IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received February 16, 2021, accepted March 1, 2021, date of publication March 4, 2021, date of current version March 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3063708

Parallel, Proactive and Power Efficient Virtual Network Embedding in a Green and Distributed SD-ODCN Architecture

IKHLASSE HAMZAOUI[1,2,3], GUILLAUME BOURGEOIS[4], BENJAMIN DUTHIL[3,4], VINCENT COURBOULAY[4], AND HICHAM MEDROMI[1,2]

[1]Research Foundation for Development and Innovation in Science and Engineering (FRDISI), Casablanca 16469, Morocco
[2]Engineering Research Laboratory (LRI), System Architecture Team (EAS), National Higher School of Electricity and Mechanic (ENSEM), Hassan II University, Casablanca 20310, Morocco
[3]EIGSI, 17041 La Rochelle, France
[4]IT, Image, Interaction Laboratory (L3I), University of La Rochelle, 17000 La Rochelle, France

Corresponding author: Ikhlasse Hamzaoui (ikhlasse.h12@gmail.com)

This work was supported as part of Green IT projects launched by L3I Laboratory and EIGSI school in La Rochelle (France), to achieve digital sobriety transition. The Moroccan Ministry of Higher Education (CNRST) and FRDISI founded this work.

ABSTRACT With a view to overcome internet ossification problem, various Virtual Network Embedding (VNE) approaches has been proposed in last few years. Nevertheless, most of prior approaches neglect some major operational requirements implied by the inherent virtualization platforms. In the case of SD-ODCN architecture, a crucial operational requirement instance is the ability to route application-specific flows or wavelengths dynamically and efficiently across multi-tenant network providers. With the perpetual bursting of clouds and IP traffics, efficient dynamic VNE not only consists on maximizing ISPs and cloud Service Provider (SP) revenues, but also involves a strong need to reduce carbon emissions. In this article, we introduce a new Parallel, Proactive and Power Efficient VNE in a Green and Distributed SD-ODCN architecture. We first formulate a Mixed Integer Linear Programming (MILP) model purposing to maximize total intra Data Center (DC)' servers and inter networking resources power efficiency as a function of users' request rates. Afterward, we proposed a new green location-aware, Parallel Global resource Topology Ranking (PGTR) method, prioritizing first the greenest server and network nodes. Depending on resulted ranking process classes, a Parallel and Proactive VNE (PPVNE) is therefore proposed to effectively maximize total DC's and networking resources power efficiency. After implementing the whole proposed algorithms namely (PGTR-PPVNE) under NSFNET network topology related data, extensive simulations results proved the improvement of the proposed (PGTR-PPVNE) approach over four other benchmark methods. More precisely, the proposed (PGTR-PPVNE) achieved 6.87% decrease, 10.77% increase, 58.54% decrease and a 1.15% increase over the proactive ACO-VNE benchmark approach, respectively in terms of total power consumption, total power efficiency, requests response times and acceptance ratio.

INDEX TERMS Parallel and proactive virtual network embedding (PPVNE), optical software defined network, power efficiency, parallel global resources topology ranking (PGTR).

I. INTRODUCTION

With the sharp increasing of dynamic high-bandwidth multimodal applications (e.g., cloud services, streaming videos, and global IP traffic), the ossified architecture of traditional network becomes inflexible for services provisioning in cloud network. Fortunately, the emergence of Software Defined Network (SDN) with Network Function Virtualization (NFV) [1] opens new perspectives to meet this challenge. Indeed, the reliability of separate centralized control plane from the entire data plane enhances the ability to dynamically route application-specific wavelengths across multi-tenant cloud services and network infrastructures. On the other hand, this flexibility has further increased with the appearance of NFV. This later concept brings the implementation of network functions in software rather than physical devices [29], through a series of Virtual Network Functions (VNFs), making hence resources

The associate editor coordinating the review of this manuscript and approving it for publication was Fan-Hsun Tseng.

provisioning independent on specific hardware equipment. Moreover, even most existing works focus on single domain network, the large core backbone Internet has evolved from limited single domain network's resources to different network domains. Those multi-domain networks are managed by several operators and network infrastructure providers [2]. Therefore, multi-domain networks' efficient resources virtualization under above emergent technologies, becomes the promising solution that surpasses internet ossification and reduces global Operational Expenditures (OPEX) and Capital Expenditures (CAPEX) in the future multi-cloud industry.

In the other hand, multi-tenant network operators and cloud service providers are nowadays rivalling each other to be in line with the Paris climate agreement [3]. Thus, more attentions start to be paid for reducing their energies consumptions and greening their entire infrastructure with many auto-generated or purchased Renewable Energies (REs). Google was the first cloud provider, which has powered their entire DCs operations by 100% REs by 2017 [4]. Amazon AWS cloud came at a second place when several new wind and solar farms were inaugurated in Europe and the United States, aiming to produce 100% green energy between 2020 and 2030 [5]. Similarly, IBM has set a target of purchasing 55 percent of IBM's global electricity consumption from renewable sources by 2025 [6]. However, the production variability of most renewable sources, pushes DCs to depend principally on the regular electrical grid as a backup or main supplier. The arising challenge consists on effectively managing REs availabilities with a view to reduce intra and inter cloud resources' carbon emissions. Recent studies have only investigated intra-cloud resources' carbon emissions reduction [28] [30], [31], without focusing on multi-domain inter-cloud networking resources' carbon emissions. This further incites the need to adopt carbon emission-based intra and inter cloud resources' ranking approach, aiming toward green substrate devices selection.

In addition to REs' adoption, there exists according to Microsoft cloud, three other key investments to reduce the environmental impact of each cloud services [7]. These three investments focus on physical equipment efficiency, operational efficiency, then cooling infrastructure efficiency. Despite recent efforts made in those directions, this ongoing challenge remains open as network traffic rates continue to surge, and global cloud resource topology remains inefficiently exploited. In fact, servers are the most energy intensive among the whole IT DCs elements, since they consume large energy consumption when underutilized, and still consume more than half their maximum energy consumption even at idle state [8], [9]. This resources' underutilization phenomenon was also widely noticed in cloud network, where network links operated only between 5 and 25% [10]. This last issue is jointly related to resources over-provisioning, driven by the interest of ensuring Quality of Service (QoS) at the expense of cloud resources wastage. The ongoing challenge becomes how to deal with the entire problem, by combining dynamic virtual Data Centers (DCs) embedding with multi-domain networks embedding. Accordingly, the main target is to achieve global resources power efficiency: as a main representative metric englobing traffic rate as a function of global virtualized inter and intra cloud resources' power consumption.

In this article, we propose a green and distributed SD-ODCN architecture containing three layers: 1) a Service Provider (SP) layer, 2) a global Virtual Resources Orchestration (VRO) layer, finally 3) a multi-domain intra and inter cloud Infrastructure Providers (IPs) layer. In this later layer, we address a Parallel and Proactive Virtual Network Embedding (PPVNE) sorting the total power efficiency of the whole network components and DCs' servers. Inspired by [11] and our previous complete survey study [12], we propose a mathematical MILP model maximizing total power efficiency in both DCs and inter cloud multi-domain networks components in the SD-ODCN architecture. Contributions in this article are synthesized as follows:

– After instantiating Virtual Network Requests (VNRs) in the SP layer, each user requiring multi-services VNRs at a fixed time stamp may benefit from a parallel and distributed processing in a timely manner. This is achieved through the consideration of a splitting and forwarding scheme provided by a Central SDN controller Node (CN).

– Still in the VRO layer, we propose a Parallel Global resources Topology Ranking (PGTR) based on a green-aware execution scenario, in which the whole topology Carbon Emission Rate (CER) values can be obtained under three ranking classes.

– Leveraging results of PGTRs, a new Parallel and Proactive VNE namely (PPVNE) was proposed in IPs layer. Results were compared to traditional reactive scenario, in which VNRs are over-provisioned based on SP' instantiated resources.

The rest of this article is organized as follows. Section II provides an overview of related works. Section III describes the proposed green and distributed SD-ODCN architecture. In Section IV, the MILP are proposed. We introduce in Section V the whole algorithms' designs. Then, collected data, simulation setups and results' discussions are presented in Section VI. Finally, Section VII concludes this article.

II. RELATED WORK

As reported from an energy profiling system simulating power consumption at network multiprotocol label switching (MPLS) level [13], the packet-processing rate at core network is the main network power driver. With the emergence of network virtualization, ISPs [14] have largely benefited from the flexibility of mapping virtual nodes on substrate active nodes generating low electricity price. However, network virtualization has also served to optimize the overall required energies to operate networks. Previous works [15], [16] have investigated the use of network coding to improve the energy efficiency of core networks. They considered either no-bypass approaches or bypass ones, where flows must visit the IP layer in each intermediate node between source and end destination.

Starting with intra DC network, the study [17] presents EnergyMap: a novel energy-efficient embedding method mapping heterogeneous MapReduce-based virtual networks into a heterogeneous DC network. As MapReduce becomes a cloud computing paradigm, this later work investigated the case where Virtual DC (VDC) embedding requires splitting the computation-based virtual nodes (mappers/reducers) and embedding them into multiple servers. The aim behind this later study is to parallelize the whole computation tasks. Another work [33] joined Sliceable-Transponders reusability together with DC consolidation, for an Energy-Aware Virtual Optical Network Embedding (EA-VONE). Still as an attempt to join network virtualization to server's consolidation, the work [28] tried to keep DC's network maintaining a Minimum Spanning Tree (MST) for minimum connectivity between servers. In doing so, VMs should cross fewer short hops, thereby reducing the power consumption of network devices. Under this scheme, some physical servers may be left underutilized, thus an "On arrival migration" (MoA) algorithm was proposed to perform every required VNRs' remapping.

However, to optimize energy consumption over the VN entire life cycle and in a realistic DC topology, this work [18] designed a novel VNE model developed through a Topology and energy aware Migration heuristic applied to both computing and networking resources. As repetitive migrations create additional bottlenecks at substrate links level, the author in [32] proposes a Markov Decision Processes evaluating optimal migration policies considering both operations' energy consumption and reconfiguration costs. Another paper [19] addresses a comprehensive node ranking method called CNRM, in order to measure each substrate node importance based on resources usage ratio. The principle behind this importance ranking was the lower the node resource utilization, the less the additional energy consumption then the larger the CNRM ranking value. Doing so, the proposed load balancing joined to VNE has reduced energy consumption and improved the load balancing in substrate network, while slightly deteriorated long-term average revenue, acceptance ratio and revenue to cost ratio. However, most above studies do not consider the whole elastic optical network components while investigating VNE. Considering the granular power consumption of various backbone network devices as well as DCs' servers, the work [21] investigated a REOVNE heuristic solving the VNE problem under several resources' capacities constraints, network flow conservation constraints as well as under a VNRs grouping approach.

Again, all above solutions focused on the economical aspect of energy consumption by maximizing revenues and reducing energy costs, while neglecting the CO2 emission factor resulting from electricity production. This work [20] comes up with a potential trade-off between centralized and distributed approaches. It is about a hybrid approach reducing communication overhead while facilitating more optimal embedding. Management nodes in this proposed architecture attempt to find nodes handling the closest capacities to the requested ones and having the smallest Co2 emission factor. Then, virtual links are assigned to the minimum number of physical links in shortest paths. Another paper [21] developed a Green Virtual Network Embedding (GVNE) framework for minimizing the use of non-renewable energy, through intelligent provisioning of bandwidth and cloud DC resources. The proposed MILP model determines how to effectively use RE during the mapping of VNRs and whether to embed virtual nodes locally or to move them to distant DCs with abundant solar energy resources. Considering a Dummy Node and an auxiliary matrix of nodes and links, another heuristic [22] performs a short path selection to minimize the non-RE consumption at network level. The effective non-RE consumption value was represented as the combined value of non-RE consumption and network migration penalty at each time interval.

As reported previously, most research papers focused on single-domain network and considered the whole network resources utilization are always powered with non-REs [21], [22]. Although some recent studies start investigating multi-domain VNE in federated SDNs [34], experimental results were not treated considering real optical network topology and granular power consumption of various optical network devices. Unlike previous research, we address in this article an intra and inter cloud resources Parallel and Proactive VNE (PGTR-PPVNE), considering various energy types' availabilities in intra and inter cloud levels. We compared all our simulated results with the whole benchmark algorithms mentioned in [11]. The following section describes in detail our proposed solution architecture aiming to maximize REs utilizations, then network and DCs' servers total power efficiency.

III. DESCRIPTION OF THE PROPOSED SYSTEM

We present in this article a new SD-ODCN architecture, as depicted in Figure 1, which is composed of three business layers: 1) Service provider layer, 2) Virtual Network control and Orchestration (VNO) layer, and 3) Physical and networking infrastructure layer. Figure 1 below presents this architecture, followed by a description of functions carried by each layer.

A. THE SERVICE PROVIDER LAYER

In this first layer, the SP receives a group of users' VNRs. Indeed, each SP is engaged through a business agreement with their subscribers, where a set of required QoS are mentioned. The main role of a SP is to instantiate received VNRs by specifying in each VNR, a list of computing and networking resources requirement (CPU, RAM, BW, DEL, Packet loss....). We assume in this architecture layer that a SP specifies in addition the required execution environment scenario of each user (e.g., the delay-sensitive scenario or / the green-aware scenario).

