Dynamic Openflow-controlled Optical Packet Switching Network

Xiaoyuan Cao, Noboru Yoshikane, Takehiro Tsuritani, Itsuro Morita, Masatoshi Suzuki
Takaya Miyazawa, Masaki Shiraiwa, Naoya Wada

(Irvised Paper)

Abstract—This paper presents and experimentally demonstrates the generalized architecture of Openflow-controlled optical packet switching (OPS) network. Openflow control is enabled by introducing the Openflow/OPS agent into the OPS network, which realizes the Openflow protocol translation and message exchange between the Openflow control plane and the underlying OPS nodes. With software-defined networking (SDN) and Openflow technique, the complex control functions of the conventional OPS network can be offloaded into a centralized and flexible control plane, while promoted control and operations can be provided due to centralized coordination of network resources. Furthermore, a contention-aware routing/rerouting strategy as well as a fast network adjustment mechanism is proposed and demonstrated for the first time as advanced Openflow control to route traffic and handle the network dynamics. With centralized SDN/Openflow control, the OPS network has the potential to have better resource utilization and enhanced network resilience at lower cost and less node complexity. Our work will accelerate the development of both OPS and SDN evolution.

Index Terms—Optical fiber networks, Optical packet switching, Openflow, Software-defined networking.

I. INTRODUCTION

Optical packet switching (OPS) [2] has been the goal for the optical switching technique evolution. OPS enables all-optical switching on a packet-basis and thus provides better network resource utilization, which is more suited for handling the future network dynamics than the current optical circuit switching (OCS) based WDM networks [3]. However, OPS network has not been commercially deployed due to obstacles related to both data and control planes. Besides the need for fast photonic switches in the data plane, the complex design and implementation in the control plane has been an even greater obstacle to the development and commercialization of OPS techniques. A conventional OPS node usually has its own control plane and works independently for hop-by-hop switching. The distributed control plane in each OPS node must be responsible for not only route calculation and resource assignment, but also contention resolution, etc. With various control functions implemented, the OPS node has become extremely complicated and heavy loaded, along with high cost and energy consumption.

However fortunately, with the introduction of the evolutionary concept of software-defined networking (SDN) and its major enabling technique known as Openflow [4, 5], this obstacle in controlling OPS node can be offloaded into a much more flexible software-based control plane, thus creating a more feasible OPS network. In a SDN/Openflow-controlled OPS (OF-OPS) network, the complex control functions of OPS network are performed in a centralized facility realized as the SDN/OpenFlow Controller [6, 7], while the simplified OPS node only needs to be in charge of the basic switch configurations and packet forwarding. In such an OF-OPS network, although packets are still switched in a distributed manner, the routing decisions are no longer made locally and independently. In other words, optimal routing decisions can be made for packet flows, thanks to the global network and resource view in the centralized Controller.

Nevertheless, a major challenge for the OF-OPS network is handling the network dynamics. In particular, since network resources are not reserved in advance for traffic flows in a statistically switched OPS network as in the OCS network, packets may experience contention during transmission in the network. Due to immature optical buffering techniques, the OPS network has to adopt extra control mechanisms and routing algorithms to alleviate network contentions. Besides, in a dynamic network such as OPS with various unpredictable traffic and an unstable environment, network status and resource distribution are constantly changing with the occurrence of contention and network failure for instance, requiring both the corresponding precaution and prompt adjustment in the network directed by the Controller, for the sake of reliable transmission and resilience enhancement. However, the good news is that with a centralized SDN control
plane that leverages global views of network resources and service demand information, promoted operations are provided for traffic routing and network dynamics can be handled more efficiently.

In light of this, this paper presents the generalized OF-OPS network architecture and the corresponding Openflow control process [1, 8]. After conducting pioneering research work on SDN/Openflow-controlled OCS network [9] and interworking between the Openflow network and an independent OPS network [10], we are the first to demonstrate OF-OPS networking. We also propose and experimentally demonstrate the advanced Openflow control mechanism for handling the network dynamics with contention alleviation and fast network adjustment ability.

