A General Framework For Performance Guaranteed Green Data Center Networking

Ting Wang*, Yu Xia*, Jogesh Muppala*, Mounir Hamdi*,†, Sebti Foufou‡
*Hong Kong University of Science and Technology, Hong Kong
†Hamad Bin Khalifa University, ‡Qatar University
*{twangah, rainsia, muppala, hamdi}@cse.ust.hk, ‡sfoufou@qu.edu.qa

***Abstract*—From the perspective of resource allocation and routing, this paper aims to save as much energy as possible in data center networks. We present a general framework, based on the blocking island paradigm, to try to maximize the network power conservation and minimize sacrifices of network performance and reliability. The bandwidth allocation mechanism together with power-aware routing algorithm achieve a bandwidth guaranteed tighter network. Besides, our fast efficient heuristics for allocating bandwidth enable the system to scale to large sized data centers. The evaluation result shows that up to more than 50% power savings are feasible while guaranteeing network performance and reliability.**

I. Introduction

The data center network connecting the servers in the data center plays a crucial role in orchestrating the data center to deliver peak performance to the users. In order to meet the high performance and reliability requirements, the data center network is usually constructed with a large numbers of network devices and links to achieve 1:1 oversubscription for peak workload and traffic bursts. However, the traffic rarely reaches the peak capacity of the network in practice [1][10]. The average link utilization ranges between 5% and 25% with wide diurnal variations [3]. Besides, the traditional non-traffic-aware routing algorithms can also cause havoc in network utilization which worsens the situation. These critical issues result in a great waste of energy consumed by the idle or under-utilized devices. What is worse, today's commodity network devices are not energy proportional (an idle switch consumes up to 90% of the peak power consumption [14]), mainly because the components of the network devices (such as transceivers, line cards, fans, etc) are always kept on regardless of whether they have data packets to transfer or not, leading to a significant energy wastage. In fact at most of time the traffic can be satisfied just by a subset of network devices and links, and the remaining ones can be put into sleep mode or powered off for the sake of saving energy [3][10][18].

Based on the above observations the main aim of this paper is to design strategies to make the data center network energy efficient, where the amount of power consumed by the network is proportional to the actual traffic workload. Noticing that the large-scale data centers are usually bandwidth hungry, but the network bandwidth is a scarce resource [6] and can have an impact on the network performance, therefore our approach is designed to perform an intelligent resource (i.e. bandwidth) allocation and energy-aware routing in a richly-connected data center. However, the energy-aware routing problem is NP-hard, which is proved in this paper. Nevertheless we achieve our goal efficiently using the Blocking Island resource abstraction technique together with well-designed heuristic algorithms. The final goal is to power off as many idle and low-utilized network devices as possible without compromising the overall performance and reliability of the entire network.

The primary contributions of this paper can be summarized as follows:

1) We formulate the energy-aware routing problem as a MCF problem with the proof of its NP-hardness and provide an efficient heuristic solution.
2) To the best of our knowledge, we are the first to apply the Blocking Island Paradigm for resource allocation into data center networks to achieve power conservation.
3) We design the Power-efficient Network System with the target of maximizing power savings and minimizing sacrifices of network performance and fault tolerance.
4) We conduct pertinent simulations to evaluate the performance of Power-efficient Network System under various network conditions and reliability requirements.

The rest of the paper is organized as follows. First we review the related research literature in Section II. Then we formulate the energy optimization problem in Section III. Afterwards, Section IV introduces our power-aware heuristic scheme followed by the evaluation in Section V. Finally, Section VI concludes this paper.

II. Related Work

The significant benefits accrued from making a data center energy-efficient has aroused considerable research interest in recent years as summarized in this section. Existing research in this area can be generally divided into five categories: (1) To use renewable or green sources of energy, like wind, water, solar energy, heat pumps, and so on. However, this requires many additional factors that must be taken into consideration including the location of the data center, climate or weather conditions, geography, and so forth. Apart from these limitations, the power generation and delivery infrastructure also yield high capital cost. The related works in this field includes GreenHadoop [8], GreenStar Network (GSN) Testbed [16], and Net-Zero Energy Data Centers [2]. (2) To design more

