Wang *et al. Journal of Cloud Computing: Advances, Systems and Applications* (2015) 4:9
DOI 10.1186/s13677-015-0035-7

Journal of Cloud Computing
a SpringerOpen Journal

RESEARCH Open Access

Towards bandwidth guaranteed energy efficient data center networking

Ting Wang[1*], Bo Qin[1], Zhiyang Su[1], Yu Xia[1], Mounir Hamdi[1,2], Sebti Foufou[3] and Ridha Hamila[3]

Abstract

The data center network connecting the servers in a data center plays a crucial role in orchestrating the infrastructure to deliver peak performance to users. In order to meet high performance and reliability requirements, the data center network is usually constructed of a massive number of network devices and links to achieve 1:1 oversubscription for peak workload. However, traffic rarely ever hits the peak capacity in practice and the links are underutilized most of the time, which results in an enormous waste of energy. Therefore, aiming to achieve an energy proportional data center network without compromising throughput and fault tolerance too much, in this paper we propose two efficient schemes from the perspective of resource allocation, routing and flow scheduling. We mathematically formulate the energy optimization problem as a multi-commodity minimum cost flow problem, and prove its NP-hardness. Then we propose a heuristic solution with high computational efficiency by applying an AI resource abstraction technique. Additionally, we design a practical topology-based solution with the benefit of Random Packet Spraying consistent with multipath routing protocols. Both simulations and theoretical analysis have been conducted to demonstrate the feasibility and convincing performance of our frameworks.

Keywords: Energy efficiency; Data center network; Energy-aware routing; Bandwidth allocation

Introduction

The data center, as a centralized repository clustering a large number of servers, has become home to essential large-scale computation, storage and Internet-based applications which provide various services like search, social networking, e-mails, gaming, cloud computing, and so on [1,2]. In order to provide high performance service with strong reliability to users, the data center network (DCN) architectures are usually over-provisioned and constructed aggressively with large number of switches and links to achieve high-capacity and high fault tolerance [3]. However, the research [4,5] shows that, in practice the average link utilization in different data centers ranges only between 5% and 25% and varies largely between daytime and night. This reveals that most network devices and links stay idle or underutilized most of the time, but an idle device consumes up to 90% of the power consumed at full loads [6], which leads to a great waste of energy. Apart from the energy wasted due to over-richly network interconnections, traditional non-energy-aware routing algorithms (like shortest path routing or its variations) can also lead to poor link utilization or even congestion, which worsens the situation.

According to current research findings [7,8], the power consumed by servers and infrastructure (i.e. power distribution and cooling) accounts for over 70% of overall power, while the network consumes around 15% 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 grows correspondingly higher. As illustrated in [4], suppose the servers are totally energy-proportional, when the data center is 15% utilized (servers and network), then the network will consume up to 50% of overall power. Even if the servers are not energy-proportional, with 15% traffic load, making the network proportional still can save as much as 975 KW (for a data center with 10,000 servers) [4]. 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.

*Correspondence: twangah@cse.ust.hk
[1] Hong Kong University of Science and Technology, Hong Kong SAR, China
Full list of author information is available at the end of the article

Based on the above observations, this paper aims to achieve a bandwidth guaranteed energy proportional data center network, where the amount of power consumed by the network is proportional to the actual traffic workload. The key principle behind this approach is that most of time the traffic can be merged and satisfied by just a certain subset of network devices and links, and the remaining ones can be put onto sleep mode or powered off for the sake of power conservation. With this goal, we propose two efficient green frameworks from the perspective of bandwidth allocation and flow scheduling. However, the bandwidth in a data center is a scarce resource [9,10], and the energy-aware routing problem is NP-hard, which is proved in Section "Problem statement". Besides, the time complexity of dynamically computing a feasible network subset to meet the traffic demands is horrible and unmanageable due to the exponential number of nodes and routes. In order to address these critical issues, derived from Artificial Intelligence our first framework employs a resource abstraction technique named Blocking Island (BI) and some well designed heuristic algorithms to efficiently reduce the searching space and significantly decreases the computation and time complexity. This framework can be applied in any arbitrary data center network topology. In the second framework, we put forward a topology-based energy-aware algorithm by computing a network subset and adopting one recently proposed multipath routing mechanism RPS [11] for flow scheduling and packet transmission.

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

1.) We formulate the energy optimization problem in DCNs mathematically and prove its NP-hardness.
2.) We propose two efficient general frameworks, which provide efficient solutions from the perspective of bandwidth allocation, routing and flow scheduling.
3.) To the best of our knowledge, we are the first to employ the AI model - Blocking Island Paradigm into data centers for resource allocation to achieve power savings.
4.) We conduct extensive simulations to evaluate and demonstrate the performance of our frameworks 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 "Related work". Then we formulate the energy optimization problem and prove its NP-hardness in Section "Problem statement". Afterwards, Blocking Island Paradigm is briefly reviewed in Section "Blocking island paradigm". Then we propose two energy-aware heuristic schemes in Section "Energy-aware heuristic schemes", followed by the evaluations and simulation results in Section "System evaluation". Finally, Section "Conclusion" concludes the paper.

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. Apart from the works on green/renewable resources [12-14], low-power hardware [15-18], energy-efficient network architecture [4,19-23], and network virtualization techniques (VM migration and placement optimization) [24,25], there are also many network-level proposals, which focus on traffic consolidation. The typical representatives include ElasticTree [5], and Energy-aware Routing Model [26].

ElasticTree is a network-wide energy optimizer, which consists of three logical modules – Optimizer, Routing, and Power Control. Once the Optimizer outputs a set of active components, Power Control toggles the power states of ports, linecards, and entire switches, while Routing chooses paths for all flows, then pushes routes into the network. The authors proposed three types of optimizers with different quality of solution and scalabilities. The formal method achieves the best solution but is computationally very expensive and does not scale beyond a 1000-node sized data center. The greedy bin-packer ends up with suboptimal solutions as with any greedy approach but is much faster than the formal method. Lastly, the topology aware heuristic needs the smallest amount of computation time but the quality of its solution is inferior to both the greedy and formal method.

Energy-aware Routing Model is also a network-wide approach, which aims to compute the routing for a given traffic matrix, so that as few switches are involved as possible to meet a predefined performance (throughput) threshold. The basic idea is that: Firstly, 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, they power off the switches not involved in the routing. However, 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.

Problem statement

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.

Problem formulation

From the perspective of routing, the crucial resource to manage in a data center is the bandwidth. To describe the bandwidth allocation problem in a data center network $G = (V, E)$, we define the constraints as follows:

1. 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));
2. Reliability requirement—each demand should be assigned FT number of backup routes (i.e., Constraint (2));
3. 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));
4. 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:

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

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

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

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

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

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

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

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 $Y_{ji}^{(k)} \in \{0, 1\}$.

NP-hardness

For the MCF problem described above, we change to its corresponding decision problem (DMCF): Is there any set of rounting 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 [27] 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 in G, 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. □

Blocking island paradigm

Derived from Artificial Intelligence, BI model provides an efficient way to represent the availability of network resources (especially bandwidth) at different levels of abstraction. The Blocking Island is defined as: *A β-Blocking Island (β-BI) 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* [28]. The key idea of BI is to abstract the original network graph into a hierarchy tree containing available bandwidth information. As shown in Figure 1, N_1 is a 50-BI for node *S3*.

BI has several fundamental properties which are very useful in routing decidability. Here we list some of the most important ones without proof.

- **Unicity:** Each node has one unique β-BI. 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.
- **Route Location:** The links of a route with β available bandwidth are all in the β-BI of its endpoints.
- **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.

