. There are two metrics for selecting removed links, our two-phase algorithm is flexible because we can replace the metric with whatever metric.

A. Algebraic Connectivity

Network topology is represented by $G = (V, E)$ as mentioned in Model section, where V is the set of vertices and E is the set of links. We say $A(G)$ is the Adjacency Matrix of graph G, that include information for which vertices of the graph are adjacent to which other vertices. $A(G)$ is a $N \times N$ matrix, where $N = |V|$ and non-diagonal entry a_{ij} equal to the number of edges from vertex i to vertex j, in this paper, a_{ij} always be 1 if $(i, j) \in E$ otherwise 0. And we stipulate the diagonal element a_{ii} be 0.

We say $D(G)$ is the Degree Matrix of graph G, which is a diagonal matrix and diagonal entry d_{ii} denote the degree of node i. It is obvious that there is $d_{ii} == \sum_j a_{ij}$.

Then we define Laplacian Matrix $L(G)$ of graph G as the difference between Degree Matrix and Adjacency Matrix :

$$L(G) = D(G) - A(G) \tag{11}$$

where $D(G)$ is the Degree Matrix and $A(G)$ is the Adjacency Matrix.

In the mathematical field of graph theory, the number of eigenvalues equal to 0 is the number of connected componements of G, so the smallest eigenvalue always be 0 in arbitrary graph . And we call the second smallest one as algebraic connectivity, which is greater than 0 if and only if graph G is connected. Further more, it measures the connectivity and stability of graph, the greater of the value, the more connective of graph; and it is a metric of average distance between any two vertices of graph G. We call the algebraic connectivity as $\lambda_2(G)$.

Topology always have different algebraic connectivity values, it is clear that when one link is added or removed from the graph, the algebraic connectivity value changes accordingly. And we say the changed value is the impact of this link on the graph. Supposed we sleep link l from graph G, new graph is described as G^*, we defined the impact of link l as :

$$\Delta_l = \lambda_2(G) - \lambda_2(G^*) \tag{12}$$

where $\lambda_2(G)$ and $\lambda_2(G^*)$ is algebraic connectivity of G and G^* respectly.

Clearly, Δ_l is always greater than 0, because graph will always lose connectivity when link is removed. Further more, some link will play a more important role in the connectivity of graph, such as the backbone link of network topology. And we say a link l affect more if Δ_l is greater.

B. Extended Robust Link Utilization

The demand-oblivious routing is not always the best one for specific TM, but good enough for a range of TM. we mentioned the definition of link utilization above when demand and routing are determined, as demand become oblivious, the definition is also not suitable. So we define a notation named extended robust link utilization (ERLU) replace the normal link utilization like :

$$u_{ij}^e = \frac{\sum_{a,b} f_{ab}(i, j)}{cap_{ij}} \tag{13}$$

Proposing this definition for two consideration: firstly, the more flows across the link, the greater the u_{ij}^e is; secondly, the more fraction of one flow across the link, the greater the u_{ij}^e is. The link with greater ERLU is also more important than others in a sense. Although there is no demand here, we mean there is more probability that this link have greater link utilization when specific demand come. Similarly, we define the impact of link l as :

$$\Delta_l = u_l^e \tag{14}$$

And there is the same conclusion that a link l affect more if Δ_l is greater.

C. Algorithm Phase One

REAR sleep as many links as possible without losing much connectivity or transportation of graph for different metrics. Originly, we should calculate impact of all the links and sleep the lowest one from the graph, then repeat calculate and remove process until arrive some specific threhold. Obviously, it is NP-Hard, following is a heuristic algorithm.

For the origin graph, we calculate the impact of every link as Δ_{l_i} , and then sort these values from small to big, output ordered list denoted as Γ:

$$\Gamma = \{..., l_i, ..., l_j, ...\} \tag{15}$$

where $\Delta_{l_i} < \Delta_{l_j}$.

Pay attention we only compute the link impact once at the begining of algorihtm, and the ordered list Γ show the order of 'importance' among links in the graph.

Now we begin selecting which links should be sleep. We denote the set of the sleeping links as S, and the output of this phase is final graph $G^* = (V, E - S)$. We set $S = \emptyset$, and repeat our selecting process, each iteration we select one link, remove it from Γ and put it into S. In iteration i, algorithm scan links as the order in Γ, we try to remove this link from the graph to check if the graph is still connected and the energy conservation have not arrive the threshold. If so, we select this one then go next iteration. Otherwise choose the next link from Γ for trying to remove. Algorithm stop until all the links in Γ is tried but no one is satisfied with both connectivity and power threhold.

