Achieving Energy Efficiency in Data Centers Using an Artificial Intelligence Abstraction Model

Ting Wang, *Student Member, IEEE*, Yu Xia, *Member, IEEE*,
Jogesh Muppala, *Senior Member, IEEE*, and Mounir Hamdi, *Fellow, IEEE*

Abstract—Today's data center networks are usually over-provisioned for peak workloads. This leads to a great waste of energy since in practice traffic rarely ever hits peak capacity resulting in the links being under-utilized most of the time. Furthermore, the traditional non-traffic-aware routing mechanisms worsen the situation. From the perspective of resource allocation and routing, this paper aims to implement a green data center network and save as much energy as possible. With the benefit of blocking island paradigm, we present a general framework trying 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 green tighter network. Moreover, our fast efficient heuristics for allocating bandwidth enable the system to scale to large sized data centers. The evaluation result shows that achieving up to more than 50 percent power savings are feasible while guaranteeing network performance and reliability.

Index Terms—Green data center, power-aware routing, energy saving, bandwidth allocation

1 Introduction

Large-scale data centers provide the core infrastructure to meet the computing and storage requirements for both enterprise IT needs, and cloud-based services, such as search (e.g., Google Search, Bing), social networking (e.g., Facebook, Twitter), Storage (e.g., Dropbox, Skydrive, Google Drive) and software applications (e.g., Google Apps, Microsoft Office 365) [1]. 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 [2]. 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 [3], [4]. The average link utilization ranges between 5 and 25 percent with wide diurnal variations [5]. 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 (an idle switch consumes up to 90 percent of the peak power consumption [6]). Based on this careful observation, it can be derived that 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 just powered off for the sake of saving energy [5], [4], [7], [8].

According to current research findings [9], [10], the power consumed by servers and infrastructure (i.e., power distribution and cooling) accounts for over 70 percent of overall power, while the network consumes around 15 percent of the total power budget. However, as the servers become more energy proportional, the fraction of the power consumed by the network in a data center will grow correspondingly higher. As illustrated in [3], suppose the servers are totally energy-proportional, when the data center is 15 percent utilized (servers and network), then the network will consume up to 50 percent of overall power. Even if the servers are not energy-proportional, with 15 percent traffic load, making the network proportional still can save as much as 975 KW (for a data center with 10,000 servers) [3]. Unfortunately, today's commodity network devices are not energy proportional, 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.

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 [11] and can have an impact on the network performance, therefore our approach provides a solution which performs 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 as has been proved in this paper by reducing it to

- *T. Wang was a PhD student at the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. E-mail: twangah@connect.ust.hk.*
- *Y. Xia is with the College of Computer Science, Sichuan Normal University, China. E-mail: rainsia@163.com.*
- *J. Muppala is with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. E-mail: muppala@cse.ust.hk.*
- *M. Hamdi is with the College of Science, Engineering and Technology in Hamad bin Khalifa University, Qatar. On leave from Hong Kong University of Science and Technology, Hong Kong. E-mail: hamdi@cse.ust.hk.*

Manuscript received 23 Aug. 2014; revised 29 Oct. 2015; accepted 10 Dec. 2015. Date of publication 0 . 0000; date of current version 0 . 0000.
Recommended for acceptance by D. Wei.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TCC.2015.2511720

a 0-1 Knapsack problem. Nevertheless we achieve our goal efficiently using the Blocking Island resource abstraction technique together with well-designed heuristic algorithms. The final target 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) We propose an efficient bandwidth allocation mechanism (BAM) and a practical power-aware routing (PAR) algorithm.
3) 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.
4) We design the Power-efficient Network System (PNS) with the target of maximizing power savings and minimizing sacrifices of network performance and fault tolerance.
5) 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 2. Then we formulate the energy optimization problem dealt with in this paper in Section 3. The Blocking Island Paradigm is briefly reviewed in Section 4. Then Section 5 introduces the power-aware heuristic routing scheme followed by the design of Power-efficient Network System in Section 6, and Section 7 presents the system evaluation and simulation results. Finally, Section 8 concludes the whole paper.

2 Related Work

A considerable amount of investigation and research for achieving a green data center have been conducted in both academia and industry due to its great potential and benefits. Current existing research efforts in achieving a green data center can be summarized as follows.

2.1 Network-Level Power Conservation Approach

This kind of research aims to save the energy consumed in the network by exploiting energy-aware routing algorithms to consolidate traffic flows to a subset of the network and power off the idle devices. Proposed schemes like Merge Networks [5], Elastic Tree [4], Energy-aware Routing Model [7] are some typical representatives of this approach.

Merge Networks [5] focuses on reducing the power consumed by a switch. The key idea is to consolidate the low traffic from N links to K links with high traffic and put the remaining ports to low power mode. For example, for a switch with eight ports, if its average link utilization is 10 percent, then all the traffic can be merged to one single link with only one port activated at maximum rate, and the rest of ports enter to low power mode. This approach, which focuses on power savings of a single switch, saves less power than other approaches which merges traffic to a subset of the network at a global view.

Elastic Tree [4] proposes multiple energy-aware strategies by finding minimum-power network subsets and shutting down the unused network elements. Elastic Tree consists of three logical modules, which are optimizers, routing and power control. Optimizer targets at finding the minimum network subset to satisfy all traffic, routing calculates the paths for flows, and power control manages the states of network devices. The authors also proposed three different optimizers with different solution qualities, which are formal model, greedy bin packing, and topology aware heuristic. The formal model aims to find the optimal flow assignment, however, with a high computational complexity, which may cost several hours for solutions. Besides, its scalability is very poor, which can only scale up to networks with less than 1,000 nodes. The greedy bin-packing heuristic always chooses the leftmost path for each flow to consolidate the them into a network subset. When all flows are assigned, then power off the unused network devices. Although this algorithm improves the scalability, its solution quality cannot be guaranteed and it requires complete knowledge of the traffic matrix. The topology aware heuristic achieves the best scalability $O(n)$ with smallest computation time cost, but its solution quality is worst compared to other two optimizers.

Energy-aware Routing Model [7] is also a network level approach, which aims to use as few switches as possible to satisfy a given traffic matrix while meeting a predefined throughput threshold. The basic idea is that: First, they take all switches into consideration and compute basic routing and basic throughput. Then, they gradually eliminate the switches from basic routing and recompute routing and throughput until the throughput reaches the predefined threshold. Finally, the switches not involved in the routing are powered off for power savings. This approach suffers inefficient computation efficiency, where it takes several seconds to calculate a non-optimal power-aware routing paths for thousands of flows and takes even hours to calculate a near optimal solution, which is intolerable for a latency-sensitive data center network.

There are other similar network level approaches, in addition to the ones presented above, such as Willow [12], DCEERS [13], GreenDCN [14] and [15]. Admittedly, the network level approaches seem to be promising, however, there are still many critical issues that need to be addressed, such as computing complexity, the sacrifice of network reliability, the impact on network performance caused by powering off devices, and so on. In response to these issues, in this paper we employ an AI resource abstraction model to aim to achieve a performance (bandwidth) guaranteed energy efficient data center network, with high computation efficiency, from the perspective of resource allocation and flow scheduling.

