




Article

# End-to-End Real-Time Demonstration of the Slotted, SDN-Controlled NEPHELE Optical Datacenter Network

Konstantinos Tokas <sup>1,\*</sup>, Giannis Patronas <sup>2,3</sup>, Christos Spatharakis <sup>1</sup>, Paraskevas Bakopoulos <sup>2</sup> , Angelos Kyriakos <sup>3</sup>, Giada Landi <sup>4</sup>, Eitan Zahavi <sup>2</sup>, Kostas Christodoulopoulos <sup>5</sup>, Muzzamil Aziz <sup>6</sup>, Richard Pitwon <sup>7</sup> , Domenico Gallico <sup>8</sup>, Dionysios Reisis <sup>3</sup>, Emmanouel Varvarigos <sup>1</sup>  and Hercules Avramopoulos <sup>1</sup>

<sup>1</sup> Institute of Communication and Computer Systems, School of Electrical and Computer Engineering, National Technical University of Athens, 9 Iroon Polytechniou Str., 15780 Athens, Greece; cspatha@mail.ntua.gr (C.S.); vmanos@mail.ntua.gr (E.V.); hav@mail.ntua.gr (H.A.)

<sup>2</sup> Mellanox Technologies, Hakidma 26, Yokneam 2069200, Israel; giannisp@mellanox.com (G.P.); paraskevasb@mellanox.com (P.B.); zahavi.eitan@gmail.com (E.Z.)

<sup>3</sup> Electronics Laboratory, Faculty of Physics, National and Kapodistrian University of Athens, 15772 Athens, Greece; akyriakos@phys.uoa.gr (A.K.); dreisis@phys.uoa.gr (D.R.)

<sup>4</sup> Nextworks, via Livornese 1027, 56122 Pisa, Italy; g.landi@nextworks.it

<sup>5</sup> Nokia Bell Labs, Lorenzstrasse 10, 70435 Stuttgart, Germany; Konstantinos.1.christodoulopoulos@nokia-bell-labs.com

<sup>6</sup> Gesellschaft für wissenschaftliche Datenverarbeitung mbH (GWDG), 37077 Göttingen, Germany; muzzamil.aziz@gwdg.de

<sup>7</sup> Resolute Photonics, 17B West Street, Fareham, Hampshire PO16 0BG, UK; rpitwon@resolutephotonics.com

<sup>8</sup> Lucense s.c.a.r.l., Via Della Chiesa di Sorbano del Giudice n. 231, 55100 Lucca, Italy; domenico.gallico@lucense.it

\* Correspondence: ktok@mail.ntua.gr; Tel.: +30-210-772-2871

Received: 26 May 2020; Accepted: 24 June 2020; Published: 25 June 2020



**Abstract:** The NEPHELE hybrid electro-optical datacenter network (DCN) architecture is proposed as a dynamic network solution to provide high capacity, scalability, and cost efficiency in comparison to the existing DCN infrastructures. The details of the NEPHELE DCN architecture and its various key parts are introduced, and the performance of its implementation is evaluated through an end-to-end NEPHELE demonstrator, which was built at the National Technical University of Athens. Several communication scenarios are demonstrated in real time, exploiting a scalable optical data-plane architecture with a software-defined network (SDN) control plane capable of slotted operation for dynamic allocation of network resources. Real-time end-to-end functionality and integration of various software and hardware components are verified in a six-host prototype datacenter cluster.

**Keywords:** optical networking; optical switching; dynamic resource allocation; datacenter architecture; software-defined networking; demonstrator

## 1. Introduction

Today, datacenters (DC) are the heart of our online applications and Internet of things (IoT) services, handling vast amounts of digital information. The continuous enrichment of these online services and applications opens new vistas in user experience and sparks demand in bandwidth and speed. More and more value-added digital services spanning from virtual reality (VR) and high-definition (HD) video streaming to cloud storage and sensor networks, forming the IoT, are sprouting all over the globe, burdening the internet hubs with heavy digital load. In the 5G era, all these activities are

becoming more and more bandwidth hungry since the users demand instant and on-the-go access at any time. As a result, the datacenter networks (DCNs) should be capable of providing ultra-high capacity interconnects for a huge number of nodes and hosts, low latency to fulfil time-critical services, and high-reliability performance to reduce service interruption time. Therefore, this profound effect emerges as an impact on the datacenter operation, driving worldwide datacenter network IP (Internet Protocol) traffic on a steep growth curve reaching 25% annually [1].

Since the traffic within a DCN is much higher than the incoming/outgoing traffic [1], DCNs are facing remarkable challenges regarding resource utilization, scalability, and management agility. To successfully handle this soaring demand and avoid any possible capacity crunch and the relentless increase of power consumption, DCN operators and equipment providers are struggling to upgrade the existing infrastructures. Current state-of-the-art intra-DCNs are based on electronic switches interconnected in fat-tree or folded-clos topologies using fibers, with electro-opto-electrical conversion at each node [2]. However, fat-tree topologies tend to underutilize resources and at the same time require a multitude of cables, fibers, and switches. Furthermore, within a conventional electrically switched DCN, as the size of the network scales and servers require more bandwidth, the switches must double their bandwidth every few years. The use of a large number of electrical packet switches contributes hugely to the energy consumption of the whole system (32-port 400-GbE switches consume almost 1000 W when fully populated with optical transceivers) [3]. It should be highlighted that roughly 90% of this energy consumption is independent of the load and, thus, savings are impossible from any load balancing/scheduling method. Finally, upgradability is a big issue for electrical networks since upgrading the communication rate of the servers requires replacing all the switches of the network. Note here that the optical transceivers that are needed to interconnect the electrical switches add up to a considerable capital expenditure, as well as significant power consumption [4].

Optical switching technologies are gaining traction as a potential vehicle to address the above-mentioned challenging requirements. Deployment of photonic components could offer network scalability, due to their inherent speed, energy efficiency, and transparency to protocol and bitrate. Several proposals based on optical technologies [5] were introduced as effective solutions within DCNs, such as space switching (e.g., using micro-electro-mechanical systems—MEMS or semiconductor optical amplifiers—SOAs [6,7]), wavelength switching (through combination of tunable lasers with arrayed waveguide grating routers—AWGRs [8,9]), or a combination thereof (e.g., using wavelength-selective switches—WSSs [10]). One of the key challenges currently pertaining to optical datacenter networks is the combination of scalability and fast reconfigurability. To this end, several efforts promote the integration of optical switching into control and orchestration frameworks, so-called software-defined networking (SDN) [11,12]. Indeed, SDN platforms in combination with orchestration algorithms provide dynamicity and scalability in DCNs and enhance the benefits of the optical switches.

During the last decade, several works proposed hybrid electrical/optical and all optical DCN concepts, and a detailed survey can be found in Reference [13]. An enhancement to the current DCNs, named c-Through, was presented in Reference [14]. The ToR switches in c-Through are connected to the legacy electrical network, as well as to an optical circuit switching network. The optical network is used to connect pairs of racks with high-bandwidth demands based on decisions taken by a traffic monitoring system that measures the bandwidth requirements. Helios, a hybrid electrical/optical network based on wavelength division multiplexing (WDM), was proposed in Reference [15]. In Helios, the circuits created by the optical switches are used for elephant flows (high bandwidth, slowly changing communication), which is limited by the ms reconfiguration time of the employed MEMS switches. Other works proposed DC interconnects completely lacking electrical switches, such as Proteus [16], an all-optical DCN architecture based on a combination of wavelength selective switches (WSSs) and MEMS. The key idea of Proteus is to use direct optical connections for the elephant flows and multi-hop connections for the shot-lived mice flows. In Reference [17], the authors introduced the Mordia microsecond switch to the DCN and studied the gains and issues for microsecond switching. They proposed a microsecond-latency control plane based on a circuit scheduling approach called

Traffic Matrix Scheduling (TMS). However, the scalability of Mordia is limited as it uses a single wavelength division multiplexing (WDM) ring whose capacity can accommodate only a few racks, while deployed resource allocation algorithms exhibit high complexity and cannot scale to large DCs. In Cboss [18], the authors proposed a DCN architecture based on a silicon photonics Wavelength Dropper, fast tunable lasers, and an SDN-based controller. Cboss consists of a WDM time-slotted ring for data transmission and a separate control channel to enable SDN functionality. Another all-optical DCN architecture, named OPSquare, was published in Reference [19], which presented a flat DCN topology with the capability for distributed control plane functionality. In Rotornet [20] and Opera [21], the authors proposed removing the need for resource allocation algorithms and instead shaping the traffic to the network's state. In Rotornet, the ToR switches are connected to an optical network for transmission of the elephant flows and to an electrical network for the rest of the traffic. Opera proposes the removal of the electrical network and follows a direct versus multi-hop transmission scheme for the elephant and the mice flows, respectively. The identification of "elephant" flows is a crucial functionality for many of the hybrid and all-optical DCN proposals, and several works focused on this [22–24].

NEPHELE [25] is a recently completed European project that developed a dynamic end-to-end optical network infrastructure for large-scale and disaggregated datacenters [26]. In this context, NEPHELE combines optical switching benefits with SDN control and orchestration to beat current datacenter challenges. To achieve this and following a vertical development approach, NEPHELE is expanding from the datacenter architecture to the overlaying control plane, in order to deliver a fully functional networking solution. NEPHELE brings two ambitious innovations.

Firstly, the NEPHELE data plane architecture leverages commercial off-the-shelf (COTS) photonic components in combination with slotted time division multiple access (TDMA) operation to enable dynamic and efficient sharing of resources [27,28].

Secondly, an SDN orchestration and control framework is responsible for the management of all the underlying data plane elements, extending the open-source SDN platforms with TDMA functionalities. From this angle, NEPHELE's framework is capable of dynamically assigning network resources directly at the optical layer [29]. Multiple algorithmic add-ons focusing on the fast resource allocation were developed and integrated to the SDN platform [30].

The rest of this manuscript is organized as follows: Section 2 summarizes the NEPHELE architecture, the data plane, and the control plane. In Section 3, we introduce the NEPHELE demonstrator [31,32] assembly, which was built at the Photonics Communications Research Laboratory in the National Technical University of Athens, while its performance is assessed through several real-time and end-to-end communication scenarios presented in Section 4. Finally, Section 5 concludes the paper.

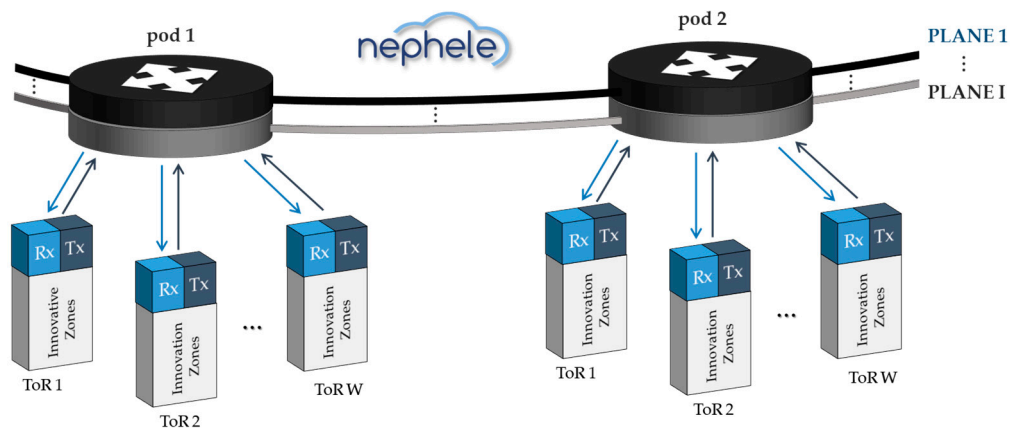
## 2. The NEPHELE Network Architecture

The proper operation of the NEPHELE DCN is based on an architecture where two tiers are collaborating and coexisting seamlessly: the data plane and the control plane. In this section, we present an overview of the NEPHELE architecture, highlighting the key innovations and functionalities of the network.