So before going ahead our algorithm, there is another thing we should done, how to measure the power of graph. We simple take an power model from [**?**] showed in Table II, and defined the difference of power consumption between two graphs as:

$$diff_p = \frac{\rho(G_{S,l}^o)}{\rho(G^o)} * 100 \tag{16}$$

where $\rho(G_{S,l}^o)$ is the power consumption of the final graph, when the links set S and link l are both removed from the origin graph G^o, and the $\rho(G^o)$ is the power consumption of the origin graph.

If we set $diff_p$ valued 90%, it means that whenever we try to remove the link l from the origin graph in iteration, the power consumption should never lower than the 90% of origin one. In another words, the output of this phase protect as much connectivity and transportation as possible. Following is our implementation:

Algorithm REAR : Phase One

Input: $G(V, E)$, $threhold$;
Output: S in which links should be switched off;
$G(V, E - S)$ which is the final network topology;

```
for each link l in E
    G* ← G(V, E − {l});
    Γ[l] ← Δ_l ← λ2(G) − λ2(G*);
Resort Γ in increasing order based on Δ_l;
S ← ∅, goon ← true;
while goon
    goon ← false;
    for each link l in Γ − S
        if G_{S,l} is connected and ρ(G_{S,l})/ρ(G) > threhold
            S ← S ∪ {l};
            goon ← true;
            break;
return S, G(V, E − S);
```

D. Algorihtm Phase Two

Once we get the output network topology from the first phase based on either metric, it is time to compute the robust routing. On one hand, computation process may cost too much time if we directly calculate the robust routing in the final graph; on the other hand, the robust routing based on the final graph may not be the best one. There is a heuristic algorithm based on the demand-oblivious routing on the origin graph, which should be obtained at first. then adjust routing in details according to the links we switched off.

The demand-oblivious routing can be computed by a single LP with $O(mn^2)$ variables and $O(nm^2)$ constraints[1] :

min r
$f_{ij}(e)$ is a routing
$\forall$ links l: $\sum_m cap(m)t(l,m) \leq r$
$\forall$ links l, $\forall$ pairs $i \to j$:
$\quad f_{ij}(l)/cap(l) \leq p_l(i,j)$
$\forall$ links l, $\forall$ nodes i, $\forall$ edges $e = j \to k$:
$\quad \pi(l, link - of(e)) + p_l(i,j) - p_l(i,k) \geq 0$
$\forall$ links l, m: $\pi(l,m) \geq 0$
$\forall$ links l, $\forall$ nodes i: $p_l(i,i) = 0$
$\forall$ links l, $\forall$ nodes i, j: $p_l(i,j) \geq 0$

where the $cap(l)$ is the capacity of link l; and $\pi(l,m)$ is the weights for every pair of links l, m; and the variables $p_l(i,j)$ for each link l and OD pair i, j is the length of the shortest path from i to j according to the link weights $\pi(l,m)$.

The routing we get indicate how to arrive at destination node from source node for every OD pair in the origin topology. What is different is that, the flow can be splited in the routing, i.e. there may be two paths ($path_1$, $path_2$) both from source node s to destination node d, and the optimal obilious routing trace 70% traffic on $path_1$ and left on $path_2$. Although splitting flow is hard handled, we take a transformation for the case like that: when an flow is coming, there is 70% probability we trace it on $path_1$, otherwise $path_2$. This is easily implementated in real world.

Because all the routing is based on the origin topology, when some links are switched off, we must adjust the routing

as well. Supposed there are some paths between two vertices s and d, maybe one link in some paths be removed, and these paths become not reachable any more, we should adjust the traffic in these paths to other paths. Of course, we can not put the whole traffic on another path, this will make some links of the path congested, so we should split the traffic to some paths 'averagely', in the sense of extended robust link utilization. Similarly to link utilization, we define the extended robust link utilization as the maximum extended robust link utilization of links.

Take pair (s, d) for example, routing includes paths from s to d, such as $p_1, p_2, ..., p_n$. While only p_1 trace the removed link, and of course it will be unreachable. We split the traffic of p_1, and put them on other paths to make the extend robust link utilization of paths almost closely.

