Multipath Routing for Reducing Network Energy

Yong Oh Lee and A. L. Narasimha Reddy
Dept. of Electrical and Computer Engineering, Texas A & M University
Email:yongoh.lee@tamu.edu, reddy@ece.tamu.edu

***Abstract*—Reducing energy consumption in networks has been recently receiving attention. Networks are typically designed to support peak loads. Energy saving is potentially achieved by appropriately provisioning network capacity, by shutting down unnecessary network elements while the capacity of remaining network elements still meets traffic demand. Traditionally, topology control techniques are employed to reduce the network to an appropriate size. In this paper, we propose techniques to construct an appropriate topology that takes the traffic demands into account. We also study how much multipath routing can contribute to energy saving compared to single/shortest path routing. We show that the proposed techniques reduce energy consumption by an additional 17% over previous schemes.**

***Index Terms*—Green network, multipath routing, bin packing problem, topology control**

I. Introduction

Reducing energy consumption has been studied in the backbone networks and the data center networks. The opportunity to save energy consumption comes from the fact that network capacity is normally provisioned for peak traffic loads. The average link utilization could be less than 30-40% and the duration of peak traffic load a small fraction of the entire day [1]. Reducing the link capacity during off-peak duration is a promising scheme to save energy consumption. To deal with the variations of traffic load at different times of the day, topology control and traffic engineering can be used to shutdown some links and nodes of the network while leaving the network connected and with sufficient capacity to carry the traffic load.

Traffic engineering approaches power off the network elements while the powered-up network capacity meets the traffic demand. Powering off the links is one of the strategies to reduce the energy consumption [2]. However, several studies report that the power consumption of a node is much higher than the power consumption of a link. In [3], authors attempt to power off the nodes first with single shortest path routing. However, the number of active node is not greatly reduced.

In this paper, we study how to reduce the number of active nodes for energy savings. We propose a new topology control based on Steiner tree approach and multipath routing based approach. Our topology control takes an approach of building an appropriately provisioned network to meet the demands of all the terminal nodes. This is in contrast with existing approaches where nodes and links are removed from the given network. Also, we study the effectiveness of employing multi-path routing to reduce energy consumption. While multi-path routing allows more possibilities for routing the traffic demands, if longer alternate paths are employed, the power consumption can actually increase. We propose an effective multi-path routing strategy for reducing energy consumption. We evaluate the proposed schemes by various simulations. In most cases, even if single shortest path is used, our topology control achieves more energy saving with a smaller number of active nodes than previous schemes. Our topology control reduces the number of iterations for finding a suitable topology. In addition, multipath routing with topology control reduces the number of active nodes further, and achieves more energy savings.

II. Related work

Traffic engineering approaches allow to power off unnecessary nodes/links while the remaining network capacity meets traffic demand [2], [3], [4]. Furthermore, the approach in [5] shuts down individual links in a bundled link. The approach in [6] also optimizes the energy consumption of data centers by powering down unneeded links and nodes in a similar manner. These approaches propose heuristic algorithms based on underlying optimization problems. All of these schemes use single shortest path routing on the topology where unnecessary nodes and links are removed. However, these schemes do not result in reducing a big fraction of the number of nodes in the network.

We consider the potential for reducing energy consumption through multi-path routing. While multi-path routing increases the routing possibilities and hence the chances of consolidating the traffic into fewer links, longer alternate paths can result in higher resource consumption. Also, traffic splitting on multiple paths could cause bad TCP performance due to the reordered packets at the destination [6]. Reordering problem can be solved by employing a flow-based routing scheme or through reorder-resistant versions of TCP [7].

We selectively power off nodes and links of the topology. As mentioned, power consumption of node is much higher than that of links. So, we minimize the number of powered-on nodes first, and then reduce the number of powered-on links. Similar strategies have been employed in [3] and [6]. Some earlier schemes focus only on link removal [2], [4], and [5]. In [14], multipath routing is used for energy saving, but only considers powering off links. Our results show that this earlier approach based on linear programming takes too long to compute a new topology in real-time when traffic demand changes.

III. Problem formulation

To minimize energy consumption, an optimization problem considering the network topology and the traffic demand is solved. An optimal solution minimizes the number of nodes and links, while all traffic demands are delivered with maximally utilized link capacities.

