Performance Analysis of Parameters Affecting Power Efficiency in Networks

Antar Shaddad Abdul-Qawy
School of Computer and Information Sciences
University of Hyderabad, Hyderabad, India
Email: eng.antar2007@gmail.com

Anupama Potluri
School of Computer and Information Sciences
University of Hyderabad, Hyderabad, India
Email: apcs@uohyd.ernet.in

***Abstract*—Networking devices in ISP networks and data centers have been deployed in an over-provisioning and redundant manner to meet the worst case traffic loads and to quickly recover from failures. These devices are idle or semi-idle most of the time with full power consumption. Today, there is widespread interest in techniques that help in reducing energy waste and achieving high levels of energy-efficiency. Solutions proposed in literature are either topology-oriented (ESTOP) or traffic-oriented (ESTA). In this paper, we propose a Hybrid Heuristic for Green Networking (HHGN) that exploits the parameters considered in both ESTOP and ESTA to identify a subset of network elements that can be switched off such that the power saving is maximized under network connectivity and edge bandwidth utilization constraints. We introduce two new heuristics, w-BFLP (weighted Betweenness, Flow, Links, Power) and w-BFP (weighted Betweenness, Flow, Power), that sort the nodes and the edges respectively in increasing order of their importance in the topology. Nodes and edges are then switched off from the least important until the connectivity threshold or maximum bandwidth utilization constraint on the edges as specified by the user is reached. We compare our approach with ESTOP and ESTA for edge optimization in terms of power gain, mean utilization of edge bandwidth, percentage of sleeping edges, fairness index, trade-off between power gain and edge utilization, and increase in the length of re-computed paths for real ISP and FatTree topologies using different power models. Experimental results show that HHGN gives the best performance independent of power models, traffic matrices and topologies tested.**

I. Introduction

The rapid growth of cloud computing with data centers and increase in Internet usage have led to a tremendous increase in power consumption by networking infrastructure. Traditionally, the development of networking systems has been focused on performance improvements; but now, the energy consumption of network elements has started to limit further performance growth. In this context, the goal of networking and communication systems' design has shifted to energy-efficiency. Green networking strategies are classified, at a high level, as individual device level and network level solutions [1]. At the network level, the traffic flow orders are regrouped on a subset of network elements during off-peak hours while maintainig a desired level of network connectivity or maximum edge utilization. Network level solutions are divided into two categories: Topology-oriented solutions [1]–[3] and Traffic-oriented solutions [4]–[6].

The topology-oriented solutions depend on the topological information available in the routing protocols such as network connectivity and betweenness, which is the number of the all-pairs shortest paths of the topology that cross the network elements. They are usually applied under connectivity-related constraints, such as network connectivity threshold, to ensure some level of redundancy. In [1], a modification of OSPF is proposed in which a set of routers, called *exporters*, send their shortest path tree (SPT) to their neighbor routers called *importers*. The importer routers modify their SPTs by using the SPTs of associated exporters and then switch off the other edges which are not included in the modified SPTs. In [3], the authors propose an algorithm called Energy Saving based on Algebraic CONnectivity (ESACON) which uses the algebraic connectivity to decide which subset of edges should be switched off. The set of edges that are switched off is called the sleeping list. This solution is improved in [2] with an algorithm called Energy Saving based on TOPology control (ESTOP) which uses the edge betweenness and algebraic connectivity to identify the sleeping list. The element will be switched off only if the connectivity of the reduced network is above a given threshold.

The traffic-oriented solutions require instantaneous measurements of the network traffic loads as a main parameter of switching off process. They are usually applied under a number of traffic-related constraints, such as maximum edge utilization threshold, to guarantee specific traffic requirements. In [6], the authors propose a solution used in IP/MPLS networks in which power off/on of an edge depends on the available capacity of the edge after rerouting the LSPs. In [4], an algorithm called Energy Saving Traffic Aware algorithm (ESTA) is proposed which uses different heuristics to sort network nodes and edges before switching them off. These heuristics are: least-flow (LF), most-power (MP), least-link (LL) and random (R). The element will be switched off only if resultant utilization of all edges in the network is below a given threshold. In [5], an algorithm to save energy in data centers called ElasticTree is provided which depends only on port counters rather than a complete traffic flow matrix. ESTA and ElasticTree provide an ILP formulation of the power saving problem and show that it can be used only for trivial cases and it is not an efficient solution for large networks.

Topology-oriented solutions are traffic-blind and focus only on the network connectivity. They do not provide any guarantee for traffic-related and quality of service constraints. On the other hand, traffic-oriented solutions consider only the current traffic patterns. They do not consider the topology and

978-1-4244-8953-4/11/$26.00 © 2015 IEEE

the importance of specific nodes and edges in that topology in terms of their participation in the shortest paths between source-destination pairs. To our knowledge, no scheme has been proposed to take advantage of both the topology and the traffic information to make the decision for switching off a subset of network elements. In this paper, we propose a hybrid approach – Hybrid Heuristic for Green Networking (HHGN) – which considers both factors to identify a maximum number of nodes and edges that can be put to sleep. We try to ensure that the power saving is maximized and edge utilization does not exceed a given threshold while maintaining the network connectivity above a predefined level. The contributions of this paper may be stated as follows:

1) Proposed a hybrid energy-efficiency solution that considers both topology and traffic parameters.
2) Extended the power model to include random and uniform power models.
3) Extensive experimentation comparing ESTOP, ESTA-LF, ESTA-MP and HHGN for edge optimization on different topologies including FatTree, using multiple power models.

The rest of the paper is organized as follows: Section II provides a more detailed explanation of the hybrid approach. Section III describes experimental setup and evaluation methodology. Experimental results and analysis are given in Section IV. Finally, we conclude the paper in Secion V.

II. Hybrid Heuristic for Green Networking (HHGN)

As shown in Figure 1, the block diagram of the proposed energy saving system consists of three main logical modules: HHGN optimizer, routing and power control modules. The power control module controls the power state of the network elements depending on the inputs coming from HHGN optimizer output. The routing module is to compute all-pairs shortest paths of the topology.

