

# Multi-Tenant Software-Defined Hybrid Optical Switched Data Centre

Shuping Peng, Bingli Guo, Chris Jackson, Reza Nejabati, Fernando Agraz, Salvatore Spadaro, Giacomo Bernini, Nicola Ciulli, and Dimitra Simeonidou

**Abstract**—We introduce a holistic solution for software-defined optical data centres (DC). Hybrid optical circuit/packet switching technologies are employed in the data plane, while a software-defined networking (SDN) controller based on OpenDaylight with significant extensions is adopted for the data centre network (DCN) control and management. Novel functional modules in the SDN controller together with its northbound (NBI) and southbound interfaces (SBI) are designed and developed. The OpenFlow protocol is extended at the SBI to support communication between the extended OpenDaylight SDN controller and the optical DCN devices. Over the NBIs, dc applications and the Cloud management system directly interact with the optical DCN. A virtual data centre (VDC) application is designed and developed that dynamically creates and provisions multiple coexisting but isolated VDCs. An optical network-aware virtual machine (VM) placement method is proposed and implemented for a single-step deployment of both network and IT (VM) resources to accommodate the VDC requests. The VDC deployment process is extensively simulated and experimentally demonstrated.

**Index Terms**—Data centre virtualisation, multi-tenancy, optical data centre network (DCN), software-defined networking (SDN), virtual data centre (VDC).

## I. INTRODUCTION

THE role of data centres (DCs) is increasingly vital in the ICT eco-system [1], [2]. A huge amount of data is being maintained and processed in DCs, on the order of TeraBytes or even PetaBytes per DC. DC providers have observed over 70% annual increase in the DC traffic volume, of which about 50%–75% is east-to-west traffic across DC networks (DCNs) [3]. Current DCN infrastructures built with massive electronic switches in tree-like hierarchical topologies [4], [5] have already been stretched by the ever-increasing demand from new and emerging cloud applications. These infrastructures also suffer from inherent disadvantages such as limited bisectional bandwidth, limited scalability, and high power consumption [6]. In order to accommodate the enormous data volume and new traffic patterns with low power consumption, DCs are moving towards

new architectures and solutions, exploring the use of optical technologies.

A plethora of solutions have been proposed to bring optical switching technologies into DCs in recent years, e.g. Helios [7] and C-through [8]. However, they are not able to deal with diverse applications using all-optical solutions. A flat optical DCN architecture empowered by Architecture-on-Demand (AoD) node [9] is proposed in [10], where a combination of optical packet switching (OPS) and optical circuit switching (OCS) technologies is adopted to dynamically accommodate the various traffic flows in DCs. The new optical DCN architecture is able to provide optical connectivity services with high throughput and low latency [11], [12].

Such an advanced optical DCN has also imposed great challenges in the DC control and management, requiring a novel framework that is able to manipulate the underlying heterogeneous optical devices and provide programmability, flexibility, and high availability of DCN connectivity services to meet the highly dynamic requirements imposed by new and emerging DC applications. The current DC control and management platforms are mostly based on rigid and slow human-driven resource provisioning procedures, resulting in a costly operation of DC infrastructure. In [13], a software-defined networking (SDN) [13] control platform is developed, which is based on the OpenDaylight (ODL) controller [15] but with some key functional modules (e.g., Forwarding Rules Manager) extended to support the control of optical network devices. OpenFlow (OF) [13] agents are built for the non-OpenFlow enabled devices, while the OpenFlow protocol is extended to enable communication between the SDN controller and DCN devices. This SDN control platform enables on-demand, flexible, and programmable optical DCN connectivity services [13].

Driven by a growing need to support cloud-based services and applications that must be provided to a wide plethora of business customers/users (e.g., 78% of all UK business have at least one cloud-based service [16]), *multi-tenancy* has become a key requirement of the next generation DCs [17]. With multi-tenancy, DC providers can efficiently multiplex customers across their physical infrastructure, while customers no longer need to buy, manage and maintain a physical DC infrastructure at their own premises. This reduces Capital Expenditure (CAPEX) and Operational Expenditure (OPEX) of both customers and DC providers. Virtualisation is the key technical enabler for the multi-tenancy in cloud DCs [18], [19].

Today, thanks to *Server Virtualisation* [20], DCN infrastructure is surrounded by clusters of extensively virtualised servers. Virtual machines (VMs) can be easily instantiated (in minutes) and run on servers that in the past were dedicated to a single

Manuscript received February 2, 2015; revised May 20, 2015; accepted May 26, 2015. The work was supported by EU funded FP7 projects LIGHTNESS (no. 318606) and COSIGN (no. 619572).

S. Peng, B. Guo, C. Jackson, R. Nejabati, and D. Simeonidou are with the High Performance Networks Group, University of Bristol, Bristol BS8 1TH, U.K. (e-mail: shuping.peng@bristol.ac.uk; allen.guo@bristol.ac.uk; cscrj@bristol.ac.uk; reza.nejabati@bristol.ac.uk; Dimitra.Simeonidou@bristol.ac.uk).

F. Agraz and S. Spadaro are with the Universitat Politècnica de Catalunya, 08034 Barcelona, Spain (e-mail: agraz@tsc.upc.edu; spadaro@tsc.upc.edu).

G. Bernini and N. Ciulli are with the Nextworks, 56122 Pisa, Italy (e-mail: g.bernini@nextworks.it; n.ciulli@nextworks.it).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JLT.2015.2438398

90 application/tenant, therefore building a multi-tenant cloud environment. However, the VMs must wait hours or days for dedicated network connectivity services to be established between them due to the multiple levels of static and human-driven configurations that must be operated. In this context, the physical DCN that interconnects storage and compute resources (i.e., IT resources) must be operated in coordination with the applications running on the IT resources in order to deliver cloud-based applications more efficiently in the multi-tenant cloud environment.

100 In this paper, dynamic virtual data centre (VDC) deployment is investigated to enable the multi-tenancy in hybrid optical switched DCs. The proposed method, *Network-aware VM Placement*, takes into account both the optical DCN characteristics (e.g., wavelength continuity constraints) and VM dynamics. The method is seamlessly plugged into the OpenDaylight SDN control platform [15] as an advanced application, which also interacts with the OpenStack cloud management system [29] for coordinated and dynamic deployment of VDC requests. The virtual DCN Manager module is developed in the SDN controller to handle the deployment of the virtual DCN requests through interaction with other functional modules.

112 The rest of this paper is organized as follows. The software-defined optical DC architecture is described in Section II, including the hybrid optical switched DCN and the SDN solution, including the specification of the southbound and northbound interfaces (SBI/NBI). In Section III, the design and implementation of the dynamic VDC deployment using the network-aware VM placement method is introduced. The simulation studies and experimental demonstrations are discussed in Section IV and Section V, respectively. Finally, Section VI concludes the paper.

## 122 II. SOFTWARE-DEFINED OPTICAL DATA CENTRE

123 The proposed software-defined optical DC architecture, as shown in Fig. 1, aims at addressing the requirements resulting from new and emerging DC applications. These requirements are mainly high bandwidth, low latency, flexibility, scalability, and programmability.

128 The overall architecture includes an advanced optical data plane based on the AoD node as well as a comprehensive SDN control plane with northbound Application Programming Interfaces (APIs) and SBI (e.g., extended OF protocol), and OF agents for heterogeneous network devices, which will be described in details in the following sections, respectively.

