Greenslater: On Satisfying Green SLAs in Distributed Clouds

Ahmed Amokrane, Rami Langar, Mohamed Faten Zhani, Raouf Boutaba, *Fellow, IEEE*, and Guy Pujolle

***Abstract*—With the massive adoption of cloud-based services, high energy consumption and carbon footprint of cloud infrastructures have become a major concern in the IT industry. Consequently, many governments and IT advisory organizations have urged IT stakeholders (i.e., cloud provider and cloud customers) to embrace green IT and regularly monitor and report their carbon emissions and put in place efficient strategies and techniques to control the environmental impact of their infrastructures and/or applications. Motivated by this growing trend, we investigate, in this paper, how cloud providers can meet Service Level Agreements (SLAs) with green requirements. In such SLAs, a cloud customer requires from cloud providers that carbon emissions generated by the leased resources should not exceed a fixed bound. We hence propose a resource management framework allowing cloud providers to provision resources in the form of Virtual Data Centers (VDCs) (i.e., a set of virtual machines and virtual links with guaranteed bandwidth) across a geo-distributed infrastructure with the aim of reducing operational costs and green SLA violation penalties. Extensive simulations show that the proposed solution maximizes the cloud provider's profit and minimizes the violation of green SLAs.**

***Index Terms*—Green SLA, virtual data center, distributed cloud, energy efficiency.**

I. Introduction

WITH the rapid development of cloud computing technologies, data centers have become a popular platform for delivering large-scale online services such as content delivery, social networking and e-commerce. However, the rapid expansion of cloud infrastructures in recent years have also raised serious concerns regarding their energy consumption and environmental impact. Recent reports [1] have revealed that the Information and Communication Technologies (ICT) account for 3% of the world's carbon emissions. Data centers by themselves accounts for about 10% of the ICT emissions worldwide.

Manuscript received March 23, 2015; revised May 27, 2015; accepted May 28, 2015. Date of publication June 5, 2015; date of current version September 4, 2015. This work was partially supported by the CNRS (France) WINDS project (Grant no. 25995), the NSERC (Canada) SAVI project (Grant no. NETGP394424-10), and the EU FP7 IRSES MobileCloud project (Grant no. 612212). The associate editor coordinating the review of this paper and approving it for publication was L. Granville.

A. Amokrane is with the CoESSI, 78360 Montesson, France (e-mail: ahmed.amokrane@lip6.fr).

R. Langar and G. Pujolle are with the Laboratoire d'Informatique de Paris 6 (LIP6), University of Pierre and Marie Curie (UPMC), 75005 Paris, France (e-mail: rami.langar@lip6.fr; guy.pujolle@lip6.fr).

M. F. Zhani is with the Department of Software and IT Engineering École de Technologie Supérieure, Montréal, QC H3C 1K3, Canada (e-mail: mfzhani@etsmtl.ca).

R. Boutaba is with the University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: rboutaba@uwaterloo.ca).

Digital Object Identifier 10.1109/TNSM.2015.2440423

Motivated by these observations, the ICT sector is witnessing an upward move towards greening cloud infrastructures and services driven by several governmental regulations and marketing considerations. For instance, a recent study [2] showed that the firms' value would decrease significantly if it has high carbon footprint or even if it withholds information about its carbon emission rates. As a result, many IT companies are voluntarily disclosing their carbon emissions and regularly reporting their efforts towards deploying environmental-friendly solutions and services [3]. At the same time, governments are imposing taxes on carbon emissions in the hopes of pushing further this shift towards the adoption of green sources of energy and the reduction of carbon footprint [4].

In current cloud environments, there are mainly two stakeholders: (1) *cloud providers* (CPs) that typically own distributed infrastructures and lease their resources in an on-demand manner to different *Service Providers* (SPs); (2) SPs use these resources to deploy their services and offer them to Internet end-users. Recent research proposals and cloud offerings [5] are advocating to offer these resources in the form of Virtual Data Centers (VDCs), i.e., a set of VMs and virtual links with guaranteed bandwidth.

Typically, CPs are responsible for allocating resources for VDCs across their distributed clouds with the goal of minimizing operational costs and maximizing the infrastructure environmental friendliness by increasing the usage of green energy [6]. However, recently, SPs were also required to take into account environmental objectives and ensure that their services are produced with the smallest carbon footprint. Many advisory boards and commissions (e.g., Open Data Center Alliance [7] and SLA Expert Subgroup of the Cloud Selected Industry Group of the European Commission [8]) are pushing towards defining green SLAs in which SPs require their CPs to limit the carbon emissions generated on their behalf. Recently, some research works advocated providing Green SLAs in the context of HPC clouds [9]–[13].

Typically, the green SLA terms require either to limit the carbon emissions generated by SPs services [9]–[12] or to set a minimum amount of renewable power to be consumed by the resources allocated to SPs [13], [14]. However, these proposals do not consider the allocation of network resources (virtual links) and aim only to allocate VMs within a single data center.

In this paper, we investigate how a CP can meet an SLA with green requirements. In particular, we consider SLAs that specify a limit on the carbon emission generated by each service provider's VDC. We, hence, propose Greenslater, a holistic framework that orchestrates the provisioning and the resource optimization for the multiple VDCs deployed across a distributed infrastructure. From the CP's point of view, the objective is to maximize revenue while minimizing operational costs and the potential green SLA violation penalties. Greenslater takes advantage of the variability in space and time of the available renewables and electricity prices in different data centers to reduce the carbon footprint and costs. It provisions VDCs and dynamically optimize resource allocation over time while fulfilling the green SLA terms. Through extensive simulations, we show that the proposed framework maximizes the CP's profit and also the usage of renewable power while minimizing SLA violation cost.

The remainder of this paper is organized as follows. Section II surveys the related works. Section III defines green SLAs and presents the proposed management framework. The mathematical formulation of the VDC embedding problem across distributed infrastructures that considers green SLAs is then presented in Section IV. Section V gives a detailed description of the proposed algorithms for VDC admission control and dynamic resource allocation and optimization. Section VI discusses simulation setup and results. Finally, we conclude the paper in Section VII.

II. Related Work

In the last few years, a large body of work has addressed the problem of reducing energy consumption and carbon footprint in cloud environments. In the following, we first survey the literature on green management in the cloud and then we focus on the proposals that advocated implementing green SLAs between cloud and service providers.

A. Green Management in the Cloud

Recently, several systems have been proposed to map VDCs onto a single data center with the goal of reducing energy consumption. For instance, Zhani *et al.* [15] proposed VDC Planner, a resource management framework that leverages dynamic VM migration to increase CP's revenue while minimizing energy consumption. Unfortunately, these solutions are designed to manage a single data center and hence do not consider the variability over time and between different locations of the electricity prices and the availability of green sources of energy.

A plethora of techniques have been also proposed to allocate resources across geographically distributed data centers in order to reduce energy costs [16]–[18], minimize the infrastructure's carbon footprint [19], [20] or achieve both objectives [6], [21], [22]. For instance, Xin *et al.* [23] proposed an algorithm that uses *minimum k-cut* to split a VDC request into partitions before assigning them to different locations so as to balance the load among different data centers. In [6], we proposed Greenhead, a framework for VDC embedding across distributed infrastructures that aims at maximizing cloud providers' revenue while cutting down the carbon footprint of the infrastructure. Unfortunately, the solutions above use static mapping and do not perform any dynamic resource optimization over time. They also do not consider green SLAs and hence do not guarantee any limit on carbon emissions of the resources leased by each SP.

B. Green SLA in the Cloud

Green SLAs stipulate that SPs are able to require their cloud providers to guarantee that the leased resources are environmental friendly. In other words, SPs can explicitly specify green constraints like, for instance, an upper limit on carbon emissions produced by the resources they lease.

Providing green SLAs has been originally proposed back in 2010 by Laszewski *et al.* [9] and then quickly adopted and supported in several research works [10]–[14], [24]–[26]. For instance, Haque *et al.* [13] considered an SLA that specifies the proportion of green power that the HPC provider should use to run the job (e.g., *x% of the job should run on green power*). Hence, the HPC provider has to pay a penalty to SPs if the green terms of the SLA are not satisfied. Similarly, Wang *et al.* [24] proposed an approach where SPs can define SLA constraints for their submitted tasks to limit the carbon emissions and the consumed power. In this case, the goal, from the CP's perspective, is to schedule parallel tasks such that the green SLAs are satisfied. Klingert *et al.* [25] proposed that data center providers consider CO2 per task or resource (in kgCO2) and the yearly average PUE as metrics to specify SPs' requirements. In a case study, the authors compared three types of SLA: (i) a standard SLA (Full Power) that does not address energy consumption at all but prioritizes performance and time; (ii) a relaxed SLA that requires key indicators to be within relaxed boundaries, and (iii) an energy-aware SLA that uses tight energy ranges for each job. The results at a small scale show significant energy saving and reduced QoS violations.

Hasan *et al.* [14] proposed a framework for defining Green SLA between the SPs (SaaS providers) and the CPs (IaaS providers). The Green SLAs define terms related to the total amount of renewable energy in percentage that should be consumed by the data center. The goal of the CP in this case is to satisfy these terms by purchasing renewable power and finding a good tradeoff between profit and SLA violation penalty. Hence, the CP negotiates with electricity providers short term contracts that would satisfy the renewable power demand based on SPs' requirements, while capping expenditures to a limited budget. To do so, the authors proposed an optimization module that uses linear programming techniques along with forecasting models that predict renewable power availability and cost.

It is worth noting that existing works such as [10], [27] proposed renegotiation of the SLA terms between the CP and the SP. The idea is that CPs incentivize SPs to relax some of the QoS constraints so as to reduce the energy consumption and/or carbon footprint. For instance, SPs can relax the constraint on the execution time of an HPC job or task to allow the CP to run it during periods of time where the renewable power is available.