The obtained BIs can be used to construct the Blocking Island Graph (BIG), which is a graph abstraction of the entire available network resources. The BIG can be

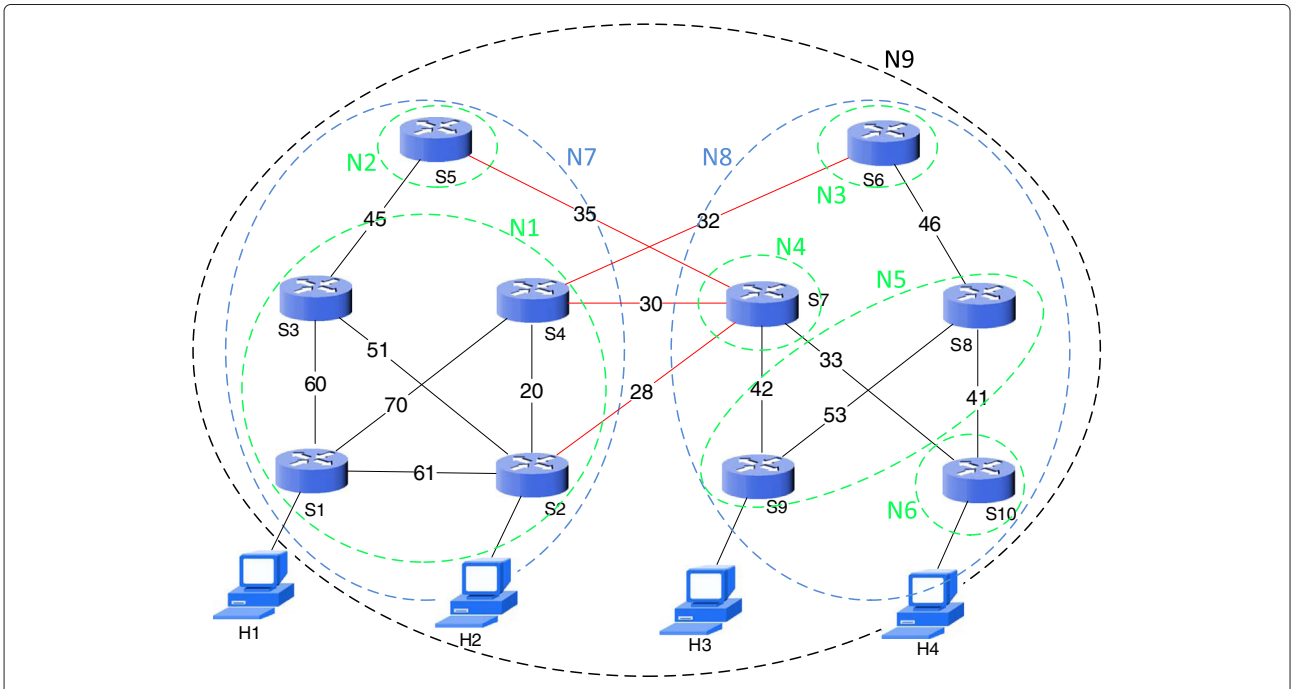

Figure 1 An example of Blocking Island Graph, in which N_1 – N_6 are 50-BIs, N_7 – N_8 are 40-BIs, and N_9 is 30-BI. The red lines are critical links between two 40-BIs. The weights on the links are their available bandwidth.

denoted as G = (*S*, *L*), where *S* indicates the set of all different β-BIs while *L* denotes the critical links between BIs. Figure 1 gives an example of BIG. We can further construct a recursive decomposition of BIGs in decreasing order of demands (βs), and the lowest level has the largest β. This layered BIG structure is named as Blocking Island Hierarchy (BIH). It can be used to identify all bottlenecks, i.e. critical links, of the network. The BIH can also be viewed as an abstraction tree when taking the father-child relation into consideration. An example of BIH is illustrated in Figure 2. The leaves of the BIH tree are the network nodes, and the other vertices denote the abstract BIs. This abstraction tree can reflect the real-time state of the available network bandwidth.

Energy-aware heuristic schemes

In this section, we propose two heuristic solutions to the energy optimization problem formulated in Section "Problem statement", one of which is based on Blocking Island Paradigm while the other one is topology based. The BI based heuristic achieves bandwidth guaranteed green networks and enjoys low computation complexity with the help of Blocking Island Paradigm for resource allocation. The topology based heuristic holds the best scalability (*O(N)*) in computation time growth where *N* is the number of servers, but the resulting solution is not as good as the BI based solution. Comparatively, the BI based heuristic provides a more attractive and practical option.

Power conservation strategy

Most existing proposals apply device-level power conservation strategy, which intends to switch off the entire device (router/switch) including fixed overheads (like fans, linecards, transceivers, etc.) only when all ports on the device are idle. This means even if only one port has data to transfer, the device (including idle ports) should be kept alive all the time. Comparatively, in our energy-aware heuristic schemes we apply the component-level strategy, which intends to power down the unused ports, and switch off the linecard if all the ports on this linecard are idle or disabled. If all linecards on a device are idle then to power off the entire device. Clearly, the component-level strategy achieves the most power savings.

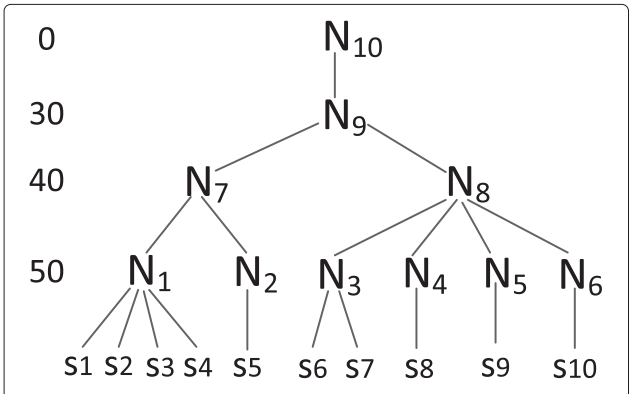

Figure 2 Abstraction tree of BIH in Figure 1, and $N_1$0 indicates 0-BI.

Consequently, the total power consumption of a switch Ω_{switch} can be computed as below:

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

where Ω_s and Ω_p are the same as described in Section "Problem statement", and N_p denotes the number of active ports on the switch.

BI-based heuristic scheme

As proved in Section "Problem statement", energy-aware bandwidth allocation and routing is an NP-hard problem with a high complexity due to the exponential number of nodes and routes. In response to this issue, we present an approach that applies the Blocking Island Paradigm to solve the problem efficiently with much lower and more manageable complexity. The BI-based Heuristic Scheme (BHS) can also be regarded as a bandwidth allocation scheme, which can help achieve a bandwidth guaranteed green data center network. Several key intuitions behind BHS can be summarized as below:

- Drawing support from BI model to guide the bandwidth allocation for the traffic demands in the most advantageous order.
- Using the energy-aware routing algorithm to compute the most beneficial routes for these allocated demands.
- Switching off devices that are not involved in the final routings for power conservation.

Bandwidth allocation mechanism

In line with the data center policy which requires fast response to the request, the *Route Existence* property of BI enables much faster decisions in determining whether a request can be satisfied just by checking whether both the endpoints are in the same β-BI, while traditional routing algorithms have to compute the routes before deciding the route's existence. For example, if we want to assign a path for a traffic demand (*H1, H3, 50*) in the network as shown in Figure 1, then we can immediately know that the route does not exist since *H1* and *H3* are not in the same 50-BI without any effort to search the whole network space and compute the routes. Moreover, if we need to find a path for (*H1, H2, 50*), then the search space can be reduced from the whole network to only {*S1, S2, S3, S4*}, which leads to a significant improvement in the efficiency of computation and bandwidth allocation. The unique β-BI for a given node *x* can be obtained by a simple greedy algorithm

(as depicted in Algorithm 1) whose complexity is linear in O(L), where L denotes the number of links. Additionally, querying two nodes in the same BI experiences a complexity of just *O(1)* since only two hashing operations are needed.

