

RESEARCH

Open Access

Software Defined Optical Networks (SDON): proposed architecture and comparative analysis



Rakesh Kumar Jha*  and Burhan Num Mina Llah

Abstract

Background: Interface SDN network with Optical network to enhance the capability of current optical network capability to achieve high data rate support to fulfilled 5G and Internet of things (IoT) backbone capacity requirement.

Methods: We use the proposed SDON design architecture based on Software defined networking and do the performance analysis so that proposed work can provide the backbone for current 5G and IoT network to support high data rate support.

Results and conclusion: A comparative analysis of results obtained in two different designs one consisting of the DWDM system and the other consists of programmable optical switches (ROADM) merged in the packet switched domain of SDN. This paper proves how SDN leads to significant improvement in the performance of the existing optical or any other Ethernet network. The flexibility of the SDON network is send the packets to the particular destination address is transmitted while the other value is either dropped or sent to the SDN controller for rerouting through a new path where flow entry is fed to the switch later.

Keywords: SDN, DWDM systems, OptiSystem, MATLAB, OpenFlow switches, BER, Q-factor

Background

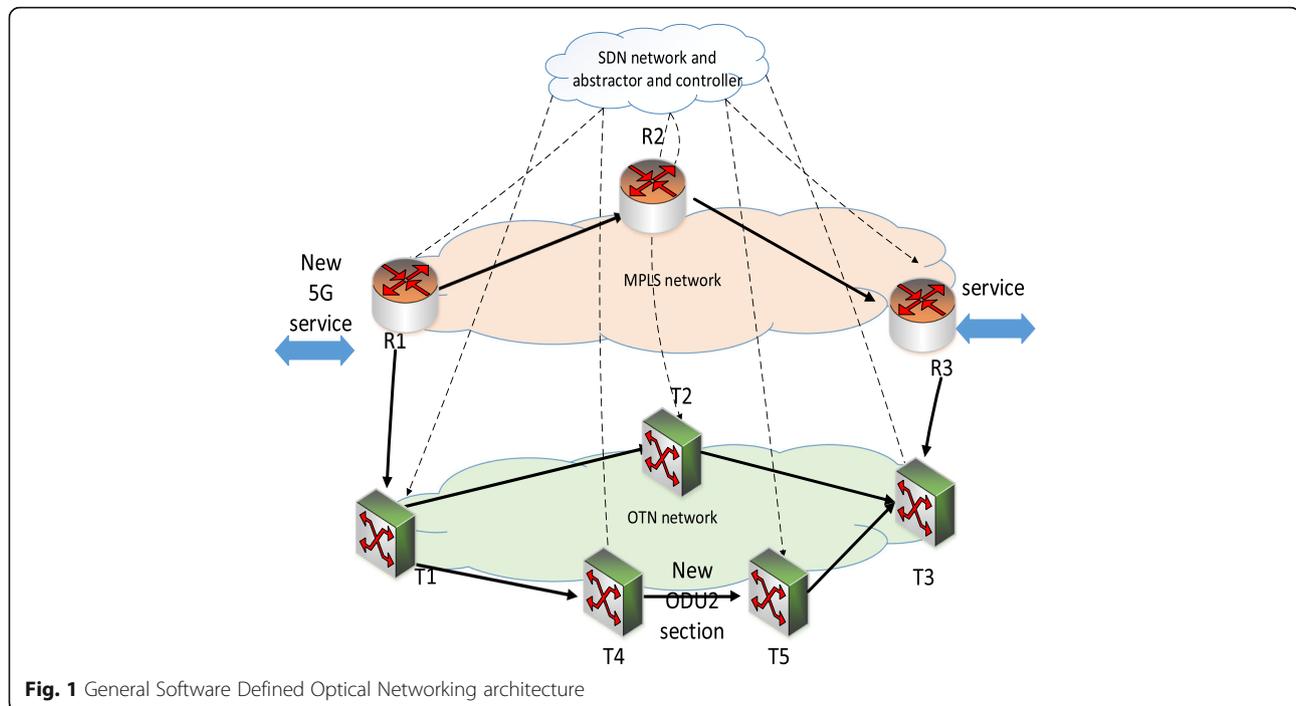
NETWORK service providers face a common problem regarding management of current networks that are made up of routers, switches and many middle boxes with complex protocols [1, 2]. Since the current networking, trends comprise of some demanding nodes like data centric nodes. These nodes require high speed and such demands are met when we merge the benefits of circuit switched networks with those of the packet switched networks. Since the existing networks demand high speed and in addition to it they also require error free delivery of packets. Optical networks provide high speed or we might say circuit switched networks because nothing can overpower the speed of light, while as error free delivery of packets is ensured by the packet switching. But merely switching to packet switching won't make much difference, we also want our networks to be flexible, programmable, and adaptable because traffic patterns

from these data centric nodes in the different networks are dynamic and hence we want our networks to adapt to such traffic demands. Thus, the current computing trends demand an evolution in the existing network and the network providers don't want to change the network as a whole. We want the nodes to reroute the packet through a path that is idle when its conventional path would be congested, in order to have such benefits in our network we need our packet switched nodes to be controlled by an SDN controller. SDN is a paradigm that separates the control plane from the forwarding plane [3, 4] the SDN controller carries out leaving behind the job of forwarding only for the switches and the control part.

Thus if we combine the error free recovery of packets of packet switching with high speed of optical network or circuit switching and finally the programmability and flexibility of networks of SDN we can lead to a revolution in the field of networking as shown in Fig. 1 and this could be a huge step towards making our networks flexible, cost efficient and innovative. Since SDN has a global view of the network as a whole thus it can lead to better optimization of resources [5, 6]. In fact, since in

* Correspondence: jharakesh.45@gmail.com

School of Electronics and Communication Engineering, Shri Mata Vaishno Devi University, Katra 182320, India



SDN the control is centralized which has a better control compared to the distributed control in other existing networks. Thus, combining SDN with the optical networks would revolutionize the networking world by making them flexible and adaptable to the demanding nodes of current scenario. It also gives a provision to the users itself to be independent of network operators as they themselves can be innovative in designing their own network as they won't have to change the infrastructure but only re-program the network.

Contributions

In the field of Software Defined Optical Networking (SDON), a testbed has been simulated in OptiSystem along with MATLAB co-simulation with OpenFlow switches represented by MATLAB components. The main aim was to design a system to incorporate both SDN and Optical networks so that both high speed and capacity ensured by Optical networks is accompanied by error free delivery of packets taken care by SDN (packet switched part) that makes the system programmable and hence adaptable to current computing trends. Therefore, a simulated design of SDON is shown in this paper accompanied by the comparative analysis of results obtained from SDON with that a simple optical network, in this case DWDM system. The results obtained clearly show how SDN improves the flexibility of system and hence lowers BER to as low as zero because we get the exact replica of input packets at the receiver side.

In this paper, the following section II contains the brief description of basics of SDN. It also presents different flow charts that show the operation of OpenFlow switches. In section, III a network design is proposed that has employed all the above-mentioned three domains and this model is implemented in the OptiSystem tool. The algorithms accompany the general architecture mentioned and flow chart the whole network employs during its operation. Section IV mentions various designs of optical networks and SDON in the simulation tool optisystem. The following section V mentions the respective results. In this section a comparative analysis is carried out between a simple optical network that comprises of the DWDM system and the network that has SDN controlling the packet switching part and optical part of the network. A comparison of BER and Quality factor is shown in this section and it is seen how SDN decreases the BER and how speed is increased by including SDN in optical networks. A detailed study of effect of SDN on the optical part of the network is analyzed based on input from the OpenFlow switch. Since in OptiSystem we can only design the optical part, the packet switched part is designed in the MATLAB and then co-simulated with the OptiSystem tool.

Methods

Software-defined networking (SDN) has gained a lot of attention in recent years, because it addresses the lack of programmability in existing networking architectures and enables easier and faster network innovation [7–9]. The

conventional communication systems comprises of switches, routers, and middle boxes. These networks have the capability to forward the packets and in addition to it they have the controlling ability embedded in them. Software Defined Networks (SDN) are different from these conventional networks in a way that SDN is the paradigm for separating the forwarding plane of networking from the control plane. This advancement in the field of networking has helped to make our networks flexible, adaptable to current computing trends. SDN has the centralized control plane that has a broader view or we can say a global view over the whole networks and it is well aware of the busy and free nodes or paths in the network. Having this global view of the existing network and having the ability to update the routing tables of the switches the controller of Software Define Network can make the packets to take the path that might be idle since a long time. Hence, it can lead to better optimization of resources.