The main limitation of the solutions described above is that they do not consider bandwidth requirements between VMs and they are designed to manage resources within a single data center. Our work considers a more general scenario with multiple data centers and where the network requirements are explicitly specified in the VDC request.

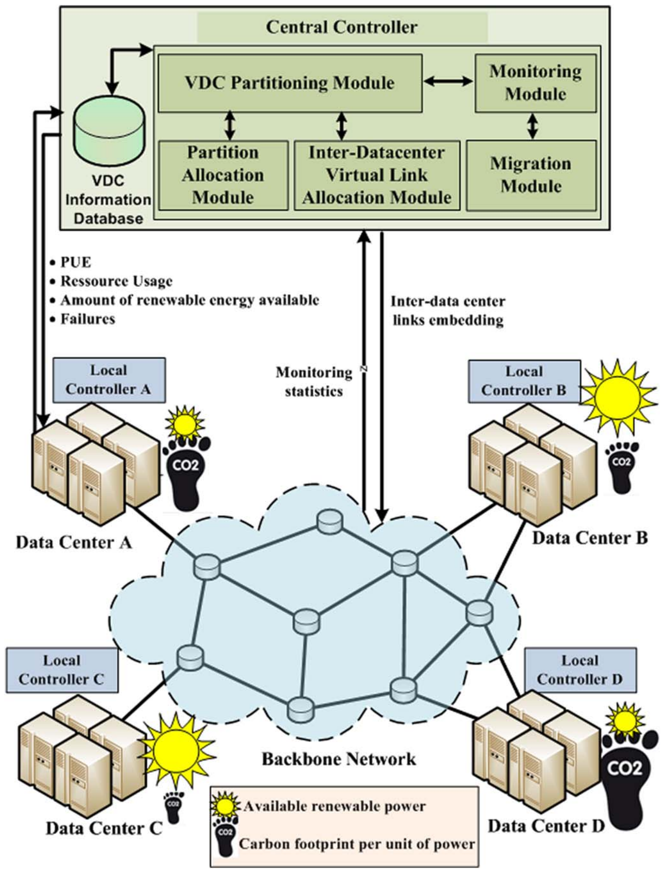


Fig. 1. Proposed framework.

III. System Architecture

In this section, we present the design architecture of the proposed solution and we discuss the definition of the Green SLA terms and how to enforce them in a distributed environments.

A. Architecture Overview

As shown in Fig. 1, we consider a distributed infrastructure consisting of multiple data centers located in different regions and interconnected through a backbone network. The entire infrastructure (including the backbone network) is assumed to be owned and managed by the same CP.

SPs send VDC request specifications to the CP, which has the responsibility of allocating the required resources. Naturally, the CP will make use of its distributed infrastructure with the objective of maximizing its revenue and minimizing energy costs and carbon footprint; this is where our proposed management framework, Greenslater, comes into play. Greenslater is composed of two types of management entities: i) a Central Controller that manages the entire infrastructure and ii) a Local Controller deployed in each data center to manage the data center's internal resources (i.e., resource allocation for VMs and virtual links inside the data center).

The central controller consists of a number of components. The *Partitioning Module* is in charge of dividing a VDC request into partitions such that inter-partition bandwidth is minimized. The *Partition Allocation Module* is then responsible for running an admission control algorithm for every received VDC request, and assigns the partitions, in case of accepted requests, to data centers based on run-time statistics collected by the monitoring module and the estimation of available renewable power. The *Inter-data center Allocation Module* allocates resources for the virtual links spanning the backbone network. Finally, the *Migration Module* dynamically relocates VDC partitions in such a way to follow renewables and reduce the carbon footprint. The *Monitoring Module* monitors and collects information about the status of physical and virtual infrastructures and stores them into *VDC Information Base*.

B. Green SLA Definition

As stated earlier, SPs have not only to specify resource requirements but also constraints on the carbon emissions generated by the CPs while hosting their VDC. Specifically, green terms in the SLA specify the limit on carbon emissions that the CP is allowed to generate to accommodate the VDC request during a period of time called hereafter the *reporting period*. The reporting period can be for instance the a billing period [7].

To enforce green SLAs, the CP should compute the carbon footprint of each VDC request. To do so, we use two metrics: (1) carbon emission per unit of bandwidth (tonCO2/Mbps) and (2) carbon emission per core (tonCO2/Core). These metrics are chosen because the bandwidth and the CPU are the major factors that determine the power consumption in data centers and they are already considered in industry. For instance, Akamai reports annually its carbon emission in CO_2 per gigabyte of data delivered (tonCO2/Gbps), Verizon reports its carbon emissions per terabyte of transported data across its network.

As the carbon footprint is computed for each VDC, the SLA is enforced at the end of each reporting period. In case of violation of the green terms (i.e., the carbon footprint for the VDC is higher than the limit specified in the SLA), the CP is required to pay a penalty (a.k.a. credit). The penalty can a percentage of the SP's bill that can go up to 100% for some providers such as Rackspace [28]. It becomes then critical to design effective VDC embedding algorithms that minimize this penalty.

IV. Problem Formulation

In this section, we formally define the VDC embedding problem across multiple data centers as an Integer Linear Program (ILP). For ease of explanation, Table I describes the notations used in our ILP model. We assume that time is divided into slots. The metrics characterizing each data center (e.g., Power Usage Effectiveness (PUE), electricity price, availability of renewable power) are measured at the beginning of each time slot and are considered constant during the corresponding time slot. Moreover, we assume that the CP reports its carbon emissions periodically every T time slots. We denote by $T^k = [t_b^k, t_e^k]$ the k^{th} reporting period, where t_b^k and t_e^k are its beginning and end time slots, respectively.

The physical infrastructure is represented by a graph $G(V \cup W, E)$, where V denotes the set of data centers and W the set of nodes of the backbone network. The set of edges E represents the physical links in the backbone network. Each link is characterized by its bandwidth capacity $bw(e)$ and propagation delay $d(e)$.

TABLE I
TABLE OF NOTATIONS

Notation	Meaning
PUE_t^i	PUE of data center i
ζ_t^i	Electricity price from the grid in data center i
η_t^i	On-site renewable power cost in data center i
z_{ik}^j	A boolean variable indicating whether data center i satisfies the location constraint of VM k of VDC j
x_{ik}^j	A boolean variable indicating whether VM k is assigned to data center i
$f_{e,e'}$	a boolean variable indicating whether the physical link $e \in E$ is used to embed the virtual link $e' \in E^j$
$P_{i,Net}^t$	Amount of power consumed by the network elements in data center i during time slot t
$P_{i,Serv}^t$	Amount of power consumed by the servers in data center i during time slot t
RN_t^i	Onsite renewable power generated in data center i during time slot t
C_t^i	Carbon footprint per unit of power from the power grid in data center i
$P_{i,L}^t$	Consumed on-site renewable power in data center i during time slot t
$P_{i,D}^t$	Consumed power from the grid in data center i during time slot t
c_j	Upper limit of carbon that should not be exceeded by the CP to accommodate VDC j
σ^{cpu}	Price per unit of cpu (core)
σ^b	Price per unit of bandwidth
σ_p	Cost per unit of bandwidth in the backbone network
p	Proportion of SP's bill to refund by the CP if the SLA is violated
$\mathcal{C}_{Serv}^t$	Total carbon emission due to the servers during time slot t
$\mathcal{C}_{Net}^t$	Total carbon emission due to the network during time slot t

A VDC request j is represented by a graph $G^j(V^j, E^j)$. The arrival time and lifetime of the request j are denoted by t^j and $\mathcal{T}^j$, respectively. Each vertex $v \in V^j$ corresponds to a VM, characterized by its CPU, memory and disk requirements. Each edge $e \in E^j$ is a virtual link that connects a pair of VMs, which is characterized by its bandwidth demand $bw(e)$ and propagation delay $d(e)$. Furthermore, each VDC j may have a constraint on carbon emissions per reporting period T, which is defined by the variable c_j. We assume the revenue generated by VDC j, denoted by $\mathcal{R}^j$, to be proportional to the amount of resources (CPU, memory and disk) and bandwidth required by its VMs and links, and inversely proportional to the carbon limit c_j. Let R denote the different types of resources offered by each node (i.e., CPU, memory and disk). The revenue generated by VDC j per time slot can be written as follows:

$$\mathcal{R}^j = \left(\sum_{v \in V^j} \sum_{r \in R} \left(C^r(v) \times \sigma^r \right) + \sum_{e' \in E^j} bw(e') \times \sigma^b \right) \times \frac{\gamma}{c_j} \quad (1)$$

where $C^r(v)$ is the demand of VM v belonging to VDC j in terms of resource $r \in R$, and σ^r and σ^b are unit price of resource r and bandwidth, respectively, and γ is a weighting factor that determines the importance of the green constraints in the pricing.

Furthermore, a VM $v \in V^j$ may have a location constraint. That is, it can only be embedded in a particular set of data centers. To model this constraint, we define a binary variable z_{ik}^j, indicating whether or not a VM k of VDC j can be embedded in a data center i.