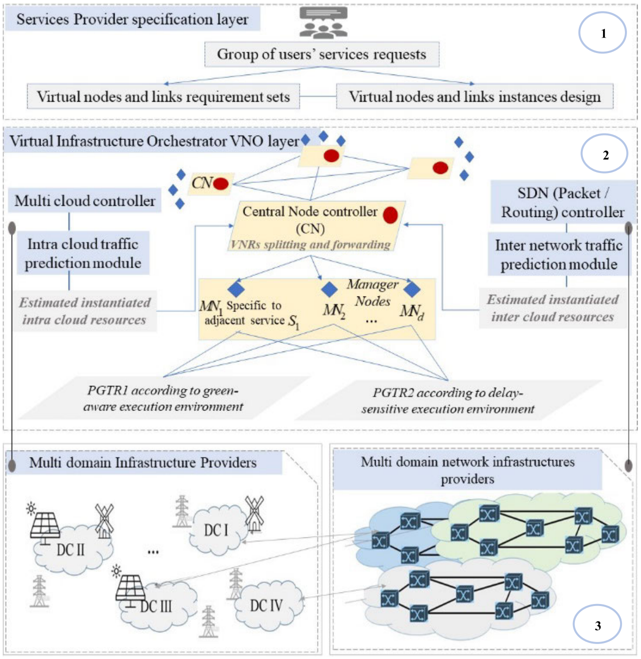

Services Provider specification layer
1
Group of users' services requests
Virtual nodes and links requirement sets
Virtual nodes and links instances design
Virtual Infrastructure Orchestrator VNO layer
2
CN
Multi cloud controller
Central Node controller (CN)
VNRs splitting and forwarding
SDN (Packet / Routing) controller
Intra cloud traffic prediction module
Inter network traffic prediction module
Estimated instantiated intra cloud resources
MN_1 Specific to adjacent service S_1
MN_2
Manager Nodes
...
MN_d
Estimated instantiated inter cloud resources
PGTR1 according to green-aware execution environment
PGTR2 according to delay-sensitive execution environment
Multi domain Infrastructure Providers
DC II
...
DC I
DC III
DC IV
Multi domain network infrastructures providers
3

FIGURE 1. The proposed three-layers SN-ODCN architecture for inter and intra cloud virtual network requests management.

B. VIRTUAL NETWORK CONTROL AND ORCHESTRATION LAYER

This second layer incorporates the main roles of a Virtual Resources Orchestrator (VRO), performed in a hybrid scheme. Time delay and scalability purposes were respected through the consideration of distributed Manager Nodes (MNs). Then, communication overhead imposed on substrate networks is reduced through the consideration of distributed Central Nodes (CNs). This means, a VRO in our architecture encompasses a Central controller Node (*CN* represented in red) connected to a set of three Distributed Manager Nodes (MN_d represented in blue). These two types of nodes are supposed to be hosted on physical nodes having maximum resource capacities (i.e., max. available CPU and bandwidth). The CN is a central control entity involved with the collection of intra cloud DCs status, Multi domain network status, as well as energy availabilities information. For less complexity, we suppose in this layer each CN is connected to three Manager Nodes (MN_1, MN_2, MN_3) performing parallel computing. In this second layer two main functions are executed.

The first splitting and forwarding function is carried by the CN. This first function consists first on splitting VNRs received from SP, into many sub-VNRs according to user required services and user required execution environment scenario (Preference p). Then, forwarding each group of sub-VNRs into the appropriate MN_d. We assume each MN_d is responsible of certain adjacent services, and that the management of those adjacent services are previously established. Then, each MN_d may execute simultaneously both execution environment scenarios. The main target of this first function is to enable parallel executions of many services required by each single user in the same time frame. Figure 2 illustrates how a VNRs group may be split and forwarded into three Sub-VNRs according to three MN_d. Each link between two required services may be categorized with at least one tag. This tag indicates the user index with their preferred execution scenario index: e.g., (p = 1: as a green execution scenario, targeted to some clients seeking resources powered by RE sources, in view of paying less carbon tax) and (p = 2 as a delay-sensitive execution scenario, for some users prioritizing response time). Accordingly, we suppose in this study, each user has the same preferred execution scenario toward its chosen services.

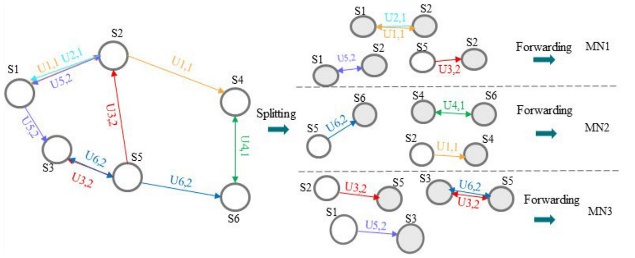


FIGURE 2. A VNRs spample composed of 6 users requesting one of the six services dispatched around three manager nodes (***MN_1, MN_2, MN_3***).

The second function is a Parallel Global Topology Ranking (PGTR) established by each distributed MN_d for its associated Sub-VNRs. In this second function, a MN_d role is to find the appropriate substrate nodes (DCs and hosts) and substrate links from physical IPs. Each MN_d executes two PGTR algorithms depending on received Sub-VNRs' preferred scenario (1 or /2). The green-aware ranking scenario (PGTR1) finds cleanest substrate processing nodes and substrate links while satisfying the virtual nodes and virtual links capacities and delay' constraints. The whole found ranking results is performed based on computed Carbon Emission Rates (CER) values and affected accordingly into three ranking classes (Q1, Q2, Q3) (See details in algorithm 2).

As for delay-sensitive ranking scenario (PGTR2), the closest substrate processing nodes are found and similarly classified into three ranking classes (Q1, Q2, Q3), using a classical minimum hop algorithm. An overall of six GTR algorithms are executed parallelly and simultaneously within the three MN_d, to benefit from a reduced execution time of requests.

On the other hand, while provisioning resources using SP instantiated required resources, it often happens to oversupply the required VNRs' resources. This leads to a wastage of VMs and VLs' assigned resources, being not fully used (Reactive Scenario). It then turns out to be necessary for a VRO to instantiate exact required VNRs' resources based on a prediction of future exact intra and inter cloud handled traffics (Proactive Scenario). In this later scenario, the proposed VRO makes use of inter and intra cloud requests predicted by a temporal-based Bidirectional Gated Recurrent Unit (B-GRU), using historical handled inter and intra VNRs workloads. On this basis, the exact required VNRs' resources are instantiated proactively by the proposed VRO layer.

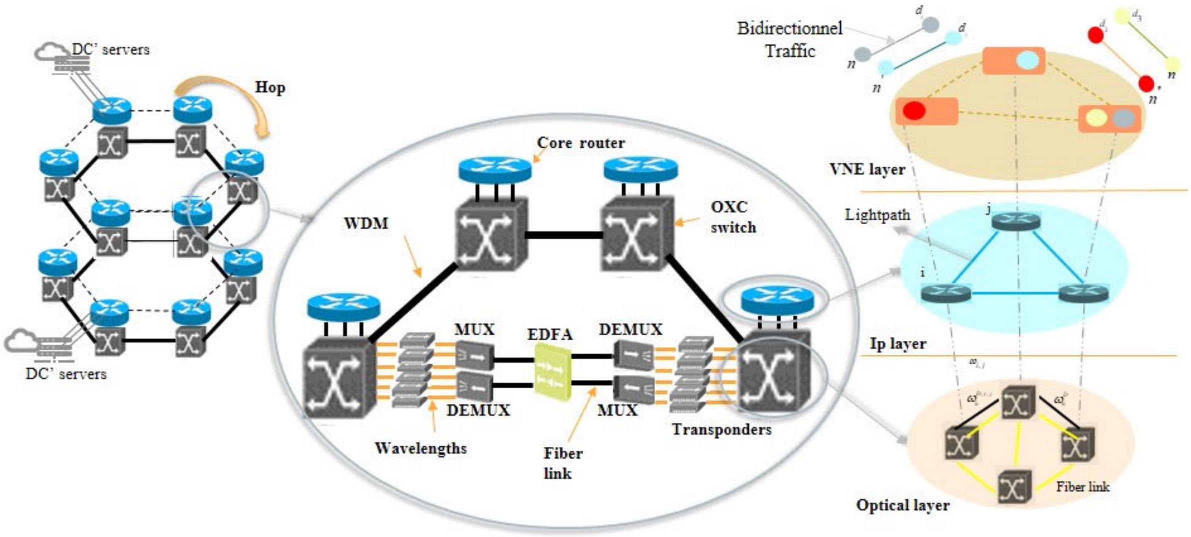


FIGURE 3. Virtual network embedding in the optical and IP layers over the WDM network.

C. MULTI DOMAIN PHYSICAL PROCESSING AND NETWORKING PROVIDERS' LAYER

For the physical layer, multiple geographically distributed IPs are interconnected by multi-domain networks infrastructure to accommodate Sub-VNRs of the VRO layer. This layer carries the last function, which is a proactive virtual nodes and virtual links embedding. In this phase, each MN_d associated Sub-VNRs together with their GTR results are entered to perform in a parallel and proactive scheme the whole Sub-VNRs embedding. This later embedding is supported by a consolidation of virtual nodes and virtual links, to maximize the global resources topology power efficiency ratio. The overall resource topology contemplated in this layer is schematized in above Figure 3. In fact, the Internet core network is composed of large optical switches (OXC) that dynamically configure high-capacity optical networks (WDM). Packets' requests are sent between core routers until arriving to the core router that directly connect to a DC. On each fiber among the total number of hops, where pair of core routers and OXC switches are not collocated, the erbium-doped fiber amplifiers (EDFA) are used to amplify the optical signal in each fiber, between two switches' long-distance transmission. In addition, a pair of multiplexers/demultiplexers are used to multiplex/demultiplex wavelengths. Then, at each switching node, the transponders provide (OEO) processing for full wavelength conversion.

The VNE problem addresses how virtualized resources are to be deployed on the substrate network. Typically, it starts with the mapping of virtual nodes into physical nodes (DCs and hosts), followed accordingly by a mapping of virtual links. Successful embedding of virtual links will therefore require resources in both IP and optical layers. Sub-VNRs' computing requests (CPU and RAM) are integrated into VMs in appropriate DCs' active servers. Then, each Sub-VNRs' bandwidth (BW) demand requires the instantiation of a virtual router in the Ip routers layer, so that the VRO is granted control of the virtual server and router and may configure any protocol to run any application. The following section presents the proposed MILP problem formulation.

IV. NETWORK MODEL AND PROBLEM FORMULATION

A. MODEL'S VARIABLES

We start by introducing respectively in Table 1 and Table 2 the substrate network variable, then the virtual network variables for each time slot t belonging to the total time interval T.

Before describing constraints related to each layer of our architecture, we present below all binary variables used in our model.

$-\alpha_{u,s,d,p}^{t,c,n} = 1$, if required virtual node of user u processing service from $Sub - VNR_{d,p}^{t}$ is mapped into server c in node n. Else $\alpha_{u,s,d,p}^{t,c,n} = 0$.

$-\ \beta_{l\in(u,s,d,p)}^{t,l_{i,j}} = 1$, if required virtual link l forwarding service request of user u from $Sub - VNR_{d,p}^{t}$ is mapped into the lightpath $l_{i,j}$. Else $\beta_{l\in(u,s,d,p)}^{t,l_{i,j}} = 0$.

-$\theta_{h}^{D} = 1$, if the hop h is belonging to network domain D.

$-\ \xi_{D,h}^{t,l_{i,j}} = 1$, if lightpath $l_{i,j}$ is mapped into a hop h of domain network D. Else, $\xi_{D,h}^{t,l_{i,j}} = 0$.

$-\ \chi_{l,l_{i,j}}^{t,D,h} = 1$, if required virtual link l forwarding service request of $Sub - VNR_{d,p}^{t}$ is mapped into virtual lightpath $l_{i,j}$, and lightpath $l_{i,j}$ is mapped into a hop h of domain network D. Else $\chi_{l,l_{i,j}}^{t,D,h} = 0$.

$-\ \sigma_{\omega}^{t,D,h} = 1$, if wavelength w is activated in physical link of a hop h of domain network D. Else $\sigma_{\omega}^{t,D,h} = 0$.

$-\ \varphi_{s,d,p}^{t,n} = 1$, if DC node n is mapped by any required virtual node processing services of $Sub - VNR_{d,p}^{t}$. Else $\varphi_{s,d,p}^{t,n} = 0$.

$-\ \tau_{D}^{t,S}/\lambda_{n}^{t,S} = 1$, if solar energy is present respectively in network domain D and DC node n. Else $\tau_{D}^{t,S}/\lambda_{n}^{t,S} = 0$.

TABLE 1. SUbstrate network variables modeling.

Variables	Significations
$n \in N_S$	A node among total DCs number
$l \in L_S$	A link among total substrate links
MN_d	The Manager Node $d \in \{1,2,3\}$
$Com(n)$	Total computing capacity of node n
$Loc(n)$	DC node location
$P_{s(d\rightarrow n)}$	Substrate path from source d to destination n
$DEL_{(d\rightarrow n)}$	Delay propagation tolerance
$N_{h\ (d\rightarrow n)}^{\ t}$	Total number of forwarded hops
$M_{h,(d\rightarrow n)}^{t}$	Number of hops computed without counting collocated routers and switches
$M'^{t}_{h,(d\rightarrow n)}$	Number of hops computed between collocated routers and switches (When WDM is not used)
$Loc(D)$	Network Domain location
$n \in N_{S_d}$	Number of nodes processing adjacent services S_d associated to MN_d
$c \in NS_n^t$	The set of active servers in DC n
L_h^D	Physical length between two nodes in a hop h in network domain D
$\omega_h^{D,l_{i,j}}$	Number of wavelength channels in virtual lightpath $l_{i,j}$ mapped in a physical link between two nodes in hop h in network domain D
C	Bandwidth capacity per wavelength
W	Number of wavelengths in each fiber link

– $\tau_D^{t,W}/\lambda_n^{t,W} = 1$, if wind energy is present respectively in network domain D and DC node n. Else $\tau_D^{t,W}/\lambda_n^{t,W} = 0$.