Algorithm 1 Construct β-BI

```
1: function ConstructBI(N = {V, E}, β)
2:     L ← {∅}                                  ▷ L: Result β-BI list
3:     for all v in V do
4:         if not visited(v) then
5:             I ← ConstructBIFromNode(N, β, v)
6:             L ← L.Add(I)
7:         end if
8:     end for
9:     return L
10: end function

11: function ConstructBIFromNode(N = {V, E}, β, x)
12:     I ← {x}                                  ▷ I: Result β-BI
13:     S ← {links incident to x and weight ≥ β}  ▷ S: stack
14:     while S ≠ ∅ do
15:         l ← pop(S)
16:         e ← another endpoint of l
17:         if e ∉ I and weight(l) ≥ β then
18:             I ← I ∪ {e}
19:             S ← S ∪ {links incident to e and weight ≥ β}
20:         end if
21:     end while
22:     return I
23: end function
```

As a known NP-hard MCF problem, it cannot be guaranteed to find an assignment to satisfy all the flows for all kinds of traffic matrix all the time. How to select the next demand to allocate bandwidth has a great impact on the future allocation success ratio, and also affects the search efficiency. There are some static methods for addressing this kind of MCF problem or constraint satisfaction problem (CSP) [29], such as first-fail principle based technique, or first selecting the largest demand. However, these static techniques are not suitable to be directly applied in the data center network which requires a more dynamic and efficient bandwidth allocation mechanism. In addition, considering the data center's own particular characteristics, these traditional static techniques do not take the mean flow completion time and deadline into consideration as well. In our approach, these concerns are effectively resolved by exploiting the advantages of Blocking Island Paradigm.

The BI-based bandwidth allocation mechanism (BAM) is mainly responsible for deciding which traffic demand should be chosen to allocate with its required bandwidth. In order to achieve a higher success ratio of bandwidth allocation and higher computation efficiency, BAM selects the unallocated traffic demands strictly following the principles as below.

(i) It firstly choose the demand of which the lowest common father (LCF) of the demand's endpoints in the BIH tree is highest. The intuition behind this principle is to first allocate the more constrained demands, which follows the fail-first principle.
(ii) If there are multiple candidate demands after the first step, then the Shortest Demand First (SDF) principle is applied, which aims to meet as many deadlines as possible and reduce the mean flow completion time. The shortest demand indicates the demand whose expected completion time is minimum $\left(\text{i.e. } min\left\{\frac{flow\ size}{required\ bandwidth}\right\}\right)$ where the *flow size* and *required bandwidth* are provided by the application layer [30,31].
(iii) In case there are still two or more satisfied demands, then the demand with the highest bandwidth requirement is preferentially selected. This criterion, apart from implying a near deadline flow, also follows the fail-first principle, where more bandwidth allocation more likely cause BI splittings and thus hinder any future allocation of other demands.
(iv) Finally, we randomly select one demand from the output of step (iii).

The demand selection rules for bandwidth allocation not only decreases the computation complexity and increases the search efficiency, but also takes the flow deadline into consideration. Moreover, they can also increase the success ratio of bandwidth allocation, which targets at simultaneously satisfying as many flows of the traffic matrix as possible. If some demands can not be allocated currently, they will be queued for a certain period until some allocated flows expire or departure so that some resources are released for further allocation.

Energy-aware routing

After selecting the most beneficial traffic demand from the traffic matrix by applying the BI-based bandwidth allocation mechanism, the energy-aware routing is designed to assign one best route for each selected demand request. However, the searching domain is too large and the valid route set is too time-consuming to be computed using a traditional routing algorithm. In order to improve the search efficiency and increase the success

ratio of bandwidth allocation, several route selection criterions are carefully customized for our energy-aware routing algorithm (ERA) as follows.

(i) The traffic should be aggregated together to the greatest extent, which would allow us to conserve more energy in a tighter data center network.
(ii) The route should use as few critical links (inter-BI links) as possible, which aims to decrease the failure ratio of future allocations and also reduce the computation cost caused by splitting/merging BIs.
(iii) The route should use as few network devices as possible, which prefer to choose the shortest path.
(iv) The current allocation should impact on the future allocation as little as possible.

Guided by these rules, the ERA assigns a route for each requested demand in an efficient and dynamic way based on the current network status. Initially, ERA searches the lowest-level BI where the two endpoints of the requested demand are located, and generates a set of feasible candidate routes. For example, as shown in Figure 1 the lowest level for demand (*s1, s4, 45*) is 50-BI N_1. This procedure aims to meet the rule i and ii, which tries to aggregate the flows into the same subnet (lowest BI) and use as few critical links as possible. Afterwards, sort these candidate routes by the number of their induced BI splittings, and choose the route(s) that cause fewest BI splittings. This step complies with rule ii and iv, which takes the computation cost and future allocation into consideration. If there are more than one such route, then choose the shortest route which tries to meet the objective of rule iii. In case there are still multiple routes, then choose the route with the maximum number of flows which can contribute to the network traffic aggregation. Finally, we randomly choose one route or just choose the first route from the sorted candidate routes. The power-aware routing procedure terminates as long as the output of the above five procedures is unique, and allocates the best route with the required bandwidth to the current demand.

Reliability satisfaction

Admittedly, the energy conservation in the way of powering off devices sacrifices the network fault tolerance, which is an inevitable conflict between them. In order to improve the robustness of the network, we need to add additional number of available backup routes according to the reliability requirements as illustrated in Constraint (3). The selection of backup routes applies the shortest-path routing algorithm other than following the aforementioned multiple route selection rules. This strategy means to reserve as few devices as possible to meet the requirements of fault tolerance. From another perspective, as indicated in [32] the switches are fairly reliable (only 5% failure rates for ToR switches per year), hence it is not so wise to sacrifice a great deal (network resources, computation time, energy, etc.) for a small probability event. Therefore, the shortest-path routing algorithm is well suited and adequate for the backup route selection.

The whole procedure of the BI-based heuristic scheme is illustrated in Figure 3. The input includes network topology, traffic matrix with the required bandwidth and the reliability requirement. The outputs are expected to be a subset of original network topology and a set of routing paths taken by flow demands with satisfied bandwidths. Firstly, based on the network topology BHS generates multiple levels of BIs according to the current available link capacities and further constructs BIG and BIH. Then, on the basis of BIH the system computes and allocates the best routes associated with a required bandwidth to each demand, applying the bandwidth allocation mechanism and energy-aware routing algorithm. Afterwards, according to the reliability requirement, a certain number of backup routes are added to guarantee the network's fault tolerance. Finally, all the ports, linecards, or switches, that are not involved in the final routings, are put into sleep mode or switched off for the sake of energy conservation.

Analysis of complexity

As aforementioned, the complexity of constructing a β-BI is $O(L)$, where L denotes the number of links. The route searching or routing decidability experiences a complexity of $O(1)$. The Blocking Island Hierarchy (BIH) reflects the real-time state of the available network bandwidth, and it needs to be updated when the link state changes. Yet we only need to update the BIs, which are involved in allocating or deallocating bandwidths, by means of splitting or merging BIs. This means there is no need to compute the whole BIH again. The complexity of updating the BIH is O(rl), where r is the number of different resource requirements (β) and l indicates the number of involved links.

Topology-based heuristic scheme

In this subsection, the topology-based heuristic scheme (THS) is described based on the multi-rooted Fat-Tree topology [33] (as shown in Figure 4), yet the idea can be extended to any other tree-like topologies. THS needs to resolve two issues:

- How many switches and ports should be sufficient to support the network traffic.
- How to distribute the traffic flows among the calculated network subset and achieve high network utilization.