2.2 Other Power Conservation Approaches

2.2.1 Use Renewable or Green Sources of Energy

This approach makes use of green sources to reduce the energy budget such as 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, which should also be counted in. The related works in this field includes GreenHadoop [16], GreenStar Network Testbed [17], and Net-Zero Energy Data Centers [18].

2.2.2 Design More Energy-Efficient Hardware

This method is dedicated to designing energy proportional servers and switches, for example, by exploiting the advantages of Dynamic Voltage and Frequency Scaling (DVFS), and vary-on/vary-off (VOVO) technique to adapt the traffic rate variations. Currently numerous hardware based proposals have been put forward, among which the typical representatives include Low Energy Switch Block for FPGAs [19], Powernap [20], Energy Management for Commercial Servers [21], Thread Motion [22], Energy-Efficient Server Clusters [23], PCPG (per-core power gating) [24], and Memory Power Management via DVFS [25].

2.2.3 Design Energy-Efficient Network Architecture

This means to design new energy-efficient data center network architectures to achieve power conservation. Examples like flattened butterfly topology [3], 3D Torus based Camcube [26], Small-World [27], NovaCube [28], Nano Data Centers [29], Proteus [30], Pcube [31], content-centric networking (CCN) based architectures which can reduce the content distribution energy costs [32], [33] can be classified into this category. These alternative approaches are attractive, but still need to be further evaluated in real data centers.

2.2.4 Taking Advantage of Virtual Machine Technology

This scheme is proposed to reduce the energy consumption by drawing support from techniques like the Virtual Machine (VM) migration and placement optimization, for example GreenCloud [34], TRP/VCS [35], VPTCA [36] and VMFLow [37]. Virtual machines are assigned or migrated depending on the characteristics of the applications and the features of network topology, which can in turn better improve the traffic engineering. However, at the same time it also brings about some additional costs induced by migrating VMs over the long-term, such as the extra time to complete the migration and the large amount of generated traffic between the source and destination servers of the VM migration, which is very bandwidth greedy.

3 Problem Statement

3.1 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, multi-channel contention problem [38], and flow scheduling problems. In the next section, we show that the energy-aware routing problem can also be formulated as an MCF problem.

3.2 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)). Constraint (6) means that if demand d_{ij} goes through any link of switch S_i then $S_i = 1$, otherwise $S_i = 0$. The right side of the inequality in Constraint (6) is a value between 0 and 1, where it will be greater than 0 once any of S_i's links are selected.

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 $\forall i, j \in V$,

TABLE 1
Summary of Notations in Problem Formulation

Notations	Descriptions
V	switch node set
E	link set
Ω_s	power consumed by fixed overheads
Ω_p	power consumed by a port
S_i	whether switch i is chosen
L_k	whether link k is chosen
$x_{ij}^{(k)}$	value of demand d_{ij} on link k
$R(d_{ij})$	the number of available paths for demand d_{ij}
FT	number of backup routes
C_k	bandwidth capacity of link k
N_i	set of links adjacent to switch i

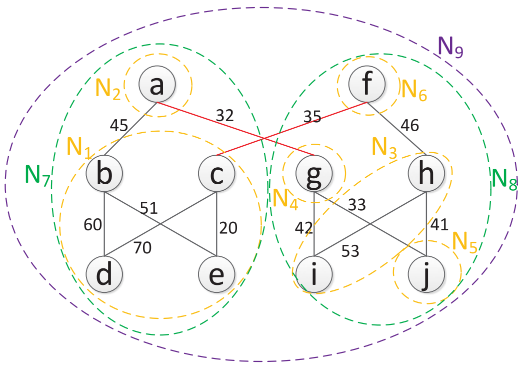


Fig. 1. An example of Blocking Island Graph, in which N$_1$-N$_6$ are 50-BIs, N$_7$-N$_8$ are 40-BIs, and N$_9$ is 30-BI. The red lines are critical links between two 40-BIs. The weights on the links are their available bandwidth.

$\sum_{k\in N_i} Y_{ji}^{(k)} \geq FT, \sum_{k\in N_i} Y_{ij}^{(k)} \geq FT$ with $Y_{ji}^{(k)} \geq x_{ij}^{(k)}/C_k$ and $Y_{ji}^{(k)} \in \{0,1\}$. The descriptions of notations are summarized in Table 1.

3.3 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 $I_1, I_2, \ldots, I_n$, 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 $\sum_{i\in S} V_i \geq P$. 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. □

4 Blocking Island Paradigm

Derived from Artificial Intelligence, Blocking Island (BI) 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* [39]. As shown in Fig. 1, N_1 is a 50-BI for node b.

β-BIs have several fundamental properties which are very useful in routing decidability. Here we list some of the most important ones.

Unicity. Each node has one unique β-BI. Thus, if S is the β-BI for node x, then S is the β-BI for all the nodes in S.

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. For example, in Fig. 1 the demand $(d, e, 50)$ can be satisfied since both d and e are in the same 50-BI while $(d, i, 50)$ cannot be allocated.

Route location. The links of a route with β available bandwidth are all in the β-BI of its endpoints.

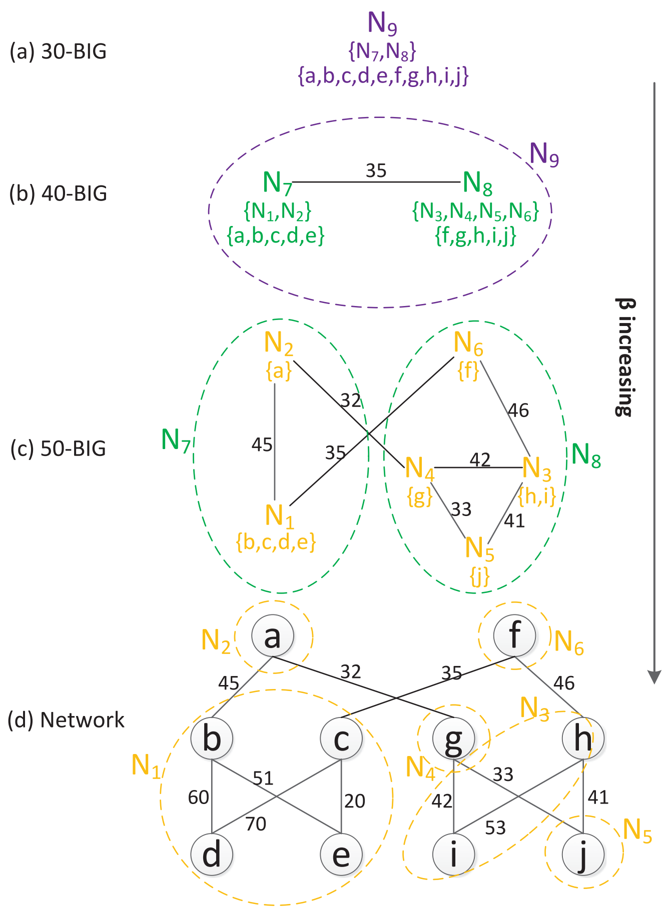


Fig. 2. An example of Blocking Island Hierarchy (BIH) for bandwidth requirements $\beta = \{50, 40, 30\}$.

