Optical Fiber Technology 26 (2015) 31–41

Contents lists available at ScienceDirect

Optical Fiber Technology

www.elsevier.com/locate/yofte

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

ARTICLE INFO

Article history:
Available online 14 September 2015

Keywords:
Access SDN
Next generation Passive Optical Networks
Metro-Access Convergence
OpenFlow
Long-Reach PON

ABSTRACT

While the concept of Software Defined Networking (SDN) has seen a rapid deployment within the data center 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 fiber 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.

1. 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 centers, 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 centers [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

* Corresponding author.

http://dx.doi.org/10.1016/j.yofte.2015.08.008
1068-5200/

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 fiber 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.

2. 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 center 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 Virtualization (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.

3. Architecture

The architecture we have chosen for our SDN control plane implementation is shown in Fig. 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 [31–33].

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