978-1-4799-3512-3/14/$31.00 ©2014 IEEE

energy-efficient hardware, for example, by exploiting the advantages of Dynamic Voltage and Frequency Scaling (DVFS) techniques, and vary-on/vary-off (VOVO) techniques. Currently many hardware-based proposals have been put forward, among which the typical representatives include Powernap [15], Thread Motion [17], PCPG (per-core power gating) [11], and Memory Power Management via DVFS [5]. (3) To design novel energy-efficient data center architectures to achieve power conservation. Examples like flattened butterfly topology [1], NovaCube [21], CamCube [4] - 3D Torus based switchless interconnection without switches/racks and associated cooling costs, Nano Data Centers which can avoid the huge cooling costs because Nano DCs are freely cooled by ambient air [20], Proteus [19], can be classified into this category. These alternative approaches are attractive, but still need to be further evaluated in real data centers. (4) To design energy-aware routing algorithms to consolidate traffic flows to a subset of the network and power off the idle devices. The proposed schemes like Elastic Tree [10], Energy-aware Routing Model [18], Merge Networks [3] are some typical representatives of this approach. (5) To reduce the energy consumption by drawing support from techniques like the Virtual Machine migration and placement optimization, for example GreenCloud [12].

III. Problem Formulation

A. MCF Problem Description

The multi-commodity flow (MCF) problem is a network flow problem, which aims to find a feasible assignment solution for a set of flow demands between different source and destination nodes. The MCF problem can be expressed as a linear programming problem by satisfying a series of constraints: capacity constraints, flow conservation, and demand satisfaction. This problem occurs in many contexts where multiple commodities (e.g. flow demands) share the same resources, such as transportation problems, bandwidth allocation problems, and flow scheduling problems. In the next subsection, we show that the energy-aware routing problem can also be formulated as an MCF problem.

B. Problem Formulation

To describe the bandwidth allocation problem in a data center network $G = (V, E)$, we define the constraints as follows: Demand completion—each traffic demand specified as a tuple (i, j, d_{ij}) should be satisfied with the required bandwidth simultaneously, with i, j, d_{ij} $(i, j \in V)$ as the source node, destination node and bandwidth request, respectively (i.e., Constraint (1)); Reliability requirement—each demand should be assigned FT backup routes (i.e., Constraint (2)); Capacity constraint—each link $k \in E$ has a bandwidth capacity C_k and none of the traffic demands ever exceed the link capacities (i.e., Constraint (3)); Flow conservation (i.e., Constraint (4)).

The objective is to find a set of optimal routing paths that minimizes the power consumption of the switches and ports involved, satisfying the above constraints. Hereby, the parameter Ω_s denotes the power consumed by the fixed overheads (like fans, linecards, and tranceivers, etc) in a switch, Ω_p represents the power consumption of a port, and α serves as a safety margin ($\alpha \in (0, 1)$ with 0.9 as default). The binary variables S_i and L_k represent whether the switch i and the link k are chosen or not (equal to 1 if chosen), $x_{ij}^{(k)}$ denotes the flow value of the demand d_{ij} that the link k carries from i to j, $R(d_{ij})$ means the number of available paths for demand d_{ij}, N_i consists of all links adjacent to the switch i, and N_i^+ (N_i^-) includes all links in N_i and carrying the flow into (out of) the switch i. Then, the MCF problem can be modeled in the following form:

$$\text{Minimize} \qquad \Omega_s \sum_{i \in V} S_i + 2\Omega_p \sum_{k \in E} L_k$$

Subject to:

$$\forall i, j \in V, \quad \sum_{k \in N_i} x_{ji}^{(k)} \geq d_{ji}, \quad \sum_{k \in N_i} x_{ij}^{(k)} \geq d_{ij}, \tag{1}$$

$$\forall i, j \in V, \quad R(d_{ij}) \geq FT, \tag{2}$$

$$\forall k \in E, \quad \sum_{i \in V} \sum_{j \in V} x_{ij}^{(k)} \leq \alpha C_k, \tag{3}$$

$$\forall i, j \in V, \quad \sum_{k \in N_i^+} x_{ij}^{(k)} = \sum_{k \in N_i^-} x_{ij}^{(k)}, \tag{4}$$

$$\forall k \in E, \quad L_k \geq \frac{1}{C_k} \sum_{i \in V} \sum_{j \in V} x_{ij}^{(k)}, \quad L_k \in \{0, 1\}, \tag{5}$$

