Global Journal of Computer Science and Technology: G
Interdisciplinary
Volume 24 Issue 1 Version 1.0 Year 2024
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Achieving Path Reversal in OTN By Control Plane Programmability

By Vishal Vasant Deshpande & Shripad Prabhakar Mohani

Abstract- SDN paradigm has been successfully used in IP networks to bring agility and programmability. Applying SDN to OTN (Optical Transport Networks) has some challenges due to circuit switched nature and physical characteristics of OTN. If SDN and OTN can be modified to adapt to each other, there are immense benefits that can be derived from this combination. This paper presents an innovative design by modifying components in OTN and incorporating SDN concepts along-with it. This design encompasses key SDN concepts such as control plane and data-plane separation, central view of network domain for decision making and software-based programmability. This design uniquely implements the flow-reversal in Optical Ring by programmability that its control plane offers. In traditional OTN, it is not only time consuming but also complex process to obtain this type of flow reversal as it requires hardware configurations. The design presented here manages to rapidly configure the flow using its centralized control-plane programmatically.

Keywords: *OTN (optical transport network), optical control plane, SDON (software defined optical network), optiSystem, openflow, SDN (software defined network).*

GJCST-G Classification: *LCC Code:TK5103.59*

ACHIEVINGPATHREVERSALINOTNBYCONTROLPLANEPROGRAMMABILITY

Strictly as per the compliance and regulations of:

Achieving Path Reversal in OTN by Control Plane Programmability

Vishal Vasant Deshpande [α] & Shripad Prabhakar Mohani [σ]

Abstract- SDN paradigm has been successfully used in IP networks to bring agility and programmability. Applying SDN to OTN (Optical Transport Networks) has some challenges due to circuit switched nature and physical characteristics of OTN. If SDN and OTN can be modified to adapt to each other, there are immense benefits that can be derived from this combination. This paper presents an innovative design by modifying components in OTN and incorporating SDN concepts along-with it. This design encompasses key SDN concepts such as control plane and data-plane separation, central view of network domain for decision making and software-based programmability. This design uniquely implements the flow-reversal in Optical Ring by programmability that its control plane offers. In traditional OTN, it is not only time consuming but also complex process to obtain this type of flow reversal as it requires hardware configurations. The design presented here manages to rapidly configure the flow using its centralized control-plane programmatically. The design was implemented and thoroughly tested using simulator tool OptiSystem. The design is generic and does not depend on any specific vendor component. The design is flexible and can co-exist in end-to-end network setup. This design aims to help fellow researchers in their research work related to SDON (Software Defined Optical Network) and can be used as-is or with suitable modifications. This paper is organized into logical sections, starting with Introduction section that gives background and problem description. The next section is Literature Review which covers review of relevant studies and gaps. Then Methodology section briefly covers research methodology used, how results are collected and stored etc. The Design section depicts the solution by explaining high-level design approach, reasons behind design choices and the detailed design. The Result section denotes the observed outcome and confirms that it matches with expected results. Finally, conclusion section summarizes key points, findings, contribution and future scope.

Keywords: OTN (optical transport network), optical control plane, SDON (software defined optical network), optiSystem, openflow, SDN (software defined network).

I. Introduction

Global mobile data traffic forecast by ITU, indicates data traffic to grow at an annual growth rate of around 55% in 2020-2030. Bandwidth demand is rapidly increasing by up-to 4x per year (predominantly between data centres). As per Cisco statistics [9], the fastest growing component of data-centre traffic is Global cloud traffic. It is seen growing by a huge 40% annual growth rate. Today's cloud, mobility and content services offered by service providers have dynamic needs. In order to meet those needs, Telecom operators require more dynamic and programmable network infrastructure with high capacity. It is essential to support rapid service deployment and provide real-time responsiveness to capacity changes.

Optical Transport Networks (OTN) promises the high capacity and reliable bandwidth fulfilment. But they are not flexible to dynamic needs as they are static and mostly manually provisioned. Optical networks are operated with wavelengths fixed in place and not designed to dynamically change. So, they are not capable of adapting to rapid changes to deliver the flexible services like cloud. Modern Network traffic consists of short bursty traffic, as well as very high bandwidth, high-duration data flows that continue for minutes. Common examples of such high bandwidth persistent traffic are VM migrations, data migrations, or Data warehousing function (MapReduce). [10]. On the other hand, OTN is traditionally a circuit switched network and the Path is set at initial design time. Changes to path requires long time and network can't adapt to dynamically to varying traffic conditions mentioned above.

