Environment-Aware Virtual Slice Provisioning in Green Cloud Environment

Kim Khoa Nguyen and Mohamed Cheriet

Abstract—Environmental footprint resulting from datacenters activities can be reduced by both energy efficiency and renewable energy in a complementary fashion thanks to cloud computing paradigms. In a cloud hosting multi-tenant applications, virtual service providers can be provided with real-time recommendation techniques to allocate their virtual resources in edge, core, or access layers in an optimal way to minimize costs and footprint. Such a dynamic technique requires a flexible and optimized networking scheme to enable elastic virtual tenants spanning multiple physical nodes. In this paper, we investigate an environment-aware paradigm for virtual slices that allows improving energy efficiency and dealing with intermittent renewable power sources. A virtual slice consists of optimal flows assigned to virtual machines (VMs) in a virtual data center taking into account traffic requirements, VM locations, physical network capacity, and renewable energy availability. Considering various cloud consolidation schemes, we formulate and then propose an optimal solution for virtual slice assignment problem. Simulations on the GSN showed that the proposed model achieves better performance than the existing methods with respect to network footprint reductions.

Index Terms—Energy-aware cloud, virtual slice allocation, multi-tenant cloud, cloud computing, virtual network embedding

1 INTRODUCTION

D ECENTLY, many giant Information and communications Received, many game incorporations, like Amazon and Google, have established data centers for hosting cloud applications. These data centers are energy-intensive, and constitute one of the largest consumers of electrical power in the economy. They are therefore one of the key sources of greenhouse gas (GHG) emissions. Growing constraints on power, cooling and distribution costs further compound the challenges faced by both data center operators and service providers who are heavily data center-dependent. New paradigms are required to power the data centers, as well as new approaches to data center design and management. The current approach to dealing with the green data center problem is improving energy efficiency, which is an attempt to reduce energy consumption at the micro level. However, an unconstrained micro-level energy efficiency approach most likely leads to an overall increase in energy consumption according to the Khazzoom-Brookes postulates (also known as Jevon's paradox) [1]. Powering data center services by a renewable energy should therefore be considered in parallel with energy efficiency efforts. As losses incurred in electricity transmission over power utility infrastructures are much higher than those caused by data transmission, relocating ICT facilities, especially power-greedy data centers, near renewable energy sources is a more efficient solution than trying to bring electricity to an existing given location. Moreover, such a solution is well aligned with the greening ICT movement that is aimed to mitigate global

Manuscript received 1 Dec. 2013; revised 19 Sept. 2014, 1 July 2014; accepted 6 Oct. 2014. Date of publication 8 Oct. 2014; date of current version 12 June 2015. 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/TSC.2014.2362544

warming progression resulting from the booming electrical consumption of ICT equipment, in particular connected personal devices which are responsible for more than 10.5 percent of the global energy consumption [2]. Developing distributed computing architectures to maximize the utilization of renewable energy for ICT services is therefore one of the most promising ICT strategies to mitigate global warming progression.

As renewable energy sources, such as solar and wind, are often intermittent, fine grained monitoring and control, as well as highly flexible internetworking mechanisms are crucial. Recently, virtualization paradigms have attracted considerable attention for building such a complex data center network. It enables convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Virtualization allows maximizing the delivery of services and hence revenue, at the same time minimizing the costs of support and maintenance, rack space, head load, and power consumption. Virtualized data centers provide the abstraction of computing resources and electrical controls through the elastic use of consolidated resources pools, and provide opportunities for higher utilization and energy savings. However, maximizing renewable energy utilization and cost effectiveness while ensuring performance and other quality of service (QoS) guarantees requires leveraging important and extremely challenging trade-offs. So far, building large-scale distributed networks serving multiple cloud service providers with various performance requirements and while taking into account environmental impacts has been an emerging research realm.

Due to high scalability and dynamicity of resources, users, protocols and architecture, virtual network provisioning with energy efficiency and renewable energy awareness

The authors are with the Automated Production Engineering, Ecole de Technologie Superieure, University of Quebec, Montreal, QC H3K1K3, Canada. E-mail: knguyen@synchromedia.ca, mohamed.cheriet@etsmtl.ca.

is emerging for next generation clouds. A key challenge resides in the ability of altering virtual network topologies and routing schemes dynamically and flexibly according to traffic requirements and the availability of renewable energy sources. As shown in [3], a "follow the sun, follow the wind" algorithm might allocate/relocate virtual machines (VMs) of a virtual data center (so called slice) across multiple nodes, which results in various virtual topologies, and so network capacity and slice power consumption would also be different. Mapping virtual slice to underlying infrastructure accurately and optimally in realtime is substantial to achieve a green cloud network. Unfortunately, recent research on virtual network mapping solutions has not yet paid attention to unique characteristics of cloud computing, such as workload consolidation and multipath forwarding in Layer 2 (or link layer in the TCP/IP network model) flat network, so far. This may result in inefficient resource allocation schemes. This paper is, therefore, dedicated to improving the existing virtual network mappings, and to provide a realistic implementation model for the problem.

The advent of software-define networking (SDN) provides a unique opportunity to effectively implement such a mapping solution. Especially, the OpenFlow framework [4] that supports programmability of network functions and protocols is the cornerstone to build a flexible and environment-aware network management framework. Based on unified cloud management framework of power control devices, electrical consumers and software applications, the green ICT service provisioning solution that we developed [5] allows achieving energy efficiency in data centers, and more importantly harvesting renewable energy sources with minimal battery or grid backup in order to power a network of geographically distributed small data centers. This strategy is complementary to traditional energy efficiency approaches, which focus on micro-processor design, computer design, and power-on-demand architectures.

The contribution of this paper is two-fold: i) we formulate and proposed an algorithm to resolve the problem of virtual slice assignment with respect to minimal footprint emissions in a neutral-carbon network taking into account both VM consolidation and multipath networking in data centers, and ii) we provide an implementation model to recommend the optimized resource allocation schemes to virtual data center service providers based on software-defined techniques.

Solutions presented in this paper have been deployed in the GreenStar Network (GSN) project [6], the first worldwide initiative for ICT service provision based entirely on renewable energy sources across Canada and around the world. The testbed has successfully demonstrated the ability of transporting user service applications to be processed in data centers located in proximity to green energy sources in an optimal fashion. This paper presents virtual networking paradigms that we have implemented in the GSN in order to keep the testbed powered entirely by intermittent renewable energy while maintaining data center service. Regarding our prior publications on carbon emission reduction [7], [8], energy renewable provisioning [3], this paper rather focuses on the architecture of virtual data center designed to support optimal VM placement in the GSN, as presented in [7], [5].

The rest of this paper is organized as follows. In the next section, we recall the architecture and working principles of the GSN, followed by its virtual network architecture. Next, we review related work, and then model the optimal virtual slice allocation problem, together with a heuristic solution. We then discuss an implementation models, and outline experiments on the GSN to validate the proposed solution. Finally, we conclude the paper and present our future work.