HHGN considers both the topology and traffic parameters to determine the importance of different nodes and edges in the topology. These parameters are: the betweenness, total traffic flowing through a node/edge, degree of nodes, and power consumption. We assign weights to each of these parameters and sort the nodes/edges according to increasing order of their importance. For example, an edge with a high betweenness or most traffic flow is not a good candidate for switching off. We call such an edge as an important edge. We start switching off nodes/edges from the least important element onwards until the connectivity and bandwidth utilization constraints are satisfied. By using weights for different parameters, a network administrator can fine tune the network based on the SLAs to be satisfied. HHGN can be run during off-peak hours in ISP backbones or enterprise networks to reduce power consumption during low load conditions. In addition, it can be implemented to control power consumption in software-defined data center (DC) networks as in B4, Google's inter-DC software-defined WAN [7] which reallocates flows to keep bandwidth utilization high. We present the notations used in our algorithm in the next section. Following that, we present our algorithm, specifically, the edge optimization algorithm in detail.

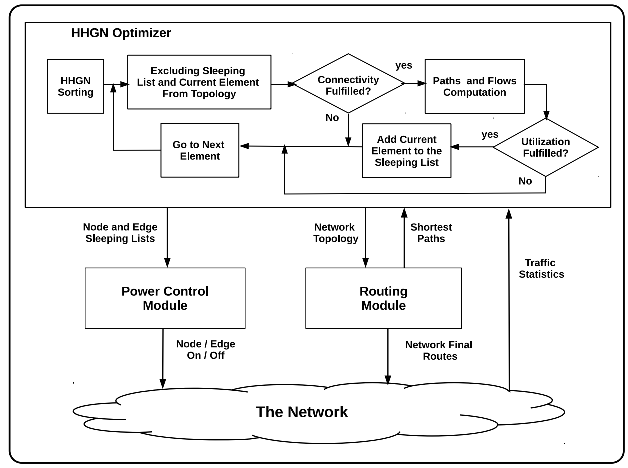


Fig. 1: HHGN Block-diagram

A. Notations

HHGN considers the network topology as an undirected graph $G(V, E)$, where V is the set of nodes in the network and E is the set of edges. $N = |V|$ and $M = |E|$ represent the number of nodes and the number of edges in the network respectively. The traffic flow matrix, TFM(G), is an $N \times N$ matrix that represents the traffic flow orders (TFODs) between every pair of source and destination nodes belonging to the network. Each TFOD in TFM(G) represents an aggregation of all the traffic flows between the source and destination nodes. Table I describes the notations and quantities that are used in HHGN.

TABLE I: Notations used in HHGN

Noatation	Description
$P(G)$	power matrix of the network graph
$\mathcal{N}_S$, $\mathcal{L}_S$	list of sleeping nodes and edges respectively
$\acute{G}$	reduced network graph after deleting $\mathcal{N}_S$ and $\mathcal{L}_S$
$\mathcal{A}_G$, $\mathcal{A}_{\acute{G}}$	algebraic connectivity of the initial and reduced network graphs respectively
$\mathcal{C}\%_{\acute{G}}$	percentage connectivity of the reduced network graph with respect to the initial network graph
$\mathcal{C}_{th}$	network connectivity threshold
$SP(s, d)$	shortest path between nodes s and d
SPs	set of $SP(s, d)$s for all (s, d) pairs
$\mathcal{N}_B(v)$, $\mathcal{L}_B(l)$	betweenness of the node v and edge l respectively
$\mathcal{N}_F(v)$, $\mathcal{L}_F(l)$	total aggregated traffic flow that crosses the node v and edge l respectively
$\mathcal{N}_P(v)$, $\mathcal{L}_P(l)$	amount of power consumed by the node v and edge l respectively
$\mathcal{L}_C(l)$	given capacity of the edge l
$\mathcal{N}_L(v)$	number of edges incident on node v
$\mathcal{U}\%_{max}$	maximum edge utilization threshold
α, β, γ	weights of $\mathcal{L}_B(l)$, $\mathcal{L}_F(l)$, $\mathcal{L}_P(l)$ respectively which are used in w-BFP
σ, η, ω and ζ	weights of $\mathcal{N}_B(v)$, $\mathcal{N}_F(v)$, $\mathcal{N}_L(v)$ and $\mathcal{N}_P(v)$ respectively which are used in w-BFLP
$\mathcal{N}_H(v)$, $\mathcal{L}_H(l)$	hybrid factors of node v and edge l respectively which are used in w-BFLP and w-BFP

To measure the network connectivity, we use algebraic connectivity ($\mathcal{A}_G$) which is defined as the second eigenvalue, λ_2, of laplacian matrix, $L(G)$, of the network graph. It is a lower bound on the node connectivity which is the minimum number of nodes that can be removed from the graph before it becomes disconnected. λ_2 becomes zero when the graph is disconnected [2], [8]. $\mathcal{C}_{th}$ represents a redundancy factor to guarantee that connectivity is preserved above a given threshold in the modified topology, i.e,

$$\mathcal{C}\%_{\acute{G}} = \frac{\mathcal{A}_{\acute{G}}}{\mathcal{A}_G} \geq \ \mathcal{C}_{th} \tag{1}$$

$\mathcal{U}\%_{max}$ represents an overprovisioning factor to guarantee that for each edge l with a given capacity $\mathcal{L}_C(l)$, the total traffic flows, $\mathcal{L}_F(l)$, on that edge should not exceed a maximum value, i.e,

$$\mathcal{L}_F(l) \ \leq \ \mathcal{L}_C(l) \ \times \ \mathcal{U}\%_{max} \qquad \forall \ \ l \ \in \ E \tag{2}$$