The problem of embedding VDC requests in a distributed infrastructure of data centers should be solved dynamically over time. In fact, the decision of embedding VMs in different data centers is modified at the beginning of every time slot in such a way to follow the renewables. Thus, for each VDC request j, and during each time slot $t \in [t^j, t^j + \mathcal{T}^j]$, the central controller should:

- Assign each VM $k \in V^j$ to a data center. Hence, we define the decision variable $x_{ik}^{j,t}$ as:

$$x_{ik}^{j,t} = \begin{cases} 1 & \text{If the VM } k \text{ of the VDC } j \text{ is assigned} \\ & \text{to data center } i \text{ during slot } t \\ 0 & \text{Otherwise.} \end{cases}$$

- Embed every virtual link either in the backbone network if it connects two VMs assigned to different data centers or within the same data center, otherwise. To do so, we define the virtual link allocation variable $f_{e,e'}^t$ as:

$$f_{e,e'}^t = \begin{cases} 1 & \text{If the link } e \in E \text{ is used to embed} \\ & \text{the virtual link } e' \in E^j \text{ during slot } t \\ 0 & \text{Otherwise.} \end{cases}$$

As a CP can reject a request due to shortage in resources or too tight constraints (delay, location), we define a binary variable X_j, which indicates whether the VDC request j is accepted for embedding or not defined as follows:

$$X_j = \begin{cases} 1 & \text{If } \sum_{t \in T^k} \sum_{i \in V} \sum_{k \in V^j} x_{ik}^{j,t} \geq 1 \\ 0 & \text{Otherwise.} \end{cases}$$

Finally, the ultimate objective of the CP when embedding VDC requests during any reporting period T^k is to maximize its profit defined as the difference between the revenue (denoted by $\mathcal{R}_k$) and the total embedding cost plus penalty cost, which consists of the embedding cost in the data centers (denoted by $\mathcal{D}_k$), the migration cost (denoted by $\mathcal{M}_k$) the embedding cost in the backbone network $\mathcal{B}_k$ and the penalty cost $\mathcal{P}_k$. Hence, our problem can be formulated as an ILP with the following objective function:

$$Maximize \ \ \mathcal{R}_k - (\mathcal{D}_k + \mathcal{B}_k + \mathcal{M}_k + \mathcal{P}_k) \quad (2)$$

Subject to:

$$x_{ik}^{j,t} \leq z_{ik}^j, \ \forall k \in V^j, \forall i \in V, \forall t, \quad (3)$$

$$\sum_{i \in V} x_{ik}^{j,t} = X_j, \ \forall k \in V^j, \forall j \in Q_t, \forall t \quad (4)$$

$$\sum_{e' \in E^j} f_{e,e'}^t \times bw(e') \leq bw(e), \ \forall e \in E, \forall t \quad (5)$$

$$\sum_{e \in E} f_{e,e'}^t \times d(e) \leq d(e'), \ \forall e' \in E^j, \forall t \quad (6)$$

$$f_{e_1,e'}^t - f_{e_2,e'}^t = x_{dst(e_1)dst(e')}^t - x_{src(e_2)src(e')}^t, \\ \forall e_1, e_2 \in E, dst(e_1) = src(e_2), \ \forall e' \in V^j, \forall t \quad (7)$$

where Q_t is the set of VDC requests being embedded during time slot t, $src(e)$ and $dst(e)$ denote the source and destination of link e, respectively. Equation (3) guarantees location constraint satisfaction. Equation (4) depicts that a VM is assigned to at most one data center. Equation (5) guarantees that link capacities are not exceeded in the backbone network, whereas (6) guarantees that delay requirements of virtual links are satisfied. Equation (7) denotes the flow continuity constraint.

The revenue for a reporting period T^k is given by:

$$\mathcal{R}_k = \sum_{t \in T^k} \sum_{j \in Q_t} \mathcal{R}^j \times X_j \tag{8}$$

Let us now focus on the expression of the embedding costs $\mathcal{D}_k$, $\mathcal{B}_k$, $\mathcal{M}_k$ and $\mathcal{P}_k$ in the data centers, the backbone network and penalty, respectively. Recall that these costs are part of the objective function.

- The cost of embedding in the data centers

In this work, we evaluate the request embedding cost in the data centers in terms of energy costs.

The total amount of consumed power in data center i is given by:

$$P_i^t = \left(P_{i,Net}^t + P_{i,Serv}^t\right) \times PUE_i^t \tag{9}$$

where $P_{i,Serv}^t$ and $P_{i,Net}^t$ are the power consumed by servers and network elements, respectively, and PUE_i^t is the power usage effectiveness of data center i during time slot t, which is used to compute the power consumed by supporting systems such as the cooling system. Note that this power consumption depends mainly on the local allocation scheme in each data center.

The mix of power used in data center i is given by:

$$P_i^t = P_{i,L}^t + P_{i,D}^t \tag{10}$$

where $P_{i,L}^t$ and $P_{i,D}^t$ denote, respectively, the consumed on-site renewable power and the amount of purchased power from the grid during time slot t. Note that $P_{i,L}^t$ should not exceed the amount of produced power, which is captured by $P_{i,L}^t \leq RN_i^t$, where RN_i^t is the amount of onsite renewable power generated in data center i, during time slot t, expressed in kW.

Hence, the total embedding cost in all data centers (expressed in \$) can be written as:

$$\mathcal{D}_k = \sum_{t \in T^k} \sum_{i \in V} P_{i,L}^t \times \eta_i + P_{i,D}^t \times \zeta_i^t \tag{11}$$

where η_i is the onsite renewable power cost in data center i (\$/kWh), ζ_i^t is the electricity price in data center i (\$/kWh).

- The cost of embedding in the backbone network

Virtual links between the VMs that have been assigned to different data centers should be embedded in the backbone network. We assume that it is proportional to their bandwidth requirements and the length of physical paths to which they are mapped. It is given by:

$$\mathcal{B}_k = \sum_{t \in T^k} \sum_{e' \in E^j} \sum_{e \in E} f_{e,e'}^t \times bw(e') \times \sigma_p \tag{12}$$

where σ_p is the cost incurred by the CP per unit of bandwidth allocated in the backbone network. Note that σ_p defines both the energy cost and any additional cost related to inter-data center bandwidth as defined in [29]. σ_p is the average cost per unit of bandwidth given the total measured cost.

- The migration cost

Let $t-1$ denote the time slot previous to time slot t. The migration cost is given by:

$$\mathcal{M}_k = \sum_{t \in T^k} \sum_{j \in (Q_{t-1} \cap Q_t)} \sum_{a \in V_j} \sum_{i_1, i_2 \in V} mig_{a,i_1,i_2}^{j,t} \times (m_{a,j} + w_{a,j,i_1,i_2}) \tag{13}$$

where $m_{a,j}$ is the cost of migrating VM a of VDC j, which corresponds to the disruption in service that might occur when migrating the VM, w_{a,j,i_1,i_2} is the energy cost for migrating VM a of VDC j from data center i_1 to data center i_2. In this paper, we use the following formula of w_{a,j,i_1,i_2} provided in [30]:

$$w_{a,j,i_1,i_2} = (0.512 \times \Delta_{mig} + 20.165) * \frac{\delta_{i_1}^t + \delta_{I_2}^t}{2}$$

where Δ_{mig} is the amount of data transferred between data centers during the migration of VMs. Note also that δ_i^t represents the power cost in data center i at time slot t, which is equal to ζ_i^t if the power is consumed from the grid and equal to η_i^t if the power is from on-site renewable source of energy. Finally, $mig_{a,i_1,i_2}^{j,t}$ is a binary variable that determines whether VM a of VDC j have been migrated to data center i_2 from data center i_1 at the beginning of time slot t. It is defined as follows:

$$mig_{a,i_1,i_2}^{j,t} = \begin{cases} 1 & \text{If } x_{i_2 a}^{j,t} = 1 \text{ and } x_{i_2 a}^{j,t-1} = 0 \\ & \text{and } x_{i_1 a}^{j,t} = 0 \text{ and } x_{i_1 a}^{j,t-1} = 1 \\ 0 & \text{Otherwise.} \end{cases}$$

Note that we assume that there is no cost for link migration as no data transfer is needed.

- The penalty cost

The penalty is paid by the CP to the SP whenever the specified green SLA is not met. At the end of every reporting period T^k, the CP reports the carbon emission related to each VDC request j that has been embedded for the whole time period T^k or during a part of it. Since the carbon emissions are due to the power consumption, we can derive the carbon emission of every data center i during a time slot t, denoted by $\mathcal{C}_i^t$, as follows:

$$\mathcal{C}_i^t = P_{i,D}^t \times C_i \tag{14}$$

where $P_{i,D}^t$ denotes the amount of purchased power from the grid by data center i during time slot t and C_i is the carbon footprint per unit of power used from the grid in data center i expressed in tons of carbon per kWh (tonsCO2/kWh).

We derive the carbon emissions, in the entire infrastructure, due to the servers (denoted by $\mathcal{C}_{i,Serv}^t$) and the network (denoted by $\mathcal{C}_{Net}^t$), as follows:

$$\mathcal{C}_{Serv}^t = \frac{1}{|V|} \sum_{i \in V} \frac{\mathcal{C}_i^t \times P_{i,Serv}^t}{P_{i,Net}^t + P_{i,Serv}^t} \tag{15}$$

$$\mathcal{C}_{Net}^t = \frac{1}{|V|+1} \times \left(\sum_{i \in V} \frac{\mathcal{C}_i^t \times P_{i,Serv}^t}{P_{i,Net}^t + P_{i,Serv}^t} + \mathcal{C}_{Bckb}^t \right) \tag{16}$$

where $\mathcal{C}^t_{Bckb}$ is the carbon emission due to embedding virtual links in the backbone network. In a similar way to the data centers, $\mathcal{C}^t_{Bckb}$ is computed for every time slot based on the power consumption and the carbon footprint per unit of power.

In this case, the average carbon emission rate of the CP per unit of VM during a reporting period T^k is given by:

$$\mathcal{C}^k_{CPU} = \frac{1}{t^k_e - t^k_b} \times \sum_{t\in[t^k_b, t^k_e]} \frac{\mathcal{C}^t_{Serv}}{\sum_{j\in Q_t}\sum_{v\in V^j} C^{cpu}(v)} \tag{17}$$

where Q_t is the set of VDC requests being embedded during time slot t and $C^{cpu}(v)$ is the capacity of VM v in terms of CPU units.