$$\forall i \in V, \; S_i \geq \frac{1}{\sum_{k \in N_i} C_k} \sum_{i \in V} \sum_{j \in V} \sum_{k \in N_i} x_{ij}^{(k)}, S_i \in \{0, 1\}, \tag{6}$$

$$\forall i, j \in V, \quad \forall k \in E, \quad x_{ij}^{(k)} \geq 0 \tag{7}$$

Note that if we assume the optimal rounting paths are link-disjoint, we can simplify Constraint (2) as $\forall i, j \in V, \sum_{k \in N_i} Y_{ji}^{(k)} \geq FT, \sum_{k \in N_i} Y_{ij}^{(k)} \geq FT$ with $Y_{ji}^{(k)} \geq x_{ij}^{(k)}/C_k$ and $Y_{ji}^{(k)} \in \{0, 1\}$.

C. NP-Hardness

For the MCF problem described above, we change to its corresponding decision problem (DMCF): Is there any set of routing paths such that satisfy $\Omega_s \sum_{i \in V} S_i + 2\Omega_p \sum_{k \in E} L_k \leq N$, and all constrains in MCF. To prove the DMCF problem is NP-hard, we show the classical 0-1 knapsack problem can be reduced to a DMCF instance. Thus, both DMCF and MCF are NP-hard due to the equivalence of hardness.

The formal definition of the 0-1 knapsack problem is given as below. There are n kinds of items $I_1, I_2, ..., I_n$, where each item I_i has a nonnegative weight W_i and a nonnegative value V_i, and a bag with the maximum capacity as C. The 0-1 knapsack problem determines whether there exists a subset of items S ($S \subseteq [n]$) such that $\sum_{i \in S} W_i \leq C$ and $\sum_{i \in S} V_i \geq P$.

Proof. Reduction: We first construct a specific instance G of the DMCF problem. Suppose there exists a source s and a sink t in G, and only one demand $(s, t, d_{st} = P)$. For each item I_i in the knapsack problem, we build a path p_i with W_i links from s to t, and each link k in p_i has capacity of $C_k = V_i/\alpha$. The parameters are set as $\Omega_p = 1$, $\Omega_s = 0$, $FT = 1$, and the predefined threshold of DMCF is set as $N = 2C$.

(i) The solution for the 0-1 knapsack problem exists $\Rightarrow$ The solution for the specific DMCF instance exists. Suppose there exists a subset of items S such that $\sum_{i \in S} W_i \leq C$ and

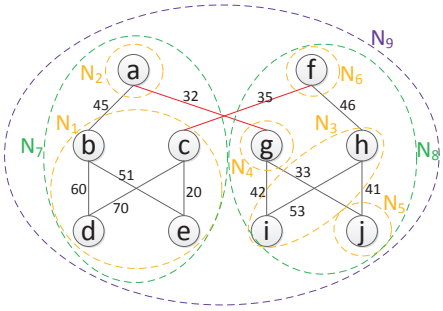

Fig. 1. An example of Blocking Island Graph, in which N_1-N_6 are 50-BIs, N_7-N_8 are 40-BIs, and N_9 is 30-BI. The red lines are critical links between two 40-BIs. The weights on the links are their available bandwidth.

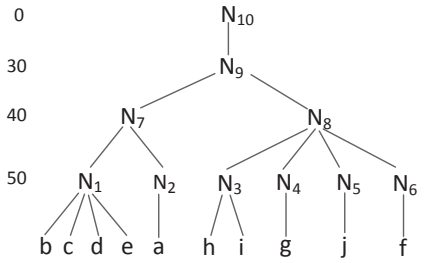

Fig. 2. Abstraction BIH tree for BIG in Fig.1, and N_{10} indicates 0-BI.