### 134 A. Hybrid Optical Switched Data Centre Network

135 In DCs, applications that generate long-lived (elephant) data flows coexist with those that exchange short-lived (mice) flows with tight latency requirements [22]. To cope with the different type of data flows and satisfy their QoS requirements, the proposed optical DCN relies on a flat architecture that integrates OPS and OCS technologies. The adoption of OPS and OCS technologies enables us to avoid expensive Optical-Electrical-Optical conversions and optical transceivers, reducing the energy consumption and cost compared to contemporary electrical

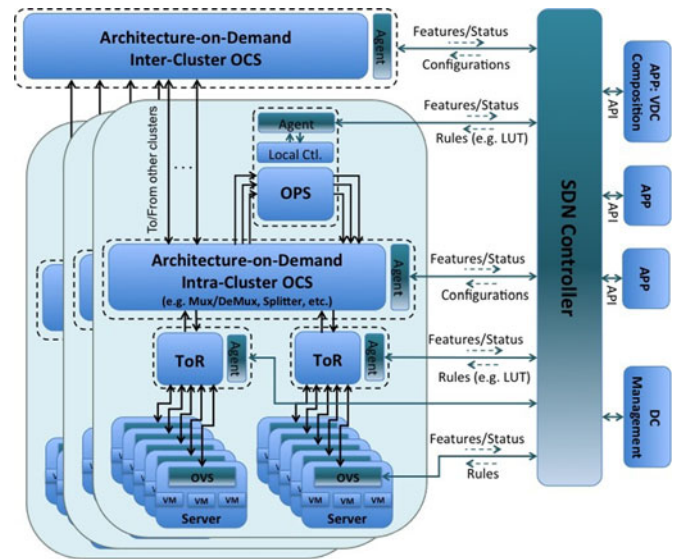


Fig. 1. Overall architecture of the software-defined optical DC (OVS: Open vSwitch [21], APP: Application).

144 solutions. A modular OPS design [10], [31] with highly distributed control for port-count independent reconfiguration time is employed. This enables high load operation with *sub- $\mu$ s* latency while having the potential to scale to over a thousand ports [31] if significant progress on cost-effective and robust OPS technology platform is made. Racks/servers are interconnected to the hybrid OPS/OCS DCN via Top-of-the-Rack (ToR) switches [10] that perform traffic aggregation and application-aware classification to either short- or long-lived data flows [23], [24]. The OCS technology can better accommodate long-lived high-capacity smooth data flows, while OPS can offer more flexible bandwidth capacity for each optical link when facing dynamic and unpredictable traffic demands with either short- or long-lived flows [25].

158 Instead of hardwired interconnections of the main network elements (i.e., OCS, OPS, and ToR switches), the designed optical DCN adopts AoD nodes for constructing both intra- and inter-Cluster connectivity, as shown in Fig. 1. The AoD node consists of an optical backplane, i.e., an optical high-radix fibre switch (e.g., Polatis Series 6000,  $192 \times 192$  ports [26]) with passive/active optical network elements (e.g., Mux/DeMux, Splitter, and Coupler) and switching modules (i.e., OPS and ToR switches) connected to it. With this AoD-based DCN architecture, various arrangements of inputs, elements/modules, and outputs can be adaptively constructed by dynamically provisioning appropriate cross-connections in the optical backplane to meet the requirements of different applications. The combination of hybrid optical switching technologies makes it possible to switch traffic in space, frequency, and time. This enhances the capability to efficiently handle both short- and long-lived traffic flows.

### 175 B. SDN Solution for the Hybrid Optical DC Network

176 For effective control and management of the underlying heterogeneous DCN devices, maintaining direct interaction with



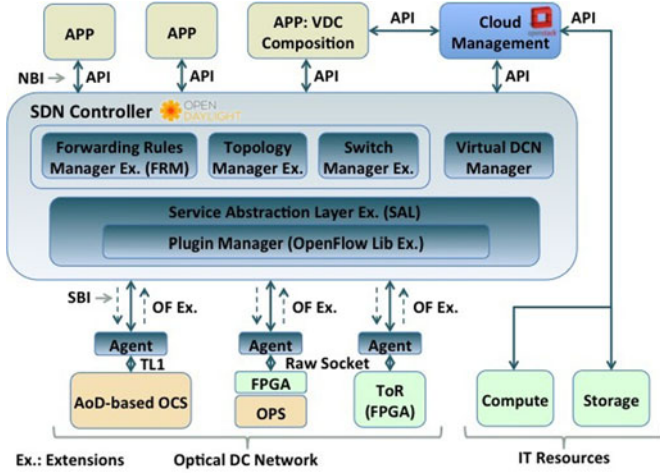


Fig. 2. SDN Solution for the Hybrid Optical DC Network.

DC applications, an SDN-based control plane equipped with both southbound and northbound interfaces is developed for the hybrid optical DCN, as shown in Fig. 2.

1) *SDN Controller Architecture and Its Key Modules*: The control intelligence is logically centralized in an SDN controller that maintains a full and detailed view of the DCN at both intra- and inter-cluster levels. The SDN controller functional architecture is based on ODL. In order to support the DCN devices, we have extended several key modules in the ODL SDN controller, including the Topology Manager, Switch Manager, Forwarding Rules Manager, Service Abstraction Layer, and Plugin Manager with an extended OpenFlow Library. The Virtual DCN Manager, responsible for the optical DCN virtualization, has also been developed, and is described in further detail in the next section.

The peer information of optical switch ports is parsed in the extended Topology Manager, which is used by ODL to build the optical DCN topology. The power level at the optical ports is collected by the ODL controller from optical switches via the OF agents, by exchanging the extended OF message *of\_port\_statistics*. Flow entry has been extended to specify optical flows [27]. The Forwarding Rules Manager is extended to construct optical flows and push them to the corresponding SBI. In the Service Abstraction Layer, new data structures have been added to represent the attributes of optical devices. With the OpenFlow Library, extended OF messages are exchanged between the controller and DCN devices. These extensions will be described in detail in the following subsection. The Plugin Manager is not limited to OF and can support multiple protocols as southbound interfaces.

2) *Southbound and Northbound Interfaces*: Unified communication between the SDN controller and the optical DCN fabric is enabled by the OF agents, which are built for each of the DCN devices, i.e., AoD backplane, OPS, and ToR switches, as shown in Fig. 2. This is to abstract the attributes of the heterogeneous DCN devices and convert the OF protocol to their own technology-specific protocols, such as TL1 and Raw Ethernet.

The OF protocol has been extended to support the optical devices. Fig. 3 shows the implemented key OF extensions in

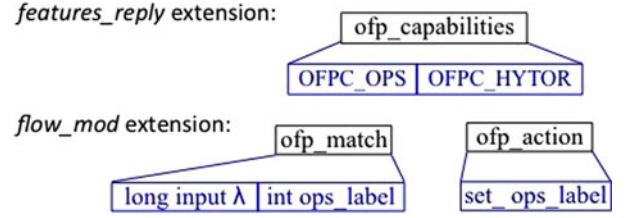


Fig. 3. OpenFlow Protocol Extensions.

the fields of *ofp\_capability*, *ofp\_match*, and *ofp\_action*. Specifically, the *ofp\_match* is extended for OPS to support the use of wavelength and label of an optical packet, while the *ofp\_action* is extended to enable the optical packet label configurations in the FPGAs for controlling OPS fabric [28].

Through the northbound interface (i.e., a set of open APIs), the SDN controller interacts with DC applications and the DC management system (e.g., OpenStack [29]). The northbound interface exposes the network functions implemented by the SDN controller to applications, so that it opens opportunities to implement advanced functionalities as DC applications running on top of the SDN controller, such as the VDC Composition that enables multi-tenancy in optical cloud environments. Moreover, using the northbound interface, the DC cloud management system is also able to perform a wide set of monitoring actions [29].

This SDN-based control provides a generalized and modular platform that can be extended to support future network technologies and devices. The adoption and support of open APIs at the northbound and standard interfaces at the southbound (with proper extensions where needed), allows DC operators (or customers) to implement network functions and applications according to their specific requirements and needs.

### III. DYNAMIC VIRTUAL OPTICAL DATA CENTRE NETWORK DEPLOYMENT WITH NETWORK-AWARE VM PLACEMENT

This study deals with dynamic VDC deployment [30], that is, VDC requests arrive dynamically and are processed on demand for their provisioning. Multiple coexisting but isolated VDCs are deployed taking into account the characteristics and availability of both optical DCN and IT resources (mostly bandwidth and VMs).

#### A. Virtual Data Centre Request

A VDC request (see Fig. 4) comprises a set of virtual nodes interconnected by virtual links. Each virtual node is a VM pool with a number of VMs, each specifying a number of CPU cores and some memory. Each virtual link has constraints on bandwidth and/or latency, etc. The VM pools are interconnected in a certain network topology to accommodate dc application requirements such as distributed computing and file replication.