And this is our implementation:

Algorithm REAR : Phase Two

Input: $G(V, E)$ which is the origin topology;
S which is the switch-off links set generated by Phase One;
R which is the Robust Routing on origin topology;
Output: $Routing$ Robust Energy-Aware Routing on new topology

```
1: G* ← G(V, E − S);
2: for each link l in S
3:    for each s, d, paths in R
4:       traffic ← 0;
5:       for each path in paths
6:          if link in path
7:             traffic ← traffic + path.traffic;
8:             paths.remove(path);
9:       yen_paths ← yens_algorithm(G*, s,d);
10:      while traffic != 0
11:         sort_paths_by_extend_robust_link_utilization(yen_paths)
12:         yen_paths[0].traffic = traffic / N
13:         traffic = traffic - traffic / N
14:         paths.add(yen_paths[0])
15:      paths.merge();
16: return R;
```

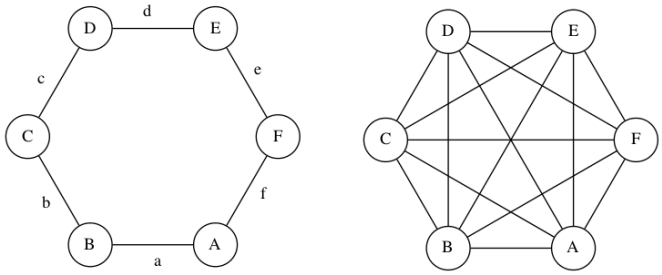
V. Samples and Proof

For showing obilious performance ratio with energy constraint really make a difference in the switching off link process, we will explain it in two simple topologies, cliques and cycles. On the other hand, we also say that there will be an upper bound for obilious performance ratio with energy constraint, and get the bound according to the topology. For simpleness, we refer the energy constraint to the quantity of removed links.

A. Circles

Cycles topology connect all the nodes by a cycle, any node in which has two links joined. Figure 1 show the six nodes cycles topology, let nodes named from A to F and links named from a to f in the figure. Now suppose a traffic in the topology then calculate the optimal routing for it, let D_a and C_a be the demand and capacity of the link, without loss of generality, we say the link d is the bottleneck, i.e. with the maximum link utilization. There are two solutions for us, switching off the link d, or the other link, such as link f.

For case one, when we switch off the link d, the origin traffic pass the link will be redirected. The worst situation is



(a) Circle (b) Clique

Fig. 2. Circle and Clique topology of 6 nodes

that all the traffic on the link d will be put to the other path : $D \leftarrow C \leftarrow B \leftarrow A \leftarrow F \leftarrow E$. After traffic redirection, the bottleneck link may become the link e, according to our definition of extended robust link utilization:

$$\frac{\frac{D_e+D_d}{C_e}}{\frac{D_d}{C_d}} = (1 + \frac{D_e}{D_d})\frac{C_d}{C_e} \tag{17}$$

Because the link utilization of link d is the maximum one, we have :

$$\frac{D_d}{C_d} > \frac{D_e}{C_e} \tag{18}$$

$$\frac{D_e * C_d}{C_e * D_d} < 1 \tag{19}$$

so we have the extended robust link utilization in case one :

$$(1 + \frac{D_e}{D_d})\frac{C_d}{C_e} < \frac{C_d}{C_e} + 1 \tag{20}$$

For case two, similar to case one, all the traffic will be redirected on the other path between F and A. There are two sub cases after traffic redirection, the link D is still the bottleneck or the link E become the new bottleneck. In sub case one:

$$\frac{\frac{D_d+D_f}{C_d}}{\frac{D_d}{C_d}} = 1 + \frac{D_f}{D_d} \tag{21}$$

Because the link utilization of link d is the maximum one in origin topology, we have as:

$$\frac{D_d}{C_d} > \frac{D_f}{C_f} \tag{22}$$

$$\frac{D_f}{D_d} < \frac{C_d}{C_f} \tag{23}$$

So we get the extended robust link utilization in this case :

$$(1 + \frac{D_f}{D_d}) < \frac{C_d}{C_f} + 1 \tag{24}$$

In sub case two, we get the new extended robust link utilization in link E:

$$\frac{\frac{D_e+D_d}{C_e}}{\frac{D_d}{C_d}} = \frac{D_e * C_d}{D_d * C_e} * \frac{C_d * D_f}{C_e * D_e} \tag{25}$$

Similarly as mentioned above, we get equations :

$$\frac{D_d}{C_d} > \frac{D_e}{C_e} => \frac{D_e * C_d}{C_e * D_d} < 1 \tag{26}$$

$$\frac{D_d}{C_d} > \frac{D_f}{C_f} => \frac{D_f * C_d}{D_d} < C_f \tag{27}$$

So we get the extended robust link utilization:

$$\frac{D_e * C_d}{D_d * C_e} + \frac{C_d * D_f}{C_e * D_d} < 1 + \frac{C_f}{C_e} \tag{28}$$

As a result, the upper bound of extended robust link utilization must be :

$$Max\{1 + \frac{C_d}{C_e}, 1 + \frac{C_d}{C_f}, 1 + \frac{C_f}{C_e}\} \tag{29}$$

Particularly, if all the links have the same capacity, the upper bound equal to 2 when switch off one link. Otherwise, the upper bound only dependent to the greatest ratio of link capacity.

