Enabling the Future Optical Internet with OpenFlow: A Paradigm Shift in Providing Intelligent Optical Network Services

Dimitra Simeonidou, Reza Nejabati, Siamak Azodolmolky
High Performance Networks Group (HPNG), School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, Colchester, Essex, CO4 3SQ, UK
Tel: +44-1206-873496, Fax: +44-1206-872900, e-mail: {dsimeo, rnejab, sazod}@essex.ac.uk
(Invited)

ABSTRACT
This paper proposes an optical networking paradigm suitable for future Internet services enabled by OpenFlow. The OpenFlow technology supports the programmability of network functions and protocols by separating the data plane and the control plane, which are currently vertically integrated in routers and switches. OpenFlow facilitates fundamental changes in the behaviour of networks and their associated protocols. This paper introduces an OpenFlow optical network architecture enabled by optical flow, optical flow switching elements and programmable OpenFlow controllers. The proposed solution allows intelligent, user controlled and programmable optical network service provisioning with the capability to operate any user defined network protocol and scenario.

Keywords: OpenFlow, Open Flow Switching, Optical Flow, Extended OpenFlow Controller, Intelligent Optical Networking, Software defined networking

1. INTRODUCTION

The Internet traffic continues to exhibit exponential growth. A major IP router manufacturer has forecast that the Internet traffic will grow at a compound annual growth rate of 34% from 2009 to 2014 [1]. This traffic growth is driven by existing and emerging high performance applications such as ultra high definition (UHD) video on demand streaming, 3D tele-presence, and cloud computing to name a few. The sum of all forms of video (TV, pee-to-peer, Internet, and video on demand) will continue to exceed 91% of global consumer traffic by 2014 [1]. These types of emerging applications and their resulting traffic, when aggregated for transport over core networks can be only handled by high-capacity WDM circuit switched optical networks. A major challenge for network operators to support the future Internet applications, is the efficient end-to-end delivery of packet switched traffic in a cost effective manner (with respect to CAPEX and OPEX), when traversing from a packet switched domain (i.e., access or aggregation) through a circuit switched domain (i.e., WDM core optical networks).

The OpenFlow paradigm has recently been proposed as a control framework that supports programmability of network protocols and functionalities (i.e., software defined networking) by decoupling the data plane and the control plane, which are currently vertically integrated in many networking equipments (e.g., routers, switches, access points) [2]. OpenFlow technology is being adopted by most of the major packet switching vendors as the key technology enabler for realization of software defined networking. There are currently a number of commercial products, which support OpenFlow technology especially on metro and campus and metro grade. OpenFlow control framework is a promising technology for integrating the control and management of packet switched domain and optical circuit switched domain. It provides a framework for development of innovative functionalities and protocols thanks to its support for software defined networking. OpenFlow adopts the concept of flow based switching and network traffic control for intelligent, user controlled and programmable network service provisioning with the capability to execute any user defined routing, control and management application in software and outside the data path, in the OpenFlow controller.

In this paper we introduce an OpenFlow-based optical networking architecture. In order to properly put the building blocks of the proposed architecture in the context, we present the key features of the OpenFlow as an enabling technology, the role of OpenFlow controllers in providing intelligence and programmability to the network, and eventually the proposed OpenFlow-based optical networking architecture.

2. OPENFLOW SWITCHING

The concept of OpenFlow switching originates from the fact that most Ethernet switches and routers include a flow table (typically constructed from ternary content-addressable memory), which run at line rate to implement switching, routing, firewalls, network address translation (NAT), Quality of Service, and other functionalities. The main idea behind the OpenFlow switching is to extract the control of the flow-table of the OpenFlow switch and to delegate it to an external programmable and flexible controller. OpenFlow provides an open protocol to program the flow table of these networking elements (i.e., routers, switches, access points, etc.).

OpenFlow abstracts each data plane switch as a flow table. In the OpenFlow switching context, flows are broadly defined, and are only limited by the capabilities of the particular implementation of the flow table. For instance, a flow could be all packets from a particular MAC address, or a particular TCP connection, or all packets destined to a specific IP address, or with the same VLAN tag, or from the same switch port. As a general or non-IPv4 packets, a flow could be defined as all packets, which match a specific (but non-standard) header.