As shown in Fig. 4, when a DC is up and running, VDC requests arrive dynamically; here, we focus on processing VDC requests sequentially. However, the situation where multiple VDC requests arrive at the same time is not excluded. The whole SDN-based framework still supports and can be extended

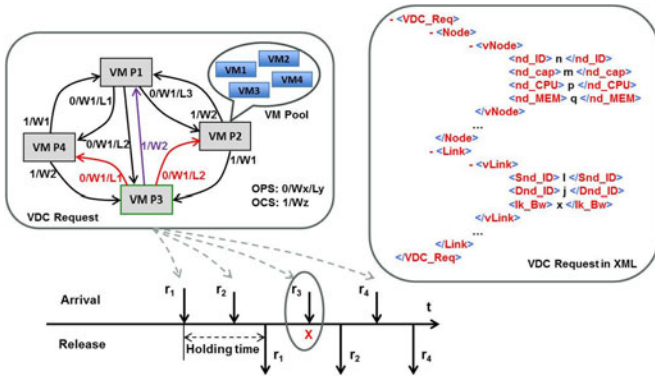


Fig. 4. Dynamic VDC requests.

261 to handle concurrent VDC requests. Each VDC request has its  
 262 own holding time (lifetime), and if the request is successfully  
 263 deployed, when its holding time expires (see the request  $r1$  in  
 264 Fig. 4), the corresponding resources (network and VMs) are re-  
 265 leased for future use. VDC requests may also be rejected (see  
 266 the request  $r3$  in Fig. 4), for example due to a lack of avail-  
 267 able resources or optical layer constraints such as wavelength  
 268 continuity and/or impairments in optical DCN.

### 269 B. Virtual Data Centre Deployment in Optical DCN

270 To deploy a requested VDC, the pool of VMs per virtual node  
 271 needs to be embedded into a rack (of servers) that has enough  
 272 available resources, while the virtual links will be accommod-  
 273 ated by optical lightpaths provisioned in the optical DCN. In  
 274 order to compose multiple coexisting but isolated VDCs to en-  
 275 able the multi-tenancy in optical DC, while taking advantage of  
 276 the benefits brought by optical technologies, the characteristics  
 277 of underlying heterogeneous optical devices need to be carefully  
 278 considered in the virtualisation process.

279 1) *Virtual Optical DCNs with Hybrid Granularities*: Lever-  
 280 aging the hybrid optical switching technologies in the optical  
 281 DC, we are able to achieve hybrid granularities of VDCs: Macro  
 282 and Micro VDCs, which can be realised by a combination of  
 283 different technologies.

284 The *Macro VDC* is a relatively stable VDC slice provid-  
 285 ing coarse granularity. In the optical DC with hybrid OPS and  
 286 OCS technologies, the Macro VDCs can be composed by using  
 287 pre-established optical circuits. This kind of VDC requires less  
 288 frequent reconfiguration, which perfectly aligns with the rela-  
 289 tively long switching time of optical circuit switches (e.g., Max  
 290 25 ms of the Polatis 6000).

291 The *Micro VDC*, providing finer granularities, is more flexible  
 292 and complementary to the Macro VDC. Multiple Micro VDCs  
 293 can be deployed on top of one Macro VDC. The statistical  
 294 multiplexing capability provided by OPS can be exploited in  
 295 composing the Micro VDCs.

296 The AoD-based hybrid optical DC design together with its  
 297 SDN solution facilitates the flexibility of composing the Macro  
 298 and Micro VDCs and achieving a rich set of granularities of  
 299 VDCs.

### Algorithm I. VM Placement based on variance calculation

for each VDC request

- 1 **READ** capacity requirement ( $vC_i$ ) of virtual nodes ( $VMP_i$ );
- 2 **READ** available capacity ( $C_j$ ) in the racks ( $R_j$ );
- 3  $COMB = \text{COMBNTNS}(\{j|R_j\}, VMP\_num)$ ;
- 4  $COMB\_NUM = \text{NCHOOSEK}(\{R\_num, VMP\_num\})$ ;
- 5 for  $k = 1 : COMB\_NUM$
- 6 **COMPARE** the **capacity** of  $VMP$  and  $COMB(k, :)$ ;
- 7 if the racks  $\in COMB(k, :)$  are able to support  $VMP$
- 8 **RECORD**  $COMB\_POSS = COMB(k, :)$ ;
- 9 **CALCULATE**
- 10  $resC_m = C_m - vC_m$  if  $m \in COMB(k, :)$
- 11  $resC_m = C_m$  otherwise
- 12  $Mean = \text{SUM}\{resC_m, \text{all } m\}$ ;
- 13  $VAR\_VM = \text{SUM}\{(resC_m - Mean)^2/R\_num, \text{all } m\}$ ;
- 14 **SORT**  $VAR\_VM$  in a descending order;
- 15 **GET** the corresponding rack no.  $R\_no$  of sorted  $VAR\_VM$ ;

2) *Network-Aware VM placement*: The VDC requests need  
 to be properly embedded in the physical DC infrastructure. The  
 actual location of the physical resources to support each VDC  
 request is not the concern of the VDC requester or its users.  
 However, the VDC composition process (including VM place-  
 ment and virtual link embedding) and the final embedded lo-  
 cations will directly affect the overall DC infrastructure utilis-  
 ation, VDC operations, and the isolation among VDCs. Energy  
 consumption is another metric, but is outside of the scope of  
 this study. In our work, we develop a coordinative method for  
 the VM placement and virtual link embedding, which is called  
*network-aware VM placement*. In this subsection, we elaborate  
 on the VM placement part (as shown in *Algorithm I*), while the  
 overall VDC composition process will be explained in detail in  
 the next section.

When a VDC request arrives, the requested VM capacity will  
 be read (Line1 (L1)) as well as the current available capacity in  
 the racks (L2). Here the capacity is defined as the number of VM  
 images (each has an initial amount of CPU and Memory capaci-  
 ties). L3 is used to generate a matrix where each row represents  
 a possible combination of the racks for embedding the requested  
 virtual nodes. Since each virtual node represents a pool of VMs  
 ( $VMP$ ), we assume that one (but only one) virtual node is em-  
 bedded into one rack.  $COMB\_NUM$  is the number of all the  
 possible combinations of racks (L4). For each possible combi-  
 nation of racks, we compare the available capacity in each rack  
 and the requested capacity (L5–6). Here the bandwidth capacity  
 of the ToR of each relevant rack is also checked. This is where  
 the network-aware VM placement mechanism resides, which  
 will interact with the VDC network embedding process (elabo-  
 rated in the following subsection). If the requested virtual nodes  
 can be supported (L7), the combination of racks  $COMB\_POSS$   
 will be recorded (L8), and the variance  $VAR\_VM$  of the residual  
 VM capacity  $resC_m$  of all the racks will be calculated (L9–13).



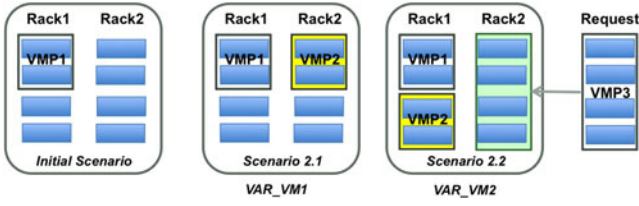


Fig. 5. The illustration of the VM placement with the variance of residual VM capacity calculated and sorted in a descending order.

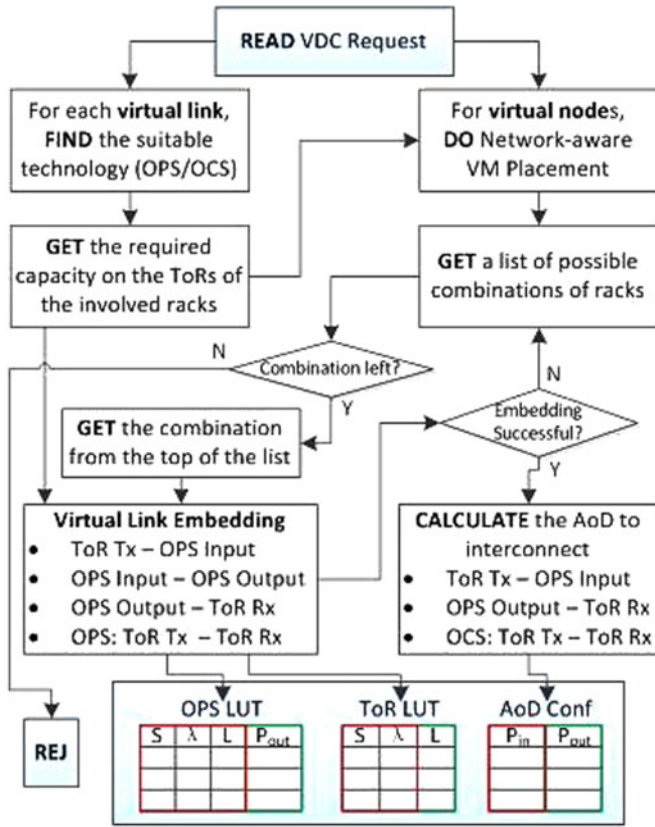


Fig. 6. The overall VDC composition workflow.