$G(V,E)$ represents the network as a directed graph that consists of a set V of nodes and a set E of links. The total number of nodes is $|V|$, and the total number of links is $|E|$. Let t^{sd} be traffic demand: the amount of traffic going from node $s=1,...,|V|$ to node $d=1,...,|E|$.

Let $x_{ij} \in [0,1], i=1,...,|V|, j=1,...,|V|$ be binary variable being the value of 1 if link (i,j) is present and powered on, and the value of 0 if link (i,j) is not present or powered off. Similarly, let $y_i \in [0,1], i=1,...,|V|$ be binary variables of router power status.

Let P_L and P_N be the power consumption of link and node, respectively. Here, we assume the power consumption of link and node is constant, because turning on network components consumes most of power, and today's network components are not energy proportional.

Given these definitions, the linear programming of minimizing the energy consumption follows.

Minimize

$$P_N \sum_{i=1}^{|V|} y_i + P_L \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} x_{ij} \quad (1)$$

Under the constraints of (1) the flow conservation of a multi-commodity minimum cost flow problem, (2) the total load on a link to be less than the link capacity and (3) powering off a node only if all bi-directional links are powered off.

This problem is a multi-commodity minimum cost flow problem known as NP-hard [9]. In addition, P_N and P_L vary widely depending on the devices, but generally P_N is much higher than P_L. So, we follow the assumption, $P_L << P_N$. In this case, this problem aims to switch off the largest possible number of network nodes first and then switches off as many links as possible.

IV. Overview of the Proposed Scheme

The proposed scheme relies on the intuition that the energy saving achieved by powering off nodes is higher than by powering off single links.

For powering off nodes, we construct a network, that has the connectivity between the terminal nodes, with a minimum number of nodes. If the capacity of the constructed network is not enough to carry the traffic demand, we add nodes until the capacity meets the traffic demand. Different from earlier schemes [3] that deletes nodes and links iteratively, our scheme has less complexity of finding a suitable topology, and reduces the number of iterations of feasibility testing. We describe the detail of this topology control in section V. The required topology can be constructed either with single path routing or multi-path routing. In single path routing, the traffic from an origin takes the shortest path to the destination and all the traffic between that od-pair takes the same path. In multi-path routing, we allow the traffic between an od-pair to be split up and sent along multiple paths. This allows more possibilities for consolidating the traffic onto fewer nodes and links and hence the constructed topology depends on the choice of the routing algorithm.

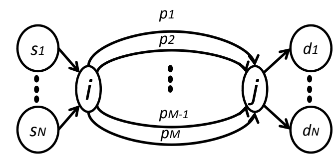


Fig. 1. Example of comparing energy saving between single path and multipath

We consider the potential for multipath routing to aid in energy savings. When the topology control checks whether the capacity of a subset of network elements is enough to carry traffic demand, multipath routing may need smaller number of network elements than single shortest path routing. We show through an example that multipath routing can power-off more nodes and links on the topology of Figure 1. The given topology has N od-pairs (s_1-d_1,s_2-d_2,...,s_N-d_N). Each od-pair has M available paths (p_1,p_2,...,p_M), and the capacity of each path between i and j is 1 ($C_{ij}(p_k)=1, k=1,2,...M$, where $C_{sd}(p)$ is the minimum link capacity on the path p from s to d). Suppose the traffic demand of each od-pair is $x \in [0,1]$. Any path can deliver $\lfloor \frac{1}{x} \rfloor$ od-pairs , if od-pairs use a single path, and do not allow traffic split. So, the minimum number of path for N od-pair is $\left\lceil \frac{N}{\lfloor \frac{1}{x} \rfloor} \right\rceil$ with single path routing. However, if od-pairs use multiple paths and allow traffic split, the minimum number of paths for N od-pair is $\lceil Nx \rceil$. We can reduce the number of path used for delivering traffic as $\left\lceil \frac{N}{\lfloor \frac{1}{x} \rfloor} \right\rceil$-$\lceil Nx \rceil$ by using spliting traffic on multiple paths.

V. The Proposed Topology Control

Since turning off a node can achieve more energy saving than turning off a link, we turn off nodes first, and then turn off links. Turning off nodes begins with finding a small number of nodes that can provide connectivity to all the terminal nodes. Finding the minimum number of nodes with connectivity is NP-hard [8]. We use a heuristic based on the construction of a minimal spanning tree approximation Steiner tree (MST-Steiner) [8].

