A SDN-based On-demand Path Provisioning Approach across Multi-Domain Optical Networks

*Md Israfil Biswas, * Philip Morrow, * Mamun Abu-Tair, *Sally McClean, *Bryan Scotney and **Gerard Parr

*School of Computing, Ulster University, Northern Ireland, UK
**School of Computing Sciences, University of East Anglia, UK
Email: *{mi.biswas, pj.morrow, m.abu-tair, si.mcclean, bw.scotney}@ulster.ac.uk, **g.parr@uea.ac.uk

Abstract— The interconnection of remote datacentres with optical networks are emerging use cases and such orchestration of multi-domains require the design of new network control, management, and orchestration architectures. Such heterogeneity needs to adopt end-to-end services like on-demand path provisioning. It is acknowledged that such scenarios are more complexed and have fundamental limitations in terms of high performance and delay. To address these issues, and as a means to cope with the complexity growth, research in this area is considering the concept of Software-Defined Network (SDN) orchestration for multi-domain optical networks to coordinated the control of heterogeneous systems. This paper presents a SDN path provisioning approach across Multi-Domain Optical Networks. The aim is to develop an efficient on-demand path provisioning platform in a software defined optical network at the control plane to dynamically manage the network's load, especially in emergency scenarios. The proposed distributed system architecture will help to solve the longstanding problem of inter-domain path provisioning. Our proposed architecture is implemented and validated in a control plane testbed to validate the approach. The paper also evaluated the factors such Quality of Service (QoS) of the network deployment associated with delay or control overhead. Our results show that the method will reduce additional delays in a multi-domain optical network, where high capacity and low latency are requirements for data-intensive applications and cloud services. The proposed method also maintains the total number of flows as low as possible to make the algorithm fast and reduce overheads.

Keywords— *SDN, Network Monitoring, Optical Network Control and Management, OpenFlow.*

I. INTRODUCTION

The isolation of the network control functions from the forwarding elements is one of the significant advantages SDN. SDN centralizes the control functions into an individual control plane, compared to the distributed and autonomous structure of current traditional networks by enabling virtualization [1]. SDN uses the OpenFlow protocol [2] to manage data and switching control. OpenFlow-like protocols are required to implement the SDN paradigm using new network elements to incorporate Network Function Virtualization (NFV) [3]. Moreover, in response to mobile (5G), enterprise and datacenter networks, the need for virtualization technologies has grown rapidly. Typically, virtual networks use address, topology, and control function isolation to achieve virtualization. Virtual networks can provide various benefits such as on-demand path provisioning, programmability, resource provisioning and utilization with agility and flexibility. Our previous paper on virtualization capabilities like geographically isolated and transparent resource provisioning [4] also shows these advantages.

However, with the emergence of data-intensive applications and dynamic provisioning in the cloud Datacentre (DC) the SDN-based service provider networks are complex and multi-layer in nature. The service providers provision and manage each layer (e.g., packet and optical) separately with the help of SDN and the transport networks are facing more challenges. Therefore, a multi-layer and multi-domain transport solution are proposed to address the threats and benefit the service providers in heterogeneous network scenarios. The traditional approach network operators take to implement customer service requests is that the operator must first study its current network topology. Moreover, it must evaluate the performance metrics for each link and path between multiple sources and destination pairs. The system must use a planning tool to compute the new path. This is to ensure that it meets the customer requirements and then make reconfiguration for the new path to provide the customer service. However, the provisioning and adding of capacity or new services requires an additional delay. The additional delay in a multi-domain optical network is an issue while high capacity and low latency are the requirements for data-intensive application and cloud services. It is expected that the inter-domain exchange should be transparent in the optical layer and a converged SDN control plane for the packet and optical networks can address these inadequacies. Hence, proper utilization of resources is essential and service providers can optimize across packet and optical layers in real-time by on-demand provisioning and management, which is our aim for this work.