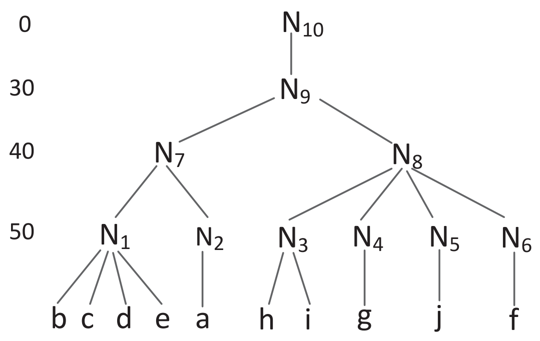


Fig. 3. Abstraction tree of BIH of Fig. 2, and N_{10} indicates 0-BI.

Algorithm 1. Construct β-BI

```
 1: function CONSTRUCTBI $N = \{V, E\}, \beta$
 2:   $L \leftarrow \{\emptyset\}$                          ▷ $L$: Result $\beta$-BI list
 3:   for all $v$ in $V$ do
 4:     if not visited($v$) then
 5:       $I \leftarrow ConstructBIFromNode(N, \beta, v)$
 6:       $L \leftarrow L.Add(I)$
 7:     end if
 8:   end for
 9:   return $L$
10: end function
11: function CONSTRUCTBIFROMNODE($N = \{V, E\}, \beta, x$)
12:   $I \leftarrow \{x\}$                               ▷ $I$: Result $\beta$-BI
13:   $S \leftarrow$ {links incident to $x$ and weight $\geq \beta$}   ▷ $S$: stack
14:   while $S \neq \emptyset$ do
15:     $l \leftarrow pop(S)$
16:     $e$     another endpoint of $l$
17:     if $e \notin I$ and $weight(l) \geq \beta$ then
18:       $I$     $I \cup \{e\}$
19:       $S$     $S \cup$ {links incident to $e$ and weight $\geq \beta$}
20:     end if
21:   end while
22:   return $I$
23: end function
```

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 (as depicted in Algorithm 1) 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), which is a graph abstraction of the whole available network resources, and it can be denoted as G = (S, L), where S indicates the set of all different β-BIs while L denotes the critical links between BIs. Fig. 1 gives an example of BIG.

We can further construct a recursive decomposition of BIGs in decreasing order of demand (βs), and the lowest level has the largest β. This layered BIG structure is called Blocking Island Hierarchy (BIH). It can be used to identify all bottlenecks, i.e., critical links, of the network. An example of BIH is illustrated in Fig. 2. The BIH can also be viewed as an abstraction tree (see Fig. 3) when taking the father-child relation into consideration. The leaves of the BIH tree are the network nodes, and the other vertices denote the abstract BIs.

BI significantly reduces the routing search space with bandwidth guarantee. Besides, in line with the Data Center policy which requires fast response to the request, 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.

5 Power-Aware Heuristic Routing 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 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.

5.1 Bandwidth Allocation Mechanism

As a common issue for the constraint satisfaction problem (CSP) or MCF problem, for some traffic matrices it cannot

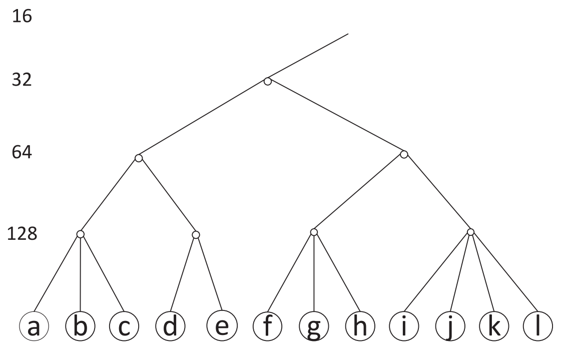


Fig. 4. The BIH example used for DSLH and DSFN.

be guaranteed to find an assignment to satisfy all the flows all the time, which is also mentioned in [4]. 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 [40], 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 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. 4, 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. 5. 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 which combines DSLH with DSFN, and it works as follows:

1. First, 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 Figs. 4 and 5, but we now have only three demands d_2, d_3, d_4 to allocate. After step 1, both 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.

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. 5. The example for comparison between DSLH and DSFN.

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.

5.2 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 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.

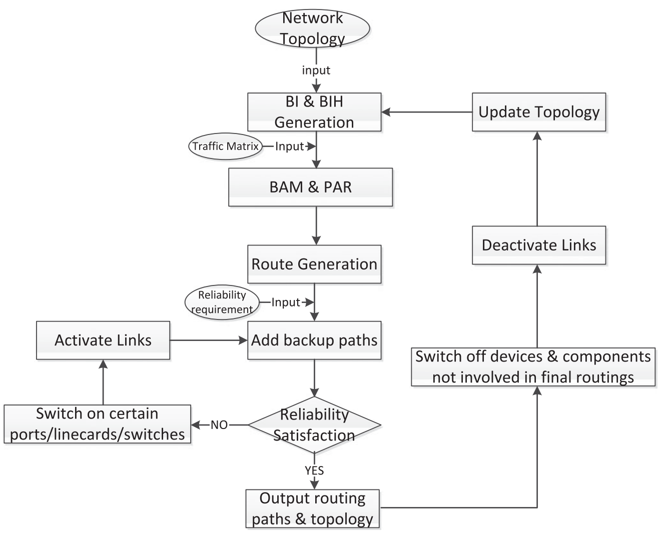


Fig. 6. The working procedure of the power-efficient network system.

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 five procedures is unique, and allocate the best route to the current demand.

6 Power-Efficient Network System Design

Based on the abstraction technique and heuristics proposed above, in this section, we present a performance-guaranteed Power-efficient Network System, which can be applied in any arbitrary data center network topology.

6.1 Power-Efficient Network System Overview

The power-efficient network system concentrates on providing an efficient approach to maximize power conservation by satisfying the traffic constraints (demand completion, reliability requirement, capacity constraint, flow conservation) and computing the optimal demand allocation decisions. The input of the system includes the original network topology, traffic matrix with bandwidth requirements and the reliability requirements. The expected final outputs of the system are a subnet of the original network topology and a set of routing paths taken by demands with satisfied bandwidth. Besides, it also provides a reasonable tradeoff between the power savings and the network fault-tolerance according to the reliability requirement.

Fig. 6 depicts the general working process of the PNS. 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 are 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.

6.2 Power Conservation Strategy

As noted in [3], [4], [6], [7], though there are no packets to transmit, the power consumed by the idle device or component is still quite significant compared to that of when the network is fully utilized. Hence, the most beneficial way for saving energy is to totally power off the idle devices or components, and only keep the required networking capacity available.