$\sum_{i\in S} V_i \geq P$. Then, we can use S to construct a solution for the specific DMCF instance. For each item I_i ($i \in S$), we choose the corresponding path p_i in G, and assign a flow of size V_i to this path, i.e., $x_{st}^{(k)} = V_i$ for all links in p_i. Thus, the capacity constraint (3) holds since $x_{st}^{(k)} = V_i \geq \alpha C_k = V_i$, the flow conservation (4) holds naturally, and then the demand completion (1) is satisfied since $\sum_{k\in N_t} x_{st}^{(k)} = \sum_{k\in N_s} x_{st}^{(k)} = \sum_{i\in S} V_i \geq P = d_{st}$, and hence the reliability requirement (2) is met due to $FT = 1$. Constraint (5) means we will choose all W_i links in the path p_i, and then the total number of chosen links is $\sum_{i\in S} W_i$, leading to the value of the objective function $2\Omega_p \sum_{k\in E} L_k = 2\sum_{i\in S} W_i \leq 2C = N$. Therefore, the found solution is indeed a solution for the specific DMCF instance.

(ii) The solution for the specific DMCF instance exists $\Rightarrow$ The solution for the 0-1 knapsack problem exists. Suppose there exists a set of S_i's and L_k's satisfying all constraint in the specific DMCF instance and $2\Omega_p \sum_{k\in E} L_k \leq N$. If a link k ($k \in N_t$) in the path p_i has $L_k > 0$, then $x_{st}^{(k)} > 0$ by Constraint (5) and $x_{st}^{(k)} \leq \alpha C_i = V_i$ by Constraint (3). For such a p_i, we choose the corresponding item i in the 0-1 knapsack problem and form a subset of item S. Then, $\sum_{i\in S} V_i \geq \sum_{k\in N_t} x_{st}^{(k)} \geq d_{st} = P$ due to Constraint (1). On the other hand, since $x_{st}^{(k)} > 0$ ($k \in N_t$) in p_i, the flow values of all links in p_i is equal to $x_{st}^{(k)} > 0$ due to the flow conservation. This means all W_i links in p_i have $L_k = 1$ by Constraints (5). Then, the total number of chosen links is $\sum_{i\in S} W_i = \sum_{k\in E} L_k \leq N/2\Omega_p = C$. Thus, we find the solution for the 0-1 knapsack problem. That ends the proof. □

IV. Power-aware Heuristic Scheme

The routing algorithms in data centers desire higher requirements and should be traffic aware with low computation complexity. However, the traditional routing algorithms, like shortest path routing or its variations, suffer high computation complexity and are not traffic aware. These kind of algorithms may also lead to poor link utilization and even congestion. Based on BI Paradigm, we propose an energy-aware bandwidth guaranteed routing scheme, which behaves efficiently with low computation complexity by reducing search space and also achieves good link utilization. This framework can be generally divided into two phases. The first phase is to apply the bandwidth allocation mechanism to select the most appropriate unallocated demands from the traffic matrix. Then the second phase uses power-aware routing algorithm to compute the best route set for the selected unallocated demands and allocate with their required bandwidth.

A. Blocking Island Paradigm

Derived from Artificial Intelligence, Blocking Island (BI) [7] provides an efficient way to represent the availability of network resources (especially bandwidth) at different levels of abstraction. It is defined as: *A β-Blocking Island for a node x is the set of all nodes of the network that can be reached from x using links with at least β available resources, including x* [7]. BI has some important fundamental properties that are very useful for bandwidth-aware routing. (1)Unicity: Each node has one unique β-BI. (2)Route Existence: An unallocated demand $d_u = (x, y, \beta_u)$ can be satisfied with at least one route if and only if both the endpoints x and y are in the same β_u-BI. (3)Inclusion: If β_i is larger than β_j, then the β_i-BI for a node is a subset of β_j-BI for the same node.

β-BI can be obtained by a simple greedy algorithm whose complexity is linear in O(L), where L is the number of links. The obtained BIs can be used to construct the Blocking Island Graph (BIG) as shown in Fig.1. We can further construct BIH tree (see Fig.2), which is a recursive decomposition of BIGs in decreasing order of demand.

BI significantly reduces the routing search space with bandwidth guarantee. The *Route Existence* property enables much faster decisions about whether a request can be satisfied just by checking whether both the endpoints are in the same β-BI, while the traditional routing algorithms have to compute the routes before deciding the route's existence.