After finding the minimum number of nodes with connectivity, traffic demand flowing through the network element is routed by the proper routing scheme. (We will discuss this routing scheme in section VI.) When the traffic is assigned, we check traffic demand constraints (Constraint 1) and the link utilization constraints (Constraint 2). If no violation is present, then we move to switching off links. If a violation occurs, we add nodes by the following sorted order.

The node set is sorted by routing density, which is the sum of routing density of links, $X_i = \sum_{j=1}^{N} x_{ij} RD_{ij} + \sum_{j=1}^{N} x_{ji} RD_{ji}$, where RD_{ij} is the routing density of link(i,j). Routing density of a link is a count of how many times a link is incident on the shortest path between different od pairs. Nodes with large value of routing density are checked first, i.e., V is sorted in decreasing value of routing density. The complexity of computing routing density is dominated by the complexity of the length and the number of all shortest paths. Consequently, the complexity of routing density is $O(|E|+|V|^2log|V|)$ [13].

We repeat adding extra nodes to the topology until traffic can be routed with no violations of constraints stated earlier. It is expected that adding highly utilized nodes will require fewer nodes and hence lead to larger energy savings.

After finding the minimum number of nodes, we try to find the minimum number of links on the subset of network elements. This method is referred as **LF** (Least Flow first) in [3]. The link set is sorted by the amount of flow on the link. Links with a smaller amount of flow on the link are checked first, i.e., E is sorted in increasing value of $F_{ij} = f_{ij} + f_{ji}$. Switching off the cable (i, j) affects link(i, j) and link(j, i),in both directions and it is expected that links with smaller amount of traffic can be switched off and the traffic on this link can be routed to other links in the reduced topology (without this cable). If switching off a link does not violate connectivity, traffic demand and link utilization constraints, the link is switched off. The algorithm progresses turning off as many links as possible and terminates when all $e \in E$ are visited.

Both our topology control and the scheme in [3] use a similar scheme for switching off links. In our proposed topology control, the complexity of the Steiner tree graph with the minimum number of nodes is $O(|V||E|(log|V|)(log|E|))$. The complexity of sorting the removed nodes and finding K-shortest paths is $O((|V|-|V_m|)|V|(|E|+|V|^2log|V|))$, where $|V_m|$ is the number of the removed nodes after 'MST-Steiner'. The latter one is greater than the former. The complexity of the proposed topology control is the complexity of sorting the removed nodes and finding K-shortest paths times the complexity of the feasibility test to ensure traffic can be routed. With single shortest path routing, the complexity of feasibility testing does not change based on the topology control algorithm. However, the number of iterations of feasibility test in the proposed topology control is $O(|V| - |V_m|)$, compared to earlier scheme of V. We can say our topology control is slightly less complex than the topology control in [3]. The complexity of feasible test with multipath routing becomes K times the complexity of feasiblity test with single shortest path routing, if k-shortest paths are used.

VI. The proposed multipath routing

In this section, we present our approach to assign the traffic demands of an od-pair among the multiple paths allowed by the multi-path routing algorithm. It is noted that our approach does not constrain the choice of multi-path routing algorithms. Among the many choices available to route the traffic between an od-pair, our algorithm allocates the whole traffic or fractions of that traffic to different paths with a view to minimizing the number of nodes and links in the network.

This problem of assigning the traffic demand to multiple paths can be seen as a modified bin-packing problem, where available paths are the bins, and the traffic demand is the object to be packed. In order to assign the traffic demand to a path, we look at the minimum residual capacity of all the links on that path and employ it as the bin capacity for that path. In single/shortest path routing, the traffic demand between an od-pair cannot be split and hence has to be assigned as a whole to one of the shortest paths. In multi-path routing, the traffic demand can be split and allocated to the available paths. The choice of paths and the choices of splitting the traffic demand can determine the performance. We propose two bin packing algorithms in order to assign the traffic demand of an od-pair to multiple paths.

Link flows are computed by multipath routing done through bin packing in order to minimize the number of nodes and links in topology control. We introduce two bin packing algorithms and explain how to use these bin packing algorithms for multipath routing. At any given time, the choice of available paths for an od-pair constrain the choices of bins. We consider all available path as bins, and divide into these bins into several set of 'closed bin set' which satisfies a certain condition, i.e. path length.

- **Closed first fit:** Closed first fit is first fit algorithm within the closed bin set.
- **Closed next fit with fragmentation:** Closed next fit with fragmentation is next fit with fragmentation algorithm within the closed bin set. When an item does not fit in an open bin, it is fragmented into two parts. The first part fills the open bin and the bin is closed. The second part is packed into a new bin which becomes the open bin.