B. HHGN Algorithm

HHGN optimizer consists of two main parts: node optimization and edge optimization which are used to switch off nodes and edges respectively. We can run node optimizer only, edge optimizer only, or both at the same time. When a node is selected to be turned off, all its incident edges will be turned off. Therefore, when we run both optimizers, we start with the node optimizer to achieve a higher power savings by switching off a node and all its incident edges as done in [4]. Then, we run the edge optimizer to switch off individual edges of other nodes that could not be switched off. Switching off all edges of a node implies that all line cards of a switch/router have been powered off. However, in standard routers, there is also a central processor that is not switched off. Secondly, if we can switch off a node without affecting the bandwidth utilization and connectivity constraints, then, many edges will be switched off in one iteration leading to a better time complexity for the algorithm. Therefore, first determining which nodes can be powered off and then the edges leads to better power efficiency as well as time complexity. Every time HHGN tries to switch off a node or an edge, it checks two parameters: $\mathcal{C}\%_{\acute{G}}$ and $\mathcal{L}_F(l)$ against $\mathcal{C}_{th}$ and $\mathcal{U}\%_{max}$ respectively. The switching off process succeeds only if both Equations 1 and 2 are fulfilled. Otherwise, the element will still be active. Edge and node optimizers use w-BFP (weighted Betweenness, Flows and Power) and w-BFLP (weighted Betweenness, Flows, Links and Power) to sort edges and nodes respectively according to the following hybridization formulas, where α, β etc. are weights assigned to the respective parameters:

$$\mathcal{L}_H(l) = \alpha \times \mathcal{L}_B(l) + \beta \times \mathcal{L}_F(l) + \gamma \times \frac{1}{\mathcal{L}_P(l)} \tag{3}$$

$$\mathcal{N}_H(v) = \sigma \times \mathcal{N}_B(v) + \eta \times \mathcal{N}_F(v) + \omega \times \mathcal{N}_L(v) + \zeta \times \frac{1}{\mathcal{N}_P(v)} \tag{4}$$

To achieve the hybridization, the values assigned to the weights in these equations should range between 0 and 1 and the sum of them must be 1. These weights can be considered as control knobs to tune the importance of each of the parameters to obtain the desired energy saving with a minimum impact on the mean edge utilization and network connectivity. This allows the administrator to determine whether the static parameters such as betweenness and power are given more weightage than the current traffic conditions or vice versa. It also allows us to understand the importance of dynamic traffic conditions in the saving of energy in networks. In the edge optimization equation 3, if we set α to 1 and β and γ to zero, it reduces to ESTOP heuristic [2]. If we set β (γ) to 1 and the others to zero, it reduces to ESTA-LF (ESTA-MP) [4] respectively. Similarly, for node optimization equation 4, if we set one of the weights σ, η, ω and ζ to 1 and the others to zero, it reduces to ESTOP, ESTA-LF, ESTA-LL [4] and ESTA-MP respectively.

Algorithm 1 Edge_Optimization

Input: $G(V, E)$, TFM(G), P(G), W(G), $\mathcal{C}_{th}$, $\mathcal{U}\%_{max}$.

1: $\mathcal{L} \longleftarrow$ w-BFP(G, TFM(G), P(G), W(G)
2: $\mathcal{L}_S \ \longleftarrow \ \phi$
3: **for** $\ i = 1 \ \rightarrow \ M$ **do**
4: $\quad \acute{E} \ = \ E \ - \ \mathcal{L}_S \ - \ \mathcal{L}[i]$
5: $\quad \acute{G} \ = \ G(V, \acute{E})$
6: $\quad \mathcal{C}\%_{\acute{G}} \ = \ \frac{\mathcal{A}(\acute{G})}{\mathcal{A}(G)} \ \times \ 100$
7: $\quad$ **if** ($\mathcal{C}\%_{\acute{G}} \ < \ \mathcal{C}_{th}$) **then Continue**
8: $\quad SPs \ = \ \phi$
9: $\quad$ **for** each (s,d) pair $\in \ V$ **where** TFM[s][d] $\neq 0$ **do**
10: $\quad\quad SP(s, d)$ = Shortest_path($\acute{G}, \ s, \ d$)
11: $\quad\quad SPs = SPs \ \cup \ SP(s, d)$
12: $\quad$ **end for**
13: $\quad$ stop = FALSE
14: $\quad$ **for** each edge l $\in \ SPs$ **do**
15: $\quad\quad \mathcal{L}_F(l)$ = Edge_util($l, \ SPs$, TFM(G))
16: $\quad\quad$ **if** ($\mathcal{L}_F(l) > \ \mathcal{L}_C(l) \ \times \ \mathcal{U}\%_{max}$) **then**
17: $\quad\quad\quad$ stop = TRUE
18: $\quad\quad\quad$ **break**
19: $\quad\quad$ **end if**
20: $\quad$ **end for**
21: $\quad$ **if** (stop $\neq$ TRUE) **then**
22: $\quad\quad \mathcal{L}_S \ = \ \mathcal{L}_S \ \cup \ \mathcal{L}[i]$
23: $\quad$ **end if**
24: **end for**
25: $\acute{G} \ = \ G(V, \ E - \mathcal{L}_S)$

Output: $\acute{G}$, $\mathcal{L}_S$.

The edge optimization phase is described in Algorithm 1. It starts by calling w-BFP heuristic to sort network edges before switching off process (line 1). The algorithm that switches off edges (lines 3-24) iteratively tries to turn off a new edge from the set of the sorted edges $\mathcal{L}$ and put it into the sleeping list $\mathcal{L}_S$. At each iteration, it removes the current edge $\mathcal{L}[i]$ from the graph (line 4). It then computes $\mathcal{A}_{\acute{G}}$ of the reduced graph $\acute{G} = G(V, \ E - \mathcal{L}_S - \mathcal{L}[i])$. If connectivity threshold is not satisfied, the edge remains active and the algorithm skips to the next iteration (lines 6-7). If connectivity threshold is fine, the algorithm checks that the maximum utilization threshold is satisfied once this edge is removed (lines 8-20). It does this by computing all SPs of the reduced graph $\acute{G}$ (lines 8-12) and the new utilization for each edge in the shortest paths and checks it against $\mathcal{U}\%_{max}$ (lines 13-20). If this is fine for all edges in SPs, the edge is definitely powered off (lines 21-23). Otherwise, it is still active and the algorithm skips to the next iteration. The node optimization phase is similar to the edge optimization, except that it switches off the nodes and their incident edges. To compute edge/node betweenness, $\mathcal{L}_B(l)$ and $\mathcal{N}_B(v)$, HHGN uses Dijkstra to find the shortest path between every node pair in the network $(s, d) \in V$ where $s \neq d$. The betweenness of every edge or node included in that path is incremented. It adds the TFOD between the pair of nodes of that path to $\mathcal{N}_F(v)$ and $\mathcal{L}_F(l)$, the total node and edge flow respectively. To compute network connectivity, we use the algebraic connectivity, $\mathcal{A}_{\acute{G}}$, of the network graph as described in Section II-A. Total flow of edges is recomputed every time an element is switched off by using Dijkstra and re-calculating the paths for the flows. Algebraic connectivity is also computed each time. These are done to ensure that the edge utilization/network connectivity satisfy the thresholds set for these parameters. In the worst case, this is done N times for node optimization and M times for edge optimization. Since Dijkstra takes $O(N^2 logN)$ and algebraic connectivity takes $O(N^2)$, in node optimization the complexity is $O(N^3 logN)$ while the edge optimization complexity is $O(MN^2 logN)$.