B. Bandwidth Allocation Mechanism

As a common issue for the constraint satisfaction problem (CSP) or MCF problem, for some traffic matrices it cannot be guaranteed to find an assignment to satisfy all the flows all the time, which is also mentioned in [10]. How to select next demand from the traffic matrix to allocate has a great impact on the future allocation success ratio, and also impacts the computation and search efficiency. As indicated in [9], the common way for this kind of problem is fail-first principle based technique which tries those tests in the given set of tests that are most likely to fail. There are also some static

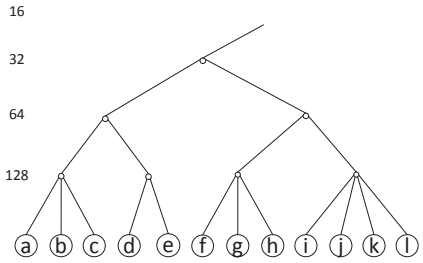

Fig. 3. The BIH example used for DSLH and DSFN

Comparison between DSLH and DSFN				
Demands	d_1	d_2	d_3	d_4
DSLH	32	64	128	64
DSFN	12	5	4	7

Fig. 4. The example for comparison between DSLH and DSFN

techniques, such as to first select the demand with the greatest value. However, they are not suitable in the data center network which requires more efficient mechanism in a dynamic way. In virtue of BIH abstraction tree's advantages, we can achieve this by proposing two dynamic heuristic demand selection approaches: DSLH and DSFN.

1. DSLH Heuristic: It intends to first choose the demand where the lowest common father (LCF) of the demand's endpoints is highest. The basic idea behind this heuristic is to first allocate the demands that use resources on more critical links, which follows the fail-first principle. The higher the LCF is, the more constrained the demand is.

2. DSFN Heuristic: It is derived from the perspective of limited network resources. Assume the LCF of the demand's endpoints is a β_d-BI, then we first choose the demand whose LCF's β_d-BI contains fewest nodes (LCF-FN). The intuition behind DSFN is that the fewer the nodes in the domain, the fewer the feasible routes between the endpoints of the demand. Hence, this heuristic also follows the fail-first principle.

Here an example is given to illustrate the principles of DSLH and DSFN. Assume that the current BIH of the network is as shown in Fig.3, and there are four demands needed to be allocated: d_1 = *(d, h, 16)*, d_2 = *(c, e, 32)*, d_3 = *(i, k, 32)*, d_4 = *(g, l, 16)*. Clearly, the LCF/LCF-FN of demand d_1, d_2, d_3, d_4 are 32-BI/12 nodes, 64-BI/5 nodes, 128-BI/4 nodes, and 64-BI/7 nodes, respectively, as shown in Fig.4. If the DSLH heuristic is applied here, the demand d_1 will be allocated first because its LCF is highest. Whereas the DSFN will select d_3 since its LCF-FN is the fewest.

Based on the above two heuristics, we propose the bandwidth allocation mechanism (BAM) which combines DSLH with DSFN, and it works as follows:

1. Firstly, the DSLH heuristic is preferentially applied to choose the next best demand to assign, since DSLH exhibits a better performance in our experiments.

2. In case there are multiple demands that have the same value for the DSLH heuristic, then the DSFN heuristic is applied. If we use the same example as shown in Fig.3 and Fig.4, but we now have only three demands d_2, d_3, d_4 to allocate. After step 1, both d_2 and d_4 are selected according to the principle of DSLH heuristic. Then DSFN is applied to select d_2 whose LCF has fewer nodes as the output.

3. If there are still more than one best demands, then the demand with the highest bandwidth requirement is preferred. This criterion follows the fail-first principle, where the more bandwidth allocation may more likely cause BI splittings and thus hinder any future allocation of other demands.

4. Finally, we randomly select one demand from the output of step 3.

BAM not only significantly increases the success ratio of bandwidth allocation satisfying all flows of the traffic matrix simultaneously, but also increases the search efficiency which in turn decreases the computation complexity.

C. Power-aware Routing Algorithm

The routing algorithm needs to assign a best route for each selected demand request using BAM. However, the domain is too large and the valid route set is too time-consuming to be computed. In order to improve the search efficiency and further select the best route as fast as possible, we need to generate the routes in the most advantageous order. The route selection criterion should be in accordance with the following rules: (1) Rule 1: The route should use as few critical links as possible. This rule not only aims to decrease the failure ratio of allocations, but also to reduce the computation cost of updating BIs. (2) Rule 2: The demands should be aggregated to the greatest extent. By means of merging traffic, we can achieve a much tighter network which can allow us to conserve more energy. (3) Rule 3: As few network resources (e.g. switches) as possible should be involved. This prefers to choose the shortest route. (4) Rule 4: The allocation for current demand should impact the future allocation as little as possible.