Software Defined Networks (SDN) paradigm promises flexibility and programmability in network operations. SDN was originally conceived for packet-based IP networks. Hence it is difficult to apply SDN to OTN as is. SDN achieves the programmability by decoupling the data plane and the control plane. In most of the OTN these planes are currently vertically integrated and inseparably hosted along-with data-plane. Even if some designs attempt separately hosted optical control plane, it only allows network management software interactions [16]. It does not expose its services to operator directly i.e. no API or interfaces exposed for issuing commands to control plane. Hence optical control planes cannot be programmed dynamically in present OTN.

This paper presents a design that attempts separation of optical control plane from data plane in OTN. The paper also explains the programmability of optical control plane through software instructions or configurations. The design helps control plane to get centralized view that enables it to take decisions at

Author α: Dept. of E & TC, Govt. College of Engg, Jalgaon Jalgaon, Maharashtra. e-mail: vish_d@outlook.com
Author σ: Head of Dept. E &TC, Govt. College of Engg, Jalgaon Jalgaon, Maharashtra. e-mail: shripad.mohani@gcoej.ac.in

network domain level (instead of at network each element level). The uniqueness of design in achieving flow control (path reversal) programmability in OTN ring.

Traditionally, the design approach for IP-plus-Optical network has been, to place all the network functions within the IP layer (routing, signalling, protection). Such design uses static optical trunks interconnecting these IP layer devices. The network controller which controls IP domains does not have capabilities to configure OTN. It treats OTN merely as a fixed pipe carrying data. This paper takes a unique approach in designing OTN Ring to perform flow-control by programming OTN. Moreover, this design ensures adherence to key concepts of SDN (such as control plane separation, centralized view of control plane, software-based programmability etc), it does not mandate OpenFlow or another southbound interface (SBI). OpenFlow is not yet a complete standard, it is still undergoing significant changes [3] to adapt to OTN. Hence it can be incorporated in this design as a future scope.

II. Literature Review

A detailed analysis of multiple articles was carried out in regards to specific architectures or models proposed by other researchers. Abhinava Sadasivarao and others have proposed [1] programmable architecture that tries to integrate with the deployment of SDN within the Data Centres. It abstracts a transport node into a programmable virtual switch and leverages the OpenFlow protocol for control. It can be extended to packet-switched transport architectures including MPLS. But the the time involved in setting up the path using SDN Controller showed a high latency (between 5s to 7s). Also, this architecture has implicit mode where the SDN controller has a view of only the edge nodes.

Cheng, Xu and Zang from China mobile in their paper [2] have explained the requirements for advanced network architecture of Packet Transport Network. It has good focus on fast provisioning, end-to-end multi-domain and multi-layer network view. Although, there is no consideration given to optical transport networks.

S. K. N. Rao in his white-paper [3] has provided a survey of SDN and its use cases explaining benefits and limitations. It explains that a typical SDN architecture has the network intelligence logically centralized in software-based controllers. This enables the control logic to be designed and executed on a global network view. This was a key take away from this whitepaper. But the architecture described in it was just high level block-diagram. Moreover, it only described few options without specific recommendations, hence it was found to be closer to study-paper.

R. Vilalta et. al. in their paper [4] proposed a network control architecture for multidomain and multivendor network which is organized in layers (Refer fig.1) The abstract network layer and the control-specific layer, results in a mesh of generic SDN controllers. This uses generalized multiprotocol label switching (GMPLS) protocols as their east/west interfaces. This architecture was tested for its performance on service provisioning latency and control plane overhead. As a limitation, the authors have clarified that application of this architecture are confined and scoped to a single or reduced number of operators with peering agreements. Verizon [5] has deployed similar hierarchical architecture for moving to 100G Packet-optical transport network.

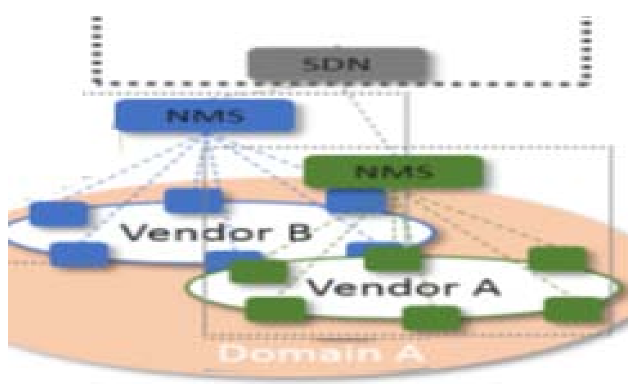


Fig.1: Hierarchical SDN Architecture for OTN

Industry body Open Networking Forum (ONF) has also proposed [6] an architecture with a parent "super" controller. The network orchestrator will abstract the details of the optical transport layer. It also enables end-to-end provisioning of services and provides open interfaces to client SDN applications. It is possible to have multiple technology controllers per domain, with this architecture.