We propose multipath routing for energy saving with the proposed bin packing algorithms.

First, we sort the order of od-pairs. Before sorting, each od-pair computes multiple paths whose path length is from the length of the shortest path to the length of the shortest path plus two extra hops. If the number of possible paths is larger than K, we restrict to K-shortest paths. To enhance usage of multipath, the order of od-pairs is sorted by the number of paths. If there is no room for assigning traffic on the shortest path, our proposed scheme can assign traffic on an alternative (possibly longer) path. However, some od-pairs have small number of paths. In order to increase the possibility of finding alternate paths, we first consider od-pairs with smaller number of paths. As traffic gets assigned to available links, the number of possible paths with sufficient capacity to route a given od-pair decrease. Hence having a larger number of choices for later assigned od-pairs is expected to increase the chances of success of assigning the traffic demand to a smaller number of links and nodes. Similarly od-pairs that have smaller paths lengths and smaller traffic demands are expected to be easier to route. Based on this rationale, we tested these sorting criterions with different orders. (We omit these results due to space limit.) We sort od-pairs by the number of paths first. Then, the second sorting criterion is path length, and the third is volume of traffic demand.

After sorting od-pairs, we assign the traffic demand to multiple paths. To assign the traffic demand of an od-pair, we sort the multiple paths by path length, routing density, and capacity. Routing density is expected to given an indication of how critical or useful this link can be in routing various flows. If we have multiple paths as options for an od-pair, we choose paths and the amount of traffic split on the paths so that remaining network capacity of the links remains as high as possible. It is expected that maximizing the remainder capacity makes it easier to route later od-pairs. The preferred path is the path with low path length, and low routing density and retains large residual capacity for other traffic demands. We sort multiple paths by path length first, then by routing density, and then by residual capacity.

We denote $PATH_0(s,d)$ is the set of paths from s to d with the shortest path length. $PATH_1(s,d)$ is the set of paths from s to d with an extra hop. $PATH_2(s,d)$ is the set of paths from s to d with two extra hops. t^{sd} is the traffic demand from s to d. We denote the path capacity as the minimum remaining link capacity on the path.

First, we assign t^{sd} on $PATH_0(s,d)$ by closed first fit. If the capacity of the paths of $PATH_0(s,d)$ is not enough to assign t^{sd}, we assign t^{sd} by closed next fit with fragmentation. We repeat these steps util t^{sd} is assigned by closed first fit and by closed next fit with fragmentation to $PATH_1(s,d)$ and $PATH_2(s,d)$, if the part of t^{sd} remains after assigning to $PATH_0(s,d)$. The pseudo code is presented in Algorithm 1. 'closedFirstFit(BinSet,Item,BinCapa)' and 'closedNextFit-Frag(BinSet,Item,BinCapa)'are closed first fit and closed next fit with fragmentation where "BinSet" is a set of bins, "Item" is item to be assigned, and "BinCapa" is a set of capacities of bins.

Algorithm 1 Multipath routing for energy saving

Input: G,odpairArray,TD,$PATH$
Output: checkFlow

```
odpairSortedArray=sorting(odpairArray, [PATH, TD])
i ← 0
checkFlow ← TRUE
for all (i, j) ∈ E do
    u(i, j) ← 0
end for
while i < N(odpairSortedArray) & checkFlow = TRUE do
    s = odpairSortedArray[i].src
    d = odpairSortedArray[i].dst
    j ← 0
    TD' ← t^sd
    while TD' > 0 & j < 3 do
        (TD', u)=closedFirstFit(PATH_j(s, d), TD', u)
        if TD' > 0 then
            (TD', u)=closedNextFitFrag(PATH_j(s, d), TD', u)
            j++
        end if
    end while
    if TD' > 0 then
        checkFlow ← FALSE
    end if
end while
```

VII. Evaluation

To evaluate the effect of our proposed techniques, we simulate various scenarios on synthetic topologies (an n-by-n grid topologies and a random topology) and real topologies (Abilene and Sprint topology).