Based on these purposeful criterions, our heuristic power-aware routing algorithm (PAR) is proposed as follows:

1. Search the lowest-level (with biggest β) BI in which the two endpoints are clustered, and generate a set of candidate routes. This is prone to use less critical links thus increasing the success ratio of future resource allocations. Consequently, it yields a tighter network with better link utilization. This step follows Rule 1 and 2, which tries to aggregate the flows into the same subnet (lowest BI) and avoid using critical links.

2. Sort the candidate routes in line with the minimal splitting criterion and select the route that causes fewest BI splittings. This complies with Rule 4 which takes any future allocation into consideration.

3. If there are multiple such routes after step 2, we follow Rule 3 to select the shortest path.

4. If in case there are still more than one best routes, we select the route with the maximum number of flows, which can contribute to the aggregation of network flows, which complies with Rule 3.

5. Finally, we randomly select one route or just select the first route from the ordered candidate routes.

We finish the route selection process as long as the output of any of the above 5 procedures is unique, and allocate the best route to the current demand.

D. Power-efficient Network System

Based on the abstraction techniques and heuristics proposed above, we present a performance-guaranteed Power-efficient Network System (PNS). This system model aims to achieve energy proportionality by powering off a subset of idle network devices while avoiding much sacrifice in the system's fault tolerance. Moreover, this model can be applied in any arbitrary data center network topology.

The general working process of the PNS can be described as follows. Originally, based on the network topology, the system generates a set of different levels of β-BIs and BIH according to the current available link bandwidth. Afterwards, based upon BIH the system computes and allocates the best routing path associated with required bandwidth to each input traffic demand applying BAM and PAR. The output of this step is a set of routes - one for each demand. Then according to the requirement of fault tolerance, to complete the reliability satisfaction procedure by means of adding a certain number of backup routes. Thereafter, the switches, ports or linecards that are not involved in the final routes should be powered off or put into sleep mode for the purpose of saving energy. If in case the reliability requirement cannot be satisfied (i.e no enough redundant paths) then power on certain switches or components, activate the related links, and then update the network topology for the future bandwidth allocation.

V. System Evaluation

In order to demonstrate the effectiveness and good performance of the Power-efficient Network System, we implement the Blocking Island Paradigm and power-aware heuristic algorithms in the DCNSim simulator [13]. Without loss of generality, we use the Fat Tree, which is the most typical data center topology, to evaluate the system with respect to different metrics, like the percentage of power savings under various conditions, the tradeoff between fault tolerance and power savings, and the computation efficiency.

As for the power savings, it depends on various factors including the level of reliability requirements, traffic patterns, data center size, and so on. Therefore, the simulations in this section are carried out under different network conditions. In the simulation, using the traditional always-on strategy as the baseline, we take the percentage of power savings, which is shown as Equation (8), as the power conservation indicator.

Percentage of Power Savings (PPS) =

$$= 100\% - \frac{\text{Power Consumed by PNS}}{\text{Power Consumed by Original Network}} * 100\% \quad (8)$$

According to our experiments, the PPS can reach up to more than 50% on the whole while guaranteeing the network performance, and ranges from 20% to 60% for different network conditions (network scales, network loads, traffic patterns, reliability requirements, etc). Moreover, reducing the network power consumption can also result in cooling energy savings proportionally, though this part of power conservation is not taken into any calculations in this paper.

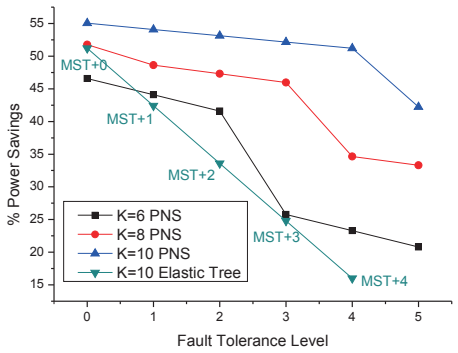

Fig. 5. The performance of the power savings in different fault tolerance levels, using All-to-All traffic pattern for k=6, k=8, k=10 Fat Tree Topology.