### 2.1. Data Plane Overview

The NEPHELE network adopts a flat and scalable topology that utilizes active and passive optical components to overcome the shortcomings of hierarchical topologies that are broadly used in electrical DCN architectures. As shown in Figure 1, NEPHELE consists of two layers of switches: the ToR (top-of-rack) and the pod switches. This way, the NEPHELE topology serves efficiently both the north–south and east–west communication, which is an important advantage compared to conventional DC topologies. In addition, in NEPHELE, the required number of network modules scales linearly with the end-nodes, while the fat-tree network scales super-linearly requiring the addition of switches

at all levels and the addition of extra levels once all ports are connected. The reference architecture of the NEPHELE network, which assumes 32K supported ports, was calculated to be about two times more expensive than the equivalent three-level fat-tree network based on current component prices and projections for volume production. Nevertheless, the cost difference decreases as the number of supported ports increases, because the cost of the NEPHELE network increases linearly as opposed to the super-linear cost increase of the fat tree. Accordingly, for 256K ports, the projected cost of the NEPHELE network is the same as the cost of an equivalent (four-level) fat tree. It is worth highlighting that the energy consumption of the reference NEPHELE network (32K supported ports) is less than half of the equivalent fat tree, and the benefits improve further as the size of the network increases [26].



**Figure 1.** Overview of NEPHELE data plane network architecture.

The NEPHELE network architecture relies on pods, which in a sense are small datacenters, accommodating several racks. Each rack is regulated by a top-of-rack (ToR) switch and consists of several hosts (i.e., disaggregated storage and compute resources, hereafter called “innovation zones”). The ToRs are connected to the pod switch following the star topology structure. Network scalability is achieved by interconnecting multiple pods in a dense wavelength division multiplexing (DWDM) multiple-fiber ring. Moreover, multiple parallel NEPHELE planes interconnecting the pods, as shown in Figure 2, further scale the overall throughput of the network. The NEPHELE optical plane refers to the ensemble of a NEPHELE multi-fiber ring along with its corresponding pod and ToR interfaces. Each optical plane is connected to a different port of each ToR.

The NEPHELE data plane operates in a slotted time division multiple access (TDMA) manner. The NEPHELE slot duration is 200  $\mu$ s with additional 10  $\mu$ s as guard time. The guard time was defined by the response of the non-ideal DC-coupled electronics and the lock time of the FPGA receiver, while the slot duration was specified at 200  $\mu$ s in order to provide 95% network utilization. Furthermore, the slotted operation facilitates the dynamic resource allocation of the NEPHELE network, offering dynamic reconfiguration with sub-wavelength granularity. The scheduling (resource allocation) process is realized in a periodic manner. The time is divided in scheduling periods, with each period consisting of 80 slots (16 ms). At the end of each scheduling period, the scheduler calculates the configuration of the network for each slot of the next scheduling period. The control plane and scheduling are discussed in Section 2.2. The following subsections provide an overview of the functionalities of the data plane elements that compose the NEPHELE network. More details on the NEPHELE data plane architecture can be found in Reference [26] along with network dimensioning studies, whereas a preliminary validation of the ToR and pod switches is presented in [28,33].

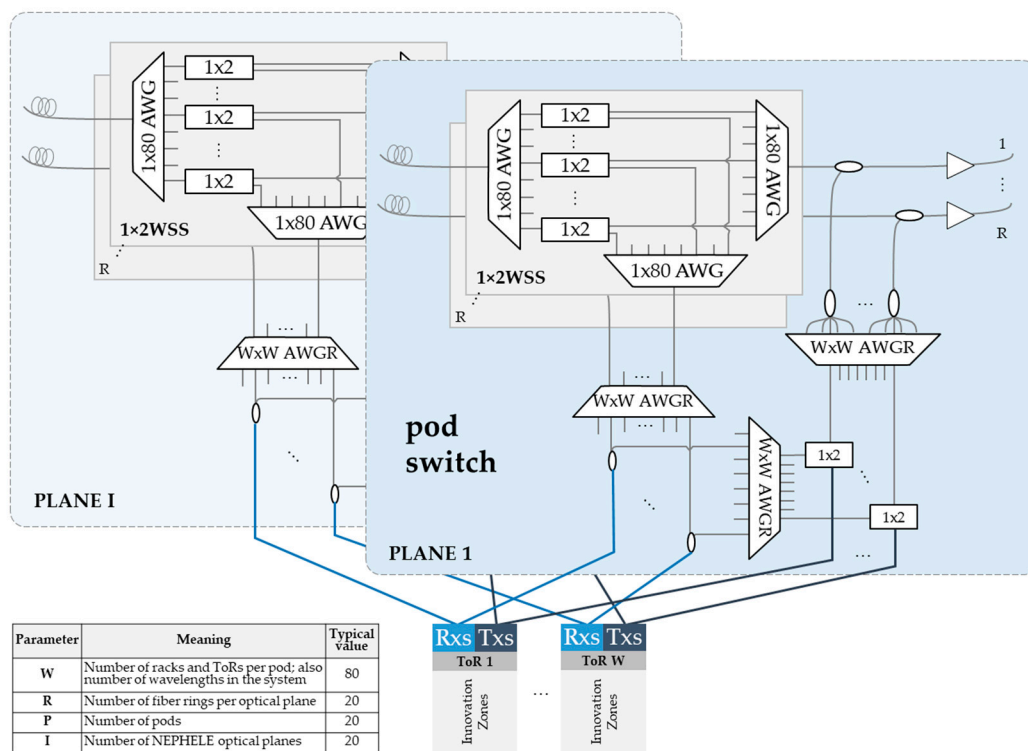


Figure 2. The pod switch schematic. Additional optical planes scale up the NEPHELE network.

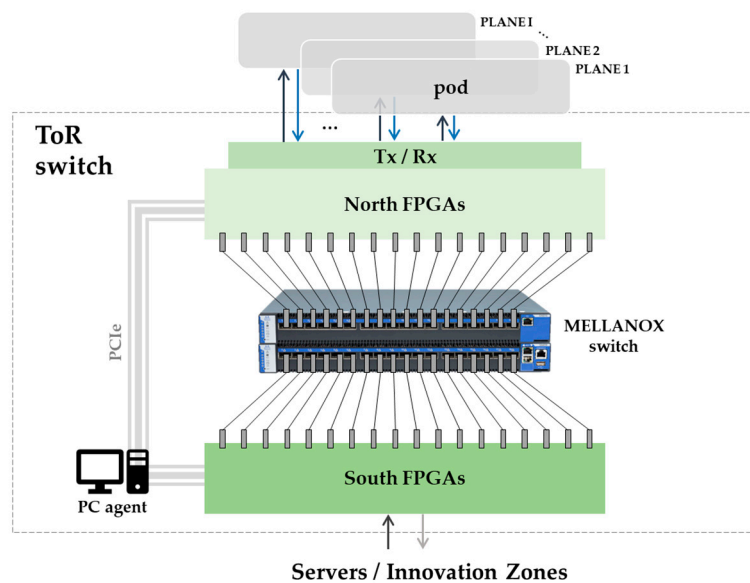
### 2.1.1. NEPHELE Pod Switch

The NEPHELE pod switch distinguishes the inter- and intra-pod traffic and is based on two types of active optoelectronic photonic modules: a wavelength-selective switch (WSS) based on the “demultiplex, switch, and multiplex” approach and a  $1 \times 2$  optical fast switch. The optical fast switches utilize the breakthrough OptoCeramic™ electro-optic materials developed by BATi for a variety of light control applications, and its nominal switching time is faster than 50 ns. A schematic of the pod switch is given in Figure 2. In the upstream direction (from the hosts to the optical network), the  $1 \times 2$  switches direct the traffic either to another ToR of the same pod (intra-pod) or to the WDM rings (inter-pod). In both cases,  $W \times W$  arrayed waveguide grating routers (AWGR) are used to route the signals to the appropriate destination. In the downstream direction (from the optical network to the hosts), the WSS drops wavelengths from the rings to the appropriate pods. The wavelengths are routed to the destination ToR through AWGRs. The combination of these modules with several passive filtering photonic elements within the network allows wavelength reuse among pods, enabling network scalability beyond the typical wavelength count of DWDM systems. The configuration of the active components of the pod is decided by the SDN controller of the network (Section 2.2) and was realized by Field Programmable Gate Arrays (FPGAs) during the experiments described in the sections below.

### 2.1.2. The NEPHELE Top-of-Rack Switch (ToR)

Each NEPHELE top-of-rack switch (ToR) interconnects the hosts in the datacenter racks, as well as to the higher network layer, handled by the pods. To support the slotted operation of the NEPHELE optical switching fabric, we developed functionality extenders of a commercial electrical ToR switch (Figure 3). The extenders are developed on FPGA boards and they are placed on the “south” (between the electrical switch and the servers) and on the “north” (between the electrical switch and the optical network) of a commercial electrical switch (Mellanox SX1024). This way, in a future NEPHELE DC, the innovation zones and the applications running on them will remain transparent to the optical network restrictions, while the NEPHELE ToR switch will undertake their seamless integration with the

slotted optical network. Note here that the FPGAs across the NEPHELE network are synchronized in time and, thus, the operations described in the subsequent subsections are coordinated in sub- $\mu$ s scale.



**Figure 3.** A schematic of the inside of a NEPHELE top-of-rack switch (ToR).

#### South FPGA (S-FPGA) Extender

The south FPGA resides between the servers and the electrical ToR switch. It receives the Ethernet frames generated from the servers in the innovation zones, parses the headers, and stores them in VOQs (virtual output queues) per destination ToR and input port. In order to efficiently utilize the available memory, the VOQs are dynamically created depending on the incoming traffic. The south FPGA forwards chunks of VLAN (virtual local area network)-tagged Ethernet frames with the same ToR destination (i.e., from the same VOQ) to the Ethernet switch. The south FPGA has a bidirectional communication channel with the control plane. It notifies the SDN controller and, therefore, the scheduler about the status of its VOQs in a periodic manner. In the opposite direction, the SDN controller sends the following instructions: (a) which VOQ will be emptied for each of the upcoming slots, and (b) a VLAN tag for each slot (which via the electrical ToR switch will define the outgoing port/plane of transmission on the optical network).

#### Legacy Electrical Switch

In the electrical switch, we use two different schemes of switching, depending on the direction of the traffic. The frames received from the south FPGA and, thus, with direction from the servers to the optical network (upstream) are switched based on their VLAN tag. The VLAN tag was inserted in the south FPGA according to the SDN controller’s instructions to steer the chunks of Ethernet frames to the appropriate NEPHELE plane. In the opposite direction, from the optical network to the servers (downstream), the switching is based on the destination MAC/IP address. This way, the frames that arrive at the ToR switch from the optical fabric are forwarded to the appropriate server/innovation zones.

#### North FPGA (N-FPGA) Extender

As described in the previous subsections, the north FPGA receives the chunks of Ethernet frames from the electrical switch. Its role is to handle the interfacing with the optics and to realize the slotted operation. The north FPGA includes a slot-sized FIFO (First-in, first-out) to fine-tune the timing on which data are sent on the optical network. The chunks of Ethernet frames are encapsulated in the NEPHELE frame; we added an  $\sim 8\text{-}\mu\text{s}$ -long preamble to facilitate the clock and the data recovery (CDR)

locking on the receiver, a delimiter, and a short header with management fields. On the receiving side, the Ethernet frames are decapsulated from the NEPHELE frame and they are forwarded to the electrical switch.

The key building block of the NEPHELE top-of-rack (ToR) switch is the wavelength-tunable transmitter (Tx), which consists of an FPGA-controlled tunable laser, an MZI modulator, and an RF driver [33]. The tunable Tx is responsible for imprinting the 10 Gb/s electrical data onto a selectable optical carrier (wavelength). The transmitter behaves in bursts and obeys the TDMA operation. The wavelength that each north FPGA uses in each slot is dictated by the SDN controller according to the destination ToR.

### 2.2. Control Plane Overview

The NEPHELE control and orchestration framework is applied on both the intra-DCN and the inter-DC domain. This means that the control plane is based on a centralized inter-domain network orchestrator, named NIDO [34] (NEPHELE inter-domain network orchestrator), which coordinates at multiple NEPHELE DCNs and the intermediate interconnection nodes to achieve end-to-end resource allocation [35]. More specifically, NIDO orchestrates the actions of lower-layer intra-domain SDN controllers, namely, OCEANIA [36] (Optical Electrical Application Aware data center network controller) for NEPHELE DCNs and JULIUS [37] for inter-DC communication, as depicted in Figure 4. This hierarchical approach enables changing the intra-domain controller and/or the related network technology of any domain (DC, core, metro, access network).