The key building blocks of an OpenFlow switch are: 1) a flow table, with an action associated with each flow entry, to instruct the OpenFlow switch how to process the flow, 2) A secure channel that connects the switch to a remote control process (i.e., OpenFlow controller), which provides a secure connectivity between the controller and the switch and 3) the OpenFlow protocol, which provides a standard and open mechanism for a controller to communicate with a switch. The entries of the flow table can be defined externally using the OpenFlow protocol.

Therefore, an OpenFlow switch is a data path element that forward packets between ports, as defined by a remote control process (i.e., OpenFlow controller). The data path of an OpenFlow switch consists of a flow table and an action associated with each flow entry. Three basic actions are defined: 1) Forward the flow's packets to a given port (or ports), 2) Drop the flow's packets, and 3) Encapsulate and forward the flow's packets to a controller (typically for the first packet in a new flow, so a controller can decide if the flow should be added to the flow table).

In the OpenFlow control framework a network is managed by a network-wide operating system running on top of a controller that controls the data plane with OpenFlow protocol. The OpenFlow controller is a server, which has the capabilities to host different network management and control applications to effectively manage the network in a centralised or distributed way. The control plane makes decisions as to how each flow is forwarded (reactively as new flows are detected, or proactively in advance), then caches its decision in the data plane's flow table. In a packet switched network a flow can be defined in a flexible way as a combination of any L2, L3, L4 headers of a packet as shown in Figure 1. Statistics field is used to gather packet statistics, which are used by the network applications to make dynamic and/or automatic decisions.

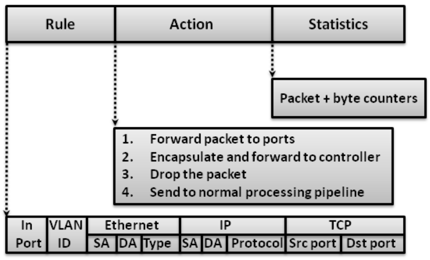

Figure 1.Structure of a flow table inside an OpenFlow switch.

In an OpenFlow controlled packet switched network, incoming packets are matched against the flow definitions; if there is a match, a set of actions are performed, and statistics will be updated. Packets that do not match any flow-table entry are (typically) encapsulated and sent to the controller. The controller can decide how to process the packet and then insert its decision in the data plane (i.e., a new rule in the flow-table). Thus, consequent packets in the flow are processed according to this new rule, without contacting the controller. Therefore, while each packet is switched individually, the flow is the basic unit of manipulation within the switch. Note that by switch we are refereeing to a generic OpenFlow enabled switch. The fields of the flow table are shown in Figure1.

3. OPTICAL OPENFLOW ARCHITECTURE

The OpenFlow controller is a server, which has the capabilities to host different network management and control applications to effectively manage the network in a centralised or distributed way. This separation between the control and data plane and the capability to treat packet and circuit traffic as flows makes OpenFlow protocol a single control for both packet and circuit networks. OpenFlow control framework is a promising technology for integrating the control and management of packet switched domain and optical circuit switched domain. There have been several attempts and proposals to control both circuit switched and packet switched networks using the OpenFlow protocol [3], [4]. A unified OpenFlow based control architecture is depicted in Figure 2. The OpenFlow protocol, the OpenFlow controller and the network applications construct a unified control plane. In this section, we propose an Optical OpenFlow networking architecture, which is enabled by the integration of OpenFlow and GMPLS control planes.

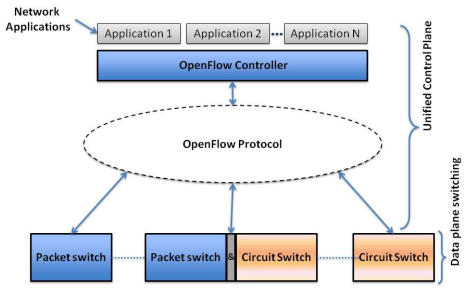

Figure 2.A unified control plane based on OpenFlow protocol, controller and network applications.

Generalized Multi Protocol Label Switching (GMPLS) developed in IETF as a generic network control plane framework is used for managing physical path and core tunnelling technologies of the Internet and Telecom service providers. The GMPLS architecture extends MPLS (Multi Protocol Label Switching) to encompass control and management of time-division (e.g., SONET/SDH, PDH, G.709), wavelength, and spatial switching (e.g., incoming port or fibre to outgoing port or fibre). The GMPLS control plane, due to its support for various optical transport technologies as well as its capability for dynamic and on demand lightpath provisioning, is widely being considered by operators as the control plane of their next generation core optical networks.