We have a three-layer software defined architecture shown in Fig. 2 in which the three respective layers are application layer, control layer, and infrastructure layer. In addition to these three layers, it also consists of two types of interfaces [7–10] i.e. *southbound interface and northbound interface*. The southbound interface is between the infrastructure layer and the control layer whereas the northbound interface exists between the application layer and the control layer. The underlying protocol responsible for communication between the control layer or control plane and the infrastructure layer or data plane is known as OpenFlow.

Software Defined Networking being based on the protocol known as OpenFlow. In fact, the switches used in the SDN are also different from the conventional switches and are termed as OpenFlow switches. We may discuss the underlying protocol and its respective switches in the following subsections.

OpenFlow protocol

OpenFlow protocol is just an option among various programmable protocols for controlling purpose in SDN [11–14], but it has proved to be the dominant one. As OpenFlow is currently in its evolving stage and many versions of its specification exist like OF 1, OF 1.1, OF 1.2, and OF 1.3; with newer releases proving to be more powerful as compared to previous versions, as their features set support IPv6, Multiprotocol Label Switching (MPLS), rerouting, metering, policing and more scalable control. The OpenFlow protocol architecture consists of three basic concepts. (1) The network is made up of OpenFlow switches that compose the forwarding plane or the data plane; (2) the control plane consists of one or more OpenFlow controllers such as NOX, POX etc.; and (3) a secure control channel, which connects the OpenFlow switches with the controller or control plane.

OpenFlow switches

An OpenFlow switch is a basic forwarding device that forwards the packets according to its flow table [15–19]. This table in the switch holds a set of flow table entries,

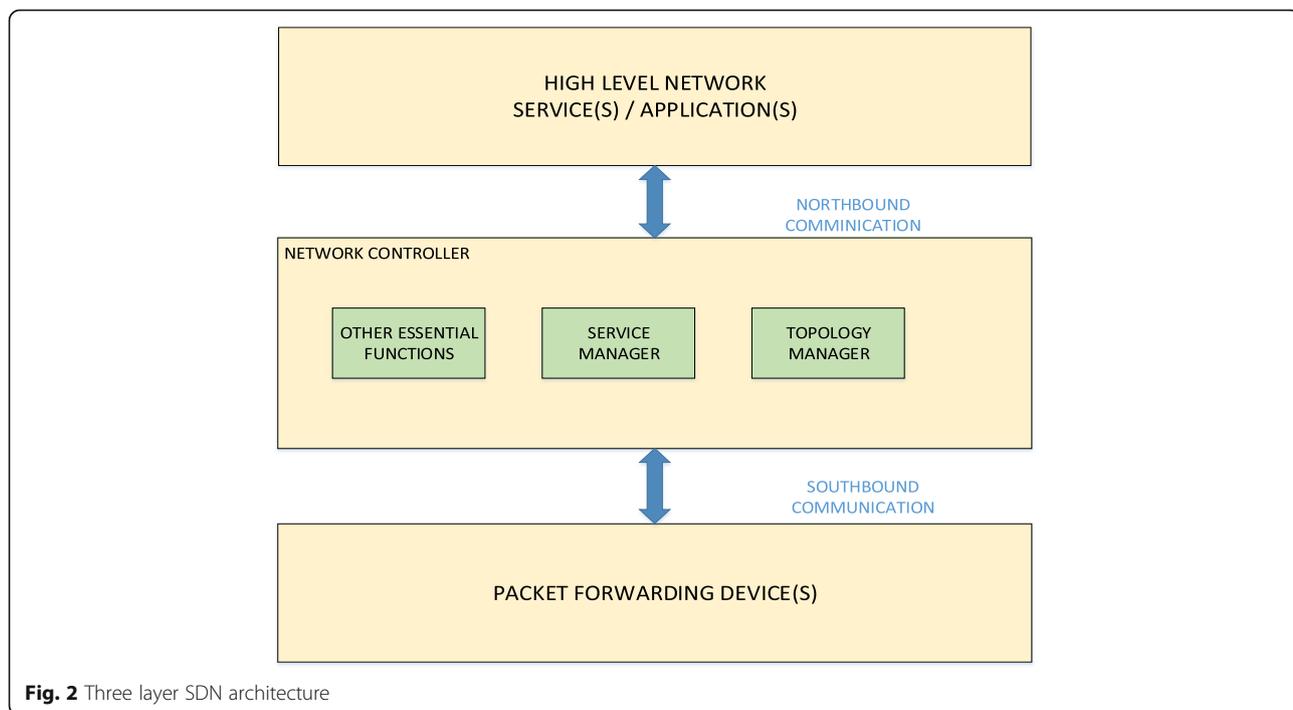


Fig. 2 Three layer SDN architecture

each of which consists of match fields, counters and instructions.

The incoming packets search for the entry in the flow table against which the header of the packet is matched and if the flow entry exists, the switch forwards the packet according to the action given in the flow entry as is shown in the flow table below in Fig. 3.

If the flow entry of the incoming packet is not present that is if the packet header is not matched with the matching field then it searches for the table miss entry. It is explained well in the below shown Fig. 4, that if the table miss entry is present then either the packet is forwarded to next flow table or it is passed to controller or in fact it drops the packet if all other flow tables are checked for.

After checking for the table miss flow entry the three decisions take on the basis of action part (instructions) of the flow entry leads to further forwarding of packets. Now, if the packet is forwarded controller it has to set up a new path and hence the flow table needs to be updated. Thus when the flow table is updated that is when a new flow entry is introduced in the table the counter is set which calculates the time for which the flow entry exists in the table and after those seconds the flow entry is automatically discarded.

SDN and optical networks

Optical networks are composed of any switch with optical interfaces in order to perform circuit switching. It is strictly confined to the optical layer: e.g. wavelength

division multiplexing (WDM) equipment, further confining to DWDM for high speed operations, reconfigurable optical add drop multiplexer (ROADM) acting as optical switches and the supporting optical link between them.

Optical networks as such are circuit switched networks and enable faster delivery of packets to destination port. Thus in order to meet the current trends and applications we need to make our networks fast as well as a flexible and adaptable to changing demands of users. This demand would be fulfilled if we will merge the benefits of SDN and optical networks by making the existing optical networks programmable and thus flexible enough to meet changing needs of users. Many ways have been proposed, employing which we can design an efficient optical network, which can be controlled by an SDN controller to make it flexible and programmable. Some efficient optical networks that can be implemented to include SDN are: (1) Generic packet/TDM/fibre switching, (2) CDC ROADM based network, (3) Sub-wavelength switched network (broadcast network with tunable receiver), (4) Sub-wavelength switched network (tunable transmitter based optical burst switch).

Proposed architecture of SDON

The architecture of an SDON [7, 8, 20–23] is depicted by Fig. 5. It consists of two clients A and B between which the communication is to be held. These clients are connected to the openflow switches, which in turn are