Generally, there are two different ways to power off a network device, namely device level and component level. Most of existing proposals, e.g., ElasticTree [4], apply the device level, which means to power off the entire device and save the overall power consumed by the device, including the power consumed by the fixed overheads (like fans, linecards, transceivers, etc). This means even if only one port has data to forward, the whole device should be kept alive. Comparatively, the component level is more competitive. If there is no data transferring on a link then disable its connected ports, and further power off a linecard if all the ports on this linecard are disabled. If all the linecards are idle then power off the entire device to achieve the most power conservation. In view of this, our proposed power conservation strategy adopt the more beneficial component level fashion. It is important to note that the devices switched off can not be critical ones that could cause disconnection of the network. And every server should be guaranteed reachable at any time regardless of the requirement degree of reliability.

As for the computation of the power consumption of a switch, it is very complicated and difficult to be actually calculated because the power consumed by the device is somehow affected by the traffic rate across them, while the traffic rate changes constantly. Nevertheless, we observe that the power consumption of an idle device or component is close to that of a fully loaded one, hence for simplicity we ignore the effects caused by traffic rate to get a slightly rough result. The total power consumption of a switch Ω_{switch} can be computed according to the equation (8):

$$\Omega_{switch} = \Omega_s + N_p * \Omega_p, \tag{8}$$

where Ω_s and Ω_p are the same as described in Section 3, noticing that Ω_p represents the approximate power consumption of a port without considering the varying traffic rate, and N_p denotes the number of active ports on the switch. Clearly, the results calculated according to equation 8 are slightly higher than the real power consumption, so the exact power conservation of our system is actually better than the evaluation results (Section 7).

6.3 Reliability Satisfaction

Typically the data center architecture is constructed with lots of redundant equal cost routes between any pair of servers against the cases of network failures. Take Fat Tree [41], BCube [42] and SprintNet [43], [44] for example, given a Fat Tree DCN with the size of 27,648 hosts using 48-port switches, there are 576 equal-cost paths between any two hosts in different pods [41], and in a BCube_k and SprintNet_k there are $k+1$ and $k+c$ parallel (node-disjoint) paths between any two servers, respectively, where c indicates the number of switches in each basic $Cell$. Therefore, it is quite feasible to power off the idle devices on these redundant paths for achieving power savings while guaranteeing the reachability of each server.

However, although powering off idle network devices for power savings seems to be attractive and promising, there is an inevitable conflict between powering off devices and maintaining the network's fault tolerance level. Basically, the power conversation strategy tends to switch off as many idle devices as possible, while on the contrary the robustness of the system requires keeping as many devices active as possible to preserve more redundant routes so as to achieve good fault tolerance. Consequently, there must be a tradeoff between power conservation and fault tolerance.

As shown in Formula 9, our proposal gives a way to make a reasonable compromise between power savings and robustness, where *PPS* is as defined in Equation (10) in Section 7, and the weighting factor α specifies the required tradeoff between fault tolerance (*FT*) and power savings (*PPS*),

$$Max\ \{FT + \alpha * PPS\}. \tag{9}$$

To selecte backup routes we apply the shortest-path routing algorithm, which is different from the Route Selection Rules. The basic reasoning of using such a route selection strategy is that we intend to use fewest network resources and preserve as few devices as possible to meet the fault tolerance requirement, and in turn maximize the power savings. From another perspective, the network failure rates in practice, after all, are relatively low (5 percent for the TOR switches per year [45]) which indicates the devices are fairly reliable, so it is not so wise to sacrifice too much (network resources, power, computation time, etc) for the small probability event. Therefore, the shortest-path routing algorithm is quite suitable and enough for the backup path selection.

6.4 Discussion on the System Implementation

The PNS can be implemented as a centralized system, where a controller is needed. The BI based bandwidth allocation mechanism and power aware routing algorithm can be implemented in the controller. The controller can be an independent server or hosted on a compute server. The network could use OpenFlow network technology to monitor the network status in runtime. The port statistics and switch status are collected by the centralized controller from OpenFlow enabled switches through the OpenFlow secure channel. In order to deal with the single point failure of the controller, multiple controllers can be used, where these controllers can have different roles: OFPCR_ROLE_EQUAL, OFPCR_ROLE_MASTER, OFPCR_ROLE_SLAVE (as specified in [46]) to guarantee the robustness of the system. The controller-to-switch messages (such as switch configuration messages, flow table configuration messages, modify-state messages, read-state messages, barrier messages, and role request messages), asynchronous messages and symmetric messages (Hello messages, Echo messages, and Experimenter messages) can be transmitted through OpenFlow secure channel either in band or out of band. With the help of OpenFlow, the controller is much easier to manage the whole network globally which can further improve the system's flexibility. Additionally, in this system the transport layer (e.g., TCP, UDP, etc.) and network layer (e.g., IP) remain the same as legacy protocols without any modifications.

7 System Evaluation

In order to demonstrate the effectiveness and good performance of the Power-efficient Network System, we implement the Blocking Island Paradigm and the power-aware heuristic algorithms in our DCNSim simulator [47] for the system's evaluation.

7.1 Simulation Environment

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. The default unidirectional link bandwidth is 1,000 MBps and each link is capable of bidirectional communications. The propagation delay of a link and the processing time for a packet at a node are set to be 5 μs and 10 μs, respectively. The default maximum transmission unit (MTU) of a link is 1,500 bytes, and the TTL of a packet is set to be 128. For the generation of packet distributions to determine the packet inter-arrival time, we adopt Weibull Distribution using the Simjava package of eduni.simjava.distributions [48]. The scale parameter λ and shape parameter k are set to be $\lambda = 1$ and $k = 1.5$ as default.

7.2 Power Conservation Indicator

How much power can be saved depends on various factors including the level of reliability requirements, traffic patterns, data center size, and so on. Therefore, the evaluations of our system PNS 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 (denoted by *PPS*), which is shown as Equation (10), as the power conservation indicator,

$$PPS = 100\% - \frac{\Omega_{PNS}}{\Omega_{ON}} * 100\%, \tag{10}$$

where Ω_{PNS} indicates the power consumed by PNS, and Ω_{ON} represents the power consumed by original network.

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.

7.3 Traffic Pattern

The traffic patterns have a great impact on system's performance and power conservation. In our experiments, we consider the traffic from two perspectives: the locality of

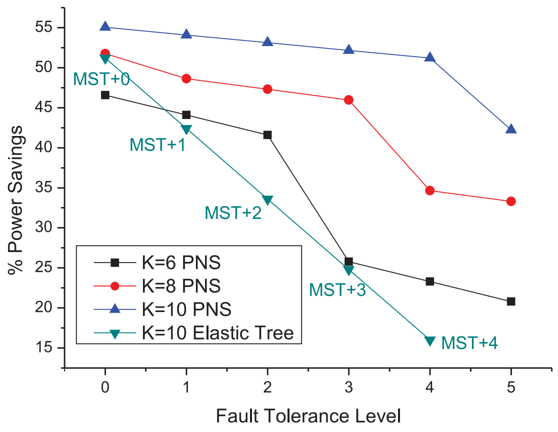


Fig. 7. The performance of the power savings in different fault tolerance levels, using All-to-All traffic pattern for k = 6, k = 8, k = 10 Fat Tree Topology.

