Universitat Politècnica de Catalunya
Optical Communications Group

Inter-Datacenter Connectivity in Flexgrid-based Optical Networks

Adrián Asensio Garcia

A thesis presented in partial fulfillment of the requirements for the degree of

Master of Science

Advisor:

Dr. Luis Velasco Esteban

Co-advisor:

Dr. Jaume Comellas Colomé

June 2014

Agraïments

En primer lloc vull agrair la dedicació i la impecable direcció d'en Luis Velasco, així com l'inestimable suport d'en Jaume Comellas. Gràcies a la seva supervisió com a directors i tutors durant aquests mesos, aquest projecte s'ha pogut dur a terme.

També voldria destacar el suport rebut per part de la resta de membres, tant professors com estudiants, del Grup de Comunicacions Òptiques de la UPC, així com de tots aquells companys que en una o altra mesura han contribuït a superar diferents obstacles durant aquest temps.

A la família i amics, especialment a la Leila i l'Eric, vull agrair-los el recolzament que m'han donat en tot moment.

Abstract

The huge energy consumption of datacenters (DC) requires an elastic resource management, e.g. by turning servers off when they are not used or turning them on to satisfy increments in the demand. Thanks to virtualization, jobs (e.g., web applications) can be encapsulated in virtual machines (VM) mixed with other workloads and consolidate them in the most proper server according to their performance goals. Local resource managers in DCs can migrate VMs from one server to another looking for reducing energy consumption while ensuring the committed quality of experience (QoE).

Additionally, cloud providers can create DC federations based on a geographically distributed infrastructure so they can manage appropriately green energy resources available in each DC, thus reducing energy expenditure. Scheduling algorithms can perform VM migration not only within a single DC but also transferring a huge amount of raw data from one DC to another to minimize operational costs while ensuring the QoE.

Since traffic between DCs is generated by VM migration, the connectivity required between two DCs highly varies along the day, presenting dramatic differences in an hourly time scale. Therefore, using a flexgrid-based optical network to interconnect DCs is an option to be considered since that technology provides fine and multiple granularity. In flexgrid optical networks the available optical spectrum is divided into frequency slices of fixed spectrum width. Optical connections can be allocated into a variable number of these slices, and its capacity can be dynamically managed by allocating or releasing slices provided that the spectrum allocated to an optical connection remain contiguous.

Network providers can facilitate the interconnection among federated DCs by allowing them to request connections' set up on demand with the desired bitrate, while tearing down those connections when they are not needed. With this aim, in the last years, huge standardization work has been done defining control plane architectures and protocols to automate connection provisioning. The Internet Engineering Task Force (IETF) is defining the Application-Based Network Operations (ABNO) architecture, which is based on standard components such as the active stateful Path Computation Element (PCE).

This thesis is devoted to characterize, evaluate and analyze the problem providing optimal VM placement so as to minimize operational costs assuming that those costs are dominated by energy and communication costs. To this aim, analytical models to optimize energy consumption in DC federations are provided. Both cloud and core optical network control architectures are explored and new connectivity models for elastic operations are proposed. Mixed integer linear programming models as well as heuristic algorithms are developed and simulations are carried out. More specifically, the main objective has been attained by developing three goals covering different open issues.

First we propose the Elastic Operations in Federated Datacenters for Performance and Cost Optimization (ELFADO) problem for scheduling workload and orchestrating federated DCs. A distributed and a centralized approach are studied.

Second we propose architectures based on ABNO, using cross-stratum orchestration and carrier SDN, as well as elastic connectivity models supported: the dynamic elastic model and a transfer mode model respectively.

Finally, we consider the centralized ELFADO and both the dynamic elastic and transfer mode connectivity models proposed and evaluate their performance.

Table of contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Cloud computing has transformed the information technology (IT) industry, shaping the way IT hardware is designed and purchased [Ar10]. Datacenters (DCs) contain hardware and software to provide services over the Internet. Because DCs consume huge amounts of energy [US07], energy expenditure becomes a prominent part of the total operational expenditures for their operators. Aiming at reducing energy expenditures, DC operators can use, or even generate themselves, *green* energy coming from solar or wind sources; green energy would replace either partially or totally energy coming from *brown*, polluting, sources. Notwithstanding, one of the principal drawbacks is that green energy is not always available, depending on the hour of the day, weather and season, among others. In contrast, brown energy can be drawn from the grid at any time, although its cost might vary along the day.

Large Internet companies, such as Google, have their own IT infrastructures consisting in a number of large DCs placed in geographically dispersed locations to guarantee the appropriate quality of experience (QoE) to users; DCs are interconnected through a wide area network [XZh13]. Using infrastructure, workloads can be moved among DCs to take advantage of reduced energy cost during off-peak energy periods in some locations while using green energy when it is available in some other locations. Physical machines, i.e. servers, are turned off when they are not used, thus minimizing their energy expenditure. Nonetheless, smaller independently operated infrastructures cannot perform such elastic operations; notwithstanding, they can cooperate by creating DC federations [Go10.1] to increase their revenue from using IT resources that would otherwise be underutilized, and to expand their geographic coverage without building new DCs.

Within a single DC, virtual machines (VM) migration for consolidation and load balancing purposes are commonly automated using scheduled-based algorithms running in the local resource manager. These algorithms target at optimizing some utility function, ensuring quality of experience and service availability [Mi12], [Go10.2]; its outcome is the set of VMs to be activated, stopped or migrated in the local DC. When a DC federation is created, scheduling algorithms need to consider not only local workload and resources but also those in the rest of federated DCs and compute VM migration towards remote DCs, as well as within the local DC. It is worth highlighting that, in a recent global cloud index study [CISCO], Cisco forecasts DC traffic to quadruple over the next years, reaching 554 EB per month by 2016. Two main components of traffic leaving DCs can be distinguished: traffic among DCs (DC2DC) and traffic between DCs and end users (U2DC). The former includes VM migration to manage the cloud elastically, whilst the latter is associated to applications, such as web, e-mail, etc.

Since elastic operations for VM migration require huge bitrate to be available among DCs for some time periods, the inter-DC network can be based on the optical technology and must provide automated interfaces to set up and tear down optical connections with the required bitrate. Thus, network providers can facilitate federated DCs interconnection by allowing them to request connections' set up on demand with the desired bitrate, while tearing down those connections when they are not needed. In the last years, huge standardization work has been done defining control plane architectures and protocols to automate connection provisioning. The Internet Engineering Task Force (IETF) is defining the application-based network operations (ABNO) architecture [Ki13], which is based on standard components such as the active stateful path computation element (PCE) [Cr13].

1.2 Thesis objectives

The main objective of this thesis is the creation of analytical models to optimize energy consumption in DC federations considering not only the inherent energy costs in DCs but also the communication costs when performing VM migration among the corresponding federated DCs. To this aim, both cloud and core optical network control architectures are explored, new connectivity models for elastic operations are proposed and compared with current fixed, static or dynamic, connectivity models. Mixed integer linear programming models as well as heuristic algorithms are developed. In addition simulations are carried out and results analyzed.

1.3 Thesis outline

The rest of this thesis is organized as follows. Chapter 2 introduces basic concepts and terminology that are relevant to the work presented in this thesis. The state of the art is reviewed and conclusions close the chapter.

Chapter 3 is based on our work in [Ve14.2]. We tackle the Elastic Operations in Federated Datacenters for Performance and Cost Optimization (ELFADO) problem. Two approaches for solving it are described: distributed and centralized. Then, the problem is formally stated and mathematical models and heuristic algorithms to solve it for both approaches are presented. Illustrative results are provided. Conclusions close the chapter.

Chapter 4 is based on our works presented in [Ve13] and [Ve14.1]. We describe the current static inter-DC connectivity and propose architectures considering cross-stratum orchestration and ABNO controller in charge of the interconnection network. Two connectivity models (dynamic and dynamic elastic) in a cloud-ready transport network so as to provide bandwidth on demand are then detailed. Illustrative results are provided to compare static, dynamic and dynamic elastic connectivity models. Then, we propose carrier software defined network (SDN) and a network-driven transfer mode for cloud operations. Illustrative results are shown. At the end of the chapter, conlusions are provided.

Chapter 5, based on our study presented in [As14], extends the scenario described in Chapter 3 adding the elastic connectivity models proposed in Chapter 4. It describes the proposed architecture for the federated DC interfacing a carrier SDN controller, which, in turn, interfaces an ABNO controller in charge of the interconnection network. Illustrative results are provided to compare elastic connectivity models.

Finally, Chapter 6 concludes the thesis, summarizes author’s publications and points out future research work.

Chapter 2

Background and state of the art

This chapter introduces basic concepts and terminology that are relevant to the work presented in this thesis. We start introducing elastic optical networks (EON), the flexgrid technology and current control plane architectures. Then we review concepts related to cloud computing such as virtualization, general DC architecture and its power model.

Additionally, to provide a view on the state of the art, we review current approaches for a cloud-ready transport network for inter-DC connections and energy expenditure minimization in DCs. Conclusions that justify this thesis close the chapter.

2.1 Elastic optical networks

On the contrary to wavelength division multiplexing (WDM) fixed grid networks, in which the width of optical channels is fixed and equal, in EON the channel allocated to a lightpath may be expanded or reduced when the required bitrate of a demand increases or decreases.

In this context, adaptive spectrum allocation with known a priori 24-hour traffic patterns has been addressed in [Ve12], [Kl13], [Sh11] and spectrum adaptation under dynamic traffic demands studied in [Ch11], [Ch13]. Concurrently, different policies for elastic spectrum allocation were proposed, including symmetric [Ve12], [Kl13], [Sh11] and asymmetric [Ve12], [Kl13], [Ch13] spectrum expansion or reduction around a reference frequency as well as entire spectrum re-allocation policy [Ve12], [Kl13].

2.1.1 Flexgrid-based elastic optical networks

The advent of the flexible spectrum grid (*flexgrid*) technology [Ge12], [Ji09] brings new opportunities to next-generation transport networks since it allows elastic, adaptive, highly-scalable, and on demand bandwidth provisioning in optical networks. Additionally, key technologies that are paving the way to devise novel EON architectures are: 1) the availability of flexgrid ready bandwidth-variable wavelength selective switches (BV-WSS) to build bandwidth-variable optical cross-connects (BV-OXC), 2) the development of advanced modulation formats and techniques, both single-carrier (such as k-PSK, k-QAM) and multi-carrier (such as O-OFDM), to increase efficiency and being capable of extending the reach of optical signals avoiding expensive electronic regeneration (3R); 3) multi-flow transponders (also known as sliceable bandwidth-variable transponders, SBVTs) that are able to deal with several flows in parallel, thus adding even more flexibility and reducing costs [Ji12]. For more details on EON architectures and proof-of-concept EON experiments we refer to [Ji09], [Ge11] and [Cu12].

2.1.2 Control plane

Core network control plane architectures based on a PCE are commonly used to control data planes based on optical networks. A set of path computation clients (PCC) send requests to the PCE, on top of the control plane. The PCE implements the corresponding label switched path database (LSP-DB), traffic-engineering database (TED) and routing algorithms. It is woth mentioning a particular case of PCE: the active stateful PCE [Cr13]. That PCE includes algorithms to compute the route and spectrum allocation (RSA) of incoming connection requests or to perform elastic operations on currently established connections.

Additionally, a global concurrent optimization (GCO) module [Le09] providing functionalities for obtaining better network-wide solutions by computing paths for a set of queries grouped together may be implemented in the control plane.

Fig. 2-1 illustrates a generic control plane based on PCE controlling an IP/MPLS network and an EON. For illustrative purposes, control plane relevant protocol messages are represented: the path computation element protocol (PCEP) [Va09] and the resource reservation protocol – traffic engineering (RSVP-TE) [Aw01].

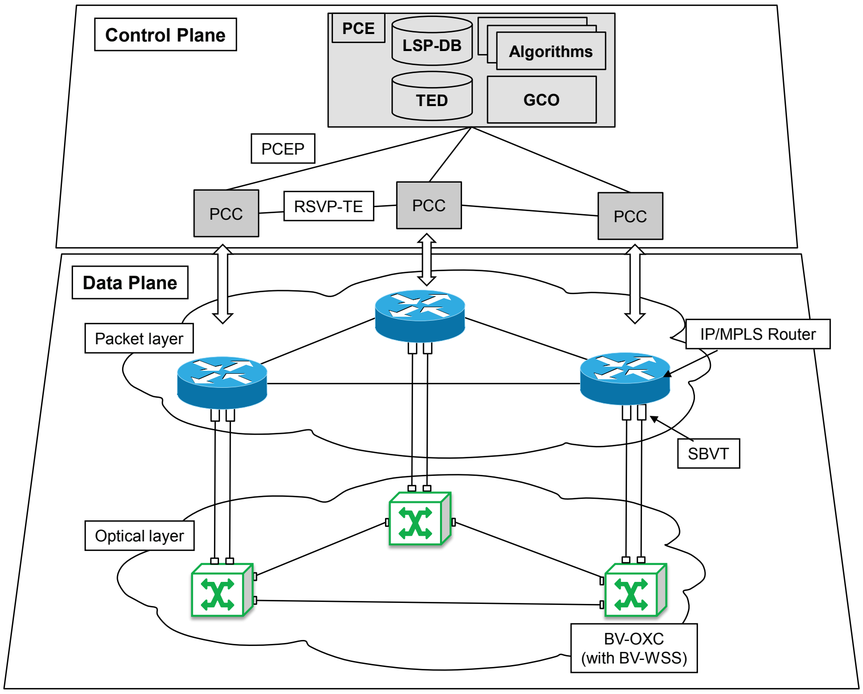

Fig. 2-1. Control and data plane scheme.

Among control plane technologies, the IETF is currently standardizing a centralized architecture named application-based network operations, ABNO, [Ki13]. ABNO is defined as an entity in charge of controlling the network in response to requests from, among others, the application layer. The ABNO architecture consists of a number of functional elements, such as an ABNO controller receiving requests; a policy agent, which enforces the set of policies received from a network management system; an active stateful PCE to perform path computation; a provisioning manager in charge of implementing connections in the network elements; and an operations, administration, and management (OAM) handler that receives notifications. Fig. 2-2 illustrates the ABNO architecture and the elements described above.

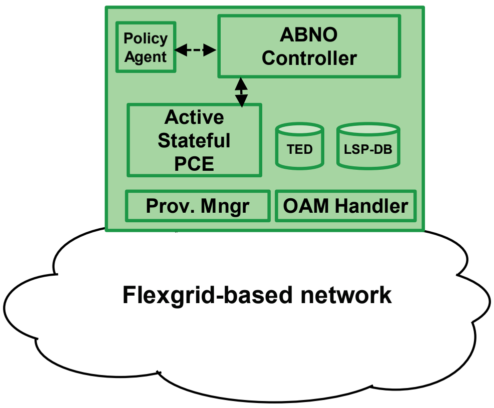

Fig. 2-2. Application-based network operations architecture.

2.2 Cloud computing

In the Internet of services, IT infrastructure providers play a critical role in making the services accessible to end customers. IT infrastructure providers host platforms and services in their DCs. The cloud initiative has been accompanied by the introduction of new computing paradigms, such as infrastructure as a service (IaaS) and software as a service (SaaS), which have dramatically reduced the time and costs required to develop and deploy a service [Ar10]. These paradigms are playing a role of paramount importance in the way companies invest their money regarding IT resources: they are moving from a model where large amounts of capital expenditure (CAPEX) are needed to build their own IT infrastructure and additional cost to maintain it (operational expenditures, OPEX) to a pure OPEX model where IT resources are requested of cloud providers in a pay-as-you-go model.

2.2.1 Virtualization

Dimensioning DCs is a challenging task since workload mixes and intensities are extremely dynamic; dimensioning DCs for peak load can be extremely inefficient, whereas reducing its capacity might result in poor quality of service (QoS), causing service level agreement (SLA) breaches. In addition, the huge energy consumption of DCs requires elastic resource management; for example, turning off physical machines (PM) when they are not used or turning them on to satisfy increments in demand.

Thanks to virtualization, mixed workloads (e.g., web applications and high-performance computing jobs) can easily be consolidated and performance isolated, their consumptions tailored, and placed in the most proper physical machine according to its performance goals. By encapsulating jobs in VMs, a cloud resource manager can migrate jobs from one PM to another looking to reduce energy consumption or some other OPEX objective, while ensuring the committed QoE with the user [Mi12], [Go10.2].

2.2.2 Datacenter architecture

When designing energy-efficient DCs, their internal architecture must be kept in mind. A certain number of switches are needed to provide connectivity between servers in the DC and to interface the DC with the Internet. Consequently, according to the DC architecture being adopted, a corresponding power is consumed, which basically depends on the number and type of switches used. Several intra-DC architectures have been studied in the literature (see [YZh13] for a detailed survey). Among them, the so-called *flattened butterfly* architecture has been identified as the most power-efficient DC architecture, since its power consumption is proportional to the number of currently used servers. However, the most widely-deployed architecture for DC is the so-called *fat-tree* topology [Fa08], which is based on a hierarchical structure where large higher-order switches represent the interface of the DC towards the network infrastructure, and are connected to the servers via a series of lower-order switches, providing the intra-DC connectivity. Fig. 2-3 illustrates an example of a fat-tree topology consisting of three switching layers; from top to bottom: *Core*, *Aggregation* and *Edge*.