2 RELATED WORK

It is worth noting that, most of prior research in the field of greening ICT has focuses on data centers, in particular computing resources such as servers and storage [5], [7]. Research in greening network is rather interested in energy efficiency and power reduction in traffic processing. For example, various projects undertaken at Berkeley Labs deal with greening Ethernet through adapting link rate to effective traffic level as higher data rates require dramatically more power [35]. The ECONET project of Europe [36] aims at aggressively modulating power consumption of network devices according to actual workloads and service requirements. However, the portion of power consumed by network is indeed much smaller than data centers as a result from the deployment of optical devices, which consume much less power than electronic devices.

In a prior paper [5], an optimized VM placement and migration model for minimizing carbon footprint has been proposed, taking into account CPU and memory resources. The model, called server-centric model, tried to optimize energy consumption of a set of VMs regardless of the links between VMs. In other words, the model places VMs onto physical servers with minimal footprint, given that each server has specific energy consumption. VMs consolidation is taken into account to allocate as many VMs as possible in greenest servers (e.g., the servers consume least energy). While computing resources are crucial for Internet-based services, the consumption of network would also be important regarding network-oriented services such as massive data downloading or telecommunications applications. In addition, the paper does not consider inter VMs or intra virtual data center traffic, which is responsible for a large amount of energy consumed by networking devices.

In [16] and [17], implementation models for power-saving flow assignment have been presented. However, they all focused on flow assignment for hosts in a tree-based intra data center interconnect. No attention has been paid to VMs requirements and traffic patterns among VMs. In addition, inter data center connections have not been taken into account so far, in particular with respect to virtual data centers spanning geographically distributed nodes in a largescale "follow the sun, follow the wind" network.

In [32], the virtual network embedding (VNE) problem has been presented using nodes and links coordination. Mapping virtual networks into physical infrastructure is decomposed into node mapping and link mapping, and a set of algorithms are proposed to coordinate these two problems in order to find optimal mapping of an entire virtual slice. The node mapping problem is resolved by a mixed integer programming (MIP) formulation, and the link mapping problem is resolved using solutions for



Fig. 1. Multipath connection among hosts in a BCube data center and mapping a virtual slice of VMs onto BCube infrastructure.

multicommodity flow problem (MFP) presented in [37]. The MIP formulation also considers link mapping, which facilitates the coordination between two problems. Although it is an efficient method to map virtual networks into substrate networks, the coordinated VNE problem does not consider cloud computing techniques, such as virtual machine consolidation and data center multipath network. It is the same for [33] and [34], in which consolidation technique has been used to combine different virtual slices instead of rescaling an existing slice.

Recently proposed intra data center networks, e.g., Fattree [11], BCube [12], DCell [23] are based particularly on symmetric port commodity switches and uses symmetric link capacity to interconnect switches and servers. In BCube, servers are connected to switches only and vice versa, and thus servers play active role in routing. In DCell, servers are connected to switches only in the lowest level of the architecture; higher levels are constructed with server to server connection, and thus servers play active role in routing. In Fat-tree, server connects only to edge switch and edge switch connects to aggregate switches and aggregate switch further connects to core switches and thus network fabric itself play role in routing. In [13], authors have compared bandwidth provided by Fat-tree, BCube and DCell; however, power consumption of these topologies has not been accurately estimated so far.

Fig. 1 gives an example of a data center based on BCube architecture with three layers: edge hosts, aggregation and core switches. Each rack contains four servers and an aggregation switch. As shown, each host has two network interfaces, one connecting to an aggregation switch and the other connecting to a core switch, and so hosts may relay traffic for each other. It is worth noting that BCube has not been designed with SDN-enabled feature in mind, and this architecture is aimed principally at improving fault-tolerance of data center interconnect. For example, two paths (e.g., Path 1 and Path 2) in Fig. 1 can be used in parallel to link hosts 1 and 6.

Recent developments in data center interconnect featured multipath forwarding paradigms which spread traffic among different paths in order to improve load balancing and fault-tolerance. Multipath techniques, such as equal cost multipath (ECMP) [30] and valiant load balancing (VLB) [31] have been deployed in both commercial products [29] and research testbeds [17]. Unfortunately, these techniques have yet been taken into account when addressing cloud resource allocation problems so far. In [14], the authors present a problem of optimizing network power consumption with respect to end-users demand, in which VM placement is optimized taking into account link power consumption. Similarly, [15] proposes an optimization model to aggregate traffic on alternated links, hence turning off idle switches and links in a data center. Both articles present theoretical work and do not consider multipath forwarding.

In this paper, we investigate the problem of optimal locations for VMs and their connection on a physical underlying infrastructure, and propose networking architecture to support such optimized scheme. In particular, we emphasize on intra virtual slice traffic, which is responsible for an important amount of footprint with regard to other kinds of traffic in the whole data center network. In a federated cloud like the GSN, a virtual slice may contain VMs located in distributed data centers. Indeed, virtual slice service is required to host large-scale network applications composed of distributed components. For example, an Internet multimedia service (IMS) system [19] involves multiple call session control function (CSPF) proxies, home subscriber server (HSS) databases, and several gateways; each component can be run on one or several VMs. A user request sent to the IMS system will therefore be processed by multiple components. Such a system can be hosted by a slice of VMs. A virtual link between two VMs can go across multiple physical servers, switches and routers, depending on the locations of the VMs. As each physical link has limited capacity, determining an optimal connection scheme which can accommodate user requirements and save power consumption of the network plane is challenging, in particular when there is a large number of nodes and links. As shown in Fig. 1, a possible mapping of such a virtual slice of IMS can be achieved taking advantage of VM consolidation and multipath features. Given three VMs in the slice, each having 2-CPU requirement, and the bandwidth among each pair of VMs is 2 Gbps, two VMs (namely CSCFs and HSS) can be consolidated into host 6, the GW VM can be mapped onto host 1, and the virtual link between CSCFs and GW is mapped onto two physical links (Path 1 and Path 2). As two VMs are hosted by a single physical server, the footprint of the virtual slice can be lower than the footprint of a different mapping which allocates two VMs in two different servers. The virtual link between these two VMs is no longer required because they can communicate to each other via internal memory of the server. In addition, a 2 Gbps bandwidth requirement can be affordable using multiple 1 Gbps physical links. This general mapping principle can be applied without any modification for various intra data center topologies, such as Tree, Fat-tree or BCube.

3 PROBLEM FORMULATION AND SOLUTION

The problem of optimized flow assignment results in an optimal virtual network of VMs allocated on physical hosts distributed across a network of data centers. We assume a data center model having an aggregation switch that plays the role of a gateway grooming all traffic and then sending to external. In reality, a host may contain multiple VMs, and



Fig. 2. Example of a mapping from a virtual slice request to underlying infrastructure resources.

communications between two VMs located on the same host do not consume network resource.

The problem recurs to a set of flow assignment and node assignment problems in an undirected graph which result in an optimal virtual network of VMs allocated on physical hosts distributed across a network of data centers, satisfying resource requirements of VMs and flows between them. Unlike the traditional substrate network mapping problem which deals only with link capacity, there is new constraints on computing power on a set of nodes in the graph (c.f. the set of hosts).