The intra-NEPHELE DCN is controlled through the OCEANIA controller, an SDN controller that extends the open-source OpenDaylight controller (Lithium version). Furthermore, OCEANIA works in-line with SDN applications, algorithms, and OpenFlow protocol extensions to efficiently operate a NEPHELE DCN. It caters for the dynamic establishment, re-configuration, and tear-down of intra-DC connections by optimizing the allocation of the space (planes) and time (time-slots) resources. In the development of the NEPHELE prototype, we achieved the integration of the OCEANIA SDN controller with the NEPHELE data plane at the southbound and the NEPHELE inter-DC orchestrator (NIDO) at the northbound. The OCEANIA controller was demonstrated successfully in Reference [29]. At the SDN application level, new resource allocation (scheduling) algorithms were implemented to enable an efficient and fully automated path allocation.

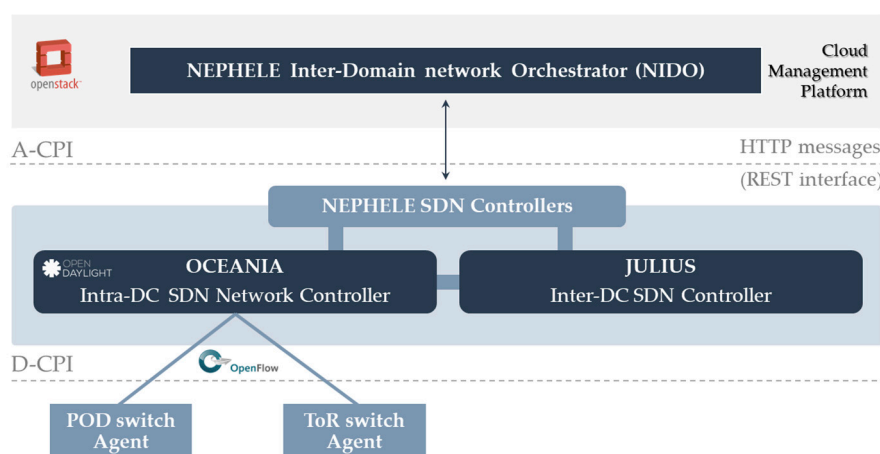


Figure 4. The NEPHELE inter-domain orchestration architecture.

The OCEANIA controller presents common characteristics with SDN controllers adopted in the context of optical DCNs [38–40]. At the southbound of the controller, OCEANIA adopts a similar abstraction approach for the data plane technologies, which allows the northbound SDN applications to work in a technology-agnostic manner. Moreover, OCEANIA implements a common

set of functionalities exposed through a REST-based (Representational state transfer) Application Programming Interface (API) toward the Cloud Management Platform, to enable the dynamic set-up and adjustment of lightpaths as integrated parts of the cloud service provisioning process. However, we can identify two key differentiators of OCEANIA controller. In terms of internal information models, OCEANIA adopts a NEPHELE-specific modeling of the optical resources, based on the possibility of allocating resources at the space and time level. This model is also reflected on the OpenFlow protocol extensions adopted between the OCEANIA and the SDN Agents of ToR and pod switches to configure the optical devices. Moreover, OCEANIA implements scheduling algorithms specifically designed for the NEPHELE DCN topology, based on loops of pods interconnected to ToRs and the related granularity enabled for the configuration of optical resources.

Extra effort was put to develop scheduling algorithms that would be efficient in large DCNs but also perform fast calculations to enable rapid network reconfiguration [30,41]. The NEPHELE data plane operates in a synchronous slotted time division multiple access (TDMA) manner. To enable efficient utilization of the network resources, a central scheduler dynamically assigns timeslots and creates end-to-end light-paths based on the traffic requirements. However, making the scheduling decisions on a per-timeslot basis, for hundreds or even thousands of end-nodes, would require high communication and processing latency. It is more efficient to perform resource allocation periodically, so that scheduling decisions are made for periods of  $T$  timeslots.

More specifically, the ToR switches periodically report the status of their queues to the controller. The controller translates the status of the queues into ToR-to-ToR communication requests and constructs the traffic matrix (TM). The scheduling algorithm takes as input the communication requests of the TM and allocates the network's resources (optical planes and wavelengths) to source–destination ToR pairs in a per timeslot basis. The resource allocation problem is essentially a matrix decomposition problem and can be solved optimally using the Birkhoff–von Neumann decomposition. Even though this approach ensures maximum utilization of the network's resources, the complexity of the optimal algorithm prohibits real-time execution [30]. In the context of NEPHELE, several scheduling approaches were proposed and evaluated: from linear, sublinear, and randomized greedy heuristics [30] to parallel implementation on FPGAs [41]. Note here that two different resource allocation approaches were proposed [30]: (a) offline and (b) incremental. The offline algorithms compute the solution from scratch based only on the new input TM, while the incremental ones update the solution of the previous scheduling period based on the traffic changes. For the demonstration scenarios presented in the upcoming sections, we used the offline linear greedy algorithm.

Regarding the TDMA network's throughput, a detailed study can be found in Reference [30], with special focus to the NEPHELE network and its variations. The maximum network throughput is defined as the maximum offered load at which the queues and the latency are finite. In short, the normalized and greedy heuristics scheduling approaches achieved a normalized throughput higher than 85% for a variety of scenarios (traffic locality, architecture variations, etc.).

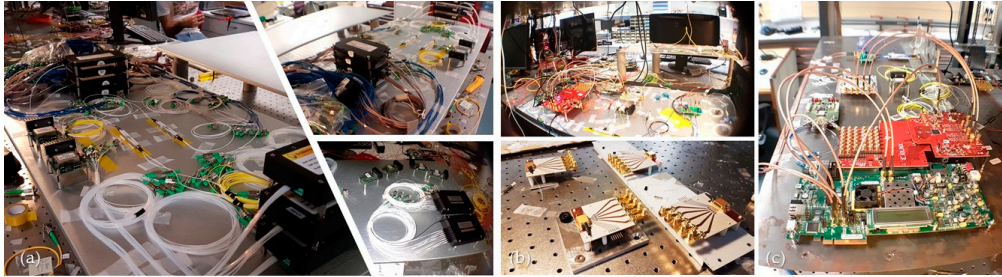
### 3. The NEPHELE Demonstrator Assembly

The NEPHELE demonstrator set-up was built step-by-step at the Photonics Communications Research Laboratory in the National Technical University of Athens [42]. Photos of the demonstrator during the preparation period are shown in Figure 5 and the implemented set-up is graphically depicted in the schematic of Figure 6. In this set-up, two NEPHELE optical planes with two pods are emulated. More specifically, pod 1 accommodates a server with two independent 10 Gb/s Ethernet interfaces/hosts (E1 and E2) through ToR 1 and a dummy ToR switch (ToR 0) for monitoring purposes. On the other hand, pod 2 handles two servers, each one accommodating two hosts through ToR 2 (E3 and E4) and ToR 3 (E5 and E6).

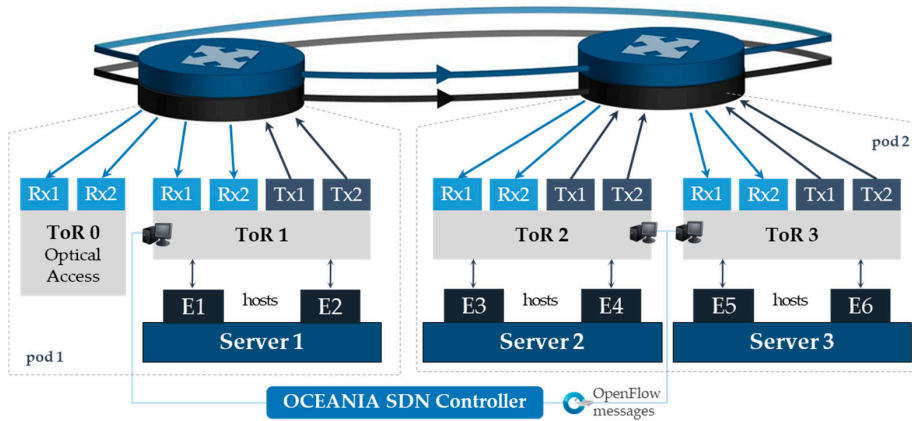
The NEPHELE demonstrator was assembled according to the NEPHELE architectural principles, in lab-scale dimensions, for the purposes of the demonstration (1st supplementary vide). Each of the NEPHELE ToR switches is equipped with two tunable transmitters (Tx) and an equal number



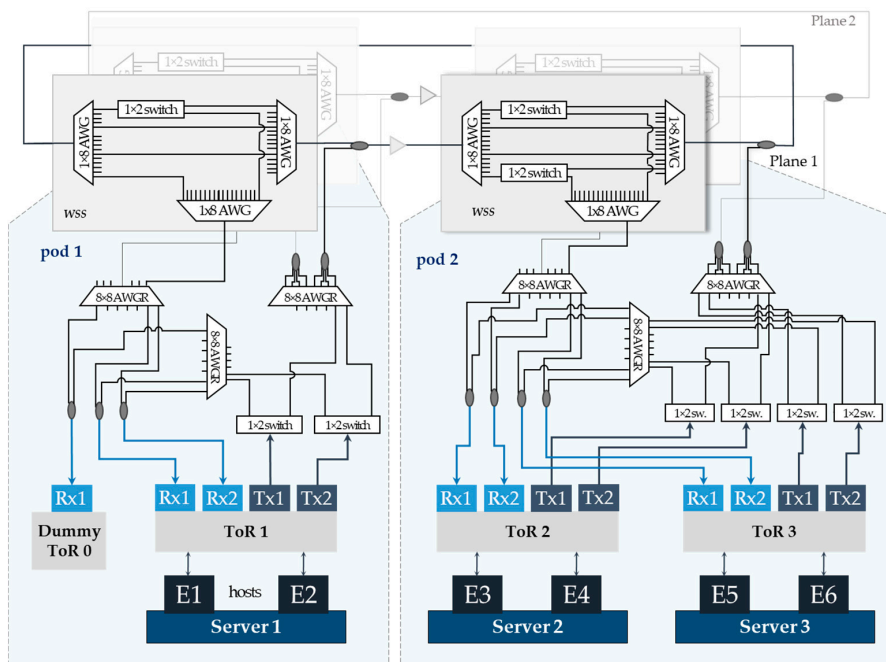
of optical receivers (Rx), as depicted in Figure 7. In this context, the demonstrator is composed of three fully equipped ToR switches, i.e., six transmitter and receiver assemblies that were developed, tested, and used for the demo [27,28,33]. Finally, a partially functional receiver was connected to the “dummy” ToR 0, which is in pod 1 and it was used for displaying purposes.



**Figure 5.** Photos of the NEPHELE demonstrator during the preparation period. (a) The pod switch, including a wavelength-selective switch (WSS) with the “demultiplex, switch, and multiplex” approach. (b) The fast tunable lasers used for the transmitter. (c) The FPGA boards that assemble the ToR switch.