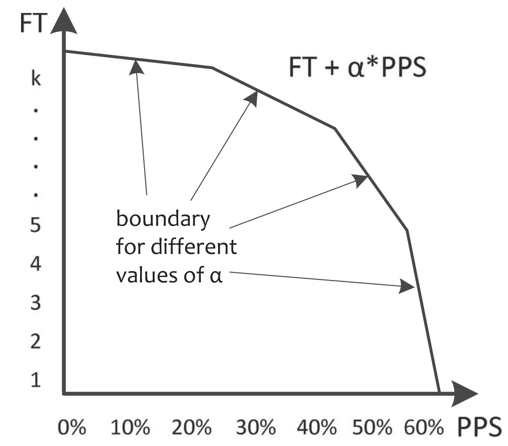


Fig. 8. The tradeoff boundary for fault tolerance and power savings.

communicators and the type of communicator-pairs. Considering the locality of communicators, we have: (1) localized traffic, where servers only communicate with other servers under the same edge switch; (2) non-localized traffic, which transfers across inter-pods; (3) random traffic, which involves both the above two traffic types. As for the type of communicator pairs, there are three typical traffic patterns in data centers, including One-to-One, One-to-Many, and All-to-All. Furthermore, according to the findings in [49] about the characteristics of the packet-level communications, the packet inter-arrival time reveals an ON/OFF pattern and its distribution follows the Lognormal mode for OFF phase, while the distribution varies between Lognormal mode, Weibull mode and Exponential mode in different data centers during the application-sensitive ON phase. Thus, Weibull distribution flow mode is applied to determine the packet inter-arrival time.

7.4 Simulation Results

This section presents the 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. 7 gives the simulation results of network power savings under different fault tolerance levels for k=6, k=8, k=10 Fat Tree topology using All-to-All traffic pattern at 20 percent network load. Here the Greedy Bin-Packing heuristic is applied for evaluating k=10 Elastic Tree with different MST configurations. Indeed, the behavior of powering off idle switches for power conservation causes the degradation of the fault tolerance, and more power savings can be achieved for lower fault tolerance level. The result also reveals that our PNS using the component level power conservation strategy achieves around 15-20 percent more power savings on average than ElasticTree. Another finding implies that more energy can be saved for larger sized data center network.

As introduced in Section 6.3, a satisfactory tradeoff between fault tolerance and power savings can be made according to $Max\{FT + \alpha * PPS\}$. Given different weights α to PPS, we can achieve different tradeoffs as illustrated in Fig. 8, where higher α gives more weights to achieving better PPS at the cost of less improvement to FT. The value of α can be decided by the network administrator according to the different requirements of reliability and power savings.

2) *Under Different Network Loads*

In this simulation, we use k=8 Fat Tree (128 servers) topology to evaluate the effect of power savings under different network loads using All-to-All traffic pattern.

Fig. 9 shows the percentage of power savings for increasing network load under different relaibility requirements. The curve for FT=1 in Fig. 9 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 result reveals that higher reliability 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. Moreover, the network load also has a great impact on the power conservation. The system achieves the most power savings for the lowest network load, and the larger the network size the more power can be saved. Admittedly, the increasing network load degrades the power conservation. When the network is fully loaded, there is no possibility to achieve any power savings. However, as noted in [5], the average link utilization in real data centers is just around 5-25 percent, which implies 30-50 percent power savings are feasible in practice. Compared with

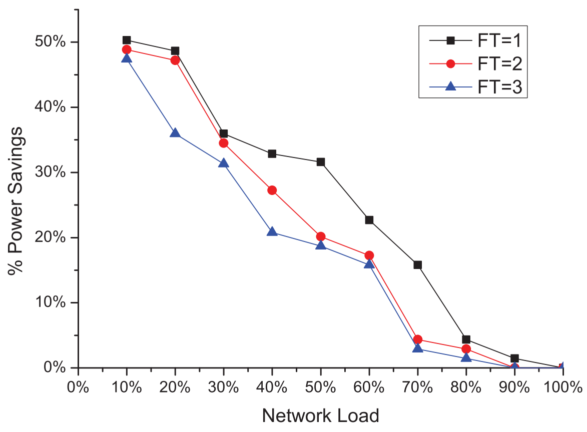


Fig. 9. The power conservation under different network loads & fault tolerance levels.

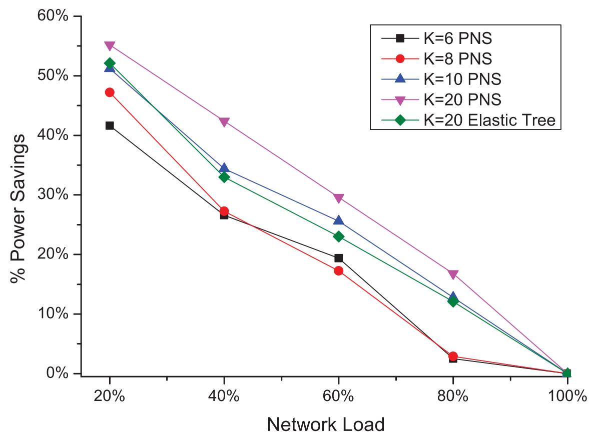


Fig. 10. The power conservation under different network loads using All-To-All traffic pattern with FT=2 for different scales of Fat Tree topology.

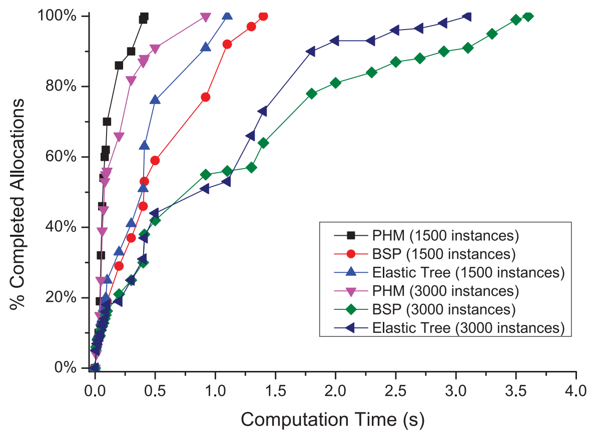


Fig. 11. The comparison of computation efficiency between PHM and BSP.

other similar works, one may find that our model cannot gain near 100 percent power savings at very low network loads. This is mainly because in our model the critical switches that may cause network disconnections cannot be powered off and each server should be guaranteed reachable at any time.