Fig. 2 illustrates an example of mapping a virtual slice into underlying infrastructure. Four VMs a, b, c, d with the same computing requirement (5) are mapped to an infrastructure having five servers A, B, C, D, E with the same computing capacity (10). Each virtual link between a pair of VMs is 10 Mbps, and each physical link between a pair of servers is 20 Mbps. A possible mapping, as given in Fig. 2C is: (a - > A, b - > B, c - > C, d - > D, ab - >AB, bc - > BC, ac - > AC, cd - > CD). The notation X/Yin the figure means a capacity X is allocated over the total capacity Y. Assume the emissions of each computing unit (i.e., one CPU) and each bandwidth unit (i.e., 1 Mbps link) are both 1g CO2. The total footprint of the virtual slice is therefore:

$$f = f_A + f_B + f_C + f_D + f_{AB} + f_{BC} + f_{AC} + f_{CD}$$

= 5 + 5 + 5 + 5 + 10 + 10 + 10 + 10 = 60. (1)

In order to formulate the problem, the entire network of data centers can be modeled as an undirected graph G(V, E), in which V is the set of vertices and $E \in V \times V$ is the set of edges. $V = H \cup S$, in which H is the set of hosts and S is the set of switches (including the gateways). Each edge $(i, j) \in E$ represents a communication link between two nodes i and j which are hosts or switches (including the gateways). The available bandwidth capacity of an edge is denoted by c(i, j). If there is no physical link between i and j, we assign c(i, j) = 0.

Let $p_L(i, j)$ refer to the power consumed on the physical link when sending a bit from i to j and vise versa. In other words, $p_L(i, j)$ is the total energy consumed by two ports of the nodes; $p_L(i, j) = 0$ if there is no link (i.e., c(i, j) = 0). Considering a virtual slice having N VMs, and there are Mvirtual links among these VMs; power consumption of the *i*th VM hosted by server *j*, $p_i(j)$, can be calculated from VM requirements and the power consumption of the host as described in [9]. As two or more VMs running on the same host communicate to each other without using any physical link, we consider the flow(s) between them has zero capacity. Therefore, without any loss of generality, we assume that each server initially hosts only a single VM belonging to a given slice. This VM represents all VMs actually hosted by the server; its processing capacity is the sum of all actual VMs, and its links is the set of outgoing links of all actual VMs. A virtual link (or a flow) between each pair of VMs in N hosts is denoted by $F_m(s_m, t_m, d_m)$, in which s_m and t_m are respectively source and destination hosts of the *m*th flow, $s_m, t_m \in N$; and d_m is the demanded bandwidth of the flow.

Let D be the number of data centers in the network. Each data center $d \in D$ is accessible through a gateway g_d which is considered as a special node. As any external request has to go through the gateway to a VM, we model external traffic going to a slice by N flows from the gateway to N VMs of the slice. If there is no external traffic to kth VM, the demanded bandwidth of the kth flow is 0. The slice will, therefore, contains $M' = M + D \times N$ flows, in which M is the number of flows among VMs, and $D \times N$ is the number of flows in the slice (note that some flows in the set of $D \times N$ flows may be zero-bandwidth, e.g., a flow between a gateway and a VM located in two different data centers).

As stated in [3], each data center *d* in the network is powered by a renewable energy source, with a green factor r_d defined in [18]. Therefore, each link (i, j) is associated with a green factor $r_L(i, j)$ which is determined by the locations of two heads of the link and the green factors of data centers $r_L(i, j) = \frac{r_i + r_j}{2}$. The green factor of a link between two data center gateways is the mean of the green factors of two data centers.

The problem is to determine a set of vertices $H' \in H$ hosting the VMs, and a set of edges $E' \in E$ linking VMs in the slice, satisfying computing and traffic demand, and minimizing the total footprint of the slice. We use two matrices to represent the results:

$$x_m(i,j) = \begin{cases} bandwidth \ on \ link \ (i,j) \ reserved \ for \ flow \ m \\ 0 \ if \ flow \ m \ is \ not \ routed \ through \ link \ (i,j), \end{cases}$$
(2)

$$y(i,j) = \begin{cases} 1 & VM \ i \ is \ hosted \ by \ the \ physical \ server \ j} \\ 0 & otherwise. \end{cases}$$
(3)

So, the problem formulation is follows:

$$\begin{aligned} Minimize & \sum_{i,j\in N, m\in M'} x_m(i,j) \times p_L(i,j) \times r_L(i,j) \\ &+ \sum_{i\in N, j\in H} y(i,j) \times p_i(j), \end{aligned}$$
(4)

such:

$$\sum_{m \in M'} x_m(i,j) \le c(i,j), \qquad \forall i,j \in V,$$
(5)

$$\sum_{k \in N} x_m(s_m, k) = \sum_{k \in N} x_m(k, t_m) = d_m, \quad \forall m \in M',$$
(6)

$$\sum_{k \in N} x_m(i,k) = \sum_{k \in N} x_m(k,i), \forall m \in M', \forall i \in V \setminus \{s_m, t_m\}.$$
 (7)

Equation (4) defines the objective function that is the minimal footprint of the virtual slice of N VMs and Mflows. Equations (5), (6) and (7) state the constraints of the problem. (5) states that the total bandwidth allocated for Mflows on any link (i, j) does not exceed the available bandwidth capacity of (i, j). Equation (6) states the demand satisfaction constraint, i.e. for any flow, the outgoing traffic at the source or the incoming traffic at the destination equal to the demand of the flow. Equation (7) states that for any intermediate node of a flow, the incoming traffic is equal to the outgoing traffic.

The aforementioned integer linear programming (ILP) multipath routing problem is NP-complete [20]. A solution for such problem can be obtained with a mathematical solver, however, it does not return results in a timely manner as required by of slice establishment operation. In reality, as the "follow the sun, follow the wind" algorithm is triggered only a few times in a day, a mathematical solver can be used to obtain the optimal result when the network has a relatively small number of nodes (order of fifty). When the number of variables increases, a heuristic algorithm is required to reduce calculation space, which can be defined based on following theorems and lemmas.

- **Theorem 1.** The optimization problem (3) results in the minimum value if and only if all 1-degree virtual nodes (VMs) are optimally consolidated with their adjacent nodes. A 1-degree node is a node which is connected to only one other node in the slice.
- Proof. Assuming an optimal mapping in which there is still a 1-degree virtual node X which is not optimally consolidated. As X can be consolidated with the node Y which is directly linked, consolidating X and Y will eliminate a virtual link between them. Without any loss of generality, we assume that the degree of Y is greater than 1, and the green factor is 1 for all servers and links. If the optimal consolidation moves X toward Y, the total power consumption will be reduced an amount $p_{XY} = (p_L(i, j))$ $\times x(i,j)) + (p_i(X) - p_j(X))$ in which *i* and *j* are two servers hosting X and Y respectively. If the optimal consolidation moves Y toward X, the total power consumption will be reduced an amount $p_{YX} = p_L(i, j) \times (x(i, j) - p_L(i, j))$ $maxflow(Y)) + (p_i(Y) - p_i(Y))$ in which maxflow(Y) is the total capacity of flows going through Y. This is because when Y moves toward X, the flow XY may need to be increased to afford existing flows going through Y. The consolidation does not affect bandwidth allocation other nodes because X is a 1-degree node, so its unique link is to Y; and as the consolidation is supposed to be optimal, eliminating a virtual link does not create a



Fig. 3. Mappings with and without 1-degree virtual node consolidation.

bottleneck. As well, the optimal consolidation of X and Y effectively reduce the power consumption (and indirectly footprint) of the network, either by an amount p_{XY} or p_{YX} . In other words, a mapping without the consolidation XY is not optimal.