Optical network model based architectures claim to provide open and vendor agnostic management of optical equipment. Thomas Szyrkowiec et. al. have investigated [7] Optical Network Models and their application to SDN Management. They have surveyed and compared important optical network models. They proposed an intent interface for creating virtual topologies which is integrated in the existing model ecosystem. This is an interesting architectural approach. It makes it easy to achieve software-based control but requires the network topology virtualization. In paper [10], a similar virtual modelling using VOLTHA was described. VOLTHA is emerging as a standard for virtualization of OLT (online termination). The modelling approaches in [7] and [10] has limitations, as assumptions like every cross-connection between input and output ports is possible, does not hold for a realistic optical network model. Also on a physical network level, the analogue nature leads to network constraints, which can't be modelled accurately.

A simplified architecture is proposed and analysed by RK Jha and Burhan NML in their paper [8]. In this architecture, control of ethernet elements (routers) and optical ring is performed by a single SDN controller. Moreover, the architecture includes standardized hardware for OpenFlow switches and standard interface for OpenFlow communication. Communication between ethernet switches and optical network devices is setup using OEO converter. Communication between SDN controller and optical network devices is setup using OpenFlow agents. Hence this architecture will need enhancements before applying to all-optical network. It can also be noted that the optical hardware (ROADM), based on banyan architecture switch, combines the data plane along with switching control inside it.

III. Methodology

The experimental design methodology was used to implement the design, run tests, collect the results and demonstrate outcomes. The type of experimental design chosen was Absolute Experiment. The input was given (in the form of WDM lambdas) and output was observed by changing the control bits (or configurations). The output was recorded in the form of signal graphs and compared with expected result. Controlled variables of this experiment included input signal wavelength, OADM's drop channel setting, selection of Optical cross-connect ports. Factors not controlled include signal characteristics (like attenuation, dispersion, noise) which were not influencing factors.

This experiment was carried out in a high-end Simulator called OptiSystem. (Design section 4.2 gives more information on choice of this simulator tool.) The results were captured in the form of signal graphs shown on optical spectrum analyser (OSA). Multiple runs (iterations) were executed that gave consistent output that matches with expected output. To decide number of test-runs by using same input and to decide the variation of inputs for additional test-runs, techniques similar to sampling techniques were used. E.g. sample size of 5 test-runs with same input is considered. non-probability sampling is used for choice of input lambdas as these are in the least attenuation region (1550nm) wavelength. Judgement sampling is used for deciding add and drop channels for alternate test runs. (Note: Even though the term 'sample' is used here, it actually refers to 'test-run' or simulator iteration).

All the observations collected from these test-runs forms the empirical data collection by experiment. Simple comparison method was used to analyze the test run results. First result was compared with the theoretically expected result of the design. Following results were compared with the first result to check if any deviation. The data storage was done inside the simulator tool using the file system.

IV. Design

Before proceeding to detailed design description, it is imperative to a indicate the high-level design aspects like topology and components. The type of OTN that is considered here is long-haul transport deployment e.g. between major cities. This is a very common deployment of OTN and generalizes other types (e.g. DCN = Data Centre network) also well. Bidirectional Ring topology was considered, since most of the OTN deployments follow Ring topology [18].

The key components in any ring are OADM (Optical Add-Drop multiplexer) and OXC (Optical Cross connect or Optical Switch). Some vertically integrated devices contain OADM, OXC, transponders etc in the same physical chassis. They perform forwarding (data-plane) and switching (control-plane) activities together inseparably. Hence as a design choice, such integrated devices were avoided. When the same signal was to be replicated on other fiber (or other direction), simple fork was utilized. Its possible to replace it with switch if any setup so requires. Since the experimental setup and outcomes are not influenced by physical characteristics like noise, BER, dispersion, there is no specific assumption on type of fiber or repeaters or amplifiers etc. The Ring input is in the form of WDM signal with four channels viz 193.1 THz, 193.2 THz, 193.3 THz and 193.4 THz. The drop and add channels of each OADMs are as shown in Table.1:

Table 1: OADM channel configuration

OADM	Drop Channel	Add Channel
A	193.1 THz	193.7 THz
B	193.2 THz	193.8 THz
C	193.3 THz	193.9 THz
D	193.4 THz	194.0 THz

a) Detailed Design

The key design considerations include.

- Data-plane and control-plane separation: Data plane focuses only on data-forwarding
- Controller having central view instead of one node (and neighbouring nodes): This helps in deciding network domain level functions like path traversed in the ring
- Controller issuing software instructions (configurations or scripted commands) to bring programmability, allowing change of flow dynamically.