3) *Different Traffic Patterns*

As mentioned above, the totally localized traffic indicates the communications between servers only take place under the same edge switch, which is the most ideal case for power conservation. The experiment shows that it can gain more than 60 percent of power savings on average for any sized network while guaranteeing the connectivity of the network. On the contrary, the completely non-localized traffic is the adverse extreme case where servers only communicate with other servers in different pods. The power conservation for non-localized traffic varies largely for different network loads. In our experiments, for example Fig. 9, we use random traffic which involves both localized and non-localized traffic, which is the most common case in practice.

Moreover, we have One-to-One, One-to-Many and All-to-All traffic patterns when considering the type of communicator pairs. The All-to-All communication simulates the most intensive network activities (such as MapReduce [11]) in the data center network, and every server communicates with all the other servers. Hence, the All-to-All traffic pattern is most detrimental to the power conservation. In order to demonstrate the performance of our system for the most rigorous case, all the conducted experiments in Section 7 use the All-to-All traffic pattern. Fig. 10 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 (2,000-server) sized Fat Tree topology. The simulation result convinces the good performance of PNS, which shows that achieving 30-50 percent power savings on average is feasible. And it can comparatively gain better achievements for One-to-One and One-to-Many traffic patterns. Besides, it also reveals that for a 2,000-server Fat Tree PNS achieves 3-9 percent more power savings than Elastic Tree with the same level fault tolerance.

4) *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 routing scheme to guide the search and achieves much higher computation efficiency by reducing the search space.

Fig. 11 gives the evaluation results of the time cost for computing 1,500 and 3,000 instances of bandwidth allocation problems using our power-aware heuristic mechanism (PHM), greedy bin-packing algorithm of Elastic Tree and the basic shortest-path algorithm (BSP) in a 2,000-server Fat Tree (k=20). The result reveals that the PHM is several times faster than both Elastic Tree and BSP on average. 100 percent of 1,500 and 3,000 allocation instances can be completed in 0.412 and 0.921 seconds, respectively, using PHM while Elastic Tree costs 1.135 and 3.142 seconds, correspondingly. The time cost of BSP is the most, which takes 1.414 and 3.602 seconds to finish 1,500 and 3,000 instances of allocations, respectively. Moreover, Table 2 gives the time cost in completing 12,000 allocation instances, which reveals that PHM saves more time in searching and computing routes for demands allocation than other traditional routing algorithms in the long run. All these simulation results verify the advantages of PHM in computation efficiency.

7.5 Discussion on PHM Implementation in Real World Scenario

We have provided theoretical analysis and simulation studies for the proposed green scheme PHM. Although the simulation conditions are very close to the real world data center environments, there are still some issues needed to be considered in real world deployment of PHM. In real

TABLE 2
Time Cost in Completing 12,000 Allocation Instances

Algorithms	Running Time (s)
PHM	12.253
ElasticTree	73.514
BSP	86.260

world data centers, the packet inter-arrival time usually reveals an ON/OFF pattern and its distribution follows the Lognormal mode for OFF phase, while the distribution varies between Lognormal mode, Weibull mode and Exponential mode in different data centers during the application-sensitive ON phase. In our simulations, we only evaluated the Weibull distribution flow mode, and the performance of PHM under other one or several mixed traffic patterns in a real world still needs further evaluation. Second, we also care about how the time cost in switching off/on a switch will affect the system performance in real data centers, which is actually a common concern of this research field. According to the findings in [4], the time cost in booting up a switch ranges from 30 seconds for Quanta switch to 3 minutes for HP switch, while powering on/off a port takes 1-3 seconds. Fortunately, nowadays most of switches have supported the sleep mode, where the switches can be put into sleep mode instead of being shut down when they are idle. The sleep mode will be significantly faster to be waken up, where the transition time from sleep mode to active mode is only 1-10 ms [50]. Third, another big concern of this research field is how to power on the device again remotely when needed. The simplest way is to power on switches manually according to the computation output. Considering the amount of switches in data centers may be very large, doing it manually is very tough and inefficient. Comparatively, there are also many state-of-the-art techniques to power on switches remotely and automatically, for example, the Remote Power Switch [51], AVIOSYS Power Switches [52], remote power control via data link [53], the Wake-on-LAN like techniques [54], and so on. What's more, switches enabled with sleep mode will be more easily and faster to be awakened/activated when needed.

8 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 propose an efficient bandwidth allocation mechanism and heuristic power-aware routing algorithm PAR. Afterwards, we further design a Power-efficient Network System by combing BAM and PAR to achieve an energy proportional data center network. Moreover, we have also discussed how to make a reasonable tradeoff between power savings and network reliability. Finally, extensive simulations are conducted to evaluate this green framework. According to our simulation results, the PPS can reach up to more than 50 percent on the whole for guaranteeing the network performance, and ranges from 20 to 60 percent for different network conditions (network scales, network loads, traffic patterns, reliability requirements, etc). This result further convinces the feasibility and good performance of our approach.

Acknowledgments

This publication was made possible in part by NPRP grant No. 6-718-2-298 from the Qatar National Research Fund. Besides, the authors would like to express their thanks and gratitudes to the anonymous reviewers for their valuable feedback and advice.