Fig. 3 illustrates two mappings with and without 1degree virtual node consolidation, using input request and infrastructure presented in Figs. 2A and 2B respectively. Assume all nodes and links having the same green factor 1, and their power consumption is proportional to the requirements, the total footprint of the mapping with virtual node consolidation (Fig. 3A) is:

$$f = f_A + f_B + f_C + f_{AB} + f_{BC} + f_{AC}$$

= 5 + 5 + 10 + 10 + 10 + 10 = 50. (8)

Server *D* and link *CD* are not involved in this equation because VM *d* is consolidated in server *C*. The footprint of the mapping without consolidation (Fig. 3B) is 60 (see Eq. (1)). The differecene in terms of footprint of two mapping schemes is therefore: 60 - 50 = 10. This example shows emission reduction resulting from a consolidation of a 1-degree virtual node.

We use Lemma 1 to calculate the footprint reduction of two consolidation scheme:

Lemma 1. Emissions change resulting from a consolidation of a 1-degree node X with an N-degree adjacent node Y (N > 1) is:

$$f_{XY} = (r_L(i,j) \times p_L(i,j) \times x(i,j)) + (r_i \times p_i(X)) - (r_j \times p_j(X)) if X is moved toward Y, or (9)$$

$$f_{YX} = (r_j \times p_j(Y)) - (r_i \times p_i(Y)) + r_L(i, j) \\ \times (x(i, j) - maxflow(Y)) \text{ if } Y \text{ is moved toward } X$$
(10)

in which i and j are two servers hosting X and Y respectively, and x(i, j) is bandwidth of the flow between X and Y.

Proof. This is straightforward calculations.

From Theorem 1 and Lemma 1, we design an algorithm to eliminate 1-degree virtual nodes, as follows.

Line 1 in Algorithm 1 walks through all 1-degree virtual nodes. Line 2 determines an adjacent node. Line 3 computes footprints of two consolidation scheme (X to Y and Y to X). Line 4 compares two schemes. Line 5 checks if two virtual nodes can be consolidated based on available computing resources on host *j*. Line 6 moves X toward Y if it is possible. Line 7 checks if the link capacity (X,Y) is greater than the



Fig. 4. Mappings with and without 4-degree virtual node consolidation.

total capacity of the flows going through Y. Line 8 checks if two virtual nodes can be consolidated based on available computing resources on host *i*. Line 9 moves Y toward X if it is possible.

Algorithm 1. Eliminating 1-degree VMs

| 1. | for each 1-degree virtual node <i>X</i> hosted by |
|----|--|
| | server <i>i</i> in the list of 1-degree nodes <i>L</i> 1 do |
| 2. | Determine virtual node Y directly linked to X and server j |
| | hosting Y |
| 3. | Compute f_{XY} and f_{YX} using (9) and (10) |
| 4. | if $f_{XY} > f_{YX}$ then |
| 5. | if capacity of $j <$ requirements of X and Y |
| | then continue |
| 6. | else Move X toward Y |
| 7. | else if $x(i,i) > max flow(Y)$ then |
| 8. | if capacity of <i>i</i> < requirements of X and Y |
| | then continue |
| 9. | else Move Y toward X |

As Algorithm 1 walks through all the virtual node, its complexity is O(N) in which *N* is the number of VMs.

- **Theorem 2.** A consolidation of an N-degree (N > 1) virtual node X toward a virtual node Y only reduces the overall footprint when the footprint of their direct links is greater than the sum of the total footprint of links toward X and the difference of footprints of X when running in the new and old locations.
- **Proof.** Consider two virtual nodes X and Y, both having a degree greater than 1, and hosted by two servers *i* and *j*. When X is consolidated toward Y, a link with capacity x(i, j) is eliminated. However, as the degree of X is greater than 1, there are more than one link leading to X. These links will be added to Y after the consolidation. As a result, the consumption of links to Y will be increased a corresponding amount. As each link has a unique power consumption and green factor, the total footprint will only be reduced when the new capacity added to Y results in a footprint being smaller than the footprint produced by the relocation of X.

Fig. 4 illustrates a mapping with a 3-degree virtual node consolidation (i.e. c and a nodes) compared to a mapping with no consolidation. With the same assumption as in Fig. 3, the total footprint of the mapping with 3-degree node consolidation (c is consolidated with a, Fig. 4A) is

$$f = f_A + f_B + f_D + f_{AB} + f_{AC} + f_{CD}$$

= 10 + 5 + 5 + 20 + 10 + 10 = 60, (11)

and of the mapping without consolidation (Fig. 3B) is also 60. This demonstrates such a consolidation does not reduce any footprint.

We use Lemma 2 to determine if an N-degree virtual node consolidation is possible and to calculate the footprint reduction resulting from the consolidation.

- **Lemma 2.** An N-degree virtual node X can be consolidated with a virtual node Y if and only if with any Z node adjacent to X $(Z \neq Y)$, the sum of available bandwidth of all flows from Z to Y is greater or equal to the capacity of the virtual link (Z, X).
- **Proof.** When X is moved toward Y, all flows from Z toward X will go directly to Y. Therefore, the total capacity of all flows from Z to Y will be augmented an amount of c_{ZX} . If this amount is greater than the total available capacity of all flows from Z to Y, communications requirements between Z and X will not be affordable. In other words, the sum of available bandwidth of all flows from Z to Y must be greater or equal to the capacity of the virtual link (Z, X).

From Theorem 2 and Lemma 2, we design an algorithm to determine whether a consolidation of two N-degree virtual nodes can be done, and the emissions changes resulting from such consolidation.

Line 1 in Algorithm 2 walks through all adjacent nodes Z of X. Line 2 determines all flows from Z to Y. Line 3 sorts all flows in ascending order of green factors; if a flow goes through multiple links, its green factor is the sum of all green factors of these links. Line 4 loops until the capacity of the virtual link (Z, X) is entirely assigned to flows from Z to Y. Line 5 caculates the available capacity of each flow from Z to Y. Lines 6 and 7 augments the flow. Line 8 continues the loop. In Line 9, the algorithm fails (i.e., two VMs cannot be consolidated) if all flows from Z to Y are visited and the capacity of the virtual link (Z, X) is not entirely assigned to them. Otherwise, the algorithm returns the new footprint resulting from flow change.

| Algorithm 2. | Consolidating | N-degree | VMs X | and Y | |
|--------------|---------------|----------|-------|-------|--|
|--------------|---------------|----------|-------|-------|--|

| | 0 0 |
|------|--|
| 1: 1 | for each Z node adjacent to X do |
| 2. | Build a list <i>L</i> of flows from <i>Z</i> to <i>Y</i> |
| 3. | Sort L in ascending order of green factors, pick the first |
| | flow $L_{i=0}$ |
| 4. | while $c_{ZX} > 0$ do |
| 5. | Calculate c_i is augementing capacity of L_i |
| 6. | if $c_i < c_{ZX}$ then Augment c_i to L_i |
| 7. | else Augment c_{ZX} to L_i |
| 8. | $c_{ZX} = c_{ZX} - c_i; i = i + 1$ |
| 9. | if $i >$ end of L then return FAIL |
| | |

The complexity of Algorithm 2 is O(degree(X)).