To accomplish these key considerations, following layers are envisaged. (Refer Fig. 2) The bottom layer is Data-plane, formed by four OADMs. On top of that resides a Control plane. It consists of OXC (optical switches) which are software-controlled by the controller. Controller has a full network domain view. It issues software commands to switches in control plane and accordingly the switches control the path flown through the data-plane.

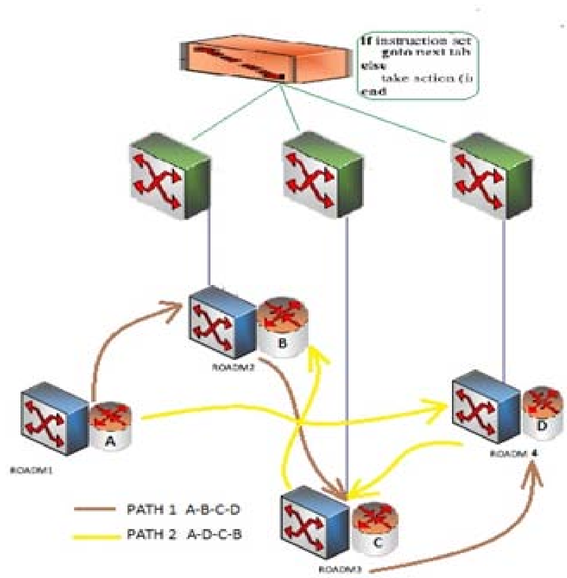


Fig. 2: Design showing control and data planes

The three OXCs are co-located i.e. placed nearby the OADMs B, C and D. But they are not hosted on or tightly couples with OADMs, like in traditional OTN. Based on instruction from controller, the OXC selects different ports and the signal coming from OADM is put onto one of the ports selected. One port allows the flow in one direction (A-B-C-D) while other port enables a reverse flow direction (A-D-C-B).

This allows to change the flow direction dynamically based on controllers input to switches in control-plane. As can be seen, OADM has no participation in flow-control. It just processes (drops a channel and adds new channel) irrespective of flow direction. This design can be used by researchers as a starting point in their SDON (Software Defined Optical Network) research, or for training others about SDN, or for integration testing in larger network setup. This design is implemented and tested using the simulator environment and output graphs are collected.

b) Simulator Selection

Simulation are technology tools and they help with the unreal and real-life entities to be modelled on to the computer and run under certain predefined conditions [11][12] Network simulators are widely used by the research community to evaluate new theories and hypotheses. [13] Selection of simulator can impact the outcomes, hence sufficient analysis is required before selection, especially in optical network research. [14] The author has analysed many simulators and has come up with detailed approach to select simulator for research in OTN. Using a simulator selection tool that the author himself created for ranking the simulators, OptiSystem simulator was ranked highest for implementing this design, testing it and collecting the results/ measurements of test-runs.

OptiSystem is specialized simulator for optical communication systems and includes all layers of Optical Transport Network [15]. It has comprehensive library of components. It offers high model accuracy, wide range of modulation formats, powerful simulation environment and hence good fit for research purpose. It has a truly hierarchical definition of components and systems including Optical sub-system. As per author's self-experience, this tool is quite easy to install and run. The GUI is intuitive and user-friendly and provides ability to quickly design optical Networks.

c) Environment Setup

Latest OptiSystem (v21.0) is used for implementing this design and setting up the test-run.

- The global parameters related to optical characteristics (such as attenuation, insertion loss, power levels) were maintained as default. (as shown in Fig. 3)
- The global parameters related to simulator model are evaluated. Some minor modifications done to bit rate, padding by no of leading/lagging zeros, central frequency (as shown in Fig. 4)
- Execution related parameters are changed as: set iterations to 2 within every run. Accordingly set the signal buffer value same as iterations (=2). Set location to store the results and filename to end with the test-run number, timestamp.

Main | Simulation | Noise | Custom order

Disp	Name	Value	Units	Mode
☑	Frequency	193.1	THz	Normal
☐	Bandwidth	10	GHz	Normal
☐	Insertion loss	0	dB	Normal
☐	Depth	100	dB	Normal

Main | Simulation | Noise | Custom order

Disp	Name	Value	Units	Mode
☐	Noise threshold	-100	dB	Normal
☐	Noise dynamic	3	dB	Normal

Figure 3: Global parameters: Optical characteristics

Simulation | Signals | Spatial effects | Noise | Signal tracing

Name	Value	Units
Simulation window	Set bit rate	
Reference bit rate	☑	
Bit rate	2.5e+009	bit/s
Time window	6.3999999999999999e-009	s
Sample rate	1.28e+012	Hz
Sequence length	16	Bits
Samples per bit	512	
Guard Bits	0	
Symbol rate	10e+009	symbols/s
Number of samples	8192	
Reference wavelength	193.1	THz