A. Traffic Pattern

The types of traffic patterns have a great impact on system's performance and power conservation. The typical traffic patterns include One-to-One, One-to-Many and All-to-All, among which the All-to-All communication simulates the most intensive network activities (such as MapReduce [6]) in the data center network. Hence, the All-to-All traffic pattern is most detrimental to the power conservation. In order to demonstrate the guaranteed performance of our system for the most rigorous case, all the conducted experiments in Section V will use the All-to-All traffic pattern, though the system can achieve more for One-to-One and One-to-Many traffic patterns. Additionally, the Distribution Flow Mode is applied in our experiments to determine the packet inter-arrival time.

B. Experimental Results

This subsection presents the experiment results of the system evaluation. In order to better illustrate the overall performance of the system, the experiment is conducted from the following four aspects.

1) Tradeoff between Power Savings and Fault Tolerance

Fig.5 exhibits the results of power savings under different fault tolerance levels for k=6, k=8, k=10 Fat Tree topology at 20% network load. Here the Greedy Bin-Packing heuristic is applied for evaluating k=10 Elastic Tree with different MST configurations. Clearly, more power savings can be achieved for lower fault tolerance level. The result also reveals that our PNS achieves around 15–20% more power savings on average than ElasticTree. Another finding shows that more energy can be saved for larger sized data center network.

2) Different Network Loads

The percentage of power savings varies under different network conditions. Here we use k=8 Fat Tree (128 servers) topology with 1Gbps links to evaluate the effect of power savings under different network loads using All-to-All traffic pattern. Fig.6-A shows the percentage of power savings for increasing network load under different reliability requirements. The curve for FT=1 in Fig.6-A is the case without considering the reliability requirement and here it is used as the baseline to be compared with higher levels of fault tolerance. The simulation result reveals that higher reliability

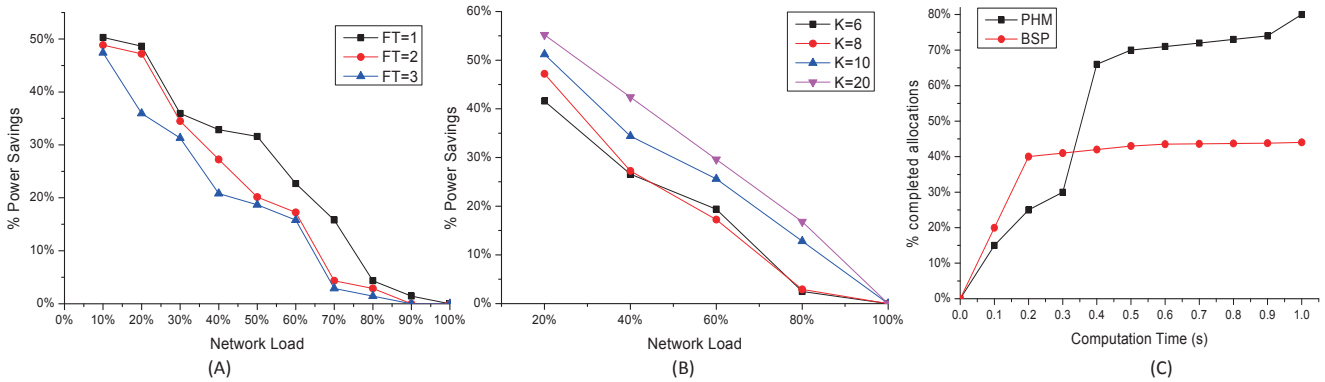

Fig. 6. The performance of PNS in power conservation under various network conditions, and the computation efficiency.

results in less power conservation, which is mainly because some additional switches and ports should be activated for obtaining more available backup paths to meet the higher fault tolerance requirements. Fig.6-B shows the performance of power conservation under different network loads using All-to-All traffic pattern with the fault tolerance level FT=2 for k=6 (54-server), k=8 (128-server), k=10 (250-server), and k=20 (2000-server) sized Fat Tree topology. The result shows that achieving 30%-50% power savings on average is feasible. Moreover, the network load also has a great impact on the power conservation, where the increasing network load degrades the power conservation. The system achieves the most power savings for the lowest network load, and the larger the network size is, the more power can be saved.