Theorem 3. When all optimal consolidations are done, the problem (3) recurs to regular virtual network embedding problem.

Proof. When there is no VM consolidation, the problem (3) seeks only to optimize a static virtual network on a physical underlying infrastructure. It is entirely the virtual network embedding problem with coordinated node and link mapping as described in [32].

Our final algorithm to resolve the optimization problem is follows.

Algorithm 3 is straightforward as a result from Algorithms 1 and 2. In general, as the algorithm has to execute the network embedding algorithm, and then walkthrough all nodes and links, the complexity of the algorithm is $O(N^M + N \times M)$.

| A | lgorithm 3. Environment-aware mapping in cloud |
|----------|---|
| 1. 2. | Run virtual network embedding algorithm if required. Run Algorithm 1 to consolidate all 1-degree virtual nodes |
| 3. | for each N-degree virtual node X (N > 1) do |
| 4. | for each N-degree virtual node Υ (N > 1) |
| | adjacent to X do |
| 5. | Run Algorithm 2 to consolidating X and Y |
| 6. | if Algorithm 2 returns TRUE then Move X toward Y |
| 7. | else continue |

It is worth noting that consolidation can reduce overall reliability of a virtual slice when it is mapped, as a single server or link failure could shutdown multiple VMs or virtual links. A possible solution for this issue is to assign a reliability parameter to each server and link, and define a scalar function of footprint and reliability. The optimization process will then try to allocate VMs and virtual links based on this function instead of individual parameters (i.e., footprint or reliability). As well, consolidation has no impact on QoS as long as physical servers have enough resource to afford VM requirements and overhead generated by hypervisors. Determining the relationship among virtualization overhead, reliability and footprint is out of the scope of this paper, which is dedicated to footprint optimization, and will be part of our future work.

4 GSN IMPLEMENTATION ARCHITECTURE

The GreenStar Network project [6], [3] was initiated by a Canadian consortium of industry, universities and government agencies with the common goal of promoting the utilization of renewable energy for ICT services and reducing GHG emissions. The idea behind the GSN project is that a carbon neutral network must consist of data centers built in proximity to renewable power sources and user applications will be moved to be executed in data centers. Such a network provides an ability to migrate entire virtual data centers to alternate data centers locations while maintaining user service according to the principle of "follow the wind, follow the sun". This is supported by a high speed optical layer having up to 100 Gbps bandwidth capacity. Note that optical networks have modest increase in power consumption, especially with new 100 and 1,000 GHz waves, compared with electronic equipment such as routers and aggregators [25].

As shown in Fig. 5A, the Canadian section of the GSN has five GSN nodes powered by sun, wind and hydroelectricity. It is connected to European green nodes in Ireland (HEAnet): wind/sun, Iceland (NORDUnet): geothermal, Spain (i2CAT): sun, the Netherlands (SURFnet): sun, and some other nodes in other parts of the world such as in China (WiCo): sun, the USA (CalIT2, UCSD): direct current (DC) power/sun, and Egypt (Smart Village): sun. The connected nodes and their main sources of energy have fully been listed in [4].



Fig. 5. The GSN map and a green node powered by solar energy at the Communication Research Centre, Ottawa, Canada.

Architecture of a GSN node is given in Figs. 5B and 5C [9], including three planes: control, data and power, which are managed by a centralized cloud middleware. Each services provider (e.g., research and educational organizations) is provided with a virtual slices consisting of virtual machines and virtual network flows linking the VMs. An environmental assessment services is available to report emissions of the slices. GSN users may choose to deploy their resources in different data centers to reduce the overall footprint.

All servers in the GSN are virtualized by hypervisors, and end-users rent computing power through VMs. GSN network slice service allows users to actively create and manage their networks of VMs (Fig. 6). A similar concept, called virtual private cloud (VPC), is also offered in the market (e.g., by Amazon [10]). However, links in a VPC are fixed when it is created, whilst a slice in the GSN is scalable and flexible thanks to a software-defined networking architecture using OpenFlow technology. Unlike the clouds where the hypervisor directly links VMs to a physical network interface (NIC) of their server, which then connects to a physical switch of the data center, each server in the GSN is equipped with a built-in virtual smart switch (called vSwitch), and hence VMs are connected to their vSwitch before the physical switch. A vSwitch allows isolating or grouping VMs running on a server according to users demand. An OpenFlow controller running on a dedicated



Fig. 6. Virtual user slice migration in the GreenStar network.

VM handles the control plane of the entire network. It controls the vSwitche's flow tables in such a way that all VMs belonging to a user slice will be put in a VLAN which may span multiple vSwitches. Users configure their slices through a web-based graphical interface [6] that translates and then relays user requests to the OpenFlow controller through the GSN cloud middleware. When a VM is moved around servers, vSwitches are dynamically reconfigured by the controller so that the VM network slice remains unchanged (e.g., in Fig. 6, slice 1 is added to the vSwitch in Host 3 after a migration). Dynamic tunnels can be configured by virtual routers when moving VMs between data centers. Flow classification algorithms are also developed for the controller to provide QoS levels to different categories of users.

With a software-defined capability, GSN users (i.e., virtual service operators) are provided with a dynamic recommendation service allowing them to allocate/relocate their virtual tenants on physical servers and network equipment with the lowest environmental impact. The features of such a dynamic service include:

 Simplifying definition of virtual tenant. Rather than defining an entirely virtual network, the service allows redefining and/or changing virtual "flows" linking the VMs. The VMs can also be dynamically migrated to optimal locations.



Fig. 7. GSN middleware with focus on network resource management.

- *Enabling a variety of actions*. In addition to regular VM consolidation as in traditional data centers, flow redirection, path splitting, virtual tenant rescaling, power source selection and other actions can be indicated.
- *Flexible behavior*. A virtual slice can be modified in a reactive way, so that new resources are dynamically added to the tenant when there is computing need. Or, it can be proactive so that the entire tenant is created in advance to computing tasks.

The entire network operates according to its targets which can be, for example, the minimization of the power consumption, minimization of GHG emissions, and maximization of the profits (in terms of traffic circulating on the network). In order to achieve such targets the GSN needs to collect information about the status of each element (at the granularity of VM) on the one hand, and on the other hand calculate precisely the power consumption and emissions, as well as produce real-time optimal allocation schemes of virtual resources. When VMs are migrated or consolidated on servers, rebuilding optimal slices is key element to successful achievement of the GSN.

As presented in [4], the GSN is controlled by a computing and power management middleware including three layers (Fig. 7):

- *Physical layer*. Contains all physical devices, such as servers (hosts), switch, routers, PDU, power generators, power meters, climate control. In general, there are many hosts and switches per data center.
- Virtual resource layer. Contains the drivers of each physical device. Each resource controls a device and offers services to a higher layer. Compute Resources control physical servers. Power Source Resources manage energy generators (e.g., wind turbines or solar panels). PDU Resources control power consumption. Climate Resources monitor environmental condition and humidity. Network resources are responsible for virtual and physical switches. A Facility Resource links Power Source, PDU, Climate, Compute and Network resources. It determines the power consumed by each server and sends notifications to managers when the power or environmental condition changes.