**Figure 6.** The NEPHELE optical network used for the real-time demonstration.

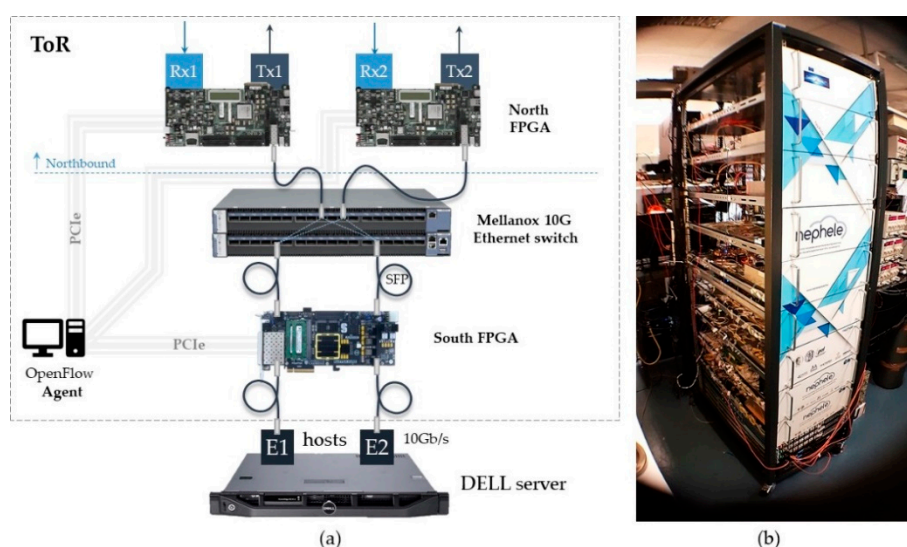


**Figure 7.** The complete experimental set-up used for the live demonstration.

Figure 8a shows a detailed schematic of the internal connections between the ToR switch and the hosts. Starting the description from the bottom to the top, three Dell servers were used as hosts inside the racks, serving as the innovation zones. Each of the servers can produce 10 Gb/s Ethernet traffic and they are connected to the south FPGA (S-FPGA) by means of two 10 Gbps SFP interfaces (Small form-factor pluggable transceiver) [43]. The interfaces are independent and, as a result, they can emulate different hosts. Indeed, server 1 located in pod 1, handled by ToR1, is connected to the S-FPGA by E1 and E2 Ethernet interfaces. After that, the traffic is forwarded to the Mellanox Ethernet switch, which sends the frames to the available north FPGA (N-FPGA) for transmission. The N-FPGAs control the tunable Tx and encapsulate the NEPHELE frames (200  $\mu$ s duration, including 10  $\mu$ s guard time). In the Tx, the NEPHELE frames (in the electrical domain) are amplified in an Radio Frequency (RF) driver and, afterward, they are introduced to a Mach–Zehnder modulator. The optical carrier is provided by a tunable laser assembled on a custom carrier, which is also fully controlled by the N-FPGA. The tunable laser is a Finisar S7500 MG-Y laser which is controlled by five input currents applied to five corresponding sections: Semiconductor Optical Amplifier (SOA), gain, phase, left reflector, and right reflector section. To enable fast wavelength tuning, the laser was assembled on a custom board with impedance matched RF interfaces. The three driving currents were provided to the board by a current digital-to-analog converter evaluation module embedded in an FPGA, as an extension. The tuning speed of the tunable laser was measured in the range of 17–23 ns [28] for all 80 wavelengths of the ITU grid, well within the NEPHELE specifications. The reception of the NEPHELE flows follows the opposite vertical direction.

From the physical layer perspective, the communication between the servers in the NEPHELE network is achieved by using different wavelengths to route the traffic, leveraging the combined effect of the fast tunable lasers and the AWGRs. On the other hand, regarding the network/protocol tier, IP addresses are assigned to each Ethernet interface. Table 1 shows the wavelengths and IP addresses that correspond to each interface according to the hosting pod and ToR.

With respect to the control plane, the SDN controller and the scheduler of the DC are running on a remote machine that is connected to the NEPHELE testbed via a 1 Gbps local area network. The SDN controller is connected with the SDN agents, which run on desktops that are also connected to the north and south FPGAs (through the Peripheral Component Interconnect Express (PCIe) interface). The control plane’s instructions are calculated, transferred, and enforced by the FPGAs in real time for the NEPHELE DC demonstration.



**Figure 8.** (a) Detailed schematic of the connections from the lower level of the host to the higher level of transmitter and receiver inside each ToR. (b) Fully populated rack, containing the optical assemblies, as well as the Mellanox Ethernet switches and dell servers serving as innovation zones.

**Table 1.** Wavelengths and IP addresses that correspond to each interface according to the hosting pod and ToR.

Destination ToR	Located Pod	Wavelength $\lambda$ (nm)	Interface	IP Address
ToR 1	1	1546.917	E1	10.1.1.1
			E2	10.1.1.129
ToR 2	2	1547.715	E3	10.2.2.1
			E4	10.2.2.129
ToR 3	2	1548.515	E5	10.2.3.1
			E6	10.2.3.129
ToR 4	1	1549.315	Dummy ToR	

### A “Day” in the Life of a NEPHELE Packet

The current subsection gives a high-level description of a packet’s journey in the NEPHELE network, aiming to provide a better understanding of the end-to-end data transmission. Each end-host is assigned with a static IP (Internet Protocol) address using DHCP (Dynamic Host Configuration Protocol) or OpenFlow. Each rack is an IP subnet; thus, the IP addresses of all the hosts in the rack begin with the same prefix (we use 24-bit prefix). The host address within the IP subnet is the index of the NIC (Network Interface Card) within the ToR.

Ethernet frames are transmitted from the NIC/host to the input ports of the ToR (ToR1 in Figure 9) through the S-FPGA. These frames carry the DMAC (Destination Media Access Control) address of the destination host. Static ARP (Address Resolution Protocol) can be used for mapping the destination host IP address to MAC address. The static ARP tables should be provided by the SDN controller. The frames sent from the hosts reach the ToR and are stored in buffers sorted by input port and destination ToR pair. These buffers for a specific input port are drawn in Figure 9 within the InP1 box. The schedule provided by the SDN controller (2nd supplementary video) and held by the ToR is executed slot after slot. The SDN controller’s instructions for each slot dictate which buffer (Input Port, Destination ToR) to send at the current time, the wavelength to be used, and through which plane the frame will be routed. If there are frames in that buffer, they are sent to the ToR Ethernet switch after they are tagged with a VLAN tag that matches the destination plane defined by the schedule. The ToR Ethernet switch forwarding inter-ToR traffic is statically configured using OpenFlow. Each VLAN tag is mapped to a different “north port” connected to a pod switch. In this way, the incoming VLAN-tagged frames to the Ethernet switch are forwarded to the correct plane.

On their way to the tuneable laser that drives the pod switch on that plane, the Ethernet frames are stripped from the VLAN tag and are encapsulated in a NEPHELE frame. Note that the entire ToR switch cannot be classified as a VOQ switch since frames are being stored per input port per destination ToR switch (to consider this as a VOQ switch, the output ports should be the ToRs and not the planes as in our case). The pod switch is also controlled by an OpenFlow agent and is performing its pre-programmed schedule. Each of the schedule slots defines the state of the fast switch and the WSS (for each color). According to the current slot, the pod 1 is aware whether this frame concerns inter-pod or intra-pod traffic. In the case of inter-pod traffic, the frame is routed via the  $1 \times 2$  fast switch and the AWGR toward the optical ring.

In Figure 9 inter-pod communication is illustrated and pod 2 is instructed to drop the wavelength that carries the data from ToR1 in the slot that the communication takes place. The arriving frames are stripped of their NEPHELE header before they reach the Ethernet switch on the destination ToR2. The Ethernet switch in ToR2 is statically configured by OpenFlow to forward intra-ToR traffic using the DMAC that is incorporated in the Eth header. In this way, the Ethernet frames are forwarded to the right port (NIC) depending on the DMAC. Optionally, the forwarding could be realized using the IP

address of the destination NIC. The same forwarding rules of the Ethernet switch allow the forwarding of the intra-ToR traffic within ToR1.

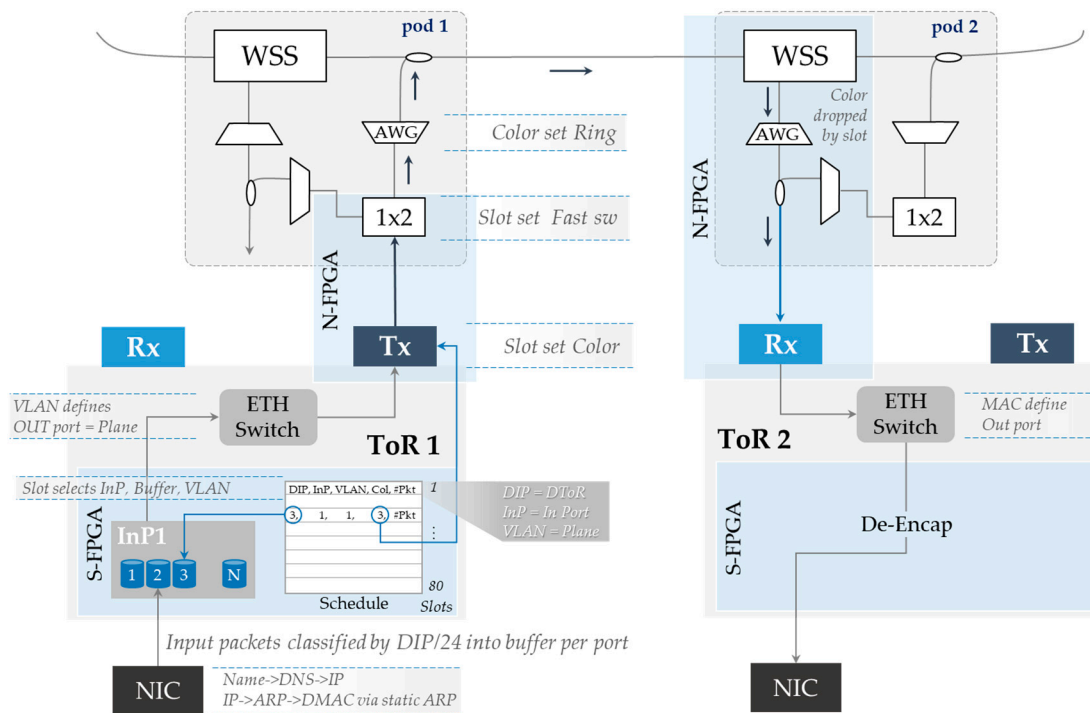


Figure 9. A “day” in the life of a NEPHELE packet.

#### 4. End-to-End Communication Scenarios and Real-Time Demo Results

In this section, we present the results of the end-to-end real-time operation of the NEPHELE demo DCN. More specifically, the intra-NEPHELE demo DCN communication scenarios are presented in Section 4.1, while an inter-NEPHELE Demo DCN interconnection between the Athens demo and the Pisa testbed in Italy is presented in Section 4.2.

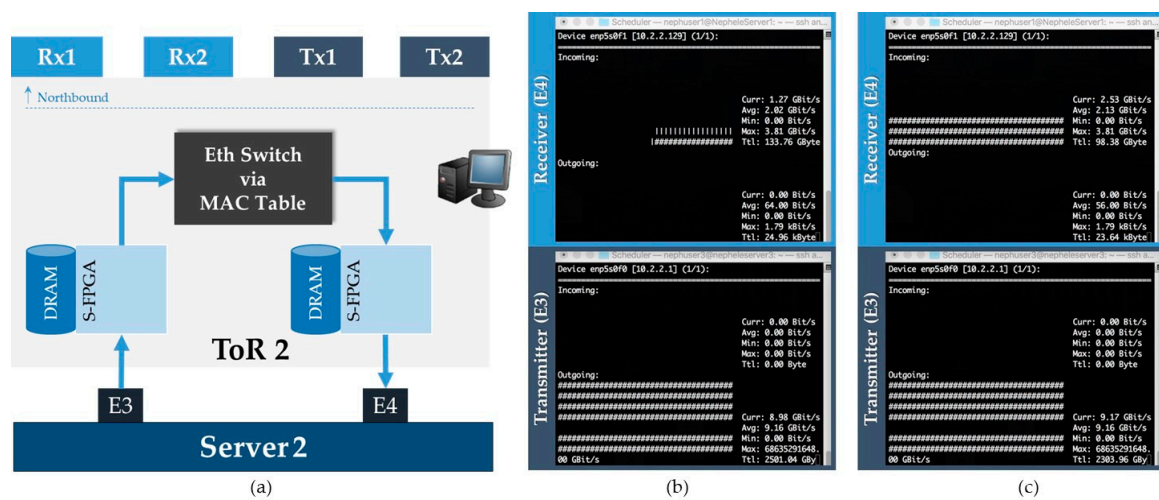
##### 4.1. Intra-Datacenter Communication

The intra-DC communication section consists of the communication scenarios that are taking place within the mini NEPHELE DCN. Taking into account the NEPHELE demonstrator of Figure 6, the scenarios are categorized as follows: intra-ToR, intra-pod, and inter-pod communication scenarios, depending on the location of the communicating hosts within the DCN.

##### 4.1.1. Intra-ToR Scenario

The first scenario demonstrated was the intra-ToR traffic case. In this scenario, the Ethernet traffic remains inside the same rack and, as a result, the northbound part of the ToR switch is not involved. Figure 10a shows the block diagram of the demo set-up, and the blue arrows indicate the traffic flow paths, from the E3 to E4 Ethernet interface within the ToR 2.

In Figure 10 b,c the *nload* command is depicted running simultaneously in all the involved host-servers. In this way, the incoming and outgoing traffic on each server is monitored. More specifically, in Figure 10b, the communication between server E3 and E4 is shown; server E3 (10.2.2.1) is sending data traffic to server E4 (10.2.2.129). The bandwidth achieved in this case is almost 1.3 Gbit/s. In Figure 10c, the traffic follows the same path but more slots are allocated by the NEPHELE scheduling engine and, as a result, the bandwidth is increased by a factor of almost two. In the described scenario, no optical part of the pod switch is used. The traffic is handled by the south part of the ToR switch, since the involved hosts belong to the same rack.