This work is an extension of our previous work [5] considering multilayer and multi-domain networks. The SDN paradigm is a feasible way to realize full virtualization solutions for business needs and a policy based framework is used for on-demand services, based on software implemented features defined in a logically centralized control plane. Using our SDN path provisioning approach, the service provider can identify the best path based on the constraints specified. This way, the service provider does not have to spend hours researching if the possible paths have enough bandwidth or which links have high latency or other path constraints. These capabilities enable network operators to accelerate time to revenue for new services and deliver a higher level of customer service. Our approach can help to maintain a low total number of flows, can add capacity based on traffic and address other considerations very quickly.

A. Related Works

Many research studies related to considering the concept of SDN orchestration for multi-domain networks to coordinate the control of heterogeneous systems have been undertaken. However, new multi-layer network architectures with on-

IEEE computer society

demand path provisioning are required to manage the packet-optical architecture effectively.

A SDN-based approach is presented in [6] to provide multi-domain on-demand bandwidth services to enable the automation of service provisioning, resources utilization, with the support for failure recovery mechanism. However, todays multi-domain networks are more complex with multi-layer heterogeneity.

In [7], a simple path provisioning architecture is proposed using a SDN controller to manage multi-layer switching for end-to-end paths. However, today's datacentre environment is more complex and requires more an optimized multi-domain transport solution.

R. Casellas et al [8] proposed and implemented an architecture for multi-domain optical networks with heterogeneous control planes supporting both Generalized Multiprotocol Label Switching (GMPLS) and OpenFlow deployments. The architecture has been validated considering mesh of SDN controllers on an experimental testbed focusing on aspects such as service provisioning latency and control plane overhead. However, this type of architecture has scalability issues for multi-domain and required peering agreements with operators.

In [9] a virtual network platform is presented considering various features including on-demand provisioning, flexibility, heterogeneous physical network infrastructure etc. with a virtual federation model. The problem with such a platform is the scalability with updates of the open-source software, especially for the promising Network Virtualization (NV) solution to meet business requirements.

Using our SDN-based path provisioning approach, service providers can quickly and easily create the data transport service path that can be used to automate fulfilment of user requests/requirements, based on various limitations. Service Level Agreement (SLA) may be applied to address a variety of path provisioning use cases, such as diversity, low latency, disaster recovery and data sovereignty. In summary, this paper is unique in the following aspects:

- We provide an on-demand provisioning approach in a multi-domain environment that will reduce additional delay by shorter response time.
- Our path reconfiguration experiments show the benefit of the proposed method by reducing latency to match varying traffic demands.
- More frequent path reconfigurations can increase the number of packet drops which leads to retransmissions and finally degrade the application performance. The proposed approach can reduce the number of packet drops and improve the network performance.
- Our approach to transport solution with SDN maintains the total number of flows as low as possible hence, makes the algorithm fast and reduces the overhead.

The rest of the paper is organized as follows: Section II describes the multi-domain transport solutions with SDN. Section III presents the proposed transport solutions for dynamic path provisioning with SDN. Section IV describes the evaluation of the proposed method with the experiment setup and results. Finally, section V provides some conclusions and a view for future work.

II. MULTI-DOMAIN TRANSPORT WITH SDN

In order to gain a service delivery path, the control and the data planes are separated in a SDN. Hence, a centralized SDN controller needs to be implemented to control and manage the multi-layer or multi-domain network elements. The controller core functions should be scalable for inter-domain exchange in a multi-layer environment. A transport solution for a multi-domain SDN architecture which is re-routable regardless of whether it is electrical (packet-switched) or optical (circuit switched) to improve the network performance is therefore required. Figure 1 shows an overview of a typical Multi-domain Optical Network. An aggregated SDN control plane for the packet and optical networks with on-demand path provisioning can help address all these inefficiencies. On-demand path provisioning across packet and optical layers can improve network performance in real-time for availability and economics.
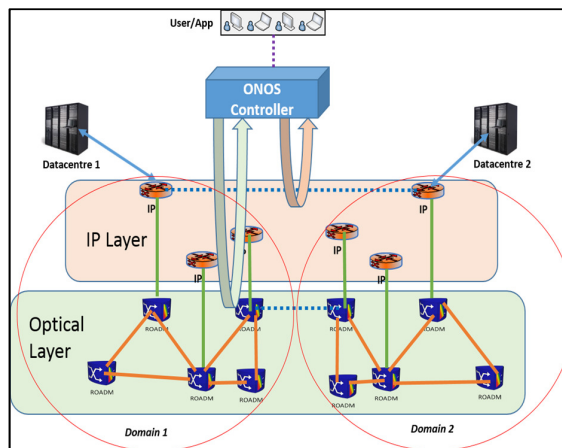

