Software-Defined Fragmentation-Aware Elastic Optical Networks Enabled by OpenFlow

Lei Liu(1), Yawei Yin(1), Ming Xia(2), Meral Shirazipour(3), Zuqing Zhu(4), Roberto Proietti(1), Qing Xu(3), Stefan Dahlfort(2), S. J. Ben Yoo(1)

(1) University of California, Davis, One Shields Ave, Davis CA 95616, USA, liulei@ieee.org
(2) Ericsson Research Silicon Valley, 200 Holger Way, San Jose, CA 95134, USA
(3) Ericsson Research, 8500 Decarie Blvd, Montreal, Quebec H4P 2N2, Canada
(4) University of Science and Technology of China, Hefei, Anhui 230027, China

Abstract *We present a software-defined fragmentation-aware elastic optical network, in which an extended OpenFlow-based control plane intelligently routes connection requests to avoid spectrum fragmentation. The overall feasibility and efficiency of the proposed scenario is validated by using both numerical simulation and experimental demonstration.*

Introduction

Optical networks are evolving from a fixed ITU-T DWDM wavelength grid to a flexible grid[1] in which the optical spectrum is divided into fixed-size spectrum slots (6.25 or 12.5 GHz each). The required spectral resources for a connection request in such a flexible grid or Elastic Optical Network (EON) are adaptively allocated by assigning the necessary number of contiguous spectrum slots according to the traffic bit rate and modulation format[2].

However, the setup and release of connections in a dynamic network scenario can fragment the optical spectrum into non-continuous small pieces. Such a spectrum fragmentation problem may lead to inefficient resource utilization and a high blocking probability if not dealt with properly.

To address this issue, several algorithms[3-6] for spectrum defragmentation have been proposed recently. Despite significant progress, it should be noted that all previous studies have focused on algorithm design. The issue regarding how to deploy a defragmentation mechanism in a real operational scenario by using a network control plane has not been addressed yet, especially through an experimental approach.

To this end, in this paper, we present a software-defined approach enabled by OpenFlow[7] for deploying a fragmentation-aware EON. All the proposed algorithms and OpenFlow protocol extensions are validated by using network simulation and experimental demonstration, verifying their overall feasibility and efficiency for spectrum defragmentation.

Network architecture

The network architecture for a software-defined fragmentation-aware EON is shown in Fig.1. The data plane is deployed with bandwidth-variable wavelength cross-connects (BV-WXC), which are implemented by using BV-wavelength selective switches (BV-WSS)[1]. Each BV-WXC is controlled by an OpenFlow agent, which is referred to as an OpenFlow-enabled BV-WXC (OF-BV-WXC). A dedicated OpenFlow controller (e.g. NOX[8]) is introduced to control all the OF-BV-WXCs through the extended OpenFlow protocol (as detailed next). Thanks to the centralized architecture of OpenFlow, the NOX controller is able to know, and can dynamically manage spectrum slots usage information (i.e., free or occupied) for all the links in the network. This information can be utilized by the NOX to perform a fragmentation-aware Routing and Spectrum Allocation (RSA) algorithm to intelligently route the incoming requests to avoid spectrum fragmentation.

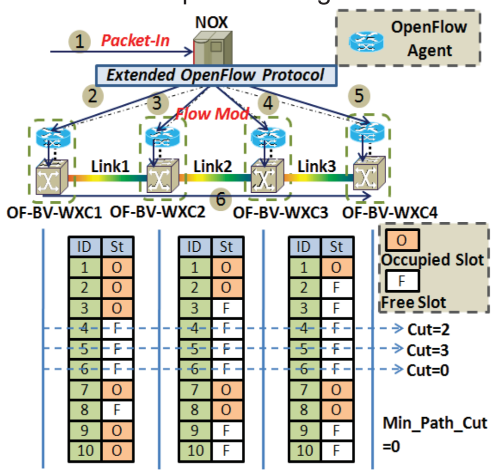

Fig. 1: Network architecture and the assessment of spectral fragmentation

To quantitatively assess the spectrum fragmentation in a EON, several metrics have been proposed[6, 9]. In this paper, we use the parameter "*Cut*"[6], which is a nonnegative integer (e.g. 0, 1, 2…) accounting for the number of consecutive spectrum that a new connection will break. For example, as shown in Fig.1, an incoming connection from WXC1 to WXC4 for one slot bandwidth can be assigned with slot 4, 5 or 6 on a given path WXC1→2→3→4. The provisioning with slot 4 will break the consecutive spectrum blocks on link 2 and link 3, so the "*Cut*" value is 2 for this assignment. Likewise, the provisioning with slot 5 or 6 has the "*Cut*" values 3 and 0 respectively. Clearly, for a given connection, the "*Cut*" value is a straightforward metric to quantitatively evaluate its newly introduced spectrum fragmentation along a path. The higher "*Cut*" value indicates that more serious spectrum fragmentation will be introduced by this incoming connection. In this paper, we also define a metric "*Min_Path_Cut*", which is the minimum "*Cut*" value for a given path to route a connection request. As shown in the aforementioned example in Fig.1, the "*Min_Path_Cut*" of the path WXC1→2→3→4 to route the incoming connection is 0.