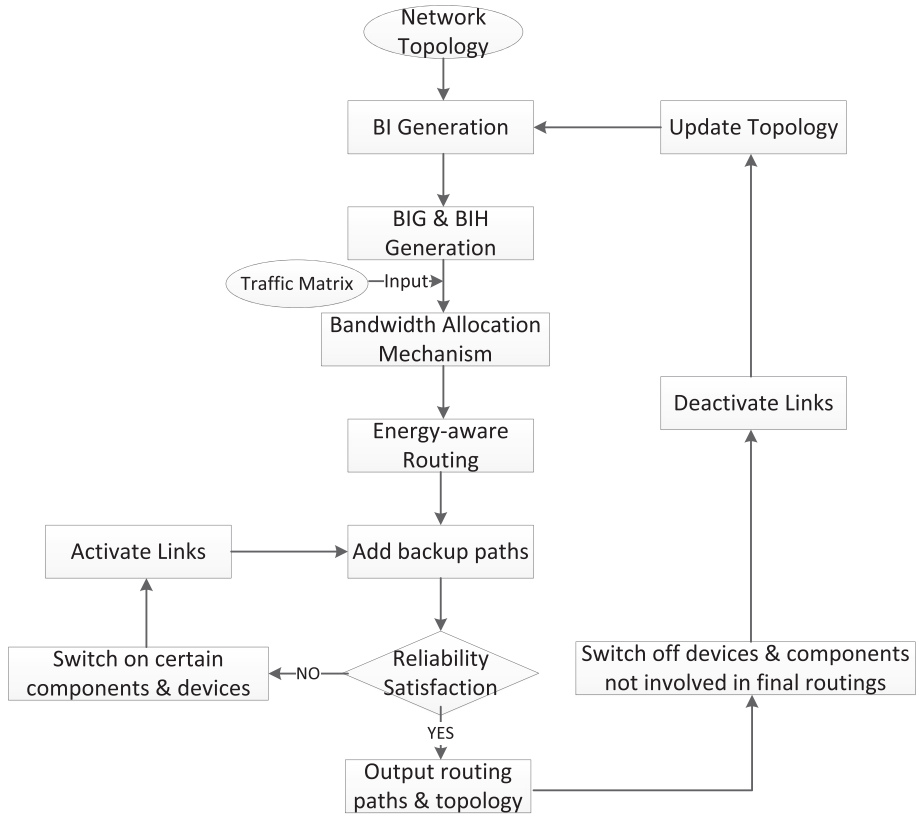

Figure 3 The working procedure of BI-based Heuristic Scheme.

Calculate the minimum network subset

In order to maximize power conservation, we need to only keep the required networking capacity available by merging the traffic flows and switch off idle devices. The minimum network subnet should be dynamically computed according to the statistics of traffic demands in runtime. The port statistics and switch status are collected by the centralized controller from OpenFlow enabled switches through the OpenFlow secure channel. It can accurately and directly obtain the statistics of the traffic matrix by using the built-in features (bytes, packet counters, etc.) for active flows kept in OpenFlow switches. In order to deal with the single point failure of the controller, THS provides multiple controllers with different roles (OFPCR_ROLE_EQUAL ,OFPCR_ROLE_MASTER and OFPCR_ROLE_SLAVE) to guarantee the robustness of the system (the same as specified in [34]).

We assume that k-port switches are used in the Fat-Tree topology. Consequently, there are k Pods, $\frac{k}{2}$ edge/aggregation switches in each Pod, and $\frac{k^2}{2}$

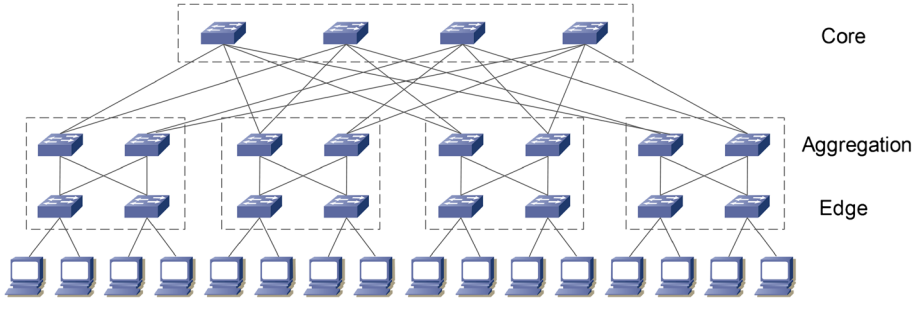

Figure 4 An example of Fat-Tree topology.

edge/aggregation switches (Equation 9) in total. The number of core switches on the network is $\frac{k^2}{4}$. Furthermore, there are $\frac{k^2}{4}$ equal-cost paths between any given pair of hosts in different Pods.

First we provide definitions of several notations used in calculating the required network subset: NS_i^{Edge} and NS_i^{Agg} denote the number of edge switches and aggregation switches in Pod i that should be activated (Equation 10 and 11), respectively, NP_i^{Edge} and NP_i^{Agg} indicate the number of active ports on each activated edge switch and aggregation switch in Pod i (Equation 13 and 14), NS^{Core} represents the number of core switches that should be turned on (Equation 12), and NP^{Core} means the number of active ports on each activated core switch (Equation 15). Besides, we define $DEdge_{ij}^{up}$, $DAgg_{ij}^{up}$ and $DCore_{ij}^{down}$ as the up-traffic of edge switch j in Pod i, the up-traffic of aggregation switch j in Pod i, and the down-traffic sending from core switch j to Pod i, respectively. Lastly, for the sake of simplicity we assume all links have the same link capacity C. Then the minimum subset of switches and ports is calculated according to the realtime traffic demands as follows:

$$NS_i^{Edge} = \frac{k}{2} \tag{9}$$

$$T = \max\left\{ \lceil \frac{\sum_{j=1}^{\frac{k}{2}} DEdge_{ij}^{up}}{C*\frac{k}{2}} \rceil, \lceil \frac{\sum_{j=1}^{(\frac{k}{2})^2} DCore_{ij}^{down}}{C*\frac{k}{2}} \rceil, 1 \right\} \tag{10}$$

$$NS_i^{Agg} = T \tag{11}$$

$$NS^{Core} = \max\left\{ \max_{i\in[1,k]} \left\{ \lceil \frac{\sum_{j=1}^{\frac{k}{2}} DAgg_{ij}^{up}}{C*k} \rceil \right\}, 1 \right\} \tag{12}$$

$$NP_i^{Edge} = \frac{k}{2} + NS_i^{Agg} \tag{13}$$

$$NP_i^{Agg} = \frac{k}{2} + NS^{Core} \tag{14}$$

$$NP^{Core} = k; \tag{15}$$

The relative notations are summarized as in Table 1. Importantly, the critical switches that may cause network disconnections cannot be powered off and each server should be guaranteed reachable at any time. Hence, all edge switches (i.e. ToR switches connecting servers) should stay alive at all times as expressed in Equation 9.

Traffic distribution using multipath routing

After obtaining the capable network subset, we need to distribute the traffic flows evenly among the subset. Since the size of each flow demand varies much, so the traffics that should be able to fill the network subset cannot fully utilize the network with unexpected low throughput when single path routing is applied. As aforementioned, the Fat-Tree DCN holds many equal-cost paths between any pair of servers. Therefore, we can divide the total flow demands by the switch capacity and distribute them evenly using multipath routing algorithms by splitting each flow into multiple sub-flows. However, the existing flow-level multipath routing relying on per-flow static hashing, like ECMP-based MPTCP [35], still cannot guarantee the traffic will be evenly distributed, which would lead to substantial bandwidth loss and significant load imbalance. Against this kind of bin-packing problem, one may apply best fit decreasing (BFD) or first fit decreasing (FFD) heuristic algorithms to mitigate this issue, but still not achieve perfect network utilization and bisection bandwidth. Intuitively, the best method is to distribute all packets evenly among all equal cost paths using packet-level multipath routing.