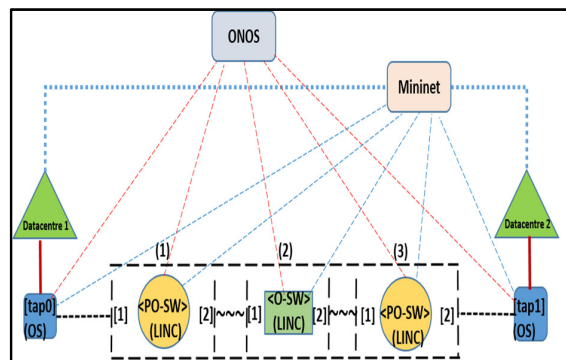Fig. 1.   Overview of a Multi-Domain Optical Network with SDN

Fig. 2.   A Typical POC environment with LINC-Switch to Simulate Simple Optical Network

This can reduce over- provisioning by adding capacity based on traffic and other considerations very efficiently and quickly manners. This will help to provide and manage efficient and faster end-to-end paths among the mixed circuit and packet network environment considering various scenarios. We have used OVS (open vSwitch) in Mininet [10] to emulate packet routers and switches. The packet switches are directly attached to hosts, which represent hosts or datacentres. The emulated packet switches are interconnected with optical ROADMs (Reconfigurable optical add-drop multiplexer) for transport.

ROADM adds the ability to remotely switch traffic from a wavelength-division multiplexing (WDM) system at the wavelength layer. The OpenFlow match/action tables for the ROADMs can be represented by the service provider's transport networks by cross-connecting the lambdas to establish paths at the optical layer. The IP routers are connected physically to the ROADMs, and the chain of ROADMs establish the transport topology. Hence, ROADMs are accountable for all traffic passing through the cross connected optical paths.

ONOS (Open Networking Operating System) [11], is known as the distributed SDN control platform motivated by the requirements of large operator networks. The requirements (e.g., performance, scalability, and availability etc.) typically operate in large and complex multi-layer networks that ONOS provides support in a multi-layer network. ONOS has the ability of on-demand provisioning of bandwidth between datacentres, and handling of failures automatically with seamless restoration in both packet and optical layers. Hence, ONOS aims to provide WAN operators with real-time on-demand network programmability.

The LINC-OE [12] is the optical emulation for the LINC-switch used in this work for optical integration considering a multi-domain network. LINC-OE helps to emulate the ROADM, whereas, the LINC-Switch is used to simulate a simple optical network as shown in Fig.2. In the figure, a logical switch number that related to the tap (e.g., tap0). tapX is the operating system tap interface number X. PO-SW indicates the packet-optical logical switch that has Ethernet ports as well as optical ports. O-SW - indicates the optical switch that has only internally emulated optical ports and links between them. The Optical Space Switch is the switching fabric of the optical network architecture built on a hybrid network [13]. The PO-SW switches (layer 2/3 switches) are simultaneously connected with an optical space switch (O-SW) that uses Direct Fibre Link (DFL) designed and implemented for the proposed approach (a switching substrate that provides an optical circuit between any idle input and output ports, without optical to electronic conversion [14]) as depicted in the figure.