The remainder of this paper is organized as follows. Section II describes the OF-OPS network architecture and the basic Openflow control process. Section III introduces the dynamic Openflow control mechanism, including a contention-aware routing/re-routing strategy to assign traffic in a load-balanced way and a network adjustment mechanism enabling OF-OPS network to promptly respond to network status changes and recover from network failure. Section IV shows the proof-of-concept demonstration and experimental results from both control and data plane. Section V concludes this paper.

II. OPENFLOW-CONTROLLED OPS NETWORK

As shown in Fig. 1, a generalized OF-OPS network essentially comprises of the centralized Controller and the distributed OPS nodes. Ideally, an Openflow-supportive OPS node is able to communicate with the Controller directly [8]. Unfortunately, such OPS node is still not available at this moment. However, with the introduction of an additional network element namely Openflow/OPS Agent (OFA) between the Controller and the conventional optical packet switch, communication is enabled between each other. The OFA is an extended Openflow switch that acts as a proxy for the southbound optical packet switch. The OFA virtualizes the resources of the switch (ports, available incoming/outgoing wavelengths and links, etc.) for the Controller, and translates the Openflow protocol messages into standard commands or messages that can be understood by the switch, thus enabling communication between the Controller and the optical packet switch through information exchange and protocol translation in the OFA. The optical packet switch and the OFA together compose the new type of OPS node, which is referred as OF-OPS Node. Extended Openflow protocol is supported in OF-OPS network, where Flowtable [11] is embedded in the OF-OPS node as the rule for packet forwarding. The Flowtable is extended to include more information (label, wavelength, etc.) in the matching field. Accordingly, the Openflow protocol messages between Controller and OF-OPS node are extended to pass on such new information.

Thanks to SDN/Openflow, conventional complex control functions can be offloaded to the centralized Controller, while the OF-OPS node is only in charge of distributed Flowtable matching and packet forwarding. In the OF-OPS network, the Controller is in charge of the main network control functions, including routing, resource assignment, contention resolution and protection/restoration, etc. An extra path computation element (PCE) [12, 13] is needed if the control functions or routing algorithms are too complex or heavy loaded to be implemented in the Controller. Likewise, external topology server [13] could be implemented for recording the up-to-date topology information and resource distributions.

The detailed control process is illustrated in Fig. 2 and further explained here as follows (from i) to v)), which indicates the relationship between different network elements and their message exchange process in the vertical view:

i) Due to the flow-based Openflow control mechanism, incoming packets are classified and aggregated into flows at the host/border node. Upon receiving a new packet, the host/border node sends a new Packet-in message [11] to the Controller if no matches are found in the Flowtable (this is typically the case for the first packet in a new flow). For OF-OPS node, the optical packet switch converts the packet header into the electrical form and forwards it to the OFA. Then based on this packet header, the OFA generates a Packet-in message and forwards it to the Controller.

ii) The Controller processes the Packet-in message and sends a request message to the PCE with the related information. Packet-in messages usually contain necessary information abstracted from the original packet header, normally but not always including IP address, input port, MAC address, VLAN ID, etc. It is up to the Controller which information would be useful and processed.

iii) PCE calculates the route and appropriate resource (output port, outgoing channel or shared wavelengths) for the new
packet according to the current topology view and the request message received from the Controller; then sends a reply message to the Controller with the information including routing decision, etc.

iv) The Controller extracts the information and sends Flowtable modification messages (OFPT_FLOW_MOD) [11] to OFAs and host/border nodes to set up Flowtables, and provides necessary information to OFAs in the meantime.

v) OFAs extract information (destination IP address, input and output ports, outgoing wavelengths, etc.) from Flowtable modification messages and information received from the Controller, construct standard commands or messages in order to configure the southbound optical packet switches. Packets will be switched accordingly afterwards.