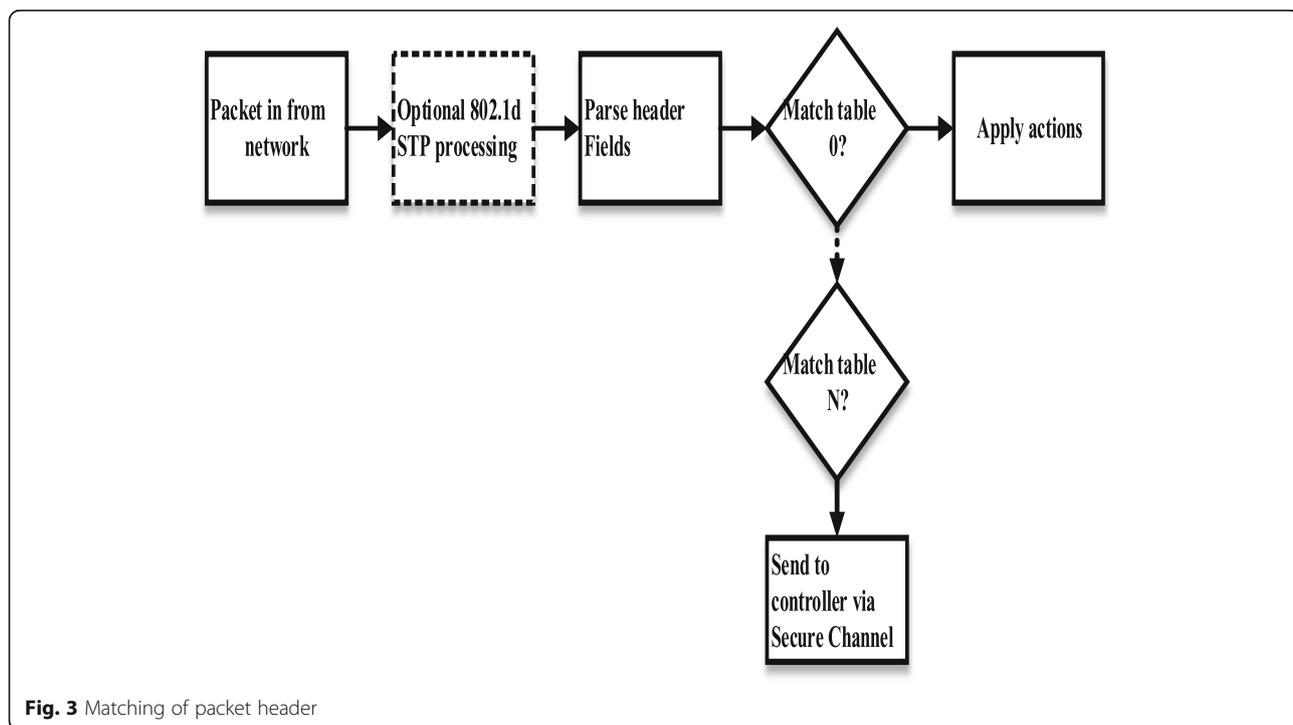
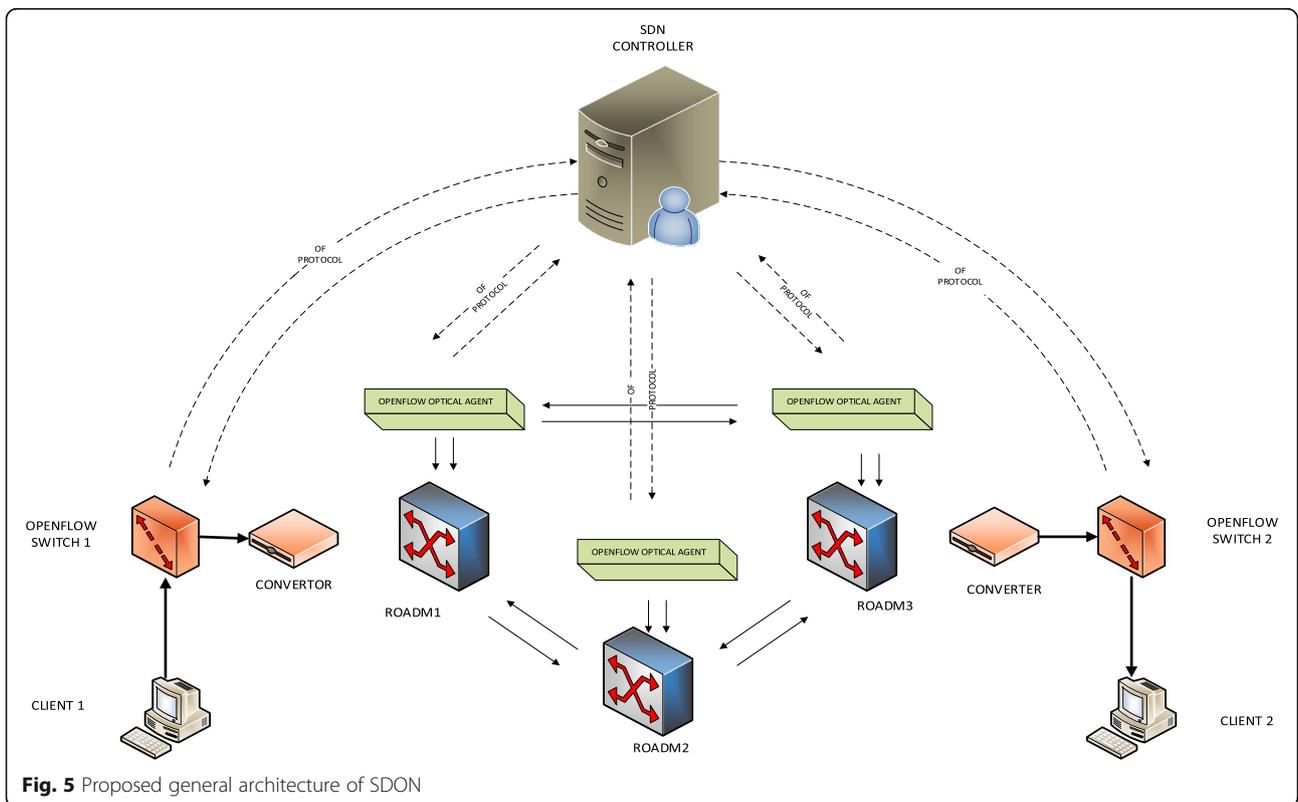
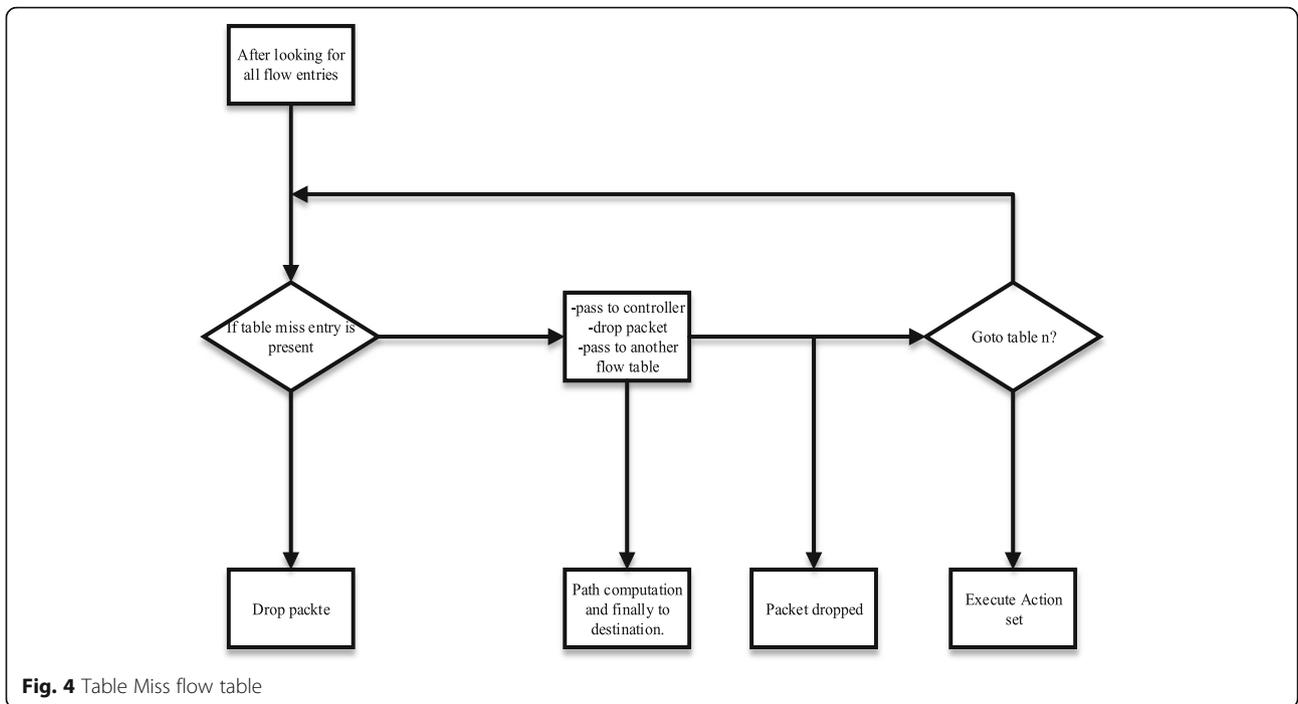


Fig. 3 Matching of packet header



controlled by the SDN controller, and by itself it just performs the forwarding part of communication only.

The optical part of the architecture as can be seen in the figure consists of ROADM switches but we can also use COADM switches, in case we have lesser number of wavelengths to dropped or added. The communication between the openflow switches and the ROADM switches is through an electrical to optical converter. Since both the optical as well packet switches are controlled by the SDN controller, for packet switches we don't require an interface but for optical switches we do require an interface to translate the electrical messages from the SDN controller to the optical signals to control the optical switches (ROADM). It's been called as the openflow optical agent which can also be programmed in Matlab as we will program SDN controller in Matlab and then interface with rest of the optical part. The interaction between the controller and the agent or openflow switch is based on the openflow protocol. We can simulate this architecture of SDON in different tools. But the tool I have worked on to implement it is OptiSystem and different Matlab interfaced components like openflow optical agent and SDN controller managing the control part of the communication network.

Algorithms and flowcharts

Algorithms based on which the incoming packets are forwarded in the above mentioned general architecture of the Software Defined Optical Network (SDON) are mentioned below in a step wise manner starting with matching the header field of the packet and finally moving towards the dropping of packets that is decided by the OpenFlow switch based on the fed instructions.

