Software Defined Networking for next generation converged metro-access networks

M. Ruffini, F. Slyne, C. Bluemm, N. Kitsuwan, S. McGettrick.
CONNECT / The Centre for Future Networks and Communications, University of Dublin, Trinity College, Ireland.

Abstract

While the concept of Software Defined Networking (SDN) has seen a rapid deployment within the data centre community, its adoption in telecommunications network has progressed slowly, although the concept has been swiftly adopted by all major telecoms vendors. This paper presents a control plane architecture for SDN-driven converged Metro-Access networks, developed through the DISCUS European FP7 project. The SDN-based controller architecture was developed in a testbed implementation targeting two main scenarios: fast feeder fibre protection over dual-homed Passive Optical Networks (PONs) and dynamic service provisioning over a multi-wavelength PON. Implementation details and results of the experiment carried out over the second scenario are reported in the paper, showing the potential of SDN in providing assured on-demand services to end-users.

Introduction

The concept of Software Defined Networking has, after only a few years from its first appearance [1], brought considerable changes in the technical and economical outlook of computer networking and telecommunications networks. The concept was swiftly adopted in data centres, whose private network operations constituted the ideal incubator for such technology, with many new startup companies coming to life in the past five years. While the acceptance of SDN by operators and the wider telecommunications industry should not be taken for granted, SDN remains very popular among the networking research community and almost every switch and router vendor has today implemented some sort of SDN interface, with many also providing an OpenFlow (OF) interface. Indeed some Wide Area Network SDN implementations already exist, most notably the two intercontinental backbone networks interconnecting Google's data centres [2]. While it could be argued that Google's backbone, although intercontinental, is still a private network, without the complexity of an operator public Internet network, it proves the suitability of the SDN framework to operate as network controller for one of the largest revenue generating Service Providers (SPs) in the world, handling over 100 PB of daily data [3].

While the SDN concept has not yet received full attention in access networks, a number of publications have shown the significant advantage they could bring. Recent work has targeted for example, problems such as video distribution optimization [4], and the application of flexi-grid transmission in access and aggregation networks [5]. In addition, operators have recently started to recognise SDN's potential as an enabler of new added-value services in the access. Verizon [6] for example has drawn up a number of use cases where SDN

could help generate new revenue: from increasing quality of service (QoS) and development of new added services to broadband users, to dynamic provisioning of leased line services and protection against failure for enterprise users.
Following from this trend, in this paper we argue that SDN could indeed revolutionize access networks, by enabling much faster provisioning of services and capacity management. As SDN blurs the conventional line between functions carried out by the control plane and those carried out by the management plane, we believe that much benefit can be gained by moving some management functions to a dynamic SDN-based control plane.

In addition, we believe that SDN will help advance network convergence in two complementary domains: the service domain, allowing multiple and diverse service type to converge into the same physical access infrastructure; and the ownership domain, creating a multi-tenancy environment that allows different network operators and service provider to operate over the same network, with different degrees of access control, thus enabling sharing of capital and operational expenditures.
Service convergence has already been widely discussed [7], and typically focused on fixed-mobile convergence. While this is not overly challenging for backhauling current mobile networks (e.g., 3G and Lomg Term Evolution - LTE), challenges start to appear when considering specific features of LTE-Advanced, with one-way delay requirements between eNodeB and Gateway of around 1 ms (e.g, implementing techniques such as Inter-Cell Interference Coordination (ICIC) and Coordinated MultiPoint (CoMP)) [8]. The challenge becomes even more problematic when considering front-hauling which requires one-way delay below 250 μs [9]: the ultra tight latency constraint and the ultra large capacity demands often requires substantial alterations to access network architectures, although some reliefs, at least on the capacity requirements, has recently appeared with the proposal of split processing of the physical layer [10].

From a network ownership perspective, sharing network infrastructure among multiple operators and service providers enables cost sharing, as the infrastructure owner can charge multiple operators for its usage. Studies [11] have shown that the cost per home connected through Fiber-to-the-home (FTTH) can decrease by 65% when three operators share the PON infrastructure. In addition, the ability to share network resources on demand and dynamically is important to increase the efficiency of resource usage, and the overall number of services delivered through the same physical infrastructure, leading to increased overall revenue. This activity has recently been taken up in a study group by the Broadband Forum (BBF), which is currently considering options for access sharing and virtualization. More details can be found in [12].

In summary, we believe that implementing SDN in the access will bring advantage to any operator, as the improved programmatic control and the adoption of open interfaces will foster the creation of new added value services, as well as reduction in complexity of network operation. We also argue that developing open-access interfaces will allow further exploitation of the ultra high capacity of the fibre access network, creating the possibility for more operators and providers to develop their own services on top of the shared infrastructure. Indeed open-access networking is a concept that is perfectly in line with the SDN philosophy, as it provides a natural framework for the development of open interfaces.

In the remainder of the paper we first describe related work and the research roadmap that has conducted to development of SDN network control planes. We then describe a possible SDN access network architecture, developed through the FP7 DISCUS project [13],[14]. We then present testbed results of dynamic service provisioning on converged metro/access nodes using next generation PONs. Finally we conclude the paper.

State of the art

While the term SDN was only coined following the development of OpenFlow in 2008, some of the concepts it promotes had been around for much longer within the research community. The idea of separating control and data planes had been already proposed in the literature, for example with the Forwarding and Control Element Separation (ForCES) concept [15], the Routing Control Platform (RCP) [16] and the 4D architecture [17] (for a comprehensive overview of roadmap to SDN the reader can refer to [18]).
The reason behind the success of OpenFlow over the previous incarnations was the idea of designing the architecture around a set of standardized open interfaces [19] operating via OpenFlow instructions that are compatible with existing hardware switching chips. Thus, a company could launch an OpenFlow-enabled switch in the market by writing new firmware to an existing product, i.e., without requiring development of new hardware.
In addition, SDN brings architectural novelty by developing open interfaces that have fostered network programmability, as the control plane is now hosted on a commodity PC, is typically open source and is developed using well-known programming languages. Instead of relying on specific vendor interfaces, often too restrictive to allow operators to develop significant modifications without assistance form the vendor, OpenFlow makes control plane programmability highly accessible. Such accessibility was further endorsed by the development of a number of now well-established OpenFlow controllers, such as ONOS, Beacon, Floodlight and Opendaylight, developed in Java; Ryu and POX, developed in python; NOX, in C++, Trema, developed in ruby, and many more. In addition the development of OpenFlow-enabled software switches has given the opportunity to any researcher around the world with a commodity PC to develop and test applications for controllers even for medium complexity networks using virtual machines and network emulators (e.g., Mininet), while the same code can then be reused to operate on hardware switches.

In addition, we should remark that the idea of centralizing the network intelligence has already been widely deployed by operators. Indeed, in order to gain control of their network and carry out operations such as traffic engineering or offering bandwidth transport services, operators make ample use of Multi Protocol Label Switching / Generalised Multiprotocol Label Switching (MPLS/GMPLS) tunnels. While MPLS/GMPLS uses dissemination protocols that can be distributed, the routing decision is often made centrally by a network management layer.

Besides currently being well accepted in data centre environments, in the past couple of years the research community has started considering SDN as a viable option for metro [20], and core telecommunications networks [21],[22]. Most popular solutions have seen SDN taking up the role of network orchestrator [23],[24], coordinating existing protocols at different layers of the network. For example recent scenarios have seen the integration of an extended OpenFlow controller for packet switched data with a GMPLS controller for optical switching and transmission [25]. In addition, the logically centralized structure of SDN has been likened to the operating system of a personal computer, fostering the idea of Network Operating System (NOS) [26],[27].