Similarly, the carbon emission rate per unit of bandwidth during a period T^k can be given as:

$$\mathcal{C}^k_{BW} = \frac{1}{t^k_e - t^k_b} \times \sum_{t\in[t^k_b, t^k_e]} \frac{\mathcal{C}^t_{Net}}{\sum_{j\in Q_t}\sum_{e\in E^j} bw(e)} \tag{18}$$

As such, the carbon emission related to a VDC request j during the period T^k, denoted by $\mathcal{C}^j_k$, can be given by:

$$\mathcal{C}^j_k = \mathcal{T}^j_k \times \left(\left(\sum_{v\in V^j} C^{cpu}(v) \times \mathcal{C}^k_{CPU}\right) + \left(\sum_{e\in E^j} bw(e) \times \mathcal{C}^k_{BW}\right)\right)$$

where $\mathcal{T}^j_k$ is the number of time slots of the period T^k during which VDC j was embedded.

Finally, a penalty is paid by the CP for an SP j at the end of the period T^k if the carbon emission for VDC j is above the limit specified in the SLA, i.e., $\mathcal{C}^j_k > c_j$, where c_j is the amount of carbon emission allowed by the SP for every reporting period. In this case, the total penalty cost for a period T^k is given by:

$$\mathcal{P}_k = \sum_{j\in(\cup_{t\in T^k} Q_t)} \left(\mathcal{R}^j \times \mathcal{T}^j_k\right) \times p, \quad \text{if } \mathcal{C}^j_k > c_j \tag{19}$$

where $p \in [0, 1]$ is the proportion of the SP's bill to be refunded by the CP in case of SLA violation. Note that p can be constant as it is common nowadays [28], or variable depending on the extent of the violation. For instance, in this paper, we use a simple penalty model as follows:

$$p = \max\left(\frac{\mathcal{C}^j_k}{c_j}, 1\right) \tag{20}$$

which makes the penalty proportional to the extent of the violation, with a maximum refund of 100% of the total amount of the bill. In this paper, we investigate both cases (i.e., constant penalty and variable penalty) and discuss them in the simulation results.

The problem described above can be seen as a combination of the bin-packing problem and the multi-commodity flow problem, which are known to be $\mathcal{NP}$-hard. Therefore, we propose a simple yet efficient and scalable solution.

V. Green SLA opTimizER (Greenslater)

Since the problem presented in the previous section is $\mathcal{NP}$-hard, we propose a greedy three-step approach. At the arrival a VDC request, the Central Controller first splits it into partitions such that the intra-partition bandwidth is maximized and the inter-partition bandwidth is minimized. It then uses an admission control algorithm that rejects VDCs with negative profit (i.e., the VDC cost is higher than the generated revenue). If the VDC is accepted, its partitions are embedded in different data centers. As the availability of renewables and electricity prices are variable over time, and the requests dynamically arrive and leave the system, we propose a reconfiguration algorithm, which migrates partitions from the data centers with no available renewables to those with available renewables. In the following, we present in details the proposed algorithms. Note that the partitioning aims at minimizing the backbone networks cost, while the reconfiguration minimizes the energy cost and limits the SLA violation by following the renewables, while taking into account the migration costs before migrating.

A. VDC Partitioning

Once received, the Central Controller divides the VDC request into partitions where the intra-partition bandwidth is maximized and the inter-partition bandwidth is minimized. Hence, each entire partition is then embedded in the same data center, which minimizes the inter-data center bandwidth. As the partitioning problem is $\mathcal{NP}$-hard [31], we use the Location Aware Louvain Algorithm (LALA), the partitioning algorithm used in [6]. LALA is a modified version of the Louvain Algorithm [32] that considers location constraints. The objective of the Louvain algorithm is to maximize the modularity, defined as an index between -1 and 1 that measures intra-partition density (i.e., the sum of the links' weights inside partitions) compared to inter-partition density (i.e., sum of the weights of links between partitions). In fact, graphs with high modularity have dense connections (i.e., high sum of weights) between the nodes within partitions, but sparse connections across partitions. Similar to the Louvain algorithm, the complexity of LALA is $O(n \log n)$ [32].

B. Admission Control

When a VDC request is received, the Central Controller checks if the request will generate profit, in which case it is accepted, otherwise it is rejected. In some cases, a request with tight carbon constraints might result in high SLA violation penalties, which reduces the CP's profit. To address this issue, we propose an admission control algorithm (Algorithm 1). The idea is to estimate the available renewable power in the next prediction window and estimate carbon emission of the requested VDC. In this paper, we consider solar panels to generate the renewable power and we use a prediction model presented in [13]. Moreover, we consider short term predictions (up to 4 hours).

Algorithm 1 Admission Control Algorithm

IN: *predictionWdW // the prediction window*
IN: *reconfigInterval // the reconfiguration interval*
IN: *vdc // the VDC to embed*
$wdw \leftarrow min(predictionWdw, reconfigInterval)$
$possible \leftarrow possibleToEmbed(vdc)$
if *possible* **then**
  $carbonRate \leftarrow getEstimationCarbonRate(wdw)$
  $carbonLimitRate \leftarrow vdc.carbonLimit/wdw$
  if carbonRate $\leq$ carbonLimitRate **then**
    Accept *vdc*
  else
    //Verify if profit can be made
    $estimatedCost \leftarrow estimatePowerCost(vdc)$
    if $revenue(vdc) \times (1-refundFactor) - estimatedCost > 0$ **then**
      Accept *vdc*
    else
      Reject *vdc*
    end if
  end if
else
  Reject *vdc*
end if

First, the central controller checks whether it is possible to embed the VDC given the available resources and constraints of the VMs in the VDC. If the request is embeddable, the central controller computes an estimation for carbon emission for the request given the current power consumption and the predicted availability of renewables for the next prediction window. To do so, we propose to use a simple estimation algorithm, which computes the estimation of carbon emission per unit of VM and per unit of bandwidth in the next prediction window, and by the same derives the estimation of carbon emission of the given VDC request. The estimated carbon of the VDC request is then compared to the limit provided in the SLA of the VDC request. In case of SLA violation, the Central Controller checks whether profit can still be made even if there is a penalty to pay. If the profit is positive, the VDC request is accepted, otherwise it is rejected. It is worth noting that as the prediction window is limited compared to the lifetime of some of the VDCs (up to weeks for long lived VDCs), the decision of accepting might be biased as the short term forecasts can show high availability of renewables.

C. Partitions Embedding

Once a request $G^j(V^j, E^j)$ is partitioned, the resulting partitions that are connected through virtual links can be seen as a multigraph $G^j_M(V^j_M, E^j_M)$ where V^j_M is the set of nodes (partitions) and E^j_M is the set of virtual links connecting them. This multigraph is then embedded into the infrastructure, partition by partition, using Algorithm 2. As reported in Algorithm 2, for each partition $v \in V^j_M$, we first build the list of data centers that satisfy the location constraints of its VMs. The Central Controller queries the Local Controller of each data center s from the list to get the embedding cost of v. The cost is returned by the remote call *getCost*(s, v).

Algorithm 2 Greedy VDC Partitions Embedding Across Data Centers

IN: $G(V \cup W, E)$, $G^j_M(V^j_M, E^j_M)$
for all $i \in V$ **do**
  $ToDC[i] \leftarrow \{\}$
end for
for all $v \in V^j_M$ **do**
  $S_v \leftarrow \{i \in V/i$ satisfies the location constraint$\}$
end for
for all $v \in V^j_M$ **do**
  $i \leftarrow s \in S_v$ with the smallest cost $getCost(s, v)$, and $LinksEmbedPossible(s, v) = true$
  if no data center is found **then**
    return *FAIL*
  end if
  $ToDC[i] \leftarrow ToDC[i] \cup \{v\}$
  for all $k \in N(v)$ **do**
    if $k \in ToDC[i]$ **then**
      $ToDC[i] \leftarrow ToDC[i] \cup \{e_{vk}\}$
    else
      if $\exists l \neq i \in V/k \in ToDC[l]$ **then**
        Embed e_{vk} in G using the shortest path
      end if
    end if
  end for
end for
return *ToDC*

The data center offering the lowest cost (provided by the procedure *getCost*(s, v)) and able to embed virtual links between v and all previously embedded partitions-denoted by $N(v)$-(verified by the function *LinksEmbedPossible*(s, v)) is then selected to host the partition. These virtual links are embedded in the backbone network using the shortest path algorithm.

This procedure is repeated until all partitions and virtual links that connect them are embedded into the distributed infrastructure. It is worth noting that the complexity of embedding the whole multigraph is $O(|V^j_M| \times |V|)$, where $|V^j_M|$ is the number of partitions and $|V|$ is the number of data centers.

D. Dynamic Partition Relocation

As the electricity price and the availability of renewables are variable over time, we propose a dynamic reconfiguration algorithm that optimizes VDC embedding over-time. Our aim is to migrate partitions that have already been embedded in data centers which may run out of renewables towards data centers with available renewable power. The second criterion to perform a migration is to move partitions to locations where the electricity price is lower.