334 All the variances are sorted in descending order (L14), and the  
 335 list of possible combinations of racks for embedding the virtual  
 336 nodes is finally obtained (L15). The outputs feed into the overall  
 337 VDC composition process.

338 The reason for sorting the variances in a descending order is  
 339 to increase the VDC request acceptance ratio. To illustrate this,  
 340 a very simple example is given in Fig. 5.

341 For the given situation with VMP1 already embedded (Initial  
 342 Scenario), the two placement options of VMP2 lead to two differ-  
 343 ent scenarios with different variances (Scenario 2.1  $VAR\_VM1$   
 344  $(= 0) < \text{Scenario 2.2 } VAR\_VM2$ ). When the next request arrives  
 345 (VMP3, which requests four VMs), it can only be deployed if  
 346 the second option of the VMP2 placement was adopted, that is,  
 347 the one with higher variance.

348 3) *Virtual Optical DC Composition Workflow*: The overall  
 349 VDC composition workflow is shown in Fig. 6, which integrates  
 350 the network-aware VM placement as described above.

When a VDC request arrives, the information of its requested  
 virtual nodes and virtual links (see Fig. 4) are read and analysed  
 in parallel. For each virtual link, according to its requested  
 bandwidth, the suitable optical switching technology (OPS or  
 OCS) is selected. By taking advantage of the statistical multi-  
 plexing capability of OPS, multiple virtual links can be merged  
 and share a single optical channel. In our work, we sort the  
 bandwidth of the virtual links starting from the same virtual  
 node in ascending order, and then add up the bandwidth of each  
 virtual link till the sum exceeds the capacity of an OPS channel,  
 which is defined as  $x\%$  of the wavelength channel bitrate. The  
 $x\%$  depends on the inherent overhead of OPS implementation  
 and the blocking due to congestion. Through the checking of all  
 the virtual links (bi-directional), the suitable optical technology  
 for each virtual link can be determined as well as the requested  
 number of channels starting from each virtual node. Since one  
 but only one virtual node will be embedded in a rack, the re-  
 quested number of channels is actually the requirement on the  
 corresponding ToR switch of the rack. The *Network-aware VM*  
*Placement* module will take this output as an input together with  
 the VM capacity requirements to finally give a list of possible  
 combinations of racks to embed the virtual nodes. We always  
 pick the top element of the list and input it into the *Virtual Link*  
*Embedding* module.

When embedding the virtual links through OPS channels  
 across the optical DCN, we need to check against the OPS  
 switching matrix as well as the optical layer constraints such  
 as wavelength continuity if there is no wavelength converter  
 employed. If the embedding is not successful, we will go back  
 to the list and choose the next possible combination if there is  
 still one.

If the deployment is successful, we will be able to obtain the  
 pairs of *ToR Tx - ToR Rx* to be connected via OPS channels. More  
 specifically, they are the sets of pairs of *ToR Tx - OPS Input*,  
*OPS Input - OPS Output*, and *OPS Output - ToR Rx*. We can determine  
 the source/destination ToRs, the suitable Input/Output fibres ( $S$ ),  
 wavelengths ( $\lambda$ ), and labels ( $L$ ) for switching optical packets  
 (see Fig. 6). Finally, the connections in the AoD are calculated  
 in order to interconnect the pairs of *ToR Tx - OPS Input*,  
*OPS Output - ToR Rx*, and *ToR Tx - ToR Rx* via OCS. The final outputs  
 will be the *Look Up Tables (LUTs)* for OPS and ToR switches,  
 and configuration files for the AoD, which are pushed by the  
 SDN controller to the devices (described in details in Section V,  
 Experimental Demonstrations).

### C. Implementation of the Dynamic VDC Deployment

The dynamic VDC deployment is implemented on a plat-  
 form with the VDC Composition application interacting with  
 the OpenDaylight SDN controller and the OpenStack cloud  
 management system, as depicted in Fig. 7.

The user (VDC requester) generates a VDC request and calls  
 the DC virtualisation application (APP: VDC Composition).  
 The VDC request is represented in an XML file as shown in  
 Fig. 4.

After a VDC request is received, the request for deploying  
 the virtual DCN is forwarded to the Virtual DCN Manager in

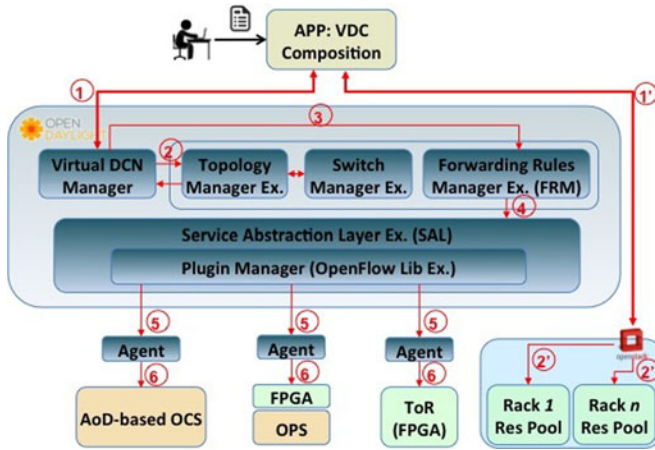


Fig. 7. The VDC deployment via OpenDaylight and OpenStack.

the SDN controller through the NBI (Step 1), while the request for placing VMs is sent to the cloud management system (Step 1'). First of all, the available network and IT resources are retrieved from the Topology Manager in the OpenDaylight controller (Step 2) and OpenStack via Compute API (Step 2'). The Topology Manager exchanges information with the Switch Manager. In Fig. 7, "Ex." indicates those modules in the original ODL controller extended to support optical devices. The retrieved information is presented in a JSON file that specifies the device capabilities and port features.

Based on the retrieved information, the VDC composition engine in the APP is called back to deploy the VDC request using the method presented in Section III-B. The VDC composition engine is implemented in Matlab R2012a.

If the VDC request can be successfully deployed, the configuration rules are specified and pushed to the Forwarding Rules Manager (Step 3) in the SDN controller and the OpenStack Computer Service NOVA. Going through the Service Abstraction Layer and OpenFlow Plugin (OpenFlow Lib) (Step 4), the configuration rules will be pushed to the OF agents of the devices (i.e., AoD backplane, OPS, and ToR switch) via extended OF protocol (see Fig. 3) (Step 5). The devices are then configured by their technology-specified instructions (AoD backplane: TL1, FPGA-based module for OPS and ToR switch: Ethernet frame), which are translated from the extended OF protocol by their own agents (Step 6).

To provision an OCS connection in the proposed hybrid optical DC scenario, the extended OF message *cf\_flow\_mod* is used to configure the Backplane cross-connect to interconnect the source and destination (s/d) ToR switches, and extended *flow\_mod* is used to configure the flow entries in the s/d ToRs. Then incoming packets that are matched against specific VLANs and/or MACs are forwarded to corresponding output ports.

To provision an OPS connection, a flow entry with the rules for packet matching and label configuration (see Fig. 3) is sent to the source ToR, while the destination ToR will receive and forward the packets to the destination servers. Also, another flow entry for the traversed OPS node will be installed to indicate how the optical packet will be forwarded according to its

attached label and assigned wavelength (specified in the extended *ofp\_match*). More details can be found in Section V.

#### IV. SIMULATION STUDIES

In order to evaluate the VDC deployment method we built a simulation platform and conducted extensive experimental studies.