III. Experimental Setup and Evaluation Methodology

To evaluate HHGN, we wrote a custom C program in Linux. We run the experiments on a platform running Ubuntu 13.04 (Kernel ver. Linux 3.8.0-19), and equipped with Intel Core2 Duo 3.00 GHz CPU and 4 GB of RAM. The input to the program is in the form of a text file including information of nodes and edges of the topology, their power consumption, and the capacity of the edges. The output file provides the evaluation parameters described in Section III-E. We evaluated HHGN *for edge optimization only*. We experimented with 10 different sets of weight values given to betweenness (α), total flow (β), and power (γ). The aim is to identify a set of α, β and γ, which enable us to achieve a high level of power gain, $\mathcal{P}_G\%$, with a minimum increase in edge mean utilization, $\mathcal{U}\%$. We compare HHGN performance with that of ESTOP, ESTA-LF and ESTA-MP. In all cases we set maximum edge utilization threshold, $\mathcal{U}\%_{max}$, to 80% and connectivity threshold, $\mathcal{C}_{th}$, is in the range 10% to 90%. The following subsections describe more about evaluation setup.

A. Network Topology Description

To thoroughly assess HHGN performance, we consider different sets of network topologies: sparsely connected topologies, medium degree topologies and data center topologies. We mainly use real ISP topologies provided by SND-lib [9] and other topologies such as typical topologies used in data center networks [10].

- The sparse set is composed of three real ISP topologies [9]: Abilene: with 12 nodes and 15 edges, France: with 25 nodes and 45 edges and Germany50: with 50 nodes and 88 edges.
- The medium set is composed of two real ISP topologies [9]: Dfn-Gwin: with 11 nodes and 47 edges and Di-Yuan: with 11 nodes and 42 edges.
- The data center topology is a FatTree with 45 nodes and 108 edges [10].

Due to lack of space, we present results of only Germany50, Dfn-Gwin and FatTree. The results for Abilene and France are similar to Germany50 and Di-Yuan has similar results to Dfn-Gwin.

B. Traffic Flow Model

We use real-life traffic matrices provided by SND-lib as well as synthetic traffic matrices that we have generated as per the models proposed in [2], [4]. We generate 40 synthetic traffic matrices and average results over these 41 traffic matrices. For synthetic traffic matrices, we generate two flows for every pair of routers: one flow has a high rate of traffic randomly selected from [20, 100] Mbps and the other flow has a low rate of traffic randomly selected from [1, 20] Mbps. A TFOD is given as: TFOD $= 0.5 \times R[1, 20] + 0.5 \times R[20, 100] Mbps$

These TFODs constitute the traffic flow matrix.

C. Edge Capacity Model

For edge capacity assignment, we first define a traffic matrix as explained in Section III-B and then assign edge capacities to match the traffic demands as specified in [2]. This guarantees that the capacity constraints are met in the initial state. The procedure is as follows: first, we route all TFODs according to shortest path routing. Second, the total traffic flow of edges is computed. Finally, modules of capacity 2.5Gbps (C_m) are assigned to each edge such that the edge utilization is less than or equal to the overprovisioning factor of 25% (C_{ov}) as in [2]. The capacity $\mathcal{L}_C(l)$ of an edge l that has a total traffic flow $\mathcal{L}_F(l)$ is given as follows:

$$\mathcal{L}_C(l) = max\left(\left\lceil \frac{\mathcal{L}_F(l)}{\mathcal{C}_m \times \mathcal{C}_{ov}} \right\rceil, 1\right) \times \mathcal{C}_m \qquad Gbps$$

D. Edge Power Model

To define the power consumption of edges for the topologies, we used three different power models: power model used in ESTOP [2], random power model that assumes different linecards in each of the routers and a uniform power model that assumes same linecards in all routers. In [11], the authors define a generalized power model for routers which takes into consideration the power consumption of the chassis, the number of line cards that are active, a scaling factor corresponding to traffic utilization and the line card in a base configuration. Since we are not dealing with node optimization here but only edge optimization, we used the power consumption values for different NICs of Cisco and Juniper routers [12], [13].

1) ESTOP Power Model: This model is used in [2]. The power consumption of an edge depends on the number of capacity modules assigned to the edge, which in turn, depends on the amount of total traffic flow. The power $\mathcal{L}_P(l)$ of an edge l that has a total capacity $\mathcal{L}_C(l)$ is given as follows:

$$\mathcal{L}_P(l) = \frac{\mathcal{L}_C(l)}{\mathcal{C}_m} \times\ 140 \qquad watts$$

2) Random Power Model: To adopt more practical scenarios, we assume heterogeneous systems in which different network interface cards (NICs) are used. We consider different types of NICs given in [12], [13] and we randomly select one of these values, RP_{NIC}, and assign it to both ends of the edge. The power $\mathcal{L}_P(l)$ of an edge l is given as follows:

$$\mathcal{L}_P(l) = 2 \times\ RP_{NIC} \qquad watts$$

3) Uniform Power Model: In this model, we use uniform power assignments. The scenario assumes homogeneous systems in which all NICs of the topology are of the same type and hence consume the same amount of power, UP_{NIC}. The power $\mathcal{L}_P(l)$ of an edge l is given as follows:

$$\mathcal{L}_P(l) = 2 \times UP_{NIC} \qquad watts$$

E. Evaluation Parameters

In this work, we used the following parameters defined in [2] to evaluate HHGN:

- edge power gain($\mathcal{P}_G\%$): the percentage of power that can be saved.
- mean utilization($\mathcal{U}\%$): the mean edge utilization of the reduced network.
- sleeping edges($\mathcal{L}_S\%$): the percentage of edges that can be switched off.
- fairness index($\mathcal{J}$): the traffic distribution fairness on all edges of the reduced network.
- increase in the length of paths(δ): measures the length of paths (number of hops) after running the heuristic.