Algorithm 3 Greedy Partition Migration Across Data Centers

1: **IN**: *predictionWdW // the prediction window*
2: **IN**: *reconfigInterval // the reconfiguration interval*
3: $wdw \leftarrow min(predictionWdW, reconfigInterval)$
4: **for all** $i \in V$ **do**
5: $Diff[i] \leftarrow EstimateRenewables(wdw, i) - FutureConsumption(wdw, i)$
6: **if** $Diff[i] < 0$ **then**
7: $part[i] \leftarrow$ list of partitions in i sorted by migration cost
8: **end if**
9: **end for**
10: **for all** $i \in V, Diff[i] < 0$ **do**
11: **while** $\ni k \in V, Diff[k] > 0$ **do**
12: $p \leftarrow part[i].first$
13: $D \leftarrow \{k \in V, Diff[k] > 0\}$
14: $done \leftarrow false$
15: **while** $!done$ && $D \neq \phi$ **do**
16: *//Take the data center with the minimum cost in the backbone network after migration*
17: $dest \leftarrow minBackboneCost(D)$
18: $Migrate(p, dest)$
19: **if** successful migration **then**
20: $done \leftarrow true$
21: Update $Diff[dest]$ and $Diff[i]$
22: **else**
23: $D \leftarrow D\backslash\{dest\}$
24: **end if**
25: **end while**
26: **end while**
27: **end for**

We, hence, propose a migration algorithm (Algorithm 3) executed every τ hours (i.e., reconfiguration interval) by the Central Controller.

Data centers are first classified into two categories: sources and destinations. A data center is considered as a source if it has not enough renewable power to support its workload and hence we will have to resort to power from the grid. In this case, in a source data center, the difference between the estimated available renewable power and the estimated power consumption is negative (cf. Line 5 of Algorithm 3). Conversely, if a data center has renewable power that exceeds its estimated power consumption, it is considered as destination data center since there is no need to reduce its workload and migrate VMs. In this case, it might be able to host more partitions if it has enough renewable power.

The idea is that partitions from source data centers should be migrated to destination data centers. To do so, the list of partitions in each source data center are sorted in increasing order of their migration cost (cf. Line 7 of Algorithm 3). For each partition, one destination data center that have a positive difference is chosen. The destination is chosen in a way that minimizes the inter-data center virtual link embedding cost after migration.

VI. Performance Evaluation

To evaluate the performance of Greenslater, we conducted several simulations using a realistic topology and real traces for electricity prices and renewable power availability. In the following, we first describe the simulation setting. Then, we present the results under two different penally cost models: a fixed penalty and a variable penalty that depends on the extent of the Green SLA violation.

A. Simulation Settings

For our simulations, we consider a physical infrastructure of 4 data centers located at four different states: New York, Illinois, California and Texas. The data centers are connected through the NSFNet topology as a backbone network, which includes 14 nodes. Each data center is connected to the backbone network through the closest node to its location. We assume all NSFNet links have a capacity of 100 Gbps. The traces of electricity prices and availability of renewable energy are provided by the US Energy Information Administration (EIA) [33]. The weather forecast is taken from the National Renewable Energy Laboratory [34] and the amount of power generated per square meter of solar panel from [35]. The carbon footprint per unit of power is provided by [36].

Similar to previous works [6], [15], VDCs are generated randomly according to a Poisson process with arrival rate λ and a lifetime following an exponential distribution with mean $1/\mu$. The number of VMs per VDC is uniformly distributed between 10 and 50 for regular VDCs, and between 5 and 10 for small VDCs. Note that the small VDCs are used only to run the exhaustive search algorithm in order to study the convergence to the optimal solution. A pair of VMs belonging to the same VDC are directly connected with a probability 0.5 with a bandwidth demand uniformly distributed between 10 and 50 Mbps and a delay uniformly distributed between 10 and 100 milliseconds. Each VM has a number of cores uniformly distributed between 1 and 4. Moreover, in each VDC, a fraction of VMs, denoted by $P_{loc} \in [0, 1]$, is assumed to have location constraints and thus cannot be migrated, i.e., it can only be embedded in a specific set of data centers. Each VDC comes with a carbon limit constraint specified in the Green SLA. This limit is assumed to be uniformly distributed between 5 and 20 kgCO_2 per day independently of the size of the VDCs to show the independence of our approach from the carbon constraints. When the Green SLA is not satisfied, the CP refunds a proportion p of the SP's bill for that specific period of time. In the first set of experiments, we consider p to be fixed to 50% of the bill. In the second set of experiments, we consider p to be proportional to the violation, i.e., the refund in percentage is equal to the proportion of violation divided by the limit of carbon of the VDC with a cap of 100%.

To assess the effectiveness of our proposal, we compare Greenslater to three other solutions: (i) Greenhead [6], (ii) Greenhead with No Partitioning (NP) (i.e., each VM is considered as a single partition), and (iii) the load balancing approach for VDC embedding [23]. Moreover, we developed an implementation of the brute force exhaustive search algorithm,

TABLE II
COMPARISON OF THE COMPUTATION TIME AND PERFORMANCE GAINS FOR THE OPTIMAL SOLUTION, GREENSLATER, GREENHEAD, GREENHEAD NP AND LOAD BALANCING

VDC size (VMs)	Approach	Computation time (ms)				Profit gain (%)
		Partitioning	Embedding	Total	Reconfiguration	
5-10	Optimal	-	-	11315 ± 31	49090 ± 94	132 ± 4.1
	Greenslater	0.00014	0.0042	0.0043 ± 0.0002	0.189 ± 0.009	126 ± 3.3
	Greenhead	0.00014	0.0016	0.0017 ± 0.0001	-	72 ± 2.6
	Greenhead NP	-	-	0.081 ± 0.011	-	39 ± 1.4
	Load Balancing	-	-	0.061 ± 0.008	-	0
10-50	Optimal	-	-	-	-	-
	Greenslater	53.2	0.123	53.323 ± 2.16	27.3 ± 1.37	129 ± 3.9
	Greenhead	55.95	0.052	56.002 ± 2.24	-	74 ± 2.7
	Greenhead NP	-	0.293	0.293 ± 0.014	-	45 ± 2.6
	Load Balancing	-	0.419	0.419 ± 0.021	-	0

that computes the optimal solution given by the ILP formulated in Section IV, to assess the convergence of our solution as well as the time complexity. The simulations are run using our own developed discrete event simulator, which extends the previous version developed in [6]. The interface between the central controller and the local controllers in each data center are implemented using remote procedure calls. Note that for each of the given results, the average values and confidence intervals of 80 consecutive runs are used.

For performance evaluation, we consider five metrics: (i) the profit of the CP, which is the difference between revenue and the sum of operational costs (i.e., power cost, backbone network cost) and the Green SLA violation cost, (ii) the acceptance ratio (defined as the ratio of embedded requests out of the total received requests by the CP), (iii) the carbon footprint generated by the whole infrastructure, (iv) the green power utilization and (v) the SLA violation penalty cost. We also measured the computation time for all the algorithms composing the solution, i.e., partitioning a VDC request, embedding the partitions, and the reconfiguration time, which is the computation time to find new embedding scheme for all partitions and virtual links.

B. Simulation Results Under Fixed Penalty Refund Factor

In this first set of simulations, we assume a fixed refund factor p. Specifically, p is set to 50%. That is, the CP refunds 50% of the SP's bill for the period of violation. Greenslater We first study the impact of the different input parameters: the arrival rate λ, the fraction of location constrained VMs P_{loc} and the reporting period T on the system performance, using different values of the reconfiguration interval τ.

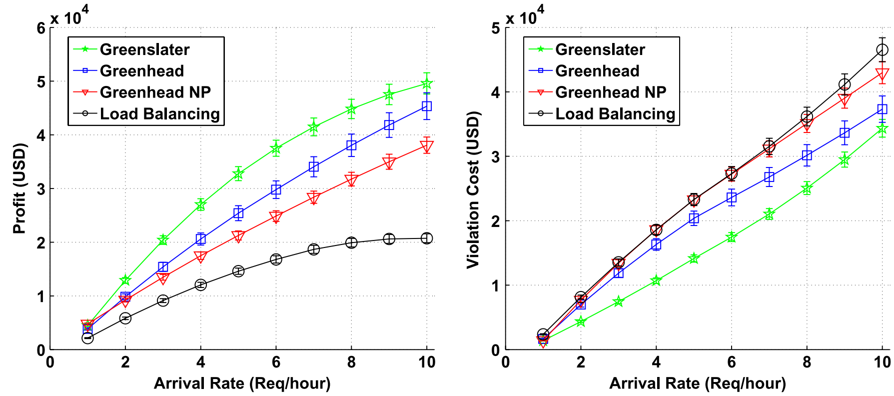
1) Computation Time and Convergence: First, we investigate the computation time of our proposed approach compared to the optimal solution given by the ILP formulation in Section IV, as well as the gain in terms of profit and SLA violation costs. To this end, we run simulations at a small arrival rate ($\lambda = 2$ Requests/hour) for small VDC requests (5–10 VMs). We implemented a brute force exhaustive search algorithm to find the optimal solution of the ILP formulated in Section IV. The brute force search algorithm iterates over all the possibilities for VM placement and virtual link allocation. Moreover, it uses the full knowledge of the available renewable power in the different data centers instead of the prediction algorithm used by Greenslater. We measured the computation time to partition, embed a VDC request and the time needed to reconfigure the infrastructure by migrating partitions. We also measured the profit gain compared to the Load Balancing approach. The results are summarized in Table II.

As reported in Table II, for small sized VDCs, we can notice that Greenslater achieves comparable gain in profit with the optimal solution, while incurring shorter embedding+partitioning time (i.e., 0.0043 ms in total) and reconfiguration time (i.e., 0.18 ms), compared to 11 seconds for embedding a request and 49 seconds to find the optimal configuration when using the optimal solution. Note that in this case, the other approaches achieve lower profit gain and higher computation time compared to Greenslater.

For large sized VDC requests, Greenslater again achieves the best profit gain with a short computation time. Specifically, the partitioning+embedding process of a VDC request takes less than 53 ms in average, which is similar to Greenhead as they use the same partitioning algorithm, while it takes less time for the other approaches as they do not partition the VDC requests. Note that the reconfiguration time in this case is less than 28 ms, which makes the algorithm usable in practice.