##### A. Simulation Scenarios and Parameters

The simulation platform is built in Matlab R2012a. The network and IT (VM) resources in the optical DC are first simulated as *DC resource database*. The VDC requests are randomly generated according to a set of predefined rules (explained in details below), and processed sequentially. If a VDC request can be successfully accommodated, it will be embedded into the optical DC and the DC resource database will be accordingly updated. When the holding time of a VDC expires, its occupied resources are released for future use.

In the hybrid optical DCN, a cluster includes 8 racks, each rack contains 20/40 servers, and each server has 10 VM images with initial CPU and Memory capacities. Each ToR has 8/16x10Gbps wavelength channels towards the AoD (Backplane). Each channel can support either OPS or OCS transmission. The OPS node supports 10 input and output ports (i.e.,  $10 \times 10$ ), each carrying 8 or 16 wavelengths. Due to the inherent overhead and congestion, the actual channel bitrate for OPS is set to  $85\% \times 10$  Gb/s [31]. Optical labels are adopted to switch the packets being transmitted in the same wavelength to different output ports (i.e., destined ToRs). The number of optical labels needed will be less than or equal to the maximum node degree of the VDC request.

VDC requests arrive dynamically following a Poisson process, while the holding time (lifetime) of each VDC request follows an exponential distribution. The normalized number of VDC requests per time unit is varied from 1 to 10. The VDC topologies (500 in total) are randomly generated with controllable parameters: the number of virtual nodes (Min: 3, Max: 5), node degree (Min: 2, Max: 3), and the probability of interconnecting virtual nodes (0.5). For the capacity requirements, each VDC requests the number of VMs per node (Min: 5, Max: 50) and the bandwidth of each virtual link (Min: 1Gbps, Max: 6 Gb/s).

Based on the simulation platform, we evaluated the impact of available optical DCN and IT resources on the dynamic VDC composition in terms of the acceptance ratio of VDC requests, as well as the benefits of introducing hybrid optical switching technologies into the optical DC and the performance improvements brought by the proposed coordinate/joint mapping of optical DCN and IT resources.

Each result is presented as statistical values with the mean calculated by running the simulation four times. The 95% confidence interval is also shown in the result figures.

##### B. The Impact of Available Optical DCN and IT Resources

The results in Fig. 8 show the impact of available physical resources in the optical DC on the VDC embedding performance



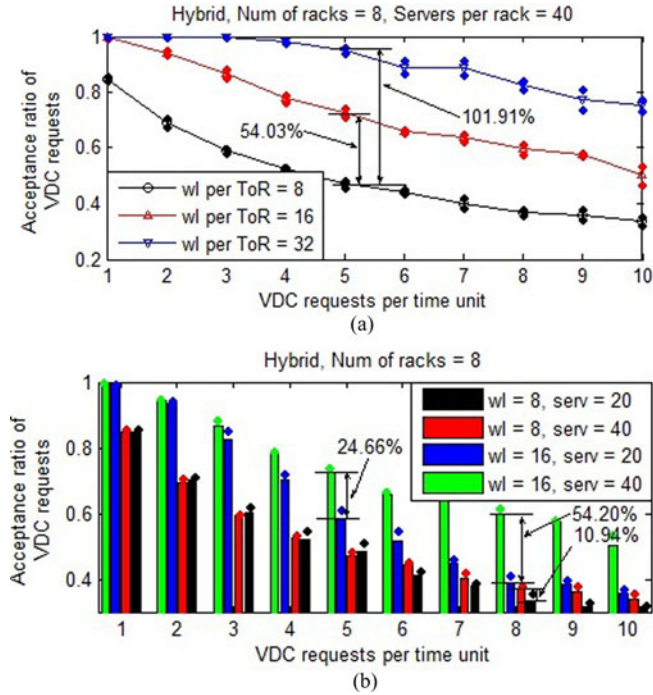


Fig. 8. (a) The impact of number of wavelenghts per ToR switch (*wl per ToR*) on the acceptance ratio of VDC requests in hybrid optical DC with 8 racks. (b) The impact of servers per rack (*serv.*) on the acceptance ratio of VDC requests in hybrid optical DC with 8 racks.

498 in terms of the acceptance ratio of VDC requests. In Fig. 8(a):  
 499 the number of wavelenghts per ToR (*wl per ToR*) [10] and in  
 500 Fig. 8(b): the number of servers (*serv*) hosted in one rack.

501 In Fig. 8(a), compared to the scenario with 8 wavelenghts per  
 502 ToR switch, 16 wavelenghts can improve the acceptance ratio  
 503 of the VDC requests by 54.03% when the normalised number  
 504 of VDC requests per time unit is 5, while 32 wavelenghts can  
 505 achieve 101.91% improvement.

506 In Fig. 8(b), we can see that when the number of wavelenghts  
 507 per ToR is limited (i.e., 8) the improvement brought by the in-  
 508 creased number of servers (IT resources) is also limited. As the  
 509 limitation on the network resources is released by increasing the  
 510 number of wavelenghts, we can see that growing improvements  
 511 are introduced by the increasing IT resources. For example,  
 512 when the normalised number of VDC requests per time unit is  
 513 8, the improvement of the acceptance ratio is 54.20% with 16  
 514 wavelenghts per ToR, and 10.94% with 8 wavelenghts per ToR.  
 515 From Fig. 8(b), we also observe that the improvements brought  
 516 by the increased IT resources are greater when more VDC re-  
 517 quests arrived per time unit, e.g., 24.66% (5 VDC requests per  
 518 time unit) versus 54.20% (8 VDC requests per time unit).

519 *C. The Benefits of Hybrid Optical Switched DC*

520 The advantages of bringing hybrid optical switching technol-  
 521 ogies (OPS and OCS) into DCN are reflected in Fig. 9. Com-  
 522 pared to the optical DCN with pure OCS technology, with 40  
 523 servers per rack, the hybrid optical switched DCN can achieve  
 524 30.11% and 14.61% improvements when the number of wave-  
 525 lenghts per ToR is 8 and 16, respectively. It shows that the

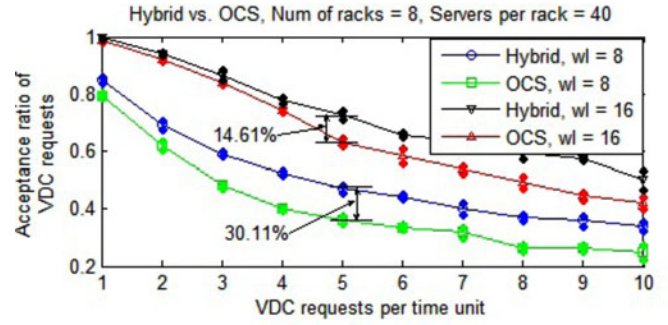


Fig. 9. The comparison of the hybrid optical switched DCN with OPS&OCS and the optical DCN with pure OCS technology.

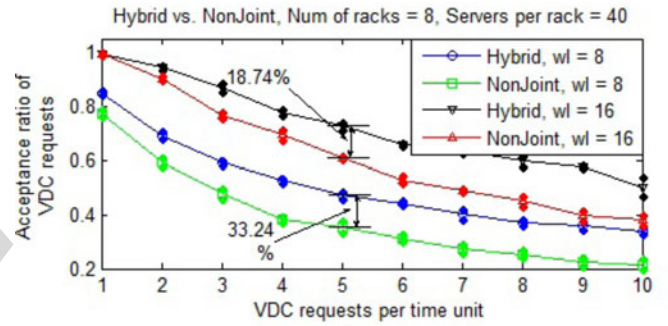


Fig. 10. Effectiveness of the joint optical DCN and IT resources mapping.

improvement becomes bigger when the limitation arising from  
 the availability of network resources (number of wavelenghts  
 per ToR) is stronger.

*D. The Benefits of Joint Mapping of Network and IT Resources*

In Fig. 10, the effectiveness of the proposed VDC composition  
 with network-aware VM placement (“Hybrid”) is evaluated and  
 compared with the one that lacks the joint mapping method  
 (“NonJoint”).

Again, we observe that the improvement becomes bigger  
 when the network resource limitation is stronger, that is, 33.24%  
 with eight wavelenghts per ToR while 18.74% with 16 wave-  
 lenghts per ToR.