III. THE PROPOSED APPROACH

The proposed method is based on the utilization of a requested link, which is able to provide the status of the link inside OpenFlow domain using ONOS. ONOS extends its on-demand provisioning of resources by supporting multi-domain operations through REST interfaces and APIs. We developed an interface that makes possible to avoid over utilization through monitoring or help recovery in emergency situations to support multi-domain path provisioning by rerouting to DFL paths.
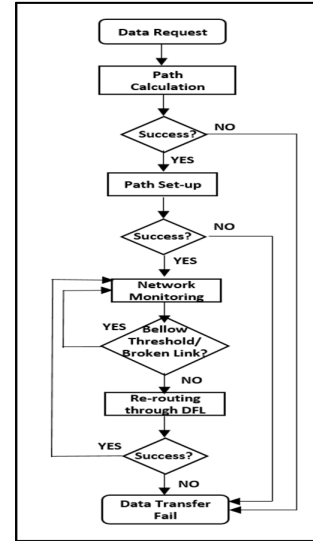

Fig. 3.   Flow Diagram of the Proposed Approach

Fig. 3 shows a flow diagram for the proposed approach for multi-domain transport with SDN. The proposed method first computes the set of all possible paths for the data transfer requests from a source to destination. For each path, the proposed approach will calculate the utilization of the link where the cost of the link (i.e. the threshold) can be accepted or, if the link is broken then the utilization is calculated for each path considering how much load can be accepted. When the flow is rerouted through high-speed DFL, the acceptable load of the path is updated.

The Network Monitoring section calculates the link utilization periodically where it sends updates to a device. The device responds with the timing profile. The controller stores and uses this information to calculate port utilization to predict link performance and status.

Our approach is able to communicate with connected devices and able to query the flow tables, fetch the flow entries as required for rerouting through the DFL. The data fetched from the Network Monitoring and during the path calculation, are used to calculate the network load and capacity. The link statistics then can be used to determine the link threshold. Therefore, the SDN controller predicts the link capacity or the broken link situation to distribute load through the DFL.

As mentioned in the Algorithm 1, the DFLshift is defined as the total load which must be shifted to avoid congestion due to over utilization. AVGtx is the average transmission rate of an outbound port. The flows are rerouted iteratively from the active path to the DFL path by the SDN controller. When the flow reaches the threshold or the selected link is broken, for each iteration the chosen flow is rerouted to the DFL path that is capable of accepting the flow's load. Hence, this policy will help to prevent congestion for emergency migration during any broken links to balance the network path after rerouting. The pseudocode used for this approach is shown in Algorithm 1 that describes how the flows are rerouted to the DFL path. Our approach maintains the total number of flows as low as possible to make the algorithm fast and reduce the overhead.

This is an iterative process to check if the requested data transfer is done i.e., flows with congested devices are free then the DFLShift is zero. If any link is broken, the requested flows will be rerouted to a DFL route flow.

Algorithm 1: Rerouting through DFL

```
1: if any Link is Broken then
2:       DFLreroute Flow <- RequestedFlows (R)
3: end if
4: DFLShift<- AVGtx - (0.7 * LinkCapacity)
5: F <- RequestedFlows (R)
6: N<- NoOfFlows(F)
7: k <- 0
8: j <- N -1
9: for (i=0; i <= j and DFLShift> 0; i++) do
10:      DFLreroute <-F[i]
11:      flowRate <-DFLreroute Flow
12:      if flowRate >= DFLShift then
13:          DFLgrantedPath <- DFLCapacityFreePath
14:             if  DFLgrantedPath != NULL then
15:                Reroute DFLreroute to DFLgrantedPath
16:                Update the granted load of DFLgrantedPath
17:                DFLShift <-0
18:                break
19:             end if
20:      end if
21: end for
22: for (DFLShift > 0 and j >= 0) do
23:      DFLreroute <-F[j]
24:      flowRate <-DFLreroute Flow
25:      DFLgrantedPath <- DFLCapacityFreePath (rerouted flows)
26:      if a DFLgrantedPath != NULL then
27:         Reroute DFLreroute to DFLgrantedPath
28:         Update the granted load of DFLgrantedPath
29:         DFLShift <-- DFLShift- flowwrate
30:            j <- j - 1
31:            i <- 0
32:            goto line 9
33:      end if
3 4:      j <- j - 1
35: end for
```