– $\tau_D^{t,N}/\lambda_n^{t,N} = 1$, if nuclear energy is present respectively in network domain D and DC node n. Else $\tau_D^{t,N}/\lambda_n^{t,N} = 0$.

– $\tau_D^{t,B}/\lambda_n^{t,B} = 1$, if brown energy is present respectively in network domain D and DC node n. Else $\tau_D^{t,B}/\lambda_n^{t,B} = 0$.

TABLE 2. Virtual network variables modeling.

Variables	Significations
VNR_a^t	A group $a \in \{1,\ldots,N_{VNR}\}$ of users Virtual Network Requests
N_v	Set of virtual nodes
L_v	Set of virtual links
$V_{n,c}$	Set of adjacent virtual nodes collocated with the required virtual node in server c and node n
$V_{l,\omega}$	Set of adjacent virtual links collocated with required virtual link in wavelength ω
U_a	Total number of users in the VNR_a group
$P_{u,a}$	A user preferred execution environment scenario
$Sub-VNR_{d,p}^t$	Sub-VNR containing users requiring one or all adjacent services S_d associated to MN_d and preferring the execution scenario p
(i,j)	Two vertices of a virtual lightpath

– $\Delta_D^{t,RE}/\Delta_n^{t,RE} = 1$, if there is no RE (solar/ wind/ nuclear) respectively in network domain D and DC node n. Else $\Delta_D^{t,RE}/\Delta_n^{t,RE} = 0$.

B. MODEL'S CONSTRAINTS

1) SPLITTING AND FORWARDING CONSTRAINTS

$$Com(VNR_a^t) = \sum_{d=1}^{3}\sum_{p=1}^{2} com(Sub - VNR_{d,p}^t), \times \forall a \in \{1,\ldots,N_{VNR}\} \tag{1}$$

$$B(VNR_a^t) = \sum_{d=1}^{3}\sum_{p=1}^{2} B(Sub - VNR_{d,p}^t), \times \forall a \in \{1,\ldots,N_{VNR}\} \tag{2}$$

Equations (1) and (2) verify respectively if all computing and bandwidth resources required by the group of $VNR_{d,p}^t$ are similarly the sum of computing and bandwidth resources required by $Sub - VNR_{d,p}^t$ for each MN_d associated Sub-VNRs, in both execution scenarios p.

2) GLOBAL TOPOLOGY RANKING CONSTRAINTS

Unlike previous GTR approach considering only global topological attributes (i.e., available computing resources and bandwidth resources) while evaluating virtual nodes and links embedding, we aim to apply a new GTR based on resources' availability as well as resource's CER values results. Considering available computing and bandwidth resources, we choose to represent each server' computing capacities and utilizations by the sum of their CPU and Memory magnitude. Similarly, we choose to represent each network' bandwidth by the sum of their wavelengths' bandwidths and delay tolerances magnitude. The available computing resources of DC node n at time slot t, is expressed by equation (3) as the difference between total servers' capacity at node n (4) and whole servers' utilizations at this same time slot (5).

$$AC^t(n) = \sum_{c=1}^{NS_n^t} (Com_c(n) - U_c^t(n)), \quad \forall n \in \{1, \ldots, N_d\} \tag{3}$$

$$Com_c(n) = \sqrt{(CPU_c)^2 + (Memory_c)^2}, \quad \forall c \in \{1, \ldots, NS_d\} \tag{4}$$

$$U_c^t(n) = \sqrt{(\sum_{v \in V_{n,c}} CPU_{v,c})^2 + (\sum_{v \in V_{n,c}} Memory_{v,c})^2}, \times \forall c \in \{1, \ldots, NS_d\} \tag{5}$$

Similarly, the available wavelength bandwidth resources of network path between node d and DC n at time slot t is computed in equation (6), as the difference between the sum of bandwidth resources capacities in this network path (7) and the utilized bandwidth resources in this network path (8).

$$AB^t_{(d \to n)} = \sum_{D=1}^{D_n} (\sum_{h=1}^{N^t_{h(d \to n)}} (B^D_{(h-1 \to h)} - BU^{t,D}_{(h-1 \to h)}).\theta_h^D) \tag{6}$$

$$B^D_{(h-1 \to h)} = \sqrt{(\sum_{\omega \in \omega_h} BC(\omega))^2 + (\sum_{\omega \in \omega_h} DEL(\omega))^2}, \times \forall D \in \{1, \ldots, D_n\} \tag{7}$$

$$BU^{t,D}_{(h-1 \to h)} = \sqrt{(\sum_{\omega \in \omega_h} (BC(\omega).\sigma_\omega^{t,D,h}))^2 + (\sum_{\omega \in \omega_h} (DEL(\omega).\sigma_\omega^{t,D,h}))^2} \times \forall D \in \{1, \ldots, D_n\} \tag{8}$$

Once resources' availabilities computed, the proposed GTR approach underpins a CER based ranking approach. Equations (9) and (10) represent respectively CER values of DC node n and its associated network path, where $EW_{Loc(n)}, EW_D, ES_{Loc(n)}, ES_D, EN_{Loc(n)}, EN_D$ and $EB_{Loc(n)}, EB_D$ are respectively the equivalent Co2 emission per KWh of wind, solar, nuclear and brown energies in different nodes' locations n and different networks domains' locations D. Then, $AW_n^t, AW_D^t, AS_n^t, AS_D^t$, AN_n^t, AN_D^t and AB_n^t, AB_D^t refer respectively to available wind, solar, nuclear and brown energies' alimentations in different DCs nodes n and network domains D at time slot t.

$$CER^t(n) = (EW_{Loc(n)}.AW_n^t).\lambda_n^{t,W} + (ES_{Loc(n)}.AS_n^t).\lambda_n^{t,S} + (EN_{Loc(n)}.AN_n^t).\lambda_n^{t,N} + (EB_{Loc(n)}.AB_n^t).\lambda_n^{t,B}.\Delta_n^{t,RE} \tag{9}$$

$$CER^t_{net}(d \to n) = \sum_{h=1}^{N^t_{h(d \to n)}} [\sum_{D=1}^{D_n} \times \begin{pmatrix} (EW_D.AW_D^t).\tau_D^{t,W} + (ES_D.AS_D^t).\tau_D^{t,S} + \\ (EN_D.AN_D^t).\tau_D^{t,N} + (EB_D.AB_D^t).\tau_D^{t,B}.\Delta_D^{t,RE} \end{pmatrix}.\theta_h^D] \tag{10}$$

In addition, we present power consumption models of a DC node (11) as the power consumption of their active servers, considering each server's associated virtual computing resources and the base power of an active server $P_{base,c}$ [11]. Then, μ_c is the power gap between server's idle power and maximum power. Thereafter, we present just after in (12) the power consumption of their associated network path, where $P_T, P_E, P_{MD}, P_R, P_O$ are respectively power consumptions of transponder, EDFA amplifier, multiplexer/demultiplexer, router and optical switch.

$$P_{DC}(n) = \sum_{c=1}^{NS} \left[(\sum_{v \in V_{n,c}} Com_{c,v}.\mu_c) + P_{base,c} \right] \tag{11}$$

$$P_{net(d \to n)} = \sum_{h=1}^{M^t_{h(d \to n)}} \left[\left(P_T.\omega_h^D\right) + (P_E.EA_h) + (P_{MD}.DM_h) + \sum_{l_{i,j}=1}^{L_{i,j}} (P_R.\omega_h^{D,l_{i,j}}) + P_O \right] + \sum_{h'=1}^{N^t_{h(d \to n)}} \left[\sum_{l_{i,j}=1}^{L_{i,j}} (P_R.\omega_h^{D,l_{i,j}}) + P_O \right] \tag{12}$$

3) VIRTUAL NODES EMBEDDING CONSTRAINTS

Equation (13) guarantees that all required active servers in each DC node n, processing adjacent services s associated to a MN_d, should provide adequate computing resources for all virtual nodes that are accommodated by this MN_d. In (14), each required virtual node processing service s of $Sub - VNR^t_{s,d,p}$ can only be hosted once in a unique DC n and unique server c.

$$\sum_{c=1}^{NS} \left(\sum_{v \in V^t_{n,c}} Com_{v,c} + \sum_{v \in Sub - VNR_{d,p}} Com(Sub - VNR_{d,p}) \right) \leq AC^t(n) \tag{13}$$

$$\sum_{n=1}^{N} \sum_{c=1}^{NS} \alpha^{t,c,n}_{u,s,d,p} = 1, \quad \forall s \in S_d \tag{14}$$

$$\varphi_{s,d,p}^{t,n} = \sum_{\forall(s,u)\in Sub-VNR_{d,p}} \alpha_{u,s,d,p}^{t,c,n} \tag{15}$$

4) VIRTUAL LINKS EMBEDDING CONSTRAINTS

Virtual links embedding constraints are shown from (15) to (21). Constraint (17) ensures the total carried bandwidth resource cannot exceed the occupied wavelength capacity, on lightpath (i, j). Then the flow conservation constraint for respectively the optical and IP layers is shown in (17) and (18). Equation (19) ensures the sum of all traffic flows through an optical link in a hop h does not exceed its available bandwidth capacity. Equation (20) ensures the number of allocated wavelengths cannot exceed the initial capacity of an optical link. The total number of occupied wavelength channels for an optical link is shown in (21).

$$\sum_{\forall l\in Sub-VNR_{d,p}} \beta_{l\in(u,s,d,p)}^{t,l_{i,j}}.B(Sub-VNR_{d,p}) \leq \omega_{l_{i,j}}.C \tag{16}$$

$$\sum_{h\in N_{h(d\rightarrow n)}^{t}} \omega_h^{D,l_{i,j}} - \sum_{h\in N_{h(n\rightarrow d)}^{t}} \omega_h^{D,l_{i,j}} = \begin{Bmatrix} \omega_h^D \\ -\omega_h^D \\ 0 \end{Bmatrix}, \begin{matrix} i\rightarrow h\in N_{h(d\rightarrow n)}^t \\ i\rightarrow h\in N_{h(n\rightarrow d)}^t \\ Others \end{matrix} \tag{17}$$

$$\sum_{h\in N_{h(d\rightarrow n)}^{t}} \xi_{D,h}^{t,l_{i,j}} - \sum_{h\in N_{h(n\rightarrow d)}^{t}} \xi_{D,h}^{t,l_{i,j}} = \begin{Bmatrix} 1 \\ -1 \\ 0 \end{Bmatrix}, \begin{matrix} i\rightarrow h\in N_{h(d\rightarrow n)}^t \\ i\rightarrow h\in N_{h(n\rightarrow d)}^t \\ Others \end{matrix} \tag{18}$$

$$\sum_{\forall l\in Sub-VNR_{d,p}} \chi_{l,l_{i,j}}^{t,D,h}.B(Sub-VNR_{d,p}) \leq \omega_h^{D,l_{i,j}}.AB_h^t \tag{19}$$

$$\sum_{i\in h}\sum_{j\in h(i\neq j)} \omega_h^{D,l_{i,j}} \leq W, \forall h\in N_{h(d\rightarrow n)}^t \tag{20}$$

$$\sum_{i\in h}\sum_{j\in h(i\neq j)} \omega_h^{D,l_{i,j}} = \omega_h^D, \forall h\in N_{h(d\rightarrow n)}^t \tag{21}$$

C. OBJECTIVE FUNCTIONS

The objective function of the proposed MILP model consists on maximizing total power efficiency of each processing DC node with its associated network path (22). Toward this end, we design a joint power consumption and user request rate ratio to compute power efficiency in both DC nodes (23) and their associated network path (24). Where, r_{cu}^t and r_{hu}^t are the number of traffic requests of user u in respectively a server c and in a hop h.

$$Max : TP_{eff} = P_{eff,DC(n)} + P_{eff,net(d\rightarrow n)} \tag{22}$$

$$P_{eff,DC(n)} = \frac{\sum_{c=1}^{NS}\sum_{u=1}^{U_{S_d}^t} r_{cu}^t \cdot \alpha_{u,s,d}^{t,c,n}}{\sum_{c=1}^{NS}\left[\left(\sum_{v\in V_{n,c}} Com_{c,v}\cdot\mu_c\right) + P_{base,c}\right]}, \times \forall s\in S_d \tag{23}$$

$$P_{eff,net(d\rightarrow n)} = \frac{\sum_{h=1}^{N_{h(d\rightarrow n)}^t}\sum_{u=1}^{U_{S_d}^t} r_{hu}^t.\beta_{l\in(s,d,p)}^{t,l_{i,j}}.\xi_{D,h}^{t,l_{i,j}}}{P_{net(d\rightarrow n)}}, \quad \forall s\in S_d \tag{24}$$

Constraints to: (1) - (2)
(3) - (12)
(13) - (15)
(16) – (21)

V. ALGORTIHMS DESIGNS

A. SPLITTING AND FORWARDING ALGORTIHM

This first algorithm inputs VNR_a^t as a 2D table of size $(U \times S)$. The table's cells incorporate some tuples of computation and networking resources for each user u requesting a service s, otherwise 0 when the service is not requested. The second input is $MNA_{services}$, which is a 2D binary table referring to 1 if the service s is belonging to Manager Node $d(MN_d)$ otherwise 0. Then, $P_{u,a}$ is a list of size U including the preference scenario of each user in the group of requests. Lines (1-3) verify if a service is belonging to a MN_d. Once the condition is verified, the algorithm verifies again in lines (4-5) whether this service is requested by some users. If so, the algorithm checks the user's preference in terms of execution scenario (1: Green-aware of 2: delay-sensitive), and creates accordingly a $Sub-VNR_{d,p}$ including Sub-VNRs associated to the MN_d and the preference scenario p (Lines 6-13). Considering three MN_d and two execution environments preferences, the algorithm outputs a maximum of six $Sub-VNR_{d,p}$. The worst-case complexity of this first algorithm is $o(3US)$, as we exactly deploy in this work three MN_d.