As in any other centralized system, scalability could be a controversial issue in OF-OPS network. As the Controller/PCE needs to deal with all the Packet-in messages and is in charge of all the underlying OF-OPS nodes, optimized algorithms and efficient CPU system are required. Fortunately, optimization and simplification of algorithms in the software-based control plane has been widely studied, while the ever-increasingly efficient CPU system nowadays is already powerful enough to take up quite a number of responsibilities. On the other side, if all these control functions were loaded into each OPS node as in the conventional fully-distributed OPS system, the OPS node would become extremely complicated and expensive, which is an obstacle to the development of OPS technology. Furthermore, a hierarchical structure is always an efficient solution for the scalability issue, where a large OF-OPS network could be divided into multiple sub-domains, which are separately controlled but coordinated by a top-level Supreme Controller namely Orchestrator. Therefore, an OPS network would definitely benefit rather than harmed from centralized but flexible and optimal SDN/Openflow control.

### III. DYNAMIC OPENFLOW CONTROL MECHANISM

Although the packet-based switching characteristic gives OPS better bandwidth utilization, it also induces unreliability to the network. As a path is not reserved for each packet flow in OPS as in OCS, contentions may happen. Extra measures should be taken in the OPS network in order to alleviate contention since buffering is no longer sufficient in the optical domain. In the conventional OPS network, it is difficult to prevent contention as each OPS node can only make local routing decisions without knowledge of the global network status. Besides, implementations of additional distributed algorithms would make the OPS node extra complicated. But such dilemma can be solved in an OF-OPS network. With the centralized control ability of SDN/Openflow, the Controller and PCE are able to leverage the global knowledge of the network topology and resource distributions. And for that, a better routing and resource assignment strategy can be provided for each packet flow in order to avoid future contention and decrease the packet loss rate.

Moreover, the network environment is constantly changing due to contention, topology mutation, resource pre-emption or even network failure, etc. It is crucial for the OPS network to be capable of reacting to these network dynamics promptly and adjust its configurations with up-to-date network status accordingly. For instance, with the change of certain port or link availability, the Controller needs to be notified in time by the OPS nodes to update the topology knowledge for route recalciulation and network reconfiguration.

Hence, we propose the following dynamic Openflow control mechanism including a contention-aware routing/rerouting strategy to allocate packet flows in a load-balanced way for OF-OPS network to alleviate potential contention or reroute traffic when network status changes, as well as a network adjustment mechanism to handle the network status changes.

#### A. Contention-aware routing/rerouting strategy

In an OF-OPS network, topology and resource information are constantly fed into the centralized control plane from the data plane, where they are stored in the topology server and used by the Controller and PCE as a reference for routing and resource allocation. The essence of the strategy is switching the packet to a route with the most available resources, so that it would have a less chance of encountering future potential contentions. The detailed strategy is illustrated in Fig. 3 and explained as follows:

1) Upon receiving a Packet-in message at each hop or a request for rerouting in the case of a network status change, the Controller informs PCE to firstly calculate K shortest paths [14] (fewest hops) for this incoming packet or request (K can be adjusted accordingly) within the current topology.

2) We sort these K paths in the order by the estimated overall resource availability along the path, which depends on the most occupied link (bottleneck link) throughout this path. Normally for an OPS network, the resource availability on one link is
related to the wavelength bandwidth (BW) and the traffic rates of packet flows in transmitting (R_p). More specifically, for a certain wavelength \( \lambda \) on link \( j \), assume the total capacity of the wavelength bandwidth is BW, the number of packet flows at wavelength \( \lambda \) on link \( j \) is \( n \), and the traffic rate of each packet flow is \( R_{pi} \). The resource availability on wavelength \( \lambda \) of link \( j \) is calculated as follows:

\[
C_j^{\lambda} = \text{BW} - \sum_{i=1}^{n} R_{pi}
\]

Assume the total number of links that this path traverses is \( L \), the overall resource availability of wavelength \( \lambda \) along the path is calculated as:

\[
C_j^{\lambda} = \min_{j=1}^{L} \{ C_j^{\lambda} \}
\]

3) We choose the path with the most resources available for the packet flow or request and assign it the corresponding output port (port, wavelength).