IV. EVALUATION AND EXPERIMENT RESULTS

A. Experiment Setup

In this section, we investigate response time with or without the DFL paths when an application adds and removes various sizes of intents. Intents are the policy-based directives in ONOS that allows applications to specify the network control requirements. The experiment uses the ONOS built-in app *push-test-intent* to push batches of point-to-point intents through its intent API. The ONOS utility creates the intents depending on the endpoints, batch size and application ID, and returns the latency while the all the intents are successfully installed. ONOS manages inventory of intents in a distributed data store that uses optimistic replication and conversation based techniques. The controller compiles the intents to flow rules as requested and write to the required distributed data stores to distribute the intents and flows. The timestamps are captured on the OF control network with the Wireshark tool, as well as ONOS Device and Graph timestamps recorded in ONOS using *Topology-event-metrics* and *Intents-events-metrics* apps. The metrics app can be enabled from *app activate org.onosproject.metrics.* Therefore, CLI command can be used to get timestamps of various events. The end-to-end timing profile is confirmed by collecting those timestamps of the

events from the initial event triggering when ONOS registers the event in its topology.

TABLE I.   CONFIGURATION PARAMETERS: EXPERIMENT –INTENT ADD REMOVE LATENCY

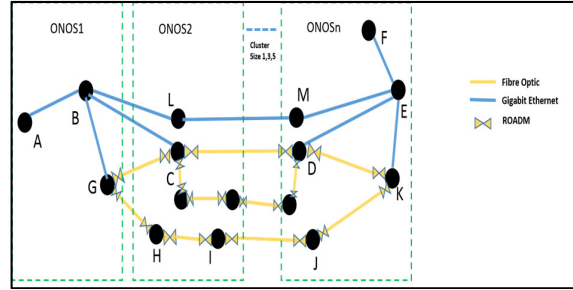| Parameters	Value
Datacentres	2
Packet Switch Nodes	6
Optical Switch Nodes	10
Number of Intents Initially Installed	1,10,100
Number of installed point-to-point intents	6
No of Runs for Each Data Points	20

Fig. 4.   Illustration of on-demand path provisoning with or without DFL
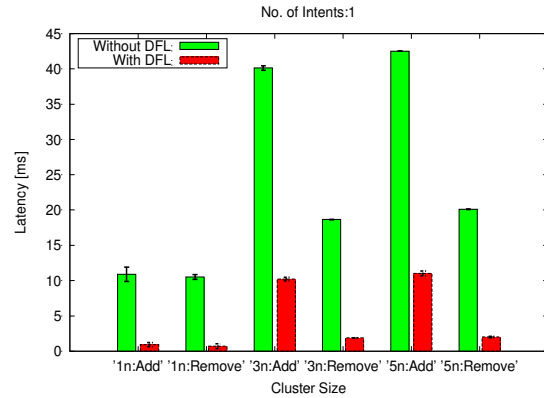
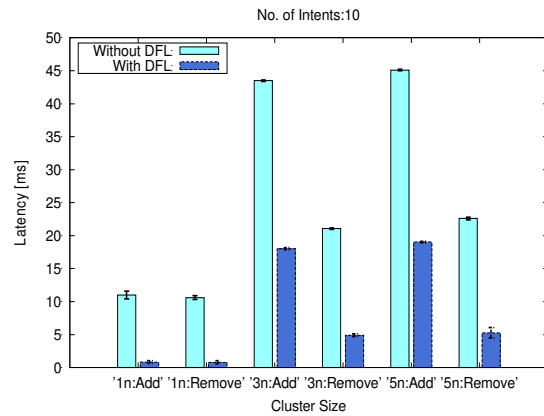

Fig. 5.   Average Latency Comparison with 1 Intent

Fig. 6.   Average Latency Comparison with 10 Intents

Each data point the mean of 20 runs (after 4 warm-up runs) with 95% confidence interval. The ONOS controller uses the North Bound API from the top stack with the help of intents and flow subsystems to the bottom of the stack. In general experiment setup, ONOS can scale from 1 node to a number of nodes. Hence, at least 5-node virtual server cluster is required for these experiments. Here, we have used Mininet version 2.2.1 and OpenFlow version 13 for all the experiments. The ONOS clusters are emulated using Mininet and the basic experimental setup for the experiments includes network connections to communicate within the clusters. Details of the configuration parameters can be found in TABLE 1.