Although the concept of SDN and network virtualization are independent, they are in practice highly correlated, as the programmable SDN framework well suits the dynamic creation and control of virtual slices of a given network architecture. The European Telecommunications Standards Institute (ETSI) for example, has defined a number of use cases for Network Function Virtualisation (NFV) services [28]. These use cases relate to the provision of virtual Customer Premises Equpment (vCPE), Fixed Access Network Function Virtualisation, virtual Provider Edge (vPE) and virtual Basestation (vBS). In addition they have developed a framework for the management and orchestration of all resources in the NFV environment, dubbed MANO [29], covering computing, networking, storage, and virtual machine (VM) resources.

Architecture

The architecture we have chosen for our SDN control plane implementation is shown in Figure 1, and it is based on a hierarchical structure of controllers. The notion of hierarchical controller architecture has recently become established in the research community: besides having made its appearance in the Open Networking Foundation (ONF) SDN architecture document [30], the concept of network orchestration appears frequently in literature [31],[32],[33].

The main benefit of the hierarchical architecture with respect to a federated architecture (see [34] for an example of a virtualization-capable federated architecture) is that it is in line to the typical centralized structure of SDN, as the network orchestrator can act as the central reference controller. However, where multiple network domains are involved, a complete hierarchical approach might not be feasible and a hybrid hierarchical/federated solution might be required.

It should be noted that this architectural model was developed specifically for the DISCUS architecture [13], summarized in Figure 2. In this view we developed the concept of a Metro-Core (MC) node, which connects on one side directly to end users, through a Long-Reach PON access that bypasses the metro transmission network; and on the other side to a transparent core network, or optical island, linking all MC nodes to each other. Although this enables a simplified view of the control architecture, we can see that a hierarchical control plane structures similar to that shown in Figure 1 has been employed for a number of different network scenarios [35].

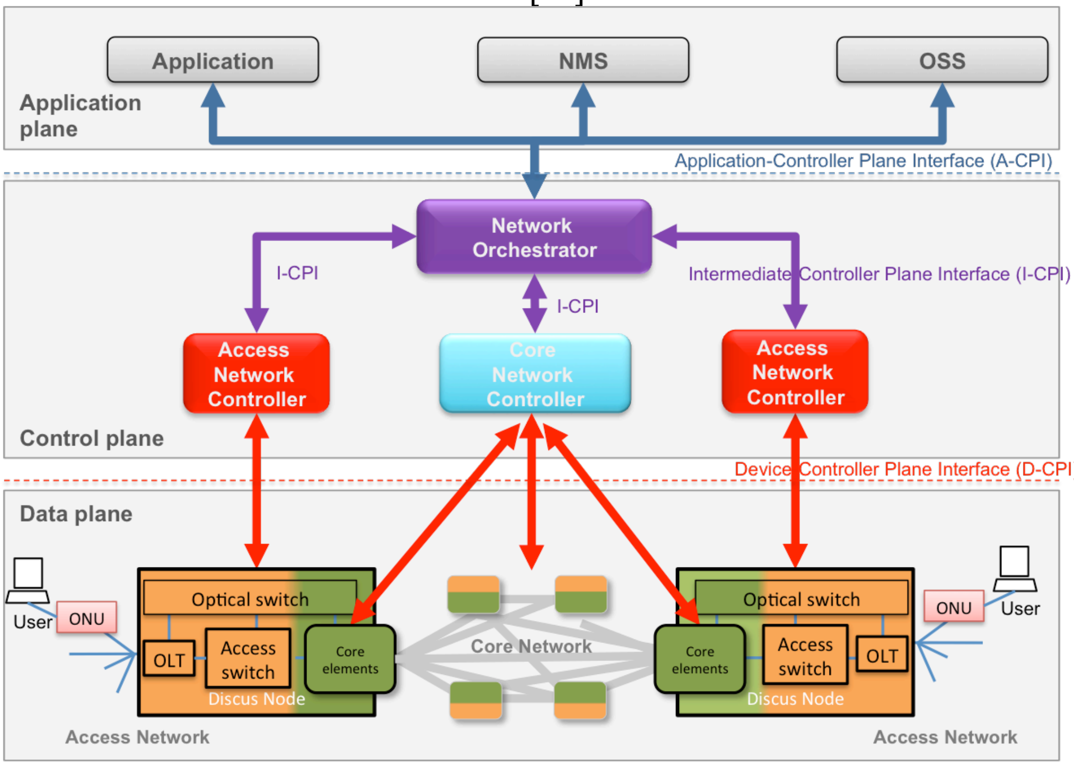

Figure 1 Overall hierarchical control pane architecture

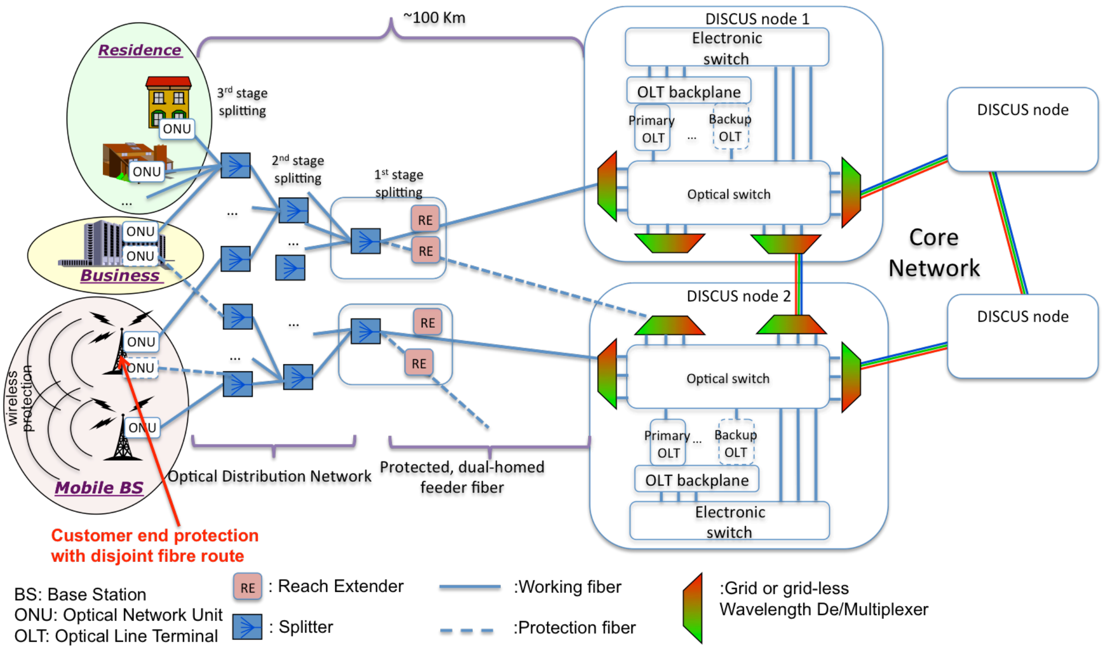

Figure 2 DISCUS architecture, based on long-reach passive optical network access and a flat optical core.

Looking at Figure 1, we can identify three main logical components for the network control plane: the access network controller, in charge of controlling the access network elements; the core network controller, in charge of controlling the elements carrying out core transmission; the network orchestrator, in charge of taking requests from the Service Provider (SP) and translating them into high-

level commands for the access and core network controllers. In addition, it should be noted that the orchestrator is also the main interface to the Network Management System (NMS) and the Operation Support System (OSS), all with different access rights to the Orchestrator APIs.
Following this hierarchical architecture, any service request is sent by an SP to the orchestrator, which calculates a high-level path in the network and then converts the service request iteratively into sub requests towards the source and destination access network controllers and the core network controller.