Algorithm 1 Packet Matching

```

x=counter count of packets arrived;
n=number of flow tables;
goto table n=1;
  if Packet Match Field=OpenFlow Switch Match Field
    x=x-1;
    output=action set;
    match field updated;
    if instruction set=goto next table;
      metadata=header+packet;
      goto next table;
      send metadata to next table;
    else
      take action (instruction set);
    end
  else
    goto next flow entry
  end
end

```

Algorithm 2 Table Miss

```

x=counter count of packets arrived;
n=number of flow tables;
  if Packet Match Field=Table Miss Flow Entry
    x=x-1;
    output=action set;
    match field updated;
    if instruction set=goto next table;
      metadata=header+packet;
      goto next table;
      send metadata to next table;
    else
      take action (instruction set);
      goto next flow entry;
    end
  else
    drop packet;
  end
end

```

Algorithm 3 Packet Send to Controller

```

Packet send to controller;
PCE computes path;
Flow entry fed back to switch;
Hardtime counter=time of arrival of entry into switch;
  if hard timeout≠0
    counter=couner-1;
    return if;
  else
    remove flow entry;
  end
idletime counter=time of flow entry execution;
if idle timeout≠0
  counter=counter-1;
  return if;
else
  remove flow entry;
end
end

```

Algorithm 4 Group Table

```

Various actions taken according to flow entries;
n=counter count;
  if action set of packet l=action set of packet n
    n=n-1;
    action set=group action;
    take group action;
  else
    action set=action of packet n;
    take action;
  end
end

```

A complete flowchart of the passage and steps that a packet follows while traversing from the client A to client B. The flowchart in Fig. 6 gives a complete scenario of how the packet moves through different stages in the packet switched path followed by the circuit switched path and again ending on the packet switched path on the destination side. It gives a clear picture as we can see that first the packet that arrives from the source client enters the OpenFlow switch and in there the header field of the incoming packet is matched against the match field of the flow table entry. If the match field of flow entry matches the header field of the packet then the packet is forwarded in accordance to the action mention in the flow entry itself. If the match is not performed then the search is carried on in finding the table miss entry. If the table miss entry is found then either of the three conditions is followed [4] i.e., (1) packet dropped (2) packet sent to controller (2) Action performed in accordance to the instructions mentioned. Along with this matching and table miss entry search we also need keep the track of counters of how long the flow entry should persist in the flow table. This is performed in a similar way as was done in the basic SDN itself. In addition to it, after the packets have been matched or not we need to take care of forming a new flow entry if the packet is sent to controller. The new path is set after the controller has a global view of the network which is ensured by the SDN controller as it is centrally located. The new path is based on the knowledge that the controller has of the network as a whole. The controller keeps

track of resources that are idle and the routes that are not in use yet and hence makes a new path and hence a new flow entry that is later updated in the flow table and hence implemented for messages that have the same matching field.

Simulation models

DWDM layout

A simple DWDM network is designed to ensure high capacity and high speed due to optical signals involved. The layout of a simple 4-channel or 16-channel DWDM network consists of the following components shown in the figure below. The only difference of DWDM network from the basic WDM network is to ensure dense wavelengths and decrease the frequency spacing. As shown in Fig. 7 the layout of DWDM network consists of a WDM transmitter followed by WDM multiplexer which feeds the signal to a loop control and the it is forwarded to Single Mode Fiber (SMF) which is amplified by an EDFA and then the dispersion needs to be compensated and hence it is passed through Dispersion Compensation Fiber (DCF). The signal after passing through a second EDFA is fed again to the loop control to complete the loop count for which the design is set. Thus after completing the respective loops of loop control, signal is demultiplexed and received by an optical receiver.

ROADM layout

The above-mentioned general architecture of SDON has been simulated in the latest version of a very advanced tool called OptiSystem. As is shown in the Fig. 7 below, in

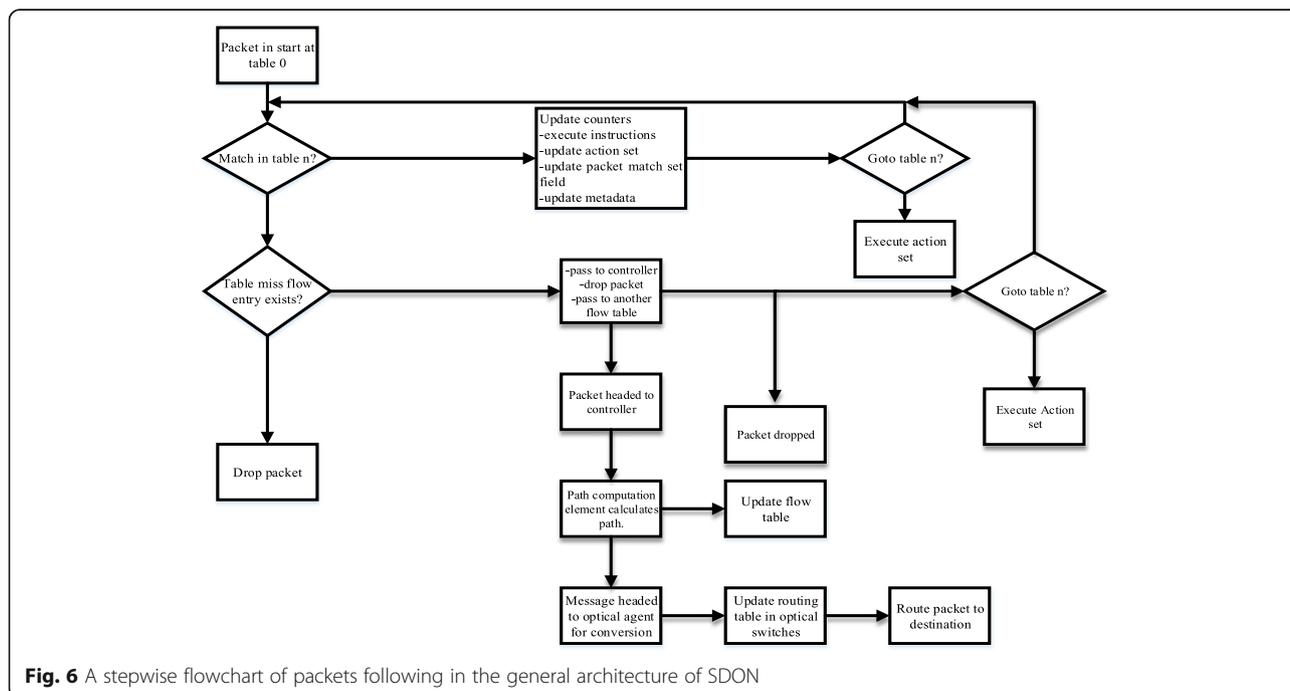
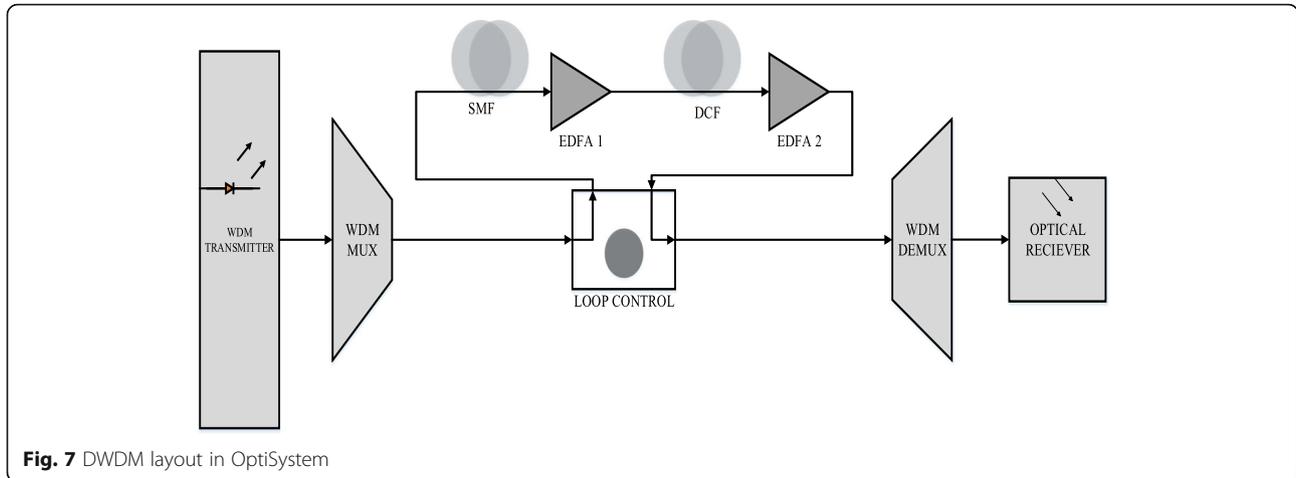


Fig. 6 A stepwise flowchart of packets following in the general architecture of SDON



or der to design a simple ROADM switch [7–10, 18–23] the most important consideration is to design banyan architecture switch which consists of optical switches in the crisscross manner as depicted in the figure itself.