Finally, the execution time of the VDC deployment algorithm  
 is also collected, which is around 35 ms.

V. EXPERIMENTAL DEMONSTRATIONS

We have also established an experimental setup to demon-  
 strate the dynamic VDC composition in the optical DC, and  
 evaluate its performance by measuring the time to successfully  
 deploy a VDC request. The configurations provisioned by the  
 SDN controller to each DCN device using the extended OF pro-  
 tocol (AoD backplane, OPS, and ToR switch) are also shown in  
 this section.

*A. VDC Composition in the Experimental Setup*

In this experiment, a Polatis switch with 192 × 192 ports  
 is used as the backplane of the AoD node. An OPS switch

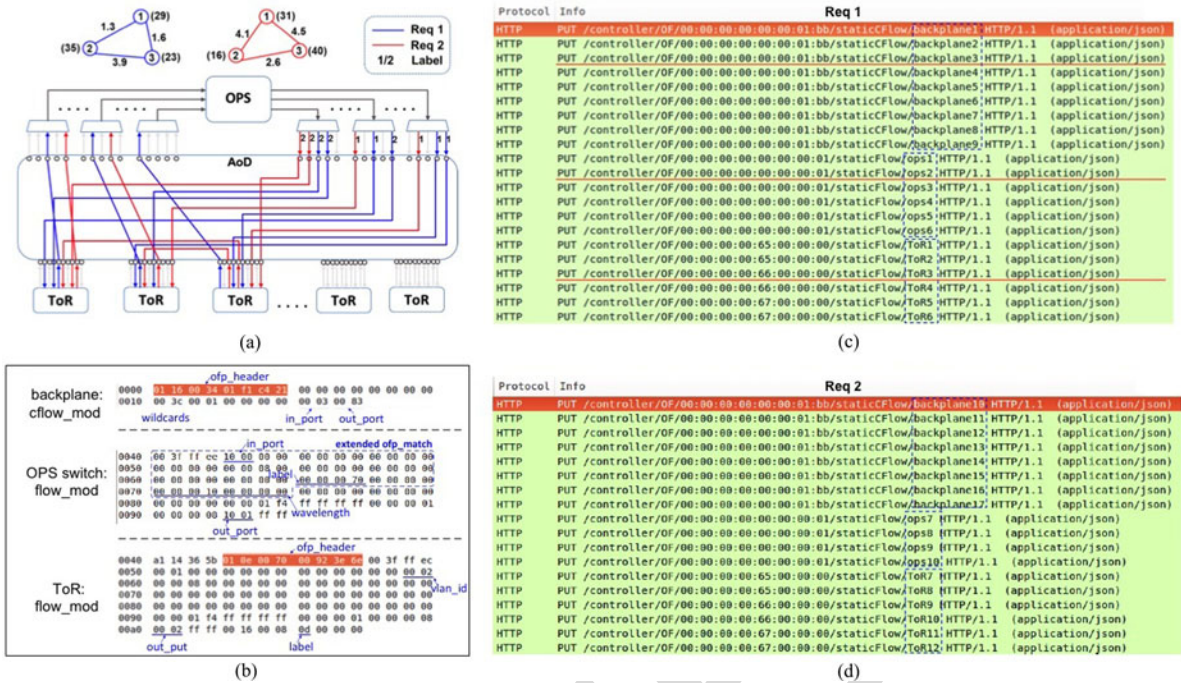


Fig. 11. Virtual optical DCs deployment and configurations

551 and ToR switches are plugged into the backplane. Servers are  
 552 connected to ToR switches via 10GE SFP+. The ToR switches  
 553 are implemented with FPGA optoelectronics with  $12 \times 10GE$   
 554 ports, which can be configured by the SDN controller via the  
 555 corresponding OF agent. The OPS switch is implemented with  
 556 SOA based fast switch and RF tones labeling technique, as well  
 557 as an FPGA-based local controller [31]. More implementation  
 558 details of each individual device can be found in [10].

559 Two VDC requests are generated randomly and processed  
 560 one by one. In Fig. 11(a), they are marked blue (Req1) and  
 561 red (Req 2), respectively. They both have a three-node topology  
 562 with capacity requirements on their virtual nodes and virtual  
 563 links, which are specified in the figure. Bidirectional communi-  
 564 cations are considered while deploying the VDC requests, that  
 565 is, one virtual link is allocated with two bidirectional physical  
 566 lightpaths. Before the VDC deployment, the VDC requester is  
 567 not concerned and is unaware of the physical location of nodes  
 568 and links allocated to the VDC. The VDC requester will call  
 569 the VDC composition APP to process the request following the  
 570 procedure described in Section III. In order to clearly show the  
 571 VDC deployment, i.e., the end-to-end interconnections between  
 572 different devices in the wavelength level, the necessary DeMux  
 573 and Mux are eliminated from the figure. Labels (1/2) are used  
 574 to indicate the switching of optical packets transmitted over the  
 575 same input port/wavelength to different output ports.

## 576 B. VDC Configurations

577 After processing a VDC request, the SDN controller will push  
 578 the devices' configurations via the extended OF protocol. These  
 579 are then translated into technology specific control messages by  
 580 the OF agents of these devices. As an example, to map the two

TABLE I  
BREAKDOWN OF THE VDC DEPLOYMENT TIME

Total (ms)	Application Execution	Process in OpenDaylight	Process in OpenStack	Device Config.
255.12	35.12 ms	Running in Parallel		25 ms
		195 ms	158 ms	

VDC requests in Fig. 11(a), the network configuration items  
 581 are shown in Fig. 11(c) and (d), and one selected configuration  
 582 item for each device is expanded and the extended OpenFlow  
 583 messages are shown in Fig. 11(b). The command lines for con-  
 584 figurations are listed in the Appendix.  
 585

Backplane: *cflow\_mod* indicates the input and output ports of  
 586 the requested optical cross connection, which is extended from  
 587 the OpenFlow 1.0 with addendum draft v0.3.  
 588

OPS: *flow\_mod* is extended by adding optical label (4 bytes)  
 589 and wavelength (8 bytes) information in the matching fields.  
 590 The corresponding input and output ports are also indicated.  
 591

ToR switch: *flow\_mod* indicates the VLAN ID for packet  
 592 matching and extended with OPS label setting action if the  
 593 matched packet is going to be delivered via an OPS connection.  
 594

In order to allocate resources (i.e., vCPU and Memory) for  
 595 VMs, the flavor list and OS (Operating System) type in the  
 596 image list need to be predefined.  
 597

## 598 C. VDC Deployment Time

599 The breakdown of the total time for successfully deploy-  
 600 ing a VDC request is given in Table I, which comprises of 1)  
 601 VDC application execution (to call the VDC deployment algo-



602 rithm engine) (35.12 ms), 2) information processing and mes-  
 603 sage exchanges in the OpenDaylight SDN controller (195 ms)  
 604 and OpenStack cloud management system (158 ms), and  
 605 3) device configurations (25 ms). The processes in the SDN  
 606 controller execute in parallel with those in OpenStack. The total  
 607 deployment time is 255.12 ms.

608 VI. CONCLUSION

609 In this paper, we introduced a flat hybrid optical-switched  
 610 DCN architecture equipped with an advanced optical data plane  
 611 and a comprehensive SDN control platform, which is able to  
 612 provision dynamic and flexible optical connectivity with high  
 613 throughput and low latency. Using this optical DC, we designed  
 614 and developed a network-aware VDC deployment method as a  
 615 DC application for enabling multi-tenancy in the optical cloud  
 616 environment. The performance evaluation shows the benefits  
 617 of bringing hybrid optical switching technologies (OPS and  
 618 OCS) into the DCN, and the effectiveness of the coordinate  
 619 deployment of virtual DCN resources and VMs. Moreover,  
 620 we demonstrated deployment of dynamic VDCs, the extended  
 621 OpenDaylight SDN control platform and the OpenStack cloud  
 622 management system. The detailed configurations of DCN de-  
 623 vices exchanged via extended OF messages and OF agents were  
 624 described. Finally, the time to successfully deploy a VDC re-  
 625 quest was measured and analysed to give insight into the imple-  
 626 mentation process.