While Figure 1 only shows a high level view of the Metro-Core node, a more detailed view is reported in Figure 3. In orange we show the access elements controlled by the access network controller, while in green the elements controlled by the core controller. In light brown are instead the IP/MPLS service nodes that might be controlled by individual Service Providers. These might not be physically located within the MC node housing, and are connected to through the optical switch. Their interface is then terminated to the access switch that will appropriately break the incoming flows into those directed towards the access served by this MC node and those directed towards other MC nodes, which are forwarded to the MPLS-TP (Transport Profile) core switch. The optical switch is the element that provides flexibility to the architecture, constituting the core element of the node that enables flexible any-to-any connectivity of the components[1]. Due to the large port number, which can be of the order of ten thousand for the largest nodes, the switch is developed as a 2-stage architecture built using single-sided switching components.
The switch is virtualized into two sides, one connecting access equipment and the other the core equipment, each managed by the respective controller.

[1] While any-to-any connectivity provides the highest flexibility it is envisaged that such high degree of freedom might not be required and constraints could be included in order to reduce the overall number of switching elements required to operate the multi-stage architecture.

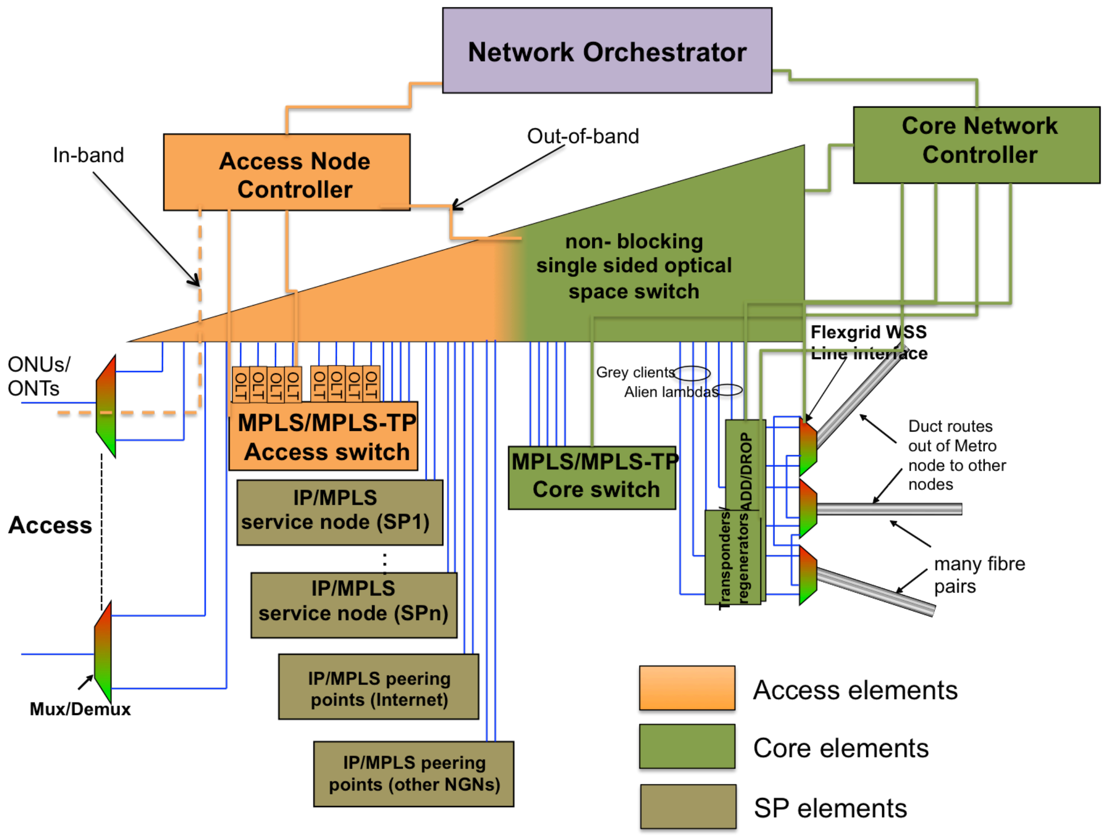

Figure 3 Detailed view of the Metro-Core node architecture

From a transport network switching and routing perspective, the data plane operates on two sets of stacked MPLS labels. The first set, the PseudoWire (PW) identifies an end-to-end path, and is assigned to a specific service type (e.g., video on demand, Internet, VoIP, Bandwidth on Demand) and it is unique for a given SP and destination OLT. The SP then aggregates multiple PWs, for transport purpose within the core network, into a Label Switched Path (LSP), which is used to route the packet through the network core. An SP can have multiple LSPs, at least one per source-destination MC node pair, but more are possible in order to operate service level distinction. An example of service implementation through LSP and PW is shown in Figure 4. For a given service flow, the SP installs a PW label into the packets header. It then aggregates this flow together with other flows going towards the same destination using a common LSP label. Packets are then forwarded to the MC node where the LSP label is examined to determine whether the aggregate flow is to be processed at this node or forwarded through the core towards a different MC node. Once forwarded to the correct MC node, the LSP label is stripped off by the access switch, which uses the PW label to determine QoS behavior of individual PW flows and reach the correct OLT. The OLT then strips off the PW label forwarding the packet to the ONU.

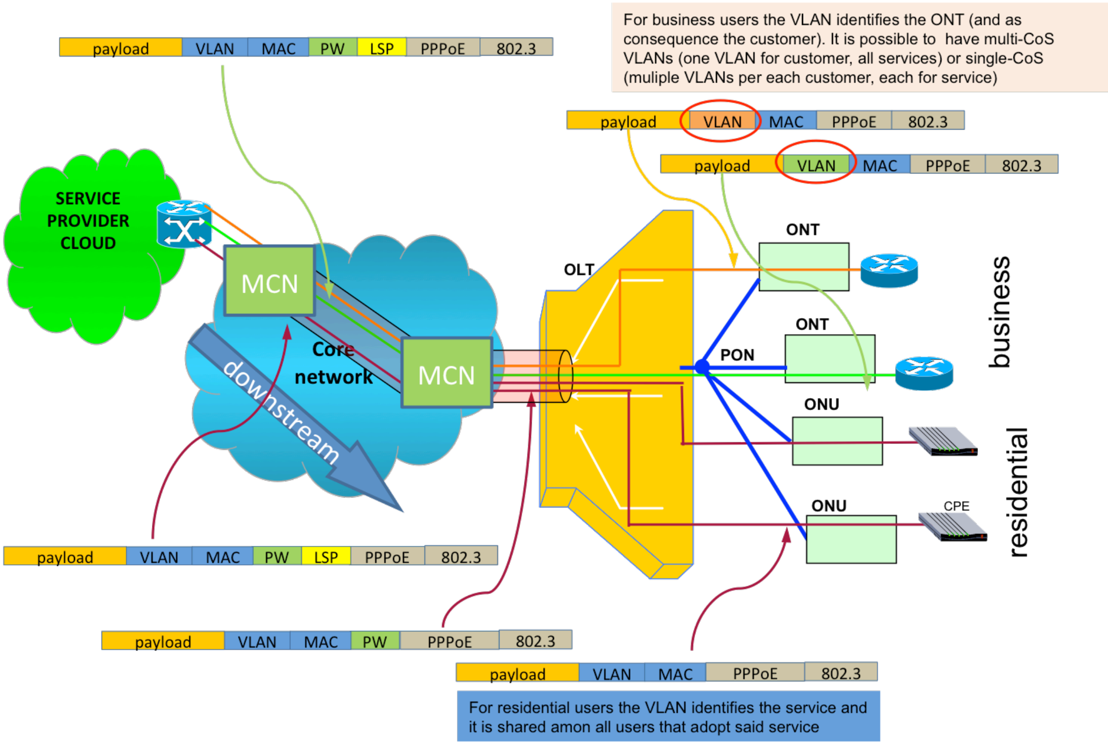

Figure 4 Examples of service implementation through LSP and PW MPLS links

Access network controller

This section provides an overview of the access network controller architecture and summarises the main tasks carried out by the access controller. It should be noted that our work so far has covered the use of SDN-based controller for dynamic service provisioning, while the development of an SDN framework for open access and virtualisation-enabled access network architecture will be targeted in future work[2].