TABLE I
Power consumption of various energy models in watts

	EM1	EM2	EM3
P_N	151	133	76
P_L	11	10.5	6.5

The proposed topology control with shortest path routing is denoted by ST-S (Steiner Tree approach-Single path routing). The proposed topology control with multipath routing is denoted by ST-M(Steiner Tree approach-Multipath routing). In each scenario, we measure the number of active nodes (N_R) and links (L_R), then calculate the ratio of active nodes to total number of node (N_R/N) and the ratio of active links to total number of link (L_R/L). Also, we measure the energy consumption on the topology (E_R) based on three different energy consumption models (EMs) of nodes and links, shown in TABLE I [6]. Then we compute energy saving ratio as

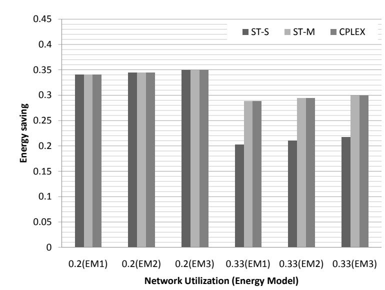
$$1 - \frac{E_R}{E} = 1 - \frac{\sum\limits_{(i,j)\in E} x_{ij} P_L + \sum\limits_{i \in V} y_i P_N}{L P_L + N P_N}, \tag{2}$$

where x is the energy model index.

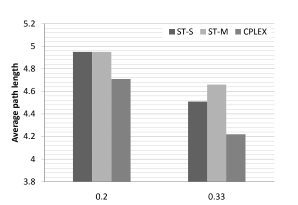
We also measure the average path length and the average link utilization after turning off nodes and links for saving energy. These metrics give an idea of the cost of the energy savings. Higher path lengths and higher link utilizations can lead to longer delays for delivering traffic.

A. Comparison between optimum solution and the hueristics on grid topologies

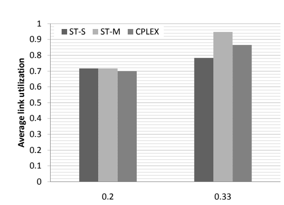
In this section, we evaluate the performance of optimum and the proposed ST-S and ST-M on an n-by-n topology with n^2 core nodes and $4n$ edge nodes. Some core nodes have



(a) Energy saving on a 4-by-4 grid topology

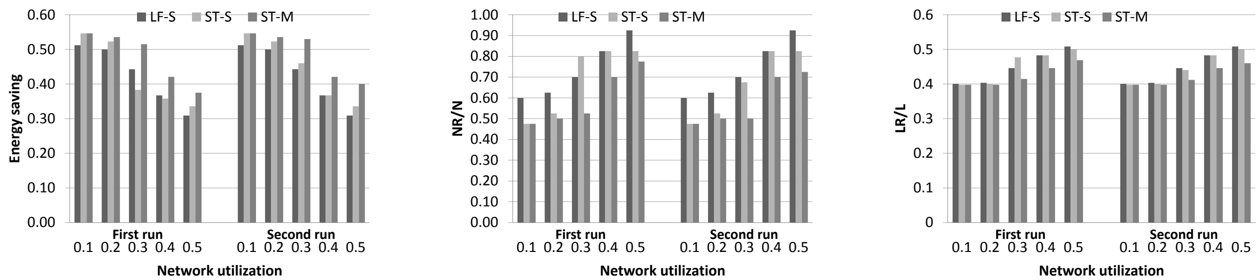


(b) Average path length on a 4-by-4 grid topology



(c) Average link utilization on a 4-by-4 grid topology

Fig. 2. Comparison between optimum and the hueristics



(a) Energy savings on a random topology (b) N_R/N on a random topology (c) L_R/L on a random topology

Fig. 3. Energy savings, N_R/N, and L_R/L on a random topology

TABLE II
N_R/N AND L_R/L OF ST-S, ST-M, AND CPLEX ON A 4-BY-4 TOPOLOGY

Network utilization	Measured metrics	ST-S	ST-M	CPLEX
0.2	N_R/N	0.75	0.75	0.75
	L_R/L	0.48	0.48	0.48
0.33	N_R/N	0.94	0.81	0.81
	L_R/L	0.53	0.52	0.52

connections to only core nodes, but other core nodes have connections to core nodes and edge nodes. Edge nodes have two or three connections to the closest edge nodes. The set of od-pairs is all the possible combinations of edge nodes. All od-pairs generate an identical amount of traffic ($t^{sd} = 1$). Link capacity is calculated corresponding to the desired average network utilization by the shortest path routing on the original topology.

We use CPLEX to find the optimal solution. The optimal solution is denoted as ‘CPLEX’.