 Management layer. Includes a Controller and a set of managers. Each manager is responsible for its individual resources. For example, the cloud Manager manages Compute resources, the Facility Manager manages the Facility resources, and the Network Manager handles data center connectivity. The Controller is the brain of the network, responsible for determining the optimized location of each VM. It computes the action plans to be executed on each set of resources, and then orders the managers to perform them. Based on information provided by the Controller, these managers execute allocation and migration tasks.

In the GSN, each data center is managed by a middleware instance running on a VM, so called domain (Fig. 7). Each domain contains a Facility resource and all computing, networking, power and climate resources of the data centers. The Facility resource can make decisions by itself to optimize workload and power consumption of the data center. A middleware instance is hosted by a dedicated VM, called virtual organization (VO), containing the managers and Controller. Managers communicate with domains through java message service (JMS) channels. Two schemes are implemented or triggering the power optimization process:

Periodical. The Controller periodically (i.e., each 30 minutes) examines all the domains to get up-to-date data. It then computes the locations of all VMs and their associated flows to build an optimized migration plan in the network. An empty plan means that no relocation is required.

Event-driven. When the Facility resource in a domain detects an abnormal condition, it will immediately send a notification to the VO to trigger the re-computation which will result in relocation of VMs and virtual flows.

In the GSN, a virtual slice is established by the Network Manager following a user request. The Network Manager interacts with an OpenFlow controller, which is running on a dedicated VM, to inject the optimal flow assignment plan to the flow table. In the implementation of GSN, a NOX controller [21] has been deployed. The Network Manager also learns the overall network topology through the OpenFlow controller, which is then served for calculating the optimal flow assignment plan. Each VM in a virtual slice is identified by a combination of MAC address, IP address, and a slice ID. When a VM in a slice is migrated to a new location, the Network Manager will recomputed all flows in the slice using Algorithm 3 and then update all virtual and physical switches involving in the flows.

When a service provider requests a virtual slice, the Controller computes an optimal plan based on data in the cloud knowledge database. If available resources can afford user's request, the Controller will send command to cloud Manager, and through Network Manager to the OpenFlow controller, to allocate required resources.

5 EXPERIMENTAL RESULTS

This section presents simulations carried out on the Canadian section of the GSN including five data centers: Montreal, Ottawa, Calgary, Halifax and Kelowna (Fig. 8). In the simulations, we evaluate network carbon footprint reductions regarding requirements of an experimental application



Fig. 8. Physical connection of GSN Canadian, USA and European nodes over CANARIE network.

named GeoChronos [24]. It is a software platform enabling the earth observation community to share data, scientific applications and to collaborate effectively. The application runs on a multi-processor clustered system with 48 cores in total. Architecture of Geochronos, as shown in Fig. 9, is composed of multiple gateways, application servers, compute servers and database servers. When being virtualized, each of these components can be hosted by a VM. There are consistent flows linking interactive application service gateway (IAS GW) and IAS components, data processing system (DPS) and data management system (DMS) components, and the Portal with IAS, DPS, and DMS. Other flows need also to be established to link Geochronos to NASA, Rackfoce centers, and end-users as well.

Given each VM is provided with eight-core processing power, we may need six VMs to host the application. In order to scale up the simulation, six applications are deployed in the network in the same time, for a total of 36 VMs. As GSN nodes are small data centers, and they are hosting multiple services in parallel, a single node is not able to handle entire Geochronos application. Thus, the virtual slice needs to span multiple nodes when it is migrated around the network according to the "Follow the wind, follow the sun" algorithm.

Table 1 provides the simulation configuration. Energy consumption of servers and VMs is given in [9]. The optimization process for this configuration is under 1 minute.

The network connection plan used to calculate power consumption of Geochronos virtual slice is showed in Fig. 8. The name of each core switch represents the province where the switch is located. [18] gives the green factor of the power grid in each province. We first calculate the power consumed when sending a bit of data between the two data center gateways. This amount is the total of consumption of optical switches linking two data centers, which is estimated as follows [22]

$$P_g = \sum_{1}^{L} \frac{8(H+1)}{B_w} \times P_S,$$
 (12)

where B_w and P_S are respectively the bandwidth capacity (Gbps) and power (kW) of a core switch, and *L* is the



Fig. 9. Geochronos architecture and its virtual slice.

number of core switches between D_N and D_i . For example, there are four core switches between Calgary and Montreal nodes in Fig. 8, namely Calgary, Chicago, Toronto and Montreal (the lightpath bypasses the Ottawa switch). *H* is the number of core hops. As all data centers are connected directly to the core network by optical links: H = 1. All core switches are Ciena OADM Core Optical Transport System model with a nominal power of 250 W [26].

We consider two scenarios: i) the slice is located at the core node (Montreal, Canada), and ii) the slice is distributed on multiple edge nodes. The core is powered by hydroelectricity and has a large capacity that enables it to maintain service in the network. Edge nodes are powered by sun and/or wind and have smaller capacity. Six VMs are allocated for Geochronos slice, together with six flows among them (see the numbered arrows in Fig. 9) and two external flows.

All local links in each data center have the same bandwidth capacity of 1 Gbps, and a connection between each pair of data centers is 10 Gbps. Assuming that all links (server-server or server-switch) consumes the same power ∂ whose value is estimated to be about 20 W [27].

Three connection schemes are used for node architecture: Tree, Fat-tree and BCube, as shown in Fig. 10 in which the red squares represent the hosts and the blue rounds represent the switches. Note that there is a gateway in Figs. 10B and 10C (Fat-tree and BCube) connecting to all top-level switches. The power consumption p(i, j) of each link is calculated as follows [28].

TABLE 1 Experiment Configuration

| Simulation Parameter | Description |
|-----------------------------------|-------------------|
| Physical server CPU capacity | 48 cores / server |
| Physical server memory capacity | 160 GB / server |
| Number of servers per data center | 16 |
| Number of datacenters | 5 |
| Core switch power | 10kWh / switch |
| VM CPU requirement | 8 cores / VM |
| VM memory requirement | 4 GB / VM |
| Number of VMs required | 36 / request |



Fig. 10. Tree, Fat-tree and BCube architecture.

For Tree architecture with two layers (Fig. 10A):

$$p(i,j) = \begin{cases} 0 & if \ i = j \\ \partial & if \left\lfloor \frac{i}{p_0} \right\rfloor = \left\lfloor \frac{j}{p_0} \right\rfloor \\ 3\partial & else, \end{cases}$$
(13)

where p_{0} is the number of switch ports (four ports in Fig. 10A).