This part of the ROADM is necessary and important as this ensures the wavelength selective switching (WSS) which is the underlying principle of ROADM. The source transmits an array of signals which are passed through the WDM multiplexer which on moving further is passed through the amplifier specifically SOA. After passing the signals through a common amplifier the signals are split and individually passed through a Bessel filter followed by an optical delay in each path. Then these waves are fed to this banyan architecture switch which performs the wavelength selective switching. And finally at the output we select one of the waves transmitted and hence it acts as a ROADM switch. On simulating this layout it was seen that how we get a different wavelength at the output when we change the respective bits in the control part of the optical switches.

The expression that allows the selection of the output wavelength from among four input wavelengths in this case, according to the physical parameters and structure of the device, is:

$$\lambda \sim \frac{x}{f} \frac{1}{\left(\frac{n}{ND} + \frac{2}{\left(\frac{M}{2} \right) d} \right)}$$

where x is the distance from the optical axis to the output fibre, f is the focal distance of the lens, d is the spatial period of the fixed diffraction grating and M has into account the number of phases.

Once the Spatial Light Modulator has been chosen, the focal distance of the lens, f , to illuminate with the collimated light the complete active surface $ND \times ND$ of

the SLM, is related with the number of pixels N and its size D by the expression:

$$f = \pi \phi_{core} \frac{ND}{4\lambda_o}$$

where ϕ_{core} is the input fibre core diameter and λ_o the central wavelength in the operation region. The 3 dB passband filter bandwidth of the holographic device, BW , is:

$$BW \geq \phi_{core} \frac{\left(\frac{M}{2} \right) d}{f} \left(1 - \frac{\lambda_o^2}{\left[\left(\frac{M}{2} \right) d \right]^2} \right)^{3/2}$$

if the condition $8D \gg D$ is reached, where d is the fixed grating spatial period.

In case of a Bessel filter, the BW (filter -3 dB bandwidth) and out-band attenuation are reached by the increment of the filter order or the variation of the polynomial coefficients. For a Gauss filter these values are obtained, directly, by the change of the standard deviation = BW :

$$Hf_{Gauss} = \frac{1}{\sqrt{2\pi BW}} e^{-\frac{(\lambda-\lambda_o)^2}{2BW^2}}$$

with $\lambda_o - 2BW < \lambda < \lambda_o + 2BW$, and

$$Hf_{Bessel} = \frac{1}{(\lambda-\lambda_o)^2 + 3(\lambda-\lambda_o) + 3}$$

SDON layout

In order to implement the above-discussed proposed general architecture of SDON we have used ROADM switches to consist as the optical switches, which can further be programmed by the SDN controller in order to get hold of the flexibility of the network as whole.

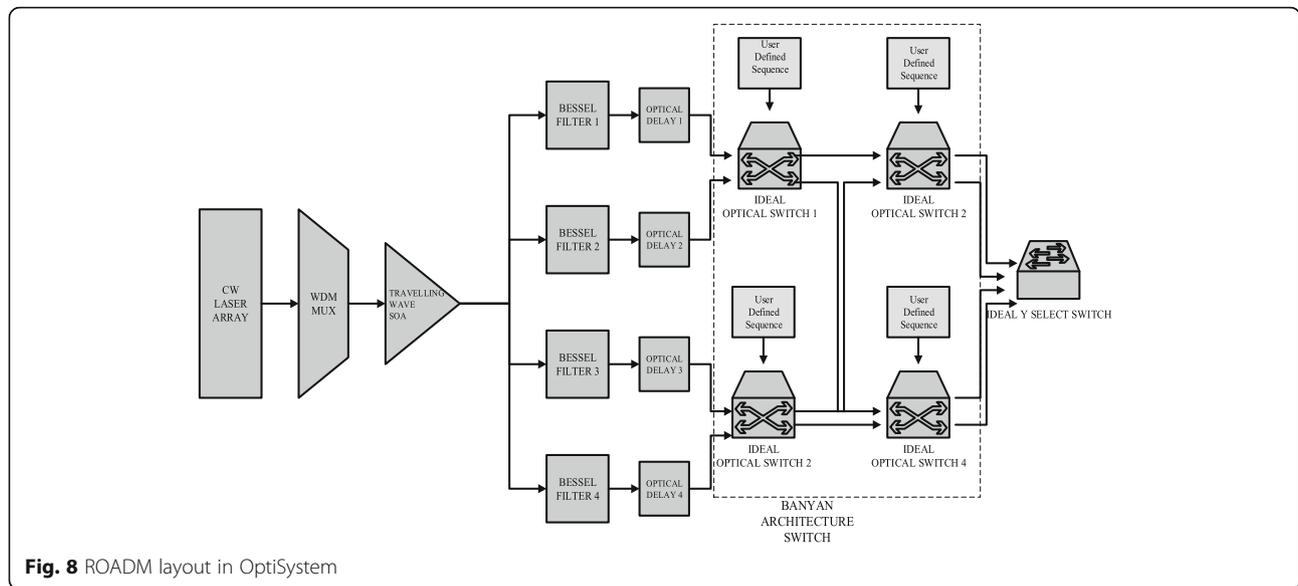


Fig. 8 ROADM layout in OptiSystem

Since the packet SDN controller can control switched part easily by importing the new route into the switch but when this SDN is merged with the optical domain we need only those optical switches that can be programmed and hence this purpose is served by ROADM very well. As can be seen in the layout shown in Fig. 8 below that for designing the packet switched part of SDON we needed to design the OpenFlow switch in MATLAB which consists of generating the packets and then followed by the flow table against which the packets are matched and hence further traversed.

The SDN controller in the layout above consists of the MATLAB component, which generates different sequences of bits that acts as the control part for ideal optical switches in the banyan architecture switch of ROADM. These bit sequences if kept in a separate subsystem would constitute a controller of ROADM as a whole. If this SDN controller gets an input packet from the OpenFlow switch then the controller, which would be a MATLAB component in OptiSystem, would take a global view of the network and would look for the possible routes this incoming packet could take and hence would form a new path for this particular packet and this path would be updated in a new flow entry which would be later updated in the existing flow table present in the OpenFlow switch. Hence, when any other packet with a similar header field arrives the input of OpenFlow switch would be routed according to this new flow entry fed from the controller.

Simulation parameters

The following table gives precise value of parameters used in the above-mentioned layouts of OptiSystem Table 1. Table 2 gives the distinct values used in

Table 1 Optical Part

Parameters	Value
Layout:	
Sensitivity	- 100 dBm
Resolution	0.1 nm
Interpolation Offset	0.5 nm
Bit rate	10 e ⁺⁰⁰⁹ bits/sec
Sample Rate	80 e ⁺⁰⁰⁹ Hz
Sequence Length	1024 bits
Samples/bit	8
Symbol Rate	10 e ⁺⁰⁰⁹ symbols/s
No. of Samples	8192
DM Laser:	
Frequency	193.1 THz
Power	10 dBm
Threshold Current	1 mA
Slope Efficiency	0.4 W/A
Linewidth	10 MHz
PIN Diode:	
Responsivity	1 A/W
Modulation Bandwidth	2 GHz
Junction Capacitance	3 pF
Centre Frequency	193.1 THz
Dark Current	10 nA
Thermal Power density	100 e ⁻⁰²⁴
Absolute Temperature	298 K