The differentiation between the network orchestrator and the access and core controllers is only logical, meaning that from a physical point of view all software components might be co-located (i.e., in the same data centre, servers, machine, etc.) or reside somewhere in the cloud. An access network controller is associated to every MC node, where it controls the optical switch, access switch, and OLTs/ONUs.

Figure 5 shows a schematic of the node controller architecture and its interaction with the network orchestrator. A JavaScript Object Notation (JSON) interface forwarding RESTful (i.e., using Representational State Transfer) API (Application Program Interface) is used to communicate with the orchestrator, which, where required, calculates end-to-end paths and then requests them to the destination node controllers.

[2] This work will be targeted through the Science Foundation Ireland funded project "O'SHARE: An open-access SDN-driven architecture enabling multi-operator and multi-service convergence in shared optical access networks" (project ID: 14/IA/2527).

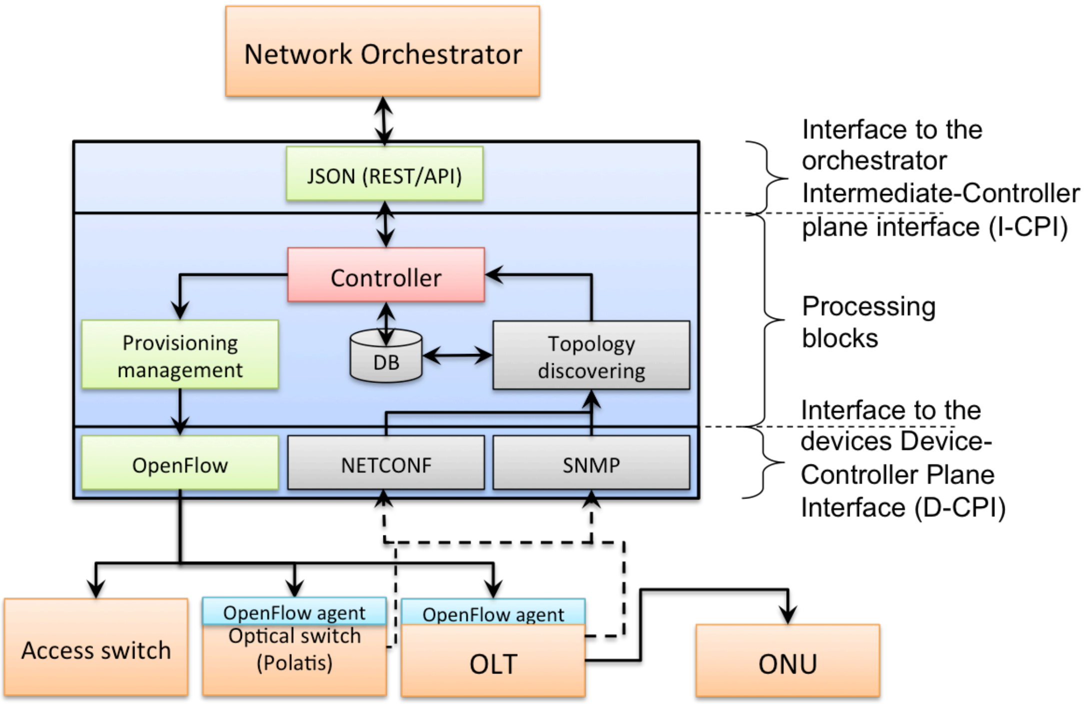

Figure 5 Access network controller architecture

After receiving a request from the orchestrator, the JSON interface passes the information to the main controller processing block, which checks the database for available capacity and ports. Topology information, together with bandwidth reservation information is stored in the database. If the request can be satisfied, the controller sends a confirmation to the SP/orchestrator and instructs the provisioning manager to carry out the appropriate actions to create the connection on the access side. The provisioning management block translates the incoming actions into messages for the appropriate protocol that can be handled by the physical devices. Our architecture uses OpenFlow as Device Controller Plane Interface (D-CPI, also known as southbound interface). The access switch uses the embedded OpenFlow interface, while for the optical switch and OLT we have developed an OpenFlow agent based on the OF v1.4 standard.

The grey blocks include protocols and modules used to maintain and manage a map of the network topology, and includes both NETCONF and SNMP for backward compatibility.

The main tasks carried out by the access network controller can be summarised as follows:

- **Maintain communication with the network orchestrator.** The access controller receives provisioning requests from the orchestrator and reports the service setup status.
- **Export abstracted network view to the controllers.** The access controller sends abstracted topological information about the resources available within its domain. To do so, the physical domain details have to be mapped in the abstracted information required by the orchestrator.

- **Operate QoS of individual or aggregate flows.** The access controller operates connection Admission Control (CAC) mechanisms and capacity reservation.

- **Handle network capacity management within the access-aggregation network.** The access controller maintains information on bandwidth availability within the access switch and each OLT. Each new connection request is assessed against the booked Committed Information Rate (CIR) at each output port of the switch, and at the OLTs (although managing OLT downstream capacity is not required where the switch port rate matches the downstream OLT rate). It also maintains connectivity information on the optical switch and on the number of OLTs and point-to-point transceivers available.

- **Translate the abstract link parameters into appropriate commands for the D-CPI (southbound) interface.**

  A summary of the main parameters include:

  - For the access switch: setting up, modifying and deleting flow entries, setup meter tables for QoS (priority tagging), MPLS and VLAN flow tagging, forwarding actions.
  - For the OLT: GPON Encapsulation Method (GEM) ports and VLAN/PW mapping/translation, DBA algorithm configuration (e.g., fixed, assured and best effort rates), downstream wavelength selection[3].
  - For the optical switch: connect/disconnect ports, incoming/outgoing port power measurement.
  - For the ONU/ONT: Messages from the OLT to the ONU are generally supported through the ONT Management and Control Interface (OMCI). Our implementations requires exchange of information on Traffic Container (T-CONT)/GEM ports and C-VLAN mapping/translation, downstream/upstream wavelength selection, OLT registration and ONU configuration parameters.

- **Access network protection.** Where access protection is required, the controller handles incoming failure messages from the OLT to operate fast protection. While our implementation targets specifically feeder fibre protection and thus is based on Loss Of Signal (LOS) alarms from the OLT, more sophisticated failure detection systems can be implemented in the system.

Controller implementation and testing

The access network controller architecture described above was implemented as a control mechanism for the DISCUS Metro-Core node. While a complete control plane implementation was out of the scope of the project, we have implemented a number of functionalities to address two main scenarios:

[3] It is assumed the receiver filter is operated by the wavelength demultiplexer.