Based on load-balanced routing, which has been proved to have higher network throughput [15], our strategy is a proactive approach to alleviate possible contention. The key to contention resolution is resource allocation. In an OPS network, contention occurs when two or more packets contend for the same output port at the same time, i.e. when resources are insufficient for all the packets. Two typical approaches to resolving contention include optical buffering, which buffers the packet until resources are available, and deflection routing, which routes the contending packets to a different output port that has more resources available [15]. Our approach, however, as an improved load-balanced routing strategy, makes use of the overall network resource utilization thanks to the centralized control of SDN, i.e. for each routing decision, the overall resource availability across multiple hops is checked instead of just one hop, making sure the routing is globally load-balanced. Thus if a packet flow traverses through a route with more resources available, it would have less chance to come across packets from other flows and contend for the limited resources along the whole path. Consider an extreme case where a packet flow could be switched to two different paths: one vacant path \( P_1 \) and one fully occupied path \( P_2 \). The obvious and reasonable choice is path \( P_1 \) as packets would be blocked in path \( P_2 \) since there are no free resources on this path.

An illustrative example of a five-node network with two available wavelengths (4G total BW capacity per wavelength) is shown in Fig. 4. The approximate traffic rates (bps) of packet flows in transmitting \( R_p \) are shown in the figure, which can be estimated by monitoring the incoming rate of the packets at the input port of the OPS nodes. When the OPS node receives a packet and processes its header in the electrical domain, the packet information is recorded at the same time, including which flow it belongs to, as well as its output link and wavelength. In this way, we can easily obtain the average traffic rate of each flow in each link. Although traffic could be dynamically or even dramatically changed in an OPS network and the flow rate is an estimated average number, it is constantly monitored and timely updated to reflect the current traffic status. Deducing from the total wavelength capacity, we can calculate the resources available in the network. Assume a new packet flow is coming to node A and destined to node E, and we need to find the next hop for packets at node A.

In the case that A is an edge node that has wavelength conversion ability: firstly we use K-shortest path algorithm to find up to three paths (K is set as 3 here): A-C-E, A-B-E, and A-B-D-E. Then we need to choose one path from these three paths according to the resource availability. For simplicity, since the total wavelength BW is the same, we directly calculate the overall resource occupancy (i.e. \( R_p \)) for each wavelength along these paths as shown in the top line of Table 1. A-B-E \( \lambda_3 \) and A-B-D-E \( \lambda_4 \) have the least resource occupancy, and thus the most resource availability. Considering the number of hops, A-B-E \( \lambda_3 \) is chosen and the packet flow should be switched to output port 1 on wavelength \( \lambda_3 \).

In the case that node A is an intermediate node: due to the wavelength continuity constraint, only the same wavelength as the incoming wavelength is considered as a candidate for output. As shown in the middle and bottom line of Table 1, if the incoming wavelength of the packet flow is \( \lambda_1 \), we choose the path A-B-E from three candidate paths A-C-E \( \lambda_1 \), A-B-E \( \lambda_1 \), A-B-D-E \( \lambda_1 \), and switch the packet to output port 1 on wavelength \( \lambda_1 \); if the incoming wavelength of the packet flow is \( \lambda_2 \), we choose the path A-B-D-E from three candidate paths A-C-E \( \lambda_2 \), A-B-E \( \lambda_2 \), A-B-D-E \( \lambda_2 \), and switch the packet to output port 1 on wavelength \( \lambda_2 \). For the downstream nodes, we apply the same strategy to find the next hop for them.

B. Advanced network adjustment mechanism

After the basic Openflow control process, the transmission in the OPS network is established. In order to adjust the network configuration when the network status changes, an advanced Openflow control mechanism is proposed. To achieve a prompt response, both the Controller and OFAs are set to be actively
listening to notification messages from the southbound devices indicating the affected node/link, via either Openflow protocols or instant socket sessions. As soon as the notification message reaches the Controller, the topology information is updated so that Controller would be able to calculate routes according to the current network status. To enable this control mechanism, the optical packet switch should have a monitor module that is able to detect any transmission failure and send trigger messages to the northbound control plane. The detailed control process is also shown in Fig. 2 and further explained as follows (from vi) to (xi):

vi) When the network status changes, the Optical packet switch notifies OFA of its port/link status change.

vii) OFA notifies the Controller and provides the information of the changed port/link.

viii) The Controller reads the notification message, updates the network topology and sends a request to PCE.

ix) PCE recalculates the route and appropriate resources according to the current topology view, and then replies to the Controller. The proposed contention-aware routing/rerouting strategy is applied here.

x) The Controller sends out Flowtable modification messages including new information to OFAs.

xi) OFAs read the new messages sent from the Controller, and construct new commands/messages for optical packet switches. Packets will be switched according to the updated node configurations afterwards.