Based on some careful studies and extensive experiments, recently A. Dixit et al. proposed a packet-level traffic splitting scheme named RPS (Random Packet Spraying) [11] for data center networks. RPS achieves near ideal load balance and network utilization, and causes little packet reordering by exploiting the symmetry of the multi-rooted tree topologies. Noticing that THS's strategy of powering off switches barely affects the symmetry of the network since we always choose the leftmost switches. Therefore, after obtaining the required subset and adding FT-redundancy, we directly borrow the packet-level RPS multipath routing scheme to spread all flows equally among multiple different equal cost shortest paths. Then

Table 1 The number of active switches and ports on the network

Devices	Number of active Switches in pod *i*	Total number of active Switches on the network	Uplink ports	Downlink ports	Number of active ports per Switch in pod *i*
Edge switch	$\frac{k}{2}$	$\frac{k^2}{2}$	NS_i^{Agg}	$\frac{k}{2}$	$NP_i^{Edge} = \frac{k}{2} + NS_i^{Agg}$
Agg switch	NS_i^{Agg}	$\sum_i^k NS_i^{Agg}$	NS^{Core}	$\frac{k}{2}$	$NP_i^{Agg} = \frac{k}{2} + NS^{Core}$
Core switch	/	NS^{Core}	/	/	$NP^{Core} = k$

switch off the switches and ports which are not involved in final subset and update the network topology. The whole THS procedure is depicted in detail in Algorithm 2.

Algorithm 2 Topology-based Heuristic Algorithm

Require:

(1) DCN topology $G = (V, E)$ using k-port switches;
(2) Traffic matrix $TM = \{(s,\ t,\ d_{st});\ s, t \in V\}$;
(3) Reliability requirement $R(d_{st})$ $(\geq FT)$;

Ensure:

(1) A set of edge switches, aggregation switches, and core switches satisfying all traffic demands;
(2) The updated network topology;
for $t \in [t_m, t_n]$ **do**
    Obtain the traffic demands of each switch on the network;
    for $i \in [1, k]$ **do**
        Calculate $\sum_{j=1}^{\frac{k}{2}} DEdge_{ij}^{up}$ and $\sum_{j=1}^{(\frac{k}{2})^2} DCore_{ij}^{down}$.
        Calculate NS_i^{Agg}, NP_i^{Edge} and NP_i^{Agg}.
        Activate the leftmost NS_i^{Agg} aggregation switches with NP_i^{Agg} ports on each switch in Pod i.
        Activate all $\frac{k}{2}$ edge switches with NP_i^{Edge} active ports on each switch.
    end for
    Calculate $\sum_{j=1}^{\frac{k}{2}} DAgg_{ij}^{up}$ in each Pod.
    Calculate NS^{Core} and activate NS^{Core} core switches.
    Activate redundant switches forming FT minimum spanning trees to meet the requirement of fault tolerance.
    Spread all traffic flows evenly into the network using RPS multipath routing algorithm.
    Switch off the unused switches and ports.
    Update the network topology $G' = (V', E')$.
end for

System evaluation

Simulation overview

In order to evaluate the performance and effectiveness of our proposed approaches (BHS and THS) to the power optimization problem, in this section we implement the blocking island paradigm and two heuristic schemes in the DCNSim simulator [36]. DCNSim can simulate several data center network topologies and compute many metrics, such as the power consumption, aggregate bottleneck throughput, network latency, average path length, fault-tolerance, and statistics of device status. Without loss of generality, all simulations in this section are conducted based on Fat-Tree topology, and all the links are capable of bidirectional communications, where the unidirectional link bandwidth is set to be 1 GBps. The default MTU of a link is 1500 bytes, the default packet size is one MTU, and the default buffer size of each switch is 10 MTU. The default processing time for a packet at a node is 10 μs while the default propagation delay of a link is 5 μs, and the TTL of a packet is set to be 128. The time interval between two packets from the same source to the same destination is set to be 5 ms as default.

Evaluation indicator

The traditional always-on strategy is used as the base line to be compared with BHS and THS in the percentage of power savings, shown as below,

$$PEC = 100\% - \frac{P_{BHS/THS}}{P_{always-on}} * 100\%, \tag{16}$$

where PEC denotes the percentage of energy conservation, $P_{BHS/THS}$ indicates the power consumed by BHS or THS, and $P_{always-on}$ represents the power consumed by the traditional always-on strategy.

To calculate the power consumption, we use the real power consumption data of Cisco Nexus 3048 Data Center Switch. According to its switch data sheet [37], the typical operating power consumption of a Nexus 3048 switch is 120 watts at 100% loads, and powering off one port of the switch saves around 1.5 watts. Moreover, reducing the power consumed by the network can also result in cooling power savings proportionally, though this part of power savings is not taken into any calculation in this paper.

Network traffic matrix

Aside from the power-aware routing and resource allocation mechanisms which mainly determine how much power can be conserved, the traffic pattern also has a great impact on power savings and network performance. In data center networks, there are several typical types of traffic patterns, including One-to-One, One-to-Many, and All-to-All. In this section, all the simulations are conducted by applying the All-to-All traffic pattern, which simulates the most intensive network activities and can evaluate the guaranteed performance under the most rigorous case. Furthermore, according to the findings in [38] 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. Here, the Exponential Flow Mode is applied to determine the distribution of packet inter-arrival times.

Simulation results

This subsection evaluates the overall performance of BHS and THS. The primary simulation results show that achieving 20% to 60% of power savings is feasible, and it varies under different network conditions (traffic loads, network scales, traffic patterns, etc.) and different reliability requirements.

System reliability

As aforementioned, achieving energy conservation by powering off network devices might sacrifice system's reliability, where a reasonable trade-off can be made according to the actual needs. For example, one possible way is to take $Max\{FT + \alpha * PEC\}$, where higher weight factor α gives more weight to achieving better *PEC* at the cost of less improvement to *FT* as illustrated in Figure 5. The value of α can be decided by the network administrator according to the different requirements of reliability and power savings. Figure 6 presents the simulation results of power savings by applying THS and BHS under different fault tolerance (*FT*) levels in a 128-server Fat-Tree topology, where the abscissa axis means the number of backup routes (in BHS) or minimum spanning trees (in THS). Several careful observations can be derived from this figure: firstly, the component-level power conservation strategy achieves more power savings than the device-level strategy using either heuristic schemes; secondly, BHS performs better than THS in power savings, and it earns more advantages for higher fault tolerance levels; thirdly, lower fault tolerance level results in more power savings, and around 55% of power savings can be achieved for $FT = 1$.

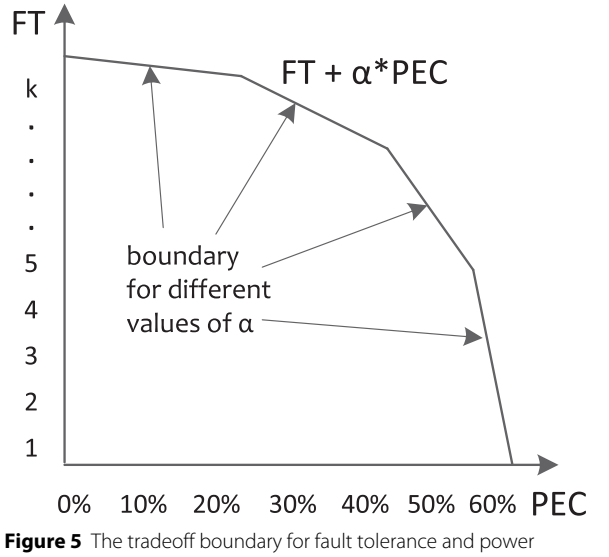

Figure 5 The tradeoff boundary for fault tolerance and power savings.