**Figure 10.** (a) The data plane structure of the ToR 2 for the intra-ToR scenario. (b) Server E3 (10.2.2.1) is sending data traffic to server E4 (10.2.2.129) reaching 1.3 Gbit/s bandwidth. (c) By allocating more slots, the bandwidth is doubled.

#### 4.1.2. Intra-Pod Scenario

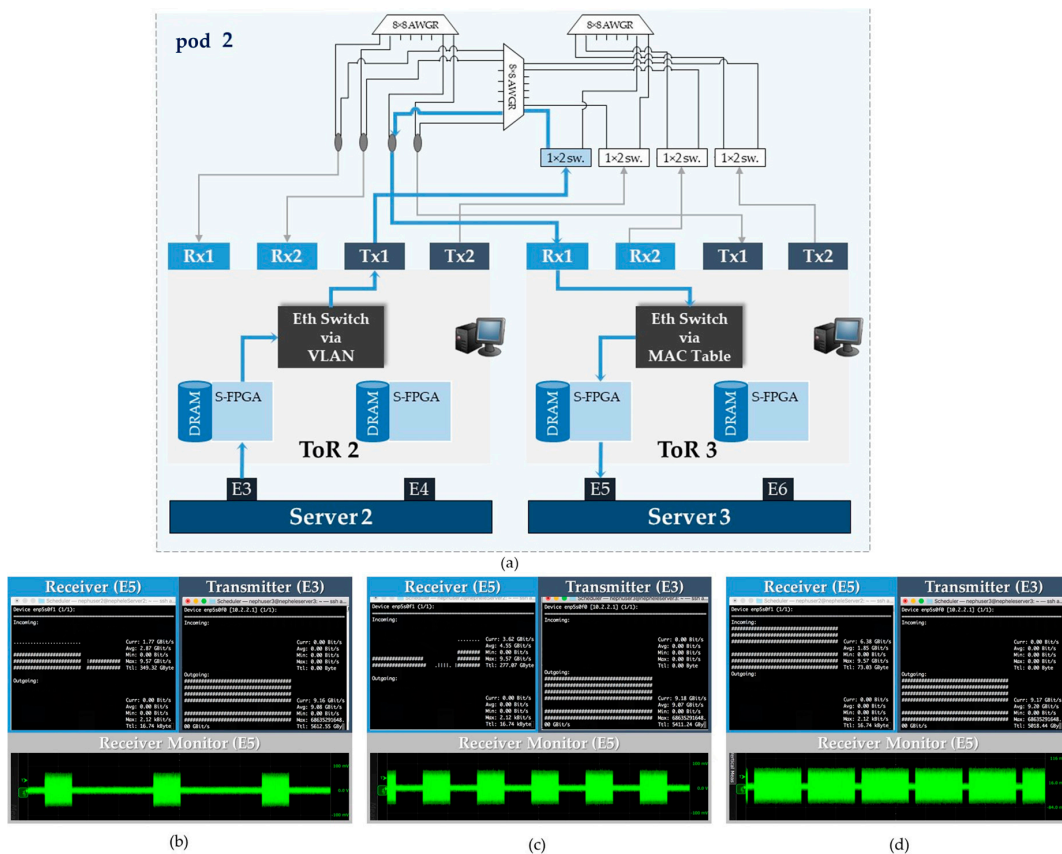
The second scenario implemented in the demo was focused on the intra-pod communication. In this scenario, the two ToRs, which reside within the pod 2, are establishing a communication path. The generated traffic originates from E3 host of ToR 2, and its destination is the E5 host that belongs to ToR 3. The experiment was carried out with three different traffic patterns: 20, 40, and 70 slots of the total 80 of the NEPHELE scheduling period. In this way, the bandwidth is consecutively increased. The commands for increasing the allocated slots are sent by the SDN controller to the FPGAs that enforce the instructions in real time.

This experiment involves three data plane modules: two ToR switches and one pod switch. On Figure 11a, a schematic diagram of the traffic route through the data plane is presented. Following the blue arrows, the frames originating from server E3 are first buffered in the S-FPGA of ToR 2. After that, the traffic is routed through the Mellanox Ethernet switch, which selects the right N-FPGA as the destination gateway to the optical rings of the NEPHELE network.

The tunable transmitter driven by the N-FPGA of ToR 2 is tuned to transmit the traffic enrolled in the correct wavelength according to the destination ToR; in this case,  $\lambda_3 = 1548.515$  nm (Table 1). The transmitter is followed by a  $1 \times 2$  optical switch which is responsible for handling the frames in accordance with their destination place (intra-pod or inter-pod). In this case, the  $1 \times 2$  optical switch, which is driven according to the schedule by the N-FPGA, routes the frames to the output that routes to the same pod. The cyclic  $8 \times 8$  AWG passively forwards the frames to the fiber that reaches the receiver embedded to the N-FPGA of ToR 3. It is important to mention that, due to the system optimization in the optical path (and in all intra-pod optical paths), there is no need for optical amplification.

After the successful optical reception of the NEPHELE frames by the N-FPGA of ToR 3, the extracted Ethernet frames are forwarded to the Mellanox switch, which in turn routes the Ethernet frames to the correct S-FPGA using the destination’s preregistered media access control (MAC) address. The S-FPGA forwards, without any additional delay or processing, the Ethernet frames to the destination E5.

The “monitor” images at the bottom of Figure 11b–d show screenshots of the NEPHELE frames traveling through the network taken by means of a real-time oscilloscope. In these figures, the screenshots represent different percentages of slot allocation. It is useful to repeat that the NEPHELE scheduling period is divided into 80 slots (200  $\mu$ s each slot) with a guard time (10  $\mu$ s) between them. Apparently, the increase in slots that are allocated for a specific connection results in a direct increase of the available bandwidth between the two endpoints. Figure 11b–d presents the *nload* command running on all the involved servers. The bandwidth scales according to the slot allocation.



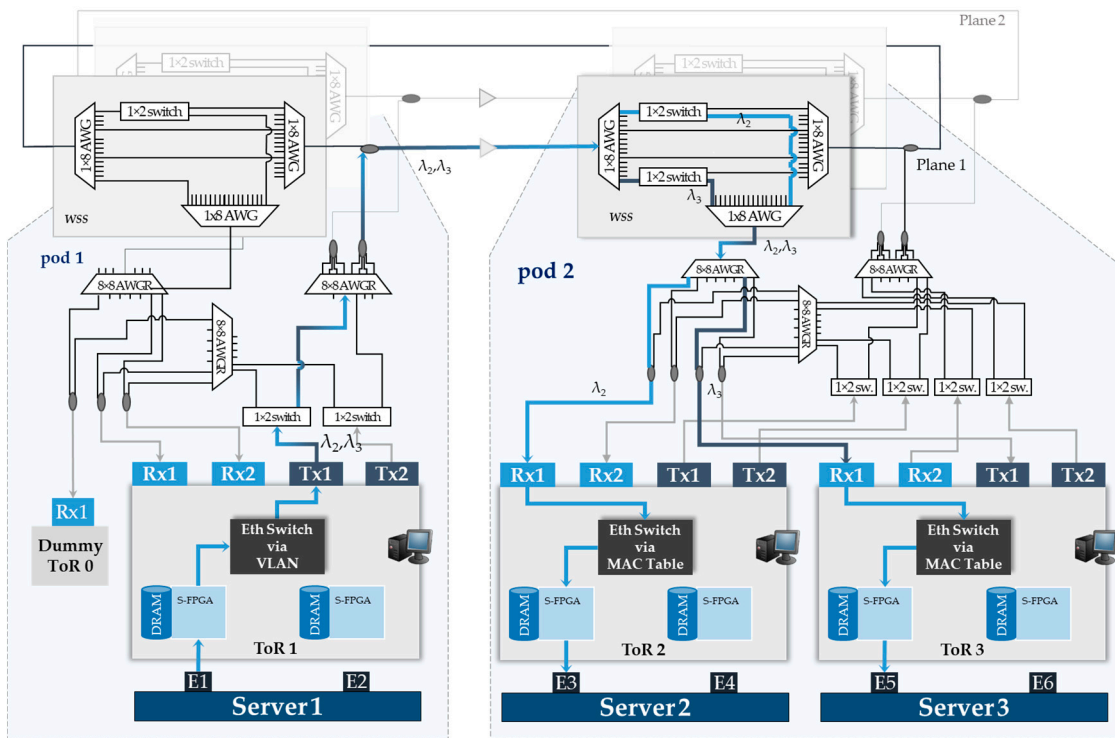
**Figure 11.** (a) The data-plane structure of the intra-pod scenario. The NEPHELE frames follow the route highlighted with the blue arrows. (b–d) Screenshots of the NEPHELE frames sent from ToR2 server E3 (10.2.2.1) and received by ToR3 server E5 (10.2.3.1). (b) In this case, 25% of the scheduling period (20 slots out of 80 slots) is occupied for the communication path between ToR2 and ToR3 achieving 1.8 Gbit/s bandwidth. (c) In this half of the scheduling period, 40 slots out of 80 slots are occupied for the communication path between ToR2 and ToR3 achieving 3.6 Gb/s bandwidth. (d) In this case, 87.5% of the scheduling period (70 slots out of 80 slots) is occupied for the communication path between ToR2 and ToR3 achieving 6.4 Gbit/s bandwidth.

#### 4.1.3. Inter-Pod Scenario I

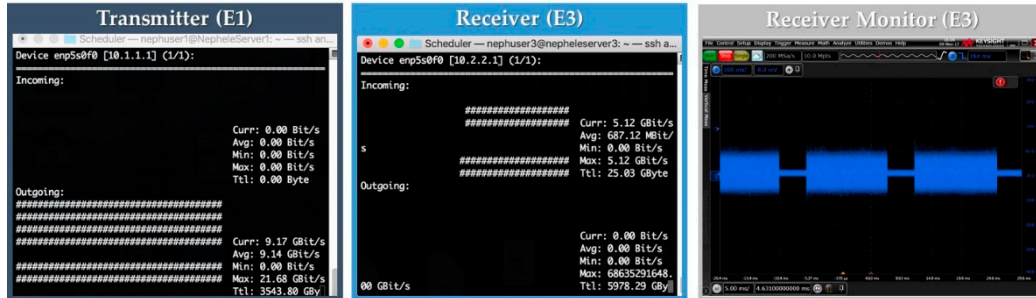
The experimental set-up used for the inter-pod communication is shown in Figure 12 (for this section, we assume only the light-blue arrows in the figure, while the dark-blue ones correspond to the scenario in Section 4.1.4). In this case, server E1 which resides inside ToR 1 of pod 1 establishes connection with server E3 which is connected to ToR 2 of pod 2. Thus, in this scenario, two pod switches and two ToR switches are involved. Moreover, the implemented traffic pattern includes 60 out of the 80 slots, which is the total NEPHELE scheduling period.

The tunable transmitter is tuned to emit the wavelength which will reach ToR 2, i.e., the wavelength  $\lambda_2 = 1547.715$  nm. The transmitted traffic passes through all the optical components of two consequent pod switches:  $1 \times 2$  switch (intra-, inter-),  $8 \times 8$  AWGR, EDFA, WSS,  $8 \times 8$  AWGR, and a coupler.

Figure 13 depicts the NEPHELE packets screenshots taken by means of a real-time oscilloscope and a constant slot allocation (60 out of 80 slots). In addition, it presents the *nload* command running on the servers and shows that the achieved bandwidth between E1 and E3 is 5.12 Gbit/s.



**Figure 12.** Experimental set-up for the inter-pod scenario; dark- and light-blue lines reveal the optical paths from source host (E1) to destination hosts E3 and E5, respectively.