B. Cliques

In clique topology, all the nodes connect to each other, it means that situation become more difficult because the traffic can be adjusted to multi paths rather than one path in circle. And we will see that splitting traffic on multi paths is always better than put all the traffic to one, so if we get the upper bound of the worst case, it must be the upper bound of other case. Similarly to circle topology, we say the link a is the bottleneck, and we will switch off it or other link like b.

In case one, after we switch off link a, the new bottleneck link must be the affected link, the link contains the origin traffic on link a. Otherwise if the link b is the bottleneck, but we adjust the traffic on other links, it means that we can do the same thing in the origin topology, and the bottleneck become link b rather than a. It is conflict with our assumption. So we have the extended robust link utilization as :

$$\frac{\frac{D_b+D_a}{C_b}}{\frac{D_a}{C_a}} = (1 + \frac{D_b}{D_a})\frac{C_a}{C_b} \tag{30}$$

Because the link a is the bottleneck link with the maximum link utilization, like what we do above, we can get

$$(1 + \frac{D_b}{D_a})\frac{C_a}{C_b} < 1 + \frac{C_a}{C_b} \tag{31}$$

In case two, the bottleneck may still be link a or the new link c. If it is link a, and no new traffic across it, the extended robust link utilization equal to 1 obiliously. Or we adjust the new traffic D_b on it, we can get the extended robust link utilization:

$$\frac{\frac{D_a+D_b}{C_a}}{\frac{D_a}{C_a}} = 1 + \frac{D_b}{D_a} < 1 + \frac{C_b}{C_a} \tag{32}$$

TABLE I
TOPOLOGIES

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

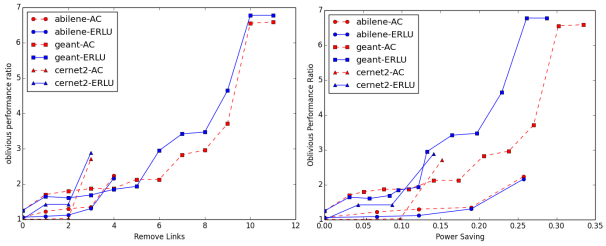


Fig. 4. OPR with Power Saving

If the bottleneck become new link c, similar to our proof of circle topology, we can get extended robust link utilization:

$$\frac{\frac{D_c+D_b}{C_c}}{\frac{D_a}{C_a}} = \frac{D_c * C_a}{D_a * C_c} + \frac{D_b * C_a}{D_a * C_c} < 1 + \frac{C_b}{C_c} \tag{33}$$

So we obtain the same conclusion as circle topology, the extended robust link utilization must be :

$$Max\{1 + \frac{C_a}{C_b}, 1 + \frac{C_b}{C_a}, 1 + \frac{C_b}{C_c}\} \tag{34}$$

C. Other Topology

We can also see that, when we switch off different link, it really matters the performance which dependent on not only the demand or traffic, but also the capacity of links. Particularly, the upper bound is just related with the capacity of links. And from our proof process, there is no special limitation for specific topology, so we can get the similar conclusion on other topology.

VI. Experiments and Results

We simulated our algorithm on real world topology, including Abilene, Geant and Cernet2, whose number of nodes and links are listed in Table I. Then we generate 1000 random traffic matrix with Gravity Model [**?**], which assume that the traffic demand between nodes is proportional to their combined capacity of connecting links. To extrapolate a complete TM, we take an attribute w to scale the traffic range from $1/w$ to w base on the basic traffic demand. Particularly, when the w limit extremity, we say the traffic matrix is really arbitrary, and our algorithm is irrelevant with traffic.