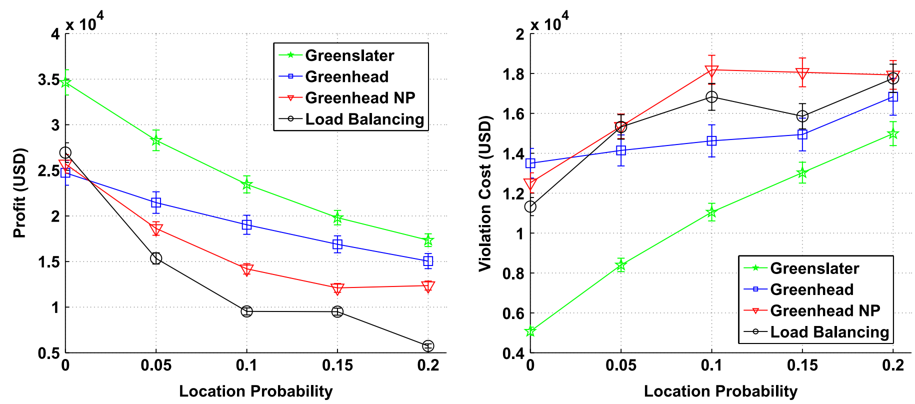
2) Impact of the Arrival Rate λ: Fig. 2 shows the impact of the arrival rate λ on both the achieved profit and SLA violation cost, when $P_{loc} = 0.05$ (i.e., low constrained locations), $T =$ 24 hours, and $\tau = 4$ hours. From this figure, we can notice that Greenslater outperforms other solutions, especially at high arrival rates (i.e., $\lambda \geq 3$). For small arrival rates (i.e., $\lambda \leq 2$), no considerable gain is observed as the number of requests being embedded is small. We can also observe that both the profit and SLA violation increase as the number of accepted requests increases. This is due to the fact that renewables are not enough to accommodate large numbers of VDCs, which leads to more power drawn from the electricity grid.

3) Impact of Location Probability Constraint P_{loc}: Let us now study how location-constrained VMs may impact the results. To do so, we have varied P_{loc} between 0 and 0.2, and fixed the values of $\lambda = 4$ requests/hour, $T = 24$ hours and $\tau = 4$ hours. We can see in Fig. 3 that Greenslater outperforms the other solutions for all the values of P_{loc}. However, as P_{loc} increases, the profit drops for all approaches since more VMs must be located in specific data centers. This limits the possibility of migrating the partitions, which may run using power from the grid. It is clear that the gain achieved by Greenslater is higher when less location constraints are considered (i.e., low P_{loc}).



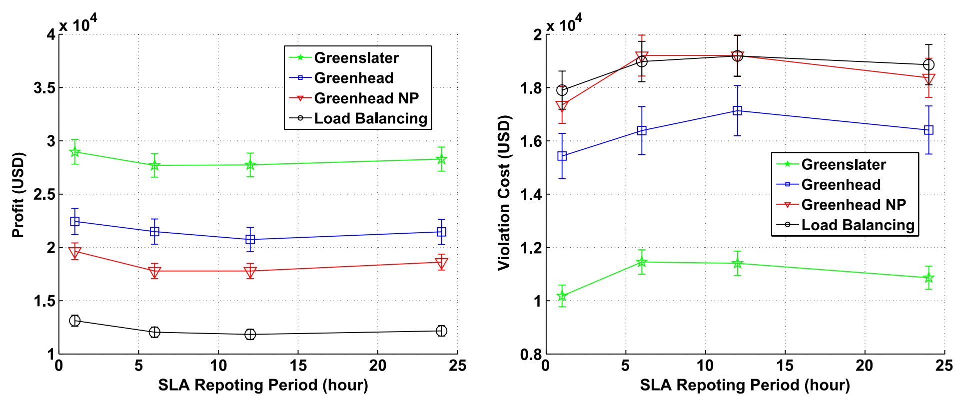
(a) (b)

Fig. 2. Impact of variable arrival rate λ ($P_{loc} = 0.05$, $T = 24$ hours, $\tau = 4$ hours). (a) Cumulative profit. (b) SLA violation cost.



(a) (b)

Fig. 3. Impact of variable location probability P_{loc} ($\lambda = 4$ requests/hour, $T = 24$ hours, $\tau = 4$ hours). (a) Cumulative profit. (b) SLA violation cost.



(a) (b)

Fig. 4. Impact of variable reporting period T ($\lambda = 4$ requests/hour, $P_{loc} = 0.05$, $\tau = 4$ hours). (a) Cumulative profit. (b) SLA violation cost.

4) Impact of Reporting Period T: Fig. 4 shows the impact of reporting period T on both the achieved profit and the SLA violation cost. In this scenario, we vary T in {1, 6, 12, 24} hours, for fixed values of $\lambda = 4$ requests/hour, $P_{loc} = 0.05$ and $\tau = 4$ hours. Note that, in this case, the carbon constraint limit specified in the Green SLA is assumed to be uniformly distributed between 5 and 20 kgCO_2 per day, and is scaled down to match the reporting period T. Again, Greenslater outperforms the baselines as it achieves higher profit and reduces the SLA violations costs. However, one can note that the profit is higher for long reporting periods (i.e., 24 hours) compared to short ones (i.e., 1, 6 and 12 hours). The rational behind this is that for long reporting periods T, the CP has more time and more flexibility. In fact, the carbon footprint is computed as an average value over the whole period T. For small values of T, the CP does not have enough leverage since, in some data

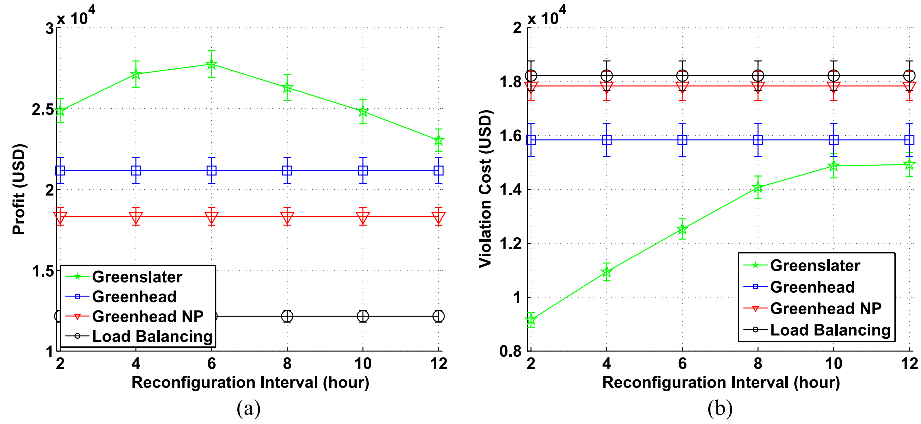


Fig. 5. Impact of variable reconfiguration interval τ ($\lambda = 4$ requests/hour, $P_{loc} = 0.05$, $T = 24$ hours). (a) Cumulative profit. (b) SLA violation cost.

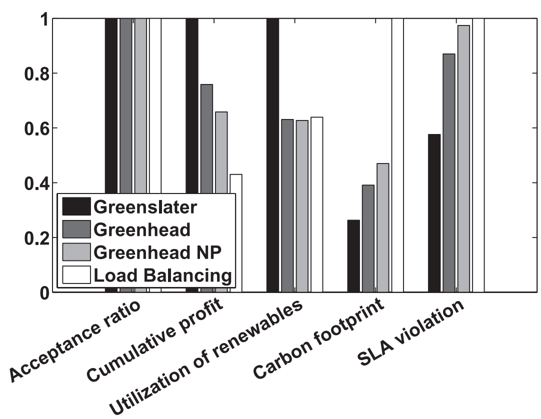


Fig. 6. Comparison of the cumulative values of the different metrics ($\lambda = 4$ requests/hour, $P_{loc} = 0.05$, $T = 24$ hours, $\tau = 4$ hours).

centers, VMs cannot be migrated even though renewables are available. This results in more frequent violation of the Green SLAs, which results in higher violations costs, as shown in Fig. 4(b), and thus lower profit (see Fig. 4(a)).

5) Impact of Reconfiguration Interval τ: We also study the impact of the reconfiguration interval τ on the profit and SLA violation cost. We varied τ between 1 and 12 hours and fixed other variables ($\lambda = 4$ requests/hour, $P_{loc} = 0.05$ and $T = 24$ hours). The results are shown in Fig. 5. From this figure, we can see that the profit for Greenslater is a concave function of τ, where the maximum profit is obtained for $\tau_{opt} = 6$ hours in our case. In addition, the SLA violation cost increases with τ, but remains low compared to the other solutions. In particular, for high values of τ, Greenslater gains decrease, since in this range of τ, the system configuration is not reoptimized to follow the renewables. Note that the variation of τ does not affect the performance of the other schemes, since they do not perform any migrations.

6) Summary of the Results: To highlight the benefits of Greenslater over existing solutions, we plotted all the studied performance metrics (acceptance ratio, cumulative profit, utilization of renewable energy, carbon footprint and SLA violation cost) in Fig. 6. It is clear that Greenslater always achieves higher profit, ensures higher utilization of renewables and lower carbon footprint with minimum SLA violation. For instance, the gain in terms of profit provided by Greenslater is respectively around 33%, 53% and 129% compared to Greenhead, Greenhead NP and the Load Balancing approach.