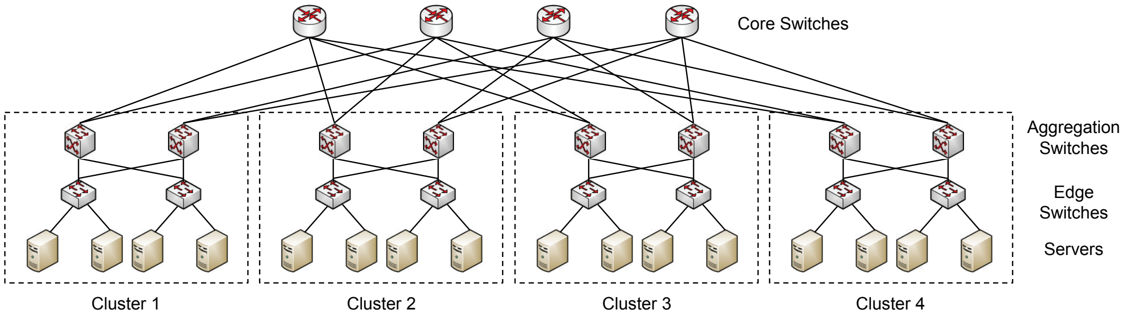

Fig. 2-3. Example of fat-tree datacenter architecture (M=4).

As it is depicted in Fig. 2-3, the lower layers –aggregation and edge– together with the servers are organized in a number of clusters, M. In each of these clusters, switches have M interfaces operating at the same bitrate. Each cluster has $M/2$ edge switches and $M/2$ aggregation switches, all with M ports; it constitutes a *bipartite* graph by connecting each edge to every aggregation switch. In each edge switch, $M/2$ ports are connected directly to servers and the other $M/2$ ports are connected to $M/2$ ports of the aggregation switches. Thus, each cluster has $M^2/4$ servers and there are $M^3/4$ servers in total in the DC. There are $(M/2)^2$ M-port core switches, each having one port connected to each cluster, whilst each cluster is connected to every core switch.

2.2.3 Datacenter power model

Considering a single DC, two main contributions to its power consumption can be distinguished: *i*) the power consumed by IT devices, P_{IT}, which comprises both the servers located in the DC as well as the switches employed to interconnect those servers; *ii*) the power consumption of the non-IT equipment, $P_{non\text{-}IT}$, such as cooling, power supplies and power distribution systems. Thus, total power consumption of a DC can be computed as $P_{DC} = P_{IT} + P_{non\text{-}IT}$. P_{IT} can be easily estimated by counting the number of servers and switches of a DC. However, it is difficult to evaluate the power consumption of non-IT devices since it depends on several details and factors which cannot be easily estimated. For instance, the power consumption of the cooling system strongly depends on the geographical location of the DC and on the building hosting that DC.

An indirect way to estimate a numerical value for $P_{non\text{-}IT}$ is to consider the power usage effectiveness (PUE) metric [GreenGrid]. PUE can be used as a measure of the energy efficiency of a DC and quantifies the amount of power consumed by non-IT equipment in that DC: $PUE = P_{DC} / P_{IT}$. Therefore, if P_{IT} and PUE can be estimated for a given DC, the total power consumed in a DC can be computed as $P_{DC}=PUE*P_{IT}$.

Regarding P_{IT}, we can distinguish between the power consumed by the servers and by network equipment. The power consumed by a server, $P_{server}(k)$, depends mainly on the CPU load (k) utilization, expressed as the ratio between the current load and the maximum capacity of the server. According to [Fa07], the power consumption of a server can be estimated as $P_{server}(k) = P_{server\text{-}idle} + (P_{server\text{-}max} - P_{server\text{-}idle})*k$, where $P_{server\text{-}idle}$ and $P_{server\text{-}max}$ represent the power consumed by the server when it is idle and when it operates at its maximum capacity, respectively. The power consumed by network equipment depends on the specific architecture of the DC.

Considering the fat-tree topology described before and assuming that clusters are active when one or more servers are loaded, otherwise the complete cluster is turned-off, the power consumption of cluster i, $P^i_{cluster}$, can be estimated as,

$$P^i_{cluster} = a^i \cdot \left(\frac{M}{2} \cdot \left(P_{agg} + P_{edge} \right) + \sum_{s=1}^{M^2/4} P_{server}(k^i_s) \right), \tag{2.1}$$

where a^i indicates whether the cluster is active or not and P_{agg} and P_{edge} denote the power consumption of aggregation and edge switches. According to (2.1), then the power consumption of the IT devices in the DC can eventually be computed as follows, where P_{core} denote the power consumption of core switches,

$$P_{IT} = \frac{M^2}{4} \cdot P_{core} + \sum_{i=1}^{M} P^i_{cluster}. \tag{2.2}$$

2.3 Cloud-ready transport network

Transport networks are currently configured with big static fat pipes based on capacity overprovisioning. The rationality behind that is guaranteeing traffic demand and QoS. The capacity of each inter-DC optical connection is dimensioned in advance based on some volume of foreseen data to transfer. Once in operation, scheduling algorithms inside cloud management run periodically trying to optimize some cost function, such as energy costs, and organize VM migration and database (DB) synchronization as a function of the bitrate available.

To avoid transference overlapping (i.e. some migration or DB synchronization not performed in the current period), which may eventually lead to performance degradation, some overdimensioning is needed. Obviously, this static connectivity configuration adds high costs since large connectivity capacity remains unused in periods where volume to transfer is low. Thus, demands of cloud services require new mechanisms to provide reconfiguration and adaptability of the transport network to reduce the amount of overprovisioned bandwidth; the efficient integration of cloud-based services among distributed DCs, including the interconnecting network, then becomes a challenge.

The cloud-ready transport network was introduced in [Co12] to handle this dynamic cloud and network interaction, allowing on demand connectivity provisioning. The considered reference architecture to support cloud-ready transport networks is shown in Fig. 2-4. A cloud-ready transport network is used to interconnect DCs placed in different locations and to provide bandwidth on demand. To support these huge adaptive connections, flexgrid is the best positioned technology, since it can create optical connections using the required spectral bandwidth based on users' requirements. Furthermore, by deploying flexgrid networks in the core, network providers can improve spectrum utilization, thus achieving a cost-effective solution to support their services.

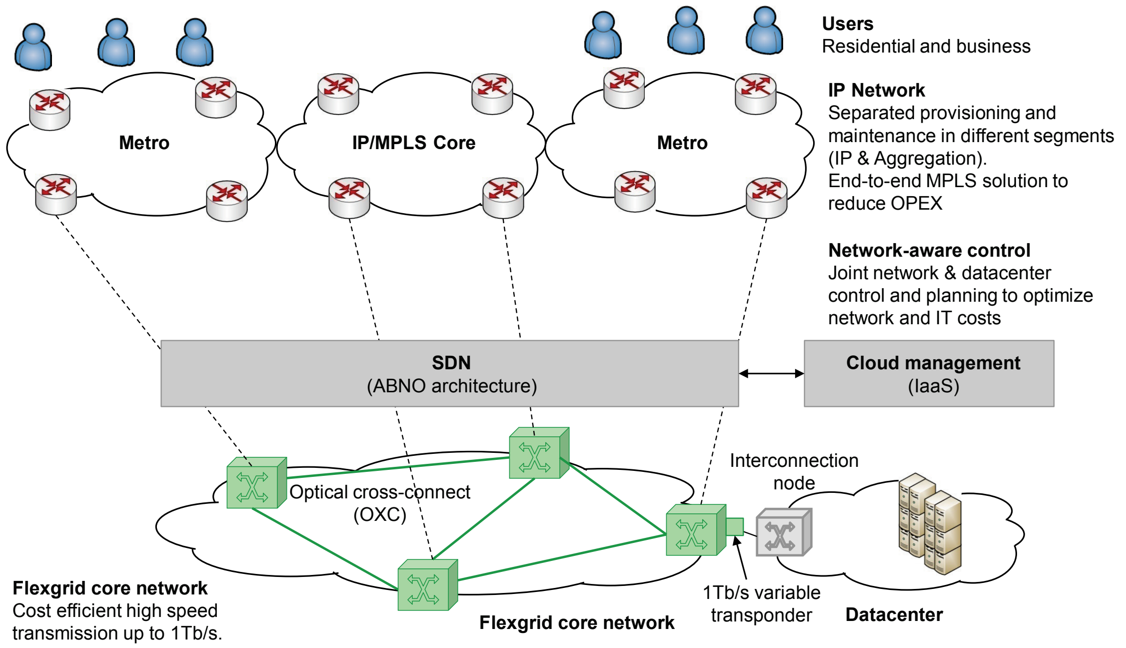

Fig. 2-4. Architecture to support cloud-ready transport networks.

2.4 Energy expenditures minimization

Since minimizing energy expenditures is really important for DC operators, many papers can be found in the literature partially addressing that problem [Go12], [LLi09], [ZLi11] and [Pi11]. In [Go12], the authors propose scheduling workload in a DC coinciding with the availability of green energy, consolidating all the jobs on time slots with solar energy available, increasing green energy consumption up to 31%. Authors in [LLi09] present a DC architecture to reduce power consumption, while guarantee QoE. They consider online-monitoring and VM placement optimization achieving energy savings up to 27%. Other works, e.g. [ZLi11], refer to the problem of load balance DC workloads geographically, following green energy availability, to reduce the amount of brown energy consumed focusing mainly on wind energy and the capability to store energy. Other works focus on the importance of counting as “energy expenditure” every element in the DC, not only computing machinery. The author in [Pi11] remarks the idea that all IT equipment counts when consuming energy, also the fluctuation of green energy production and energy transportation are important factors.

As elastic operations for VM migration require huge bitrate to be available among DCs for some time periods, the inter-DC network can be based on the optical technology and must provide automated interfaces to set-up and tear down optical connections with the required bitrate. Some works consider optical networks to interconnect DCs. For instance, the authors in [Bu13] present routing algorithms considering both routing and scheduling and compare energy savings with respect to a scenario where routing and scheduling problems are solved separately. In addition, some works using flexgrid networks to interconnect DCs are currently appearing in the literature. Authors in [JZh13] propose an application controller that interfaces an OpenFlow controller for the flexgrid network, similarly to the approach followed by Google [XZh13]. Notwithstanding, some network operators are supporting ABNO in the IETF, so there is a lack of consensus on the architecture.

2.5 Conclusions

In the view of the above state of the art, to the best of our knowledge, no work compares the way to compute scheduling considering both energy and communications costs in a single framework. In addition, we focus on solar energy, which is more predictable, and take more advantage of our network capabilities to migrate workload. Besides, in this work we assume IETF’s architecture, supported by major European network operators within the IDEALIST project [Ve14.3], [Ve14.4]. All the above is considered in the Elastic Operations in Federated Datacenters for Performance and Cost Optimization, ELFADO, problem.

Chapter 3

Cloud computing and networking

The huge energy consumption of DCs providing cloud services over the Internet has motivated different studies regarding cost savings in DCs. Since energy expenditure is a predominant part of the total operational expenditures for DC operators, energy aware policies for minimizing DCs' energy consumption try to minimize energy costs while guaranteeing a certain QoE. Federated DCs can take advantage of its geographically distributed infrastructure by managing appropriately the green energy resources available in each DC at a given time, in combination with workload consolidation and VM migration policies. In this scenario, inter-DC networks play an important role and communication costs must be considered when minimizing operational expenditures. In this chapter we tackle the Elastic Operations in Federated Datacenters for Performance and Cost Optimization, ELFADO, problem for scheduling workload and orchestrating federated DCs. Two approaches, distributed and centralized, are studied and mixed integer linear programming (MILP) formulations and heuristics are provided. Using those heuristics, cost savings are analyzed with respect to a fixed workload placement. Simulation experiments have been carried out considering realistic scenarios.

3.1 Orchestrating federated datacenters

In this section the main objective of elastic operations, i.e. minimizing operational costs by taking advantage from available green energy and cheap brown energy, and a distributed and a centralized approaches for orchestrating federated DCs are described.

3.1.1 Minimizing energy expenditures

A first optimization to reduce energy expenditures is to perform consolidation, placing VMs so as to load servers as much as possible and switching off those servers that become unused. To further reduce energy consumption, consolidation can be performed by taking into account clusters structure, and switching on/off clusters as single units. Those servers in switched on clusters without assigned load remain active and ready to accommodate spikes in demand. In addition, as stated in the introduction, DC federations can perform elastic operations, migrating VMs among DCs aiming at minimizing operational costs by taking advantage from available green energy in some DCs and off-peak cheap brown energy in other while ensuring the desired QoE level, e.g latency experienced by the users of a service is used as a QoE mesure. We face then, the ELFADO problem, which orchestrates federated DCs providing optimal VM placement so as to minimize operational costs. We assume that operational costs are dominated by energy and communication costs, so we focus on specifically minimizing those costs.

Two approaches can be devised to orchestrate federated DCs: *i*) *distributed* (Fig. 3-1), where scheduling algorithms running inside DC resource managers compute periodically the optimal placement for the VMs currently placed in the local DC; *ii*) *centralized* (Fig. 3-2), where a *federation orchestrator* computes periodically the global optimal placement for all the VMs in the federated DCs and communicates said computation to each DC resource manager. In both approaches, local resource managers interface the rest of DCs to coordinate VM migration and the control plane controlling the interconnection network to request optical datacenter-to-datacenter connections' set up and tear down.

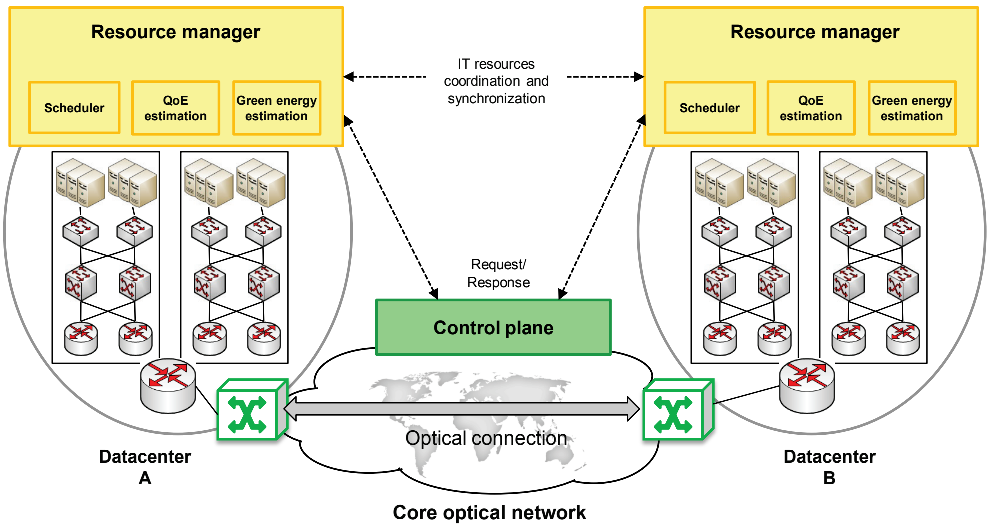

Fig. 3-1. Distributed federated datacenters orchestration.

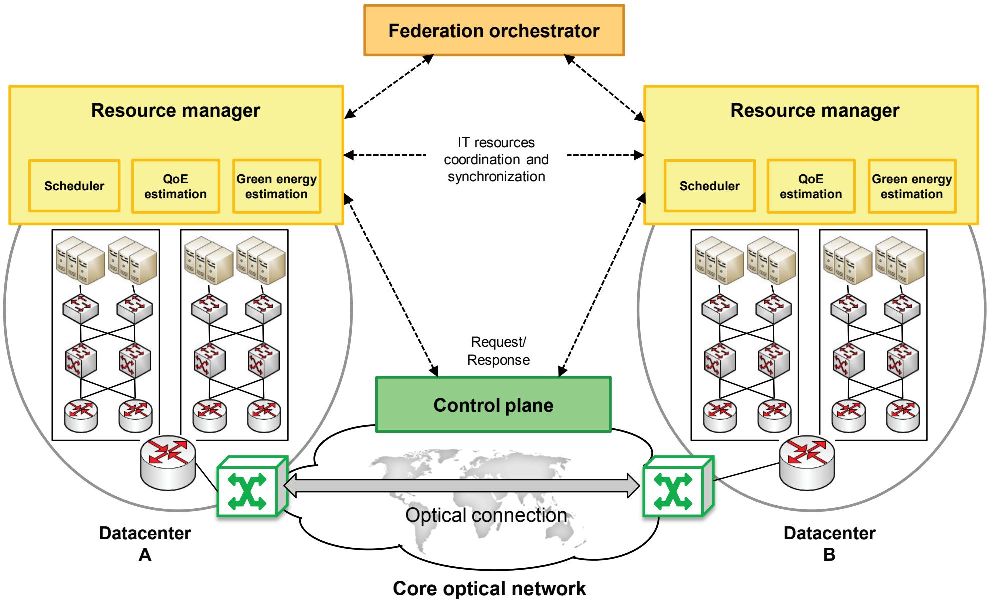

Fig. 3-2. Centralized federated datacenters orchestration.

To solve the ELFADO problem some data must be available, such as an estimation of QoE perceived by the users, the amount of green energy available in each DC, the cost of brown energy, among others. QoE can be estimated by a specialized module inside each resource manager [Verizon]. The cost of brown energy comes from the contract each DC has with the local power supply company, which varies with the time of day. Finally, the amount of green energy that will be likely available in the next period can be predicted using historical data and weather forecast [Sh10]. Each local resource manager can flood all that data to the rest of resource managers in remote DCs.