Algorithm 1 : Splitting and Forwarding

```
Input: $VNR_a^t$ = Tab[$u \in \{1...U\}$,
$s_1 \in \{1...s\}$],
$MNA_{services}$ = Tab[$s_1 \in \{1...S\}, d \in \{1...3\}$], $P_{u,a}$
Output: $Sub-VNR_{d,p} = Tab[u \in \left\{1...U_{S_d}^t\right\}$,
$s \in \{1...S_d\}$],
1:  for $d = 1$ to 3 do
2:      for $s_1 = 1$ to $S$ do
3:          if $MNA_{services}[s_1, d] \neq 0$ then
4:              for $u = 1$ to $U$ do
5:                  if $VNR_a^t[u, s_1] \neq 0$ then
6:                      $p \leftarrow sort.P_{u,a}[u]$,
7:                      create $Sub-VNR_{d,p}\left[U_{S_d}, S_d\right]$,
8:                      add. $VNR_a^t[u, s_1]$ in
                        $Sub-VNR_{d,p}$,
9:                  end if $u++$ (Return to 4)
10:             end for
11:         end if $s_1++$ (Return to 2)
12:     end for
13:     sort $Sub-VNR_{d,p}$, $d++$(Return to 1),
14: end for
```

Algorithm 2 : PGTR1

```
Input: $Com(Sub - VNR_{d,1})$, $B(Sub - VNR_{d,1})$
$R = Tab[1...2, r \in \{1...N_{S_d} = L_{S_d}\}], \forall d \in \{1...3\}$
Output: $Q_1 = Tab[1...2, 1...L_1]$,
$Q_2 = Tab[1...2, 1...L_2]$
$Q_3 = Tab[1...2, 1...L_3]$
Initialization:
$AR = Tab[1...2, r' \in \{1...N'_{S_d} \le N_{S_d}\}] = 0$,
$CER = Tab[1...2, r' \in \{1...N'_{S_d}\}] = 0$,
$T_1 = V_1, T_2 = V_2, Q_1 = 0, Q_2 = 0, Q3 = 0$
1: for $r = 1$ to $N_{S_d}$ do
2:     compute $AC^t(R[1, r])$,                Equation (3)
3:     compute $AB^t_{(d \to n)}(R[2, r])$,    Equation (6)
4:     $((AC^t(R[1, r]) \succ Com(Sub - VNR^t_{d,p}))\&$
5:     if                                        then
           $(AB^t_{(d \to n)}(R[2, r]) \succ B(Sub - VNR^t_{d,p})))$
6:         add. $R[1, r]$ in $AR[1, r']$,
7:         add. $R[2, r]$ in $AR[2, r']$,
8:     end if $r++$ (Return to 1),
9: end for
10: sort $AR$,
11: for $r' = 1$ to $N'_s$ do
12:     compute $CER^t(AR[1, r'])$,               Equation (9)
13:     compute $CER^t_{net,(d \to n)}(AR[2, r'])$, Equation (10)
14:     add. $CER^t(AR[1, r'])$ in $CER[1, r']$,
15:     add. $CER^t_{net(d \to n)}(AR[2, r'])$ in $CER[2, r']$,
16:     if $((CER^t[1, r'] \le V_1)\&(CER^t[2, r'] \le V_2))$
        then
17:         add. $AR[1, r']$ in $Q_1[1, r']$,
18:         add. $AR[2, r']$ in $Q_1[2, r']$,
19:     else
20:         compute $P^t_{DC}(AR[1, r'])$ ,   Equation (11)
21:         compute $P^t_{net}(AR[2, r'])$,   Equation (12)
22:         if                                then
                $((CER^t[2, r'] \succ V_2)\&$
                $(P^t_{DC}(AR[1, r']), \succ P^t_{net}(AR[2, r'])))$
23:             add. $AR[1, r']$ in $Q_2[1, r']$,
24:             add. $AR[2, r']$ in $Q_2[2, r']$,
25:         else
26:             add. $AR[1, r']$ in $Q_3[1, r']$,
27:             add. $AR[2, r']$ in $Q_3[2, r']$,
28:         end if
29:     end if
30:     Return to 11,
31: end for
32: sort $Q_1, Q_2, Q_3$
```

B. PARALLEL GLOBAL TOPOLOGY RANKING ALGORITHM FOR THE GREEN-AWARE SCENARIO

The second proposed algorithm is a Parallel Global resources Topology Ranking performed in the case of the chosen execution scenario $p = 1$. This second algorithm enters the following inputs: first, $Com(Sub - VNR_{d,1})$ as the magnitude (CPU, RAM) values required by the $Sub - VNR_{d,1}$ as proposed in (equation 4). Second, $B(Sub - VNR_{d,1})$ is similarly the magnitude of (BW, deadline) values required by the same $Sub - VNR_{d,1}$ as demonstrated in (equation 7). Third, a 2D table R containing all MN_d associated candidate DCs nodes in the first line, with their associated network paths trajectories in the second table line. From initialization to line 10, the algorithm checks DCs and networks' resources availabilities for each request in the subgroup, and stores nodes with sufficient capacities in an initially empty table AR. Lines (11-32) incorporate the ranking process. At this level, the algorithm calculates and stores in lines (11-15) the CER values of each DC (equation 9) and its associated network (equation 10). If the two obtained CER values are well below CER thresholds initialized at the beginning (V1= V2= 500 KgeqCO2) (line 16), indicating thus the DC with its associated network are green, their indices are therefore stored in the first class Q1 (lines 17-18). In case the complete condition is not satisfied (line 19), the algorithm extends the Ranking process based on DC power consumption (line 20 - equation 11) and network power consumption (line 21 - equation 12) at time t. The condition of (line 22) verifies then the compensation of the brown power consumed in network by the green power consumed in intra DC. Once this compensation is satisfied, the DC index r' with its associated network path is stored in Q2 class (lines 23-24). The remaining resources' indices satisfying the remaining cases (line 25) where both CER values are higher than the fixed thresholds or where network's brown power consumption is not compensated by that green one consumed in DC, are then classified in the last class Q3 (lines 26-27). At the end (line 32), the three obtained Ranking classes are sorted. Considering three MN_d and two execution environment scenarios, we perform a maximum of three-parallel execution of (PGTR1) algorithm according to preferred execution scenario $p = 1$. This was executed simultaneously with a maximum of three-parallel execution of a classical minimum hop (PGTR2) algorithm, according to preferred execution scenario $p = 2$. At this stage, a total of 6 (PGTR) algorithms will be performed parallelly. The computed worst-case complexity of (PGTR1) algorithm is $\boldsymbol{O}(N_{S_d}.NS^t_n.N^t_{h(d \to n)}) + \boldsymbol{O}(N_{S_d})$.

C. PARALLEL AND PROACTIVE VIRTUAL NETWORK EMBEDDING

The VNE algorithm (Algorithm 5) requires two sub-algorithms: a Virtual Node Embedding (VNoE: Algorithm 3) and a Virtual Link Embedding (VLiE: Algorithm 4). For each MN_d, the above VNoE (algorithm 3) enters the appropriate proactive instantiated $Sub - VNR_{d,1}$ together with the three Ranking classes Q_1, Q_2, Q_3 results, sorted by previous PGTR algorithm of the same MN_d. As long as there are enough resources' nodes in the first Q1 class (line 1), the algorithm outputs for each DC node along the total Q1 list length (L_1) (line 3), a list of active servers RNS in an ascending order of their residual capacities (line 4). For each server index c(line 5), the algorithm outputs similarly a list its running

Algorithm 3 : VnoE

```
Input: $Sub-VNR_{d,p} = Tab[u \in \left\{1...U^t_{S_d}\right\}$,
$s \in \{1...S_d\}]$,
$Q_1, Q_2, Q_3 \quad \forall d \in \{1...3\}, \quad \forall p \in \{1...2\}$
Output: $E_{r',c,v}$, $P_{eff,DC(r')}$
Initialization: $E_{r',c,v} = 0$, $P_{eff,DC(r')} = 0$,
1:  while $(r' \in (Q = Q_1) \leftarrow true)$
2:      for $u = 1$ to $U_{S_d}$ do
3:          for $r' = 1$ to $L_1$
4:          $R_{NS}(r') \leftarrow$ sort $NS(r')$,
5:          for $c = 1$ to $LR_{NS}$ do
6:              $R_v(c) \leftarrow$ sort $V_{r',c}$,
7:              for $v = 1$ to $LR_V$ do
8:                  if $(Com(Sub-VNR_{d,p}[u,s]) \prec$
                    $Com(R_v[v]))$
9:                      sort $E_{r',c,v}$,
10:                     update resources,
11:                 end if    $v++$ (Return to 7)
12:             end for
                $(E_{r',c,v} = 0$ &
13:             if                                  then
                    $(Com(Sub-VNR_{d,p}[u,s]) \prec$
                    $Com(R_{NS}[c]))$
14:                 create a new VM in c,
15:                 sort $E_{r',c,v}$,
16:                 update resources,
17:             end if    $c++$ (Return to 5)
18:         end for
19:         end for
20:         compute $P_{eff,DC(r')}$,          Equasion (23)
21:     end for
22: end while $(Q++)$
23: end
```

VMs (R_v) in an ascending order of their residual capacities (line 6). This ascending order sorting based on residual capacities is used for consolidation purposes, avoiding hence energy-intensive consuming inter migrations.

For each selected VM index v (line 7), the condition in (line 8) verifies the availability of a user requested (CPU, RAM) values, within the selected VM residual capacities. Once the condition is satisfied in a VM, the algorithm outputs in (line 9) the index of this allocation and updates resources (line 10) for consideration during a next single user's requests. If the condition is not met for any VM, while there is in addition sufficient residual capacity in the selected server (line 13), the algorithm proceeds to create a new VM (line 14) with the requested capacity, and similarly updates resources (line 16). After finding the allocation indices (line 15), the algorithm calculates and outputs in (line 20) the DC's power efficiency resulting from this allocation. The computed worst-case of VnoE algorithm is $\boldsymbol{O}(U_{S_d}.L_1.LRNS.LR_v)$.

Again, the following Virtual link Embedding (algorithm 4) inputs the previous (algorithm 3) output, which is: the virtual node allocation indexes $E_{r',c,v}$, along with the associated Manager Node proactive instantiated Sub-VNR. For each hop (line 2) in the network path associating the MN_d source

Algorithm 4 : VLiE Pseudo Code

```
Input: $Sub-VNR_{d,p} = Tab[u \in \left\{1...U^t_{S_d}\right\}$,
$s \in \{1...S_d\}]$, $E_{r',c,v}$ $\forall d \in \{1...3\}$, $\forall p \in \{1...2\}$
Output: $E_{h,l_{i,j}}$, $P_{eff,net(d \to r')}$
Initialization: $E_{h,l_{i,j}} = 0$, $P_{eff,net(d \to r')} = 0$,
1:  for $u = 1$ to $U_{S_d}$
2:      for $h = 1$ to $N^t_{h(d \to r')}$
3:          $Rw(h) \leftarrow$ sort the total number of residual
                wavelength $\omega^D_h$,
4:          $R_{l_{i,j}}(h) \leftarrow$ sort the total virtual lightpaths $L_{i,j}$
5:          for $l_{i,j} = 1$ to $LR_{l_{i,j}}$
6:              if
                $(B(Sub-VNR_{d,p}[u,s]) \prec B\left(R_{l_{i,j}}\left[l_{i,j}\right]\right))$
7:                  sort $E_{h,l_{i,j}}$,
8:                  update resources,
9:              end if    $l_{i,j}++$ (Return to 5)
10:         end for
            $((E_{h,l_{i,j}} = 0)$ &
11:         if                                          then
                $(B(Sub-VNR_{d,p}[u,s]) \prec B(Rw[h])))$
12:             create new virtual lightpath$l$
13:             sort $E_{h,l_{i,j}}$,
14:             update resources,
15:         end if    $h++$ (Return to 2)
16:     end for
17:     Compute $P_{eff,net(d \to r')}$,          Equation (24)
18: end for
19: end
```

Algorithm 5 : PPVNE Pseudo Code

```
Input: $Sub-VNR_{d,p} = Tab[u \in \left\{1...U^t_{S_d}\right\}$,
$s \in \{1...S_d\}]$,
$Q_1, Q_2, Q_3 \forall d \in \{1...3\}, \forall p \in \{1...2\}$
Output:$TP_{eff}$, $A_R$
Initialization: $TP_{eff} = 0$, $A_R = 0$,
1:  for $u = 1$ to $U_{S_d}$
2:  execute VNoE using algorithm 3,
3:  if $E_{r',c,v}$ then
4:      execute VLiE using algorithm 4,
5:      if $E_{h,l_{i,j}}$ then
6:          compute $TP_{eff}$,
7:      else
8:          reject the request,
9:          update acceptance ratio $A_R$,
10:     end if
11: else
12:     reject the request,
13:     update acceptance ratio $A_R$,
14: end if
15: sort $TP_{eff}$, sort $A_R$,          Equation (22)
16: end
```