Figure 3 shows the architecture of the proposed integrated OpenFlow-GMPLS control plane. It utilizes an extended packet switch OpenFlow controller integrated with a GMPLS control plane in an overlay model. The GMPLS control plane follows the standard ASON model and its building blocks include: the network connection controller (NCC) responsible for handling and processing the connection requests, the signalling controller (SC), which implements the RSVP-TE protocol and is responsible for handling the GMPLS signalling, the routing controller (RC) comprising OSPF-TE protocol and a path computation algorithm for calculating the end to end path, the link resource manager (LRM) which is responsible for monitoring and information collection regarding status of network elements and the transport network resource controller (TNRC) which provides the configuration and monitoring interface between GMPLS and network elements. The extended OpenFlow controller comprises: the flow processor (FP), which is responsible for processing new flows and creating flow rules and updates the flow tables in switches, the path computation element (PCE), responsible for path computation for each flow within each packet switched domain, the discovery agent (DA), which is responsible for discovering the network topology and connectivity including end points in each packet switched domain and the OpenFlow Gateway (OFGW) that provides the interface between the OpenFlow controller and the GMPLS control plane through a user network interface (UNI). OpenFlow protocol controller (OFPC) is responsible for interfacing OpenFlow Controller and OpenFlow enabled switches for configuration and monitoring of their flow table.

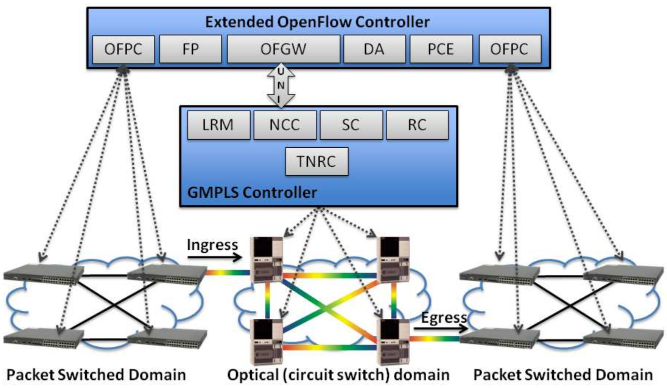

Figure 3.Integrated GMPLS-OpenFlow control plane.

In the proposed architecture the packet switched domain and the GMPLS optical domain are controlled by an extended OpenFlow controller. The extended controller provides functionality for requesting optical connectivity from the optical domain. It is assumed that the edge packet switch, which interconnects the optical domain to the packet switched domain, is equipped with tuneable WDM interfaces and is connected to an add/drop port of the ingress/egress nodes of the optical domain as shown in Figure 3. Initially the flow tables of the switches are empty. The first packet of the request will be forwarded to the extended OpenFlow controller. Based on the destination address of the request, the extended OpenFlow controller identifies the end point of the flow. In this context, the *Optical Flows* can be defined as those flows, which are destined to another packet switch domain through the optical (circuit switched) domain (i.e., from one packet switch domain to another one through the optical core network). In fact for the optical flows, the extended OpenFlow controller identifies the end point of the lightpath in the circuit switch domain and requests (via OFGW) a lightpath from GMPLS control plane. The interface for this request is UNI. Once the lightpath is established the extended controller updates the flow table of the ingress and egress switches, finalizing the establishment of an end-to-end optical flow path. The extended OpenFlow controller also maintains a list of established lightpaths in terms of lightpath identifiers. After this process the new optical flow entry will be inserted to the flow table of the ingress switch, which instructs it to put the optical flows to the proper port and wavelength in order to traverse the circuit switched domain. The flow table entry of the switch after processing of a new flow is shown in Figure 4a.When the flow table of the ingress and egress switches are updated, the packet stream from the packet domain will enter the ingress switch. The flow entry in the ingress switch maintains the stream of the packets (i.e., optical flows) to the proper switch port (for instance indicated as port ‘X’ in Figure 4a) and on proper wavelength (e.g., λ_i as indicated in Figure 4a). Since the end-to-end lightpath is already established inside the circuit switch domain, the optical flow traverses the optical core network and leaves the circuit switch domain at the egress boundary switch. The flow entry of the egress switch guides the traffic to the proper egress port towards the packet switch domain in the destination.