Fragmentation-aware RSA

Two fragmentation-aware RSA algorithms are proposed, which are referred to as the *Minimum Path Cut (MPC)* algorithm and the *Minimum Path Cut with Network Resource Optimization (MPC-NRO)* algorithm respectively.

The MPC algorithm calculates all the feasible paths and then selects a path with the minimum "*Min_Path_Cut*" value to route the request, which can minimize the newly introduced fragmentation for an incoming connection.

MPC Algorithm Description:

Step 1: for an incoming connection request, the algorithm uses the *K*-shortest path first computation to calculate all the routes between a source-destination node pair;
Step 2: according to the path distance and flow bit-rate, the algorithm assigns appropriate modulation formats, and calculates the required number of spectrum slots for each route;
Step 3: the algorithm computes the "*Min_Path_Cut*" value for all the routes which can meet the spectrum continuity constraint[2];
Step 4: the algorithm selects a route with the minimum "*Min_Path_Cut*" value. If multiple routes have the same "*Min_Path_Cut*" value, the algorithm randomly selects one route;
Step 5: the algorithm assigns the spectrum slots with the minimum "*Cut*" value on the selected route for the request. If the connection can be assigned with different spectrum slots with the same "*Cut*" value, the algorithm uses the random mechanism to select spectrum slots.

However, the MPC algorithm only considers the fragmentation for RSA. In order to improve the overall network performance, we propose the MPC-NRO algorithm which is based on the MPC but with two additional steps for network resource optimization.

MPC-NRO Algorithm Description:

Step 1 to Step 3: same with the MPC algorithm;
Step 4: the algorithm selects a route with the minimum "*Min_Path_Cut*" value. If multiple routes have the same "*Min_Path_Cut*" value, the algorithm selects the shorter path in distance, aiming at reducing the resource occupation;
Step 5: if multiple paths are with the same minimum "*Min_Path_Cut*" value and the same length, the algorithm selects the path with more available spectrum slots, aiming at improving load balancing of the whole network;
Step 6: if tie still exists among multiple paths, the algorithm randomly selects one path;
Step 7: the algorithm uses the same approach, as described in the Step 5 of the MPC algorithm, for spectrum slot assignment.

OpenFlow protocol extensions

Fig.1 shows the procedure for path provisioning in a fragmentation-aware EON. As shown in the step 1 in Fig.1, a *Packet In* message triggers the path provisioning according to the OpenFlow methodology. Here, we extend the *Packet In* message to carry the bit-rate information of the incoming client (e.g. IP) traffic[10], and then, according to the source-destination addresses and traffic bit-rate information in the *Packet In* message, the NOX performs the propose MPC or MPC-NRO algorithms to intelligently route the incoming flow to avoid fragmentation. After completion of the fragmentation-aware RSA computation, the NOX controller allocates suitable frequency slots, and then controls corresponding OF-BV-WXCs along the computed path to create a connection with appropriate optical spectrum range through extended OpenFlow *Flow Mod* messages[10, 11], as shown in the steps 2-6 in Fig.1. The extended *Flow Mod* message carries the RSA results from the NOX, including input and output ports, central frequency, number of spectrum slots, and modulation format for each OpenFlow agent to control the underlying hardware.

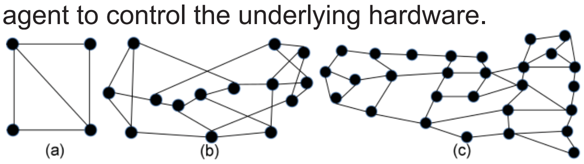

Fig. 2: (a) 4-node mesh network; (b) 14-node NSFNET; (c) 24-node US Backbone

Performance evaluation

We firstly use the experimental demonstration to verify the overall feasibility of the proposed solutions, and then utilize network simulations to validate the efficiency of the MPC and MPC-NRO algorithms in terms of defragmentation.

Fig.2(a) illustrates the testbed setup, which consists of the deployed control plane and an emulated data plane. All the nodes are connected to a dedicated NOX controller which is implemented with both the MPC and MPC-NRO algorithms. The DWDM links are characterized by 64 individual slots of 12.5 GHz each. In this experiment, we consider three different bit rates 100Gb/s, 200Gb/s and 400Gb/s, and three modulation formats DP-64QAM, DP-16QAM and DP-QPSK. Upon receiving an extended *Packet In* message, as the Wireshark capture in Fig.3 illustrates, the NOX performs MPC or MPC-NRO algorithms to route the requests, and then uses the extended *Flow Mod* messages to create an appropriately-sized optical path. We observed that the overall control plane latency to provision a path by using the MPC-NRO algorithm was around 59 ms, which was slightly longer than the MPC algorithm (~51 ms) due to the additional two steps for resource optimization.