The experimental results of this set of tests provide network operators and application developers with an intensive to design applications based on the response time. Our set of experiments show comparisons with or without DFL including a various size of operating intents, in terms of how large the intents are, and how the cluster size affects the response time. Fig. 4 presents the packet-optical datacentre network topology used for the experiments to compare the path provisioning approach with or without DFL route. For example, a route without a DFL path can be chosen as A->B->L->M->E->F and a route with a DFL route can be chosen as A->B->C->D->E->F. Fig. 5 shows that using 1 intent, increasing the cluster size from 1 node to 5 nodes the average latency can be reduced at least 60% using DFL route compared to without DFL (i.e., packet switching nodes) routes.

Similarly, removing intents using DFL paths reduces the latency compared to using no DFL paths and the average latency can be reduced by at least 70%. Similar results can be found by increasing the number of intents as shown in Fig. 6. We observed that the average latency is increased with the increase of the cluster size. The average latency can be reduced by at least 50% while installing the intents over DFL route. The intents withdrawal latency can also be reduced by at least 72% using DFL paths. Using 100 intents, similar results can be observed comparing the average latency with or without DFL routes. Fig.7 shows that adding intents over DFL route the average latency can be reduced by at least 42%, whereas the average intent removal latency can be reduced by at least 23% using DFL path.
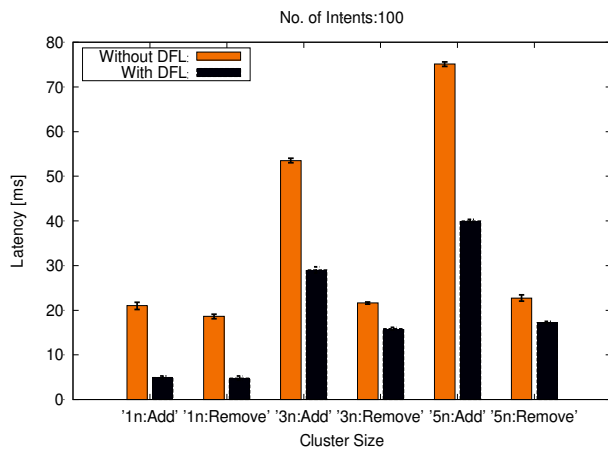
The second experiment characterises the response time in an event when the added intents need to be re-routed due to a path change, e.g. the shortest path is no longer available. The experiments are performed over 1000 run with 95% confidence interval and the VMs are created using iperf [15] tool.

We first install intents confirming that the intents are not automatically removed. For simulation, we break the shortest path and take the initial event timestamp t0 by inspecting the ONOS log for when the new topology description is taken in by the controller. Finally, ONOS re-compiles the intents and flows, and installs them. The final Intent added timestamp (t1) is captured from *Intents-events-metrics*, after verifying all intents are added successfully. The timestamp difference (t1 - t0) is considered as the latency incurred for re-routing the intent(s).