IV. PROOF-OF-CONCEPT DEMONSTRATION

A. Optical Packet and Circuit Integrated (OPCI) node

The optical packet switch used in our demonstration is the OPCI node introduced in [10, 16], although only the OPS part was used here. Together with the corresponding OFA, they comprise the OF-OPS node. As shown in Fig.5, the OPCI node switches packets according to a label called OP-ID assigned at the edge node. The optical packet transponder (OP-TF) has a Label mapping table that associates the destination IP address (first three bytes, i.e. X.X.X/24) of the incoming packet to a designated OP-ID, according to which the node attaches this OP-ID to each packet for later switching. The Ethernet frame is converted to a multi-wavelength optical packet [10, 16] at the edge OPCI node. Each OPCI node holds another Forwarding table, according to which the switch is configured to forward the packet with a certain OP-ID from an input port to the appropriate output port. In this experiment, OFA enables Openflow control over the OPS network via virtualizing the OPCI node and translating the Flowtable modification message into commands to configure the Label mapping table and Forwarding table. Since the OPCI node is not able to discover neighboring nodes (thus links) and dynamically provide port information to the OFA, we manually/statically set up the virtual port/link of the OFA to represent the OPCI node. For each packet, at the edge node, a unique OP-ID is found in the database according to its destination IP address extracted from the Packet-in message and commands are sent from OFA to OPCI node to set up the Label mapping table. Currently, the OPCI node only uses the destination IP address for OP-ID mapping as a packet classification mechanism. However, with upgraded OPCI node in the future, it is possible to utilize more information in the Packet-in message for OP-ID mapping and therefore better packet classification. Then at each OPCI node (both edge and intermediate node), the route is calculated in the Controller/PCE and a command is sent from OFA to OPCI node to set up the Forwarding table. As traffic is transmitted in multi-wavelength optical packets form at the same waveband,
the routing decision is only needed to indicate the output port, regardless of the wavelength.

B. Experimental demonstration

We set up our experimental network as shown in Fig. 6 and Fig. 7, including four OPCI nodes (N1, N2, N3, N4) with attached OFAs, host nodes/edge Openflow switches (H1, H2, H3, H4), as well as a centralized Controller (POX) and PCE. Information exchange between Controller and PCE was enabled through TCP-based Socket sessions. Each link had the same total capacity. As shown in Fig. 7, Ethernet traffic with different bit rates were generated using 10GbE generators from three host nodes H1 (0.6 Gbps), H2 (1.25 Gbps), and H3 (2.5 Gbps), converted into optical packets at N1, N2, N3, and all destined to the fourth host node H4. Originally, with the calculated routing result from PCE, the packet flow from host H1 denoted as P1 was assigned to the route from N1 to N4, the packet flow from host H2 denoted as P2 was assigned to the route from N2 to N4, and the packet flow from host H3 denoted as P3 was assigned to the route from N3 to N4. All flows were given an OP-ID of 0x02 according to their destination IP address (10.0.2.1/24).

The demonstration of the Openflow control process is shown and explained in Fig. 6, including the OP-ID assignment in the Controller, the route calculation in PCE for traffic P1, the sample Flowtable modification message from the Controller to OFA1, and the sample commands sent from OFA1 to N1 for table configurations. The Flowtable modification message was modified to include the “Set VLAN ID” action in order to pass on the OP-ID information to OFA1. Input port number and destination IP address were read from the “Match” field. Output port number and OP-ID were read from two “Action” fields. Fig. 8 shows the optical spectra of the multi-wavelength optical packet. We also monitored the waveforms of these three optical packet flows at several measurement points (marked in Fig. 7) in the network as shown in Fig. 9, i.e. P1 at point A between N1 and N4 (Fig. 9 (a)), P2 at point B between N2 and N4 (Fig. 9 (b)), P3 at point C between N3 and N4 (Fig. 9 (d)).