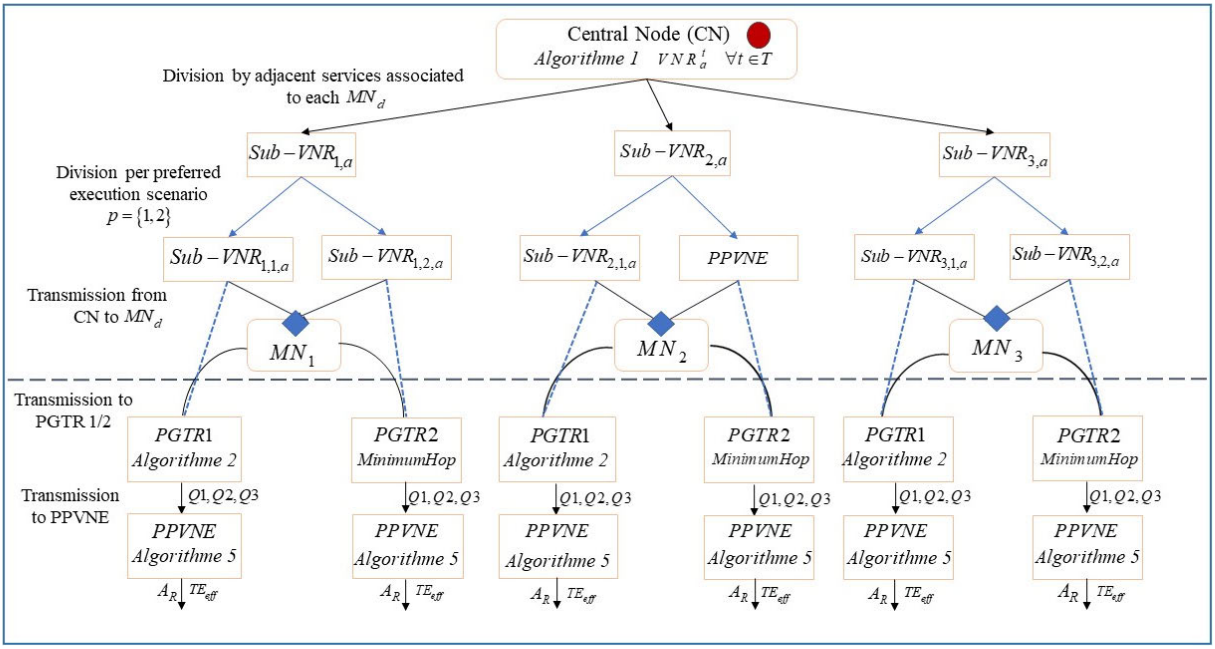


FIGURE 4. Synthesis and summary of the whole implemented algorithms.

with the selected DC index destination sorted by previous (algorithm 3), the algorithm stores the total number of available residual wavelengths (i.e., not assigned to any virtual lightpath) in a list $Rw(h)$ (line 3). Similarly, the algorithm stores all existing virtual lightpaths in a list $R_{l_{i,j}}(h)$, in an ascending order of their residual wavelength's capacities (line 4). The condition in (line 6) verifies the availability of requested bandwidth (BW) within the residual wavelength capacity of the selected virtual lightpath. Once the condition is satisfied, the algorithm outputs the allocation index (line 7) and similarly updates resources (line 8). If no virtual lightpath has the residual capacities satisfying the requested bandwidth, and there is sufficient residual capacity in the remaining hop's wavelengths (line 11), the algorithm creates new virtual lightpath (line 12) and accordingly updates resources (line 13). Once the allocation indices are found, the algorithm calculates and outputs in (line 16), the power efficiency of the network path linked to DC index r'. The computed worst-case complexity of VLiE algorithm is $\boldsymbol{O}(U_{S_d}.N^t_{h(d\rightarrow r')}.LR_{l_{i,j}})$.

If previous both algorithms (3 and 4) are jointly well executed, the global Parallel and Proactive VNE (Algorithm 5) calculates the overall power efficiency TP_{eff}. Otherwise, an acceptance ratio A_R is computed. The worst-case complexity of this global PPVNE algorithm becomes then the sum of previous algorithms' (3 and 4) complexities. At this level, six PPVNE algorithms are again executed parallelly according to the six previous sorted three ranking classes. Figure 4 illustrates how the set of implemented algorithms are connected to be executed in parallel.

VI. PERFORMANCE EVALUATION

We simulated the proposed Green and Distributed SD-ODCN architecture under NSFNET network topology as mentioned in (Figure 5), considering availabilities of different kinds of energy. We have implemented algorithms synthesized in (figure 4) on the latest version of CloudSim [23]. We extended the following classes: the CloudCoordinator class, network topology classes, resources provisioning classes, cloudlet scheduler classes, as well as allocation policies classes. Experiments were run on a DELL G5-15-5590 Intel(R) (core (TM) i7-9750H, 12 CPUs (2.6GHz), 16GB DDR4 RAM). For evaluation purpose, we compared our simulated results with four benchmark algorithms presented in [11]. The following (Section VI.A) highlights the simulation setups with data collection sources and further simulation details. Then, performance evaluation metrics and all obtained results analysis were discussed in (Section VI.B).

A. SIMULATION SETUP AND DATA COLLECTION SOURCES

Following benchmark algorithms [11], we simulated our model under the NSFNET network topology shown in (Figure 5). As a first step, we created our NSFNET network topology using a BRITE network generator. To route therefore different links, we used an extended CloudsimSDN, where we incorporated the WDM optical network depicted in Figure 3. After implementing our solution, we performed the following three configurations: A physical topology configuration including physical hosts, switches/routers and WDM physical links formed between our NSFNET DCs' nodes; a virtual topology configuration consisting of virtual machines and virtual links between VMs; Finally, a workload configuration.

The simulated NSFNET network topology of (Figure 5) contains 14 DCs located in 14 different geographical locations in the USA. Each node is represented by its ID and location. Additionally, inspired by The Global IT Network

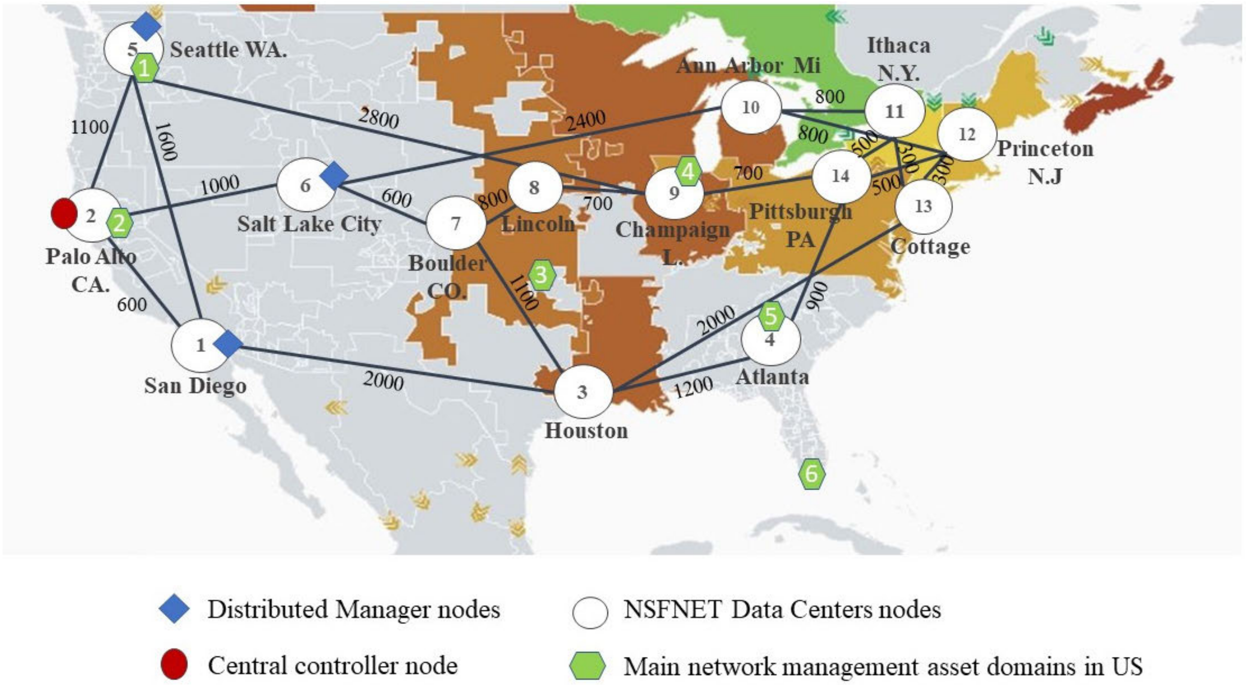


FIGURE 5. The simulated NSFNET network, drawn on the top of the ElectricityMap.

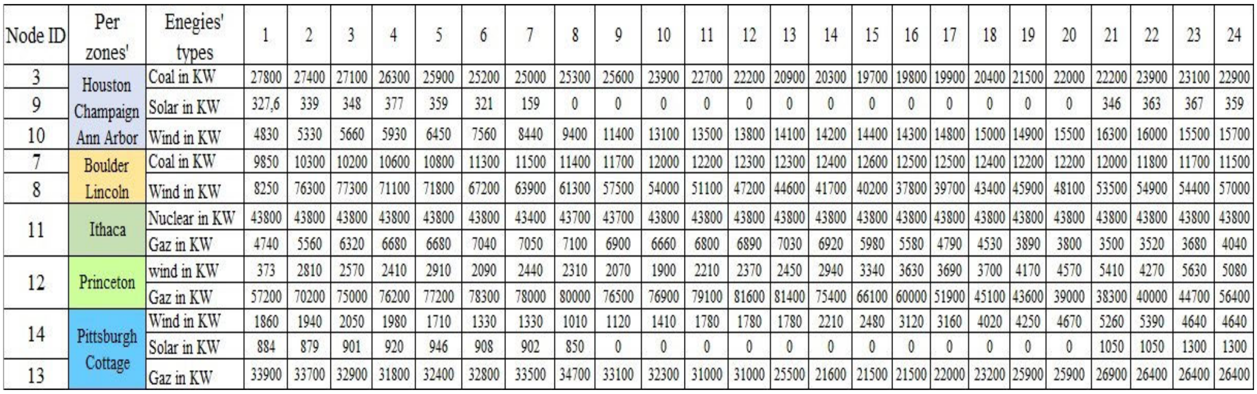
Communication in USA [24], we have chosen to categorize all NSFNET network nodes by network management categories. Indeed, examples of "green nodes" depicted in (Figure 5) refer to localizations of main network management assets in 'The Global IT Network' service. This solution [24] provides 1Gbps to 100Gbps optical fibers connecting hundreds of markets and being able to combine the performance of Internet services with that of cloud computing DCs. This network service is comprised of reliable technologies and vendors such as Cisco, Juniper, and Tier 1 network operators such as: Verizon, GTT, Internap, PCCW, Centurylink, ATT, etc.... All DCs and network nodes of this topology were mapped on the top of electricity and carbon emissions map, provided by the ElectricityMap platform [25] from which energy data were collected. As a large part of the network lacks data related to energy availability, we have chosen on this basis to assign to nodes in this area the roles of Central Controller (indicated in red) and of the three MN_d (indicated in blue). The following (Table 3) summarizes a taxonomy of DCs handling adjacent services of each MN_d, as well as DCs nodes' taxonomy by inter-network category.

In (Table 4), we provide the 24h gathered data on the availability of different types of energy at each node location in the NSFNET network. We were also inspired by the International Energy Agency (IEA) platform [26], to gain knowledge about dominant energy types in each region. According to the recent report on Energy Innovation (Policy and technology LLC) [27] global DCs likely consumed around 205 terawatt-hours (TWh) in 2018, the equivalent of 1% of global electricity use. Driven by this fact, we extracted only 1% of each collected energy availability. Therefore, we converted all obtained data in KW. From the same platform, we also collected carbon intensity in Kg/KWh associated to USA region as depicted in Table 5. As summarized in (Table 6), we tried to keep the same simulation setup [11] to remain compatible with benchmark algorithms. Therefore, we considered additionally hosts and VMs characteristics shown in (Table 7).

TABLE 3. Taxonomies of nsfnet dcs nodes dedicated to each manager node and network management domain.

Description	ID	LOCATION ZONES
DCs nodes processing adjacent services $S_1 = \{1, 2\}$ associated to MN_1 .	9	Champaign L.
	14	Pittsburgh PA.
	12	Princeton N. J
DCs nodes processing adjacent services $S_2 = \{4, 6\}$ associated to MN_2 .	7	Boulder CO
	8	Lincoln
	10	Ann Arbor Mi
	11	Ithaca N. Y
DCs nodes processing adjacent services $S_3 = \{3, 5\}$ associated to MN_3	3	Houston
	4	Atlanta
	13	Cottage
DCs nodes connected to network management domain whose ID=4	9	Champaign L.
	12	Princeton N. J
	10	Ann Arbor Mi
	14	Pittsburgh PA
	11	Ithaca N. Y
DCs nodes connected to network management domain whose ID=3	3	Houston
	7	Boulder CO
	8	Lincoln
	4	Atlanta
DCs nodes connected to network management domain whose ID=5	4	Atlanta
	12	Princeton N. J
	14	Pittsburgh PA.
	13	Cottage
	3	Houston

TABLE 4. Energy availabilities at nsfnet nodes from electrictymap platform.