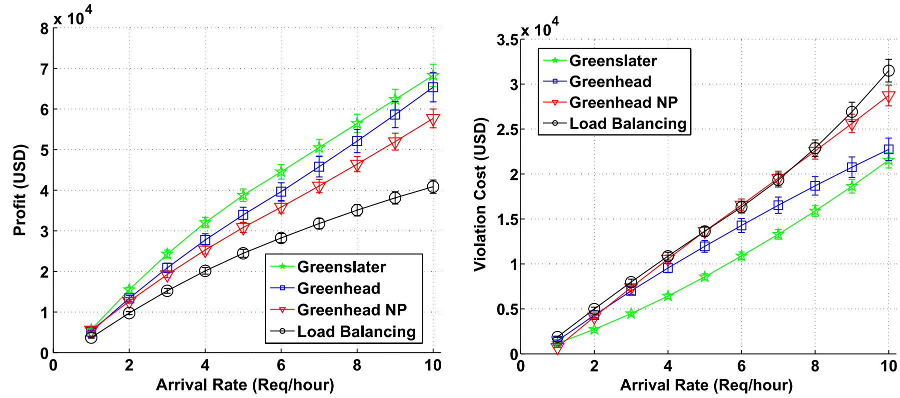
C. Simulation Results for Variable Penalty Cost

Now, we present the simulation results when the penalty cost is proportional to the Green SLA violation. More specifically, we assume the violation penalty is a percentage of the SP's bill to refund. This percentage is proportional to the violation of the carbon limit constraint defined in the Green SLA. Hence, we consider the penalty formula defined in equation (20).

We studied the impact of arrival rate λ, the reporting period T and the reconfiguration interval τ. Fig. 7 shows the profit and SLA violation costs under variable arrival rate λ, when $P_{loc} = 0.05$ (i.e., low constrained locations), $T = 24$ hours, and $\tau = 4$ hours. Similar to the case of fixed penalty cost, Greenslater achieves higher profit while reducing the SLA violation cost under different arrival rates. The achieved gain is negligeable under low arrival rates $\lambda \leq 2$, but considerable under higher arrival rates. For instance, the gain in profit culminates at 20%, 30% and 33% compared to Greeanhead, Greeanhead NP and the Load Balancing approaches, respectively.

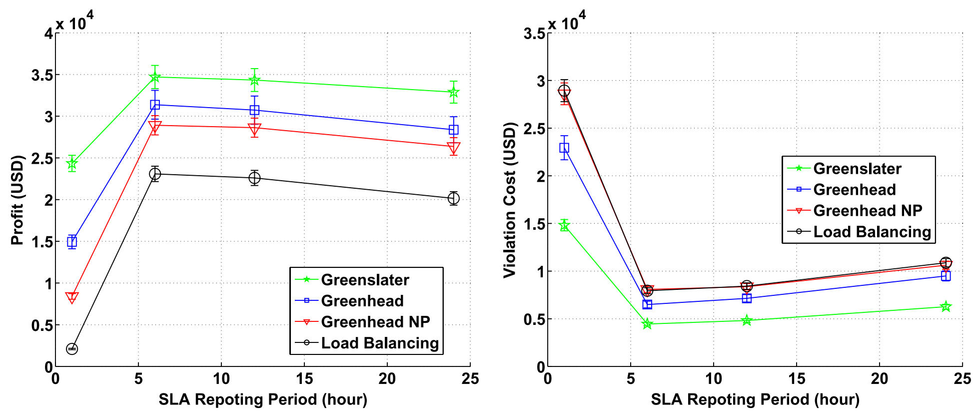
In another set of simulations, we varied the reporting period $T \in \{1, 6, 12, 24\}$ hours, while we fixed the values of $\lambda = 4$ requests/hour, $P_{loc} = 0.05$ and $\tau = 4$ hours. Fig. 8 shows the achieved profit and the SLA violation cost. From this figure, we can see that Greenslater always achieves higher profit and reduced SLA violation costs. In particular, the highest profit is achieved when the reporting period is equal to 6 and 12 hours, while reducing the reporting period (i.e., 1 hour) gives the worst results in profit. Note that this is different from the case of fixed penalty cost. This is explained by the fact that, on the one hand, the violations observed in the reporting periods 6–12 hours are very low in magnitude (small violations only) compared to the violations observed in small reporting periods (i.e., 1 hour). On the other hand, the magnitude of the violation is not taken into account in the case of fixed penalty cost.

We also studied the impact of the reconfiguration interval τ (varied between 1 and 12) on the profit and SLA violation cost, when $\lambda = 4$ requests/hour, $P_{loc} = 0.05$ and $T = 24$ hours. The



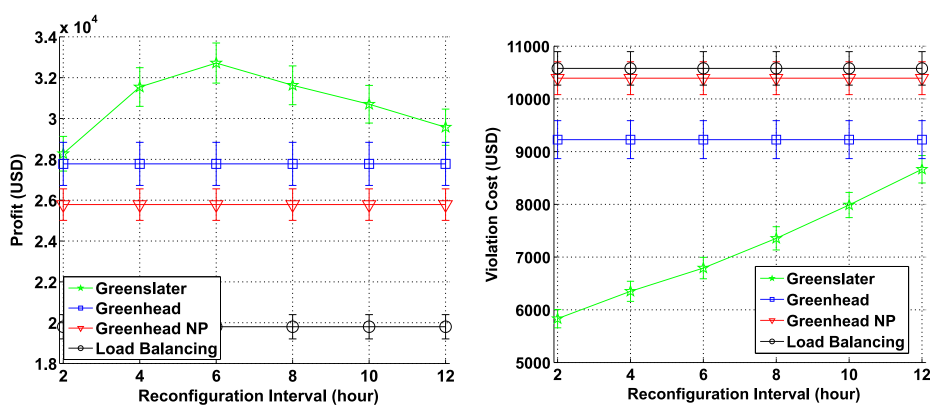
(a) (b)

Fig. 7. Impact of variable arrival rate λ under variable penalty cost ($P_{loc} = 0.05$, $T = 24$ hours, $\tau = 4$ hours). (a) Cumulative profit. (b) SLA violation cost.



(a) (b)

Fig. 8. Impact of variable reporting period T under variable penalty cost ($\lambda = 4$ requests/hour, $P_{loc} = 0.05$, $\tau = 4$ hours). (a) Cumulative profit. (b) SLA violation cost.



(a) (b)

Fig. 9. Impact of variable reconfiguration interval τ under variable penalty cost ($\lambda = 4$ requests/hour, $P_{loc} = 0.05$, $T = 24$ hours). (a) Cumulative profit. (b) SLA violation cost.

results are shown in Fig. 9. Similar to the case of fixed penalty cost, the maximum profit is obtained for $\tau_{opt} = 6$ hours, and the same behavior is observed for the SLA violation cost, which increases with τ.

Finally, Fig. 10 illustrates a summary of additional metrics (acceptance ratio, cumulative profit, utilization of renewable energy, carbon footprint and SLA violation cost), when $\lambda = 4$ requests/hour, $P_{loc} = 0.05$, $T = 24$ hours, $\tau = 4$ hours. From this figure, we can note that Greenslater achieves higher profit and ensures higher utilization of renewables and lower carbon footprint with minimum SLA violation compared to the baseline approaches. For instance, the gain in profit for Greenslater are 19%, 25% and 67% compared to Greenhead, Greenhead NP and the Load Balancing approach.

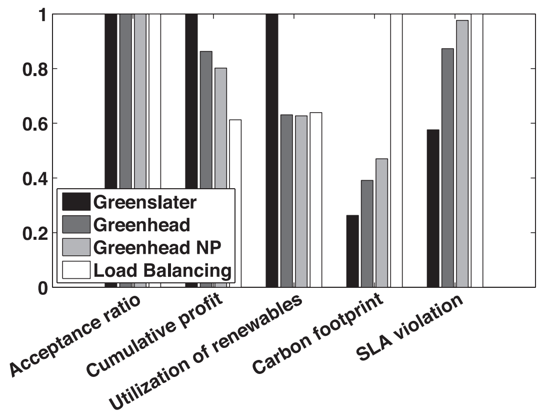


Fig. 10. Comparison of the cumulative values of the different metrics under variable penalty cost ($\lambda = 4$ requests/hour, $P_{loc} = 0.05$, $T = 24$ hours, $\tau = 4$ hours).

VII. Conclusion

As the environmental impact of cloud infrastructures and services has become increasingly significant, governments and environmental organizations are in a ramping effort to urge SPs to require guarantees from their CPs that the carbon emission generated by the leased resources is limited. Hence, in this paper, we addressed the problem of including green constraints in the SLAs in order to cap the carbon emission of the resources allocated to each SP. We proposed Greenslater, a holistic framework that allows CPs to provision VDCs across a geographically distributed infrastructure with the goal of minimizing the operational costs and green SLA violation penalties. More specifically, Greenslater incorporates admission control to wisely select which VDC requests to accept, and a dynamic reconfiguration algorithm to allow the CP to relocate parts of the VDCs in data centers with available renewable energy. The simulation results showed that, compared to existing solutions, Greenslater achieves high profit by minimizing operational costs and SLA violation penalties, while maximizing the utilization of the available renewable power, under both fixed and variable SLA violation penalty models. More specifically, Greenslater achieves profit gains of up to 33%, 53% and 129% compared to Greenhead, Greenhead NP and the Load Balancing approach, respectively.

References

[1] "Toolkit on environmental sustainability for the ICT sector (ESS)," Int. Telecommun. Union (ITU), Geneva, Switzerland, 2012. [Online]. Available: http://www.itu.int/ITU-T/climatechange/ess/index.html

[2] "Carbon footprint stomps on firm value," KPMG Int. Rep., Amsterdam, The Netherlands, Dec. 2012. [Online]. Available: http://goo.gl/migJnq

[3] "Carbon Disclosure Project website." [Online]. Available: www.cdp.com

[4] "Carbon risks and opportunities in the s&p 500," Investor Responsibility Res. Center Instit. (IRRCi)/Trucost, London, U.K., Jun. 2009.