We also propose a new parameter, fitness function $\mathcal{F}_{pu}$, to measure the trade-off between maximizing power gain, $\mathcal{P}_G\%$, and minimizing the increase in edge mean utilization, $\mathcal{U}\%$. $\mathcal{F}_{pu}$ is defined as follows:

$$\mathcal{F}_{pu} \;=\; \frac{\mathcal{P}_G\%}{\mathcal{U}\%} \;\times\; 100$$

The higher $\mathcal{F}_{pu}$ value implies a better tradeoff between $\mathcal{P}_G\%$ and $\mathcal{U}\%$.

IV. Experimental Results and Analysis

In this section, we present the evaluation results and analysis of the proposed scheme. We present tradeoff function, $\mathcal{F}_{pu}$, of HHGN when the weights given to betweenness (α), total flow on the edges (β) and the power of the edges (γ) equal to (0.1, 0.1, 0.8), (0.1, 0.2, 0.7) and (0.2, 0.1, 0.7) which give a better tradeoff than the other combinations which we tried. For the other evaluation parameters: power gain ($\mathcal{P}_G\%$), mean utilization ($\mathcal{U}\%$), sleeping edges ($\mathcal{L}_S\%$), fairness index ($\mathcal{J}$), and increase in the length of paths (δ), we present the results when α, β and γ equal to (0.1, 0.1, 0.8) for sparse and medium degree topologies and (0.1, 0.2, 0.7) for FatTree topology.

A. Sparsely Connected Topologies

Table II reports $\mathcal{F}_{pu}$ of different heuristics using the random power model for Germany50. Figures 2 and 3 show $\mathcal{P}_G\%$, $\mathcal{U}\%$, $\mathcal{L}_S\%$, and $\mathcal{J}$ of the four heuristics with all the power models.

In the random and ESTOP power models, we observe that ESTA-MP, on occasion, does not give a high power gain even though it switches off most power-consuming edges first. This is due to the fact that these edges are also those with a maximum amount of traffic flow, especially in ESTOP power model. Switching off such edges increases maximum utilization of the other edges beyond the configured threshold and so it is unable to switch off many edges. HHGN, on the other hand, does better than ESTA-MP despite the fact that we are giving a weight of 0.8 for the power of edges. This shows that hybridization does help in improving tradeoff and also allow more edges to be switched off by their reordering due to betweenness and least flow. We observe that HHGN performs better with the random power model. This is because the edge power consumption does not depend on the edge flow unlike in ESTOP power model in which the edge with the most power also has the highest amount of traffic.

TABLE II: $\mathcal{F}_{pu}$ of ESTOP, ESTA-LF, ESTA-MP and HHGN with the three best combinations of weights and different values of $\mathcal{C}_{th}$ for Germany50 with the random power model.

Heuristic \ $\mathcal{C}_{th}$		10%	20%	30%	40%	50%	60%	70%	80%	90%	Average $\mathcal{F}_{pu}$
ESTOP		115	115	115	118	115	113	107	95	88	109
ESTA-LF		117	117	116	115	116	113	112	93	80	109
ESTA-MP		112	112	112	113	116	106	85	67	86	101
HHGN	(0.1, 0.1, 0.8)	123	124	125	125	126	127	119	125	104	122
	(0.1, 0.2, 0.7)	122	123	123	127	129	127	122	126	109	123
	(0.2, 0.1, 0.7)	121	121	122	122	122	126	119	125	108	121

ESTOP and ESTA-LF have a higher $\mathcal{L}_S\%$ and less $\mathcal{U}\%$ than ESTA-MP and HHGN with the random and ESTOP power models. They have a worse performance in terms of $\mathcal{P}_G\%$ because they do not consider edge power in the sorting heuristic. Hence, they have less $\mathcal{F}_{pu}$ than HHGN. From the experiments, we found that the least betweenness edges also have the least amount of traffic flows and least effects on network connectivity, while the highest betweenness edges have more traffic flows and the highest effects on connectivity. Therefore, ESTOP and ESTA-LF perform similarly with all three power models.

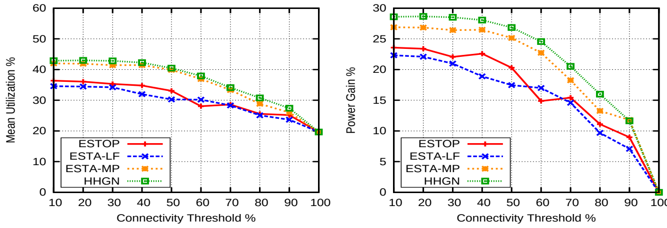
In the uniform power model, all the edges have the same power consumption. Therefore γ does not have any effect. However, $\mathcal{P}_G\%$ depends on the number of sleeping edges while $\mathcal{U}\%$ depends on the flow of these edges. Therefore, the performance of ESTA-MP is the worst, whereas the performance of HHGN in this power model is comparable to that of ESTOP and ESTA-LF. This is because it depends on the edge betweenness and the edge flow with the same value of α and β. $\mathcal{L}_S\%$ of HHGN is comparable to that of ESTOP and ESTA-LF and higher than that of ESTA-MP.

We also observe that $\mathcal{J}$ of HHGN is almost similar to that of ESTOP and ESTA-LF with the random and uniform power models. This is because the redirected flows are fewer as implied from the less $\mathcal{U}\%$ resulting from these two power models. While with ESTOP power model, it is almost as worse as that of ESTA-MP. This is because the redirected flows, which are high, are unfairly redistributed to the active edges.

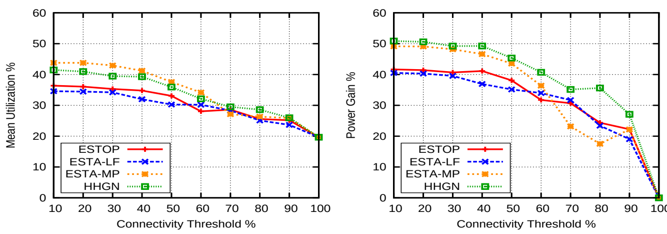
Figure 4 shows the maximum increase in the length of recomputed paths (δ) of the different heuristics with the random power model for Germany50. It can be seen that HHGN, on the average, has a comparable δ to the other heuristics. In ESTOP power model, it has less δ than the other heuristics; while in the uniform power model, it is comparable to ESTOP and ESTA-LF, which are better than ESTA-MP.