627 APPENDIX

628 **Backplane:**

629 `curl -u admin:admin -H 'Content-type: application/json'`  
 630 `-X PUT -d '{"installInHw": "true", "name": "backplane",`  
 631 `"node":`  
 632 `{ "id": "00:00:00:00:00:00:01:bb", "type": "OF" }, "ingress-`  
 633 `Port": "x",`  
 634 `"outgressPort": "y", "inWavelength": "1", "outWavelength":`  
 635 `"1", "wildcards": "OFPCW_FIBER_PORT", "actions":`  
 636 `["OFFPAT_CKT_OUTPUT"]}' 'http://localhost:8080/`  
 637 `controller/nb/v2/flowprogrammer/default/node/OF/00:00:00:`  
 638 `00:00:00:01:bb/staticCFlow/backplane'`

639 **OPS:**

640 `curl -u admin:admin -H 'Content-type: application/json' -X`  
 641 `PUT -d '{"installInHw": "true", "name": "ops", "node":`  
 642 `{ "id": "00:00:00:00:00:00:00:01", "type": "OF" }, "in-`  
 643 `gressPort": "x", "priority": "500", "etherType": "0x800",`  
 644 `"opsLabel": "y", "opsInWavelength": "z", "actions":`  
 645 `["OUTPUT = n"]}' 'http://localhost:8080/controller/nb/v2/`  
 646 `flowprogrammer/default/node/OF/00:00:00:00:00:00:00:01/`  
 647 `staticFlow/ops'`

648 **ToR switch:**

649 `curl -u admin:admin -H 'Content-type: application/json' -X`  
 650 `PUT -d '{"installInHw": "true", "name": "ToR", "node":`  
 651 `{ "id": "00:00:00:00:00:00:0x", "type": "OF" }, "ingress-`  
 652 `Port": "y", "priority": "500", "etherType": "0x800", "vlanId":`  
 653 `"z", "actions": ["OUTPUT = n", "SET_LABEL = m",`  
 654 `"SET_TIMESLOT = p"]}' 'http://localhost:8080/controller/`

`nb/v2/flowprogrammer/default/node/OF/00:00:00:00:65:00:`  
 655 `00:00/staticFlow/ToR`  
 656

657 **VM:**

658 `curl -i http://nova_controller_ip:8774/v2/tenant_id/servers -`  
 659 `X POST -H "X-Auth-Project-Id: admin" -H "Content-Type:`  
 660 `application/json" -H "Accept: application/json" -H "X-Auth-`  
 661 `Token: token_id" -d '{ "server": { "name": "instance_name",`  
 662 `"imageRef": "image_id", "key_name": "repleced_with_key-`  
 663 `name", "flavorRef": "flavor_id", "max_count": 1, "min-`  
 664 `count": 1 } }'`

665 REFERENCES

666 [1] R. Villars, J. Kopyy, and K. Quinn, "From IDC Predictions 2013: The new  
 667 data center dynamic," Dec. 2012.  
 668 [2] A. Von Lehmen, R. Doverspike, G. Clapp, D. M. Freimuth, J. Gannett,  
 669 K. Kim, H. Kobrinski, E. Mavrogiorgis, J. Pastor, M. Rauch,  
 670 K. K. Ramakrishnan, R. Skoog, B. Wilson, and S. L. Woodward,  
 671 "CORONET: Testbeds, cloud computing, and lessons learned," presented  
 672 at the Opt. Fiber Commun. Conf. Exhib., San Francisco, CA, USA, 2014.  
 673 [3] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of  
 674 data centers in the wild," in *Proc. 10th ACM SIGCOMM Conf. Internet*  
 675 *Measurement*, Nov. 2010, pp. 267–280.  
 676 [4] L. A. Barroso and U. Hölzle, "The data centre as a computer: An intro-  
 677 duction to the design of warehouse-scale machines," in *Synthesis Lectures*  
 678 *on Computer Architecture*, San Rafael, CA, USA: Morgan & Claypool,  
 679 2009.  
 680 [5] F. Abel, C. Minkenberg, I. Iliadis, T. Engbersen, M. Gusat, F. Gramsamer,  
 681 and R. P. Luijten, "Design issues in next-generation merchant switch  
 682 fabrics," *IEEE/ACM Trans. Netw.*, vol. 15, no. 6, pp. 1603–1615, 2007.  
 683 [6] S. J. B. Yoo, "Energy efficiency in the future internet: The role of opti-  
 684 cal packet switching and optical-label switching," *IEEE J. Sel. Topics*  
 685 *Quantum Electron.*, vol. 17, no. 2, pp. 406–418, Mar./Apr. 2011.  
 686 [7] N. Farrington, G. Porter, S. Radhakrishnan, H. H. Bazzaz, V. Subramanya,  
 687 Y. Fainman, G. Papen, and A. Vahdat, "Helios: A hybrid electrical/optical  
 688 switch architecture for modular data centres," in *Proc. ACM SIGCOMM*,  
 689 2010, pp. 339–350.  
 690 [8] G. Wang, D. G. Andersen, M. Kaminsky, K. Papagiannaki, T. E. Ng,  
 691 M. Kozuch, and M. Ryan, "c-Through: Part-time optics in Data centres,"  
 692 in *Proc. ACM SIGCOMM*, 2010, pp. 327–338.  
 693 [9] M. Garrich, N. Amaya, G. Zervas, P. Giaccone, D. Simeonidou, "Archite-  
 694 cture on demand: Synthesis and scalability," presented at the 16th Int.  
 695 Conf. Opt. Netw. Design Modeling, Colchester, U.K., 2012.  
 696 [10] S. Peng, D. Simeonidou, G. Zervas, R. Nejabati, Y. Yan, Y. Shu, S. Spadaro,  
 697 J. Perelló, F. Agraz, D. Careglio, H. Dorren, W. Miao, N. Calabretta,  
 698 G. Bernini, N. Ciulli, J. Carlos Sancho, S. Iordache, Y. Becerra,  
 699 M. Farrera, M. Biancani, A. Predieri, R. Proietti, Z. Cao, L. Liu, and  
 700 S. J. B. Yoo, "A novel SDN enabled hybrid optical packet/circuit switched  
 701 data centre network: The Lightness approach," presented at the IEEE Eur.  
 702 Conf. Netw. Commun., Bologna, Italy, Jun. 2014.  
 703 [11] [Online]. Available: <http://www.ict-lightness.eu/>  
 704 [12] J. Perello, S. Spadaro, S. Ricciardi, D. Careglio, S. Peng, R. Nejabati,  
 705 G. Zervas, D. Simeonidou, A. Predieri, M. Biancani, H. J. S. Dorren,  
 706 S. Di Lucente, J. Luo, N. Calabretta, G. Bernini, N. Ciulli, J. C. Sancho,  
 707 S. Iordache, M. Farreras, Y. Becerra, C. Liou, I. Hussain, Y. Yin, L. Liu,  
 708 and R. Proietti, "All-optical packet/circuit switching- based data center  
 709 network for enhanced scalability, latency and throughput," *IEEE Netw.*  
 710 *Mag.*, vol. 27, no. 6, pp. 14–22, Nov./Dec. 2013.  
 711 [13] B. Guo, S. Peng, C. Jackson, Y. Yan, Y. Shu, W. Miao, H. Dorren,  
 712 N. Calabretta, F. Agraz, J. Perelló, S. Spadaro, G. Bernini, R. Monno,  
 713 N. Ciulli, R. Nejabati, G. Zervas, and D. Simeonidou, "SDN-enabled  
 714 programmable optical packet/circuit switched intra data centre network,"  
 715 presented at the Opt. Fiber Commun. Conf., Los Angeles, CA, USA, Mar.  
 716 2015, Paper Th4G.5.  
 717 [14] [Online]. Available: <https://www.opennetworking.org/>  
 718 [15] [Online]. Available: <http://www.opendaylight.org/>  
 719 [16] Cloud Industry Forum. [Online]. Available: <http://cloudindustryforum.org>  
 720 [17] J. Mudigonda, P. Yalagandula, J. Mogul, B. Stiekes, and Y. Pouffary,  
 721 "NetLord: A scalable multi-tenant network architecture for virtualized  
 722 datacenters," in *Proc. ACM SIGCOMM*, 2011, pp. 62–73.  
 723 [18] M. F. Bari, R. Boutaba, R. Esteves, L. Z. Granville, M. Podlesny,  
 724 M. G. Rabbani, Q. Zhang, and M. F. Zhani, "Data center network