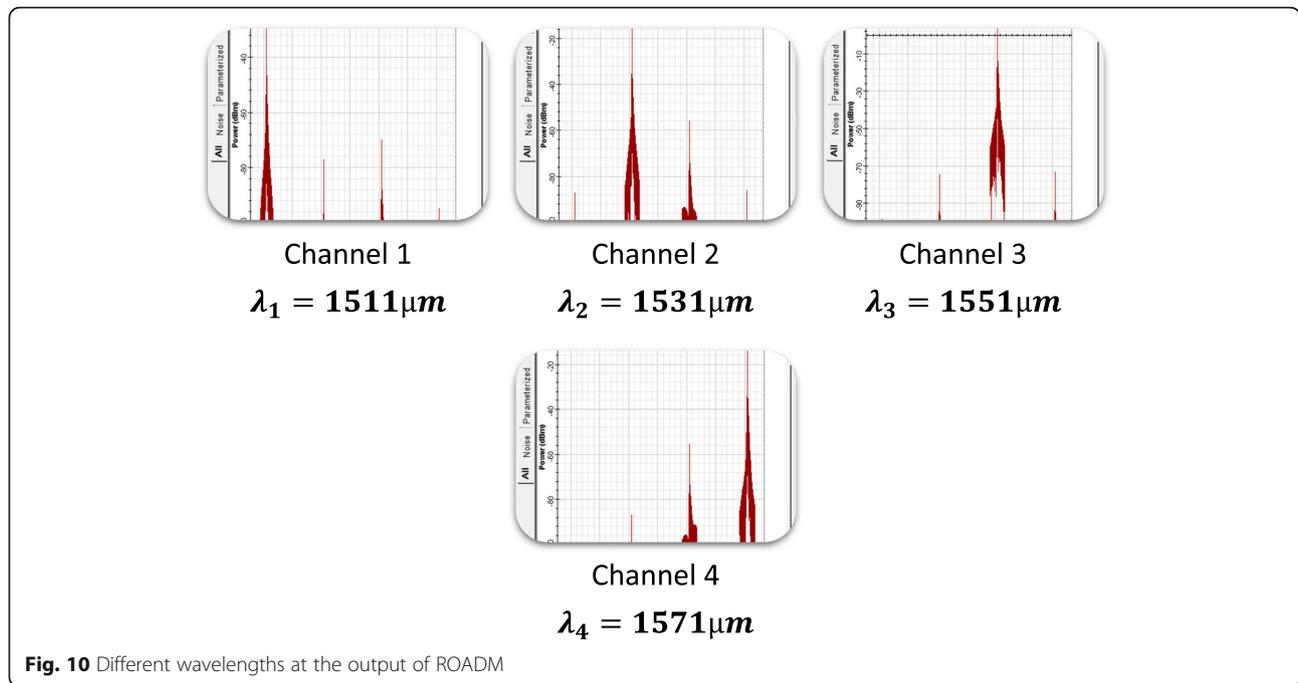


Fig. 10 Different wavelengths at the output of ROADM

Different wavelengths at the ROADM output (Figs. 10 and 11)

When the above-mentioned layout of ROADM shown in Fig. 7 is simulated in OptiSystem we get four different wavelengths at the output when we change the bit sequence fed from a user defined sequence which acts as the controller for an ideal optical switch. Table 3 shows the combination of different bit sequences that result in different wavelength at the output of the ROADM. The bit sequences or in this case the user defined sequences controlling the ideal switches can be replaced by a MATLAB component which would give the respective sequence of bits to the switches to transmit a

particular wavelength and drop the other. As we can see in the table below that, we just need to give a particular sequence of bits to the banyan architecture switch and not the y-select switch that selects any of the wavelengths.

Comparative analysis of received signal between DWDM and SDON

A comparison of received signal in two cases, i.e. (1) DWDM network and (2) SDON network, is carried out and the results are seen to vary drastically. When the simple optical network in this case DWDM network is compared to a network in which SDN is included, we see that BER is lowered to as low as zero as we get the exact replica of the packets sent from the transmitter end. However, in practical optical networks we see that dispersion and other attenuating factors play a vital role and may lead to intersymbol interference and hence would lead to lower but still some amount of BER, which is seen in the Fig. 12 below.

The Q-factor as seen in the figure below in case of DWDM is 5.8 but for SDON we can see that the increase of Q-factor to 1.5exp (047) is quite huge and this is ensured by the merging of packet switching in the optical infrastructure. While as decrease in BER of 2.9exp (-009) in case of DWDM to 0(zero) in case of SDON is also quite significant and hence leads to the requirement of SDON in place of current optical networks as it would serve various needs of the users.

For this reason an experiment was carried out as to see how the network reacts if a packet is sent whose

Table 3 Control bit sequence for banyan architecture switch

$\lambda_1=1511\mu m$						$\lambda_2=1531\mu m$							
Wavelength λ_1 selected:						Wavelength λ_1 selected:							
Bit Sequence	A	B	C	D	E	F	Bit Sequence	A	B	C	D	E	F
Ideal Switch 1	0	1	0	0	0	0	Ideal Switch 1	1	0	1	0	1	0
Ideal Switch 2	0	1	1	0	0	0	Ideal Switch 2	0	1	0	1	0	1
Ideal Switch 3	1	1	0	1	1	0	Ideal Switch 3	1	0	1	0	1	0
Ideal Switch 4	1	1	1	1	1	0	Ideal Switch 4	0	1	0	1	0	1
$\lambda_3=1551\mu m$						$\lambda_4=1571\mu m$							
Wavelength λ_3 selected:						Wavelength λ_4 selected:							
Bit Sequence	A	B	C	D	E	F	Bit Sequence	A	B	C	D	E	F
Ideal Switch 1	1	1	1	1	1	1	Ideal Switch 1	1	0	1	1	1	1
Ideal Switch 2	0	0	0	1	1	1	Ideal Switch 2	1	0	0	1	1	1
Ideal Switch 3	0	0	0	0	0	0	Ideal Switch 3	0	0	1	0	0	1
Ideal Switch 4	1	1	1	0	0	0	Ideal Switch 4	0	0	0	0	0	1

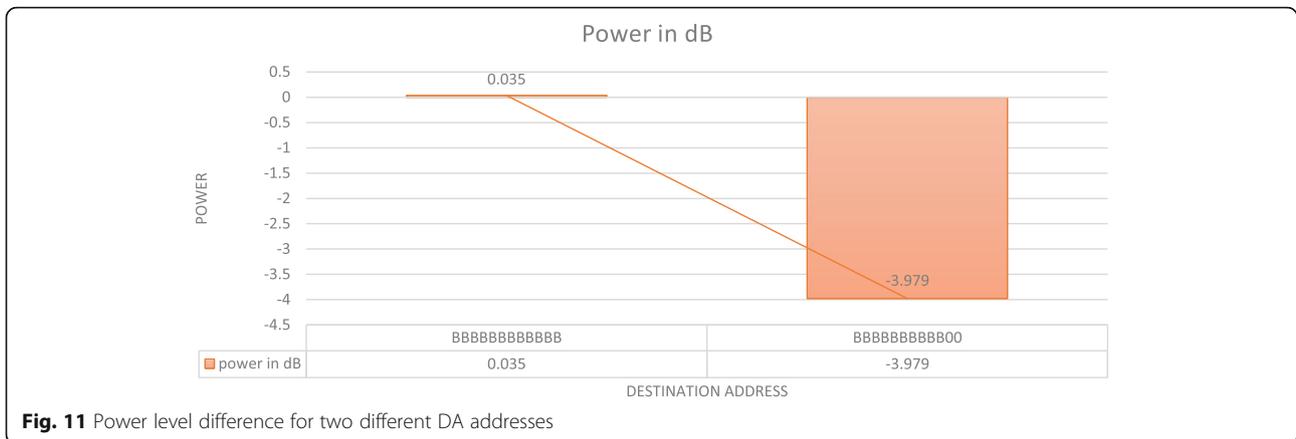


Fig. 11 Power level difference for two different DA addresses

header or in this case the Destination Address (DA) does not match any of the match field of any flow entry. We can see in Fig. 13, a clear distinction as to when the destination address is BBBBBBBBBB, from the condition when the destination address is changed to BBBBBBBBB00. Since in that case this packet should be sent to controller and a net path should be set. But since the controller is not yet programmed to set

a net path in this test-bed therefore we can see that this packet won't be transmitted further.

Conclusion

After carrying on the simulations of an optical network specifically a DWDM network and in comparison to it the SDON network proved design that is more efficient and led to making the networks