A. OPR versus Power Saving

In Figure 4(a), we see that the value of OPR is at least 1 even in the origin topology, particularly in Geant the value is 1.24, it means the robust routing always not be the optimal for all TMs even no links removed from topology. It is reasonable

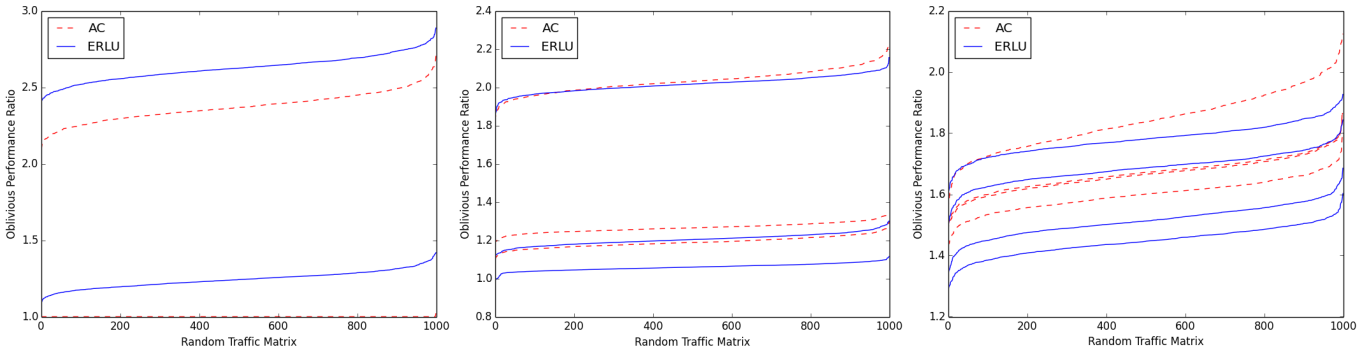


Fig. 3. Sort: (a). Abilene (b).Geant

TABLE II
GREEN TE POWER MODEL

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

because our robust routing is used for a range of traffic rather than a specific TM.

When we switch off links, the topology lose connectivity and some routing paths will be failed, traffic on those paths will be adjusted to others, which result in the value of OPR increased. We take margin as 1.5 and observe this phenomena from Figure 4(a). And all the curves can split in two phases according to the growth speed of OPR, for example in Abilene, the OPR rise geantly before 4 links are removed, and rise rapidly since then. It is revelatory for implementing our algorithm in real life, controlling the number of removed links less than the turning point is more efficient. The another reason for avoiding removeing too much links is that the robust routing will be less 'robust', because there is only one path between two vertices in the minimum connected topology.

On the other hand, we concern more about how is the OPR varifying when achived specific power saving target. Obviously, the more links we removed, the more power we saved, but how to quantify the power of one link is difficult. Green TE [**?**] proposed a simple power model, which can be represented in Table II. In which, the total energy of topology is dominant by

We compute the power saving ratio as the total power of the removed links over the total power of all the links. And we can save 19% power with only 34% worse than the worst case for all the traffic in Abilene topology. In Geant and Cernet2, the OPR is little higher. Also we can see that two ways for removing links perform almost the same at the begin in Geant, and varified at the end. It shows that we may find another better way for removing the links in the future.

For computing the worst case, we generate 1000 traffic

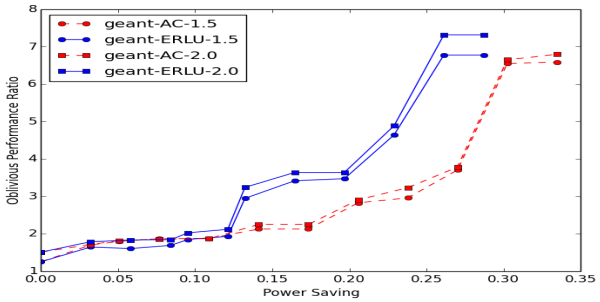


Fig. 5. OPR with w relationship

matrix for every topoloy and the w with the Gravity Model. We show the each result in the Figure x and x when we remove the first 4 links in each topology. In Abilene, the result is more tighter than Geant. Comparsion in two ways, we can see that in Abilene the AC method is worse than the ERLU one, and the downmost zone is the same in two ways, because the robust routing is same in the origin topology.

From Figure xx and Figure xx, we sort the 1000 TMs by their OPR value, and see that most of the TMs are quite average. It is said that, our metric OPR is always not be achived for most traffic matrix.

We know when we compute the robust routing for specific topology, the w is very important. When the w limit extremely, our robust routing is said without knowledge of traffic matrix. Figure xx, show the OPR varified when w changes. The greater the w is, the greater the OPR is. It is obvious because when the w is greater, the traffic matrix is random in more wider range, and our OPR is the worst case of all the traffic matrix.

VII. FUTURE AND ISSUE

Future and Issue

VIII. CONCLUSION

The conclusion goes here.

REFERENCES

[1] M.Zhang, C.Yi, B.Liu and B.Zhang, "GreenTE: Power-Aware Traffic Engineering".

[2] D.Applegate and E.Cohen, "Making Intra-Domain Routing Robust to Changing and Uncertain Traffic Demands: Understanding Fundamental Tradeoffs".

[3] 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