For illustrative purposes, Fig. 3-3 plots unit brown energy cost, c_d, and normalized availability of green energy, δ_d, for DC d as a function of the time of day. Brown energy cost varies with the time showing on-peak and off-peak periods, where energy during on-peak is approximately 40% more expensive than during off-peak periods. Regarding green energy availability, large variations during the day can be expected. In the view of Fig. 3-3 it is clear that some advantage can be taken from orchestrating the federated DCs, moving VMs to place them in the most advantageous DC.

Let us assume that DCs are dimensioned to cover some proportion β_d of the total energy consumption for the maximum dimensioning. Then, green coverage in DC d, α_d, can be estimated as, $\alpha_d(t) = \beta_d * \delta_d(t)$, and the amount of green energy available can be estimated as $g_d(t) = \alpha_d(t) * Energy\_MaxDimensioning$, where *Energy_MaxDimensioning* represents the amount of energy consumed for the maximum dimensioning.

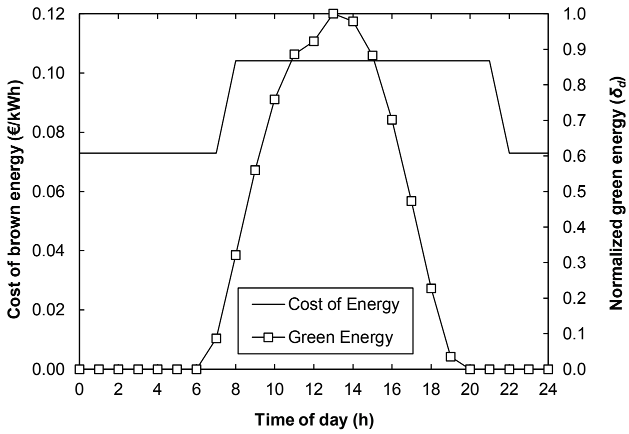

Fig. 3-3. Unit cost of brown energy and normalized availability of green energy against the time of day.

In the distributed approach, local DCs do not know the amount of VMs that will be placed in each DC in the next period, since that decision is to be taken by each DC resource manager in the current period. Therefore, the amount of VMs that can take advantage from green energy availability in each DC in the next period cannot be computed. To overcome that problem, estimation on the unitary energy cost in each DC should be made. We use eq. (3.1),

$$\hat{c}_d = (1 - \alpha_d) \cdot c_d \, , \tag{3.1}$$

i.e. the cost of the energy in each DC is estimated by decrementing the cost of brown energy with the expected green coverage value. As an example, the estimated cost of the energy is 0.0729 €/kWh at 2am and 0 €/kWh at 1pm (assuming ßd=1).

In general, however, green energy covers only partially, even in the generation peak, total energy consumption, thus ßd<1. Therefore, if several DCs take the decision of migrating local VMs to one remote DC in the hope of reducing costs, it may happen that some brown energy need to be drawn from the grid if not enough green energy is available, which may result in higher energy cost in addition to some communication cost.

In contrast, the amount of VMs to be placed in each DC in the next period is known in the centralized approach since the placing decision is taken in the centralized federation orchestrator. Therefore, one can expect that better VM placements can be done in the centralized approach, which might result into further cost savings.

3.2 The ELFADO problem

In this section, the ELFADO problem is formally stated and the corresponding MILP models and heuristic algorithms for solving efficiently both distributed and centralized approaches are presented.

3.2.1 Problem statement

The ELFADO problem can be formally stated as follows:

Given:

- a set of federated datacenters D.
- the set of optical connections E that can be established between two datacenters,
- a set of VMs $V(d)$ in each datacenter d,
- a set of client locations L, where n_l is the number of users in location l to be served in the next period,
- PUE_d, brown energy cost c_d, and green coverage level a_d in datacenter d for the next period,
- the data volume k_v and the number of cores $cores_v$ of each VM v,
- energy consumption of each server as a function of the load k, $w_{server}(k) = P_{server}(k)$*1h,
- the performance p_{ld} perceived in location l when served from a virtual machine placed in datacenter d,
- a threshold th_v for the performance required at any time for accessing the service in virtual machine v.

Output: the datacenter where each VM will be placed the next time period.

Objective: minimize energy and communications cost for the next time period ensuring the performance objective for each service.

3.2.2 Mathematical formulations

As previously stated, the ELFADO problem can be solved assuming either a distributed or a centralized approach. The following sets and parameters have been defined for both approaches:

Sets:

D	set of federated datacenters, index d.
E	set of optical connections that can be established, index e.
$E(d_1)$	set of optical connections between d_1 and any other datacenter.
V	set of virtual machines, index v.
$V(d_1)$	set of virtual machines in datacenter d_1.
L	set of client locations, index l.

Users and performance:

p_{ld}	performance perceived in location l when accessing datacenter d.
n_l	number of users in location l.
th_v	the threshold performance to be guaranteed for v.

Datacenter architecture and VMs:

M	maximum number of clusters per datacenter.
n_{server}	number of cores per server.
k_v	size in bytes of VM v.
n_v	number of cores needed by VM v.

Energy:

a_d	green energy cover in datacenter d.
g_d	amount of green energy available in datacenter d.
PUE_d	PUE for datacenter d.
c_d	brown energy cost per kWh in datacenter d.
w_v	energy consumption of VM v. It can be computed assuming that the server where it is placed is fully loaded, so $w_v = w_{server\_max}/n_v$.

Connections:

k_e	maximum amount of bytes to transfer without exceeding the maximum capacity assigned in connection e. k_e includes the needed overhead from TCP/IP downwards to the optical domain.
c_e	cost per Gb transmitted through connection e.

Additionally, the decision variables are:

x_{vd} binary, 1 if virtual machine v is placed in datacenter d, 0 otherwise.

y_d real positive, energy consumption in datacenter d.

z_e integer positive, bytes to transfer through optical connection e.

The MILP formulation for the ELFADO problem assuming the distributed approach is as follows. It is worth highlighting that this problem is solved by each of the DCs separately; in the model, d_1 identifies the local DC.

$$(Distributed\ ELFADO) \quad \text{minimize} \quad \sum_{d \in D} (1-\alpha_d) \cdot c_d \cdot y_d + \sum_{e \in E(d_1)} 8 \cdot c_e \cdot z_e \tag{3.2}$$

subject to:

$$\frac{1}{\sum_{l \in L} n_l} \cdot \sum_{l \in L} \sum_{d \in D} n_l \cdot p_{ld} \cdot x_{vd} \leq th_v \quad \forall v \in V(d_1) \tag{3.3}$$

$$\sum_{d \in D} x_{vd} = 1 \quad \forall v \in V(d_1) \tag{3.4}$$

$$y_d = PUE_d \cdot \sum_{v \in V(d_1)} w_v \cdot x_{vd} \quad \forall d \in D \tag{3.5}$$

$$z_{e=(d_1,d_2)} = \sum_{v \in V(d_1)} k_v \cdot x_{vd_2} \quad \forall d_2 \in D \setminus \{d_1\} \tag{3.6}$$

$$z_e \leq k_e \quad \forall e \in E(d_1) \tag{3.7}$$

The objective function in equation (3.2) minimizes the total cost for the VMs in a given DC d_1, which consists on the estimated energy costs plus the communication costs for the VMs that are moved to remote DCs.

Constraint (3.3) guarantees that each VM is assigned to a DC if the on-average performance perceived by the users is above the given threshold. Constraint (3.4) ensures that each VM is assigned to one DC. Constraint (3.5) computes the energy consumption in each DC as a result of moving VM from the local DC. Constraint (3.6) computes the amount of data to be transferred from the local to each remote DC. Finally, constraint (3.7) assures that the capacity of each optical connection from the local DC is not exceeded.

The MILP formulation for the centralized one is presented next. Although the model is similar to the distributed approach, this problem computes a global solution for all the DCs and as a result, the total amount of VMs that will be placed in the next period in each DC can be computed. Therefore, the centralized ELFADO computes the cost of the energy in each DC given the amount of green energy available.

Two additional decision variables are defined:

γ_d positive integer with the number of servers operating with some load in datacenter *d*.

ρ_d positive integer with the number of clusters switched on in datacenter *d*.

$$(Centralized\ ELFADO)\quad \text{minimize}\quad \sum_{d\in D} c_d \cdot y_d + \sum_{e\in E} 8 \cdot c_e \cdot z_e \tag{3.8}$$

subject to:

$$\frac{1}{\sum_{l\in L} n_l} \cdot \sum_{l\in L}\sum_{d\in D} n_l \cdot p_{ld} \cdot x_{vd} \le th_v \quad \forall v \in V \tag{3.9}$$

$$\sum_{d\in D} x_{vd} = 1 \quad \forall v \in V \tag{3.10}$$

$$\gamma_d \ge \frac{1}{n_{server}} \cdot \sum_{v\in V} n_v \cdot x_{vd} \quad \forall d \in D \tag{3.11}$$

$$\rho_d \ge \frac{4}{M^2} \cdot \gamma_d \quad \forall d \in D \tag{3.12}$$

$$y_d \ge PUE_d \cdot \left(\begin{array}{l} \frac{M^2}{4} \cdot w_{core} + \frac{M}{2} \cdot \left(w_{agg} + w_{edge}\right) \cdot \rho_d \\ + w_{server-\max} \cdot \gamma_d + w_{server-idle} \cdot \left(\frac{M^2}{4} \cdot \rho_d - \gamma_d \right) \end{array} \right) - g_d \quad \forall d \in D \tag{3.13}$$

$$z_{e=(d_1,d_2)} = \sum_{v\in V(d_1)} k_v \cdot x_{vd2} \quad \forall d_1, d_2 \in D, d_1 \ne d_2 \tag{3.14}$$

$$z_e \le k_e \quad \forall e \in E \tag{3.15}$$

The objective function (3.8) minimizes the total cost for all DCs in the federation, which consists on the energy costs plus the communication costs for the VMs that are moved between DCs.

Constraint (3.9) guarantees that each VM is assigned to a DC if the on-average performance perceived by the users is above the given threshold. Constraint (3.10) ensures that each VM is assigned to one DC. Constraint (3.11) computes, for each DC, the amount of servers where some VM is to be placed, whereas constraint (3.12) computes the number of clusters that will be switched on. Constraint (3.13) computes the brown energy consumption in each DC as the difference between the effective energy consumption, computed as eq. (2.1)-(2.2), and the amount of green energy available in the next period in each DC. Note that $w_{(\cdot)}=P_{(\cdot)}$*1h. Constraint (3.14) computes the amount of data to be transfer from each DC to some other remote DC. Finally, constraint (3.15) assures that the capacity of each optical connection is not exceeded.

The ELFADO problem is *NP-hard* since it is based upon the on the well-known capacitated plant location problem which has been proved to be *NP-hard* [Di02]. 0Regarding problem sizes, the number of variables and constraints for each approach are detailed in Table 3-1. Additionally, an estimation of problems' size is calculated for the scenario presented in Section 4.

Table 3-1. Size of the ELFADO problem.

	Constraints	**Variables**
Distributed	$O(\lvert V\rvert+\lvert D\rvert)$ (10^4)	$O(\lvert V\rvert\cdot\lvert D\rvert)$ (10^5)
Centralized	$O(\lvert V\rvert+\lvert D\rvert^2)$ (10^5)	$O(\lvert V\rvert\cdot\lvert D\rvert)$ (10^5)

Although the size of both MILP models is limited, they must be solved in real time (in the order of few seconds). In the experiments described later in this chapter, we used a commercial solver such as CPLEX [CPLEX] to solve each approach. The distributed approach took tens of minutes on average to be solved; more than 1 hour in the worst case, whereas the centralized approach took more than one hour on average. As a consequence, in the next section we propose heuristic algorithms that provide much better trade-off between optimality and complexity to produce solutions in practical computation times, short enough to be used for schedule real federated DCs.

3.2.3 Heuristic algorithms

The heuristic algorithm for the distributed approach (Table 3-2) schedules the set of VMs in the local DC. For each VM, all feasible, in terms of performance (p_{vd}), placements are found and the cost for that placement is computed (lines 2-9). If the placement is in the local DC, only energy costs are considered, whereas if it is in a remote DC communication costs are also included. Note that energy costs are estimated using the green energy cover to decrement the cost of the energy in the considered DC. The list of feasible placements is ordered as a function of the cost (line 10). Each VM is placed afterwards in the cheapest DC provided that the amount of data to be transferred through the optical connection does not exceed the maximum available, in case of a remote placement (lines 11-17). The final solution is eventually returned (line 18).

Table 3-2. Heuristic for the distributed ELFADO.

```
INPUT d1, V(d1), D
OUTPUT Sol
Sol ← Ø
for each v∈V(d1) do
  for each d∈D do
    if pvd ≤ thv then
      if d ≠ d1 then
        let e=(d1, d)
        C[v].list ← {d, e, (1-αd)*cd*wv + ce*kv}
      else
        C[v].list ← {d, Ø, (1-αd)*cd*wv}
  sort (C[v].list, Ascending)
for each v∈V(d1) do
  for i=1..C[v].list.length do
    {d, e} ← C[v].list(i)
    if e≠Ø && ze+kv > ke then continue
    if e≠Ø then ze ← ze + kv
    Sol ← Sol ∪ {(v, d)}
    break
return Sol
```

The heuristic algorithm for the centralized approach (Table 3-3) schedules the set of VMs in all the federated DCs. The proposed heuristic focuses on taking advantage from all the available green energy, only considering the cost of brown energy and communications when no more green energy is available. The perceived performance of each VM in its current placement is computed; those infeasible placements (the perceived performance is under the threshold) are added to set U whereas those which are feasible to the set F (lines 2-7). Next, the remaining green energy in each DC is computed, considering the available green energy and the energy consumption of those feasible placements (line 8). The set R stores those DCs with remaining green energy available.

The remaining green energy in the DCs (if any) is used to place infeasible placements in set U; the cheapest feasible placement if found for each VM in U provided that the energy consumption of that VM can take advantage from remaining green energy (lines 12-15). If a feasible placement is finally found, the remaining green energy for the selected DC is updated (line 16) and if no green energy remains available, that DC is eventually removed from set R. The same process of maximizing available green energy is performed for the feasible placements in set F (lines 19-25).

Every remaining not yet considered, feasible or unfeasible, placement is stored in the set F to be jointly considered (line 26) and an algorithm similar to the one for the distributed approach is then followed (lines 27-42). The only difference is that the cost of new placements is computed considering that all the energy will come from brown sources (lines 32 and 34). Finally, the solution for all the DCs is returned.

Table 3-3. Heuristic for the centralized ELFADO.

```
INPUT V, D
OUTPUT Sol
Initialize Sol ← Ø; U ← Ø; F ← Ø; R ← Ø
for each d∈D do
  U_d ← Ø; F_d ← Ø
  for each v∈V(d) do
    if p_vd > th_v then
      U_d ← U_d ∪ {(v, d)}
    else F_d ← F_d ∪ {(v, d)}
  r_d ← g_d computeEnergy (F_d)
  U ← U ∪ U_d; F ← F ∪ F_d
  if r_d < 0 then
    R ← {(d, r_d)}
if R≠Ø then
  for each (v, d_1)∈U do
    find (d_2, r_d2)∈R feasible for v such that r_d > w_v with min
    comm cost
    Sol ← Sol ∪ {(v, d_2)}
    r_d2 ← r_d2 - PUE_d2*w_v
    if r_d2 <=0 then
      R ← R \ {(d_2, r_d2)}
if R≠Ø then
  for each (v, d_1)∈F do
    find (d_2, r_d2)∈R feasible for v such that r_d > w_v with min
    comm cost
    Sol ← Sol ∪ {(v, d_2)}
    r_d2 ← r_d2 - PUE_d2*w_v
    if r_d2 <=0 then
      R ← R \ {(d_2, r_d2)}
F ← F ∪ U
for each {v, d_1}∈F do
  for each d_2∈D do
    if p_vd2 ≤ th_v then
      if d_2 ≠ d_1 then
        let e=(d_1, d_2)
        C[v].list ← (d_2, e, c_d2*w_v + c_e*k_v )
      else
        C[v].list ← (d_2, e, c_d2*w_v)
  sort (C[v].list, Ascending)
for each (v, d_1)∈F do
  for i=1..C[v].list.length do
    (d_2, e) ← C[v].list(i)
    if e≠Ø && z_e+k_v > k_e then continue
    if e≠Ø then z_e ← z_e + k_v
    Sol ← Sol ∪ {(v, d_2)}
    break
return Sol
```

The performance of each of the proposed heuristic algorithms was compared against the corresponding MILP model. In all the experiments performed, the heuristics were able to provide a much better trade-off between optimality and computation time; in all the tests the optimal solution was found within running times of hundreds of milliseconds, in contrast to tens of minutes (for the distributed) and even hours (for the centralized) needed to find the optimal solution with the MILP models. Thus, we use the heuristics to solve the instances in the scenario presented in the next section.

3.3 Performance evaluation

In this section, we present the scenario considered in our experiments and we show the results from solving the ELFADO problem considering a realistic instance; we evaluate the impact in the cost when distributed and centralized approaches are used for scheduling VM placement compared to a fixed placement, where no scheduling is done.