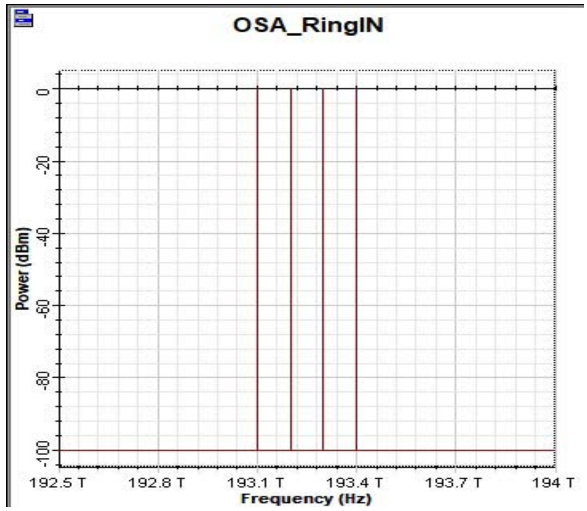
Figure 4: Global parameters: Simulation Model characteristics

V. Results

There are two iterations carried out to test run the two flows: First flow is OADM A-to-B-to-C-to-D and second flow is reverse, OADM A-to-D-to-C-to-B. Both flows worked as designed and successfully showed the designed output.

Results are collected as the graphs from Optical Spectrum Analyzer (OSA). Depending upon the test, the OSA is placed at various points to traverse the flow of signal step-by-step and check the correctness of channels for that step. The resulting graphs at OADM A, B, C and D are shown below for both the test runs. These results are stored in flat file. The file names end with test-run, iteration number, timestamp information. OADM A is a common Ring ingress point for both the flows. In both the test-runs, same input signal (WDM) is given to Ring at OADM-A as shown in fig. 5.

Flow 1 Output: Output taken at OADM-A, B, C and D shows each OADM drops a channel and adds a channel on its input signal. So, at output of D (i.e. Ring output), as expected the original four channels are dropped and new four channels are added, as seen in fig 6 (OADM A & B) and Fig 7 (C & D).