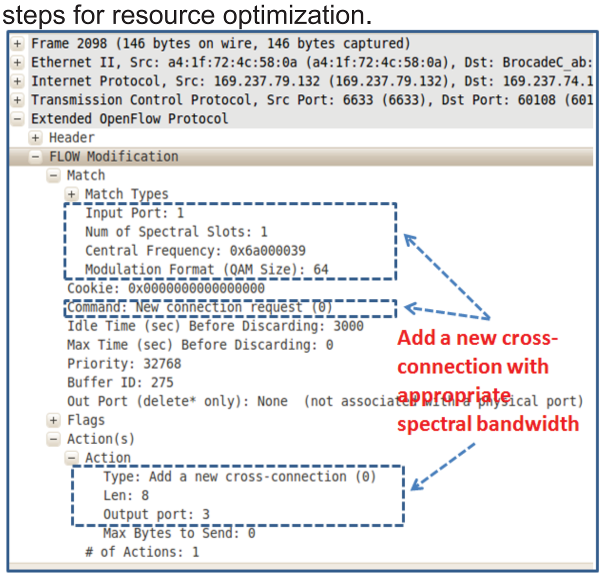

Fig. 3: Wireshark capture of an extended OpenFlow *Flow Mod* message for elastic path setup

We also conducted network simulations to evaluate the proposed algorithms. Dynamic connection arrival and departure events are simulated on a 14-node NSFNET network (Fig.2(b)) and a 28-node US backbone network (Fig.2(c)) respectively. Each fiber link has 400 spectrum slots, and each source-destination pair is generating connection requests randomly according to a Poisson process averages at 0.8~10 arrivals per time units. The holding time of each connection follows a negative exponential distribution averages at 5 time units. Fig.4 shows the blocking probability comparison using different algorithms. The blocking performance results indicate that the MPC algorithm can obtain a similar defragmentation performance compared with the recently proposed alignment-aware RSA[6] algorithm, and that it outperforms the common benchmark RSA algorithm, namely the shortest path routing with first-fit spectrum assignment (SP). The MPC-NRO performs the best in reducing blocking probability since it considers not only the fragmentation, but also the network resource optimization. Simulation on 24-node US backbone network in Fig.5 shows similar results.

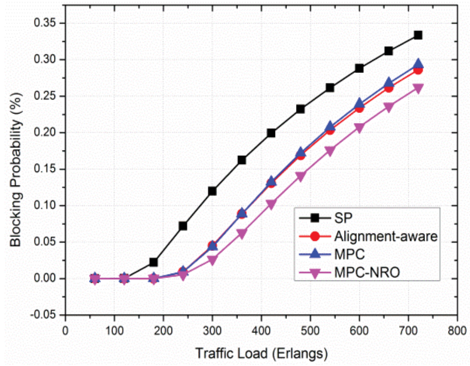

Fig. 4: Blocking probability comparison of different algorithms in the 14-node NSFNET network

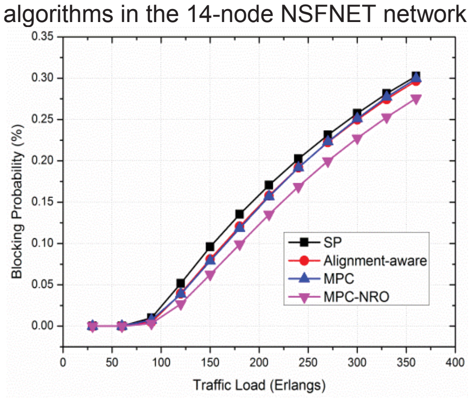

Fig. 5: Blocking probability comparison of different algorithms in the 24-node US backbone network

Conclusions

In this paper, we present a software-defined EON enabled by OpenFlow. Experimental and simulation results validated the overall feasibility and efficiency of the proposed solutions for spectrum defragmentation, and also indicated that OpenFlow is a promising control plane solution to address the fragmentation problem in elastic optical networks, due to its centralized architecture, open interfaces and high flexibility.

References

[1]M. Jinno et al., IEEE Commun Mag, **47**(11),66-73 (2009).
[2]M. Jinno et al., IEEE Commun Mag, **48**(8),138-45 (2010).
[3]A. N. Patel et al., OFC/NFOEC'11, Paper OTuI8 (2011).
[4]T. Takagi et al., ECOC'11, Paper Mo.2.K.3 (2011).
[5]Y. Sone et al., ECOC'11, Paper Mo.1.K.3 (2011).
[6]Y. Yin et al., OFC/NFOEC'13, Paper OW3A.5 (2013).
[7]OpenFlow [Online], http://www.openflow.org/.
[8]NOX [Online], http://www.noxrepo.org/.
[9]X. Wang et al., OFC/NFOEC'12, Paper OTh1A.2 (2012).
[10]L. Liu et al., ECOC'12, Paper Mo.2.D.3 (2012).
[11]M. Channegowda et al., ECOC'12, PD-Th.3.D.2 (2012).