3.3.1 Scenario

For evaluation purposes, we implemented the proposed heuristic algorithms for the distributed and centralized ELFADO approaches on a scheduler in the OpenNebula cloud management middleware [OpNeb]. For comparison, a *fixed* approach, where the total workload is evenly distributed among the federated DCs, was also implemented.

We consider the global 11-location topology depicted in Fig. 3-4. Each location collects user traffic towards the set of federated DCs, which consists of five DCs strategically located in Taiwan, India, Spain, and Illinois and California in the USA. A global telecom operator provides optical connectivity among DCs, which is based upon the flexgrid technology. The number of users in each location was computed considering Wikipedia's audience by regions [MetaWiki] and it was scaled and distributed among the different locations in each region. Latency between location pairs was computed according to [Verizon].

Table 3-4 briefly presents the value considered for some representative energy parameters. Daily PUE values were computed according to [Go12] using data obtained from [USEIA]. Green energy coverage was obtained from [USEIA], [USEERE] and [Ki04] and brown energy cost for each DC was estimated from their respective local electric company rates (e.g. [EUEP] and [USBLS]). Servers in DCs are assumed to be HP ProLiant DL580 G3 [HP], equipped with four processors, 2 cores per processor, with $P_{server\text{-}idle}$ = 520W and $P_{server\text{-}max}$ = 833W.

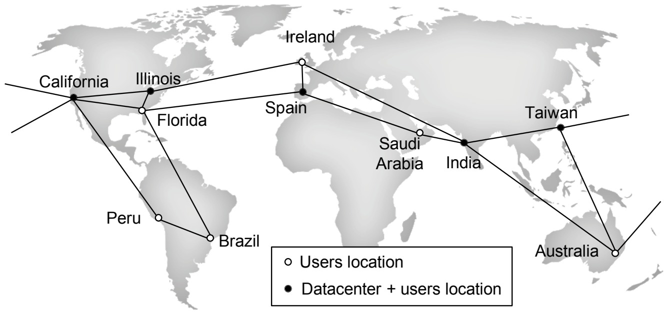

Fig. 3-4. Scenario: federated datacenters, locations and inter-datacenter network.

In line with [Fa08], DCs are dimensioned assuming a fat-tree topology with a maximum of M=48 clusters with two levels of switches and $M^2/4$=576 servers each. The number of VMs was set to 35,000, with individual image size of 5 GB; we assume that each VM runs in one single core. An integer number of clusters is always switched on, so as to support the load assigned to the DC; those servers without assigned load remain active and ready to accommodate spikes in demand. Green cover was set to ensure, at the highest green energy generation time, a proportion of energy β_d when all VMs run in DC d.

We consider a different type of switch, and thus a different power consumption value, for each layer of the intra-DC architecture. We selected the Huawei [HUAWEI] CloudEngine switches series; Table 3-5 specifies the model, switching capacity and power consumption for each considered switch.

Table 3-4. Value of energy parameters.

Datacenter	c_d (on/off peak) (€/kWh)	β_d	PUE (max/ avg)
Taiwan	0.0700 / 0.0490	0.5	1.671 / 1.632
India	0.0774 / 0.0542	0.9	1.694 / 1.694
Spain	0.1042 / 0.0729	0.9	1.670 / 1.457
Illinois	0.0735 / 0.0515	0.2	1.512 / 1.368
California	0.0988 / 0.0692	0.5	1.385 / 1.303

Table 3-5. Characteristics of Huawei CloudEngine switches.

Layer	Model	Switching capacity	Power consumption
Core	12812	48 Tb/s	P_{core} = 16,200 W
Aggregation	6800	1.28 Tb/s	P_{agg} = 270 W
Edge	5800	336 Gb/s	P_{edge} = 150 W

Finally, we consider that each DC is connected to the flexgrid inter-DC network through a router equipped with 100 Gb/s bandwidth variable transponders. Therefore, the actual capacity of optical connections is limited to that value. To compute the real throughput, we consider headers for the different protocols, i.e. TCP, IP, and GbE. The maximum amount of bytes to transfer, k_e, was computed to guarantee that VM migration is performed in less than 40 minutes.

3.3.2 Illustrative results

Results obtained in the different simulations carried out are presented and analyzed in the following paragraphs.

Fig. 3-5 (left) shows the availability of green energy as a function of the time (GMT) at each DC, $a_d(t)$, for a typical spring day, whereas the two rightmost graphs in Fig. 3-5 illustrate the behavior of the distributed (center) and centralized (right) ELFADO approaches. The distributed approach places VMs in DCs where the cost of energy (plus communication costs) is expected to be minimal in the next period; equation (3.1) is used for said energy cost estimation. However, in view of Fig. 3-5 (center), it is clear that equation (3.1) does not provide a clear picture, since all VMs are placed in India and Spain during the day periods where more green energy is available in those locations, thus exceeding green energy availability and paying a higher cost. In contrast, DC in Illinois seems to be very little utilized. Interestingly, the centralized approach reduces the percentage of VMs in those DCs with higher green coverage, to place only the amount of VMs (translated into powered-on clusters and servers) that the available green energy can support and placing the rest considering brown energy (and communication) costs. In fact, the DC in Illinois is more used in the centralized approach as a consequence of its cheaper brown energy cost compared to that of California.

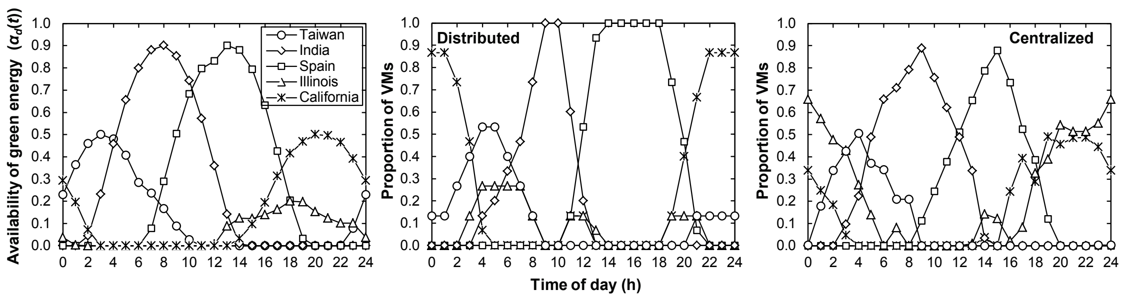

Fig. 3-5. Availability of green energy vs. time in all datacenters (left). Percentage of VMs in each datacenter when the distributed (center) and the centralized (right) approaches are applied.

Fig. 3-6 presents costs and performance as a function of the time for all three approaches; cost per transmitted bit was set to 1e-9 €/Gb*km. Energy costs per hour plots in Fig. 3-6 (left) show a remarkable reduction in energy costs when some ELFADO approach is implemented, with respect to the fixed approach. Daily comparison presented in Table 3-6 shows savings of 11% for the distributed and over 52% for the centralized approach. Hourly plot for the distributed approach clearly highlights how by placing VMs in DCs where the estimated energy is cheaper, results in a high amount of brown energy being drawn from the grid at a more expensive price. In contrast, the centralized approach leverages green energy arriving to virtually zero energy cost in some periods.

Regarding communication costs (Fig. 3-6 (center)), the distributed approach shows a more intensive use, presenting three peaks, exactly when the DC in Illinois is used to compensate energy costs between green energy availability peaks in the rest of DCs. However, although the centralized approach is less communications intensive, the total daily communications costs are only under 6% cheaper compared to the distributed approach, as shown in Table 3-6.

Aggregated daily costs are detailed in Table 3-6 for all three approaches. As shown, the distributed approach saves only 2% of total cost when compared to the fixed approach. Although, that percentage represents more than 100€ per day, it is just a small portion of the savings obtained by the centralized approach, which are as high as just over 44% (more than 2.6 k€ per day).

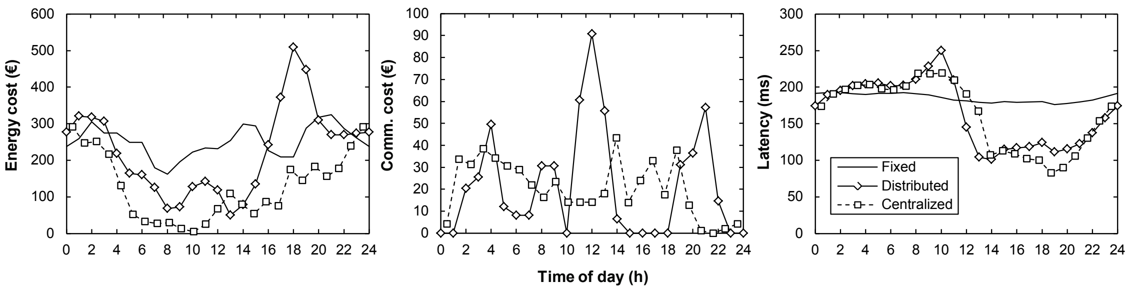

Fig. 3-6. Energy (left) and communication (center) cost per hour against time. Latency vs. time (right).

Table 3-6. Comparison of daily costs and performance.

Approach	Energy cost	Comm. cost	Total cost	Average latency
Fixed	6,048 €	-	6,048 €	185.2 ms
Distributed	5,374 € (11.1%)	537 €	5,912 € (2.3%)	164.2 ms (11.3%)
Centralized	2,867 € (52.6%)	508 € (5.8%)	3,376 € (44.2%)	161.5 ms (12.8%)

Regarding performance (latency), both the distributed and the centralized approach provide figures more than 10% lower than that of the fixed approach as shown in Table 3-6. Hourly plots presented in Fig. 3-6 (right) show that latency is slightly higher during some morning periods under both, the distributed and the centralized ELFADO, with respect to that of the fixed; however, in after noon periods, both approaches reduce latency extraordinary since VMs are placed closer to users.

The results presented in Fig. 3-6 were obtained by fixing the value of th_v to 1.3*average(*latency_fixed*) (specified in Table 3-6), so as to allow obtaining worse hourly performance values in the hope of obtaining better daily ones. Fig. 3-7a gives insight of the sensitivity of costs to the value of that threshold. Fixed costs are also plotted as a reference. Costs in the centralized approach show that even for very restrictive thresholds, noticeable cost savings can be obtained. In addition, when the threshold is set to the average latency in the fixed approach or above, obtained costs are almost constant. In contrast, the distributed approach proves to be more sensible to that threshold, reaching a minimum in terms of costs when the threshold value is 30% over the average latency in the fixed approach.

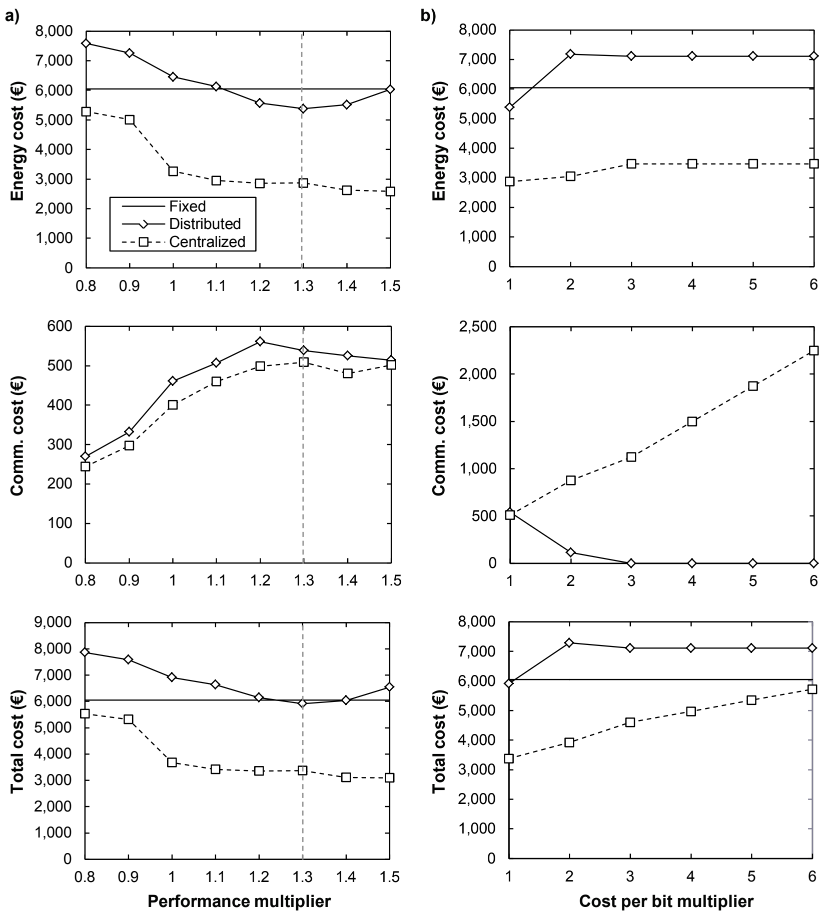

Fig. 3-7. Cost per day vs. performance threshold (a) and against cost per bit (b).

Finally, Fig. 3-7b illustrates the influence of the cost per bit to transfer VMs from one DC to another. As before, fixed costs are plotted for reference. Energy costs in the distributed approach increase sharply when the cost per bit doubles, almost preventing from moving VMs, as clearly shown Fig. 3-7b (middle). Nonetheless, energy costs are almost stable in the centralized approach. Recall that the proposed heuristic focuses on green energy availability as the first indicator for placing VMs. In fact, communication cost increase linearly with the increment in the cost per bit. However, it is not until the cost per bit increases more than 6 times when the centralized approach cost equals that of the fixed approach.

3.4 Conclusions

The enormous energy consumption of DCs translates into high operational expenditures for DC operators. Although the use of green energy allows reducing the energy bill, its availability is limited depending on the hour of the day, weather and season, etc. Federating DCs can be a way for independent DC operators to not only increase their revenue but also reduce operational expenditures. Aiming at optimizing costs whilst ensuring the desired QoE for users, this chapter described and formally stated the ELFADO problem to orchestrate federated DCs, placing workloads in the most convenient DC.

Two approaches to solve the ELFADO problem were compared, distributed and centralized, where mathematical formulations as well as heuristic algorithms for scheduling VM placement were proposed. The distributed approach is based on running scheduling algorithms inside DC resource managers to compute periodically the optimal placement for the VMs currently in the local DC. VMs are placed in DCs where the cost (energy and communications) is expected to be minimal for the next period. In this approach, energy costs are estimated since the total amount of VMs to be placed in each DC is computed in a distributed manner. Therefore, the available green energy could not be enough to cover the whole energy consumption in each DC. In contrast, the centralized approach, proposes a federation orchestrator to compute the global optimal placement for all the VMs in the federated DCs. VMs are placed in DCs so as to take full advantage from green energy availability. This is possible as a result of computing the placement of all VMs in the proposed federation orchestrator at the same time.

Results showed that both ELFADO approaches improve QoE by reducing average latency more than 10% with respect to a fixed approach where no scheduling is performed. Regarding costs, the distributed approach can save up to 11 % of costs with respect to the fixed approach. However, when communication costs are considered, total cost savings are reduced to only 2%. The centralized approach showed remarkable energy cost savings circa 52%, resulting in 44% when communication costs are taken into account. Finally, it was shown that the centralized approach provides cost savings even when the cost per bit increases 6 times.

Chapter 4

Inter-datacenter networks

Current inter-DC connections are configured as static big fat pipes, which entails large bitrate overprovisioning and thus high operational costs for DC operators. Besides, network operators cannot share such connections between customers, because DC traffic varies greatly over time. Those connections are mainly used to perform VM migration and DB synchronization among federated DCs, allowing thus elastic DC operations. To improve resource utilization and save costs, dynamic inter-DC connectivity is currently being targeted from a research point of view and in standardization form.

In this chapter, we show that dynamic connectivity is not enough to guarantee elastic inter-DC operations and might lead to poor performance provided that not enough overprovisioning of network resources is performed. To alleviate it to some extent a dynamic and elastic connectivity model taking advantage of elastic network resources allocation in flexgrid-based optical networks is proposed. Flexgrid technology enables finer spectrum granularity adaptation and the ability to dynamically increase and decrease the amount of optical resources assigned to connections.

Additionally, transfer mode requests for cloud operations are proposed. To provide an abstraction layer to the underlying network, a new stratum on top of the ABNO, the carrier SDN, could be deployed, implementing a northbound interface to request transfer operations and using application-oriented semantic, liberating application developers from understanding and dealing with network specifics and complexity.

4.1 Dynamic connection requests

Evolution towards cloud-ready transport networks entails dynamically controlling network resources, considering cloud requests in the network configuration process. Hence, that evolution is based on elastic data and control planes, which can interact with multiple network technologies and cloud services. A cross-stratum orchestrator (CSO) between the cloud and the interconnection network is eventually required to coordinate resources in both strata in a coherent manner.

When considering cloud-ready transport network, for the control plane to dynamically set up and tear down connections, the entry point from applications to the network is ABNO [Ki13].

To manage connections dynamically, we propose a CSO module to coordinate cloud and network (Fig. 4-1). In addition to the cloud and network management for local DC resources, the proposed CSO implements new components to facilitate DC federation operations:

- An IT resources coordination and synchronization module in charge of coordinating VM migration and DB synchronization among federated DCs.
- A connection manager and a virtual topology DB.