Ring Input (WDM) at OADM-A

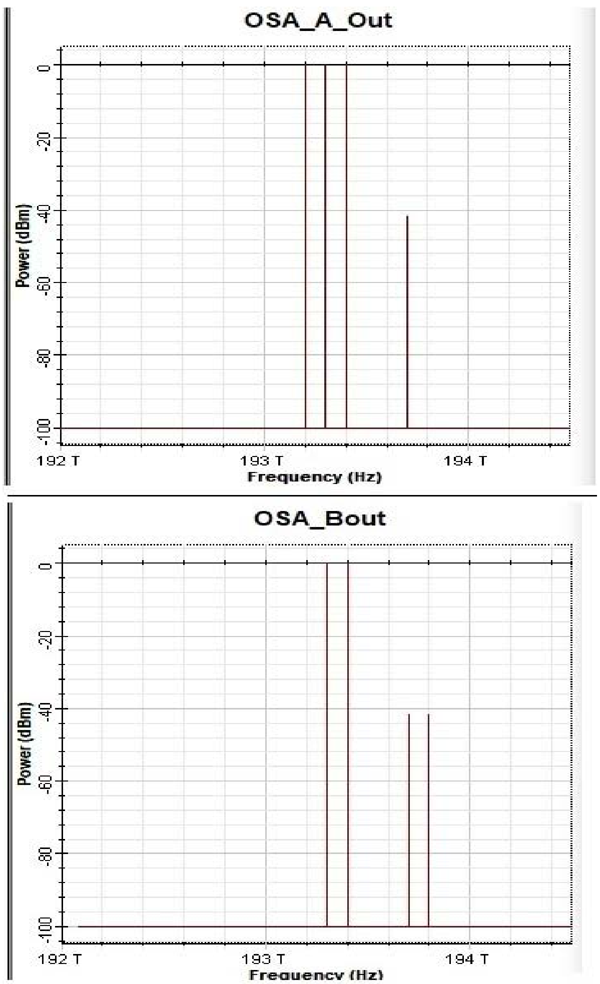


Fig. 5: Output at OADM-A & B for flow 1