1) The setup of a service with guaranteed capacity (e.g., for Video On Demand or more general Bandwidth on Demand services).
2) A fast feeder fibre protection mechanism with N:1 sharing of backup OLTs.

Our access SDN controller was implemented on the RYU OpenFlow platform, which provides software components with well-defined APIs to enable the development of new network management and control applications. It supports several protocols for managing network devices, such as OpenFlow (versions 1.0, 1.2, 1.3, and 1.4), NetConf, and OF-config.

Figure 6 shows the interaction of the control plane components implementing the first scenario. While our implementation has focused on the access network controller, we have implemented a lightweight network orchestrator, to handle communication to/from the SP (shown in the figure as a Web Portal) and to/from the access controller. We have also implemented a simple OpenFlow-based core controller to enable testing of scenarios through Openflow core networks (for example scenario 2 was tested through the core Europe-wide GEANT OpenFlow network).

For the first scenario we assume that a portal exists where the user can submit (step 1) a service request indicating the service type (committed and peak capacity, wavelength range and channel allocation type – e.g., shared vs. dedicated), and destination.

The request to the portal is passed to the SP, which forwards the request to the network orchestrator (step 2). The orchestrator finds the access node controller where the user is located and forwards the request (step 3). The access controller needs to assess how the capacity can be provided to the user, considering the preferences and abilities expressed by the user. For example the user might require capacity on a shared channel, but the current channel might not have enough capacity to satisfy the request. The user's ONU can thus be moved to a different channel, provided it is able to tune to one of the available channels. If the capacity is available the access controller proceeds to configure the access switch, OLT and ONU to carry out the connection (step 5) and then confirms the service request to the orchestrator (step 6). The orchestrator confirms the request to the SP, which will pass the information to the web portal to give feedback to the user (step 7).

The main scope of this scenario is to show the dynamic provisioning of a new end-to-end path across the node, thus creating a new flow entry (comprised of Pseudowire (PW) and Label-Switched Path (LSP) tags, as required) and also using multi-wavelengths allocation in the PON. This scenario assumes that user traffic is terminated at the MC node, i.e. does not cross transparently the node[4].

Further details on the envisaged API interfaces can be found in [36].

---

[4] Due to the high loss of the PON section, the signal is regenerated and then groomed into appropriate core transmission channels.

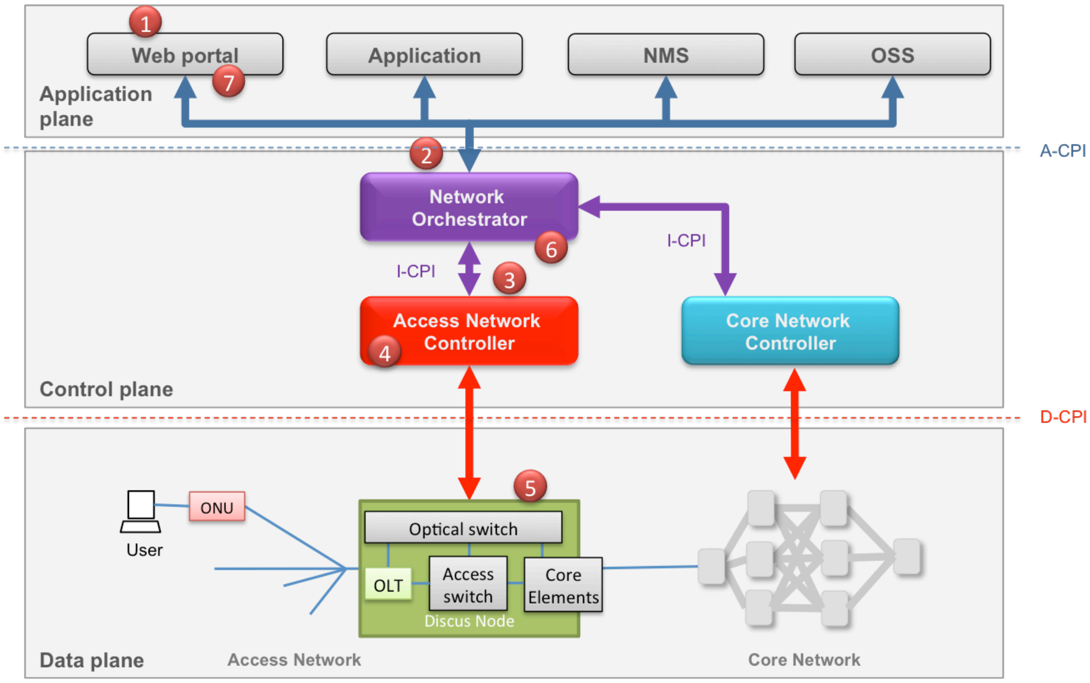

Figure 6 Scenario 1: dynamic provisioning of bandwidth-on-demand services

The second scenario demonstrates fast PON protection against cuts of the feeder fibre. The PON is protected by a backup OLT that is located at a different MC node in order to provide diversity of fibre path and OLT location. N:1 protection is achieved by connecting the feeder fibres and OLTs to the optical switch (as in Figure 3), so that any standby OLT can protect for any feeder fibre failure. The N:1 value determines the ratio of active to standby OLTs, obtained by oversubscribing the number of standby OLTs over the number of feeder fibres to be protected. This concept was theoretically investigated in [37].

While it is debatable whether fast feeder protection is required for an access network mainly servicing residential customers, we argue that the current trend towards service convergence in access networks will pose strict requirements on the overall network availability. For example, the same PON fibre could provide both residential and business services (such as private lines or back-hauling/front-hauling of mobile stations), which might require Service Level Agreements (SLA) with high network availability. In addition, fast feeder protection can enable protection sharing mechanisms thus lowering the overall cost for protection [38],[39].

The logical view of the control plane interaction for scenario 2 is shown in Figure 7. The fibre cut is identified by a LOS alarm[5] from the OLT to the access controller (step 1). Upon failure detection, the control plane operates as follows. The access controller in MC node 1 informs the network orchestrator about the failure (step 2). The orchestrator informs the core controller to setup the pre-calculated core backup paths corresponding to the failure (step 3). The orchestrator then communicates to the access controller in the MC node hosting the standby OLT, passing the information required to activate the OLT as well as

[5] The LOS alarm is activated whether there is a failure of the upstream path or the downstream path. Detection of downstream path failure rests upon the fact that ONUs stop any transmission as soon as it looses contact with the OLT, (e.g., following the fibre cut).

the appropriate protection paths that need to be activated in the node (step 4). In the meantime the core controller will configure the core nodes (step 5) while the access controller of the standby MC node will configure the access elements (step 6). The core controller will then provide feedback to the orchestrator (step 7) on the outcome of the core traffic redirection, and the access controller of the standby node on the access protection (step 8).
It should be noticed that in order to speed up the protection process a number of operations are operated asynchronously and in parallel (e.g., without waiting for confirmation from the access and core controllers).
Finally, this scenario assumes that a network management system has already pre-calculated standby PWs, that are associated to existing or other standby LSPs, during protection configuration. Protection setup schemes for PW redundancy have already been standardized [40].