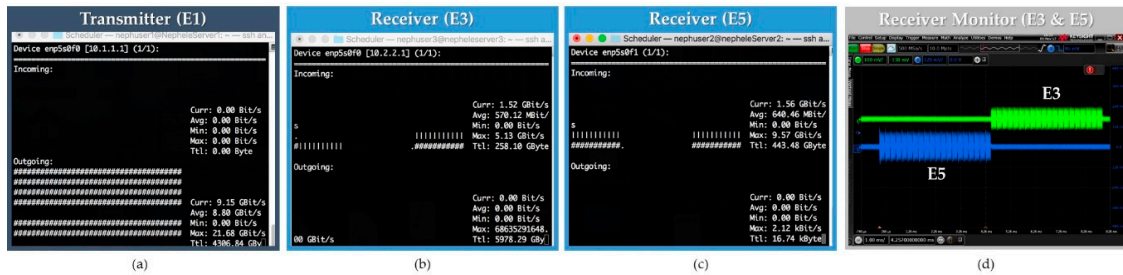
**Figure 13.** The NEPHELE packets sent from the E1 (10.1.1.1) ToR1 and received by the E3 (10.2.2.1) ToR2. In this case, 75% of the scheduling period (60 slots out of 80 slots) is occupied for the communication path between ToR1 and ToR2 achieving 5.12 Gb/s bandwidth.

#### 4.1.4. Inter-Pod Scenario II

In this scenario, the traffic follows an inter-pod route as well. In this case, however, the server of ToR1, which resides inside pod 1, sends traffic to two different ToRs in pod 2. More specifically, server E1 (ToR 1-pod 1) generates and sends traffic to server E3 (ToR 2-pod 2) and E5 (ToR 3-pod 2) alternately. The source server (E1) transmits Ethernet frames for both destinations concurrently, while the scheduling commands, sent by the SDN controller, impose the slot allocation for the scheduling period. The experimental set-up for the inter-pod scenario with two destination ToRs is depicted in Figure 12. The traffic originating from the E1 interface is optically transmitted by Tx1 of ToR 1 and it is forwarded into the optical ring of plane 1 by means of a 1 × 2 optical switch (intra- or inter-pod) and an 8 × 8 AWGR. As mentioned above, the tunable transmitter Tx1 creates two traffic flows enrolled in two wavelengths alternately according to the targeted ToR;  $\lambda_2 = 1547.715$  nm for ToR 2 and  $\lambda_3 = 1548.515$  nm for ToR 3. An optical amplifier (EDFA) is used to compensate for the optical losses between the two pods.

The wavelength-selective switch (WSS) of pod 2, which is the responsible module for dropping specific wavelengths from the WDM rings, forwards the optical flows towards its ToRs according to the SDN controller’s commands. The switches inside the WSS are driven by the N-FPGA with TTL signals. An additional passive AWGR, as shown in Figure 12, finally routes the traffic to ToR 2 and ToR 3.

The green trace depicted in Figure 14 d represents the NEPHELE frames that reach ToR 2, and the blue one represents the corresponding frames which are received by ToR 3. It is obvious that the NEPHELE traffic is divided into two equal parts for each destination and, more specifically, 25% of the scheduling period (20 slots out of 80) is occupied for each of the two ToRs (Figure 14a–c). Finally, the bandwidth reached almost 1.4 Gb/s at both servers E3 and E5, as shown in Figure 14.

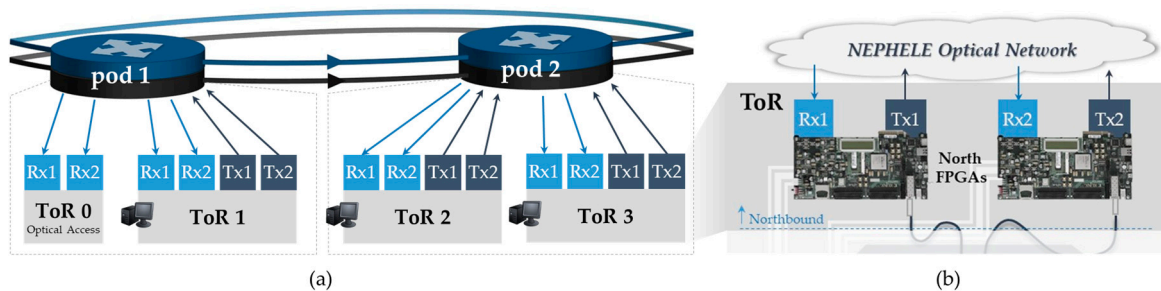


**Figure 14.** (a) Server E1 (10.1.1.1) is sending data traffic to (b) server E3 (10.2.2.1) and (c) server E5 (10.2.2.129), achieving almost 1.4 Gb/s bandwidth at each server. (d) Screenshots of the NEPHELE frames originating from ToR1 and captured in ToR 2 (blue trace) and ToR 3 (green trace). The total traffic is equally divided into two destination ToRs (20 slots reaching each ToR).

#### 4.1.5. Combined Intra-Pod and Inter-Pod Communication Scenarios

The following sections present additional, more complex scenarios that were tested on the developed innovative NEPHELE mini-datacenter testbed. Extra functionalities were implemented in order to confirm the system stability despite the additional complexity.

To begin with, the scenarios that are presented in the next sections were carried out on the same demo testbed (Figure 15a) with a main difference from the previous ones; communication between the ToRs takes place from the northbound interfaces of each ToR switch as depicted in Figure 15b, since the end-to-end real time communication was proven in previous sections. This means that the modules located vertically below the north FPGAs are not involved in these experiments.



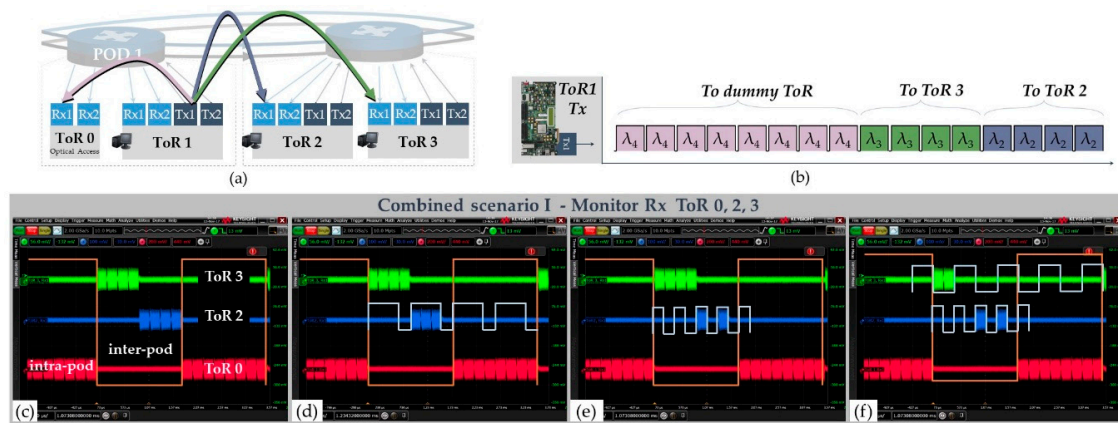
**Figure 15.** (a) The experimental set-up used for the evaluation of the combined Intra-pod and inter-pod scenarios. (b) The northbound data plane interfaces schematic of the ToR switches.

More specifically, the N-FGPA generates “dummy” NEPHELE frames filled with pseudorandom binary sequence (PRBS) instead of Ethernet data. This dummy traffic meets all the specifications of the NEPHELE architecture with respect to timing and rate. Furthermore, the north FPGA generates and provides the control signals (TTL) for the optical modules which are involved: 1 × 2 optical switches for the intra- and inter-pod communication and 1 × 2 WSS (the “demultiplex, switch, and multiplex” approach of the WSS module includes multiple 1 × 2 optical switches).



### Combined Scenario I

The first scenario which combines intra-pod and inter-pod traffic is shown in Figure 16a. According to this, ToR 1 of pod 1 sends traffic to the ToR 0 of pod 1 (intra-pod communication) and to ToR 2 and ToR 3 of pod 2 (inter-pod communication). Hence, the tunable transmitter of ToR 1 is set to transmit 16 repeated NEPHELE frames enrolled in different wavelengths (in  $\lambda_4$ ,  $\lambda_2$ , and  $\lambda_3$ ) as shown in Figure 16b.



**Figure 16.** (a) Combined scenario I. (b) The repeating sequence of NEPHELE frames in three wavelengths produced by the tunable Tx of ToR 1: eight frames in  $\lambda_4$ , four in  $\lambda_2$ , and four in  $\lambda_3$ . (c–f) Screenshots of the received NEPHELE frames to each ToR receiver; the red trace corresponds to the ToR 0 receiver, while blue trace corresponds to the ToR 2 receiver, and the green trace corresponds to the ToR 3 receiver.

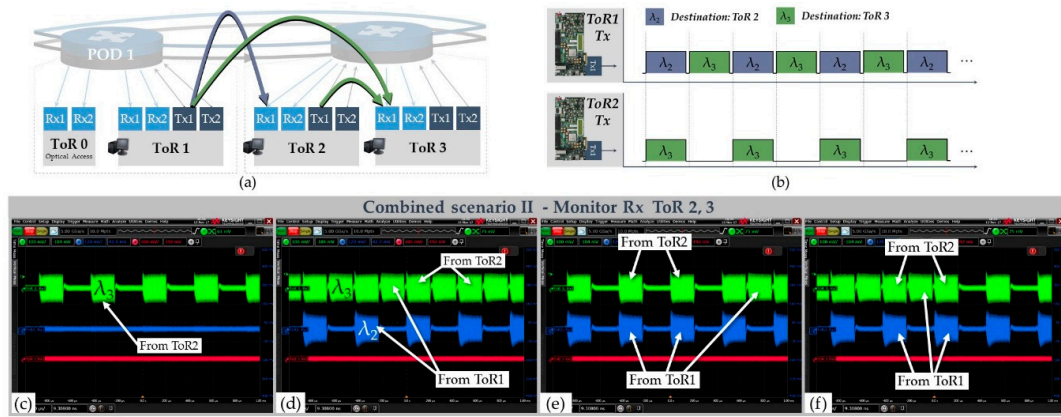
Figure 16c–f portrays the screenshots captured after the receivers by means of a real-time oscilloscope. The oscilloscope’s channels are connected to each of the three photoreceivers; the red trace corresponds to ToR 0, the blue one corresponds to ToR 2, and the green one corresponds to ToR 3. In all the screenshots (c–f), the orange trace represents the driving signal that is applied to the  $1 \times 2$  optical switch which distinguishes the intra- and inter-pod traffic in pod 1. In addition, the white traces in (d–f) screenshots represent the driving signal for the  $1 \times 2$  optical switches which are located inside the WSS of pod 2 and correspond to  $\lambda_2$  and  $\lambda_3$ .

### Combined Scenario II

The second combined scenario that was implemented is presented in Figure 17a. As shown, ToR 1 of pod 1 transmits NEPHELE frames to ToR 2 and ToR 3 of pod 2 (inter-pod communication). At the same time and synchronously, the transmitter of ToR 2 sends traffic to ToR 3 (intra-pod communication). As a result, the receiver of ToR 3 receives traffic from both ToR 1 and ToR 2.

The traffic generated by ToR 1 (pod 1) and ToR 2 (pod 2) is depicted in Figure 17b. The tunable transmitter of ToR 1 is able to produce a sequence of NEPHELE frames at  $\lambda_2$  and  $\lambda_3$ . Similarly, the ToR 2 transmitter is programmed to generate NEPHELE frames at  $\lambda_3$  with an empty slot between them in order to avoid any possible conflict with  $\lambda_3$  frames transmitted by ToR1.

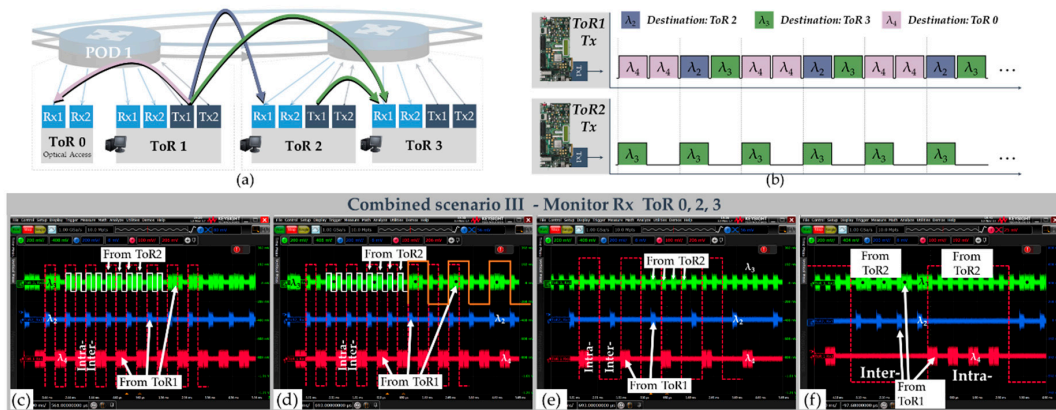
Combinations of the above-mentioned traffic flows are presented in the screenshots (c–f) of Figure 17. Firstly, Figure 17c shows the NEPHELE frames at  $\lambda_3$  that reach ToR 3 and are transmitted by ToR 2, without any other transmitted traffic. Afterward, the transmitter of ToR 1 starts sending NEPHELE frames at  $\lambda_2$  and  $\lambda_3$  as observed in Figure 17d. In this case, the  $1 \times 2$  switch of pod 1 forwards the traffic to the inter-pod path and the WSS of pod 2 drops the two wavelengths. The next two screenshots (Figure 17e–f) show cases in which the WSS of pod 2 drops some of the frames at  $\lambda_3$ .