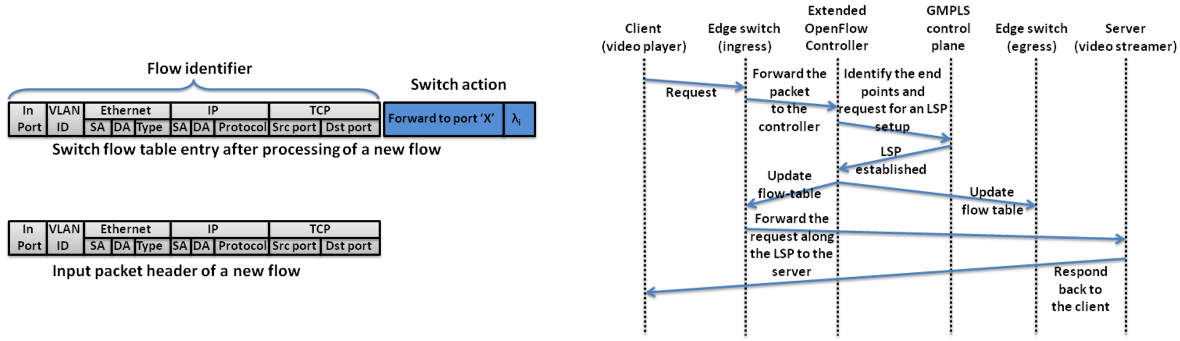

Figure 4. a) Extended flow table entry in the ingress switch (left) b) Timing diagram for a use-case (right).

As a use-case we can consider the on demand streaming scenario of UHD video content. The UHD player (i.e., the client) sends the request to the ingress switch. Since its flow table is initially empty the ingress switch encapsulates the packet and forwards it to the extended OpenFlow controller. The extended OpenFlow controller identifies the egress switch based on the global routing information and establishes a lightpath to from ingress to the egress node (using GMPLS control plane and OFGW interface). Then, the flow table of the switches will be updated and the request will be forwarded to the video server. Given the established lightpath, the UHD content will be sent back to the client for presentation. The corresponding timing diagram for this scenario is depicted in Figure 4b. Unifying the control of the packet and circuit switching domains using an extended OpenFlow controller and sharing the databases of the GMPLS control plane and extended OpenFlow controller, pave the way for novel and intelligent network applications (e.g., intelligent load balancing, location based content delivery, enforcing different QoS levels, etc.) that can be developed on top of the extended (and unified) OpenFlow controller.

4. CONCLUSIONS

In this work we presented a novel approach to unify the packet switching and circuit switching domains utilizing an extended OpenFlow control framework. The main idea behind the OpenFlow is to decouple the data path of the generic switching elements (i.e., routers, switches and access points) from the control plane. On the other side the GMPLS control plane is the main control plane candidate of the core optical networks. Extending the capabilities of the OpenFlow controller, when integrated with a GMPLS control plane, paves the way for a unified control framework, which can serve the development of intelligent network applications. We also presented a use-case scenario for UHD content delivery, across the packet and circuit switched domains.

ACKNOWLEDGEMENTS

The work presented in this paper is partially funded by the EU FP7 funded project OFELIA.

REFERENCES

[1] Cisco,“Cisco Visual Networking Index:Forecast and Methodology, 2009–2014,”white paper, 2 June 2010.

[2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, “OpenFlow: enabling innovation in campus networks,” ACM SIGCOMM Comp. Comm. Rev., vol. 38, no. 2, pp. 69-74, April 2008.

[3] S. Das, G. Parulkar, P. Singh, D. Getachew, L. Ong, N. McKeown, “Packet and Circuit Network Convergence with OpenFlow,” Optical Fiber Conference (OFC/NFOEC10), San Diego, March 2010.

[4] V. Gudla, S. Das, A. Shastri, G. Parulkar, N. McKeown, L. Kazovsky, “Experimental Demonstration of OpenFlow Control of Packet and Circuit Switches,” Optical Fiber Conference (OFC/NFOEC’10), San Diego, March 2010.