Path reconfiguration is introduced in this architecture in order to compare response time with background traffic to match varying traffic demand. In this experiment iperf TCP connections are used during reconfiguration with a VM size of 50MB and the average topology discovery time is calculated using *Topology-event-metrics*. *Topology-event-metrics* in ONOS calculates the time when ONOS gets connections from OVS switches as the start time and the time when ONOS handles the last Topology Event as the end time.
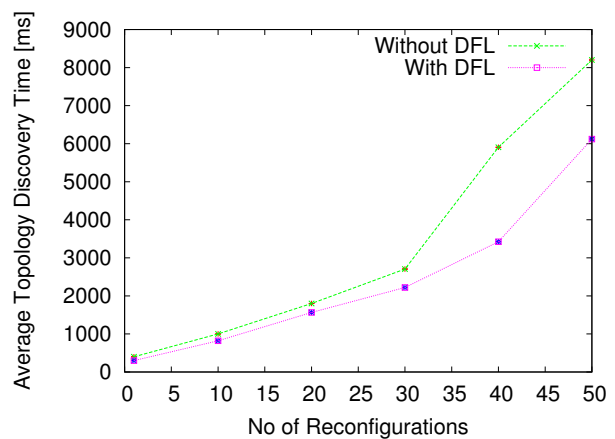

Fig. 8. Average Topology Dscovery Time Comparison during Re-routing

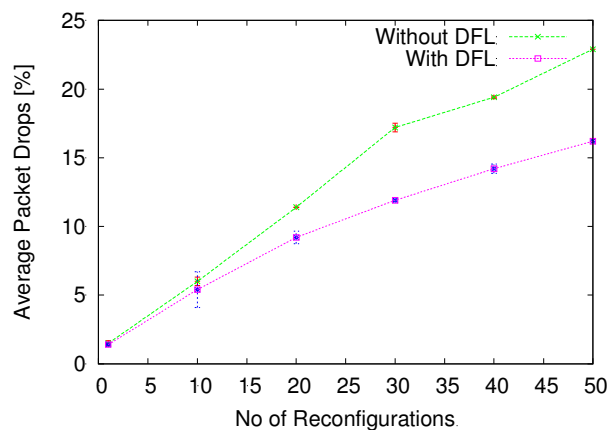Fig. 7. Average Latency Comparison with 100 Intents

Fig. 9. Average Packet Drops Comparison during Re-routing

For example, this tool will simulate a certain amount of switches (in this case with or without DFL), and the switches connect to each other to form a certain network topological structure, test the time of the used by the controller to completely discovery this topological structure. Fig.8 shows the comparison of average topology discovery time during re-routing with or without DFL.

The more the reconfigurations are, the longer it will take to find the topology of the tested controllers. On condition of the same number of switches, the more the paths of topology are, the longer it will take to find the correspondent topology. Generally, the time spent on finding the topology is in a positive linear correlation with the complexity of the paths. During the experiment, DFL path shows lower discovery time with the increase in the number of reconfigurations. This could reduce up to 25% as shown in the figure for 50 reconfigurations. However, a path reconfiguration may interrupt the transfer, and then this leads to packet drops. More frequent path reconfigurations can increase the number of such packet drops which leads to retransmissions and finally degrade application performance. Moreover, the frequency of path reconfiguration is usually enlarged by severe variation of traffic. However, few research studies have addressed the adverse influence of frequent reconfiguration as well as the mechanisms to avoid it. In this experiment, iperf UDP connections are used with a constant bit rate of 100 Mbit/sec to measure the network performance by calculating the average packet drops for 10 sec with or without DFL. According to our experimental results as shown in Fig.9 the growth of reconfiguration frequency leads to higher dropped packet counts without DFL, whereas with DFL paths the packet drops can be reduced by up to 8%.

Therefore, our proposed method of path provisioning not only reduces latency or response time but also improves the network convergence time and affects the ability of the controller to respond to network events in a minimal time. Moreover, the proposed approach significantly reduces the packet loss and hence, can provide minimum overhead to the controller. The results show the increase the overall performance of the SDN controller in a multi-domain optical network and capable to handle unpredicted variable network load while maintaining a continuous network operation, even when there is a link failure.

V.   CONCLUSION AND FUTURE WORK

This paper proposes a SDN path provisioning solution for a multi-domain optical network by providing an efficient algorithm to improve the network performance. In this method, the SDN controller monitors the network conditions and determines if there is any link broken or any link is over utilized. If any link is broken or the link utilization reaches the threshold point, the controller reroutes the flow through a DFL path. Our experiments show the proposed method has a short response time with improved network performance during path rerouting. Our path reconfiguration experiments have observed more interruptions in data transfer without DFL path, which leads to higher packet drops without the proposed approach. The paper is an initial attempt towards on-demand path provisioning handling the reliability, scalability, interoperability, and fault-tolerance in multi-domain optical networks.