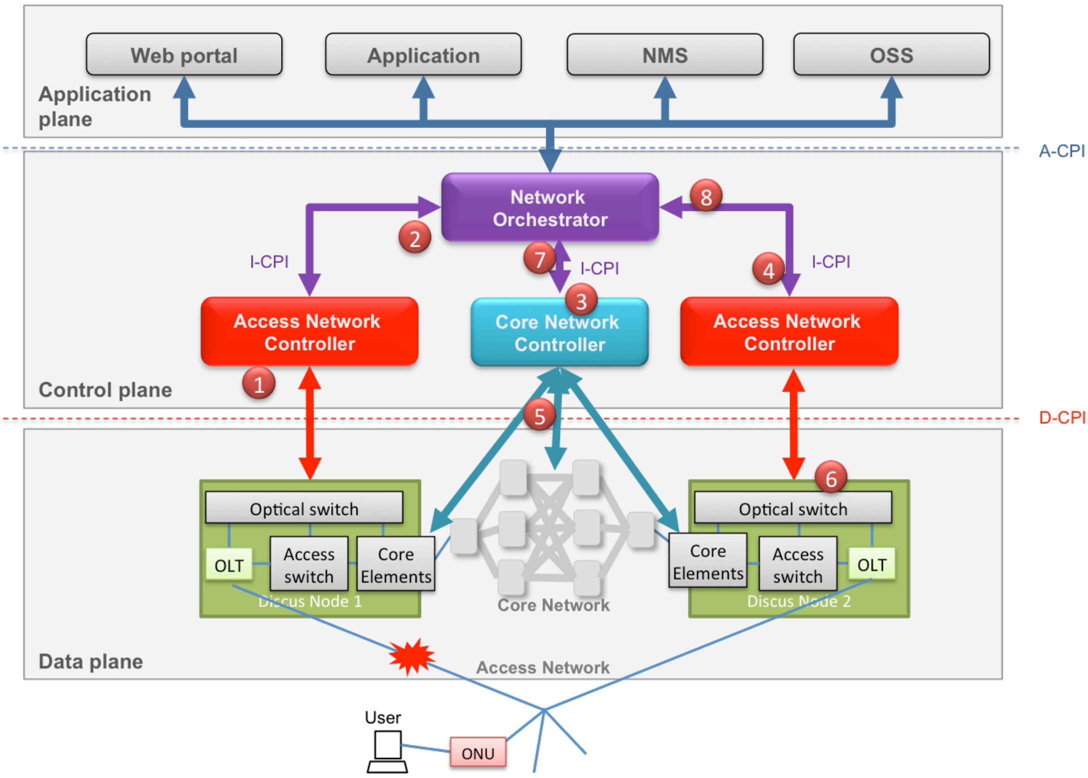

Figure 7 Scenario 2: fast PON feeder fibre protection

Demonstration of dynamic bandwidth provisioning

While the network protection scenario has been demonstrated in 1+1 [41],[42], 1:1 [43] and N:1 OLT sharing [44], here we report a demonstration of the dynamic Bandwidth-on-demand service provisioning with dynamic channel selection.

The testbed experiment we report shows the dynamic provisioning of a service using an SDN controller with OpenFlow as southbound interface. It is assumed that while an ONU is receiving a service with a given reserved capacity (which we refer to as service-A) by an OLT (which we refer to as OLT-1), it also requests capacity for an additional service, for example Bandwidth-on-demand (which we refer to as service-B). The scenario is setup so that the wavelength channel used by the ONU does not have enough capacity to support the requested Bandwidth-on-Demand (BoD) service. The controller needs thus to activate a second OLT (which we refer to as OLT-2), in order to deliver service B. Since the ONU only has one (tunable) transceiver, both services A and B need to be delivered over the same wavelength, by OLT-2. This implies redirecting service A to OLT-2, which requires communication with the SP to move service A into a different PW which can be routed to OLT-2. The aim of the experiment is to demonstrate the ability of OpenFlow and SDN to handle such end-to-end service provisioning and to measure the time required to provision service-B and the impairment generated for rerouting service-A.

It should be noticed that this experiment focuses on the control plane activities rather than on the physical layer. Thus for example the tunable lasers are tunable DWDM Small Form-factor Pluggable transceivers (SFP+) with fast tuning time and precise wavelength locking, while on the other hand we have used slow Wavelength Selective Switches (WSS) as tunable filters. While this might not represent the standard for current low-cost ONU devices, we argue that our results remain valid from a control plane timing perspective and additional delays could be accounted for in estimating overall provisioning time of lower-cost transceivers.

The experiments are run on our Optical Network Architecture access testbed with proprietary ONU and OLTs implemented in Field-Programmable Gate Array (FPGA) hardware [43],[44], a proprietary network orchestrator implemented in python and an access network controller built on top of the RYU SDN platform.

A logical view of the testbed components and operations is shown in Figure 8, showing the different steps required to provision the service[6]. Following the user request for a new BoD service-B, the Orchestrator forwards a JSON message to the OF controller (step 1) indicating the capacity requested and the range of wavelengths the ONU can support. Once the controller has verified the capacity availability it acknowledges the orchestrator, which communicates with SP2 to activate the new service-B and with SP1 to reroute service-A into a different PW (Step 2). The orchestrator then informs the core controller to add a new LSP path for routing service B to the correct MC node (step 3). In parallel the Metro-Access controller communicates with OLT-2 through the OpenFlow agent translating OF commands into proprietary control messages sent through the UART interface, indicating to activate its service on the desired wavelength.

The Metro Access controller prepares the physical and link layer between OLT-2 and the ONU. OLT-2 is instructed to set the frequency of its downstream tunable laser and is given the explicit parameters used to bind the PW to the alloc_id for each of both service-A and service-B (step 4).

The OLT's and ONU's do not present native Openflow interfaces, but instead are controlled over a high-speed serial Universal Ssynchronous Receiver/Transmitter (UART) interface running at 406kbps. The Microblaze (i.e., a simple general purpose processor implemented in the FPGA hardware) on the host FGPA boards presents an interface for directly programming and interrogating PON control registers, which are then accessible over the high-speed UART interface. Run-time control of the PON is executed through the

[6] Although we described the process in sequential steps, some of the actions described are actually carried in parallel in order to speed up the provisioning process.

interfaces on OLT-1 and OLT-2 which in turn relay control instructions to the remote ONU using Physical Layer Operations, Administration and Maintenance (PLOAM) messages. The run-time functionality includes configuration of the laser frequencies of the OLT and ONU tunable lasers, the configuration of alloc_id's at the ONU for appropriate XG-PON encapsulation Method (XGEM) packets and the re-homing of ONU from OLT1 to OLT2.
An Openflow agent wrapper around both OLT-1 and OLT-2 was developed so as to present an Openflow v1.4 compatible interface to the Metro-Access Controller. Openflow v1.4 facilitates the control of optical parameters of OpenFlow compatible switches and devices through the 'OFPPortModPropOptical' method. These parameters include the transmission centre frequency or wavelength, a frequency offset from the centre frequency and the transmission power level (dB) and are a subset of those which we are looking to control within the PON. Because we need runtime control of additional non-standard parameters, we enhanced both the v1.4 protocol and the agent to allow configuration of the XGEM, Alloc_ID and PseudoWire tags associated to a given flow through the Metro-Access Controller.
The controller then installs the new entry for the PW, including meter tables for QoS and capacity reservation in the switch flow table (step 5).
In parallel, The Metro Access controller, through a PLOAM message from OLT-1, instructs the ONU to tune the filter to the new wavelength (step 6). A serial link between a secondary UART on the ONU and the tunable filter, tunes the filter to the new wavelength thereby directing the signal to OLT2. (step 7).