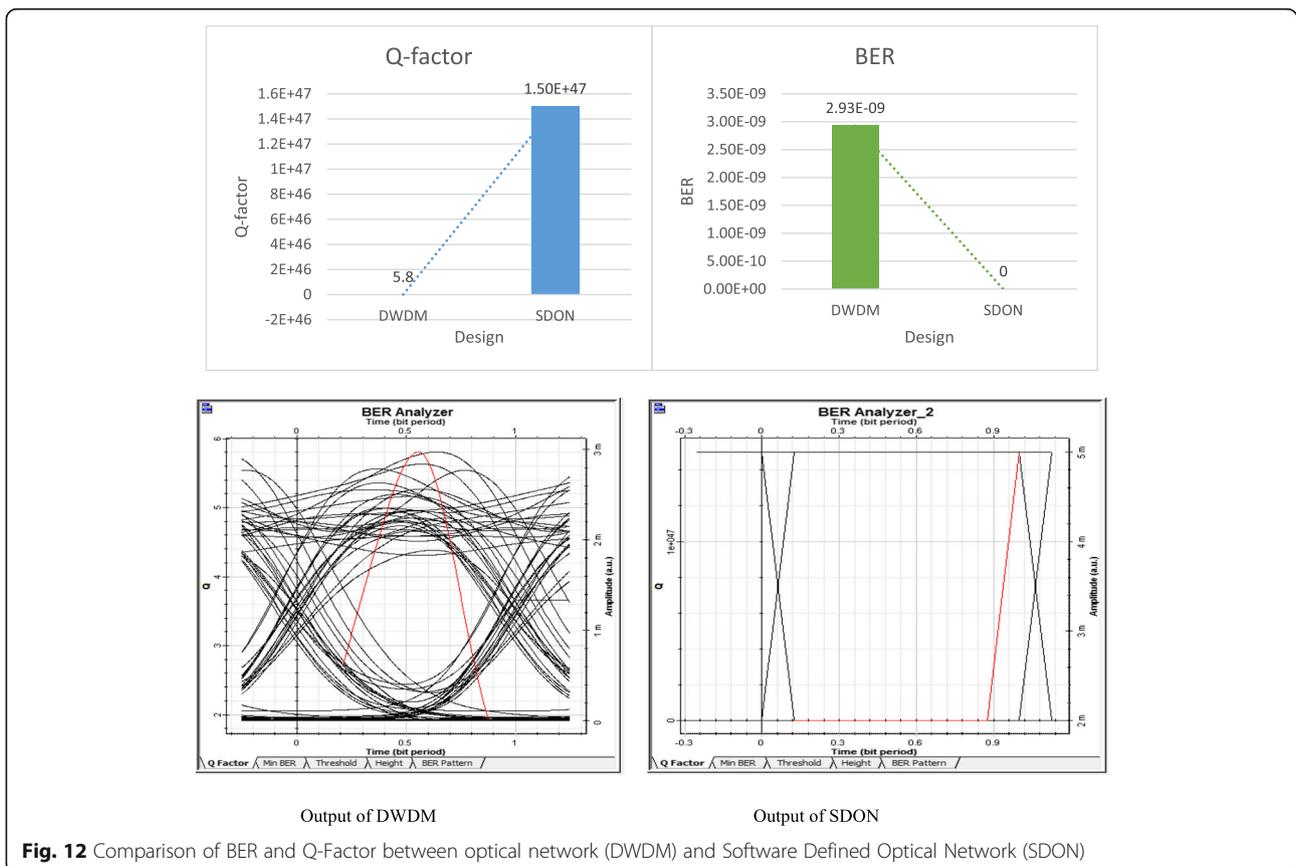
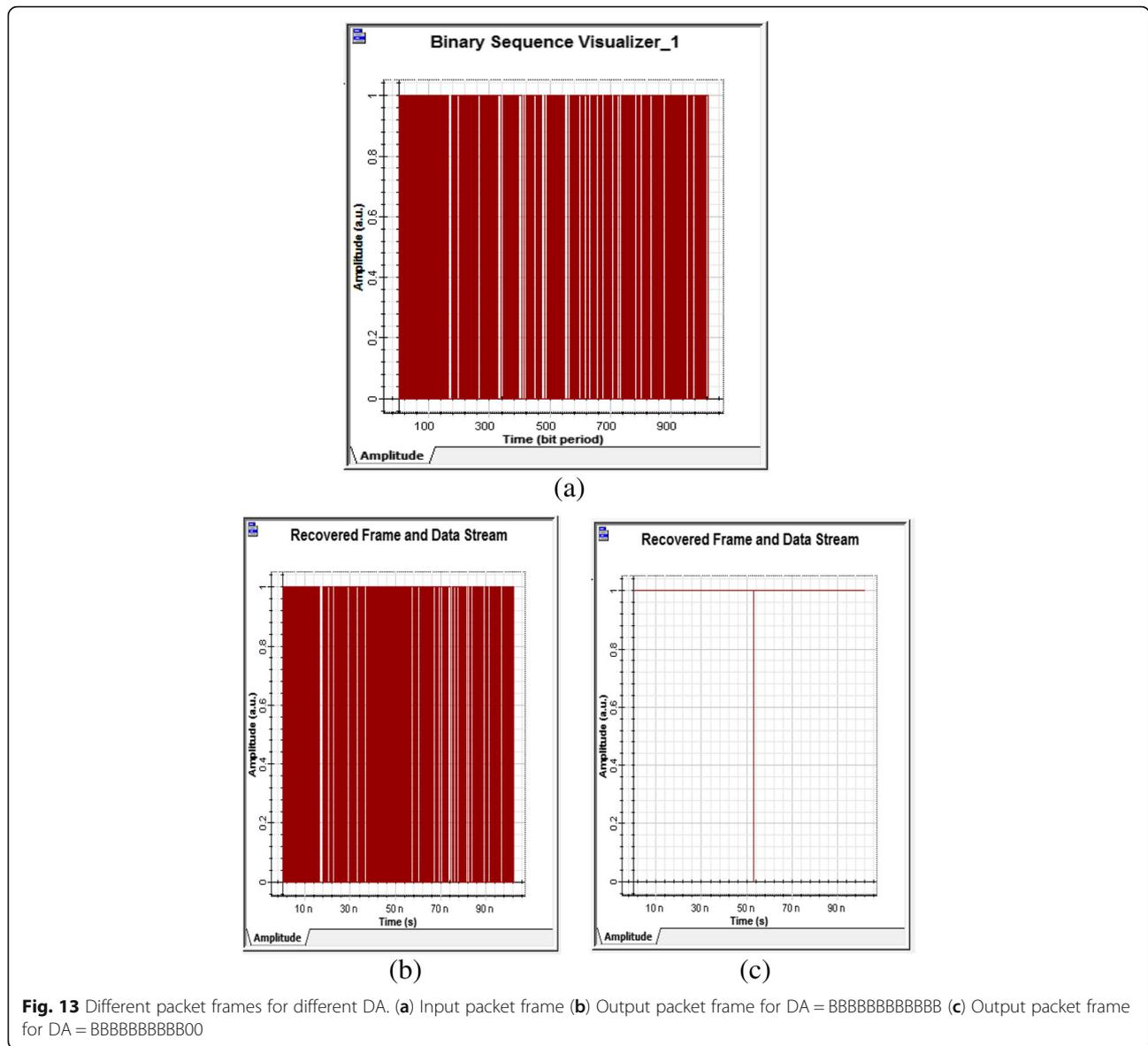


Fig. 12 Comparison of BER and Q-Factor between optical network (DWDM) and Software Defined Optical Network (SDON)



flexible and programmable. The BER and Q-factor obtained from the SDN were far better than obtained in case of just an optical network. The flexibility of the SDN network was proved when the network was able to transmit a particular packet of data, which contained a specific destination address that would match the match field of flow entry. While as, the packet with some other destination address, which does not match the match field of flow, entry either is dropped or lead towards the SDN controller, which in turn has a global view of the network and hence is able to reprogram the network according to the demands. Thus, this paper gives a clear distinction between the SDN and an optical network consisting of DWDM networks, and proves

as to how SDN leads to adaptability of network to current computing trends.

Appendix

Co-Simulation of MATLAB and OptiSystem

In OptiSystem, all the available components sometimes prove to be insufficient for the requirements of the new design that could meet the current computing trends. Thus, we want to design such components in some other tool and then co-simulate with our base tool like OptiSystem. Now the best possible tool available to us in which we can design any type of component and ensure anything required in our design is MATLAB. Thus, we do have a MATLAB component present in OptiSystem version 13 or 14. In order to co-simulate a MATLAB file

with the rest of the OptiSystem design we need to follow below mentioned steps:

❖ **STEP 1:** In the **Default**, section of OptiSystem library there is a subsection named **External Software Library** wherein we can find the **MATLAB Component**.

❖ **STEP 2:** Import the respective MATLAB component at a place where you need to have the component as per your requirements.

❖ **STEP 3:** Write the MATLAB code separately in the MATLAB tool and store it at some specific location in your PC.

❖ **STEP 4:** Double click on the MATLAB component of OptiSystem and there you will find a section named **Path** wherein you need to provide the complete path of the .m file, which you have already stored at some specific location in PC.

❖ **STEP 5:** after providing the path of .m file another section named Run Command is seen, where the name of the respective .m file is written followed by **.m**.

❖ **STEP 6:** The last and final step is to check the **Load** section in the properties block of MATLAB component, which would load the .m file and then you run your design and automatically the MATLAB component would be co-simulated with your OptiSystem design.

Acknowledgements

This work was supported in part by the 5G and IoT Lab, Department of Electronics and Communication Engineering.

Authors' contributions

BNML performed the simulation. RJ supervised the work and proposed the architecture and entire flow. RJ and BNML wrote the manuscript. Both authors read and approved the final manuscript.