For Fat-tree architecture with 4-port switches (Fig. 10B):

$$p(i,j) = \begin{cases} 0 & if \ i = j \\ \partial & if \left\lfloor \frac{2i}{p_0} \right\rfloor = \left\lfloor \frac{2j}{p_0} \right\rfloor \\ 3\partial & if \left\lfloor \frac{2i}{p_0} \right\rfloor \neq \left\lfloor \frac{2j}{p_0} \right\rfloor \text{ and } \left\lfloor \frac{4i}{p_0^2} \right\rfloor = \left\lfloor \frac{4j}{p_0^2} \right\rfloor \\ 5\partial & \left\lfloor \frac{4i}{p_0^2} \right\rfloor \neq \left\lfloor \frac{4j}{p_0^2} \right\rfloor. \end{cases}$$
(14)

For BCube architecture (Fig. 10C):

$$p(i,j) = \begin{cases} 0 & \text{if } i = j \\ 2\partial \times hamming \ (addr(i), addr(j)) - 1 & \text{if } i \neq j. \end{cases}$$
(15)

Fig. 11 presents a comparison of three edge nodes in terms of carbon emissions (and indirectly power consumption) when the Geochronos slice is mapped to underlying infrastructure using the traditional virtual network embedding algorithm [32]. Indeed, a number of heuristic algorithms



Fig. 11. Carbon emissions resulting from Geochronos flows assignment for three intra data center interconnect architectures (first scenario).

TABLE 2 Summary of Virtual Node Consolidation

| Architecture | 1-Degree consolidation | 2-Degree consolidation | 3-Degree consolidation |
|--------------|---------------------------|------------------------|---------------------------|
| Tree | 11.3 | 6 | 0 |
| Fat-tree | 10.6 | 5.4 | 0 |
| BCube | 12 | 9.7 | 3.3 |

have been proposed to resolve the virtual network mapping problem, such as hierarchical node mapping [38], survival mapping [39], and distributed mapping [40], among the others (a nearly complete list of existing algorithms can be found in [41]). However the coordinated mapping algorithm proposed in [32] is the widest used and strongest in terms of node and link mapping coordination, which is often used as a building block to establish the other algorithms.

As shown, the Fat-tree architecture consumes the largest amount of energy since it has the largest number of network levels and intermediate switches. The BCube architecture achieves the best performance in terms of energy consumption as a result of its multipath routing feature.

Table 2 depicts the average number of virtual node consolidation operations (over 36 allocations) on the three architectures. The actual number of consolidations is case-

Fig. 12. Carbon emissions resulting from Geochronos flow assignment using traditional IP routing and the proposed solution (first scenario).

Fig. 13. Carbon emissions resulting from Geochronos flow assignment using traditional IP routing and the proposed solution (second scenario).

by-case dependent, following the availability capacity of physical servers and links. The highest node degree in the slice is of the portal VM (3-degree). Table 2 suggests that BCube may allow the largest number of consolidation, which is explained by the highest degree of its physical nodes among these three architectures (Fig. 10).

Fig. 12 gives a comparison of the proposed solution and the traditional virtual network embedding algorithm when assigning Geochronos flows in a data center having Tree and Fat-tree architectures respectively. In the Tree architecture, the proposed solution may reduce about 40 percent of network power consumption compared to the traditional algorithm. This number may increase up to 60 percent in the Fat-tree architecture. In reality, if only 1-degree virtual nodes are considered in the consolidation process, network power consumption is reduced about 26 percent in Tree architecture, and about 40 percent in Fat-tree architecture. If only N-degree (N > 1) virtual nodes are consolidated, network power consumption is reduced about 13 percent in Tree architecture, and about 20 percent in Fat-tree architecture.

Fig. 13 shows carbon emissions of the traditional mapping algorithm and the proposed solution when the Geochronos slice is distributed across multiple data centers in the network (the second scenario). It is worth noting that all data centers in the network use the same intra interconnect architecture. Similarly to Fig. 12, the proposed solution clearly achieves a better performance compared to the traditional algorithm in the Tree and Fat-tree architecture. It is worth noting that this improvement could still increase when we compare these two algorithms in a larger scale network. Indeed, the number of data centers involved in our experiment is still small (five nodes) and the number of links between these data centers is also small (six lighpaths).

6 CONCLUSION

In this paper we faced a problem of optimal flow assignment for virtual slice based on software-defined paradigms, in order to provide environment-aware recommendation services for cloud operators. This is a piece of work in our global green cloud optimization framework taking into account computing, network, and storage resources. Simulations show that the proposed solution achieves different performance when intra data center network topology is considered in our optimization model. In other words, having a good intra data center network topology plays an important role in reducing network carbon footprint.

To simplify the calculation in our experiments, we have assumed that all local links have the same power, which is not necessarily true in reality. However, it does not change the problem formulation as each link is associated with a variable representing power consumption when sending a bit. Also, the power consumption of each local switch may not be proportional with processing capacity.

The synchronization of Network Manager and GSN Controller is also an issue to be investigated, in order to prevent situations in which a flow would be established when VMs had been migrated.

Our future work will address the global optimization problem and quality of services hosted by the GSN.

ACKNOWLEDGMENTS

The authors thank CANARIE for funding the GSN project, and all GSN partners for their contribution in this project. Research on Telco-grade cloud applications is supported by Ericsson, Inocybe Technologies and NSERC.

REFERENCES

- [1] F. Schneider, "The Jevons paradox and the myth of resource efficiency improvements," J. Cleaner Prod., vol. 18, no. 6, pp. 600-602, 2010
- The Climate Group, "SMART2020: Enabling the low carbon econ-[2] omy in the information age," London, U.K., Rep. Behalf Global eSustainability Initiative (GeSI), 2008.
- M. Lemay, K. K. Nguyen, B. St Arnaud, and M. Cheriet, "Toward [3] a zero-carbon network: Converging cloud computing and network virtualization," IEEE Internet Comput., vol. 16, no. 6, pp. 51-59, Nov. 2012.
- [4] N. McKeown, T. Anderson, H. Balakrishnan, "OpenFlow: Enabling innovation in campus networks," ACM SIGCOMM Comput. Commun. Rev., vol. 38, no. 2, pp. 69-74, 2008.
- K. K. Nguyen, M. Cheriet, M. Lemay, V. Reijs, A. Mackarel, and [5] A. Pastrama, "Environmental-aware virtual data center network," J. Comput. Netw., vol. 56, no. 10, pp. 2538-2550, 2012.
- The GreenStar network project [Online]. Available: http://green-[6] starnetwork.com, 10/2014.
- [7] F. F. Moghaddam, M. Cheriet, and K. K. Nguyen "Low carbon virtual private clouds," in Proc. IEEE CloudCom, 2011, pp. 259-266.