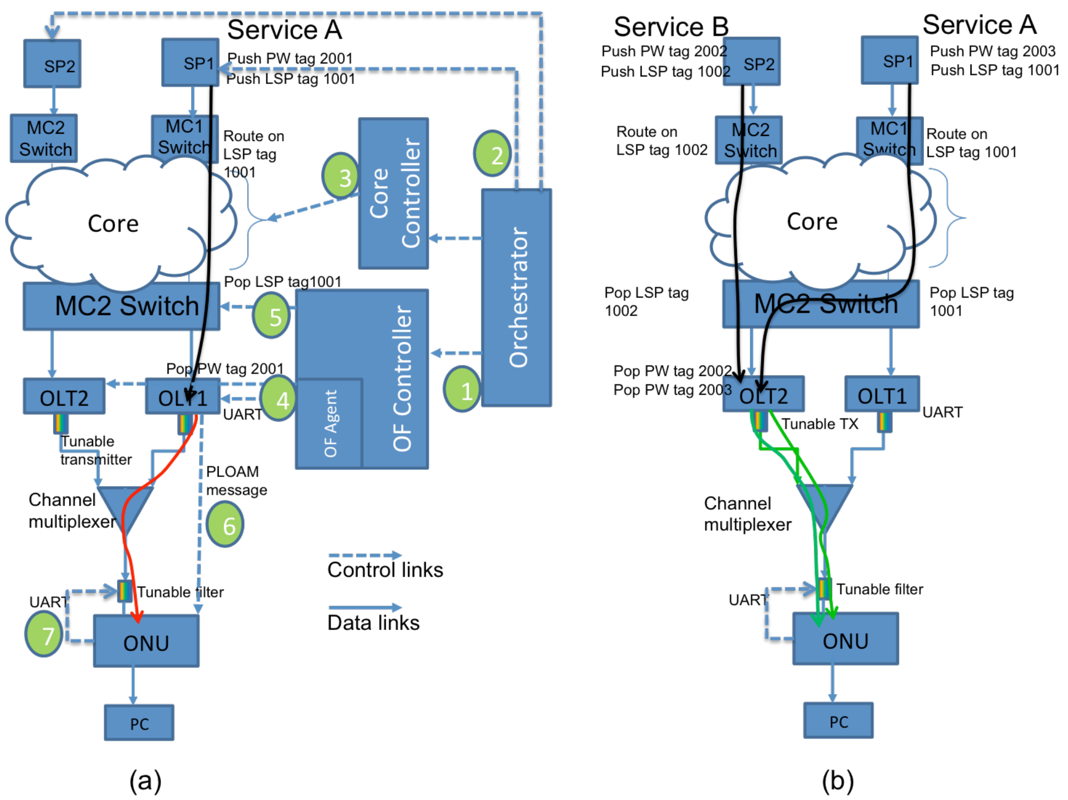

Figure 8 (a) Operation of SDN control plane to provision service B while operating service A; (b) network state after provisioning of service B and re-direction of service A to OLT2.

The timing results of the testbed experiment are reported in Figure 9, averaged over ten testbed runs. From the time the service is requested to the orchestrator (step 0), the orchestrator and access controller introduce a 444 ms processing delay (steps 1 to 3 as also reported in Figure 8), after which the OLT receives the command to activate a new channel (step 4) and the access switch, in parallel, receives the OpenFlow commands for updating the PW tags in the flow table (step 5). The PLOAM message containing the information on the new wavelength channel is then received and acknowledged by the ONU within 1.5 ms (step 6), triggering it to activate the tunable filter and laser to the correct wavelength channel and causing the OLT-1 to de-register the ONU. In our testbed the tunable filter was operated by a WSS device, which introduced a sizable latency of about 2 seconds. Once the filter is tuned to the new channel (step 7), the ONU is re-registered on the new OLT within 7 ms (step 8).

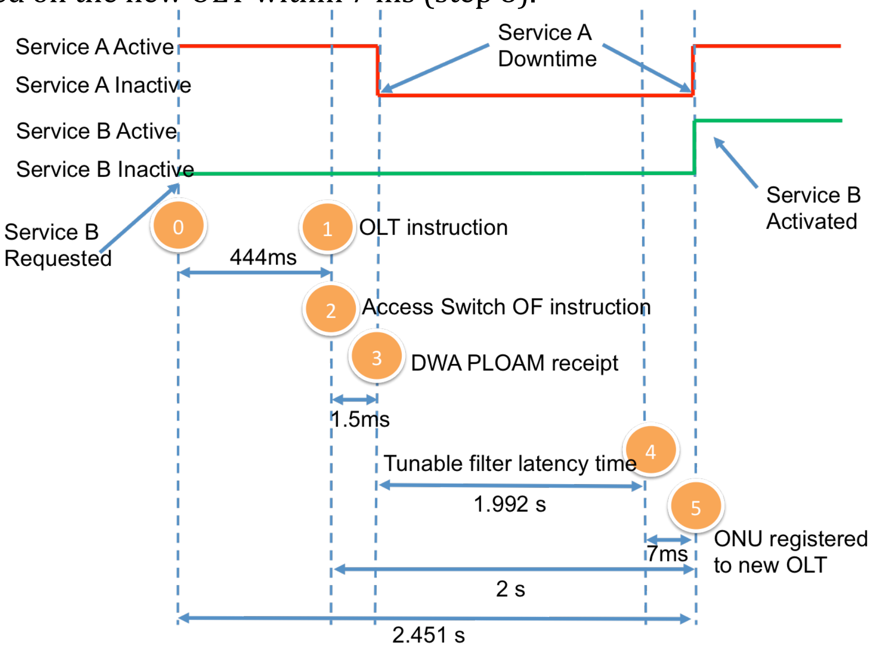

Figure 9 Timing diagram of the service activation experiment

The results show that about 2.45 seconds are required in order to move the ONU to a new channel, with a disruption time on existing services of about 2 seconds. It should be noticed however that most of the latency was due to the slow switching time of the WSS. This could be reduced to about 50 ms by using commercially available tunable filters, which would reduce the downtime on existing services to about 60 ms, while activating the new service in about 500 ms.

In addition we believe that better optimization of the orchestrator and controller could be operated to further reduce the activation time of the new service.

Conclusions

This paper has presented a hierarchical control plane architecture for next generation broadband networks, focusing on the details of an OpenFlow-based controller for a converged Metro-Access network. Besides giving a description of the main component and interfaces, the paper has provided implementation details based on two main scenarios, fast protection of a dual-homed PON system and dynamic service provisioning over a multi-wavelength PON.

A testbed experiment was carried out for the second scenario, emulating a user request through Service Provider portal, operating through a network orchestrator and an OpenFlow-based Metro-Access SDN controller. The latency of our experiment was strongly dominated by the slow tuning time of the WSS we used as tunable filter, showing wavelength service setup times of about 2.5 seconds, while introducing a 2 seconds disruption on existing services. However, we recognize that the use of appropriate tunable filters could reduce service disruption times to about 60 ms and service activation time to less than half a second.

While more work is required to assess the viability of a fully SDN-based controller for a telecommunications metro-access network, we believe the demonstration shown in this paper, together with other recent implementations, are creating an important roadmap for the programmability of the network control and management planes. These will be essential to maintain a healthy balance between the constant increase in link capacity and the need for more dynamic and automated network control and management.

Acknowledgments

The research leading to this article has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n. 318137 (Collaborative project "DISCUS"), and Science Foundation Ireland under grants no. 12/IA/1270 and 10/CE/I1853. The authors would also like to acknowledge Victor Lopez, Andrea Di Giglio, Marco Schiano and Laura Serra for the fruitful discussions on control plane architecture and data labelling models.

