Traffic Management in SDN-enabled Optical Packet Switching Intra-Datacenter Network

E. Dutisseuil, B. Uscumlic, J.M. Estaran Tolosa, H. Mardoyan, Q. Pham Van,
M. Dallaglio, A. Dupas and Y. Pointurier
Nokia Bell Labs, France
eric.dutisseuil@nokia-bell-labs.com

***Abstract*— Slotted optical packet switching (OPS) is foreseen as future technology for intra datacenter network that will have to provide dynamic reconfiguration capacity, scalability, low latency and QoS guarantees to support fast growing emerging services. In this paper, we describe a prototype based on this technology and reconfigurable thanks to implementation of Software Defined Network (SDN) functions. We demonstrate plug-and-play network upgrade, reconfiguration capability within a few microseconds and low latency controlled within a few ns. Then we present how SDN functions can be distributed on-the-fly into the network through the control plane, again within microseconds.**

Keywords—datacenter; optical ring; optical slot switching; SDN

I. Introduction

Internet of Things (IoT) will have strong presence in future, where an ever-increasing number of connected devices and machines will generate new traffic patterns and services in the networks. Among them, connected vehicles and medical alarms are examples of future applications that will require Quality-of-Service (QoS) guarantees -as capacity and latency- in the networks including datacenters. Additionally, fast growing cloud-based applications or High Performance Computing (HPC) will require low latency guarantees within dynamically reconfigurable traffic [1]. These applications will be limited by current datacenter architectures based on IP and Ethernet technologies that cannot guaranty QoS with high capacity. Moreover, many papers have assessed the increasing need for scalability and higher bandwidth to sustain traffic growth, and for flexible and dynamic resource allocation to efficiently support the diversity and variation of traffic characteristics [2 and references herein]. Introduction of all-optical network based on Wavelength Division Multiplexing (WDM) optical slotted burst switching has been proposed as a solution to overcome the aforementioned limitations inside datacenter [3, 4]. In [5], we proposed an optical packet switched fabric called BOSS ("Burst Optical Slot Switching") that ensures the inter-connection of the servers at the top-of-rack (ToR) level by nodes using high-speed WDM interfaces. The proposed topology is a 2D-torus fabric based on interconnected optical rings as shown in Fig. 1. Data slots and their control information are physically separated using different WDM channels travelling synchronously in time-slotted windows along a single fiber. This separation allows the network to be operated using a Software Defined Network (SDN) approach.

In this paper, we describe for the first time a real implementation of key SDN functions for such network including the interaction between an open-source SDN controller and the network nodes through a control plane. This paper is organized as follows: In section II we present the experimental test-bench including the node architecture (A), the control plane operations (B) and the electronic processing implementation (C). Then we present in Section III the SDN operations performed and key results, before concluding in Section IV.

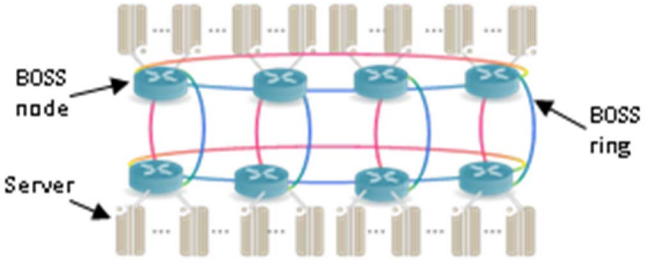

Fig. 1. Datacenter inter-connections based on Burst Optical Slot Switching (BOSS) rings.