Authors' information

Dr. Rakesh K Jha (S'10, M'13, SM 2015) is currently an Associate Professor in School of Electronics and Communication Engineering, Shri Mata Vaishno Devi University, Katra, Jammu and Kashmir, India. He is carrying out his research in wireless communication, power optimizations, wireless security issues and optical communications. He has done B.Tech. in Electronics and Communication Engineering from Bhopal, India and M. Tech from NIT Jalandhar, India. Received his PhD degree from NIT Surat, India in 2013. He has published more than 35 Science Citation Index Journals Papers Including many IEEE Transactions, IEEE Journal and more than 25 International Conference papers. His area of interest is Wireless communication, Optical Fiber Communication, Computer networks, and Security issues. Dr. Jha's one concept related to router of Wireless Communication has been accepted by ITU (International Telecommunication Union) in 2010. He has received young scientist author award by ITU in Dec 2010. He has received APAN fellowship in 2011,2012, 2017 and 2018 and student travel grant from COMSNET 2012. He is a senior member of IEEE, GISFI and SIAM, International Association of Engineers (IAENG) and ACCS (Advance Computing and Communication Society). He is also member of ACM and CSI, many patents and more than 1361 Citations in his credit.

Miss. Burhan Num Mina Llah received the B.E. degree in Electronics and Communication Engineering from University of Kashmir, Jammu and Kashmir, India, in 2012. She is currently pursuing the M. Tech degree in Electronics and Communication Engineering at Shri Mata Vaishno Devi University, Katra, Jammu and Kashmir, India. Her research interest includes the emerging technologies of Interface between SDN and Optical Networks. Currently she is doing her research work in Software Defined Optical Networks. She is working on Matlab, OptiSystem, Estinet for SDN and Optical Networks. She is receiving the teaching assistantship from MHRD.

Dr. Rakesh K Jha (S'10, M'13, SM 2015) is currently an Associate Professor in School of Electronics and Communication Engineering, Shri Mata Vaishno Devi University, Katra, Jammu and Kashmir, India. He is carrying out his research in wireless communication, power optimizations, wireless security issues and optical communications. He has done B.Tech. in Electronics and Communication Engineering from Bhopal, India and M. Tech from NIT Jalandhar, India. Received his PhD degree from NIT Surat, India in 2013. He has published more than 35 Science Citation Index Journals Papers Including many IEEE Transactions, IEEE Journal and more than 25 International Conference papers. His area of interest is Wireless communication, Optical Fiber Communication, Computer networks, and Security issues. Dr. Jha's one concept related to router of Wireless Communication has been accepted by ITU (International Telecommunication Union) in 2010. He has received young scientist author award by ITU in Dec 2010. He has received APAN fellowship in 2011,2012, 2017 and 2018 and student travel grant from COMSNET 2012. He is a senior member of IEEE, GISFI and SIAM, International Association of Engineers (IAENG) and ACCS (Advance Computing and Communication Society). He is also member of ACM and CSI, many patents and more than 1361 Citations in his credit.

Miss. Burhan Num Mina Llah received the B.E. degree in Electronics and Communication Engineering from University of Kashmir, Jammu and Kashmir, India, in 2012. She is currently pursuing the M. Tech degree in Electronics and Communication Engineering at Shri Mata Vaishno Devi University, Katra, Jammu and Kashmir, India.

Her research interest includes the emerging technologies of Interface between SDN and Optical Networks. Currently she is doing her research work in Software Defined Optical Networks. She is working on Matlab, OptiSystem, Estinet for SDN and Optical Networks.

She is receiving the teaching assistantship from MHRD.

Funding

This work was supported in part by the 5G and IoT Lab, Department of Electronics and Communication Engineering, TEQIP-III, and in part by TBIC- Shri Mata Vaishno Devi University, Katra, India.

Availability of data and materials

NA

Competing interests

The authors declare that they have no competing interests.

Received: 6 May 2018 Accepted: 16 May 2019

Published online: 28 June 2019

References

- Bahnasy, M., Idoudi, K., Elbiaze, H.: OpenFlow and GMPLS unified control Planes: testbed implementation and comparative study. *J Optical Commun Netw.* **7**(4), 301–313 (2015)
- Pupatwibul, P., Banjar, A., Sabbagh, A.A.L., Braun, R.: A comparative review: accurate OpenFlow simulation tools for prototyping. *J Netw.* **10**(5), 322–327 (2015)
- Son, J., Dastjerdi, A.V., Calheiros, R.N., Ji, X., Yoon, Y., Buyya, R.: CloudSimSDN: modeling and simulation of software-defined cloud data centers. In: Cluster, Cloud and Grid Computing (CCGrid), 15th IEEE/ACM International Symposium (2015) first quarter
- The Open Networking Foundations, "The OpenFlow switch specifications" Version 1.3.4, 2014
- Nunes, B.A.A., Mendonca, M., Nguyen, X.-N., Obraczka, K., Turletti, T.: A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks. *IEEE Commun Surv Tutor.* **16**(3), (2014)
- Bhaumik, P., Zhang, S., Chowdhury, P., Lee, S.-S., Lee, J.H., Mukherjee, B.: Software-defined optical networks (SDONs): a survey Springer Science+ Business Media New York. *Photon Netw. Commun.* **28**, 4_18 (2014)
- Singh, S., Jha, R.: SDOWN: a novel algorithm and comparative performance analysis of underlying infrastructure in software defined heterogeneous network. *Fiber Integr Opt.* **38**, 1–33 (2018)
- Singh, S., Jha, R.: A survey on software defined networking: architecture for next generation network. *J. Netw. Syst. Manag.* **25**(2), 321–374 (2017)
- Singh, S., Jha, R.K.: SDOWN: a novel algorithm for better quality of service and experience in software defined optical wireless network. *Optik.* **176**, 662–684 (2019)

10. Gupta, A., Jha, R.K.: A survey of 5G network: architecture and emerging technologies. *IEEE Access*. **3**, 1206–1232 (2015)
11. Zhao, Y., Ji, Y., Zhang, J., Li, H., Xiong, Q., Qiu, S.: Software defined networking (SDN) controlled all optical switching networks with multi-dimensional switching architecture. *Opt. Fiber Technol.* **20**, 353–357 (2014)
12. Introductory White Paper: Optical Software Defined Networking. MRV Communications (2014)
13. Introductory White Paper, “Software-Defined Networking: the New Norm for Networks (White Paper)”. <https://www.opennetworking.org/sdn-resources/sdn-library/whitepapers>. Accessed 2014
14. Zhao, Y., Zhang, J., Yang, H., Yu, Y.: Which is more suitable for the control over large scale optical networks, GMPLS or OpenFlow? In: Proceedings of OFC/NFOEC (2013)
15. Liu, L., Choi, H.Y., Casellas, R., Tsuritani, T., Morita, I., Martínez, R., Muñoz, R.: Demonstration of a dynamic transparent WSON employing flexible transmitter/receiver controlled by an open-flow/stateless PCE integrated control plane. In: Proceedings of OFC/NFOEC (2013)
16. Gringeri, S., Bitar, N., Xia, T.J.: Extending Software Defined Network Principles to Include Optical Transport. In: IEEE Communications Magazines. New Paradigms in Optical Communications and Networks. **5**(1), 32–40 (2013)
17. Casellas, R., Muñoz, R., Martínez, R.: Path computation elements (PCEs): applications and status. In: Proceedings of OFC/NFOEC (2013)
18. Martín-Minguez, A., Horche, P.R.: Simulation of transmission performance for equalized holographic ROADM designs. In: *Óptica Pura Y Aplicada*, section: Optoelectronics (2012)
19. Shirazipour, M., John, W., Kempf, J., Green, H., Tatipamula, M.: Realizing Packet-Optical Integration with SDN and OpenFlow 1.1 Extensions. In: IEEE international conference. Communication (ICC), 6633–6633 (2012)
20. Azodolmolky, S., Nejabati, R., Escalona, E., Jayakumar, R., Efsthathiou, N., Simeonidou, D.: Integrated OpenFlow-GMPLS Control Plane: An Overlay Model for Software Defined Packet over Optical Networks. In: IEEE conference and exhibition (2012) Optical Communication (ECOC), 37th European Conference
21. Liu, L., Tsuritani, T., Morita, I., Casellas, R., Martínez, R., Muñoz, R.: Control plane techniques for elastic optical networks: GMPLS/PCE vs OpenFlow. In: IEEE GlobecomWorkshops, pp. 352–357 (2012)
22. Liu, L., Tsuritani, T., Morita, I.: Experimental demonstration of OpenFlow/GMPLS interworking control plane for IP/ DWDM multi-layer optical networks. In: IEEE 14th Int. Conf. On Transparent Optical Networks (ICTON), pp. 1–4 (2012)
23. Doria, A., Hadi Salim, J., Haas, R., Khosravi, H., Wang, W., Dong, L., Gopal, R., Halpern, J.: “Forwarding and Control Element Separation (ForCES) Protocol Specification”. RFC 5810 (Proposed Standard) (2010)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