References

[1] N. McKeown et al., OpenFlow: Enabling Innovation in Campus Networks. ACM SIGCOMM Computer Communication Review, 38(2), April 2008.

[2] U. Holze, "Openflow @ Google", Open Networking Summit 2012, available at: http://opennetsummit.org/archives/apr12/hoelzle-tue-openflow.pdf

[3] K. Amin, KMS technologies. Available form: http://www.slideshare.net/kmstechnology/big-data-overview-2013-2014

[4] X. Li, et al., Joint Bandwidth Provisioning and Cache Management for Video Distribution in Software-Defined Passive Optical Networks. Paper Tu3F.5, OFC 2014.

[5] N. Cvijetic, et al., SDN and OpenFlow for Dynamic Flex-Grid Optical Access and Aggregation Networks. IEEE Journal of Lightweight Technology, 32(4), 2014.

[6] N. Bitar, SDN and Potential Applicability to Access Networks. Paper Th1G.4, OFC 2014.

[7] T. Pfeiffer, A Physical Layer Perspective on Access Network Sharing. Elsevier OFT special issue on access networks, November 2015.

[8] Stoke white paper, Latency Considerations in LTE, September 2014. Available at http://www.mavenir.com/files/doc_downloads/Stoke_Documents/130-0029-001_LTELatencyConsiderations_Final.pdf

[9] NGMN Alliance, Fronthaul requirements for C-RAN, March 2015. Available at: https://www.ngmn.org/uploads/media/NGMN_RANEV_D1_C-RAN_Fronthaul_Requirements_v1.0.pdf
[10] U. Doetsch, et al., Quantitative Analysis of Split Base Station Processing and Determination of Advantageous Architectures for LTE, Bell Labs Technical Journal 18(1), January 2013.
[11] J. Rendon, Network sharing with FTTH/PON architectures. Huawey Commnunicate, No 68, Jan. 2013. Available at: http://www.huawei.com/en/static/HW-201348.pdf
[12] B. Cornaglia et al., Fixed Access Network Sharing. Elsevier OFT special issue on access networks, November 2015.
[13] M. Ruffini, et al., DISCUS: An end-to-end solution for ubiquitous broadband optical access. IEEE Com. Mag., vol. 52, no. 2, February 2014
[14] M. Ruffini, et al., DISCUS: End-to-end network design for ubiquitous high speed broadband services (Invited). ICTON conference, 2013.
[15] L. Yang, et al. Forwarding and Control Element Separation (ForCES) Framework. Internet Engineering Task Force, RFC 3746, 2004.
[16] N. Feamster, et al.. The case for separating routing from routers. ACM SIGCOMM Workshop on Future Directions in Network Architecture, Sept. 2004
[17] A. Greenberg, et al. A clean slate 4D approach to network control and management. ACM SIGCOMM Computer Communications Review, 35(5), 2005.
[18] N. Femster, et. Al. "The Road to SDN: An intellectual history of programmable networks", ACM SIGCOMM Computer Communication Review, Vol. 44, No. 2, 2014.
[19] Z. Wang et al., IETF Internet draft "Analysis of Comparisons between OpenFlow and ForCES draft-wang-forces-compare-openflow-forces-01", March 2012
[20] S. Hubbard, Software-Defined Metro Networks: Virtualizing the Network & Services Edge. Light Reading White Paper, December 2012, available at: http://www.overturenetworks.com/sites/default/files/white_paper/Heavy Reading_WP_Metro SDN_EdgeVirtualization_12-18-12.pdf
[21] P. Iovanna, et al., E2E Traffic Engineering Routing for Transport SDN. Paper W1K.3, OFC 2014.
[22] R. Casellas, et al., SDN based Provisioning Orchestration of OpenFlow/GMPLS Flexi-grid Networks with a Stateful Hierarchical PCE. Paper Th3I.2, OFC 2014.
[23] A. Aguado, et al., ABNO: a feasible SDN approach for multi-vendor IP and optical networks. Paper Th3I.5, OFC 2014.
[24] A. Autenrieth, et al., Introducing OFELIA and OpenFlow extensions to support Optical Networking Equipment. ONTC Meeting, OFC 2012. Presentation available at: http://www.fp7-ofelia.eu/assets/Publications-and-Presentations/2012-03-08-ONTC- Autenrieth-OFELIA-final.pdf
[25] S. Azodolmolky et al., Integrated OpenFlow–GMPLS control plane: an overlay model for software defined packet over optical networks. Optics express, 19(26), 2011
[26] N. Gude, et al. NOX: Towards an Operating System for Networks. ACM SIGCOMM Computer Communication Review, 38(3), 2008.

[27] V. López et al., Towards a Network Operating System. Paper Th3I.6, OFC 2014.
[28] ETSI standard GS NFV 001, Network Functions Virtualisation (NFV): Use Cases. October 2013.
[29] ETSI standard GS NFV-MAN 001, Network Functions Virtualisation (NFV): Management and Orchestration. December 2014.
[30] ONF. SDN Architecture, Issue 1, June 2014. www.opennetworking.org
[31] F. Francois, et al., Experimental Assessment of ABNO-based Network Orchestration of end-to-end Multi-layer (OPS/OCS) Provisioning across SDN/OpenFlow and GMPLS/PCE Control Domains. Paper We.3.2.4, ECOC 2014.
[32] R. Casellas, et. Al., SDN Orchestration of OpenFlow and GMPLS Flexi-grid Networks with a Stateful Hierarchical PCE , Journal of Optical Communications and Networking, Vol. 7, No. 1, January 2015.
[33] R. Munoz, et al., Transport Network Orchestration for end-to-end Multi-layer Provisioning Across heterogeneous SDN/OpenFlow and GMPLS/PCE Control Domains. Journal of Lightwave Technology, Vol.33, No. 8, April 2015.
[34] P. Szegedi, et al., Enabling future internet research: the FEDERICA case. IEEE Communications Magazine, vol. 49, no. 7, July 2011.
[35] R. Vilalta, et al. The Need for a Control Orchestration Protocol in Research Projects on Optical Networking. To appear in proceedings of EuCNC 2015.
[36] M. Ruffini, at al., DISCUS FP7 project Deliverable D6.3: "Report on the design of the interfaces of the control plane", August 2014.
[37] A. Nag, et al., N:1 Protection Design for Minimising OLTs in Resilient Dual-Homed Long-Reach Passive Optical Network. Paper Tu2F.7, OFC 2014
[38] M. Ruffini, et al., Deployment case studies of an energy efficient protected LR-PON architecture. Proceedings of *IEEE Optical Network Design and Modelling*, 2012
[39] M. Ruffini, et sl., Protection Strategies for Long-Reach PON. ECOC 2010.
[40] Muley at al., IETF RFC 6718, Pseudowire Redundancy, 2012.
[41] S. McGettrick, et al., Ultra-fast 1+1 Protection in 10 Gb/s Symmetric Long Reach PON. ECOC 2013.
[42] S. McGettrick, et al., Improving Hardware Protection Switching in 10Gb/s Symmetric Long Reach PONs, *Optical Fibre Copnference (OFC), Anaheim, CA*, 2013
[43] F. Slyne, et al., Design and experimental test of 1:1 End-to-End Protection for LR-PON using an SDN multi-tier Control Plane. Paper Mo.3.2.4, ECOC 2014
[44] S. McGettrick, et al., Experimental End-to-End Demonstration of Shared N:1 Dual Homed Protection in Long Reach PON and SDN-Controlled Core. Paper Tu2E.5, OFC 2015.