Figure 2 shows energy savings, path length and link utilization of ST-S, ST-M, and CPLEX on a 4-by-4 grid topology. Table II shows N_R/N and L_R/L of ST-S, ST-M, and CPLEX.

In this simulation, ST-M improves energy savings, matching CPLEX, while ST-S has lower energy savings than CPLEX. We can find that the higher savings is a result of smaller number of live nodes (N_R/N) in Table II. However, the different choices of active nodes and links in ST-M and ST-S from CPLEX results in higher path length and higher link utilization than CPLEX. Our focus here is on reducing the energy consumption. Re-configuring link weights of active links would be one solution to mitigate the extra path lengths. We leave this problem as a future work.

We show the simulation results of CPLEX only on a 4-by-4 topology, because CPLEX is hard to compute. For these experiments, the simulation time of CPLEX took a few hours. For larger topologies, the running time of CPLEX may exceed the dynamics of timescales of traffic fluctuations and hence it is not practical.

B. Comparison between the existing scheme and the proposed schemes on realistic topologies

In this section, we consider realistic topologies: random topologies [3], Abilene and Sprint topology [14]. Here, we consider EM1 when we compute energy consumption based on the active nodes and links.

We compare our approaches ST-S and ST-M to earlier work in [3]. We denote the earlier scheme as LF-S. Figure 3 shows simulation results of LF-S, ST-S, and ST-M at 0.1-0.5 network utilization. Left bars in each graph are the first run of the proposed topology control.

In Figure 3(a), energy saving of LF-S, ST-S, and ST-M are shown. At low average network utilization, energy saving gap between ST-S and ST-M is small. At 0.3 and higher average network utilization, multipath routing substantially improves energy consumption, ranging from 8 to 17% of base energy consumption. Multipath routing achieves significant energy savings by using fewer nodes in the network.

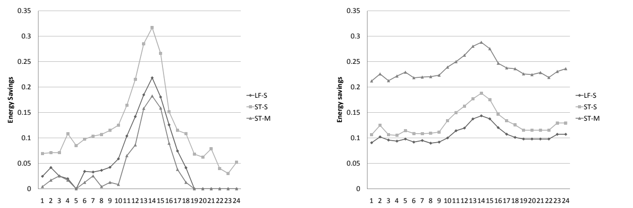
The fraction of live nodes (N_R/N) and live links (L_R/L) are shown in Figure 3(b) and 3(c). The gap of N_R/N between ST-S and ST-M is 0 - 0.35. The gap of L_R/L between ST-S and ST-M is 0 - 0.04, smaller than the gap of N_R/N between ST-S and ST-M. The dominant factor for the extra energy savings in multipath routing comes form the reduced number of nodes.

N_R/N of ST-S increases from 0.58 to 0.85 when average network utilization changes from 0.2 to 0.3, but N_R/N of ST-M increases by only 0.03. This result shows that multipath routing uses the increased network capacity more effectively than single path routing.

LF-S shows better energy saving than ST-S, when the average network utilization is 0.3 and 0.4. In order to meet the traffic demand of an od-pair, a number of internal nodes may have to be added and as a result the network capacity grows in bursts. During this capacity growth bursts, our topology control may require more tuning.

To improve energy savings, we iteratively run our algorithm on the reduced topology obtained from the first run. Adding a highly utilized node on the currently-used topology is expected to turn off more nodes in the topology control than adding a highly utilized node on the original topology.

The simulation results of the second run are shown in right bars in each graph of Fig. 3. With the second run of topology control, ST-S shows better energy saving than LF-S at 0.3 and 0.4 avearage utiliztion because N_R/N of ST-S is reduced at 0.3 and 0.4 avearge utiliztion, compared to the first run of ST-S. Some more iterations of the proposed topology control can



(a) Energy savings on Abilene network (hourly)

(b) Energy savings on Sprint network (hourly)

Fig. 4. Energy savings on real networks

achieve more energy savings.

Abilene and Sprint network [15] are employed to evaluate the performance of the proposed scheme. Since the topology information is POP-level information, all nodes have traffic to send or to receive. We randomly select a set of nodes that do not generate or receive traffic in this simulation to simulate the impact of powering off node first on energy savings. We randomly choose 1/3 nodes in Abilene and 1/5 nodes in the Sprint network that can be removed without the loss of the connectivity. Traffic demand of Abilene is obtained from [16]. Traffic demand for Sprint is generated by the gravity model [17] and daily pattern of the average network utilization from [16]. We configure the topology where the nodes and links are minimized at every 5 minutes. The proposed schemes are able to configure these topologies with heuristics, while using linear programming to obtain these configurations is not feasible for real deployment in real topologies.