Network latency

The queuing delay, which plays a dominant role in the resulting network latency, is usually caused by the network congestion because of exceeding the network capacity. However, guaranteeing the required bandwidth for each flow's transmission could help mitigate network congestion and decrease queuing delay. Here we use the average global packet lifetime (the time from packet's generation to the arrival at its destination) to measure the overall network latency, and the always-on scheme applying two-level routing tables [33] is used as the baseline to be compared with BHS and THS. With the benefit of BI-based bandwidth allocation mechanism and carefully designed routing algorithm, BHS implements a bandwidth guaranteed network and achieves even lower network latency than the original always-on scheme using its original two-level routing, as illustrated in Figure 7. However, THS receives higher network latency without any bandwidth-guarantee mechanism. Admittedly, the packet level multipath routing algorithm applied in THS could help fill the computed network subset perfectly, but it also may incur the incast problem at the last hop, resulting in queuing delays caused by traffic collisions or even packet loss, where packet retransmission further delays the flow completion time. However, this can be mitigated by many already existing techniques dealing with incast issues and reducing flow completion time, which is beyond the scope of this paper.

Network scales

Our efficient heuristic schemes have a good scalablity enabling that the system can easily adapt to larger sized data centers with good performance. Figure 8 exhibits the performance of BHS and THS in power savings by applying component-level strategy under various network loads in 128-server, 250-server and 2000-server sized Fat-Tree DCNs. The fault tolerance level is set to be $FT = 2$. The simulation results show that both BHS and THS achieve more power savings for larger sized network and lower network load, and BHS gains a better performance than THS for all three cases. Either BHS or THS conserves no energy at full network loads.

Computation efficiency

The traditional routing algorithm suffers a bad exponential time complexity due to the huge search space. However, utilizing blocking island to guide the search and bandwidth allocation, BHS achieves a much lower computation complexity. Comparatively, THS only needs to compute the capable minimum network subset based on the sum of traffics without doing any bandwidth allocations for each particular flow, thus THS scales at roughly O(N) where N is the number of servers. Figure 9 presents the time costs for computing different number

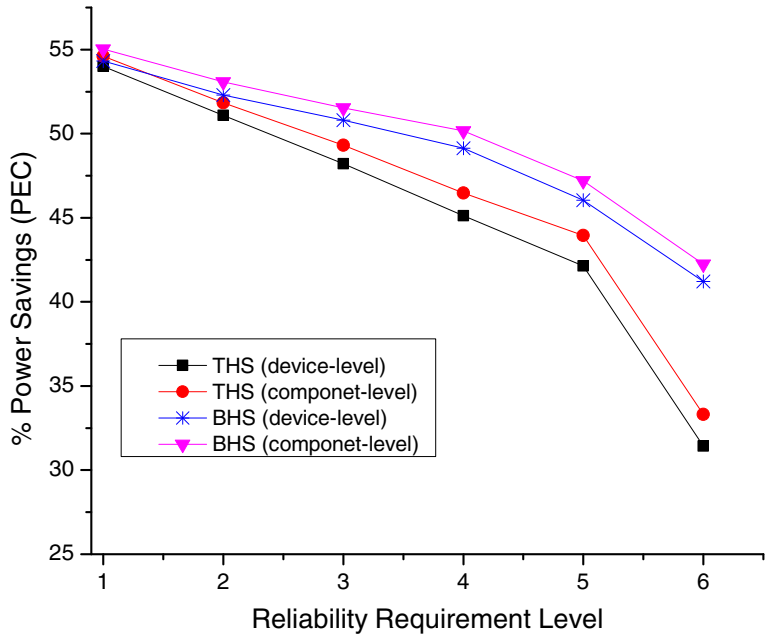

Figure 6 The performance of power savings under different reliability requirements in 128-server Fat-Tree.

of bandwidth allocation instances using BHS and basic shortest path routing (BSP) under the 3456-server sized FatTree topology (using 24-port switches). The result reveals that BHS is several times faster than BSP on average, where BHS takes 0.41 and 1.05 seconds in computing 2000 and 4000 allocation instances, respectively, while BSP costs 1.91 and 4.72 seconds correspondingly. This further witnesses the high computation efficiency of BHS, though the maintenance of BIH may take some time.

The implementation of BHS and THS in real world scenario

We have provided theoretical analysis and simulation studies for the proposed two green schemes BHS and THS. Although the simulation conditions are very close

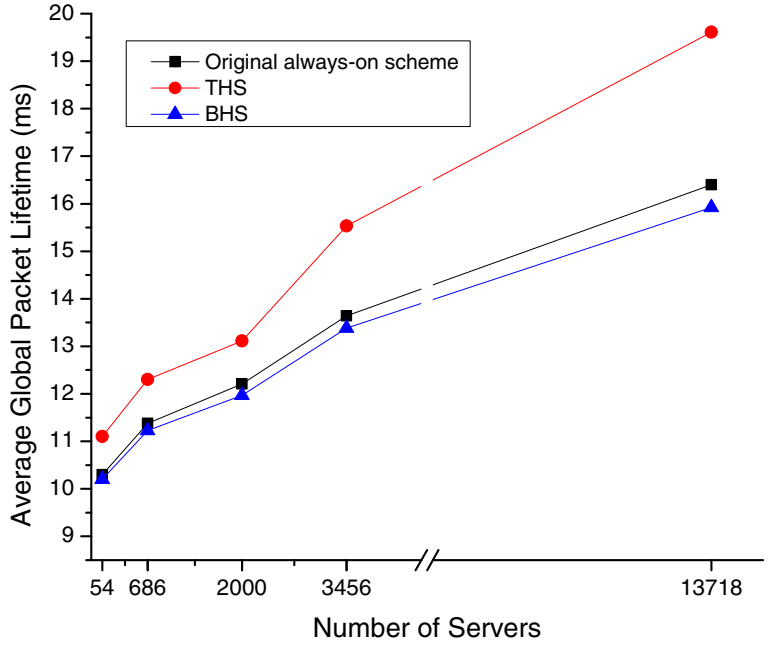

Figure 7 The performance of network latency in different sized Fat-Tree networks.

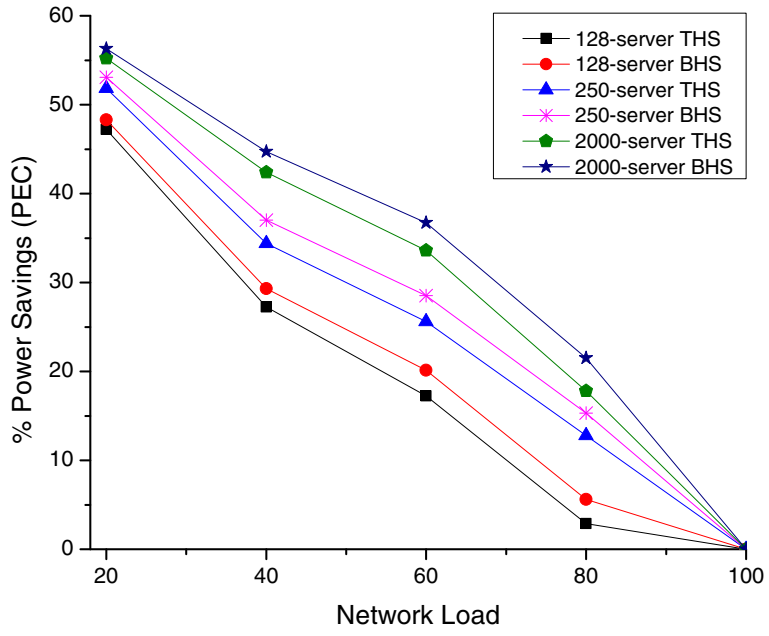

Figure 8 The performance of power savings under different network scales.