B. Medium Degree Topologies

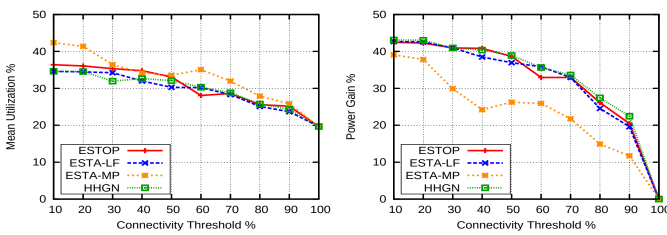
In these topologies, the number of edges that can be switched off is higher for the same given connectivity thresh-



(a) ESTOP power model

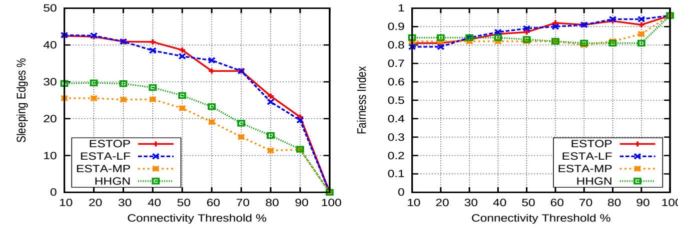


(b) Random power model

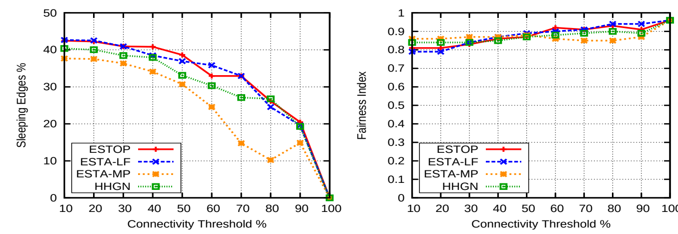


(c) Uniform power model

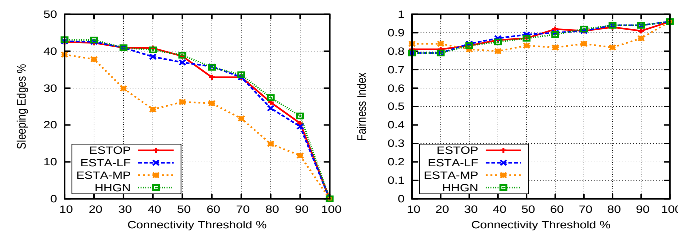
Fig. 2: Power gain, $\mathcal{P}_G\%$, and mean utilization, $\mathcal{U}\%$, of ESTOP, ESTA-LF, ESTA-MP and HHGN when α, β and γ = (0.1, 0.1, 0.8) for Germany50.



(a) ESTOP power model



(b) Random power model



(c) Uniform power model

Fig. 3: Sleeping edges, $\mathcal{L}_S\%$, and fairness index, $\mathcal{J}$, of ESTOP, ESTA-LF, ESTA-MP and HHGN when α, β and γ = (0.1, 0.1, 0.8) for Germany50.

old. Table III shows $\mathcal{F}_{pu}$ of the different heuristics with the random power model for Dfn-gwin. From the experimental results, we find that all the heuristics have similar performance for ESTOP and uniform power models. This is due to the fact that most of the edges have similar amounts of traffic flowing through them. In ESTOP power model, this implies that they are assigned the same capacity. This results in a uniform edge power when using ESTOP power model. Moreover, we observed that most of edges have the same betweenness. Thus, all factors being the same, all heuristics have similar performance for these topologies for all evaluation parameters considered for these power models. For the random power model, the power gain of ESTA-MP and HHGN is better than ESTA-LF and ESTOP (Fig. 5) as the number of edges switched off is same in all the heuristics but, HHGN and ESTA-MP switch off those edges that consume more power leading to a higher power gain. We also observe that the sleeping edges and fairness index of HHGN and ESTA-MP are similar to ESTA-LF and ESTOP (Fig. 5). The increase in average length of paths (Fig. 6) is higher for HHGN and ESTA-MP only for 10% connectivity threshold whereas they are mostly similar in other cases. Thus, we can conclude that HHGN and ESTA-MP are better for medium degree topologies for the random power model.

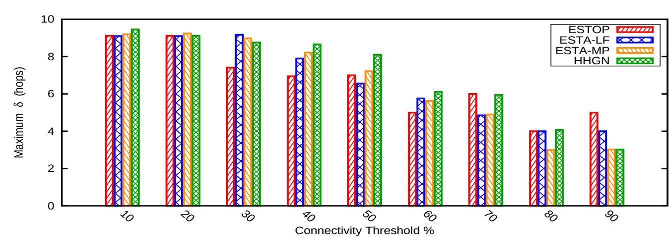


Fig. 4: Maximum increase in the length of paths, δ, of ESTOP, ESTA-LF, ESTA-MP and HHGN when α, β and γ = (0.1, 0.1, 0.8) for Germany50 with the random power model.

TABLE III: $\mathcal{F}_{pu}$ of ESTOP, ESTA-LF, ESTA-MP and HHGN with the three best combinations of weights and different values of $\mathcal{C}_{th}$ for Dfn-Gwin with the random power model.

Heuristic \ $\mathcal{C}_{th}$		10%	20%	30%	40%	50%	60%	70%	80%	90%	Average $\mathcal{F}_{pu}$
ESTOP		230	250	289	357	385	397	396	434	449	354
ESTA-LF		231	264	304	347	403	417	443	463	473	372
ESTA-MP		239	306	366	467	459	453	573	595	638	455
HHGN	(0.1, 0.1, 0.8)	232	308	357	390	455	481	557	574	622	442
HHGN	(0.1, 0.2, 0.7)	233	298	359	393	450	483	518	564	594	432
HHGN	(0.2, 0.1, 0.7)	231	288	362	390	453	495	505	558	596	431

C. FatTree Topology

In FatTree topology, the communications are either between servers within the data center or between the servers