In future, this work will consider implementation of the proposed approach for software defined optical networks including the performance evaluation of a single centralized controller compared with a multi-controller network. This work will also consider simulations for the proposed approach for a set of diverse topology including 5G networks for better network utilization and efficient software defined monitoring, specially energy efficiency for 5G technologies.

REFERENCES

[1] M. Alenezia, K. Almustafaa and K. A. Meerjab, "Cloud based SDN and NFV architectures for IoT infrastructure", Egyptian Informatics Journal, 2018.

[2] ONF, OpenFlow Switch Specification Version 1.5.0, https://www.opennetworking.org/images/stories/downloads/sdnresource s/onfspecifications/openflow/openflow-switchv1.5.0.noipr.pdf, December 2014.

[3] Network Functions Virtualisation (NFV); Infrastructure; Hypervisor Domain, ETSI GS NFV-INF 004 V1.1.1 (2015-01),http://www.etsi.org/deliver/etsi_gs/NFV-INF/001_099/004/01.01.01_60/gs_nfv-inf 004v010101p.pdf

[4] Md Israfil Biswas, Gerard Parr, Sally McClean, Philip Morrow and Bryan Scotney, "SLA-Based Scheduling of Applications for Geographically Secluded Clouds", 1st workshop on Smart Cloud Networks & Systems (SCNS'14), December 3-5 Paris, France, 2014..

[5] Md Israfil Biswas, Mamun Abu-Tair, Philip Morrow, Sally McClean, Bryan Scotney and Gerard Parr, "A Policy Based Framework for Software Defined Optical Networks", The 12th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 3PGCIC-2017 November 8-10, 2017, Palau Macaya, Barcelona, Spain.

[6] Alaitz Mendiola, Jasone Astorga, Eduardo Jacob, Kostas Stamos, Artur Juszczyk, Krzysztof Dombek, Jovana Vuleta-Radoičić, Jordi Ortiz, "Multi-domain bandwidth on-demand service provisioning using SDN", NetSoft Conference and Workshops (NetSoft) 2016 IEEE, pp. 353-354, 2016.

[7] S. Baik, C. Hwang and Y. Lee, "SDN-based architecture for end-to-end path provisioning in the mixed circuit and packet network environment", Network Operations and Management Symposium (APNOMS), 2014 16th Asia-Pacific, 17-19 Sept. 2014, Hsinchu, Taiwan.

[8] R. Casellas, R. Martínez, R. Muñoz, R. Vilalta and L. Liu, "Control and Orchestration of Multidomain Optical Networks With GMPLS as Inter-SDN Controller Communication"", Journal of Opt. Communications and Networking, V. 7, N. 11, Nov. 2015.

[9] Y Han, T Vachuska, A Al-Shabibi, J Li, H Huang, W Snow and JWK Hong, "ONVisor: Towards a scalable and flexible SDN-based network virtualization platform on ONOS", International Journal of Network Management, 2017

[10] "Mininet: An Instant Virtual Network on your Laptop (or other PC) - Mininet." [Online]. Available: http://mininet.org/

[11] ONOS. [Online] Availibie: http://github.com/opennetworkinglab/onos

[12] LINC-OE: LINC-Switch for optical emulation, https://github.com/FlowForwarding/LINC-Switch/blob/master/docs/optical_extension.md

[13] Payman Samadi, Varun Gupta, Junjie Xu, Howard Wang, Gil Zussman, Keren Bergman, "Optical multicast system for data center networks", Optical Society of America, Optics Express, 17(23), p. 22162, 2015.

[14] "Calient S320 optical circuit switching," http://www.calient.com

[15] iPerf - The network bandwidth measurement tool, https://iperf.fr/