to the real world data center environments, there are still some issues needed to be considered in real world deployment of BHS and THS. Firstly, as aforementioned, the traffic patterns and packet inter-arrival time change time to time in the real world, though they may follow some disciplines (Lognormal, Exponential, Weibull, etc.) on the whole in the long run. We only simulated the Exponential flow mode according to the findings about the real world traffic characteristics in [38], and the performance of BHS and THS under other one or several mixed traffic patterns in a real world are left for further evaluation. Secondly, 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. Another issue needed to be considered is the deployment of BHS. BHS requires a centralized controller which plays a very important role in the BI/BIG/BIH generation, bandwidth allocation and routing rules computation. How to guarantee the robustness of the centralized controller is a big concern. Besides,

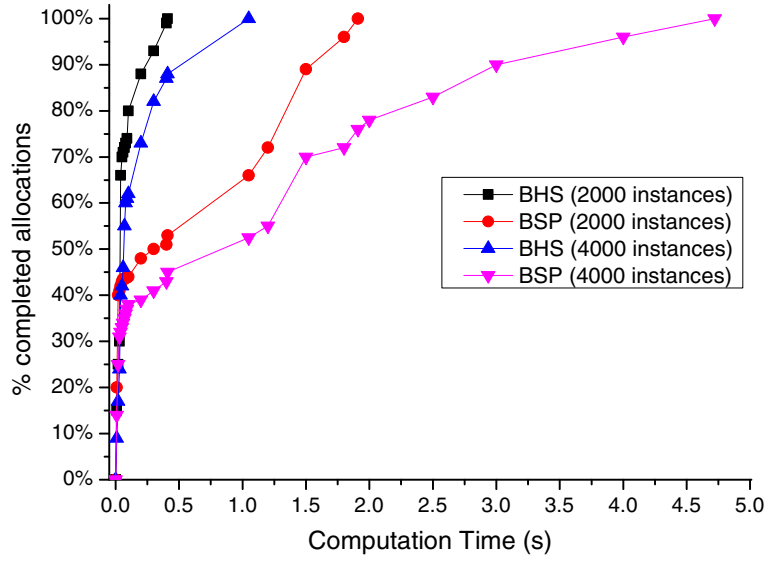

Figure 9 The performance of computation efficiency.

the choice of communication method (in-band or out-of-band) between controller and switches is also needed to be weighted. All of the mentioned issues above are difficult to be simulated in simulators, which should be considered in real world scenarios.

Conclusion

In this paper, firstly we rigorously formulated the power optimization problem in data center networks into an MCF problem, and proved its NP-hardness. In response to this NP-hard problem, inspired by an Artificial Intelligence abstraction technique, we proposed a Blocking Island based heuristic scheme (BHS) by designing an energy-aware bandwidth allocation mechanism and an energy-aware routing algorithm, which can decrease the computation complexity and increase the success ratio of bandwidth allocation. To the best of our knowledge, we are the first to employ the BI paradigm into data center networks to achieve power conservation. Furthermore, we proposed a topology-based heuristic scheme (THS), which focuses on how to compute the minimum network subset and how to distribute the traffic flows properly among the network subset. THS performs faster than BHS and holds the best scalability ($O(N)$), but the quality of the resulting solution is not as good as BHS. BHS achieves a bandwidth guaranteed data center network irrespective of network topologies with a high quality solution and low computation cost. Comparatively, BHS provides a more attractive and practical solution. The conducted simulations further confirmed their feasibility and good performance.

Competing interests

This paper is supported in part by NPRP grant from the Qatar National Research Fund and HKUST RGC Fund. Except this, the authors declare that they have no other competing interests.

Authors' contributions

Author TW proposed the idea of this paper, carefully designed two green frameworks, and drafts the manuscript. Author BQ was responsible for the problem formulation and proof of the NP-hardness of the MCP problem. Author ZS was involved in the experiments and evaluations of two frameworks. Author YX conducted a deep investigation in related works and revised the manuscript draft critically from the system design to the experiments. Author SF and RH discussed frequently with us about this research and gave us many useful and constructive suggestions to perfect the idea of this paper. Author MH supervised our research group about this research topic and given final approval of the version to be published. All authors read and approved the final manuscript.

About the Authors

Ting Wang received his Bachelor Sci. degree from University of Science and Technology Beijing, China, in 2008, and received his Master Eng. degree from Warsaw University of Technology, Poland, in 2011. From 02.2012 to 08.2012 he interned as a research assistant in the Institute of Computing Technology, Chinese Academy of Sciences. He is currently working towards the PhD degree in Hong Kong University of Science and Technology. His research interests include data center networks, cloud computing, green computing, and software defined network.

Bo Qin is currently a PhD student in Department of Computer Science and Engineering, the Hong Kong University of Science and Technology. He was a visiting PhD student at Aarhus University and University of Southern California. His research interests include spectral graph theory, algorithmic game theory and combinatorial optimization.

Zhiyang Su received his B.E. degree in computer science and technology from China University of Geosciences (Beijing) in 2009, and M.S degree in computer network and application from Peking University in 2012. Currently, he is pursuing Ph.D. degrees in Hong Kong University of Science and Technology. His research interests focus on software defined networking (SDN) and data center networking, especially on improving the performance of SDN.

Yu Xia is currently a postdoctoral fellow in Department of Computer Science and Engineering, the Hong Kong University of Science and Technology. He received the Ph.D. in computer science from Southwest Jiaotong University, China. He was a joint Ph.D student and a visiting scholar at Polytechnic Institute of New York University. His research interests include high-performance packet switches, data center networks and network architectures.

Mounir Hamdi received the B.S. degree from the University of Louisiana in 1985, and the MS and the PhD degrees 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 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 an IEEE Fellow for contributions to design and analysis of high-speed packet switching. Prof. Hamdi 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.

Sebti Foufou obtained a Ph.D. in computer science in 1997 from the University of Claude Bernard Lyon I, France, for a dissertation on parametric surfaces intersections. He worked with the computer science department at the University of Burgundy, France from 1998 to 2009 as an associate professor and then as a full professor. He was invited as a guest researcher at NIST Maryland, USA, in 2005-2006. He is with the computer science department at Qatar University since September 2009. His research interests concern image processing, geometric modelling, and data representations and processing for product lifecycle management.

Ridha Hamila received the Master of Science, Licentiate of Technology with distinction, and Doctor of Technology degrees from Tampere University of Technology (TUT), Department of Information Technology, Tampere, Finland, in 1996, 1999, and 2002, respectively. Dr. Ridha Hamila is currently an Associate Professor at the Department of Electrical Engineering, Qatar University, Qatar. Also, he is adjunct Professor at the Department of Communications Engineering of TUT. From 1994 to 2002 he held various research and teaching positions at TUT within the Department of Information Technology, Finland. From 2002 to 2003 he was a System Specialist at Nokia research Center and Nokia Networks, Helsinki. From 2004 to 2009 he was with Etisalat University College, Emirates Telecommunications Corporation, UAE. His current research interests include mobile and broadband wireless communication systems, cellular and satellites-based positioning technologies, synchronization and DSP algorithms for flexible radio transceivers. In these areas, he has published over 60 journal and conference papers most of them in the peered reviewed IEEE publications, filed two patents, and wrote numerous confidential industrial research reports. Dr. Hamila has been involved in several past and current industrial projects Qtel, QNRF, Finnish Academy projects, TEKES, Nokia, EU research and education programs. He supervised a large number of under/graduate students and postdoctoral fellows.

Acknowledgement

We thank Mrs. Shauna Dalton who carefully revised this paper for grammar and spelling.