II. Experimental Testbed

A. Experimental setup

We focus on the implementation of a generic single-ring topology and its operation with an open-source SDN control. Our BOSS ring includes three distinct and asynchronous (each running with its own clock) nodes dispatched along a single 537m fiber link that transports one data channel and the control channel on two different wavelengths (Fig. 2). So, although BOSS concept uses WDM scheme for data plane, only one wavelength is used here which is sufficient for the experiments reported here. As represented in Fig. 2, each node is connected on its client interface to a computer through a 10 Gb/s optical

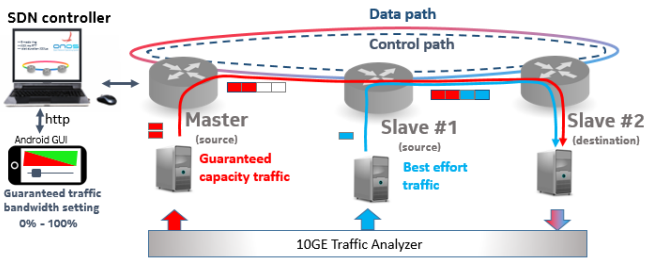

Fig. 2. Demonstrator architecture. The ONOS SDN controller allocates capacity for guaranteed traffic (red); unreserved slots can be used by Best Effort traffic (blue).

Ethernet adapter and to the optical ring on its line interfaces. One of the nodes called "master" is responsible for the ring synchronization (see Section III-A) and for the interconnection to the SDN controller. The other nodes are called "slaves". Each node is built around an FPGA platform that performs all the electronic processing and is equipped with 10 Gb/s Small Form Factor (SFP+) modules for data and control channel emission and reception. Two sets of wavelength-(de)mux and 1:2-(de)mux ensure the separation/mixing and duplication/insertion of channels as shown in Fig. 3. Electro-optical data conversions are performed only at source and destination edges, transit through intermediate nodes being transparent. The experiments presented here do not involve the so-called wavelength-reuse [5] operation, so no device for slot erasing after drop is used.

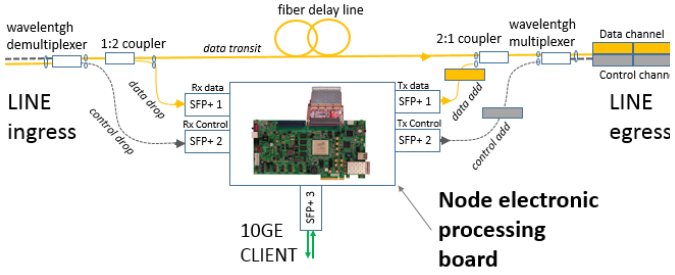

Fig. 3. BOSS node implementation.

B. Electronic processing

Our demonstrator uses one electronic board -equipped with a Kintex xcku040 FPGA- per node that hosts all electronic operations. The code architecture (Fig. 4) includes the parallel processing of the Control Channel information and the add/drop operation in the Data Channel. In between the SDN controller interacts with the control block and manage the authorizations for the data reception and emission between the Data Channel and the client 10GE interface. This management is assured on one hand by an embedded local database containing the scheduling rules specific to each node that can be modified on the fly using dedicated field bits inside the control channel frames (denoted "SDN control field" in Fig. 6), and one the other hand by the data slot information also transported on the control frame ("slot control field" in Fig. 6). Each control slot contains routing and availability information for the associated synchronous data slot as shown in Fig. 6. For both channels the line side edge blocks assure the data

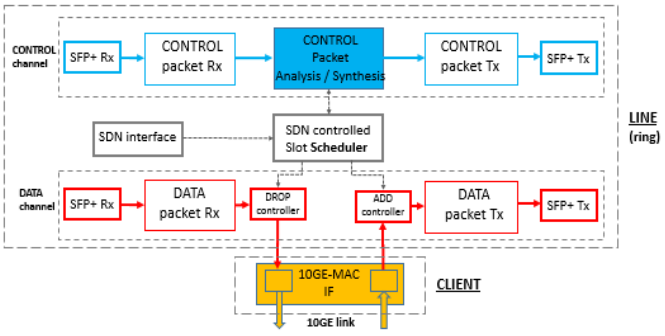

Fig. 4. BOSS node implementation FPGA code architecture ("master" node version with its SDN interface).