Node ID	Per zones'	Enegies' types	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
3	Houston	Coal in KW	27800	27400	27100	26300	25900	25200	25000	25300	25600	23900	22700	22200	20900	20300	19700	19800	19900	20400	21500	22000	22200	23900	23100	22900
9	Champaign	Solar in KW	327,6	339	348	377	359	321	159	0	0	0	0	0	0	0	0	0	0	0	0	0	346	363	367	359
10	Ann Arbor	Wind in KW	4830	5330	5660	5930	6450	7560	8440	9400	11400	13100	13500	13800	14100	14200	14400	14300	14800	15000	14900	15500	16300	16000	15500	15700
7	Boulder	Coal in KW	9850	10300	10200	10600	10800	11300	11500	11400	11700	12000	12200	12300	12300	12400	12600	12500	12500	12400	12200	12200	12000	11800	11700	11500
8	Lincoln	Wind in KW	8250	76300	77300	71100	71800	67200	63900	61300	57500	54000	51100	47200	44600	41700	40200	37800	39700	43400	45900	48100	53500	54900	54400	57000
11	Ithaca	Nuclear in KW	43800	43800	43800	43800	43800	43800	43400	43700	43700	43800	43800	43800	43800	43800	43800	43800	43800	43800	43800	43800	43800	43800	43800	43800
		Gaz in KW	4740	5560	6320	6680	6680	7040	7050	7100	6900	6660	6800	6890	7030	6920	5980	5580	4790	4530	3890	3800	3500	3520	3680	4040
12	Princeton	wind in KW	373	2810	2570	2410	2910	2090	2440	2310	2070	1900	2210	2370	2450	2940	3340	3630	3690	3700	4170	4570	5410	4270	5630	5080
		Gaz in KW	57200	70200	75000	76200	77200	78300	78000	80000	76500	76900	79100	81600	81400	75400	66100	60000	51900	45100	43600	39000	38300	40000	44700	56400
14	Pittsburgh	Wind in KW	1860	1940	2050	1980	1710	1330	1330	1010	1120	1410	1780	1780	1780	2210	2480	3120	3160	4020	4250	4670	5260	5390	4640	4640
		Solar in KW	884	879	901	920	946	908	902	850	0	0	0	0	0	0	0	0	0	0	0	0	1050	1050	1300	1300
13	Cottage	Gaz in KW	33900	33700	32900	31800	32400	32800	33500	34700	33100	32300	31000	31000	25500	21600	21500	21500	22000	23200	25900	25900	26900	26400	26400	26400

TABLE 5. Carbon intensity associated to USA region in Kg/KWh.

Energy type	Carbon Intensity in Kg/KWH
Coal	0,820 KgCo2eq/KWh
Gaz	0,490KgCo2eq/KWh
Solar	0,045KgCo2eq/KWh
Wind	0,011KgCo2eq/KWh
Nuclear	0,012KgCo2eq/KWh

TABLE 6. Simulation parameters.

Parameters Notations	Values
Distance between the adjacent EDFAs [Km]	80
Number of fibers associated to a physical link	1-2
Number of wavelengths in a fiber	[50,80]
Capacity of each wavelength [Gbps]	40
Number of hops between collocated routers and switches	[6,15]
Number of hops between distant routers and switches	[9,15]
Total number of hops	[15,30]
Power consumption of a router port [KW]	1
Power consumption of a transponder [KW]	0,073
Power consumption of an EDFA [KW]	0,008
Power consumption of multiplexer/demultiplexer [KW]	0,016
Power consumption of an optical switch [KW]	0,085

B. RESULTS PERFORMANCE EVALUATION

We first analyze in (Section VI.B.1) how different uses of the five considered energy kinds vary under various time slots and various numbers of VNRs groups. Then, we present the improvement of CER values in the green-aware scenario compared to the delay-sensitive one. In (Section VI.B.2), we demonstrate the scalability of our proposed solution under various numbers of VNRs groups. Then, we prove the improvement of obtained results over other benchmark algorithms, in terms of four evaluation metrics.

TABLE 7. Hosts characteristics and the four considered EC2 VMS instances parameters.

Power of full load server [kW]			0.356	
Power of idle server [kW]			0.112	
MIPS			1000-5000	
Bandwidth			10GB	
RAM/Storage			32GB/1TB	
EC2 VMs	vCPU	RAM	Bandwidth	Storage
Tiny	1	0.5GB	100MBits/s	5GB
Small	1	2GB	100MBits/s	20GB
Medium	2	4GB	100MBits/s	40GB
Large	4	8GB	100MBits/s	80GB

1) ENEGIES' UTILIZATIONS VARIATION RESULTS

In this section, we begin by comparing variations in uses of the five considered energy types, considering the green-aware scenario performed by the (PGTR1) algorithm and the delay-sensitive one insured by the classical minimum hop algorithm (PGTR2). Figure 6 above illustrates these variations under different numbers of requests. (Figure 6, a) proves a 100% use of green energy with a slight use of nuclear energy in the green-aware scenario. This result is prompted by the green resources' ranking process proposed in the PGTR1 algorithm. Assuming the considered DCs nodes use primarily REs once they are available, we notice from (Figure 6.b) the use of a 52.11% share of green energy (including nuclear energy) even for a "delay-sensitive" scenario. As we presume the existence of REs in all considered NSFNET DCs' nodes as mentioned in (Figure 5) and (Table 4), the 47.89% share of grid energy in this later scenario results therefore from a shortfall of REs availabilities. Figure 7 similarly emphasizes the use of these five types of energy during each time slot. When several groups of requests are executed throughout the considered total 24 hours interval in the "delay-sensitive" scenario, we clearly perceive from (Figure 7.b) the use of 52.88% of grid energy over

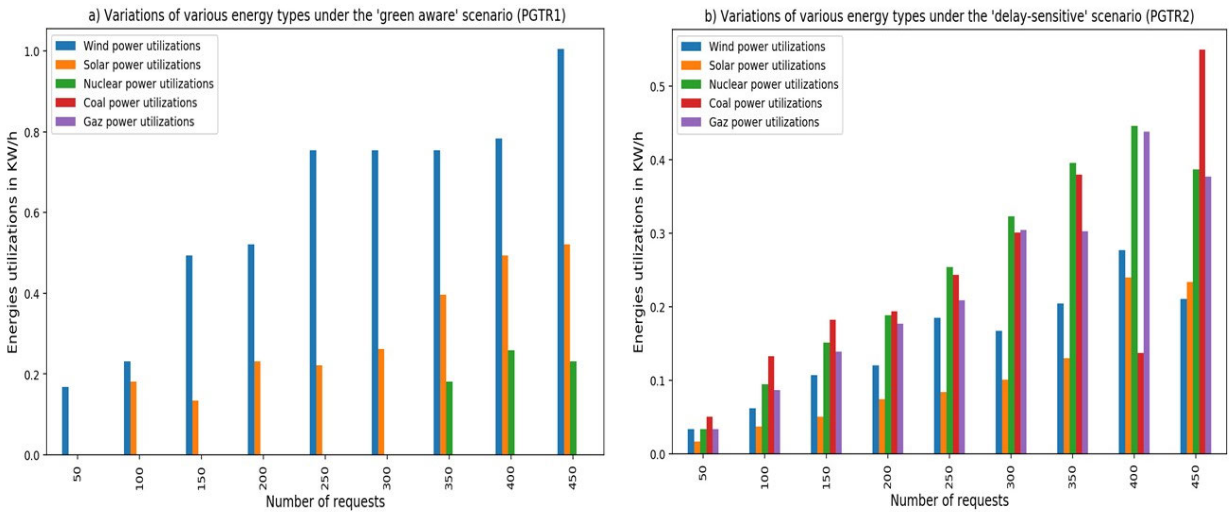


FIGURE 6. Variation of energies utilizations under different requests numbers in both: a) the green-aware scenario of PGTR1 algorithm and b) the delay-sensitive scenario of PGTR2 algorithm.

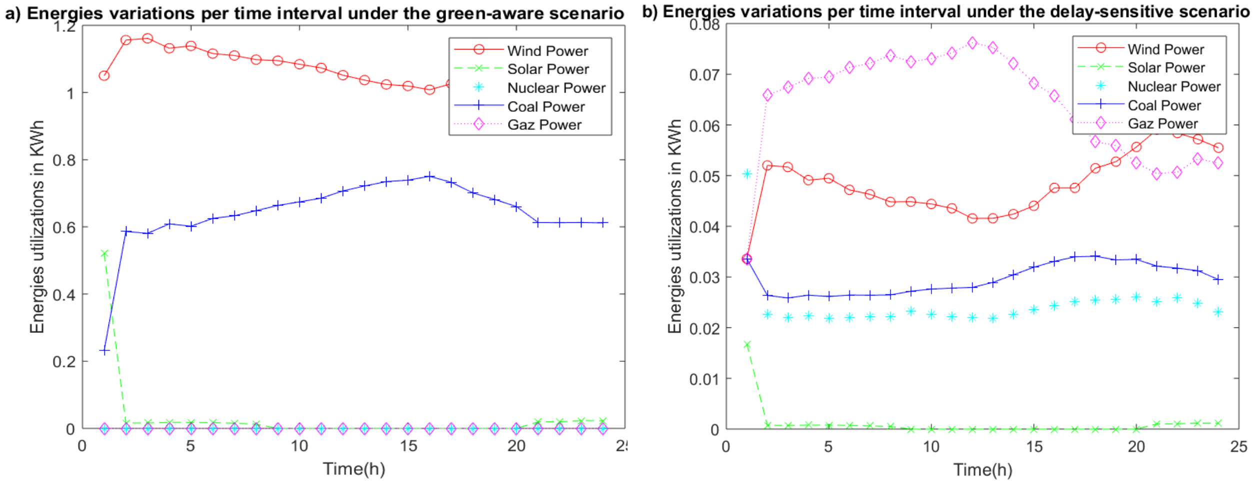


FIGURE 7. Variation of energies utilizations per time slots in both scenarios: a) the green-aware scenario of PGTR1 algorithm and b) delay sensitive scenario of PGTR2 algorithm.

47.12% of green energy (including nuclear energy). Nevertheless, a 100% primordial use of green energies (including nuclear power) results from the proposed green-aware scenario throughout the 24 hours interval, as shown in (Figure 7.a). This last scenario calls initially for green energies (wind, solar and nuclear), according to the increasing order of CER values established by the PGTR1 first sorted Q1 ranking class and according to green energies availabilities (Table 4).

2) COMPARAISON AND ANALYSIS OF PERFORMANCE METRICS RESULTS

To evaluate the performance and scalability of our proposed solution, we measured the following five performance metrics:

– CER: refers to the total Carbon Emission Rate metric computed as the sum of DC's CER value and network's CER value as described respectively in equations (9) and (10) in (Section IV.B.2).

– The total power consumption P_{Total}: refers to the sum of DC's power consumption $P_{DC(n)}$ and network's power consumptions $P_{net(d\rightarrow n)}$, described respectively in equations (11) and (12) in (Section IV.B.2).

– The total power efficiency TP_{eff}: refers to the sum of DC's power efficiency $P_{eff,DC(n)}$ and network's power efficiency $P_{eff,net(d\rightarrow n)}$, described respectively in equations (23) and (24) in (Section IV.C).

– Running time: refers to the average response time spent per the whole parallel algorithms while executing a group of requests.

– Acceptance ratio: refers to the ratio of all executed requests on the total number of received requests.

Starting with CER metric (Figure 8), the sum of CER values obtained in the "green-aware" scenario throughout different group of requests accounts only for 6.3% of the sum

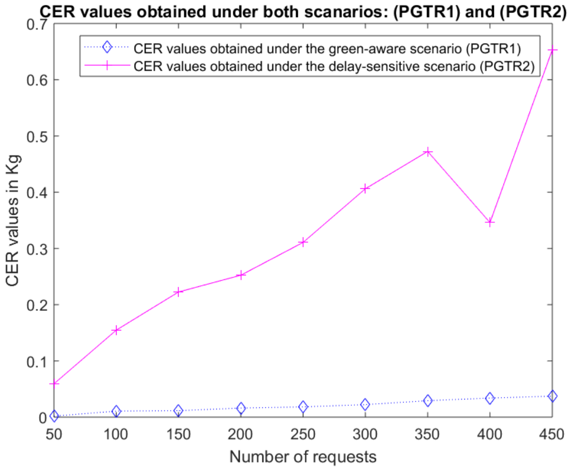


FIGURE 8. **CER values obtained under PGTR1 and PGTR2 algorithms.**

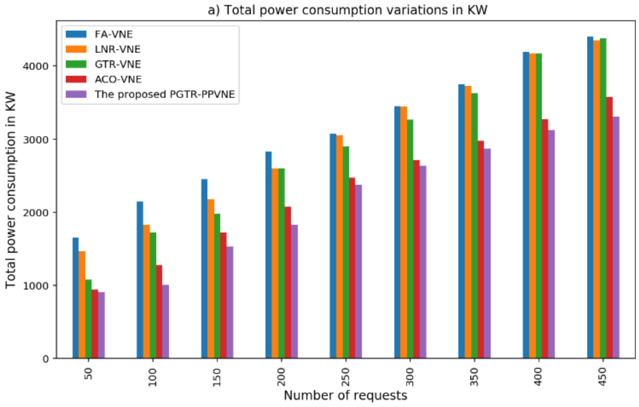


FIGURE 9. **Comparison of total power consumption results in the NSFNET network.**

of CER values obtained in the "Delay-sensitive" one, which is equivalent to a 93% decrease of carbon emissions.