**Figure 17.** (a) Schematic for the combined intra-pod and inter-pod scenario II. (b) Repeating sequence of NEPHELE frames transmitted by ToR 1 and ToR 2 simultaneously and synchronously at  $\lambda_2$  and  $\lambda_3$ . (c–f) Screenshots of the received NEPHELE frames to ToR 2 and ToR 3 receivers; blue and green traces represent the receivers at ToR 2 and ToR 3, respectively.

### Combined Scenario III

The third and final scenario that was carried out on the NEPHELE demo testbed is a combination of the two previous scenarios, and it is shown in the schematic of Figure 18a. The traffic produced by ToR1 is both intra-pod (ToR 0-pod 1) and inter-pod (ToR 2 and ToR 3-pod 2). In the meantime, ToR 2 generates another intra-pod traffic flow, but this time inside pod 2 by transmitting NEPHELE frames with the destination as ToR 3. In addition, three wavelengths are involved ( $\lambda_2$ ,  $\lambda_3$ , and  $\lambda_4$ ) in this scenario and, obviously, the two transmitters are synchronized.



**Figure 18.** (a) Schematic for the combined intra-pod and inter-pod scenario III. (b) Repeating sequence of NEPHELE frames transmitted by ToR 1 and ToR 2 simultaneously and synchronously at  $\lambda_4$ ,  $\lambda_2$  and  $\lambda_3$ . (c–f) Screenshots of the received NEPHELE frames to each ToR receiver; the red trace corresponds to the dummy ToR receiver, the blue trace corresponds to the ToR 2 receiver, and the green trace corresponds to the ToR 3 receiver.

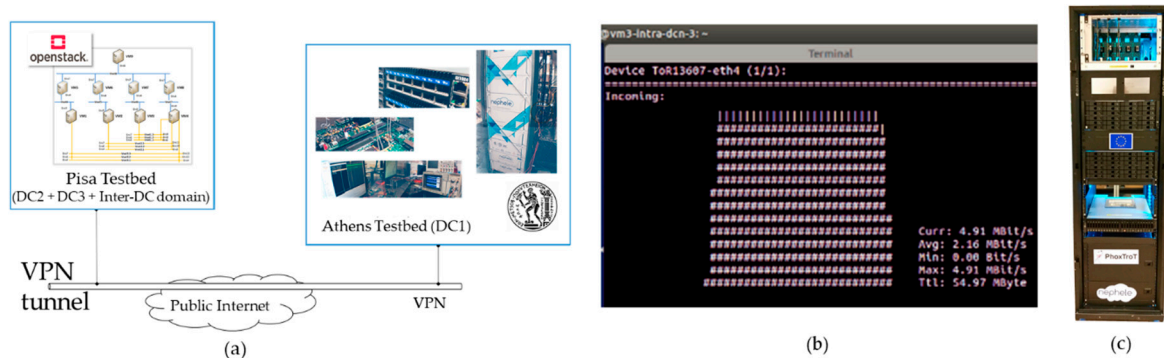
The traffic that is generated by the two tunable transmitters is depicted in Figure 18b. The transmitter of ToR 1 (pod 1) creates a sequence of NEPHELE frames enrolled in  $\lambda_4$ ,  $\lambda_2$ , and  $\lambda_3$ . As it happened in the previous combined scenario, the ToR 2 (pod 2) transmitter is programmed to generate NEPHELE frames in  $\lambda_3$  with an empty slot between them.

Figure 18c–f represents the screenshots of the NEPHELE frames that reach the three destination ToRs. The red frames represent the ones at  $\lambda_4$  that are received by the ToR 0 Rx. Similarly, the blue and green traces show the frames that arrive to ToR 2 and ToR 3, respectively. It is important to mention that, in all screenshots, the two transmitters are set to emit the sequences shown in Figure 18b.

In addition, the dotted red waveform indicates the separation of intra- and inter-pod traffic of pod 1. White arrows and transmitter labels are highlighted on the following figures, indicating the origin of each packet. More specifically, in Figure 18c,d the frames transmitted by ToR 2 (intra-pod inside pod 2) are shown in the green waveform. Additionally, the orange trace and the black bullets highlighted in Figure 18 represent the WSS on/off state for  $\lambda_3$  and the frames transmitted by ToR 1, respectively.

#### 4.2. Inter-Datacenter Communication—Pisa and Athens Testbed Communication

In order to verify the capability of the NEPHELE framework to control heterogeneous technologies, we validated it in a multi-site inter-DC testbed deployed at IRT Testbed (Interoute) and NTUA (National Technical University of Athens) facilities, in Pisa Italy and Athens Greece, respectively. The environment includes two DCNs emulated by Mininet, connected through an inter-DC network (also emulated) to a third physical DCN, based on the NEPHELE data plane. The emulated networks are running in virtual machines (VMs) placed in the IRT testbed, while the physical NEPHELE DCN is located at the NTUA premises (see Figure 19a). The two sites are interconnected through a VPN tunnel instantiated between two hosts acting as gateways, with the VPN playing the role of an inter-domain link.



**Figure 19.** (a) Multi-site testbed for integrated intra-DC and inter-DC tests. (b) The bandwidth transported over the public internet was measured at 4.91 Mbit/s. (c) The industrial demonstrator of NEPHELE complementing the Athens Testbed demonstrator including the presented technologies.

This environment combines physical and emulated network domains and allows us to verify the correct cooperation between control and data plane for both intra-DC and inter-DC scenarios. Moreover, we are also able to demonstrate the applicability of the NEPHELE system to scenarios involving multiple network domains deploying different technologies, a key feature for the backward compatibility of the NEPHELE solution and its possible deployment in existing multi-DC infrastructures.

The experiment is triggered through the NIDO Graphical User Interface (GUI) sending a request for an end-to-end path between an emulated server located in the IRT testbed and a physical server in the NTUA testbed (3rd supplementary video). At the control plane level, the interaction between NIDO and the SDN controllers OCEANIA and JULIUS, as well as the exchange of OpenFlow messages between the controllers and the data plane, are executed and the end-to-end connection is successfully established.

In order to verify the actual connectivity between the two servers, we use the *iPerf* tool to generate traffic between the servers and we verify the exchanged traffic using the *tcpdump* tool. Although the bandwidth reported (4.91 Mbit/s, Figure 19b) is very low for NEPHELE standards, the traffic is transported through the VPN over the public internet; thus, the actual transfer rate is not a significant indication of the performance of the physical DCN infrastructure.

Finally, the industrial demonstrator of NEPHELE complementing the demonstrator in Athens and the NEPHELE technologies is depicted in Figure 19c.

## 5. Discussion

The NEPHELE DC network [26] is a dynamic optical infrastructure that leverages optical switching and SDN control and orchestration. For the proof-of-concept demonstrator presented in this paper, we implemented a variety of novel functionalities and interfaces across the Open Systems Interconnection (OSI) networking layers. In the SDN controller (and the SDN agents), we extended prominent SDN platforms with TDMA functionality, adding the capability to dynamically assign network resources directly at the optical layer. In addition, fast resource allocation (scheduling) algorithms were integrated to the SDN platform. On the data plane, the functionalities of commercial Ethernet switches were extended with FPGAs. The FPGAs were programmed to perform several novel functionalities such as (1) communication with the extended SDN platform, (2) buffering and traffic handling for adjusting Ethernet traffic to the TDMA scheme according to the SDN commands, and (3) interfacing with the optical network and the control of the optical components (tunable lasers, optical switches, etc.). On the physical layer, our work focused on the optimization of the link quality while introducing novel optical components in an architecture that can scale to thousands of end-nodes. The most challenging part, however, was the integration and the interfacing of the different technologies and innovations. The integration process revealed challenges that will also be relevant for the wider adoption of optical switching. Several of these challenges were addressed during the project, leading to the successful real-time system operation. From our point of view, further collaboration across the broader community covering different networking layers is needed to make optical switching a commercial technology. Optical switching is a paradigm shift and, to exploit its full potential, we will need to make radical changes to the networking environment. An important part of the network, which was not studied in NEPHELE, is the application layer. Making the applications and the developers aware of the slotted operation and its implications will be essential for creating efficient end-to-end optically switched networks.

## 6. Conclusions

We presented several real-time communication scenarios carried out on the NEPHELE optical network demonstrator. End-to-end communication was successfully achieved between the hosts of the prototype datacenter cluster. SDN and orchestration frameworks supervised the slotted operation of the optical network with remarkable synchronization, facilitated by the FPGA boards. NEPHELE demos proved with emphasis the project's concept and put the NEPHELE architecture in a prominent position as an ambitious solution for future DCNs.

**Supplementary Materials:** Videos of the NEPHELE DCN live demonstrator are available online at <https://youtu.be/bJYhRsNwPMU>, <https://youtu.be/J9GbZEtPFC>, and <https://youtu.be/n9mC5NH4IGA>.

**Author Contributions:** Conceptualization, K.T., G.P., C.S., P.B., and K.C.; methodology, K.T., G.P., C.S., P.B., and E.Z.; software, G.P., A.K., G.L., D.G., and M.A.; validation, K.T., G.P., C.S., L.G., D.G., and R.P.; writing—review and editing, K.T., G.P., C.S., E.Z., K.C., and P.B.; visualization, K.T.; supervision, D.R., E.V., and H.A.; project administration, H.A.; funding acquisition, H.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by EU's Horizon 2020 research and innovation program under grant agreement No645212 (NEPHELE).