3) *Computation Efficiency*

One of the biggest advantages of our framework lies in its computation efficiency. Generally, the traditional routing algorithms (like the shortest-path based routings) have a bad exponential time complexity due to the huge searching space. However, our framework, with the advantage of Blocking Island Paradigm, applies power-aware heuristic scheme to guide the search and achieves a much lower computation complexity by reducing the search space.

Fig.6-C gives the experiment results of the time cost for computing 20000+ instances of bandwidth allocation problems using our power-aware heuristic mechanism (PHM) and the basic shortest-path algorithm (BSP). The result reveals that the PHM is several times faster than the BSP on average. Approximately 80% of allocations can be completed within one second and 100% in several seconds using PHM while BSP only finishes around 40% within one second and tens of seconds for 100%. This demonstrates the great performance of PHM in computation efficiency even though the generation and maintenance of the BIH may take some time.

VI. Conclusion

In this paper, we formulate the energy optimization problem as a MCF problem and prove its NP-hardness. By drawing the inspiration from an artificial intelligence abstraction technique BI, we have proposed an efficient bandwidth allocation mechanism BAM and heuristic power-aware routing algorithm PAR. Afterwards, we further design a general and efficient green framework by combing BAM and PAR to achieve an energy proportional data center network. Finally, extensive simulations are conducted to evaluate this green framework and the results convince its good performance.

VII. Acknowledgement

This paper is supported in part by NPRP grant No.6-718-2-298 from the Qatar National Research Fund.

References

[1] Dennis Abts, et al. Energy proportional datacenter networks. In *ACM SIGARCH Computer Architecture News*, pages 338–347. ACM, 2010.
[2] M. Arlitt, et al. Towards the design and operation of net-zero energy data centers. In *13th IEEE Intersociety Conference*. IEEE, 2012.
[3] A. Carrega, et al. Applying traffic merging to datacenter networks. In *ACM e-Energy*. ACM, 2012.
[4] P Costa, et al. Camcube: a key-based data center. Technical report, Technical Report MSR TR-2010-74, Microsoft Research, 2010.
[5] H. D., et al. Memory power management via dynamic voltage/frequency scaling. In *8th international conference on Auto computing*. ACM, 2011.
[6] Jeffrey Dean, et al. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.
[7] C. Frei, et al. A dynamic hierarchy of intelligent agents for network management. In *IATA*. Springer, 1998.
[8] Íñigo Goiri, et al. Greenhadoop: leveraging green energy in data-processing frameworks. In *7th EuroSys Conference*. ACM, 2012.
[9] Robert M Haralick, et al. Increasing tree search efficiency for constraint satisfaction problems. *Artificial intelligence*, 14(3):263–313, 1980.
[10] Brandon Heller, et al. Elastictree: Saving energy in data center networks. In *NSDI*, volume 3, pages 19–21, 2010.
[11] Jacob Leverich, et al. Power management of datacenter workloads using per-core power gating. *Computer Architecture Letters*, 8(2):48–51, 2009.
[12] L. Liu, et al. Greencloud: a new architecture for green data center. In *the 6th ICAC conference on industry session*. ACM, 2009.
[13] Y. Liu and J. Muppala. Dcnsim: A data center network simulator. In *the 3rd international workshop on Data Center Performance*. IEEE, 2013.
[14] Priya Mahadevan, et al. On energy efficiency for enterprise and data center networks. *Communications Magazine, IEEE*, 49(8):94–100, 2011.
[15] David Meisner, et al. Powernap: eliminating server idle power. In *ACM Sigplan Notices*, volume 44, pages 205–216. ACM, 2009.
[16] Kim-Khoa Nguyen, et al. Powering a data center network via renewable energy: A green testbed. *Internet Computing, IEEE*, 17(1):40–49, 2013.
[17] K. R, et al. Thread motion:fine-grained power management for multicore systems. In *ACM SIGARCH Computer Architecture News*. ACM, 2009.
[18] Y. S., et al. Energy-aware routing in data center network. In *Proceedings of the 1st ACM SIGCOMM workshop on Green networking*. ACM, 2010.
[19] A. Singla, et al. Proteus: a topology malleable data center network. In *the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, 2010.
[20] V. Valancius, et al. Greening the internet with nano data centers. In *the 5th CoNEXT conference*. ACM, 2009.
[21] Ting Wang, et al. NovaCube: A Low Latency Torus-Based Network Architecture for Data Centers. IEEE GlobeCom, 2014. to appear.