In this sub-section, we evaluate our proposed Parallel Global Topology Ranking – Parallel and Proactive VNE (PGTR-PPVNE) results using the last four metrics. We compared then obtained results with four other benchmark solutions namely: The First Available-VNE (FA-VNE), the Local Nodes Ranking-VNE (LNR-VNE), the Global Topology Ranking-VNE (GTR-VNE) and the And Colony Optimization (ACO-VNE).

Figure 9 illustrates the total energy consumption results obtained with the proposed PGTR-PPVNE algorithm, and those of the other four reference algorithms. Total power consumption results in the proposed PGTR-PPVNE show respectively 6.87%, 23.88%, 26.97% and 29.97% decrease over ACO-VNE, GTR-VNE, LNR-VNE and FA-VNE. Reasons behind those improvements are the adopted resources provisioning, together with the followed consolidation scheme in the proposed PGTR-PPVNE. Therefore, the closest benchmark algorithm with regards to the obtained model's performance is the ACO-VNE. This is mainly due to its proactive probabilistic capabilities for resource provisioning, based on pheromone updates.

In (Figure 10, b), DCs nodes power consumption of the proposed PGTR-PPVNE declined by 24.14 %, 29.83% and 33.95% respectively over GTR-VNE, LNR-VNE and FA-VNE, with a small increase of 0.29% over ACO-VNE. This slight increase was however largely compensated by the decreased power consumption at network level by 13.93% over ACO-VNE, then by 23.58%, 23.38% and 24.75% respectively over GTR-VNE, LNR-VNE and FA-VNE, as shown in (Figure 10, a).

According to (Figure 11, a), the obtained total power efficiency results under the proposed proactive scenario showed respectively 10.47%, 33.06%, 36.43% and 39.42% increase over ACO-VNE, GTR-VNE, LNR-VNE and FA-VNE. Nevertheless, while directly using SP' level instantiated VNRs resources, the obtained total power efficiency under this reactive scenario achieved 13.35%, 16.22% and 18.77% increase respectively over GTR-VNE, LNR-VNE and FA-VNE, as represented in (Figure 11, b). While the same obtained reactive scenario total power efficiency results showed 5.89% and 14.81% decrease respectively comparing to the proactive ACO-VNE and the proposed proactive PGTR-PPVNE.

In (Figure 12, b), DCs' nodes power efficiency of the proposed PGTR-PPVNE increased by 31.87%, 40.74% and 47.25% respectively over GTR-VNE, LNR-VNE and FA-VNE, with a small decrease of 0.4% over ACO-VNE. As highlighted previously, this slight decrease was however largely compensated by the increased power efficiency at network level by 20.04% over ACO-VNE, then by 50.43%, 33.24% and 33.72% respectively over GTR-VNE, LNR-VNE and FA-VNE, as shown in (Figure 12. a).

It is worth noting how response times differ completely from one simulation environment to another. It is obvious from (Figure 13) how the first three reactive algorithms (LA-VNE, LNR-VNE and GTR-VNE) have very low response times compared to both proactive algorithms (ACO-VNE and the proposed PGTR-PPVNE), given their low complexities. Considering the whole simulated algorithms under an NSFNET network in CloudSim simulator, we noticed that requests' response times increased by 2 seconds along an increase in the number of requests' group. This same 2 seconds increase rate was also noticed for the ACO-VNE algorithm. Nevertheless, a 58.54% decrease in requests response time is achieved compared to ACO-VNE response time.

On the first hand, this response time reduction is strongly due to the separated implementation environment of the adopted Bidirectional-Gated Recurrent Unit (B-GRU) prediction model, ensuring the treated proactive intra and inter cloud resources provisioning scenario. Since then, traffics prediction results preliminarily released separately every hour as a traffic seasonality interval, have been used as an input to the proposed PPVNE algorithms. On the second hand, this response time reduction has further been improved

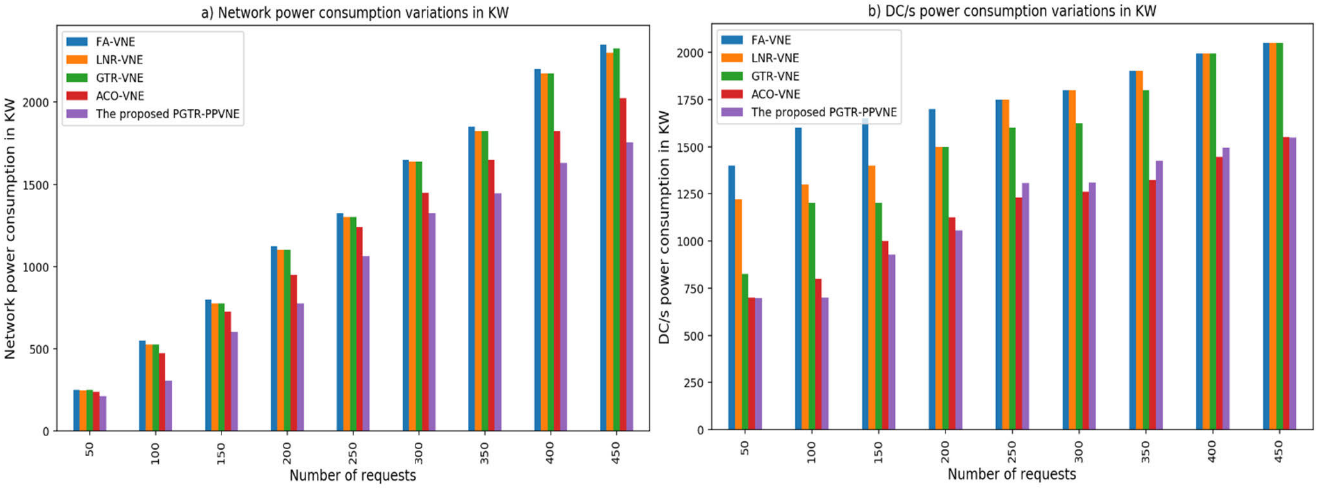


FIGURE 10. Comparison of power consumption results in a) inter cloud optical network and b) intra cloud data centers.

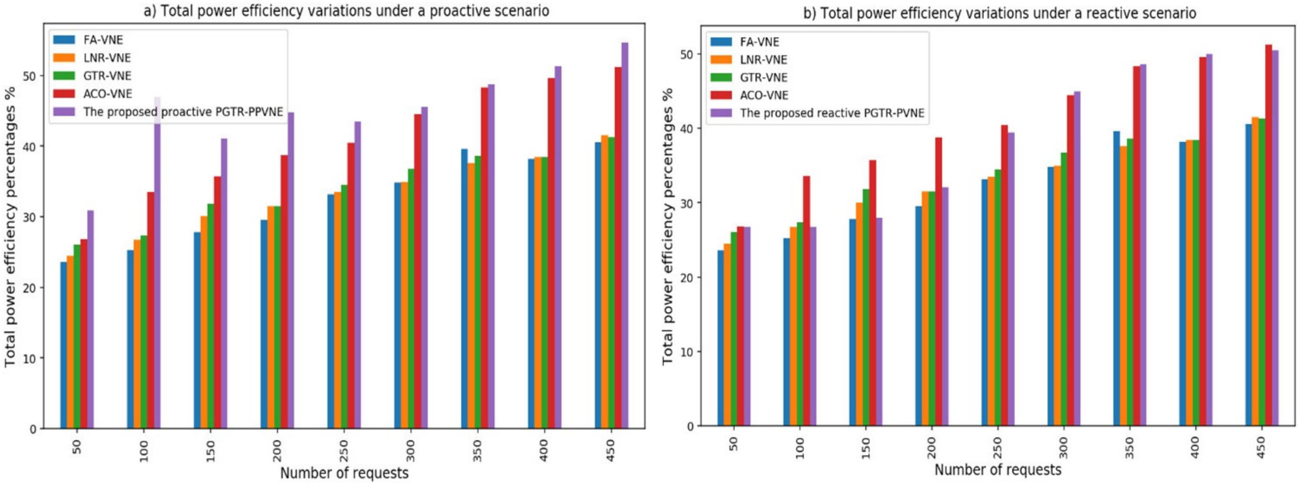


FIGURE 11. Total power efficiencies results obtained under a) the proposed proactive scenario and b) the reactive scenario.

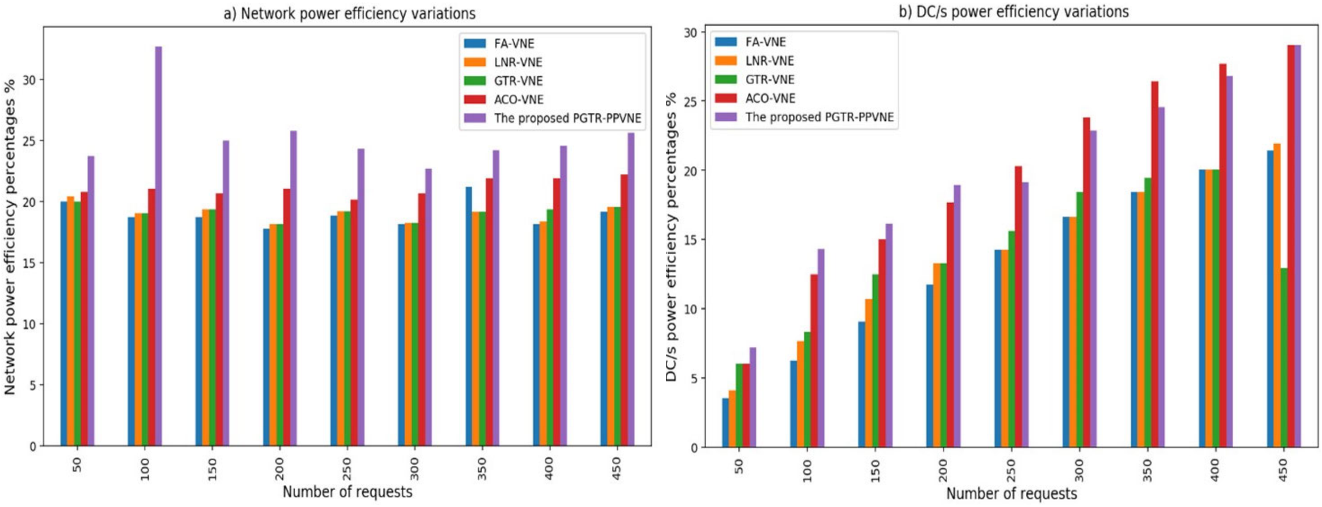


FIGURE 12. Comparison of power efficiency results in a) inter cloud optical network and b) intra cloud data centers.

thanks to the proposed parallel simulated algorithms scheme, executed simultaneously under both: the green-aware scenario and the delay-sensitive one.

The acceptance rate on the other hand completed 100% for requests less than 400, in the case of GTR-VNE, ACO-VNE and the proposed PGTR-PPVNE, as presented in (Figure 14).

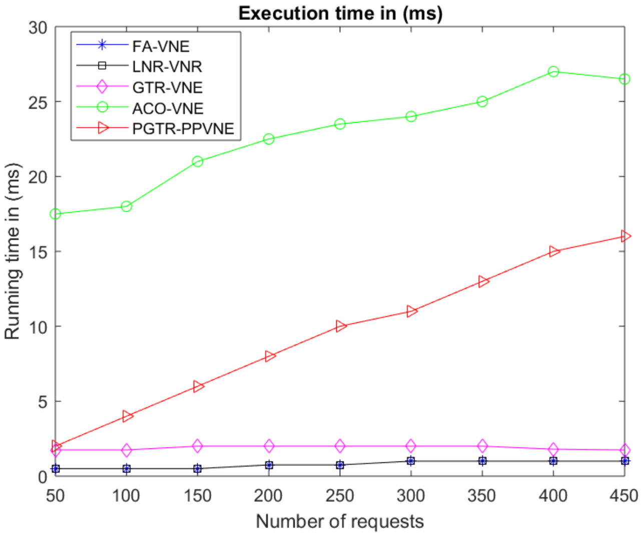


FIGURE 13. Average requests' running time in the simulated NSFNET network.

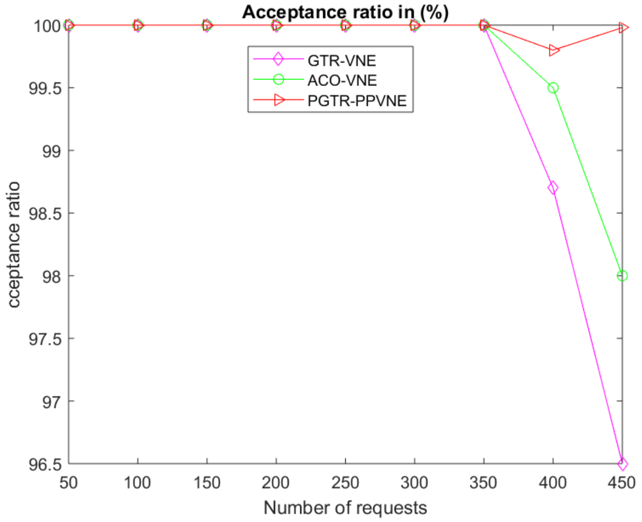


FIGURE 14. Acceptance ratio results in the simulated NSFNET network.

The proposed PGTR-PPVNE achieved until 99,8% and 99,9 % compared to ACO-VNE obtained acceptance ratios of 99,5% and 98%, respectively for requests number above 400 and 450.