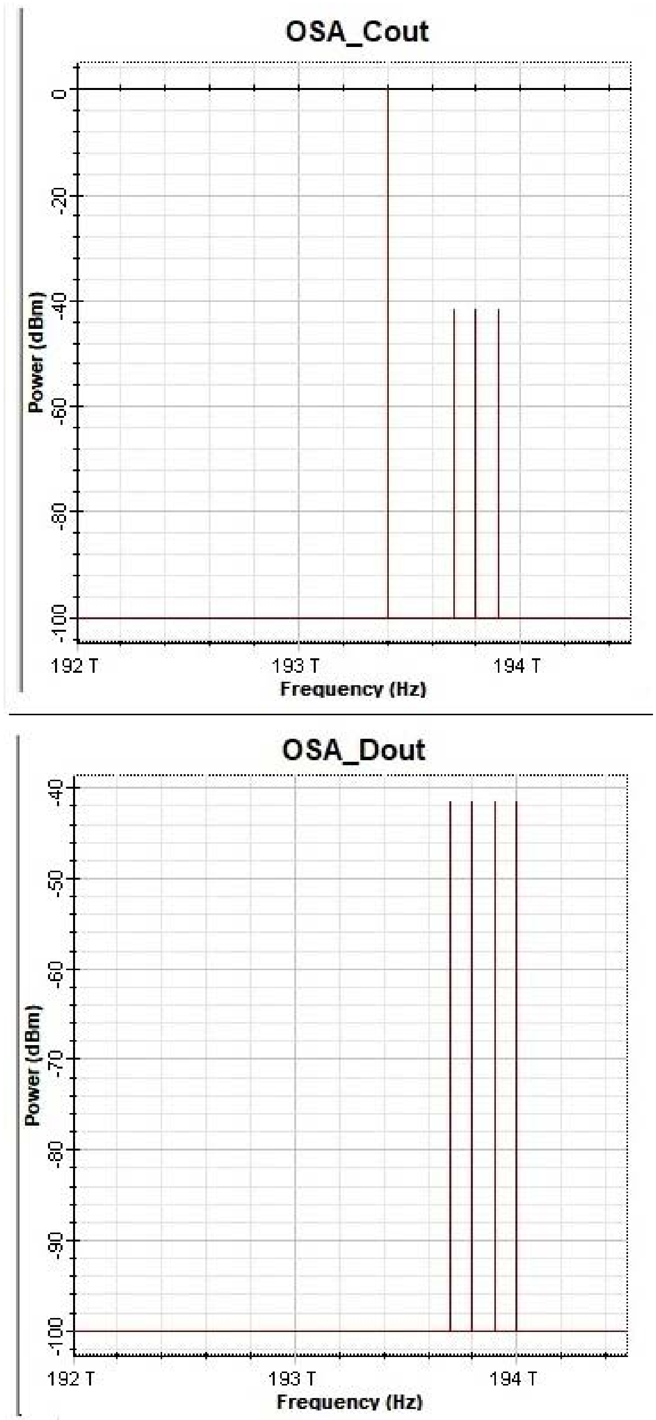
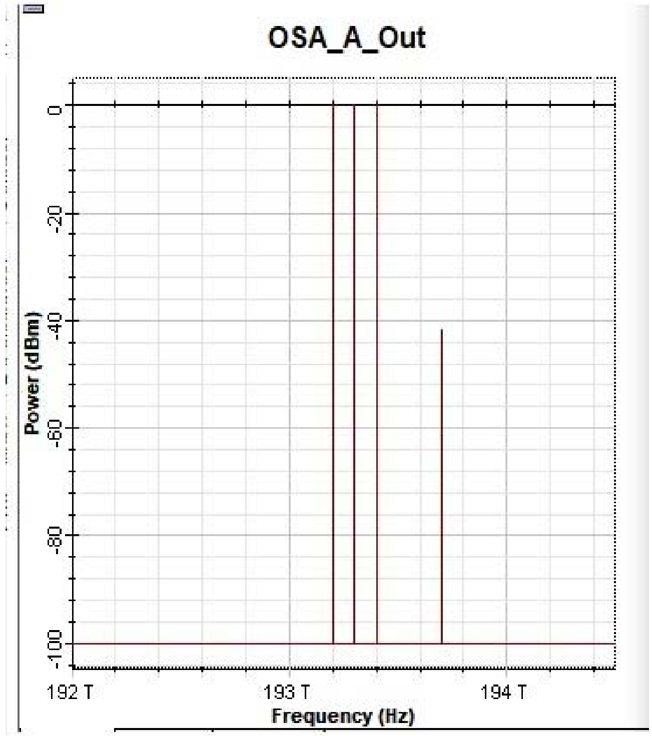
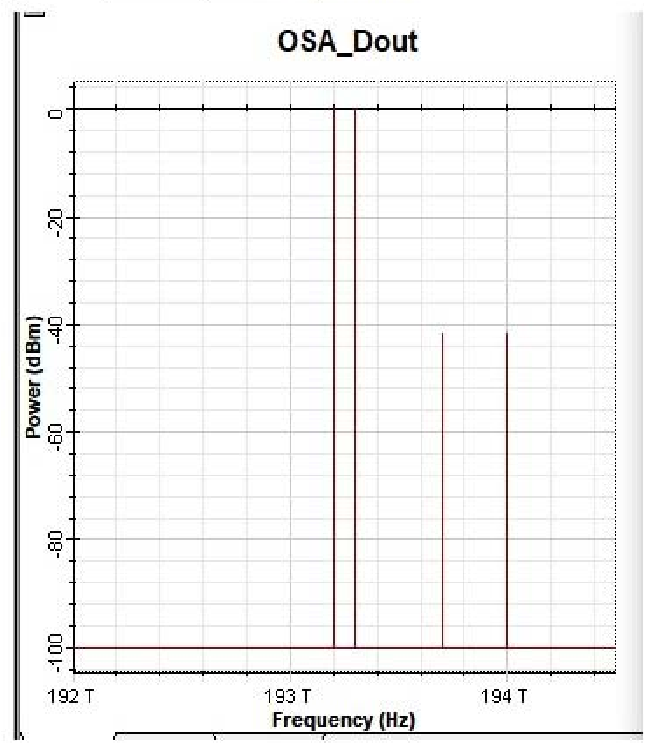