Fig. 4(a) and 4(b) show the energy savings in the Abilene and Sprint networks over 24 hours. Each point in the plot is the average of energy savings of 5 minute bins over an hour. The energy savings varies over time because the number of active nodes and links are changed according to the amount of traffic demand. When traffic demand is high (1-9 hour and 19-24 hour) the energy savings of ST-S is slightly higher than LF-S. When traffic demand is low (10-18 hour), the difference between ST-S and LF-S is increased. Using multipath routing enhances the energy savings. ST-M shows 5-14% more energy savings than LF-S in Abilene network, and 12-14% more energy savings than LF-S in Sprint network.

VIII. Conclusion

In this paper, we studied saving network energy through topology control and multipath routing. We proposed a topology control algorithm that built the necessary network connecting all the traffic sources and sinks based on a Steiner tree approach. We proposed a multipath routing algorithm based on bin packing to meet the traffic demands with minimal network resources. We simulated the proposed algorithms in different networks with different load and capacity constraints. We showed that the proposed topology control resulted in better energy savings in more scenarios than a previous topology control algorithm with single path routing. Also, our topology control has slightly less compleixy than the existing scheme due to less number of iteration when single shortest path routing is used.

When combined with multipath routing, our approach resulted in significantly better energy savings, by up to 17% of the base energy consumption, than previous approaches.

In the future, we will study the impact of our approach in data center networks. We will study the combination of our proposed scheme with hardware sleeping or rate-adaptation schemes that achieve energy savings on short time scales.

References

[1] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, C. Diot "Packet-Level Traffic Measurements from the Sprint IP Backbone" *IEEE Network*, 2003.

[2] M. Gupta and S. Singh "Dynamic Ethernet Link Shutdown for Power Conservation on Ethernet Links" *ICC*,2007

[3] L. Chiaraviglio, M. Mellia, and F. Neri "Reducing power consumption in backbone networks" *ICC*, 2009.

[4] N. Vasic and D. Kostic "Energy-aware traffic engineering" *EPFL Technical report*, 2008

[5] W. Fisher, M. Suchara, and J. Rexford "Greening Backbone Networks: Reducing Energy Consumption by Shutting Off Cables in Bundled Links" *Green Networking Workshop in SIGCOMM*, 2010.

[6] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. MacKeown "ElasticTrees: Saving energy in data center networks" *USENIX NSDI*, 2010.

[7] Sumitha Bhandarkar, A. L. Narasimha Reddy, Mark Allman and Ethan Blanton, Improving the robustness of TCP to Non-Congestion Events , IETF RFC 4653, August 2006.

[8] L. Kou, G. Markowsky, and L. Berman "A fast algorithm for Steiner trees" *Acta Inform., 12(2):141-151*, 1981

[9] S. Even, A. Itai, and A. Shamir "On the complexity of Time Table and Multi-Commodity Flow Problem" *16th Annual Symposium on Foundations of computer science, page 184-193*, October 1975

[10] M.E.J. Mark "Networks: an introduction" *Oxford University Press*, 2010

[11] D. Xu , M. Chiang , J. Rexford "DEFT: Distributed exponentially weighted flow splitting" *INFOCOM*, 2007

[12] Y. Lee and A.L.N. Reddy "Constructing disjoint paths for failure recovery and multipath routing" *Elsevier Journal on Computer Networks*, 2012

[13] U. Brandes "A faster algorithms for betweenness centrality" *Journal of Mathematical Sociology*, 2001

[14] Mingui Zhang, Cheng Yi, Bin Liu, Beichuan Zhang "GreenTE: Power-Aware Traffic Engineering" *International Conference on Network Protocols (ICNP)* , 2010

[15] Rocketfuel topology mapping. *WWW http://www.cs.washington.edu*

[16] Yin Zhangs Abilene TM. [Online]. Available: http://www.cs.utexas. edu/ yzhang/research/AbileneTM/

[17] S. Kandula, D. Katabi, B. Davie, and A. Charny, Walking the Tightrope: Responsive Yet Stable Traffic Engineering, *Proceedings of ACM SIGCOMM*, 2005