Q1

Q2

Q3

- virtualization: A Survey,” *IEEE Commun. Surveys Tut.*, vol. 15, no. 2, pp. 909–928, 2013.
- [19] S. Peng, R. Nejabati, and D. Simeonidou, “The role of optical network virtualization in cloud computing,” *IEEE/OSA J. Opt. Commun. Netw.*, vol. 5, no. 10, pp. A162–A170, Oct. 2013.
- [20] J. Daniels, “Server virtualization architecture and implementation,” *ACM Crossroads*, vol. 16, no. 1, pp. 8–12, Sep. 2009.
- [21] [Online]. Available: <http://openswitch.org/>
- [22] T. Benson, A. Anand, A. Akella, and M. Zhang, “Understanding data center traffic characteristics,” in *Proc. ACM SIGCOMM Workshop: Res. Enterprise Netw.*, 2009.
- [23] P. Peter, S. Panchen, and N. McKee, “InMon corporation’s sFlow: A method for monitoring traffic in switched and routed networks,” *Network Working Group*, RFC 3176, 2001.
- [24] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, “Hedera: Dynamic flow scheduling for data center networks,” in *Proc. 7th USENIX Conf. Netw. Syst. Design Implementation*, vol. 10, 2010, p. 19.
- [25] Y. Shu, G. Zervas, Y. Yan, S. Peng, S. Yan, E. Hugues-Salas, and D. Simeonidou, “Programmable optical packet/circuit switched data center interconnects: Traffic modeling and evaluation,” presented at the Eur. Conf. Opt. Commun., Cannes, France, Sep. 2014.
- [26] [Online]. Available: <http://www.polatis.com/press-releases/polatis-introduces-192-fibre-single-sided-all-optical-switch-for-datacentre-and-telecom-networks.asp>
- [27] M. Channegowda, R. Nejabati, M. Rashidi Fard, S. Peng, N. Amaya, G. Zervas, D. Simeonidou, R. Vilalta, R. Casellas, R. Martínez, R. Muñoz, L. Liu, T. Tsuritani, I. Morita, A. Autenrieth, J. P. Elbers, P. Kosteci, and P. Kaczmarek, “Experimental demonstration of an OpenFlow based software defined optical network employing packet, fixed and flexible DWDM grid technologies on an international multi-domain testbed,” *OSA Opt. Exp.*, vol. 21, no. 5, pp. 5487–5498, 2013.
- [28] F. Agraz, W. Miao, A. Ferrer, G. Bernini, H. J. S. Dorren, N. Calabretta, N. Ciulli, J. Perelló, S. Peng, G. Zervas, D. Simeonidou, G. Junyent, and S. Spadaro, “Experimental assessment of SDN-based control of OPS switching nodes for intra-data centers interconnect,” presented at the Eur. Conf. Opt. Commun., Cannes, France, Sep. 2014.
- [29] [Online]. Available: <http://www.openstack.org>
- [30] S. Peng, R. Nejabati, B. Guo, Y. Shu, G. Zervas, S. Spadaro, A. Pages, and D. Simeonidou, “Enabling multi-tenancy in hybrid optical packet/circuit switched data center networks,” presented at the Eur. Conf. Opt. Commun., Cannes, France, Sep. 2014.
- [31] W. Miao, S. Di Lucente, J. Luo, H. J. S. Dorren, and N. Calabretta, “Low latency and efficient optical flow control for intra data center networks,” *Opt. Exp.*, vol. 22, pp. 427–434, 2014.

**Shuping Peng** received the B.S. degree in physics, and the Ph.D. degree in communications and information system from Peking University, Beijing, China. She is currently working as a Research Fellow in High Performance Networks Group, University of Bristol, Bristol, U.K. She is involved in multiple international, EU, U.K. projects. Her research interests include optical network virtualization, software-defined optical data centers, network modeling, and algorithm design. She is author and coauthor of more than 60 papers, and has served as a TPC member and the session chair of several IEEE/ACM conferences.

**Bingli Guo** received the B.S. degree in telecommunication engineering from Jilin University, Jilin, China, in 2005, and the Ph.D. degree in electromagnetic field and microwave technology from the Beijing University of Post and Telecommunication, Beijing, China, in 2011. He is currently a Research Associate at the High Performance Network Group, University of Bristol, Bristol, U.K. From 2011 to October 2013, he was a Researcher in Peking University, Beijing, China. His research interests include routing algorithm in converged IP and optical network, survivability strategies in integrated network and cloud computing system, and software defined optical network.

**Chris Jackson** received the M.Eng. and Ph.D. degrees from the University of York, York, U.K., and Bristol in 2007 and 2013. He is currently a Research Associate with the High Performance Networks Group, University of Bristol. He has worked previously in Business Intelligence and Analytics. He has authored and coauthored papers on energy-efficient on-chip interconnection networks, including routing algorithms and adaptive architectures. His current research interests include optical networks, software-defined networking, abstractions for network virtualization, disaggregated data center architectures and programming language development.

**Reza Nejabati** is a Lecturer at the University of Bristol, Bristol, U.K. His research interest include the application of high-speed network technologies, design and control of software-defined, service-oriented and programmable network, cross-layer network design, network architecture and technologies for e-science and cloud computing. He is author and coauthor of more than 150 papers and three standardization documents. He is involved in several national/international projects. He has served as a TPC member, the chair and a organizer of several IEEE conferences and workshops.

**Fernando Agraz** received the M.Sc. degree in computer science in 2005 and the Ph.D. degree in signal theory and communications in 2012 from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain. He is currently working as a Research Engineer in the Optical Communications Group of the Signal Theory and Communications Department of the UPC where he is the responsible of several software developments. He has participated in various FP6 and FP7 European research projects such as LIGHTNESS, EULER, STRONGEST, DI-CONET, etc. His expertise includes software engineering, communication protocols, and systems administration. His research interests focus on all-optical networks routing, control and management, as well as their application to data center networks and virtualization.

**Salvatore Spadaro** received the M.Sc. degree in 2000 and the Ph.D. degree in telecommunications engineering from Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 2005. He also received the Dr.Ing.degree in electrical engineering from Politecnico di Torino, Torino, Italy, in 2000. He is currently an Associate Professor in the Optical Communications Group of the Signal Theory and Communications Department of UPC. He has participated in various FP6 and FP7 European research projects. He is currently the UPC scientific coordinator for the running EU-funded LIGHTNESS and COSIGN projects. He has published more than 150 articles in international journals and conference proceedings. His research interests include the fields of all-optical networks for data center interconnects, with emphasis on network control and management, resilience and network resources virtualization.

**Giacomo Bernini** received the Italian Laurea degree in telecommunication engineering from the University of Pisa, Pisa, Italy, in 2006. He is currently the R&D Project Manager at Nextworks, and his research interests include SDN, NFV, cloud computing, NSI, ASON/GMPLS control plane, and PCE frameworks. He recently participated to design, development, and demonstration activities in several FP6 and FP7 projects as well as to industrial projects. He is currently active in the FP7 LIGHTNESS, TRILOGY 2, PACE, CONTENT, and H2020 ORCHESTRA projects.

**Nicola Ciulli** is a Co-founder and the Head of Research and Development activities at Nextworks, an innovative and dynamic SME, based in Tuscany, Italy, with a lot of investment in R&D activities. He holds a degree in telecommunication engineering from the University of Pisa and SSSUP S. Anna (1997). He has coauthored and participated in several (20+) EU FP4-FP7 and H2020 projects in collaboration with a wide range of EU actors, in National and Regional research initiatives, and in industrial projects. He is a frequent external expert in EC evaluations, project reviews, and call observations, with 10+ years of experience (FP6, FP7, and H2020).

**Dimitra Simeonidou** is a Professor of High Performance Networks Group, University of Bristol, Bristol, U.K. He is a leading Academic in optical networks, future internet research and experimentation (fire), grid and cloud computing, and a Founder of transport software defined networking. She has chaired a number of international conferences and committees across these technical fields. She is the author and co-author of more than 350 publications of which many have received best paper awards, 11 patents, and standards.