Fig. 6: Output at OADM-C & D for flow 1

In the reverse flow test-run, output of OADM-A is same (as same input is provided). Thereon, there is a difference in flow. Output at B is now Ring output due to flow reversal and B shows only new four channels added (as expected for reverse flow). Output of OADM A, D show in Fig 8 and output of OADM C & B is shown in Fig 9.





Output at OADM-A & D for flow 2

This confirms that flow-reversal design has worked as desired. The controller changes the configurations and sends instruction (bit-sequence) to switches to reverse the flow. This shows that the programmability of flow in the OTN can be achieved by control-plane.

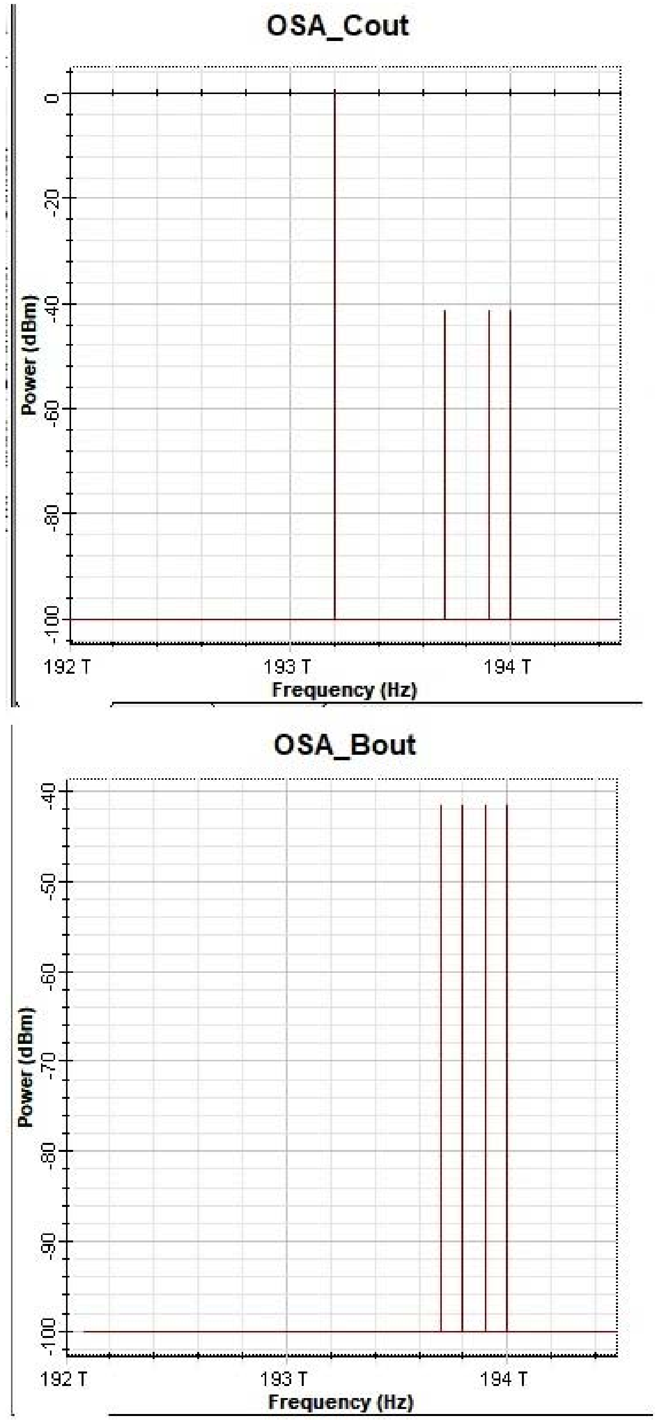


Fig. 7: Output at OADM-C & B for flow 2

VI. Conclusion

It was experimentally shown that the proposed design is capable of achieving flow-control in OTN with the help of programmability offered by its central control plane. The design when implemented works as desired and flow reversal in Ring topology was observed. This unique design follows key concepts in SDN (like control-data plane separation, software-based programmability, central view of network domain etc). Hence it can be inferred that it is possible to apply SDN concepts to OTN, although SDN can't be applied to OTN in the same way it is applied to IP networks. SDN expects optical networks to be modified to adapt to SDN e.g. by clear separation of data-plane and control-plane. This paper presents a design with optical control-plane and data-plane logically separated. Flow control is dynamically achieved by programmability offered by central control-plane that has full view of optical ring network. Thus, the paper has illustrated SDN-like programmability in OTN using flow-control scenario. Researchers can reuse this design in their SDON research work as-is or by modifying as per their needs, the design is flexible. Academician can use it for training SDON concepts or for integration testing with another network setup. This tested, working and flexible to modify network design acts as a reliable starting point. With it, researchers can make alterations with confidence and refer back to this reliable stratum. It can also be concluded that OptiSystem simulator was suitable choice for implementing and testing this OTN design. As a future work, OpenFlow can be added to this design for SBI (southbound interface)

Acknowledgment

The author wishes to thank OptiWave OptiSystem [15] for allowing temporary access to licensed copy of latest version (v21.0) of OptiSystem Simulator.

References Références Referencias

1. Abhinava Sadasivarao et. al, Bursting Data Between Data Centers: Case for Transport SDN, IEEE 21st Annual Symposium on High-Performance Inter-connects, pp87-90.
2. Weiqiang Cheng, ACTN Use-cases for Packet Transport Network, China Mobile, IETF, Jul 2014.
3. S. K. N. Rao, SDN and its Use-Cases - NV and NFV: A State-of-the-Art survey, presented at NTIL 2014 event.
4. R. Vilalta et. al, Control and Orchestration of Multidomain Optical Networks With GMPLS as Inter-SDN Controller Communication, Journal of Optical Communications and Networking Vol. 7, Issue 11, pp. B46-B54.
5. Verizon readies its metro for next-generation P-OTS, Gazettabyte, June 2014.
6. ONF-TR 538 : Use Cases for Carrier Grade SDN
7. Thomas Szyrkowiec, Achim Autenrieth and Wolfgang Kellerer,Optical Network Models and Their Application to Software-Defined Network Management, International Journal of Optics, Volume 2017.
8. Rakesh Kumar Jha and Burhan Num Mina Llah, SDON: proposed architecture and comparative analysis, Journal of the European Optical Society-Rapid Publications (2019).
9. Cisco Global Cloud Index: Forecast and Methodology, 2015–2020, White Paper.
10. Calient White Paper, "The Software Defined Hybrid Packet Optical Datacenter Network"
11. J. Heidemann, K. Mills, S. Kumar, Expanding Confidence in Network Simulation, IEEE Net-work Magazine, Vol. 15, No. 5, Apr. 2000, pp. 58- 63.
12. Siraj, Saba, A. Gupta, and Rinku Badgujar. Network simulation tools survey, International Journal of Advanced Research in Computer and Communication Engineering Vol. 1, Issue 4, pp. 199-206, 2012.
13. Atta ur Rehman Khana, Sardar M. Bilalb, Mazliza Othmana, A Performance Comparison of Network Simulators for Wireless Networks, IEEE International Conference on Control System, Computing and Engineering, Malaysia, 2012, pp. 34-38.
14. Vedran Mileći, Branko Mikac, Matija Dzanko, Modelling optical network components: A network simulator-based approach, ResearchGate Conference Paper, October 2012.
15. Optisystem product website https://optiwave.com/products/optisystem.
16. Yao Li and D. Kilper, Optical Physical Layer SDN, Journal of Optical Communication Networks, Vol-10, pp110-121.
17. Shu Namiki et. al, Dynamic Optical Path Network, The 12th International Conference on Optical Internet Proceedings.
18. Raimena Viesllari et. al., Scalability Analysis of SDN-controlled Optical Ring MAN with Hybrid Traffic, IEEE ICC 2014 - Optical Networks and Systems, pp3283-3288.