After the basic transmission setup was finished, we emulated the network dynamics by cutting off the link between N1 and N4, which affected the original traffic P1, making N4 notifies OFA4 (loss of light at the input port of N4) and then the Controller of the network status change (link failure). In our demonstration, instant socket sessions were used to transmit notification messages, including the affected port and link. Upon receiving the notification, the Controller updated the network topology information and requested PCE to recalculate the route for traffic P1. The proposed contention-aware routing/rerouting strategy was applied here. PCE firstly found two candidate rerouting options (N1-N2-N4 and N1-N3-N4), and then checked their resource availability by comparing the current traffic bit rate in transmitting (R_p) on the route for both options. As link N2-N4 was partially taken up by P2 (1.25 Gbps) and link N3-N4 was partially taken up by P3 (2.5 Gbps), PCE chose path N1-N2-N4 as the new routing since P3 has a higher bit rate, which could cause more potential contention. Therefore, the original traffic P1 was switched (denoted as new
traffic P1’ to a new output port 1 from the previous output port 2 at node N1, under the instructions of OFA1. Traffic P1’ was statistically multiplexed with P2 at node N2, and then multiplexed with P3 at node N4. Thanks to the packet-based switching characteristic of OPS, multiple traffic flows were able to share the same wavelengths on the common links along the paths. The aggregated traffic P1’ & P2 at monitor point B and aggregated traffic P1’ & P2 & P3 at monitor point D are shown in Fig. 9 (e) and Fig. 9 (f) respectively.

To further demonstrate the dynamic Openflow control mechanism and the contention-aware rerouting strategy, we created a second scenario where the bit rate of traffic P2 was increased to 5 Gbps (denoted as new traffic P2* and shown in Fig. 9 (c)). We repeated the rerouting process for traffic P1 after cutting off the link between N1 and N4. At this time, since P2* has a higher bit rate than P3 which makes path N1-N2-N4 more occupied, path N1-N3-N4 was chosen instead. Therefore, the original traffic P1 was switched (denoted as new traffic P1”) to a new output port 3 from the previous output port 2 at node N1. Afterwards, traffic P1” was statistically multiplexed with P3 at node N3, and further multiplexed with P2* at node N4. The aggregated traffic P1” & P3 at monitor point C and aggregated traffic P1” & P2* & P3 at monitor point D are shown in Fig. 9 (g) and Fig. 9 (h) respectively.

As can be seen in the waveforms in Fig. 9, among P2, P2* and P3, it is easier to insert traffic P1 into P2 (Scenario 1) and P3 (Scenario 2) without interfering with each other as there is larger interval between two adjacent packets, which was ensured by our contention-aware routing/rerouting strategy. Fig. 10 shows the control process and forwarding table re-configuration in these two different scenarios, where commands sent from OFA1 to node N1 indicate different rerouting operations at node N1 when network dynamic occurs under different network resource distributions.

Although performing dynamic Openflow control has extra yet limited process overheads, better performance in terms of alleviating potential contention and guaranteed transmission quality can be provided. The measured responding time from OFA receiving the notification to OFA sending out the new commands was only 0.56 ms, mainly depending on the recalculation process in the Controller and PCE.

V. CONCLUSION

For a long time, the dilemma between the vision of OPS and the reality of its complicated and costly control plane design has been perplexing both academic researchers and business partners. No doubt that the introduction of SDN/Openflow has opened a new window for the development of OPS networks. In view of this, we have presented and also experimentally demonstrated the OF-OPS network architecture with the corresponding Openflow control process. An additional network element called OFA is introduced into the OF-OPS network to enable Openflow control over an OPS network. Furthermore, we have proposed and demonstrated a dynamic Openflow control mechanism for contention alleviation and fast network adjustment with prompt response to network status changes. With centralized SDN/Openflow control, the OPS network has the potential to have better resource utilization and enhanced network resilience.

REFERENCES