The set of VMs to migrate and the set of DBs to synchronize are computed by the scheduler inside the cloud manager, which takes into account the availability of PMs in the rest of the DCs, and the high-level performance and availability goals of the workloads hosted in the DCs. Once those sets are computed, the CSO module coordinates intra and inter-DC networks to perform the transfers. The CSO interfaces with the local network controller, remote CSOs, and inter-DC controller, implementing the ABNO architecture described in Chapter 2.

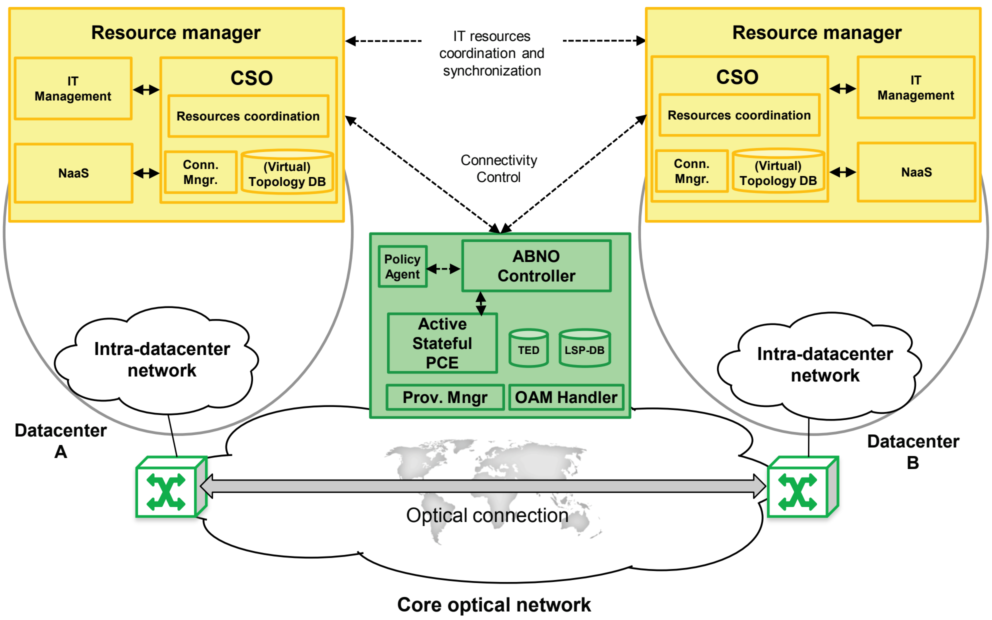

Fig. 4-1. Cross-stratum orchestration architecture.

When a request is received from a CSO, the controller in the ABNO verifies rights, asking the policy agent to check maximum bandwidth, origin and destination nodes, and so on. Then the controller requests to the PCE a label switched path (LSP) between both locations. Once the PCE finds a path for the requested capacity, it delegates its implementation to the provisioning manager. The provisioning manager creates the path using some interface; PCEP is proposed in this work to forward the request to the source node in the underlying flexgrid network, so that node can start signaling the connection [ZAl13]. After the connection has been set up properly, the source node notifies the provisioning manager, which in turn updates the TED and the LSP-DB. A response is sent back to the originating CSO after the whole process ends.

In contrast to static connectivity, dynamic connectivity allows DCs to manage optical connections to remote DCs, requesting connections as they are really needed to perform data transfers and releasing them when all data has been transferred. Furthermore, the fine spectral granularity and wide range of bitrates in elastic optical networks makes the actual bitrate of the optical connection closely fit connectivity needs. After requesting a connection and negotiating its capacity as a function of the current network availability, the resulting bandwidth can be used by scheduling algorithms to organize transferences.

Nonetheless, the availability of resources is not guaranteed, and the lack of network resources at request time may result in long transference times and even in transference period overlapping. Note that a connection's bitrate cannot be renegotiated and remains constant along the connection's holding time. To reduce the impact of the unavailability of required connectivity resources, we propose to use the elasticity for resources allocation that the flexgrid technology provides, allowing the amount of spectral resources assigned to each connection to be increased (or decreased), and thus its bitrate. This adaptation is done if there are not enough resources at request time so that more bandwidth may be requested at any time after the connection has been set up. We refer to this type of connectivity as dynamic elastic.

Using the architecture in Fig. 4-1 the CSO is able to request optical connections dynamically to the ABNO negotiating its bitrate. The sequence in Fig. 4-2 for the dynamic connectivity illustrates messages exchanged between CSO and ABNO to set up and tear down an optical connection. Once the CSO has computed a transfer to be performed, it requests an 80 Gb/s optical connection to a remote DC; the policy agent inside ABNO verifies local policies and performs internal operations (in our implementation, it forwards the message to the PCE) to find a route and spectrum resources for the request. Assuming that not enough resources are available for the bitrate requested, an algorithm inside ABNO finds the maximum bitrate and, at time t_1, it sends a response to the originating CSO with that information. Upon the reception of the maximum available bitrate, 40 Gb/s in the example, the CSO recomputes the transfer, reduces the amount of data to transfer, and requests a connection with the corresponding available bitrate. When the transfer ends, t_2, the CSO sends a message to ABNO to tear down the connection, and the used resources are released so they can be assigned to any other connection.

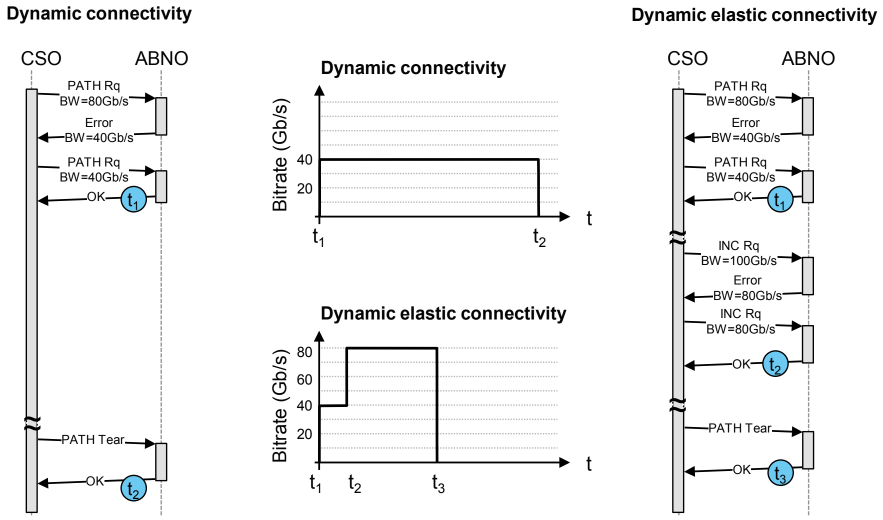

Fig. 4-2. Dynamic and dynamic elastic connectivity models.

In the dynamic elastic connectivity model, the CSO is able to request increments in the bitrate of already established connections. In the example in Fig. 4-2, after the connection has been established in t_1 with half of the initially requested bitrate, the CSO sends periodical retrials to increment its bitrate. In the example, some resources are released after the connection has been established, and after a request is received in t_2, they can be assigned to increment the bitrate of the already established connection; the assigned bitrate increases then to 80 Gb/s, which reduces the total transfer time. Note that this is beneficial for both DC federation, since better performance could be achieved, and the network operator, since unused resources can be immediately occupied. Finally, the connection is torn down at t_3.

4.1.1 Performance evaluation

For evaluation purposes, we implemented CSO on a scheduler in the OpenNebula cloud management middleware, whereas the flexgrid network and ABNO architecture were implemented as an ad-hoc event-driven simulator in OMNeT++. An XML-based protocol was developed to communicate between the CSO and the ABNO. In the scheduler inside cloud management, a follow-the-work strategy was implemented for VM migration, where VMs are moved to DCs closer to the users, reducing thus the user-to-service latency. The DB synchronization policy tries to update the differential images between services running in all the DCs; in the case that an image cannot be synchronized in time, the next update will attempt to synchronize the whole DB image, increasing the traffic among DCs (DC2DC) overhead. Regarding the PCE, the algorithms described in [Ca12] and [Kl13], for routing and spectrum allocation and elastic operations, respectively, were implemented.

Similarly as in the previous chapter, we consider a scenario consisting of 11 locations around the globe; each location is used as source for traffic between users and DCs. DCs are placed in: Illinois, Spain, India, and Taiwan. DC2DC and U2DC traffic compete for resources in the physical network. Users' traffic connection requests arrive following a Poisson process and are sequentially served without prior knowledge of future incoming connection requests. The destination DC of each U2DC connection is the closest DC to the user's location in case of requesting a distributed service or to the DC containing the VM in case of singular services. To represent a user's activity along a day, the bitrate demanded by each U2DC connection request is proportional to the number of active users at the requesting time, for example, several gigabits per second during office hours and a few megabits per second at night. The holding time of connections is exponentially distributed with the mean value equal to 2 h. Different values of offered network load can be considered by changing the arrival rate while keeping the mean holding time constant. Specifically, in this section two offered loads for U2DC traffic are considered:

- Low load, unleashing U2DC blocking probability < 0.2%
- High load, unleashing blocking probability < 0.6%

To conduct all the simulations, optical spectrum width was set to 4 THz, spectral granularity to 6.25 GHz, and the capacity of transponders interfacing the flexgrid network to 1 Tb/s. As for the DC federation, the number of VMs was set to 35,000, with an image size of 5 Gbytes each, while the number of DBs was set to 300,000 distributed among DCs, each with a size of 5 Gbytes, about half the size of Wikipedia [WiSize]; at each scheduling round we assume that only 450 MB change. In addition, protocol stack overheads were considered: raw data to be transferred is transported on TCP packets, which are encapsulated into IPv4 packets; multiprotocol label switching (MPLS) and gigabit Ethernet (GbE) headers are added afterward.

We set a scheduling round each hour, so the objective is to perform the required VM migration and DB synchronization within each time period. In fact, the shorter the transfer time, the better the offered network service, by having the VMs in their proper locations earlier and the DBs up to date. Fig. 4-3 presents results for the static connectivity. The bitrate of static connections is plotted in Fig. 4-3a as a function of the average time to complete transfers so as to ensure that all transfers are performed within 1 h for both VM migration and DB synchronization. It is clear that the larger the bitrate of the connections, the shorter the transfers. To minimize the bitrate overprovisioned for the static connections, let us assume an objective of 30 minutes on average for the transferences. Thus, connections of 200

Gb/s and 150 Gb/s are needed for DB synchronization and VM migration, respectively; although one connection would be enough in this case, two different connections are established for the sake of differentiating usage. Using those bitrates, time-to-transfer VMs and DBs as a function of the time along one day in the connection between DCs in Spain and Illinois are plotted in Fig. 4-3b. As shown, time-to-transfer is always lower than 60 minutes, although it presents peaks of 50 minutes and above.

The used bitrate to synchronize DBs and get VMs migrated is also illustrated in Fig. 4-4. Note that since the bitrate of the optical connections has been fixed to 200 Gb/s for DBs and 150 Gb/s for VMs, different times are obtained in line with the time-to-transfer plots.

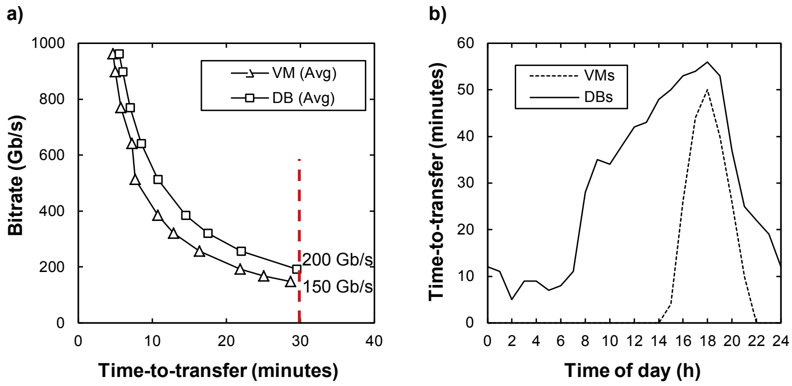

Fig. 4-3. Bitrate vs. time-to-transfer (a) and time-to-transfer vs. day-hours (b) for static connectivity.

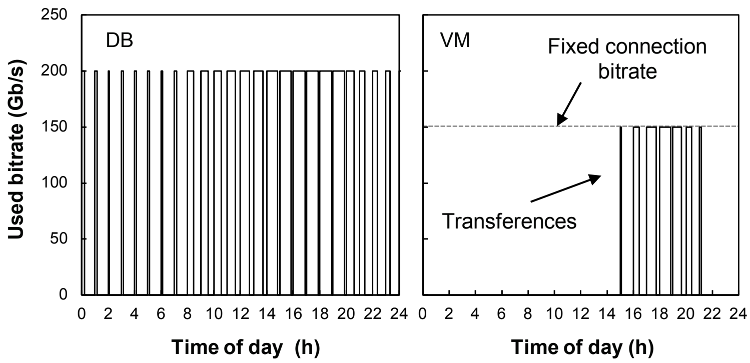

Fig. 4-4. Used bitrate vs. time for static connectivity.

Fig. 4-5 and Fig. 4-6 show the used bitrate for the dynamic and dynamic elastic connectivity models and two different cases regarding U2DC traffic: low (Fig. 4-5) and high (Fig. 4-6). Recall that U2DC traffic and DC2DC traffic compete for spectral resources.

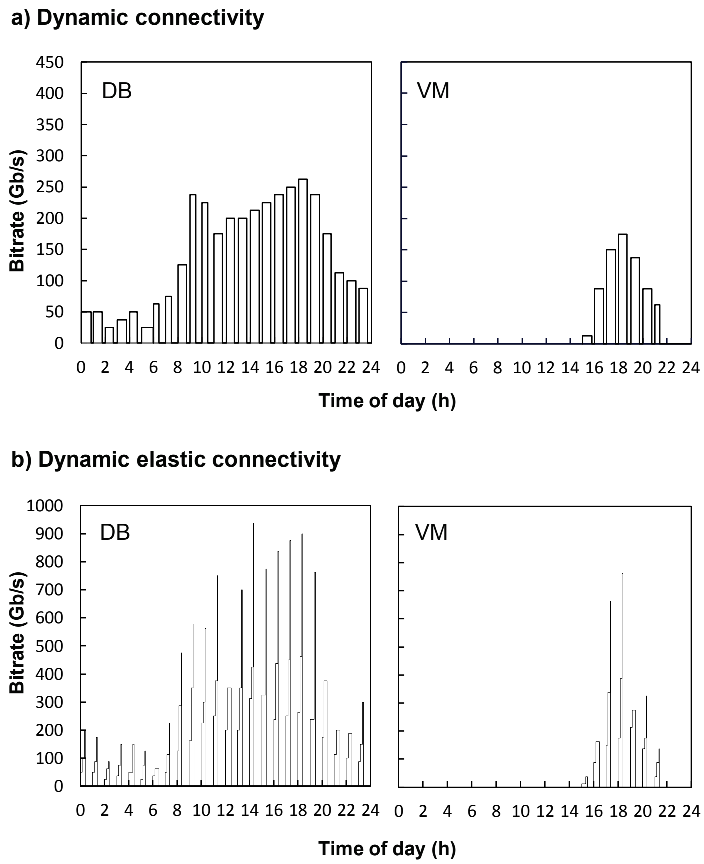

Fig. 4-5. Used bitrate vs. time in low U2DC traffic scenario.

In the dynamic model under low U2DC, time-to-transfer is kept clearly under 60 minutes for both DBs and VMs. Note, however, that a connection's bitrate varies with the amount of data to transfer; connection bitrates for DBs are as low as 50 Gb/s and as high as 250 Gb/s, in contrast to the constant 200 Gb/s bitrate used in the static model. However, as soon as the U2DC traffic increases, the amount of needed bitrate for the dynamic connections might not be available at the time of the request, so this model is not able to perform data transfers within 1 h, leading to period overlapping, as shown in Fig. 4-6a; this fact yields poor performance as some DBs become degraded and users perceive an increased latency.

The dynamic elastic model is able to keep time-to-transfer under 60 minutes for both low and high U2DC traffic. As shown, the initial bitrate for the connections is comparable to that of the connections for the dynamic model. However, since the CSO can perform elastic bitrate operations on the established optical connections, it is able to increase a connection's bitrate even under high U2DC traffic. In fact, as detailed in Table 4-1, transfer times are clearly lower than those obtained with the dynamic and even the static connectivity models.

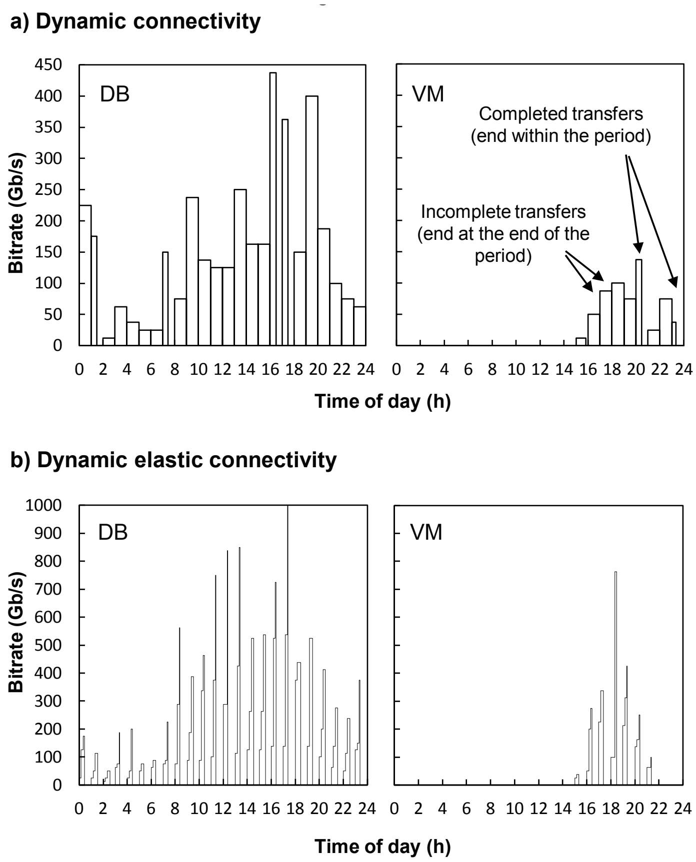

Fig. 4-6. Used bitrate vs. time in high U2DC traffic scenario.