References

[1] T. Wang, Z. Su, Y. Xia, and M. Hamdi, "Rethinking the data center networking: Architecture, network protocols, and resource sharing," *IEEE Access*, vol. 2, pp. 1481–1496, 2014.
[2] T. Wang, Y. Xia, D. Lin, and M. Hamdi, "Improving the efficiency of server-centric data center network architectures," in *Proc. IEEE Int. Conf. Commun.*, 2014, pp. 3088–3093.
[3] D. Abts, M. R. Marty, P. M. Wells, P. Klausler, and H. Liu, "Energy proportional datacenter networks," in *Proc. ACM SIGARCH Comput. Archit. News*, 2010, vol. 38, pp. 338–347.
[4] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, "Elastictree: Saving energy in data center networks," in *Proc. 7th USENIX Conf. Netw. Syst. Des. Implementation*, 2010, vol. 3, pp. 19–21.
[5] A. Carrega, S. Singh, R. Bolla, and R. Bruschi, "Applying traffic merging to datacenter networks," in *Proc. 3rd Int. Conf. Future Energy Syst.: Where Energy, Comput. Commun. Meet*, 2012, p. 3.
[6] P. Mahadevan, S. Banerjee, P. Sharma, A. Shah, and P. Ranganathan, "On energy efficiency for enterprise and data center networks," *IEEE Commun. Mag.*, vol. 49, no. 8, pp. 94–100, Aug. 2011.
[7] Y. Shang, D. Li, and M. Xu, "Energy-aware routing in data center network," in *Proc. 1st ACM SIGCOMM Workshop Green Netw.*, 2010, pp. 1–8.
[8] T. Wang, Y. Xia, J. Muppala, M. Hamdi, and S. Foufou, "A general framework for performance guaranteed green data center networking," in *Proc. IEEE Global Commun. Conf.*, 2014, pp. 2510–2515.
[9] A. Greenberg, J. Hamilton, D.A. Maltz, and P. Patel, "The cost of a cloud: Research problems in data center networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 68–73, 2008.
[10] S. Pelley, D. Meisner, T. F Wenisch, and J. W VanGilder, "Understanding and abstracting total data center power," in *Proc. Workshop Energy-Efficient Des.*, 2009.
[11] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.
[12] D. Li, Y. Yu, W. He, K. Zheng, and B. He, "Willow: Saving data center network energy for network-limited flows," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 9, pp. 2610–2620, Sep. 2015.
[13] J. Shuja, K. Bilal, S. A. Madani, and S. U. Khan, "Data center energy efficient resource scheduling," *Cluster Comput.*, vol. 17, no. 4, pp. 1265–1277, 2014.
[14] L. Wang, F. Zhang, J. A. Aroca, A. V. Vasilakos, K. Zheng, C. Hou, D. Li, and Z. Liu, "Greendcn: A general framework for achieving energy efficiency in data center networks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 1, pp. 4–15, 2014.
[15] Q. Yi and S. Singh, "Minimizing energy consumption of fattree data center networks," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 42, no. 3, pp. 67–72, 2014.
[16] Í. Goiri, K. Le, T. D. Nguyen, J. Guitart, J. Torres, and R. Bianchini, "Greenhadoop: Leveraging green energy in data-processing frameworks," in *Proc. 7th ACM Eur. Conf. Comput. Syst.*, 2012, pp. 57–70.
[17] K.-K. Nguyen, M. Cheriet, M. Lemay, M. Savoie, and B. Ho, "Powering a data center network via renewable energy: A green testbed," *IEEE Internet Comput.*, vol. 17, no. 1, pp. 40–49, Jan./Feb. 2013.
[18] M. Arlitt, C. Bash, S. Blagodurov, Y. Chen, T. Christian, D. Gmach, C. Hyser, N. Kumari, Z. Liu, M. Marwah, et al. "Towards the design and operation of net-zero energy data centers," in *Proc. 13th IEEE Thermal Thermomech. Phenomena Electron. Syst.*, 2012, pp. 552–561.
[19] R. Krishnan and J. Pineda de Gyvez, "Low energy switch block for fpgas," in *Proc. 17th Int. Conf. VLSI Des.*, 2004, pp. 209–214.
[20] D. Meisner, B. T. Gold, and T. F. Wenisch, "Powernap: Eliminating server idle power," *ACM SigPlan Notices*, vol. 44, pp. 205–216, 2009.
[21] C. Lefurgy, K. Rajamani, F. Rawson, W. Felter, M. Kistler, and T. W. Keller, "Energy management for commercial servers," *Computer*, vol. 36, no. 12, pp. 39–48, 2003.
[22] K. K. Rangan, G.-Y. Wei, and D. Brooks, "Thread motion: Fine-grained power management for multi-core systems," in *Proc. ACM SIGARCH Comput. Archit. News*, vol. 37, pp. 302–313, 2009.
[23] E. N. Elnozahy, M. Kistler, and R. Rajamony, "Energy-efficient server clusters," in *Proc. 2nd Int. Conf. Power-Aware Comput. Syst.*, 2003, pp. 179–197.

[24] J. Leverich, M. Monchiero, V. Talwar, P. Ranganathan, and C. Kozyrakis, "Power management of datacenter workloads using per-core power gating," *Comput. Archit. Lett.*, vol. 8, no. 2, pp. 48–51, 2009.

[25] H. David, C. Fallin, E. Gorbatov, U. R. Hanebutte, and O. Mutlu, "Memory power management via dynamic voltage/frequency scaling," in *Proc. 8th ACM Int. Conf. Autonom. Comput.*, 2011, pp. 31–40.

[26] H. Abu-Libdeh, P. Costa, A. Rowstron, G. O'Shea, and A. Donnelly, "Symbiotic routing in future data centers," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, pp. 51–62, 2010.

[27] J.-Y. Shin, B. Wong, and E. G. Sirer, "Small-world datacenters," in *Proc. 2nd ACM Symp. Cloud Comput.*, 2011, p. 2.

[28] T. Wang, Z. Su, Y. Xia, B. Qin, and M. Hamdi, "Novacube: A low latency torus-based network architecture for data centers," in *Proc. IEEE Global Commun. Conf.*, 2014, pp. 2252–2257.

[29] V. Valancius, N. Laoutaris, L. Massoulié, C. Diot, and P. Rodriguez, "Greening the internet with nano data centers," in *Proc. 5th Int. Conf. Emerging Netw. Exp. Technol.*, 2009, pp. 37–48.

[30] A. Singla, A. Singh, K. Ramachandran, L. Xu, and Y. Zhang, "Proteus: A topology malleable data center network," in *Proc. 9th ACM SIGCOMM Workshop Hot Topics Netw.*, 2010, p. 8.

[31] L. Huang, Q. Jia, X. Wang, S. Yang, and B. Li, "Pcube: Improving power efficiency in data center networks," in *Proc. IEEE Int. Conf. Cloud Comput.*, 2011, pp. 65–72.

[32] U. Lee, I. Rimac, and V. Hilt, "Greening the internet with content-centric networking," in *Proc. 1st Int. Conf. Energy-Efficient Comput. Netw.*, 2010, pp. 179–182.

[33] K. Guan, G. Atkinson, D. C. Kilper, and E. Gulsen, "On the energy efficiency of content delivery architectures," in *Proc. IEEE Int. Conf. Commun. Workshops*, 2011, pp. 1–6.

[34] L. Liu, H. Wang, X. Liu, X. Jin, W. B. He, Q. B. Wang, and Y. Chen, "Greencloud: A new architecture for green data center," in *Proc. 6th Int. Conf. Ind. Session Autonom. Comput. Commun. Indus. Session*, 2009, pp. 29–38.

[35] F. Teng, D. Deng, L. Yu, and F. Magoules, "An energy-efficient vm placement in cloud datacenter," in *Proc. Int. Conf. High Perform. Comput. Commun., IEEE 6th Int. Symp. Cyberspace Safety Security, IEEE 11th Int. Conf. Embedded Softw. Syst.*, 2014, pp. 173–180.

[36] T. Yang, Y. C. Lee, and A. Y. Zomaya, "Energy-efficient data center networks planning with virtual machine placement and traffic configuration," in *Proc. IEEE 6th Int. Conf. Cloud Comput. Technol. Sci.*, 2014, pp. 284–291.

[37] V. Mann, A. Kumar, P. Dutta, and S. Kalyanaraman, "Vmflow: Leveraging vm mobility to reduce network power costs in data centers," in *Proc. 10th Int. IFIP TC 6 Conf. Netw.*, 2011, pp. 198–211.

[38] L. Wang, K. Wu, J. Xiao, and M. Hamdi, "Harnessing frequency domain for cooperative sensing and multi-channel contention in crahns," *IEEE Trans. Wireless Commun.*, vol. 13, no. 1, pp. 440–449, Jan. 2014.

[39] C. Frei and B. Faltings, "A dynamic hierarchy of intelligent agents for network management," in *Proc. Int. Agents Telecommun. Appl.*, 1998, pp. 1–16.