VII. CONCLUSION AND PERSPECTIVES

As VNE represents a key problem in network virtualization, in this article we examined this issue under power efficiency and carbon emission reduction perspectives. Specifically, we proposed a green distributed SD-ODCN architecture, in which green DCs' servers and networking nodes are prioritizing first through a new proposed Parallel Global resource Topology Ranking process (PGTR). The proposed green PGTR achieved until 93% decrease of carbon emissions compared to a classical ranking approach. Accordingly, another proposed power efficient, Parallel and Proactive (PPVNE) was addressed. The proposed (PGTR-PPVNE) solved efficiently the formulated MILP problem and outperformed other four benchmarks VNE approaches in terms of four performance metrics. More precisely, the proposed (PGTR-PPVNE) achieved 6.87% decrease, 10.77% increase, 58.54 % decrease and a 1.15% increase over the proactive ACO-VNE benchmark approach, respectively in terms of total power consumption, total power efficiency, requests response times and acceptance ratio. Coming up to future perspectives, we intend first to include REs storage process to study various energy type availabilities in a more realistic scheme. Afterward, we are striving again to address the proposed (PGTR-PPVNE) under others network topologies, and even improved traffic forecasts.

ACKNOWLEDGMENT

This work was supported as part of Green IT projects launched by L3I Laboratory and EIGSI school in La Rochelle (France), to achieve digital sobriety transition. The Moroccan Ministry of Higher Education (CNRST) and FRDISI founded this work.

Nomenclature	
SD-ODCN	Software Defined Optical Data Center Networking
VM/s	Virtual Machine/s
VL/s	Virtual Links
VNR/s	Virtual Network Requests
Sub-VNR/s	Sub-Virtual Network Requests
VNO	Virtual Networks Orchestrator
VIP	Virtual Infrastructure Provider
MN/s	Manager Node/s
CN	Central Node
DC/s	Data Center/s
SDN	Software Defined Network
RE/s	Renewable Energy/ies
QoS	Quality of Service
GTR	Global resources Topology Ranking
CER	Carbon Emission Rate
OEO	Optical Electrical Optical conversion

REFERENCES

[1] M. Otokura, K. Leibnitz, Y. Koizumi, D. Kominami, T. Shimokawa, and M. Murata, "Evolvable virtual network function placement method: Mechanism and performance evaluation," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 1, pp. 27–40, Mar. 2019, doi: 10.1109/TNSM.2018.2890273.

[2] C. Zhang, X. Wang, A. Dong, Y. Zhao, Q. He, and M. Huang, "Energy efficient network service deployment across multiple SDN domains," *Comput. Commun.*, vol. 151, pp. 449–462, Feb. 2020, doi: 10.1016/j.comcom.2020.01.019.

[3] *Cop-La COP21 Sur Le Climat (déFinition, Enjeu, Résumé). L'accord de Paris | Agence Parisienne du Climat*. Accessed: Aug. 6, 2020. [Online]. Available: https://www.apc-paris.com/cop-21

[4] U. Hölzle. (Dec. 2016). *Google Sustainability*. Accessed: Nov. 3, 2020. [Online]. Available: https://sustainability.google/progress/projects/announcement-100/

[5] (2019). *AWS Sustunability*. Accessed: Nov. 3, 2020. [Online]. Available: https://aws.amazon.com/fr/about-aws/sustainability/

[6] IBM Cloud. (Feb. 27, 2015). *IBM and the Environment—News and Recognition—IBM Receives Receives 2018 Climate Leadership Award for GHG Management—Goal Achievement*. Accessed: Nov. 3, 2020. [Online]. Available: https://www.ibm.com/ibm/environment/news/climate_leadership_award_2019.shtml

[7] Microsoft Cloud. (2018). *The Microsoft Cloud Can Save Customers 93 Percent and More in Energy and Carbon Efficiency*. Accessed: Nov. 3, 2020. [Online]. Available: https://www.comparex-group.com/web/microsites/microsoft/news/news/Microsoft_Cloud_saves_customers_93_in_energy_and_carbon_efficiency.htm

[8] D. Kliazovich, P. Bouvry, and S. U. Khan, "GreenCloud: A packet-level simulator of energy-aware cloud computing data centers," *J. Supercomput.*, vol. 62, no. 3, pp. 1263–1283, Dec. 2012, doi: 10.1007/s11227-010-0504-1.

[9] I. Raïs, A.-C. Orgerie, M. Quinson, and L. Lefèvre, "Quantifying the impact of shutdown techniques for energy-efficient data centers," *Concurrency Comput., Pract. Exper.*, vol. 30, no. 17, Sep. 2018, Art. no. e4471, doi: 10.1002/cpe.4471.

[10] A. Carrega, S. Singh, R. Bruschi, and R. Bolla. (Aug. 12, 2012), *Traffic Merging for Energy-Efficient Datacenter Networks—IEEE Conference Publication*. Accessed: Dec. 3, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/6267016

[11] Y. Zong, Y. Ou, A. Hammad, K. Kondepu, R. Nejabati, D. Simeonidou, Y. Liu, and L. Guo, "Location-aware energy efficient virtual network embedding in software-defined optical data center networks," *J. Opt. Commun. Netw.*, vol. 10, no. 7, p. B58, Jul. 2018, doi: 10.1364/JOCN.10.000B58.

[12] I. Hamzaoui, B. Duthil, V. Courboulay, and H. Medromi, "A survey on the current challenges of energy-efficient cloud resources management," *Social Netw. Comput. Sci.*, vol. 1, no. 2, p. 73, Mar. 2020, doi: 10.1007/s42979-020-0078-9.

[13] R. Lent, "Simulating the power consumption of computer networks," in *Proc. 15th IEEE Int. Workshop Comput. Aided Modeling, Anal. Design Commun. Links Netw. (CAMAD)*, Dec. 2010, pp. 96–100, doi: 10.1109/CAMAD.2010.5686955.

[14] S. Su, Z. Zhang, A. X. Liu, X. Cheng, Y. Wang, and X. Zhao, "Energy-aware virtual network embedding," *IEEE/ACM Trans. Netw.*, vol. 22, no. 5, pp. 1607–1620, Oct. 2014, doi: 10.1109/TNET.2013.2286156.

[15] M. O. I. Musa, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Energy efficient core networks using network coding," in *Proc. 17th Int. Conf. Transparent Opt. Netw. (ICTON)*, Jul. 2015, pp. 31–34, doi: 10.1109/ICTON.2015.7193630.

[16] M. O. I. Musa, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Network coding for energy efficiency in bypass IP/WDM networks," in *Proc. Int. Conf. Transparent Opt. Netw.*, Aug. 2016, pp. 1–3, doi: 10.1109/ICTON.2016.7550699.

[17] E. Ghazisaeedi and C. Huang, "EnergyMap: Energy-efficient embedding of MapReduce-based virtual networks and controlling incast queuing delay," in *Proc. 8th IEEE Int. Conf. Commun. Softw. Netw. (ICCSN)*, Jun. 2016, pp. 698–702, doi: 10.1109/ICCSN.2016.7586614.

[18] X. Guan, B.-Y. Choi, and S. Song, "Energy efficient virtual network embedding for green data centers using data center topology and future migration," *Comput. Commun.*, vol. 69, pp. 50–59, Sep. 2015, doi: 10.1016/j.comcom.2015.05.003.

[19] P. Zhang, "Incorporating energy and load balance into virtual network embedding process," *Comput. Commun.*, vol. 129, pp. 80–88, Sep. 2018, doi: 10.1016/j.comcom.2018.07.027.

[20] N. Triki, N. Kara, M. E. Barachi, and S. Hadjres, "A green energy-aware hybrid virtual network embedding approach," *Comput. Netw.*, vol. 91, pp. 712–737, Nov. 2015, doi: 10.1016/j.comnet.2015.08.016.

[21] L. Nonde, T. E. H. Elgorashi, and J. M. H. Elmirgahni, "Virtual network embedding employing renewable energy sources," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6, doi: 10.1109/GLOCOM.2016.7842376.

[22] S. K. Dey and A. Adhya, "Delay-aware green service migration schemes for data center traffic," *J. Opt. Commun. Netw.*, vol. 8, no. 12, pp. 962–976, Dec. 2016, doi: 10.1364/JOCN.8.000962.

[23] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Softw., Pract. Exper.*, vol. 41, no. 1, pp. 23–50, Jan. 2011, doi: 10.1002/spe.995.

[24] GLOBAL IT Communications. *Network—Global IT*. Accessed: Nov. 18, 2020. [Online]. Available: https://globalit.com/about-globalit/network-solutions/#

[25] ElectricityMap. *electricityMap | Emissions CO? de la Consommation électrique en Temps réel*. Accessed: Nov. 19, 2020. [Online]. Available: https://www.electricitymap.org/map

[26] *IEA—International Energy Agency*. Accessed: Nov. 19, 2020. [Online]. Available: https://www.iea.org/

[27] Energy Innovation Policy & Technology LLC. *Home—Energy Innovation: Policy and Technology*. Accessed: Nov. 19, 2020. [Online]. Available: https://energyinnovation.org/

[28] T. M. Nam, N. H. Thanh, H. T. Hieu, N. T. Manh, N. V. Huynh, and H. D. Tuan, "Joint network embedding and server consolidation for energy–efficient dynamic data center virtualization," *Comput. Netw.*, vol. 125, pp. 76–89, Oct. 2017, doi: 10.1016/j.comnet.2017.06.007.

[29] V. Eramo and F. G. Lavacca, "Proposal and investigation of a reconfiguration cost aware policy for resource allocation in multi-provider NFV infrastructures interconnected by elastic optical networks," *J. Lightw. Technol.*, vol. 37, no. 16, pp. 4098–4114, Aug. 15, 2019, doi: 10.1109/JLT.2019.2921428.

[30] C. Gu, L. Fan, W. Wu, H. Huang, and X. Jia, "Greening cloud data centers in an economical way by energy trading with power grid," *Future Gener. Comput. Syst.*, vol. 78, pp. 89–101, Jan. 2018, doi: 10.1016/j.future.2016.12.029.

[31] T. Baker, B. Aldawsari, M. Asim, H. Tawfik, Z. Maamar, and R. Buyya, "Cloud-SEnergy: A bin-packing based multi-cloud service broker for energy efficient composition and execution of data-intensive applications," *Sustain. Comput., Informat. Syst.*, vol. 19, pp. 242–252, Sep. 2018, doi: 10.1016/j.suscom.2018.05.011.

[32] V. Eramo, E. Miucci, and M. Ammar, "Study of migration policies in energy-aware virtual router networks," *IEEE Commun. Lett.*, vol. 18, no. 11, pp. 1919–1922, Nov. 2014, doi: 10.1109/LCOMM.2014.2360190.

[33] M. Zhu, Q. Sun, S. Zhang, P. Gao, B. Chen, and J. Gu, "Energy-aware virtual optical network embedding in sliceable-transponder-enabled elastic optical networks," *IEEE Access*, vol. 7, pp. 41897–41912, 2019, doi: 10.1109/ACCESS.2019.2892993.

[34] M. H. Dahir, H. Alizadeh, and D. Gözüpek, "Energy efficient virtual network embedding for federated software-defined networks," *Int. J. Commun. Syst.*, vol. 32, no. 6, p. e3912, Apr. 2019, doi: 10.1002/dac.3912.

IKHLASSE HAMZAOUI received the Engineering degree in computer science and industrial system management from the ESITH School, Casablanca, Morocco, in 2018, after two years of preparatory classes and three years of engineering school. She is currently pursuing the Ph.D. degree in computer science with the ENSEM School, Hassan II University, Casablanca. Having acquired multidisciplinary skills, her current research interests include green IT, green software engineering, cloud computing, multimodal computer communications, and advanced deep learning.

GUILLAUME BOURGEOIS received the master's degree in computer science with a specialization in computer systems for industrial and service logistics engineering (SIGLIS) from the University of Pau and Pays de l'Adour, France. He is currently pursuing the Ph.D. degree in computer science with the L3I Laboratory, La Rochelle University, France. Having strong inter-disciplinary skills in the fields of computer science and supply chain, data visualization, interactive software programming, artificial intelligence, and big data, his current research interest is to reconceptualize digital services from an eco-design perspective.

BENJAMIN DUTHIL received the Ph.D. degree in computer science from Montpellier 2 University, Montpellier, France, in 2012. He has strong skills and expertise in information extraction, document annotation, text mining, visual saliency approaches, and knowledge representation. He also has extensive practice with fraud detection in administrative documents. He is currently a Teacher-Researcher and a Coordinator of the ''Networks and Information Systems Integration'' Program with the EIGSI School and also a member of the INR Institute, La Rochelle, France. His research interests include different aspects of footprint reduction, including digital environmental footprint, digital social footprint, and digital economic footprint.

VINCENT COURBOULAY received the Ph.D. degree in medical imaging from La Rochelle University, La Rochelle, France, in partnership with General Electric Medical Systems. He continued his research towards the Habilitation to Direct Research (HDR), obtained about ten years later. His research has been oriented towards sustainable IT, otherwise known as computer science and sustainable development. After his involvement in a Green IT club, the blossoming of various projects led him to set up in 2018, the INR Institute of La Rochelle, where he currently holds the position of Scientific Director.

HICHAM MEDROMI received the Ph.D. degree in engineering science from the University of Nice Sophia Antipolis, Nice, France, in 1996. He is responsible for the System Architecture Team, National Higher School of Electricity and Mechanics (ENSEM), Hassan II University, Casablanca, Morocco. Since 2003, he has been a Full Professor of automatic and computer sciences with ENSEM, Hassan II University. His main research interest includes control architectures of mobile systems based on multi agents' systems.