**Acknowledgments:** We gratefully acknowledge the work of Mr. Marco Capitani in the demo execution. The IRT Pisa testbed was hosted at gtt's facilities (former Interoute) in Italy and we are grateful for their support. The work of Angelos Kyriakos was supported by the Hellenic Foundation for Research and Innovation (HFRI) under the HFRI PhD Fellowship grant (Fellowship Number: 29).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Cisco. Cisco Annual Internet Report (2018–2023), White Paper. Available online: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html> (accessed on 23 June 2020).
2. Singh, A.; Ong, J.; Agarwal, A.; Anderson, G.; Armistead, A.; Bannon, R.; Boving, S.; Desai, G.; Felderman, B.; Germano, P.; et al. Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network. *ACM SIGCOMM Comput. Commun. Rev.* **2015**, *45*, 183–197. [CrossRef]
3. DellEMC. Data Center Networking—Quick Reference Guide. November 2019. Available online: [https://www.dell.com/resources/en-us/asset/quick-reference-guides/products/networking/Dell\\_EM\\_C\\_Networking\\_-\\_QRG\\_-\\_Data\\_Center.pdf?fbclid=IwAR2l6ueMbyuXNgCjGoNJ38hfa\\_5tRIRtUeRrMmiAa8IIC55PzAsLqvp93k](https://www.dell.com/resources/en-us/asset/quick-reference-guides/products/networking/Dell_EM_C_Networking_-_QRG_-_Data_Center.pdf?fbclid=IwAR2l6ueMbyuXNgCjGoNJ38hfa_5tRIRtUeRrMmiAa8IIC55PzAsLqvp93k) (accessed on 23 June 2020).
4. OIF to Present “Cu (see you) Beyond 112 Gbps” Webinar to Debate Requirements for Next Generation Electrical Interconnects, Including Networking Trends and Cloud Scale Applications. Available online: <https://www.businesswire.com/news/home/20200601005611/en/OIF-Present-%E2%80%9CCu-112-Gbps%E2%80%9D-Webinar-Debate> (accessed on 23 June 2020).
5. Mellette, W.M. A Practical Approach to Optical Switching in Data Centers. In Proceedings of the 2019 Optical Fiber Communications Conference and Exhibition (OFC), San Diego, CA, USA, 3 March 2019; pp. 1–3.
6. Zang, D.; Chen, M.; Sun, N.; Proietti, R.; Yoo, S.J.; Cao, Z. OpticV: An energy-efficient datacenter network architecture by MEMS-based all-optical bypassing. In Proceedings of the 2016 IEEE Optical Interconnects Conference (OI), San Diego, CA, USA, 9 March 2016; pp. 70–71.
7. Ezra, Y.B.; Lembrikov, B.I. Application of All-Optical Memory for Advanced Modulation Formats in Communication Intra-Datacenter Networks (Intra-DCNs). In Proceedings of the 20th ICTON, Bucharest, Romania, 1 July 2018; pp. 1–4.
8. Zhong, Z.; Guo, C.; Shen, G. Scheduling Traffic Switching in An All-Optical Intra-Datacenter Network with Sub-Waveband Switching. In Proceedings of the Asia Communications and Photonics Conference (ACP), Hangzhou, China, 26 October 2018; pp. 1–3.
9. Sato, K.I.; Hasegawa, H.; Niwa, T.; Watanabe, T. A Large-Scale Wavelength Routing Optical Switch for Data Center Networks. *IEEE Commun. Mag.* **2013**, *51*, 46–52. [CrossRef]
10. Saridis, G.M.; Peng, S.; Yan, Y.; Aguado, A.; Guo, B.; Arslan, M.; Jackson, C.; Miao, W.; Calabretta, N.; Agraz, F.; et al. Lightness: A Function-Virtualizable Software Defined Data Center Network With All-Optical Circuit/Packet Switching. *J. Lightwave Technol.* **2016**, *34*, 1618–1627. [CrossRef]
11. Xue, X.; Prifti, K.; Wang, F.; Yan, F.; Pan, B.; Guo, X.; Calabretta, N. SDN-Enabled Reconfigurable Optical Data Center Networks Based on Nanoseconds WDM Photonics Integrated Switches. In Proceedings of the 21st International Conference on Transparent Optical Networks (ICTON), Angers, France, 9 July 2019; pp. 1–4.
12. Peng, S.; Guo, B.; Jackson, C.; Nejabati, R.; Agraz, F.; Spadaro, S.; Bernini, G.; Ciulli, N.; Simeonidou, D. Multi-Tenant Software-Defined Hybrid Optical Switched Data Centre. *J. Lightwave Technol.* **2015**, *33*, 3224–3233. [CrossRef]
13. Kachris, C.; Tomkos, I. A survey on optical interconnects for data centers. *IEEE Commun. Surv. Tutor.* **2012**, *14*, 1021–1036. [CrossRef]
14. Wang, G.; Andersen, D.G.; Kaminsky, M.; Papagiannaki, K.; Ng, T.S.E.; Kozuch, M.; Ryan, M. c-Through: Parttime optics in data centers. *ACM SIGCOMM* **2010**, 327–338. [CrossRef]
15. Farrington, N.; Porter, G.; Radhakrishnan, S.; Bazzaz, H.H.; Subramanya, V.; Fainman, Y.; Papen, G.; Vahdat, A. Helios: A hybrid electrical/optical switch architecture for modular data centers. In Proceedings of the ACM SIGCOMM 2010 Conference, New Delhi, India, 30 August–3 September 2010; pp. 339–350. [CrossRef]
16. Singla, A.; Singh, A.; Ramachandran, K.; Xu, L.; Zhang, Y. Proteus: A topology malleable data center network. In Proceedings of the ACM SIGCOMM Workshop on Hot Topics in Networks, Los Angeles, CA, USA, 20 October 2010; p. 8.
17. Porter, G.; Strong, R.; Farrington, N.; Forencich, A.; Pang, C.-S.; Rosing, T.; Fainman, Y.; Papen, G.; Vahdat, A. Integrating microsecond circuit switching into the data center. *ACM SIGCOMM Comput. Commun. Rev.* **2013**, *43*, 447–458. [CrossRef]

18. Benzaoui, N.; Estarán, J.M.; Dutisseuil, E.; Mardoyan, H.; De Valicourt, G.; Dupas, A.; Van, Q.P.; Verchere, D.; Ušćumlić, B.; Gonzalez, M.S.; et al. Cboss: Bringing traffic engineering inside data center networks. *IEEE/OSA J. Opt. Commun. Netw.* **2018**, *10*, 117–125. [[CrossRef](#)]
19. Miao, W.; Yan, F.; Raz, O.; Calabretta, N. OPSquare: Assessment of a novel flat optical data center network architecture under realistic data center traffic. In Proceedings of the Optical Fiber Communication Conference, Anaheim, CA, USA, 20 March 2016; pp. 1–3.
20. Mellette, W.M.; McGuinness, R.; Roy, A.; Forenchich, A.; Papen, G.; Snoeren, A.C.; Porter, G. Rotornet: A scalable, low-complexity, optical datacenter network. In Proceedings of the SIGCOMM '17, New York, NY, USA, 7 August 2017; pp. 267–280. [[CrossRef](#)]
21. Mellette, W.M.; Das, R.; Guo, Y.; McGuinness, R.; Snoeren, A.C.; Porter, G. Expanding across time to deliver bandwidth efficiency and low latency. In Proceedings of the 17th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 20) 2020, Santa Clara, CA, USA, 25–27 February 2020.
22. Kanonakis, K.; Yin, Y.; Ji, P.N.; Wang, T. SDN-controlled routing of elephants and mice over a hybrid optical/electrical DCN testbed. In Proceedings of the 2015 Optical Fiber Communications Conference and Exhibition (OFC) 2015, Los Angeles, CA, USA, 22 March 2015; pp. 1–3. [[CrossRef](#)]
23. Mehmeri, V.D.; Olmos, J.J.; Monroy, I.T.; Spolitis, S.; Bobrovs, V. Architecture and evaluation of software-defined optical switching matrix for hybrid data centers. In Proceedings of the Advances in Wireless and Optical Communications (RTUWO), Riga, Latvia, 3 November 2016; pp. 55–58. [[CrossRef](#)]
24. Wang, C.; Zhang, G.; Chen, H.; Xu, H. An ACO-based elephant and mice flow scheduling system in SDN. In Proceedings of the IEEE 2nd International Conference on Big Data Analysis (ICBDA), Beijing, China, 10 March 2017; pp. 859–863. [[CrossRef](#)]
25. NEPHELE Project Website. Available online: <http://www.nepheleproject.eu/> (accessed on 23 June 2020).
26. Bakopoulos, P.; Christodouloupoulos, K.; Landi, G.; Aziz, M.; Zahavi, E.; Gallico, D.; Pitwon, R.; Tokas, K.; Patronas, I.; Capitani, M.; et al. NEPHELE: An end-to-end scalable and dynamically reconfigurable optical architecture for application-aware SDN cloud data centers. *IEEE Commun. Mag.* **2018**, *56*, 178–188. [[CrossRef](#)]
27. Bakopoulos, P.; Tokas, K.; Spatharakis, C.; Avramopoulos, H. Slotted optical datacenter network with sub-wavelength resource allocation. In Proceedings of the IEEE Photonics Society Summer Topical Meeting Series (SUM), San Juan, Puerto Rico, 10–12 July 2017; pp. 161–162.
28. Tokas, K.; Patronas, I.; Spatharakis, C.; Reisis, D.; Bakopoulos, P.; Avramopoulos, H. Slotted TDMA and optically switched network for disaggregated datacenters. In Proceedings of the 19th International Conference on Transparent Optical Networks (ICTON), Girona, Catalonia, Spain, 2–6 July 2017; pp. 1–5.
29. Landi, G.; Patronas, I.; Kontodimas, K.; Aziz, M.; Christodouloupoulos, K.; Kyriakos, A.; Capitani, M.; Hamedani, A.F.; Reisis, D.; Varvarigos, E.; et al. SDN control framework with dynamic resource assignment for slotted optical datacenter networks. In Proceedings of the Optical Fiber Communication Conference OFC 2017, Los Angeles, CA, USA, 19 March 2017; pp. 1–2.
30. Christodouloupoulos, K.; Kontodimas, K.; Siokis, A.; Yiannopoulos, K.; Varvarigos, E. Efficient bandwidth allocation in the NEPHELE optical/electrical datacenter interconnect. *IEEE/OSA J. Opt. Commun. Netw.* **2017**, *9*, 1145–1160. [[CrossRef](#)]
31. Tokas, K.; Spatharakis, C.; Patronas, I.; Bakopoulos, P.; Landi, G.; Christodouloupoulos, K.; Capitani, M.; Kyriakos, A.; Aziz, M.; Pitwon, R.; et al. Real Time Demonstration of an End-to-End Optical Datacenter Network with Dynamic Bandwidth Allocation. In Proceedings of the 2018 European Conference on Optical Communication (ECOC), Rome, Italy, 23 September 2018; pp. 1–3.
32. Spatharakis, C.; Tokas, K.; Patronas, I.; Bakopoulos, P.; Reisis, D.; Avramopoulos, H. NEPHELE: Vertical Integration and Real-Time Demonstration of an Optical Datacenter Network. In Proceedings of the 20th International Conference on Transparent Optical Networks (ICTON) 2018, Bucharest, Romania, 1 July 2018; pp. 1–4.
33. Bakopoulos, P.; Tokas, K.; Spatharakis, C.; Patronas, I.; Landi, G.; Christodouloupoulos, K.; Capitani, M.; Kyriakos, A.; Aziz, M.; Reisis, D.; et al. Optical datacenter network employing slotted (TDMA) operation for dynamic resource allocation. In Proceedings of the Optical Interconnects XVIII, San Francisco, CA, USA, 22 February 2018. [[CrossRef](#)]
34. NIDO Orchestrator. Available online: <https://github.com/nextworks-it/nephele-nido> (accessed on 23 June 2020).

35. Landi, G.; Capitani, M.; Kretsis, A.; Kokkinos, P.; Christodoulopoulos, K.; Varvarigos, E. Joint intra-and inter-datacenter network optimization and orchestration. In Proceedings of the Optical Fiber Communications Conference and Exposition (OFC), San Diego, CA, USA, 11 March 2018; pp. 1–3.
36. OCEANIA SDN Controller. Available online: <https://github.com/nextworks-it/oceania-dcn-controller> (accessed on 23 June 2020).
37. Kretsis, A.; Corazza, L.; Christodoulopoulos, K.; Kokkinos, P.; Varvarigos, E. An emulation environment for SDN enabled flexible IP/optical networks. In Proceedings of the 18th International Conference on Transparent Optical Networks (ICTON), Trento, Italy, 10 July 2016; pp. 1–4.
38. Peng, S.; Simeonidou, D.; Zervas, G.; Nejabati, R.; Yan, Y.; Shu, Y.; Spadaro, S.; Perelló, J.; Agraz, F.; Careglio, D.; et al. A novel SDN enabled hybrid optical packet/circuit switched data centre network: The LIGHTNESS approach. In Proceedings of the 2014 European Conference on Networks and Communications (EuCNC), Bologna, Italy, 23 June 2014; pp. 1–5. [CrossRef]
39. Shu, Y.; Yan, S.; Jackson, C.; Kondepu, K.; Salas, E.H.; Yan, Y.; Nejabati, R.; Simeonidou, D. Programmable OPS/OCS hybrid data centre network. *Opt. Fiber Technol.* **2018**, *44*, 102–144. [CrossRef]
40. Kondepu, K.; Jackson, C.; Ou, Y.; Beldachi, A.; Pagès, A.; Agraz, F.; Moscatelli, F.; Miao, W.; Kamchevska, V.; Calabretta, N.; et al. Fully SDN-Enabled All-Optical Architecture for Data Center Virtualization with Time and Space Multiplexing. *J. Opt. Commun. Netw.* **2018**, *10*, B90–B101. [CrossRef]
41. Patronas, I.; Gkatzios, N.; Kitsakis, V.; Reisis, D.; Christodoulopoulos, K.; Varvarigos, E. Scheduler Accelerator for TDMA Data Centers. In Proceedings of the 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), Cambridge, UK, 21 March 2018; pp. 162–169.
42. Photonics Communications Research Laboratory Website. Available online: <https://www.photonics.ntua.gr/> (accessed on 23 June 2020).
43. QLogic 57810 Dual Port 10Gb Network Adapter, Product Specifications. Available online: <https://www.dell.com/en-my/shop/qlogic-57810-dual-port-10gb-direct-attach-sfp-network-adapter-full-height/apd/540-bbgs/networking> (accessed on 23 June 2020).



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).