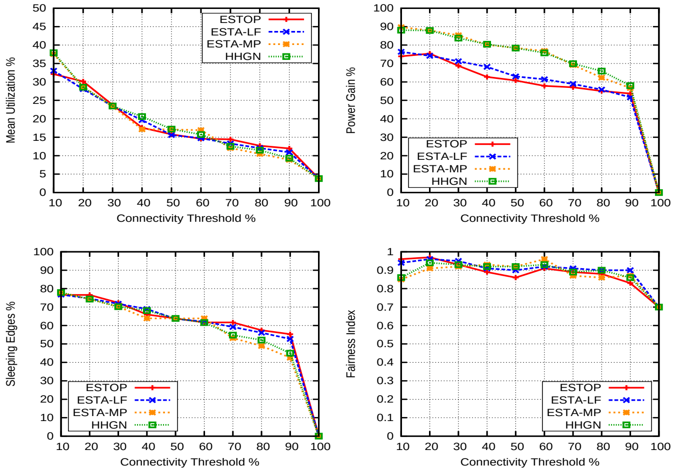


Fig. 5: Power gain, $\mathcal{P}_G\%$, mean utilization, $\mathcal{U}\%$, sleeping edges, $\mathcal{L}_S\%$, and fairness index, $\mathcal{J}$,of ESTOP, ESTA-LF, ESTA-MP and HHGN when α, β and γ = (0.1, 0.1, 0.8) for Dfn-Gwin with the random power model.

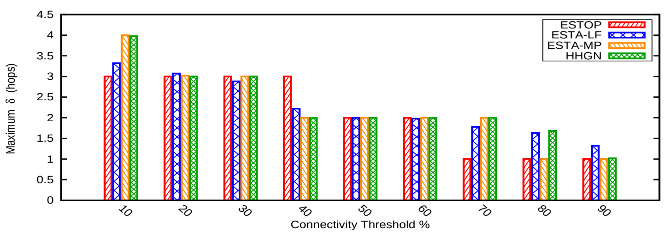


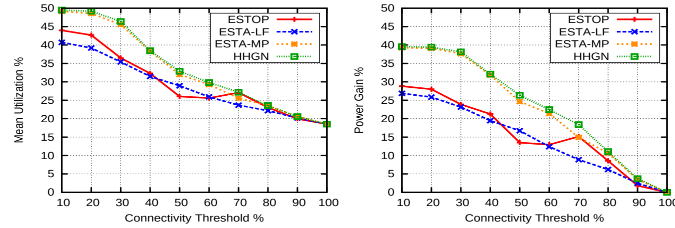
Fig. 6: Maximum increase in the length of path, δ, of ESTOP, ESTA-LF, ESTA-MP and HHGN when α, β and γ = (0.1, 0.1, 0.8) for Dfn-Gwin with the random power model.

and outside users. Therefore, TFODs generated as specified in Section III-B, are only from edge switches to edge switches and from edge switches to core switches and vice versa. Table IV shows $\mathcal{F}_{pu}$ of different heuristics using the random power model. Figures 7 and 8 show $\mathcal{P}_G\%$, $\mathcal{U}\%$, $\mathcal{L}_S\%$, and $\mathcal{J}$ with all the power models.

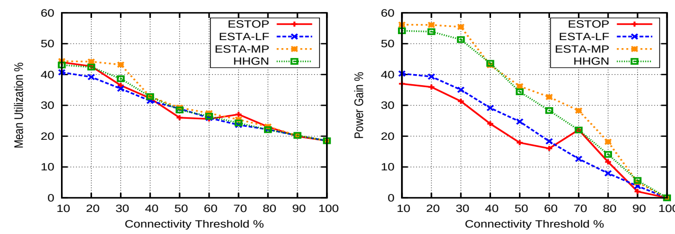
TABLE IV: $\mathcal{F}_{pu}$ of ESTOP, ESTA-LF, ESTA-MP and HHGN with the three best combinations of weights and different values of $\mathcal{C}_{th}$ for FatTree topology with the random power model.

Heuristic \ $\mathcal{C}_{th}$		10%	20%	30%	40%	50%	60%	70%	80%	90%	Average $\mathcal{F}_{pu}$
ESTOP		84	84	86	74	69	63	82	51	10	67
ESTA-LF		99	100	99	93	85	71	53	36	19	73
ESTA-MP		127	127	128	131	124	119	111	79	25	108
HHGN	(0.1, 0.1, 0.8)	125	126	131	133	119	108	96	69	31	104
	((0.1, 0.2, 0.7)	126	127	133	133	121	107	91	64	28	103
	(0.2, 0.1, 0.7)	118	118	122	123	107	92	89	69	31	97

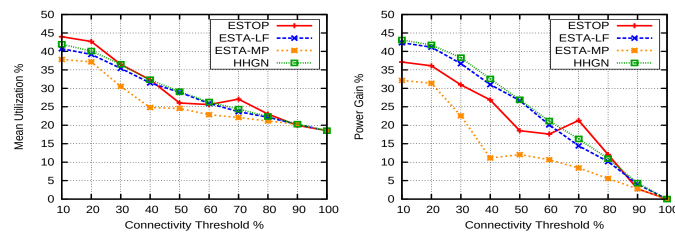
From the results, we can see that HHGN has better performance than the other heuristics with ESTOP and uniform power models. In fact, with the uniform power model, it has the highest power gain, most number of sleeping edges and highest fairness index whereas ESTA-MP has the worst performance



(a) ESTOP power model



(b) Random power model



(c) Uniform power model

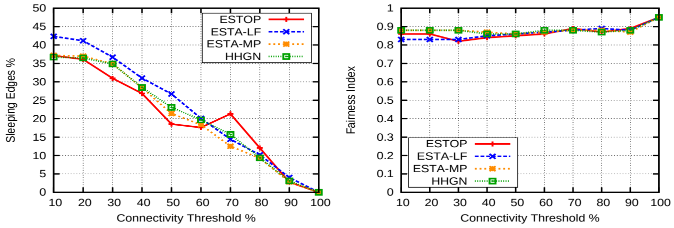
Fig. 7: Power gain, $\mathcal{P}_G\%$, and mean utilization, $\mathcal{U}\%$, of ESTOP, ESTA-LF, ESTA-MP and HHGN when α, β and γ = (0.1, 0.2, 0.7) for FatTree topology.

on all parameters. This shows that in data centers using FatTree topologies and uniform interconnection between switches (i.e., uniform NICs), ESTA-MP must not be used and HHGN is the best heuristic. With the random power model, HHGN performance is better than that of ESTOP and ESTA-LF and less than that of ESTA-MP. The better performance of ESTA-MP in the random power model, in which the power edge does not depend on the traffic flow, is due to its ability to switch off a number of edges comparable to that of the other heuristics as we can see in Fig 8. Since these edges are the ones with more power, its power gain is highest.