The cost of elasticity is in the control plane, since the amount of messages that need to be processed is slightly increased; 40.8 request messages/h to increase connections' capacity in addition to 36.8 messages/h for connections' set up and tear down. More than half of requests (53.8%) were successful, and the optical connection was expanded; moreover, each connection got 1.67 increments on average during each period. It is worth highlighting that although the remaining 46.2% of requests were not successful, the corresponding connections maintained their prior assigned capacity.

Finally, it is worth pointing out that the bitrate savings that can be obtained by implementing any of the proposed dynamic connectivity models is as high as 60% with respect to the static model (Table 4-2). Note that in the static model 12 connections for DB synchronization and 4 connections for VM migration need to be permanently established.

Although DC resource managers can request optical connections and control their capacity dynamically, this involves the resource managers to implement algorithms and interfaces to deal with network specifics and complexity. To solve those issues, in the next section, transfer mode requests are prososed taking advantage of carrier SDN, as a new element in between resource managers and ABNO.

Table 4-1. Connectivity models comparison.

<table>
<tr><td colspan="2" rowspan="2"></td><th colspan="3">Time-to-transfer (minutes)</th></tr>
<tr><th>Static</th><th>Dynamic</th><th>Elastic</th></tr>
<tr><th rowspan="2">Time-to-transfer (minutes)</th><th>DB synchronization (Max. / Avg.)</th><td>54.0 / 28.5</td><td>58.0 / 39.9</td><td>49.0 / 24.7</td></tr>
<tr><th>VM migration (Max. / Avg.)</th><td>50.0 / 28.7</td><td>48.0 / 39.9</td><td>40.0 / 24.4</td></tr>
<tr><th colspan="2">Bitrate savings</th><td>-</td><td>59.1%</td><td>57.9%</td></tr>
</table>

Table 4-2. Elastic operations per period.

Total signaling	**% success**	**# per connection**
40.8	53.8%	1.67

4.2 Transfer mode requests

Even though local resource managers can request optical connections' set up, tear down, and on demand adapt their capacity, the flexgrid interconnection network supports additional traffic for different services and clients. Therefore, competence for network resources could lead to connections' capacity being reduced or even blocked at requesting time. In that case, resource managers can either perform connection request retries, similar to I/O polling (software-driven I/O) in computers, to increase the bitrate of already established connections or set up new ones, although without guarantees of success, resulting in a poor cloud performance.

To alleviate to some extent the dependency between cloud management and network connectivity, in this section we propose a novel network-driven connectivity model. As illustrated in Fig. 4-7, an abstraction layer on top of the ABNO-based control architecture, the carrier SDN, could be deployed. A carrier SDN controller implements a northbound interface to request transfer operations. Those applications' operations are transformed into network connection requests. The northbound interface uses application-oriented semantic, liberating application developers from understanding and dealing with network specifics and complexity.

It is worth noting that the dynamic elastic model proposed before can also request connections to carrier SDN instead of the ABNO. In this case, the carrier SDN acts as a proxy between applications and network.

The SDN controller is in charge of managing inter-DC connectivity; if not enough resources are available at requesting time, notifications (similar to interruptions in computers) are sent from the ABNO to the SDN controller each time specific resources are released. Upon receiving a notification, the SDN controller takes decisions on whether to increase the bitrate associated to a transfer. Therefore, we have effectively moved from polling to a network-driven transfer mode. Fig. 4-8 illustrates both control architectures supporting dynamic and transfer mode requests.

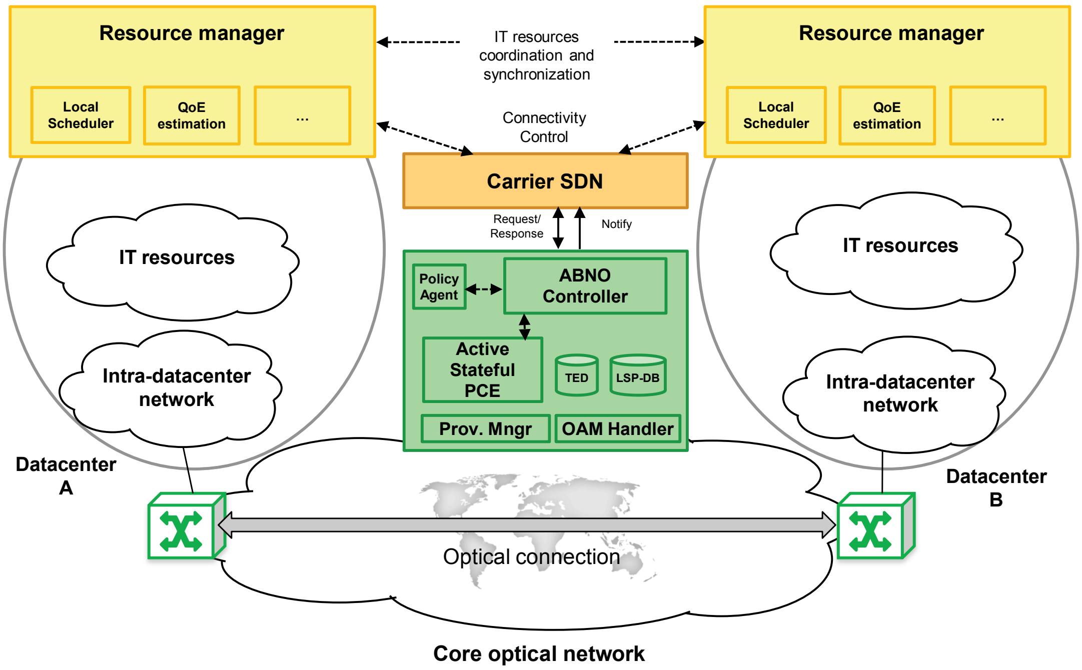

Fig. 4-7. Carrier SDN implementing transfer operations.

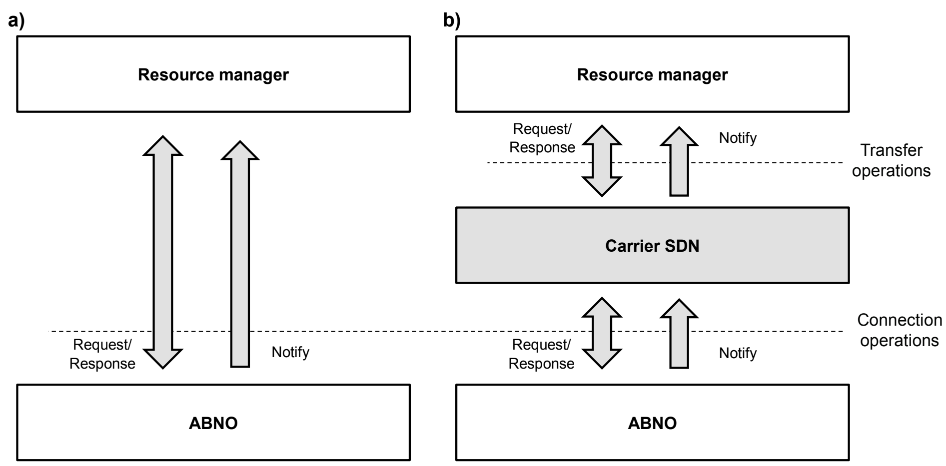

Fig. 4-8. Control architectures supporting dynamic connections (a) and transfer mode (b) requests.

In the network-driven model (Fig. 4-9), applications request transferences instead of connectivity. The corresponding source cloud resource manager sends a transfer request to the SDN controller specifying the destination DC, the amount of data that needs to be transferred, and the maximum completion time. Upon its reception, the SDN controller requests the ABNO controller to find the greatest spectrum width available, taking into account local policies and current service level agreements and sends a response back to the resource manager with the best completion time. The source resource manager organizes data transference and sends a new transfer request with the suggested completion time. A new connection is established and its capacity is sent in the response message; in addition, the SDN controller requests ABNO controller to keep it informed upon more resources are left available in the route of that LSP. ABNO controller has access to both the traffic engineering database and the LSP-DB. Algorithms deployed in the ABNO controller monitor spectrum availability in those physical links. When resource availability allows increasing the allocated bitrate of some LSP, the SDN controller performs elastic spectrum operations so as to ensure committed transfer completion times. Each time the SDN controller modifies bitrate by performing elastic spectrum operations, a notification is sent to the source resource manager containing new throughput. Cloud resource manager then can optimize VM migration as a function of the actual throughput while delegating ensuring completion transfer time to the SDN controller.

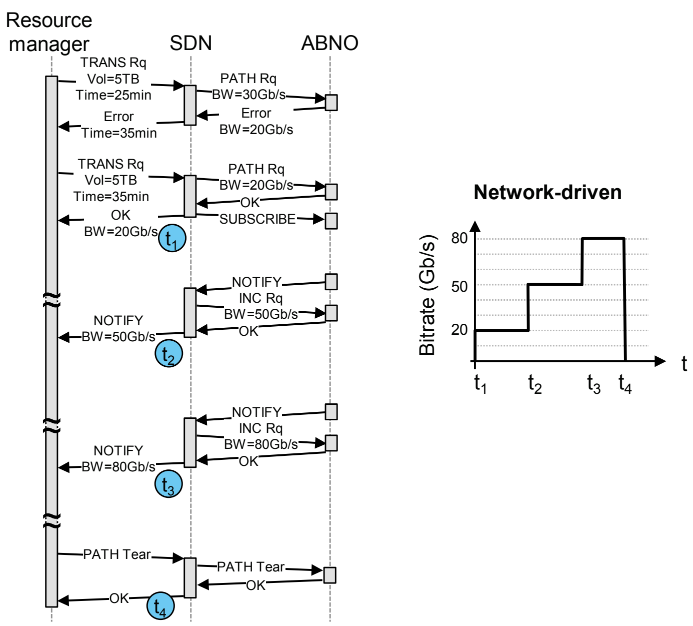

Fig. 4-9. Network-driven transfer mode requests.

4.2.1 Performance evaluation

Similarly as for the evaluation of the dynamic and dynamic elastic models, we developed scheduling algorithms in an OpenNebula-based cloud middleware emulator. Federated DCs are connected to an ad-hoc event-driven simulator developed in OMNET++. In this case, the simulator implements the SDN controller as well as the flexgrid network with an ABNO controller on the top, as described in Fig. 4-8. Regarding PCE, the algorithm described in [As13] for elastic operations was implemented.

We assume the global 11-node topology depicted in Fig. 4-10. These locations are used as source for traffic between users and DCs. In addition, four DCs are strategically placed in Illinois, Spain, India, and Taiwan. DC2DC and U2DC traffic compete for resources in the physical network. We fixed the optical spectrum width to 4 THz, the spectral granularity to 6.25 GHz, the capacity for the ports connecting DCs to 1 Tb/s, the number of VMs to 35,000 with an image size of 5 GB each and we considered 300,000 DBs with a differential image size of 450 MB and a total size of 5 GB each; half the size of Wikipedia [WiSize]. Additionally, TCP, IPv4, GbE and MPLS headers have been considered.

Fig. 4-11 shows the required bitrate to migrate VMs and synchronize DBs in 30 minutes. VM migration is performed as a follow-the-work strategy and thus, connectivity is used only during part of the day. In contrast, DB synchronization is performed along the day, although bitrate depends on the amount of data to be transferred, i.e. on users' activity.

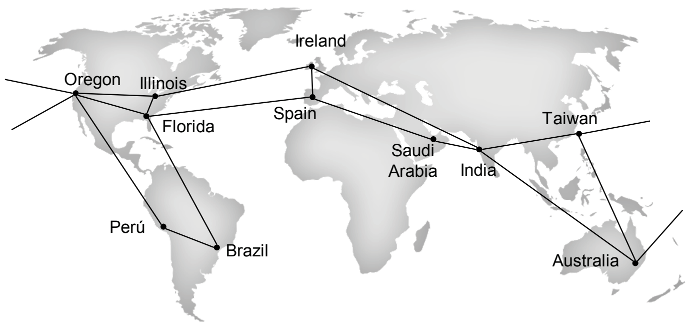

Fig. 4-10. Worldwide topology.

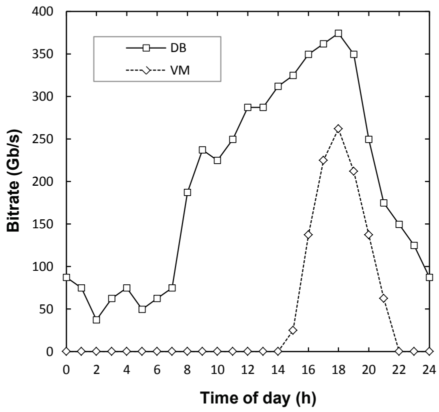

Fig. 4-11. Required bitrate vs. time.

Fig. 4-12a depicts the assigned bitrate for DB synchronization and VM migration, between two DCs during a 24 hours period when the dynamic elastic model is used. Fig. 4-12b shows the assigned bitrate when the network-driven model is used. The dynamic elastic model tends to provide longer transfer times as a result of not obtaining additional bitrate in retries. Note that intervals tend to be narrower using the network-driven model. The reason is that, said model, assigns additional bitrate to the connections as soon as resources are released by other connections.

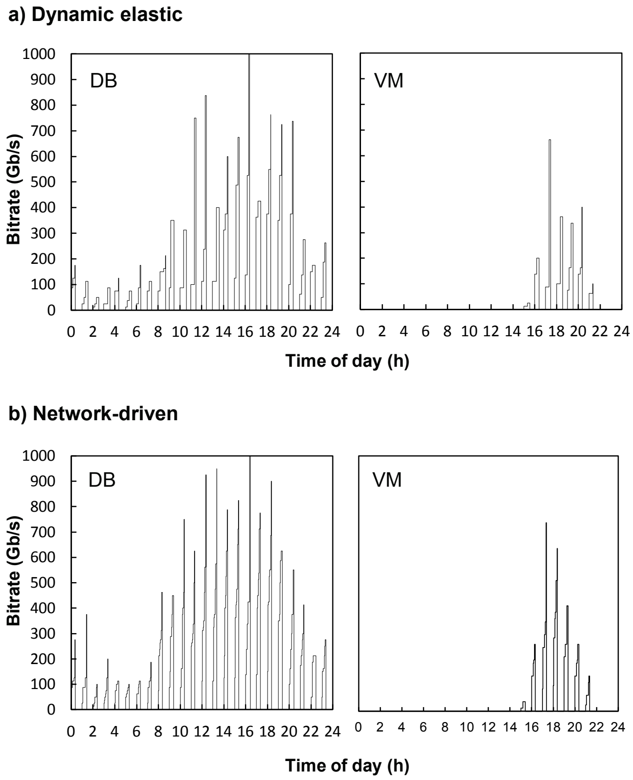

Fig. 4-12. Used bitrate for the dynamic elastic (a) and network-driven (b) models.

Table 4-3 shows that when using the network-driven model both the maximum and average required time-to-transfer are significantly lower than when the dynamic elastic model is used. The longest transfers could be done in only 28 minutes when the network-driven model was used compared to just below 60 minutes using the dynamic elastic model. Note that the amount of requested bitrate is the same for both models. Additionally,

Table 4-4 shows the number of required requests messages per hour needed to increase bitrate of connections for the whole scenario. As illustrated, only 53% of those requests succeeded to increase connections' bitrate under the software-driven model, in contrast to 100% reached under the network-driven model.

Table 4-3. Time-to-transfer.

	Max/Avg Time-to-transfer (minutes)	
	DBs	**VMs**
Dynamic elastic	58.0 / 29.0	54.0 / 28.2
Network-driven	28.0 / 22.4	27.0 / 22.2

Table 4-4. Elastic operations.

	Requests	
	#/h	**% success**
Dynamic elastic	43.1	53.5%
Network-driven	65.3	100%

4.3 Conclusions

This chapter presented an architecture for a distributed cross-stratum orchestrator to coordinate the cloud and network based on elements under standardization at IETF. The CSO allows elastic DC operations to be performed as well as the dynamic establishment and tear down of inter-DC connections. To control the interconnection network, an ABNO architecture based on an active stateful PCE was considered. The CSO and ABNO controller negotiate the parameters of each connection to be established, including its bitrate.

Even with dynamic connectivity, some resource overprovisioning still needs to be done to guarantee that resources are available when needed. Aiming to improve the performance of dynamic connectivity, we propose to perform elastic operations on already established connections allowing the CSO to retry when not enough resources are available at a connection's set up. Elastic connections are supported by flexgrid-based interconnection networks, which enable finer spectrum granularity and the ability to increase and decrease the amount of optical resources assigned to connections already established.