[40] R. M. Haralick and G. L. Elliott, "Increasing tree search efficiency for constraint satisfaction problems," *Artif. Intell.*, vol. 14, no. 3, pp. 263–313, 1980.

[41] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *Proc. ACM SIGCOMM Comput. Commun. Rev.*, 2008, vol. 38, pp. 63–74.

[42] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, "Bcube: A high performance, server-centric network architecture for modular data centers," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 63–74, 2009.

[43] T. Wang, Z. Su, Y. Xia, Y. Liu, J. Muppala, and M. Hamdi, "Sprintnet: A high performance server-centric network architecture for data centers," in *Proc. IEEE Int. Conf. Commun*, 2014, pp. 4005–4010.

[44] T. Wang, Z. Su, Y. Xia, J. Muppala, and M. Hamdi, "Designing efficient high performance server-centric data center network architecture," *Comput. Netw.*, vol. 79, pp. 283–296, 2015.

[45] P. Bodík, I. Menache, M. Chowdhury, P. Mani, D. A. Maltz, and I. Stoica, "Surviving failures in bandwidth-constrained datacenters," in *Proc. ACM SIGCOMM Conf. Appl., Technol., Archit. Protocols Comput. Commun.*, 2012, pp. 431–442.

[46] Openflow switch specification v1.3.0, 2012.

[47] Y. Liu and J. Muppala, "Dcnsim: A data center network simulator," in *Proc. 3rd Int. Workshop Data Center Perform.*, 2013.

[48] F. Howell and R. McNab, "SimJava: A discrete event simulation library for java," *Simul. Series*, vol. 30, pp. 51–56, 1998.

[49] T. Benson, A. Akella, and D.A. Maltz, "Network traffic characteristics of data centers in the wild," in *Proc. 10th Annu. Conf. Internet Meas.*, 2010, pp. 267–280.

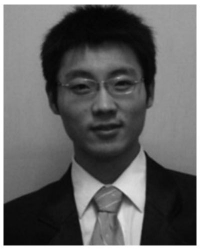
[50] G. Ananthanarayanan and R. H. Katz, "Greening the switch," in *Proc. Conf. Power Aware Comput. Syst.*, 2008, p. 7.

[51] [Online]. Available: www.remotepowerswitch.com, Remote power switch

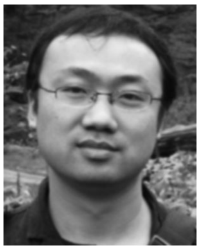
[52] [Online]. Available: http://www.aviosys.com/,Aviosys power switches

[53] R. J. Gallagher, K. H. Hoppe, A. J. Perri, M. S. Styduhar, J. M. Taylor, and B. W. Weidle, "Remote power control via data link, March 7 1995," US Patent 5,396,636, Mar. 7, 1995.

[54] P. Lieberman, Wake-on-lan technology, 2010.



Ting Wang received the bachelor sci degree from the University of Science and Technology Beijing, China, in 2008, the master eng degree from the Warsaw University of Technology, Poland, in 2011, and the PhD degree in computer science and engineering from the Hong Kong University of Science and Technology, Hong Kong SAR China, in 2015. From 02.2012 to 08.2012, he visited as a research assistant in the Institute of Computing Technology, Chinese Academy of Sciences, China. He is currently a research scientist in Alcatel Lucent Bell Labs, Shanghai, China. His research interests include data center networks, cloud computing, green computing, software defined network, and next generation networks. He is a student member of the IEEE.



Yu Xia received the PhD degree in computer science from Southwest Jiaotong University, China, in 2012. He was also a joint PhD student and a visiting scholar at the Polytechnic Institute of New York University, Brooklyn, from 2010 to 2012. After his graduation, he was a postdoctoral fellow at the Hong Kong University of Science and Technology, Hong Kong, starting from 2013. He is currently with the College of Computer Science, Sichuan Normal University. His research interests include high-performance packet switch designs, network protocols, and data center networks. He is a member of the IEEE.

Jogesh Muppala received the PhD degree in electrical engineering from Duke University, Durham, NC in 1991, the MS degree in computer engineering from The Center for Advanced Computer Studies, University of Southwestern Louisiana (now University of Louisiana at Lafayette), Lafayette, LA in 1987, and the BE degree in electronics and communication engineering from Osmania University, Hyderabad, India in 1985. He is currently an associate professor in the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. He is also currently serving as the program director for the master of science in information technology (MSc(IT)) program. He also served as the undergraduate studies director in the Computer Science Department from August 1999 to August 2001, and the chair of facilities committee in the department from September 2005 to August 2007. He was previously a member of the technical staff at Software Productivity Consortium (Herndon, Virginia) from 1991 to 1992, where he was involved in the development of modeling techniques for systems and software. While at Duke University, he participated in the development of two modeling tools, the Stochastic Petri Net Package (SPNP) and the symbolic Hierarchical Automated Reliability and Performance Evaluator (SHARPE), both of which are being used in several universities and industry in the USA. He cofounded the International Workshop on Dependability of Clouds, Data Centers, and Virtual Machine Technology (DCDV). He was also the program cochair for the 1999 Pacific Rim International Symposium on Dependable Computing held in Hong Kong in December 1999. He also cofounded and organized The 1st Asia-Pacific Workshop on Embedded System Education and Research (APESER). He has also served on program committees of many international conferences. He is a senior member of the IEEE.

Mounir Hamdi received the BS degree in electrical engineering-computer engineering minor (with distinction) from the University of Louisiana in 1985, and the MS and PhD degrees in electrical engineering from the University of Pittsburgh in 1987 and 1991, respectively. He is a chair professor at the Hong Kong University of Science and Technology, and was the head of the Department of Computer Science and Engineering. Now he is the dean of the College of Science, Engineering and Technology at the Hamad Bin Khalifa University, Qatar. He is/was on the editorial board of various prestigious journals and magazines including *IEEE Transactions on Communications*, *IEEE Communication Magazine*, *Computer Networks*, *Wireless Communications and Mobile Computing*, and *Parallel Computing* as well as a guest editor of *IEEE Communications Magazine*, guest editor-in-chief of two special issues of *IEEE Journal on Selected Areas of Communications*, and a guest editor of *Optical Networks Magazine*. He has chaired more than 20 international conferences and workshops including The IEEE International High Performance Switching and Routing Conference, the IEEE GLOBECOM/ICC Optical networking workshop, the IEEE ICC High-speed Access Workshop, and the IEEE IPPS HiNets Workshop, and has been on the program committees of more than 200 international conferences and workshops. He was the chair of IEEE Communications Society Technical Committee on Transmissions, Access and Optical Systems, and vice-chair of the Optical Networking Technical Committee, as well as member of the ComSoc technical activities council. He received the best paper award at the IEEE International Conference on Communications in 2009 and the IEEE International Conference on Information and Networking in 1998. He also supervised the best PhD paper award among all universities in Hong Kong. He is a fellow of the IEEE for contributions to design and analysis of high-speed packet switching.