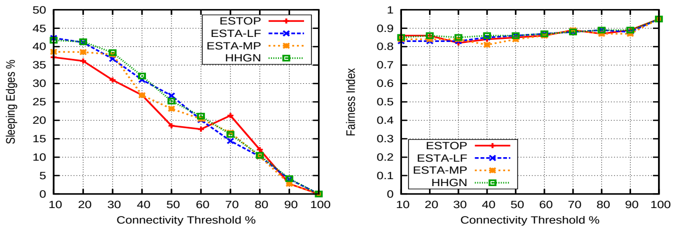
Figure 9 shows the maximum increase in the length of paths (δ) in terms of number of hops with the random power model. It can be seen that HHGN in most of the cases is comparable to other heuristics. ESTOP has a higher increase when $\mathcal{C}_{th}$ ranges between 10% and 30%. The same is true for the ESTOP power model, whereas in the uniform power model, ESTA-MP has less δ than other heuristics. This can be explained by the fact that it switches off the least number of edges. All the other heuristics have similar increase in length of paths for the uniform power model.

V. Conclusion

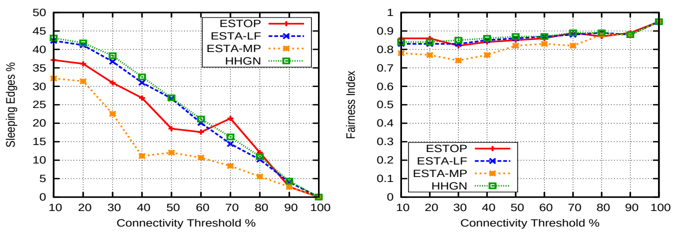
In this paper, we proposed a hybrid solution, HHGN, to exploit both topology and traffic parameters to identify a subset



(a) ESTOP power model



(b) Random power model



(c) Uniform power model

Fig. 8: Sleeping edges, $\mathcal{L}_S\%$, and fairness index, $\mathcal{J}$, of ESTOP, ESTA-LF, ESTA-MP and HHGN when α, β and γ = (0.1, 0.2, 0.7) for FatTree topology.

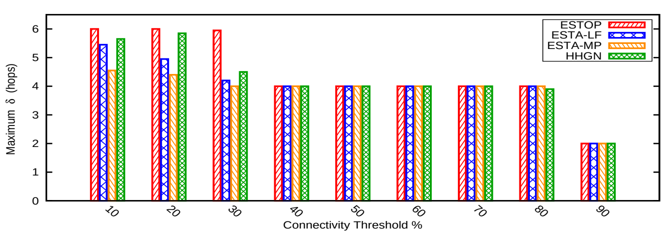


Fig. 9: Maximum increase in the length of path, δ, of ESTOP, ESTA-LF, ESTA-MP and HHGN when α, β and γ = (0.1, 0.2, 0.7) for FatTree topology with the random power model.

of nodes and edges that can be switched off under connectivity and maximum edge utilization constraints. In general, we find that ESTA-MP has a better performance than ESTA-LF and ESTOP for random and ESTOP power models. It has the worst performance for uniform power model whereas ESTA-LF and ESTOP are good with this model. For sparse topologies, we find that HHGN gives the best tradeoff between power gain and mean utilization compared to all other heuristics for all power models considered. HHGN switches off more edges than ESTA-MP which has equivalent power gain for these topologies while mean utlization is smaller than for ESTA-MP. This shows that it is far better than ESTA-MP for such topologies. For FatTree topologies, commonly seen in data center networks, HHGN has the highest power gain, more number of sleeping edges and high fairness index and similar mean utilization as ESTOP and ESTA-LF for uniform power model making it the heuristic of choice for such networks using similar interconnection NICs. HHGN and ESTA-MP are both suitable for FatTree topologies with random power model, i.e., data center networks which use heterogeneous NICs. We find that the results are similar when we consider only the single real life traffic matrix available for these topologies. HHGN has as good a performance as ESTA-MP with random and ESTOP power models and as good as ESTOP and ESTA-LF for uniform power model. This proves that hybridization leads to an overall improved performance for different types of topologies with different power models.

In our future work, we intend to evaluate for node optimization only and using both node and edge optimization. We plan to evaluate the schemes using higher degree ISP topologies and other data center topologies such as VL2, B-Cube, and large FatTree topologies. We also plan to modify HHGN to switch on nodes and edges based on traffic load conditions and fault tolerance requirements.

REFERENCES

[1] A. Cianfrani, V. Eramo, M. Listanti, M. Marazza, and E. Vittorini, "An energy saving routing algorithm for a green OSPF protocol," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2010, pp. 1–5.

[2] F. Cuomo, A. Cianfrani, M. Polverini, and D. Mangione, "Network pruning for energy saving in the internet," *Computer Networks*, vol. 56, no. 10, 2012.

[3] F. Cuomo, A. Abbagnale, A. Cianfrani, and M. Polverini, "Keeping the connectivity and saving the energy in the internet," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2011, pp. 319–324.

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

[5] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, "Elastictree: saving energy in data center networks," in *Proceedings of the 7th USENIX conference on Networked systems design and implementation*, 2010, pp. 17–17.

[6] A. Coiro, F. Iervini, and M. Listanti, "Distributed and adaptive interface switch off for internet energy saving," in *Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*, 2011, pp. 1–8.

[7] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat, "B4: experience with a globally-deployed software defined wan," in *ACM SIGCOMM 2013 Conference*, 2013, pp. 3–14.

[8] M. Fiedler, "Algebraic connectivity of graphs," in *Czech. Math. J. 23*, 1973, p. 298305.

[9] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, "SNDlib 1.0–Survivable Network Design Library," in *Proceedings of the 3rd International Network Optimization Conference (INOC 2007), Spa, Belgium*, April 2007.

[10] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 63–74, Aug. 2008.

[11] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power awareness in network design and routing," in *INFOCOM 2008*, April 2008, pp. 1130–1138.

[12] http://www.cisco.com/c/en/us/products/routers/product-listing.html.

[13] http://www.juniper.net/in/en/products-services/routing/.