Illustrative results showed that dynamic elastic connectivity improves the performance of the dynamic one in scenarios where the physical optical network is shared by several services, so competence to use network resources arises. Finally, it was shown that dynamic connectivity could entail bitrate savings as high as 60 percent with respect to static connectivity. This fact becomes remarkable for DC operators willing to reduce OPEX and paves the way to devise scheduling algorithms that might use cheaper inter-DC connectivity to improve service performance, reducing OPEX by, for example, minimizing energy consumption.

In addition, carrier SDN has been studied in a DC federation scenario. A carrier SDN controller implementing a northbound interface with application-oriented semantic has been considered and a transfer mode requests model proposed. In contrast to the dynamic elastic model, which needs periodical retries requesting to increase connection's bitrate and may do not translate into immediate bitrate increments, the network-driven model takes advantage from the use of notify messages, thus being able to reduce time-to-transfer remarkably.

Finally, it is worth highlightning that the proposed network-driven model opens the opportunity to network operators to implement policies so as to dynamically manage connections' bitrate of a set of customers and fulfill simultaneously their SLAs.

Chapter 5

Performance evaluation using elastic connectivity

Extending the scenario for ELFADO problem presented in Chapter 3, in this chapter cost savings considering both energy and communication costs are analyzed in a DC federation when using the dynamic elastic connectivity and the network-driven models proposed in the previous chapter.

5.1 Federation orchestrator and elastic connectivity

In Chapter 3, we concluded that the centralized approach for ELFADO problem results in higher savings in operational costs in contrast with a distributed approach when dominated by energy and communication costs. Therefore, we assume a federation orchestrator computing periodically the global optimal placement for all the VMs in the federated DCs so as to minimize operational costs whilst ensuring QoE. In addition, in this chapter local resource managers interface not only the rest of DCs to coordinate VM migration but also the ABNO controller and the carrier SDN controller to request optical DC2DC connections' set up, tear down and capacity increments and transfer mode operations respectively. Fig. 5-1 and Fig. 5-2 illustrate the architecture considered when using the dynamic elastic, also referred as application-driven, (Fig. 5-1) and transfer mode, network-driven, (Fig. 5-2) connectivity models.

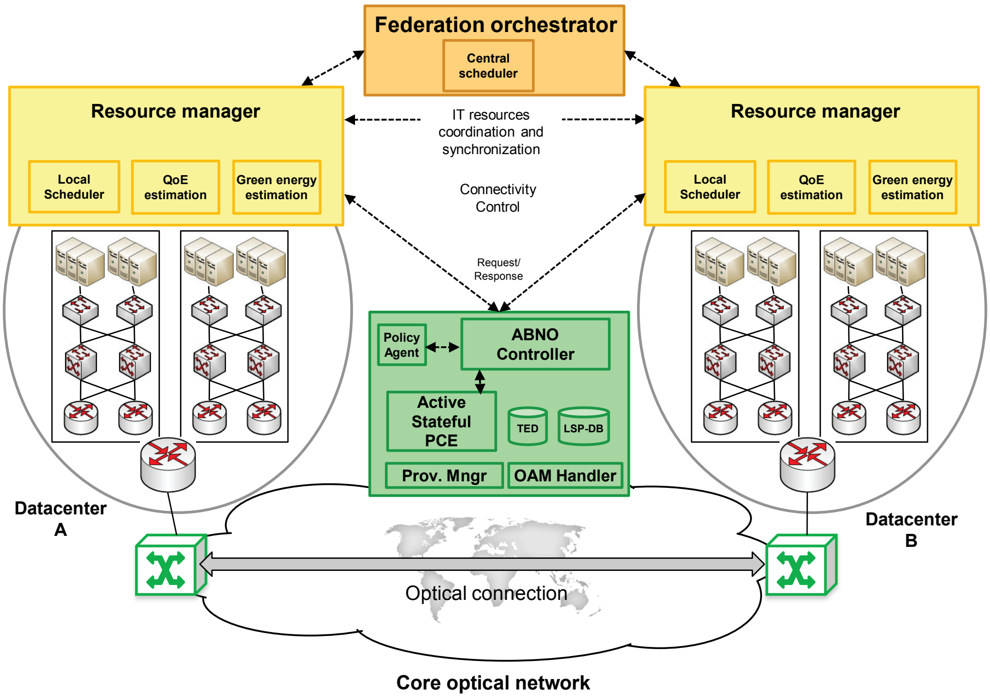

Fig. 5-1. Architecture for dynamic elastic connection requests.

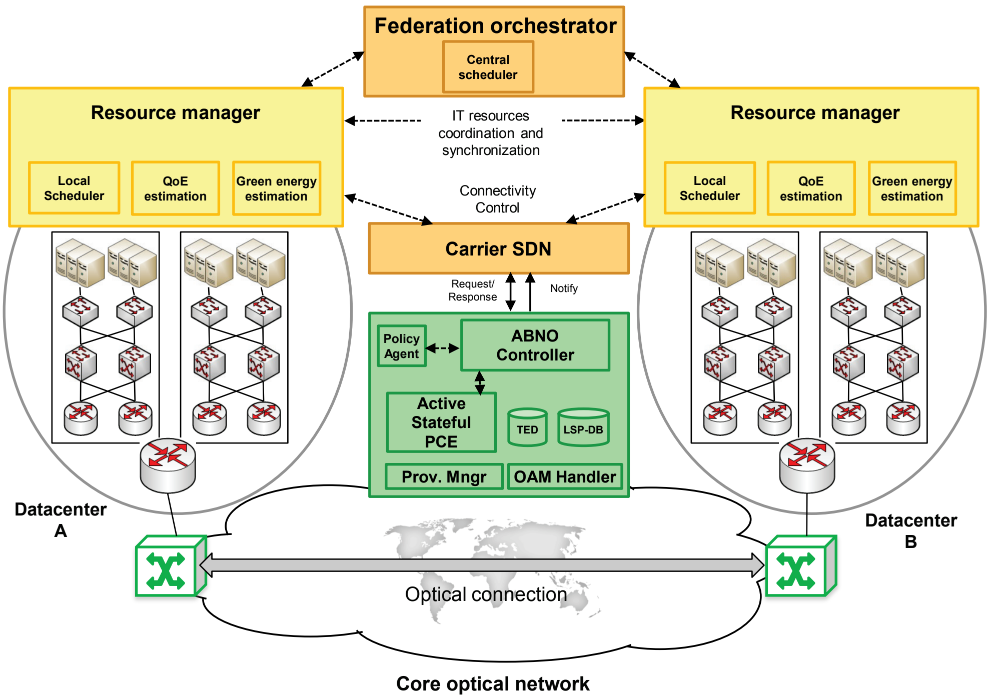

Fig. 5-2. Architecture for transfer mode requests.

It is worth mentioning that, as in Chapter 3, in this chapter it is assumed that DCs are dimensioned to cover some proportion β_d of the total energy consumption for the maximum dimensioning. Then, green coverage in DC d, α_d, can be estimated as, $\alpha_d(t) = \beta_d * \delta_d(t)$, where δ_d(t) is the normalized availability of green energy as a function of the time of day in the location of DC d. The amount of green energy available at period t can be estimated as $g_d(t)= \alpha_d(t)*Energy\_MaxDimensioning$. Therefore, knowing the VMs to be placed in each DC in the next period, the federation orchestrator can compute precisely the amount of workload in each DC, compute the green energy available and thus, the optimal VM placement. Finally, after the federation scheduler schedules the next period, local resource managers can start performing VM migration requesting dynamic elastic connectivity to ABNO or transfer mode operations to the carrier SDN controller.

Table 5-1 presents the algorithm implemented in the carrier SDN controller for transfer requests. It translates requested data, amount of data to transfer and completion time, into a required bitrate, taking into account frequency slice width in the flexgrid network (line 1). Next an optical connection request is sent towards the ABNO controller, specifying source and destination of the connection and the bitrate (line 2). In case of lack of network resources (lines 3-6), the maximum available bitrate between source and destination DCs is requested to the ABNO controller, its result is translated into the minimum completion time, which is used to inform the requesting DC resource manager. If the connection could be established, the carrier SDN requests a subscription to the links in the route of the connection, so as to be aware of available resources as soon as they are released in the network (line 7). Finally, the actual completion time is recomputed taking into consideration the connection's bitrate and communicated back to the requesting DC resource manager.

Table 5-1. Algorithm for transfer mode requests.

```
INPUT source, destination, dataVol, rqTime, sliceWidth
OUTPUT Response
1:  minBitrate ← translateAppRequest (dataVol, rqTime, sliceWidth)
2:  netResp ← requestConnection (source, destination, minBitrate)
3:  if netResp==KO then
4:    maxBitrate ← getMaxBitrate (source , destination)
5:    minTime ← translateNetResponse (maxBitrate)
6:    return {KO, minTime}
7:  requestSubscription (netResp.connId.route)
8:  time ← translateNetResponse (dataVol, netResp.connId.bitrate)
9:  return {OK, netResp.connId, time}
```

5.2 Performance evaluation

For evaluation purposes, we developed resource managers in an OpenNebula-based cloud middleware emulator. The federation orchestrator with the centralized scheduling algorithm was implemented as a stand-alone module in Java. Federated DCs are connected to an ad-hoc event-driven simulator developed in OMNET++. The simulator implements the carrier SDN controller and the flexgrid network with an ABNO controller on the top, as illustrated in Fig. 5-2. Finally, the algorithm described in [As13] for elastic spectrum allocation was implemented.

In line with the previous chapters, a global 11-node topology is considered and locations are used as source for U2DC traffic collecting user traffic towards the set of DCs, which consists of five DCs strategically located in Taiwan, India, Spain, and Illinois and California in the USA. A global telecom operator provides optical connectivity among DCs, which is based upon the flexgrid technology. The number of users in each location was computed considering Wikipedia's audience by regions that was scaled and distributed among the different locations in each region. Latency was computed according to Verizon's data [Verizon]. Brown energy cost for each DC, servers' model and DC dimensioning parameters such as the number of clusters, number and characterization of VMs, as well as DC architecture and switches' models considered are detailed in Chapter 3.

It is worth remembering that an integer number of clusters is always switched on, to support the load assigned to the DC; those servers without assigned load remain active and ready to accommodate spikes in demand. Green cover was set to ensure, at the highest green energy generation time, a proportion of energy β_d when all VMs run in DC d.

Finally, a dynamic network environment was simulated for the scenario under study, where background incoming connection requests arrive following a Poisson process and are sequentially served without prior knowledge of future incoming connection requests. Background traffic competes with the one generated by the federated DCs for network resources.

Fig. 5-3 plots daily energy and communication costs as a function of the normalized background traffic intensity. We observe a clear increasing trend when the background traffic increases, as a consequence of connections' initial capacity decreases from 55 Gb/s to only 12 Gb/s on average. To try to increase that limited initial connections' capacity, elastic capacity increments need to be requested. The results obtained when each connectivity model is applied are however different. Both models behave the same when the background traffic intensity is low or high, which is as a consequence of the percentage of VMs that could not be migrated in the scheduled period (see Fig. 5-4 left). When the intensity is low, there are enough resources in the network so even in the case that elastic connection operations are requested, both models are able to perform scheduled VM migration in the required period. When the background intensity is high connections requests are rejected or are established with a reduced capacity that is unlikely modified. As a result, a high percentage of scheduled VM migrations could not be performed.

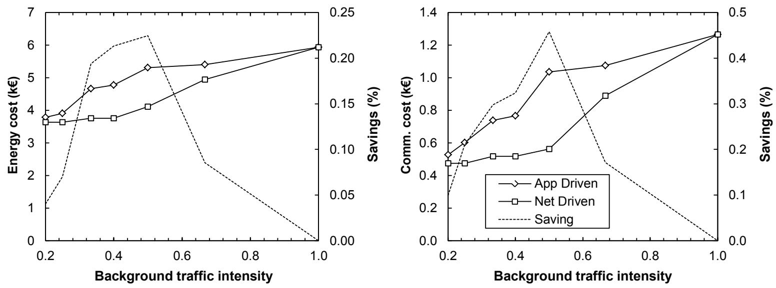

Fig. 5-3. Daily energy cost (left) and communication cost (right).

However, when the background load increases without exceeding 5% of total blocking probability, the behavior of the analyzed connectivity models is different; the proposed network-driven model provides a constant energy and communications costs until the normalized background load is greater than 0.4, in contrast to the remarkable cost increment provided using the application-driven model. In fact, costs savings as high as 20% and 40% in energy and communications, respectively are obtained when the network-driven model is applied with respect to those of the application-driven. When the normalized background load increases from 0.4, the lack of resources starts affecting also the network-driven model and, although costs savings reach their maximum for a load of 0.5, energy and communication costs start increasing and relative savings decreasing.

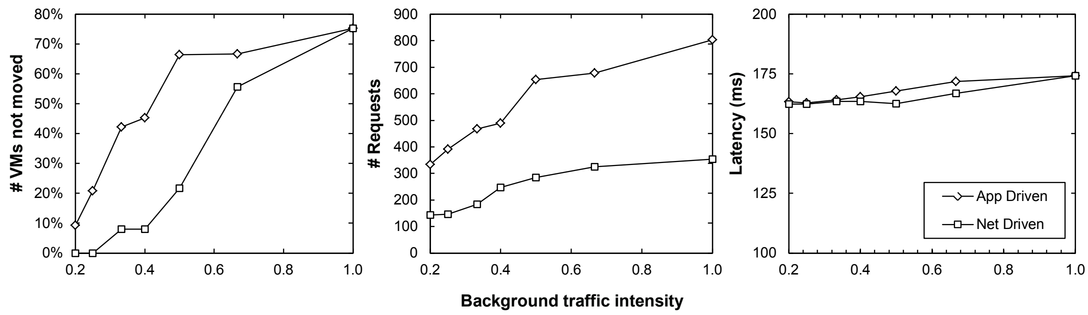

Fig. 5-4. Percentage of VMs not moved as first scheduled (left), number of connection requests (center), and latency experienced by users (right).

It is also interesting to see the total number of requests generated when each connectivity model is used. Fig. 5-4 (center) plots the amount of requests for set up, elastic capacity increment or decrement, and tear down that arrive to the ABNO controller. When the application-driven model is used, the number of requests is really high compared to that number under the network-driven model. However, since the requests are generated by the DC resource managers without any knowledge of the state of the resources, the majority of those requests are blocked as a result of lack of resources. Such high utilization of the network resources is the target for the network operator. In contrast, in the network-driven model, elastic capacity increment or decrement requests are generated by the carrier SDN, which knows that some resources in the route of established connections have been released and elastic capacity operation could be successfully applied. In this case, the amount of requests is much lower but many of them are successfully completed (although some few can be also blocked). Regarding latency, both models are able to provide similar performance, as shown in Fig. 5-4 (right). This fact, however, is as a result of the scheduler algorithm that focuses at guaranteeing the committed QoE.

Finally, Fig. 5-5 illustrates hourly variation in the energy and communications costs when the application-driven and the network-driven models are applied, for three different background traffic loads. The behavior of both models is basically the same and slight hourly energy cost savings can be appreciated, although they are clearly evident for the intermediate load. In contrast, there are some periods with a totally different behavior between application-driven and the network-driven models, especially in the intermediate load. That is as a consequence of that VMs can be placed in those locations so as to minimize cost in the network-driven model so no new migrations are required, whereas massive migrations need to be performed in the application-driven model, which further increases communications needs.

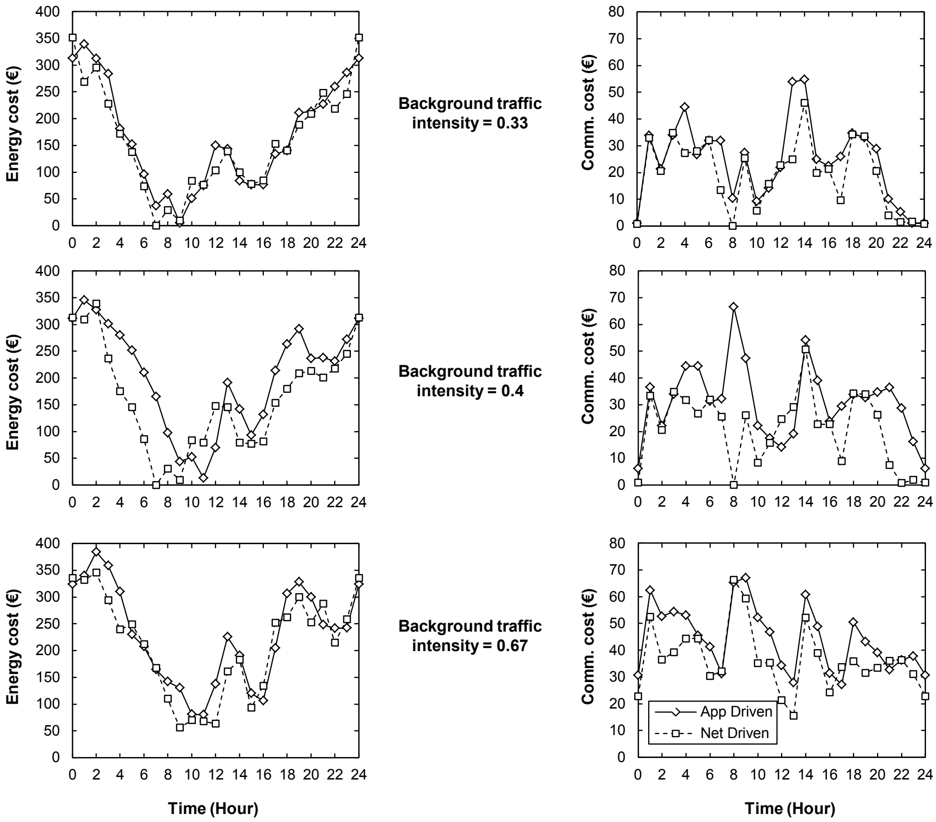

Fig. 5-5. Hourly costs for several background traffic intensities.