Author details
[1]Hong Kong University of Science and Technology, Hong Kong SAR, China. [2]Hamad Bin Khalifa University, Doha, Qatar. [3]Qatar University, Doha, Qatar.

Received: 18 November 2014 Accepted: 23 April 2015
Published online: 10 May 2015

References

1. Ting W, Zhiyang S, Yu X, Yang L, Jogesh M, Mounir H (2014) SprintNet: A high performance server-centric network architecture for data centers. In: Communications (ICC), 2014 IEEE International Conference on. IEEE. pp 4005-4010
2. Ting W, Zhiyang S, Yu X, Yang L, Jogesh M, Mounir H (2015) Designing efficient high performance server-centric data center network architecture. Comput Netw 79:283-296
3. Ting W, Yu X, Jogesh M, Mounir H, Sebti F (2014) A general framework for performance guaranteed green data center networking. In: Global Communications Conference (GLOBECOM), 2014 IEEE. IEEE. pp 2510-2515
4. Abts D, Marty MR, Wells PM, Klausler P, Liu H (2010) Energy proportional datacenter networks. In: ACM SIGARCH Computer Architecture News. ACM Vol. 38. pp 338–347
5. Heller B, Seetharaman S, Mahadevan P, Yiakoumis Y, Sharma P, Banerjee S, McKeown N (2010) Elastictree: Saving energy in data center networks. In: NSDI Vol. 10. pp 249-264
6. Mahadevan P, Banerjee S, Sharma P, Shah A, Ranganathan P (2011) On energy efficiency for enterprise and data center networks. Commun Mag IEEE 49(8):94–100
7. Greenberg A, Hamilton J, Maltz DA, Patel P (2008) The cost of a cloud: research problems in data center networks. In: ACM SIGCOMM computer communication review 39(1):68-73
8. Pelley S, Meisner D, Wenisch TF, VanGilder JW (2009) Understanding and abstracting total data center power. In: Workshop on Energy-Efficient Design
9. Ting W, Zhiyang S, Yu X, Mounir H (2014) Rethinking the Data Center Networking: Architecture, Network Protocols, and Resource Sharing. In: Journal of IEEE Acess 2(1):1481-1496
10. Dean J, Ghemawat S (2008) Mapreduce: simplified data processing on large clusters. Commun ACM 51(1):107–113
11. Dixit A, Prakash P, Hu YC, Kompella RR (2013) On the impact of packet spraying in data center networks. In: INFOCOM, 2013 Proceedings IEEE (pp. 2130-2138). IEEE
12. Arlitt M, Bash C, Blagodurov S, Chen Y, Christian T, Gmach D, Hyser C, Kumari N, Liu Z, Marwah M, McReynolds A, Patel C, Shah A, Wang Z, Zhou R (2012) Towards the design and operation of net-zero energy data centers. In: 13th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)
13. Goiri Í, Le K, Nguyen TD, Guitart J, Torres J, Bianchini R (2012) Greenhadoop: leveraging green energy in data-processing frameworks. In: Proceedings of the 7th ACM european conference on Computer Systems. ACM. pp 57-70
14. Nguyen K-K, Cheriet M, Lemay M, Savoie M, Ho B (2013) Powering a data center network via renewable energy: A green testbed. Internet Comput IEEE 17(1):40-49
15. David H, Fallin C, Gorbatov E, Hanebutte UR, Mutlu O (2011) Memory power management via dynamic voltage/frequency scaling. In: Proceedings of the 8th ACM international conference on Autonomic computing. ACM. pp 31-40
16. Leverich J, Monchiero M, Talwar V, Ranganathan P, Kozyrakis C (2009) Power management of datacenter workloads using per-core power gating. Comput Architecture Lett 8(2):48–51
17. Meisner D, Gold BT, Wenisch TF (2009) Powernap: eliminating server idle power. In: ACM SIGARCH Computer Architecture News 37(1):205-216
18. Rangan KK, Wei G-Y, Brooks D (2009) Thread motion:fine-grained power management for multicore systems. In: ACM SIGARCH Computer Architecture News (Vol. 37, No. 3, pp. 302-313). ACM
19. Ting W, Zhiyang S, Yu X, Bo Q, Mounir H (2014) NovaCube: A low latency Torus-based network architecture for data centers. In Global Communications Conference (GLOBECOM), 2014 IEEE (pp. 2252-2257). IEEE
20. Abu-Libdeh H, Costa P, Rowstron A, O'Shea G, Donnelly A (2010) Symbiotic routing in future data centers. ACM SIGCOMM Comput Commun Rev 40(4):51–62
21. Hong C-Y, Caesar M, Godfrey P (2011) Pcube: Improving power efficiency in data center networks. In: Cloud Computing (CLOUD), IEEE International Conference on. IEEE. pp 65-72
22. Singla A, Singh A, Ramachandran K, Xu L, Zhang Y (2010) Proteus: a topology malleable data center network. In: Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks. ACM. p 8
23. Valancius V, Laoutaris N, Massoulié L, Diot C, Rodriguez P (2009) Greening the internet with nano data centers. In: Proceedings of the 5th international conference on, Emerging networking experiments and technologies. ACM. pp 37-48
24. Liu L, Wang Hm, Liu X, Jin X, He WB, Wang QB, Chen Y (2009) Greencloud: a new architecture for green data center. In: Proceedings of the 6th international conference industry session on Autonomic computing and communications industry session. ACM. pp 29-38
25. Mann V, Kumar A, Dutta P, Kalyanaraman S (2011) Vmflow: leveraging vm mobility to reduce network power costs in data centers. In: NETWORKING. Springer Berlin, Heidelberg. pp 198-211
26. Shang Y, Li D, Xu M (2010) Energy-aware routing in data center network. In: Proceedings of the first ACM SIGCOMM workshop on Green networking (pp. 1-8). ACM
27. Martello S, Toth P (1990) Knapsack problems: algorithms and computer implementations. John Wiley and Sons, Inc.
28. Frei C, Faltings B (1997) Simplifying network management using blocking island abstractions. Internal Note from the IMMuNe Project
29. Mackworth AK (1992) Constraint satisfaction problems. Encyclopedia of AI, 285, 293
30. Hong C-Y, Caesar M, Godfrey PB (2012) Finishing flows quickly with preemptive scheduling. ACM SIGCOMM Computer Communication Review 42(4):127-138
31. Wilson C, Ballani H, Karagiannis T, Rowtron A (2011) Better never than late: meeting deadlines in datacenter networks. In: ACM SIGCOMM Computer Communication Review (Vol. 41, No. 4, pp. 50-61). ACM
32. Bodík P, Menache I, Chowdhury M, Mani P, Maltz DA, Stoica I (2012) Surviving failures in bandwidth-constrained datacenters. In: Proceedings of the ACM SIGCOMM 2012 conference on, Applications, technologies, architectures, and protocols for computer communication. ACM. pp 431-442
33. Al-Fares Mohammad, Loukissas Alexander, Vahdat Amin (2008) A scalable, commodity data center network architecture. In: ACM SIGCOMM Computer Communication Review Vol. 38. pp 63-74
34. (2012) Openflow switch specification v1.3.0. www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.0.pdf
35. Ford Alan, Raiciu Costin, Handley Mark, Bonaventure Olivier and others (2011) Tcp extensions for multipath operation with multiple addresses. Internet Engineering Task Force, RFC, 6824
36. Liu Y, Muppala J (2013) DCNSim: A data center network simulator. In: Distributed Computing Systems Workshops (ICDCSW), 2013 IEEE 33rd International Conference on. IEEE. pp 214-219
37. Nexus C (2012) 3048 switch data sheet
38. Benson T, Akella A, Maltz DA (2010) Network traffic characteristics of data centers in the wild. In: Proceedings of the 10th ACM SIGCOMM conference on Internet measurement. ACM. pp 267-280