- A. Daouadji, K. K. Nguyen, M. Lemay, and M. Cheriet, [8] "Ontology-based resource description and discovery framework for low carbon grid networks," in Proc. IEEE SmartGridComm, 2010, pp. 477-482
- [9] K. K. Nguyen, M. Cheriet, and M. Lemay, "Powering a data center network via renewable energy: A green testbed," IEEE Internet Comput., vol. 17, no. 1, pp. 40–49, Jan./Feb. 2013. [10] Amazon Inc. Amazon virtual proviate cloud (Amazon VPC)
- [Online]. Available: http://aws.amazon.com/vpc, 10/2014.
- [11] R. Niranjan Mysore, A. Pamboris, N. Farrington, N. Huang, P Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat, "PortLand: A scalable fault-tolerant layer 2 data center network fabric," in Proc. ACM SIGCOMM, 2009, pp. 39-50.
- [12] 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," in Proc. SIGCOMM, 2009,
- pp. 63–74.
 [13] L. Gyarmati and T. A. Trinh, "Scafida: A scale-free network inspired data center architecture," ACM SIGCOMM Comput. Commun. Rev., vol. 40, no. 5, pp. 4–12, 2010.
- [14] 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.
- [15] W. Fang, X. Liang, S. Li, L. Chiaraviglio, and N. Xiong, "VMPlanner: Optimizing virtual machine placement and traffic flow routing to reduce network power costs in cloud data centers," J. Comput. Netw., vol. 57, no. 1, pp. 179-196, 2013.
- [16] B. Heller, S. Seetharaman, and P. Mahadevan, "ElasticTree: Saving energy in data center networks," in Proc. 7th USENIX Symp. Netw. Syst. Des. Implementation, 2010, p. 17.
- [17] M Al-Fares, S. Radhakrishnan, and B. Raghavan, "Hedera: Dynamic flow scheduling for data center networks," in Proc. 7th USENIX Symp. Netw. Syst. Des. Implementation, 2010, p. 9.
- [18] "Emission factors from canada's greenhouse gas inventory," Environment Canada. (2010) [Online]. Available: http://ec.gc.ca.
- [19] M Poikselkä and G. Mayer, The IMS: IP Multimedia Concepts and Services. Hoboken, NJ, USA: Wiley, 2009. [20] R. Banner and A. Orda, "Multipath routing algorithms for conges-
- tion minimization," IEEE/ACM Trans. Netw., vol. 15, no. 2, pp. 413–424, Apr. 2007. [21] N. Gude, T. Koponen, J. Pettit, B. Pfaff, and M. Casado, "NOX:
- Towards an operating system for networks," ACM SIGCOMM Comput. Commun. Rev., vol. 38, no. 3, pp. 105–110, 2008. [22] J. Baliga, R. Ayre, K Hinton, and W. V. Sorin, "Energy consump-
- tion in optical IP networks," J. Lightw. Technol., vol. 27, no. 13, pp. 2391–2403, Apr. 2009.
- [23] C. Guo, H. Wu, K Tan, L. Shi, Y. Zhang, and S. Lu, "Dcell: A scalable and fault-tolerant network structure for data centers," in *Proc. SIGCOMM*, 2008, pp. 75–86. [24] R. Curry, C. Kiddle, and R. Simmonds, "An on-line collaborative
- data management system," in Proc. IEEE Gateway Comput. Environ. Workshop, 2010, pp. 1-10.
- [25] S Figuerola, M. Lemay, V. Reijs, M. Savoie, and B. St Arnaud, "Converged optical network infrastructures in support of future internet and grid services using IaaS to reduce GHG emissions," J. Lightw. Technol., vol. 27, no. 12, pp. 1941-1946, May 2009.
- [26] Product data sheet. Ciena Con gurable OADM for CoreStream Agility Optical Transport System. (2008) [Online]. Available: http://www.ciena.com/files/Configurable_OADM_A4_DS.pdf
- [27] C. Gunaratne, K Christensen, and B. Nordman, "Managing energy consumption costs in desktop PCs and LAN switches with proxying, split TCP connections, and scaling of link speed," Int. J. Netw. Manage., vol. 15, no. 5, pp. 297–310, 2005.
- [28] X Meng, V. Pappas, and L. Zhang, "Improving the scalability of data center networks with traffic-aware virtual machine placement," in Proc. IEEE INFOCOM, 2010, pp. 1-9.
- [29] Cisco data center infrastructure 2.5 desin guide [Online]. Available: http://www.cisco.com/univercd/cc/td/doc/solution/dcidg21. pdf, 12/2007.
- [30] C. Hopps, "Analysis of an equal-cost multi-path algorithm," IETF RFC 2992, 2000.
- [31] R. Zhang-Shen and N. McKeown, "Designing a predictable internet backbone network," presented at the ACM HotNets, San Diego, CA, USA, 2004.

- [32] M Chowdhury, M. R. Rahman, and R. Boutaba, "ViNEYard: Virtual network embedding algorithms with coordinated node and link mapping," *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 206–219, Feb. 2012.
- [33] S Su, Z. Zhang, X. Cheng, and Y. Wang, "Energy-aware virtual network embedding through consolidation," in *Proc. IEEE INFO-COM Workshop Comput. Commun. Workshops*, 2012, pp. 127–132.
 [34] M. F. Zhani, Q Zhang, G. Simon, and R. Boutaba, "VDC Planner:
- [34] M. F. Zhani, Q Zhang, G. Simon, and R. Boutaba, "VDC Planner: Dynamic migration-aware virtual data center embedding for clouds," in *Proc. IFIP/IEEE Integr. Netw. Manage. Symp.* (IM)., Ghent, (Belgium), 2013, pp. 18–25.
- [35] A. P. Bianzino, C. Chaudet, D. Rossi, and J. Rougier, "A survey of green networking research," *IEEE Commun. Surv. Tutorials*, vol. 14, no. 1, pp. 3–20, 1st Quarter 2012.
- [36] R. Bolla, F. Davoli, R. Bruschi, K. Christensen, F. Cucchietti, and S. Singh, "The potential impact of green technologies in next-generation wireline networks: Is there room for energy saving optimization?" *IEEE Commun. Mag.*, vol. 49, no. 8, pp. 80–86, Aug. 2011.
- [37] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: Substrate support for path splitting and migration," ACM SIGCOMM Comput. Commun. Rev., vol. 38, no. 2, pp. 17–29, 2008.
- [38] T. Ghazar and N. Samaan, "A hierarchical approach for efficient virtual network embedding based on exact subgraph matching," in *Proc. IEEE Global Telecommun. Conf.*, 2011, pp. 1–6.
 [39] G. Sun, H. Di, H. Yu, L. Li, and V. Anand, "The framework and
- [39] G. Sun, H. Di, H. Yu, L. Li, and V. Anand, "The framework and algorithms for the survivable mapping of virtual network onto a substrate network," *IETE Tech. Rev.*, vol. 28, no. 5, pp. 381–391, 2011.
- [40] I. Houidi, W. Louati, D. Zeghlache, "A distributed virtual network mapping algorithm," in *Proc. IEEE Int. Conf. Commun.*, 2008, pp. 5634–5640.
- [41] A. Fischer, J. F Botero, M. Till Beck, H. de Meer, and X. Hesselbach "Virtual network embedding: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 1888–1906, Feb. 2013.

Kim Khoa Nguyen received the PhD degree in electrical and computer engineering from Concordia University. He is a research fellow at the École de Technologie Supérieure, University of Quebec. He is a key architect of the GreenStar Network project and leading R&D in the Green Sustainable Telco Cloud project. His expertise includes green ICT, cloud computing, smart grid, high-speed routing architecture, wireless networks, and smart city. He has published more than 40 papers, and holds a number of patents in the field.

Mohamed Cheriet received the PhD degree in computer science from the University of Pierre et Marie Curie. He has been a professor in the Department of Automated Manufacturing Engineering at ÉTS since 1992. He has focused on cloud computing and network virtualization in the area of green/sustainable ICT. In addition, he is an expert in computational intelligence, pattern recognition, and machine learning. He has published more than 300 technical papers in the field. He is a Canada Research chair Tier 1 on Smart

Sustainable Eco-Cloud. He cofounded the Laboratory for Imagery, Vision, and Artificial Intelligence (LIVIA) and founded and directs the Synchromedia Consortium (Multimedia Communication in Telepresence) at the University of Quebec.

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.