5.3 Conclusions

In line with the ELFADO problem presented in Chapter 3, a carrier SDN controller implementing a northbound interface with application-oriented semantic has been considered as a new abstraction layer between DC resource managers and the ABNO controller in the control plane of flexgrid-based interconnection networks. Each resource manager can request transfer operations specifying the destination DC, the amount of data to be transferred and the desired completion time.

The above connectivity model, named network-driven, has been compared against the dynamic elastic model or application-driven, where the local resource managers are in charge of requesting connections directly to the ABNO controller. The application-driven model needs periodical retries requesting increase connection's bitrate, which do not translate into immediate bitrate increments and could have a negative impact on the performance of the network control plane.

Energy and communications costs and QoE on a DC federation were analyzed. Some green energy is available in each of the locations as a function of the time, whilst the cost of brown energy shows differentiated on/off peak costs. A federation orchestrator computes periodically the global optimal placement for all the VMs in the federation so as to minimize operational costs whilst ensuring QoE.

From the results, we observed that when the network operates under low and medium traffic load costs savings as high as 20% and 40% in energy and communications, respectively can be obtained when the network-driven model is applied with respect to those of the application-driven. Besides, both connectivity models allow scheduling algorithms to provide the committed QoE.

Chapter 6

Closing discussion

6.1 Main contributions

This section summarizes the main contributions and conclusions of this thesis.

We reviewed the state of the art for energy expenditures minimization in DCs and the inter-DC connectivity models. We concluded that, to the best of our knowledge, no work compared the way to compute scheduling for VM migration considering both energy and communications costs in a single framework.

Then we proposed the Elastic Operations in Federated Datacenters for Performance and Cost Optimization problem for scheduling workload and orchestrating federated DCs, taking into account the solar energy available in DCs. Two approaches were proposed and evaluated: distributed and centralized. Results showed that both ELFADO approaches improve QoE by reducing average latency more than 10% with respect to a fixed approach where no scheduling is performed. The centralized approach showed remarkable energy cost savings about 52%, resulting in 44% when communication costs were considered. VMs were placed in DCs so as to take full advantage from green energy availability, since the placement of all VMs was computed at the same time in the proposed federation orchestrator in the centralized approach.

After that, we presented an architecture for a distributed cross-stratum orchestrator to coordinate the cloud and network, allowing elastic DC operations to be performed as well as the dynamic establishment and tear down of inter-DC connections. Furthermore, we proposed to perform elastic operations on already established connections allowing the CSO to retry requests trying to increase connection's capacity when not enough resources are available at a connection's set up. We considered flexgrid-based interconnection networks, since elastic connections are supported by that technology. Results showed that dynamic connectivity could entail bitrate savings as high as 60 percent with respect to static connectivity and that dynamic elastic connectivity improves the performance of the dynamic one in scenarios where the physical optical network is shared by several services.

In addition, carrier SDN was proposed in a DC federation scenario. A carrier SDN controller implementing a northbound interface with application-oriented semantic was described. The dynamic elastic model (application-driven) was compared to the transfer mode model (network-driven), which takes advantage from the use of notify messages, thus being able to reduce time-to-transfer remarkably and opens the opportunity to network operators to implement policies so as to dynamically manage connections' bitrate of a set of customers and fulfill simultaneously their SLAs.

Finally, energy and communications costs and QoE on a DC federation were analyzed consireding: a federation orchestrator computing periodically the global optimal placement for all the VMs in the federation so as to minimize operational costs whilst ensuring QoE, carrier SDN and both application-driven and network-driven connectivity models.

From the results, we observed that when the network operates under low and medium traffic load costs savings as high as 20% and 40% in energy and communications, respectively were obtained when the network-driven model was applied with respect to those of the application-driven. Besides, both connectivity models allow scheduling algorithms to provide the committed QoE.

6.2 Publications

6.2.1 Journals and magazines

- L. Velasco, **A. Asensio**, J.Ll. Berral, V. López, D. Carrera, A. Castro, and J.P. Fernández-Palacios, "Cross-Stratum Orchestration and Flexgrid Optical Networks for Datacenter Federations," IEEE Network Magazine, vol. 27, pp. 23-30, 2013.
- L. Velasco, **A. Asensio**, J.Ll. Berral, A. Castro, V. López, "Towards a Carrier SDN: An example for Elastic Inter-Datacenter Connectivity," OSA Optics Express, vol. 22, pp. 55-61, 2014.
- L. Velasco, **A. Asensio**, J. Ll. Berral, E. Bonetto, F. Musumeci, V. López, "Elastic Operations in Federated Datacenters for Performance and Cost Optimization," accepted in Elsevier Computer Communications, 2014.

6.2.2 Conferences and workshops

- **A. Asensio**, A. Castro, L. Velasco and J. Comellas, "An Elastic Networks OMNeT++ -based Simulator," in Proc. IEEE 15th International Conference on Transparent Optical Networks (ICTON), 2013.
- **A. Asensio**, M. Klinkowski, M. Ruiz, V. López, A. Castro, L. Velasco, J. Comellas, "Impact of Aggregation Level on the Performance of Dynamic Lightpath Adaptation under Time-Varying Traffic," in Proc. IEEE 17th International Conference on Optical Network Design and Modeling (ONDM), 2013.
- **A. Asensio**, L. Velasco, M. Ruiz, and G. Junyent, "Carrier SDN to Control Flexgrid-based Inter-Datacenter Connectivity," in Proc. IEEE 18th International Conference on Optical Network Design and Modeling (ONDM), 2014.

6.3 Future work

The results shown in this work motivate further study for the improvent of future flexgrid-based optical networks and cloud management and control architectures. Transfer mode requests and carrier SDN for controlling inter-DC connections, open issues to extend this work while exploring other functionalities.

My interest for participating in future projects within the GCO research group brings me the opportunity to continue developing my research work with the intention to work towards the Ph.D. degree.

Among the topics of research that are in relation with this thesis, stand out the following:

- Improving the weather estimation in ELFADO problem for predicting green energy availability in DCs using a statistical approach instead of only using historical weather information.
- Exploting transfer mode requests, carrier SDN and ABNO-based control plane architecture to solve not only the routing and spectrum allocation problem but also scheduling resources on demand so to fulfill the committed SLAs of several customers while improving network performance
- Exploring new scenarios where DCs require other types of connectivity, for example for UltraHD television distribution.

List of Acronyms

ABNO	Application-Based Network Operations
BV-OXC	Bandwidth-Variable Optical Cross-Connect
BV-WSS	Bandwidth-Variable Wavelength Selective Switches
CSO	Cross-Stratum Orchestrator
DB	Database
DC	Datacenter
DC2DC	Datacenter-to-datacenter traffic
ELFADO	Elastic Operations in Federated Datacenters for Performance and Cost Optimization
EON	Elastic Optical Network
GbE	Gigabit Ethernet
IETF	Internet Engineering Task Force
IP	Internet Protocol
IT	Information Technology
k-PSK	k- Phase-Shift Keying
k-QAM	k- Quadrature Amplitude Modulation
LSP	Label Switched Path
LSP-DB	Label Switched Path Database
MILP	Mixed Integer Linear Programming
MPLS	Multiprotocol Label Switching
O-OFDM	Optical Orthogonal Frequency Division Multiplexing
OPEX	OPerational EXpenditures
PCE	Path Computation Element
PCEP	Path Computation Element Protocol

PM	Physical Machine
PUE	Power Usage Effectiveness
QoE	Quality of Experience
QoS	Quality of Service
SBVT	Sliceable Bandwidth-Variable Transponder
SDN	Sofware Defined Network
SLA	Service Level Agreement
TCP	Transmission Control Protocol
TED	Traffic Engineering Database
U2DC	User-to-Datacenter traffic
VM	Virtual Machine

References

[Ar10] M. Armbrust et al., “A View of Cloud Computing,” Communications of the ACM, vol. 53, pp. 50-58, 2010.

[As13] A. Asensio et al., “Impact of Aggregation Level on the Performance of Dynamic Lightpath Adaptation under Time-Varying Traffic,” in Proc. IEEE International Conference on Optical Network Design and Modeling (ONDM), 2013.

[As14] A. Asensio, L. Velasco, M. Ruiz, and G. Junyent, “Carrier SDN to Control Flexgrid-based Inter-Datacenter Connectivity,” in Proc. IEEE International Conference on Optical Network Design and Modeling (ONDM), 2014.

[Aw01] D. Awduche, et al. “RSVP-TE: Extensions to RSVP for LSP Tunnels,” IETF RFC 3209, 2001

[Bu13] J. Buysse et al., “Energy-Efficient Resource-Provisioning Algorithms for Optical Clouds,” IEEE/OSA Journal of Optical Communications and Networking, vol. 5, pp. 226-239, 2013.

[Ca12] A. Castro et al., “Dynamic Routing and Spectrum (Re)Allocation in Future Flexgrid Optical Networks,” Elsevier Computer Networks, vol. 56, pp. 2869–83, 2012.

[Ch11] K. Christodoulopoulos et al., “Dynamic bandwidth allocation in flexible OFDM-based networks,” in Proc. OFC 2011.

[Ch13] K. Christodoulopoulos et al., “Time-Varying Spectrum Allocation Policies in Flexible Optical Networks,” IEEE Journal on Selected Areas in Communications, vol. 31, pp. 13-25, 2013.

[CISCO] Cisco, Global Cloud Index, 2012.

[Co12] L. Contreras et al., “Toward Cloud-Ready Transport Networks,” IEEE Communications Magazine, vol. 50, pp. 48–55, 2012.

[CPLEX] CPLEX, http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/.

[Cr13] E. Crabbe, J. Medved, R. Varga, and I. Minei, “PCEP Extensions for Stateful PCE,” IETF draft, 2013.

[Cu12] F. Cugini et al., “Demonstration of flexible optical network based on path computation element,” Journal of Lightwave Technology, vol. 30, no. 5, pp. 727-733, 2012.

[Di02] J. Díaz and E. Fernández, “A Branch-and-Price Algorithm for the Single Source Capacitated Plant Location Problem,” Journal of the Operational Research Society (JORS), vol. 53, pp. 728-740, 2002.

[EUEP] Europe’s Energy Portal: http://www.energy.eu/

[Fa07] X. Fan, W. Weber, and L. A. Barroso, “Power Provisioning for a Warehouse-sized Computer”, In Proc. of ACM International Symposium on Computer Architecture (ISCA), 2007.

[Fa08] M. Al-Fares, A. Loukissas, and A. Vahdat, “A scalable, commodity data center network architecture,” in Proc. ACM SIGCOMM, 2008.

[Ge11] D. J. Geisler et al., “The first testbed bemonstration of a flexible bandwidth network with a real-time adaptive control plane,” in Proc. of ECOC, Geneva, Switzerland, Sep. 2011.

[Ge12] O. Gerstel et al., “Elastic optical networking: A new dawn for the optical layer?,” IEEE Communications Magazine, vol. 50, no. 2, 2012.

[Go10.1] I. Goiri, J. Guitart, and J. Torres, “Characterizing Cloud Federation for Enhancing Providers’ Profit,” in Proc. IEEE International Conference on Cloud Computing, 2010.

[Go10.2] I. Goiri et al., “Energy-aware Scheduling in Virtualized Datacenters,” in Proc. IEEE International Conference on Cluster Computing, 2010.

[Go12] I. Goiri, et al. “GreenHadoop: Leveraging Green Energy in Data-Processing Frameworks,” in Proc EuroSys, 2012.

[GreenGrid] The Green Grid: www.thegreengrid.org/

[HP] Hewlett-Packard, http://www.hp.com/

[HUAWEI] Huawei, http://www.huawei.com

[Ji09] M. Jinno et al., “Spectrum-efficient and scalable elastic optical path network: Architecture, benefits, and enabling technologies,” IEEE Communications Magazine, vol. 47, no. 11, 2009.

[Ji12] M. Jinno et al., “Multiflow Optical Transponder for Efficient Multilayer Optical Networking,” IEEE Communications Magazine, vol. 50, no. 5, 2012.

[JZh13] J. Zhang, et al. “First demonstration of enhanced Software Defined Networking (eSDN) over elastic Grid (eGrid) Optical Networks for Datacenter Service Migration,” in Proc. OSA OFC PDP5B.1, 2013.

[Ki04] D. King, W. Boyson and J. Kratochvil, “Photovoltaic Array Performance Model,” Sandia National Laboratories Report, SAND2004-3535, 2004.

[Ki13] D. King and A. Farrel, “A PCE-based Architecture for Application-based Network Operations,” IETF draft, 2013.

[Kl13] M. Klinkowski et al., "Elastic spectrum allocation for time-varying traffic in flexgrid optical networks," IEEE J. Sel. Areas Commun., vol. 31, pp. 26-38, 2013.

[Le09] Y. Lee, et al., "Path Computation Element Communication Protocol (PCEP) Requirements and Protocol Extensions in Support of Global Concurrent Optimization," IETF RFC 5557, 2009.

[LLi09] L. Liu et al., "GreenCloud: A New Architecture for Green Data Center," in Proc. ICAC-INDST, 2009.

[MetaWiki] Meta-Wiki. http://meta.wikimedia.org/wiki/User:Stu/comScore_data_on_Wikimedia#Geographic_breakdown.

[Mi12] M. Mishra et al., "Dynamic Resource Management Using Virtual Machine Migrations," IEEE Communications Magazine, vol. 50, pp. 34-40, 2012.

[OpNeb] OpenNebula, http://www.opennebula.org/

[Pi11] J.M. Pierson, "Green Task Allocation: Taking into Account the Ecological Impact of Task Allocation in Clusters and Clouds," Journal of Green Engineering, vol. 1, pp. 129–144, 2011.

[Sh10] N. Sharma et al., "Cloudy Computing: Leveraging Weather Forecasts in Energy Harvesting Sensor Systems," in Proc. SECON, 2010.

[Sh11] G. Shen et al., "Maximizing time-dependent spectrum sharing between neighbouring channels in CO-OFDM optical networks," in Proc. ICTON 2011.

[US07] US Environmental Protection Agency, "Report to Congress on Server and Data Center Energy Efficiency," 2007.

[USBLS] U.S. Department of Labor, Average Energy Prices, http://www.bls.gov/ro5/aepchi.htm

[USEERE] US Department of Energy, http://apps1.eere.energy.gov/buildings/energyplus/weatherdata_about.cfm

[USEIA] U.S. Department of Energy. US Energy Information Administration. 2009. Web site http://www.eia.doe.gov/.

[Va09] JP. Vasseur and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)," IETF RFC 5440, 2009

[Ve12] L. Velasco, et al., "Elastic Spectrum Allocation for Variable Traffic in Flexible-Grid Optical Networks," in Proc. OFC 2012.

[Ve13] L. Velasco, et al., "Cross-Stratum Orchestration and Flexgrid Optical Networks for Datacenter Federations," IEEE Network Magazine, vol. 27, pp. 23-30, 2013.

[Ve14.1] L. Velasco, at al., "Towards a Carrier SDN: An example for Elastic Inter-Datacenter Connectivity," OSA Optics Express, vol. 22, pp. 55-61, 2014.

[Ve14.2] L. Velasco, et al., “Elastic Operations in Federated Datacenters for Performance and Cost Optimization,” to appear in Elsevier Computer Communications, 2014.

[Ve14.3] L. Velasco, et al., “Solving Routing and Spectrum Allocation Related Optimization Problems: from Off-Line to In-Operation Flexgrid Network Planning,” (Invited Paper) to appear in IEEE/OSA Journal of Lightwave Technology (JLT), 2014.

[Ve14.4] L. Velasco, et al., “In-Operation Network Planning,” IEEE Communications Magazine, vol. 52, pp. 52-60, 2014.

[Verizon] Verizon, http://www.verizonbusiness.com/about/network/latency/

[WiSize] Wikipedia, http://en.wikipedia.org/wiki/Wikipedia_talk:Size_of_Wikipedia#size_in_GB

[XZh13] X. Zhao et al., “The Prospect of Inter-Data-Center Optical Networks,” IEEE Communications Magazine, vol. 51, pp. 32-38, 2013.

[YZh13] Y. Zhang, and N. Ansari, “On Architecture Design, Congestion Notification, TCP Incast and Power Consumption in Data Centers,” IEEE Communications Surveys & Tutorials, vol. 15, pp. 39-64, 2013.

[ZAl13] Z. Ali et al., “Path Computation Element Communication Protocol (PCEP) Extensions for Remote-Initiated GMPLS LSP Setup,” IETF draft, 2013.

[ZLi11] Z. Liu et al., “Geographical Load Balancing with Renewables,” in Proc. of ACM Greenmetrics, 2011.