[5] Amazon Virtual Private Cloud. [Online]. Available: http://aws.amazon.com/vpc/

[6] A. Amokrane, M. F. Zhani, R. Langar, R. Boutaba, and G. Pujolle, "Greenhead: Virtual data center embedding across distributed infrastructures," *IEEE Trans. Cloud Comput.*, vol. 1, no. 1, pp. 36–49, Jan./Jun. 2013.

[7] "Open data center alliance usage: Carbon footprint values," Open Data Center Alliance, Beaverton, OR, USA, 2011. [Online]. Available: http://goo.gl/QcEfhG

[8] Report of the second meeting of the Cloud Selected Industry Group, Service Level Agreements Expert Subgroup, Apr. 2013. [Online]. Available: http://bit.ly/KUhR8v

[9] G. Laszewski and L. Wang, "Greenit service level agreements," in *Grids and Service-Oriented Architectures for Service Level Agreements*, P. Wieder, R. Yahyapour, and W. Ziegler, Eds. New York, NY, USA: Springer-Verlag, 2010, pp. 77–88.

[10] C. Bunse, S. Klingert, and T. Schulze, "GreenSLAs: Supporting energy-efficiency through contracts," *Energy Efficient Data Centers*, vol. 7396, pp. 54–68, 2012.

[11] A. Galati *et al.*, "Designing an SLA protocol with renegotiation to maximize revenues for the CMAC platform," in *Proc. Web Inf. Syst. Eng. Workshops*, 2013, pp. 105–117.

[12] C. Atkinson, T. Schulze, and S. Klingert, "Facilitating greener it through green specifications," *IEEE Softw.*, vol. 31, no. 3, pp. 56–63, May 2014.

[13] M. Haque, K. Le, I. Goiri, R. Bianchini, and T. Nguyen, "Providing green SLAs in high performance computing clouds," in *Proc. IGCC*, 2013, pp. 2–11.

[14] M. S. Hasan, Y. Kouki, T. Ledoux, and J.-L. Pazat, "Cloud energy broker: Towards SLA-driven green energy planning for IaaS providers," in *Proc. IEEE Int. Conf. HPCC*, Aug. 2014, pp. 1–8.

[15] 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 IM*, May 2013, pp. 18–25.

[16] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for Internet-scale systems," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 123–134, Aug. 2009.

[17] Q. Zhang, Q. Zhu, M. F. Zhani, and R. Boutaba, "Dynamic service placement in geographically distributed clouds," in *Proc. ICDCS*, 2012, pp. 526–535.

[18] D. Hatzopoulos, I. Koutsopoulos, G. Koutitas, and W. Van Heddeghem, "Dynamic virtual machine allocation in cloud server facility systems with renewable energy sources," in *Proc. IEEE ICC*, 2013, pp. 4217–4221.

[19] P. X. Gao, A. R. Curtis, B. Wong, and S. Keshav, "It's not easy being green," in *Proc. ACM SIGCOMM*, 2012, pp. 211–222.

[20] Y. Guo, Y. Gong, Y. Fang, P. Khargonekar, and X. Geng, "Energy and network aware workload management for sustainable data centers with thermal storage," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 8, pp. 2030–2042, Aug. 2014.

[21] J. He, X. Deng, D. Wu, Y. Wen, and D. Wu, "Socially-responsible load scheduling algorithms for sustainable data centers over smart grid," in *Proc. IEEE Int. Conf. SmartGridComm*, Nov. 2012, pp. 406–411.

[22] Z. Abbasi, M. Pore, and S. Gupta, "Impact of workload and renewable prediction on the value of geographical workload management," in *Proc. 2nd Int. Workshop E2DC*, vol. 8343, pp. 1–15, 2014.

[23] Y. Xin *et al.*, "Embedding virtual topologies in networked clouds," in *Proc. Int. CFI Technol.*, 2011, pp. 26–29.

[24] L. Wang *et al.*, "Energy-aware parallel task scheduling in a cluster," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1661–1670, Sep. 2013.

[25] S. Klingert, T. Schulze, and C. Bunse, "Green SLAs for the energy-efficient management of data centres," in *Proc. Int. Conf. Energy-Efficient Comput. Netw.*, 2011, pp. 21–30.

[26] D. Rincn *et al.*, "A novel collaboration paradigm for reducing energy consumption and carbon dioxide emissions in data centres," *Comput. J.*, vol. 56, no. 12, pp. 1518–1536, 2013.

[27] L. Wang, G. von Laszewski, J. Dayal, and F. Wang, "Towards energy aware scheduling for precedence constrained parallel tasks in a cluster with DVFS," in *Proc. 10th IEEE/ACM Int. Conf. CCGrid*, May 2010, pp. 368–377.

[28] S. A. Baset, "Cloud SLAs: Present and future," *SIGOPS Oper. Syst. Rev.*, vol. 46, no. 2, pp. 57–66, Jul. 2012.

[29] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: Research problems in data center networks," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 68–73, Dec. 2008.

[30] H. Liu, C.-Z. Xu, H. Jin, J. Gong, and X. Liao, "Performance and energy modeling for live migration of virtual machines," in *Proc. 20th Int. Symp. HPDC*, 2011, pp. 171–182.

[31] S. E. Schaeffer, "Graph clustering," *Comput. Sci. Rev.*, vol. 1, no. 1, pp. 27–64, 2007.

[32] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Statist. Mech.: Theory Exp.*, vol. 10, no. 10, p. 8, Oct. 2008.

[33] U.S. Energy Information Administration. [Online]. Available: http://www.eia.gov

[34] National Renewable Energy Laboratory, Feb. 2014. [Online]. Available: http://www.nrel.gov/gis/solar.html

[35] The Renewable Resource Data Center (RReDC), 2012. [Online]. Available: http://www.nrel.gov/rredc/
[36] Carbon Footprint Calculator, 2012. [Online]. Available: http://www.carbonfootprint.com

Ahmed Amokrane received the M.Sc. degree in computer science from École Normale Supérieure de Cachan, Cachan, France, in 2011 and the Ph.D. degree in computer science from the Pierre and Marie Curie University, Paris, France, in 2014. He was a Visiting Ph.D. Student for nine months with the University of Waterloo, ON, Canada, and a Research Intern for six months with IBM T.J Watson Research Center, Yorktown Heights, NY, USA. He is currently working as a Consultant and R&D Engineer with CoESSI, Montesson, France. His research interests include energy efficiency, green networking, and security in wireless networks, cloud computing, and software-defined networks.

Rami Langar received the M.Sc. degree in network and computer science from the Pierre and Marie Curie University, Paris, France, in 2002 and the Ph.D. degree in network and computer science from Telecom ParisTech, Paris, France, in 2006. He is currently an Associate Professor with Laboratoire d'Informatique de Paris 6 (LIP6), Pierre and Marie Curie University. In 2007 and 2008, he was with the School of Computer Science, University of Waterloo, Waterloo, ON, Canada, as a Postdoctoral Research Fellow. His research interests include mobility and resource management in cloud radio access networks, wireless mesh, vehicular ad hoc and femtocell networks, green networking, green cloud, and quality-of-service support.

Mohamed Faten Zhani received the Ph.D. degree in computer science from the University of Quebec, Montréal, QC, Canada, in 2011. He is an Assistant Professor with the Department of Software and IT Engineering, École de Technologie Supérieure (ÉTS), University of Quebec. Before that, he was a Postdoctoral Research Fellow with the David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada. His research interests include cloud computing, virtualization, big-data analytics, software-defined networks, and resource management in large-scale distributed systems.

Raouf Boutaba (M'93–SM'01–F'12) received the M.Sc. and Ph.D. degrees in computer science from the Pierre and Marie Curie University, Paris, France, in 1990 and 1994, respectively. He is currently a Professor of computer science with the University of Waterloo, Waterloo, ON, Canada. His research interests include resource and service management in networks and distributed systems. Dr. Boutaba is a Fellow of the IEEE, the Engineering Institute of Canada, and the Canadian Academy of Engineering. He served as a Distinguished Speaker for the IEEE Computer and Communications Societies. He is the founding Editor-in-Chief of the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT (2007–2010), and he is on the editorial boards of other journals. He was the recipient of several Best Paper Awards and other recognitions, such as the Premier's Research Excellence Award, the IEEE Hal Sobol Award in 2007, the Fred W. Ellersick Prize in 2008, the Joe LociCero Award and the Dan Stokesbury Award in 2009, the Salah Aidarous Award in 2012, and the McNaughton Gold Medal in 2014.

Guy Pujolle received the Ph.D. and "These d'Etat" degrees in computer science from the University of Paris IX and Paris XI, in 1975 and 1978, respectively. He is currently a Professor at Pierre et Marie Curie University-Paris 6, a Distinguished Invited Professor at POSTECH, Korea, a member of the Institut Universitaire de France, and a member of The Royal Physiographical Academy of Lund, Sweden. He spent the period 1994–2000 as Professor and Head of the computer science department of Versailles University. He was also Professor and Head of the MASI Laboratory at Pierre et Marie Curie University (1981–1993), Professor at ENST (1979–1981), and a member of the scientific staff of INRIA (1974–1979). He is the French representative at the Technical Committee on Networking at IFIP. He is an Editor for *ACM International Journal of Network Management, Telecommunication Systems*, and Editor in Chief of *Annals of Telecommunications*. He is a pioneer in high-speed networking having led the development of the first Gbit/s network to be tested in 1980. He has participated in several important patents like DPI or virtual networks. Guy Pujolle is Co-founder of QoSMOS (www.qosmos.fr), Ucopia Communications (www.ucopia.com), EtherTrust (www.ethertrust.com), Virtuor (www.VirtuOR.fr), and Green Communications (www.green-communications.fr).