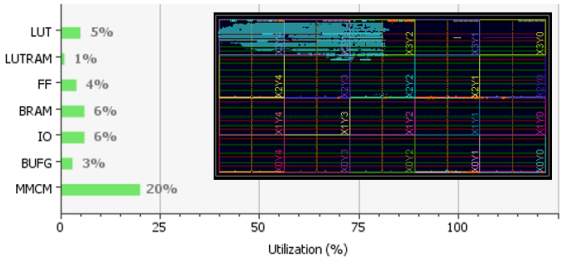

Fig. 5. BOSS node implementation FPGA code occupancy: percentage of resources utilization, resources area occupancy (inset).

encapsulation/decapsulation in/from the time slots, the 8b/10b encoding/decoding, and the framing/deframing. The master node code also supports a dedicated communication interface with the SDN controller. Fig. 5 shows the localization and amount of logic elements used in the FPGA of the master node, all electronic processing requires a very low amount of logic resources. Our implementation uses 12300 Look-Up-Table (LUT) and 18500 registers (FF), which represents for the FPGA used here less than 5 % of available logic elements.

C. SDN-controller and interaction with nodes

Right after the initialization process (described in Section III-A), the master node transmits the network topology (ring length, number of nodes, ..) to the SDN controller. This information is stored in an Open Network Operating System (ONOS) [7] database and displayed on a graphic user interface (see Fig. 2, top left). On the downstream path, an end-user sets new requirements (e.g. capacity request) through an Android device and an ONOS Rest API (Application Programming Interface) to the SDN controller which determines all scheduling rules and transmits them to the master node. This node retransmits the rules for the network operation to all slave nodes through the dedicated "SDN control field" in the control frame. These parameters are then stored into the local database of each node. Other fields of the control frames contain information regarding the associated data slot that may concerns the source, destination, Class of Service (CoS), Quality of Service (QoS), slot occupancy or else. These fields are received by each downstream slave node and their content is analyzed and modified if needed, and forwarded to the next node. At each node, the local scheduler controls any decision regarding data dropping or insertion on the data channel using the information of the associated control packet and the local database. Each node is then able to decide if the incoming line packet has to be dropped, and if incoming client data are allowed to be transmitted in an available slot. As soon as a transmission is allowed, the next awaiting client Ethernet frame is read from the Medium Access Control (MAC) First-in-First-out buffer

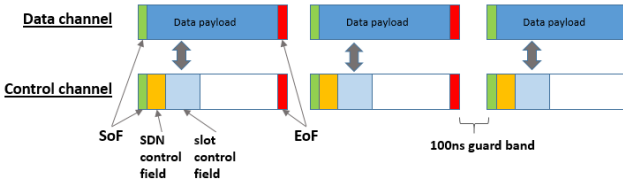

Fig. 6. Data and control slot structure.

(FIFO), then encapsulated into a 8b/10b frame and sent on the line ring within the slot boundaries.

III. RESULTS

A. Timing control

Precise timing control of the slotted operations along a ring topology is crucial in order to avoid any packet collision, mainly for packet insertion mechanism. Several works as [6] used a pulsed trigger signal generated from a master node to operate the slots synchronization, where an adjustment of the pulse period to the Round Trip Time (RTT) of the ring has to be performed. In our demonstrator, using standard 8b/10b encoding/framing for data and control packet allows to easily retrieve at each node the packets starting (SoF) and ending (EoF) boundaries using the dedicated 8b/10b K-control symbols that serve here in place of any added physical trigger signal (Fig. 6). These packets boundaries detected at each slave nodes receiver are used for packet insertion without collision in between those in transit optically delayed (Fig. 3). This mechanism provides an automatic self-synchronized insertion of packet inside the ring. On the other hand, insertion of packet without collision at master node that sources the packet flow, relies on the simple method that consist in constraining the slot length as an integer sub-division of the ring length. At any reset of the ring network, the master node initiates a clock counter and sends on the control channel a single frame containing a dedicated data flag. After a ring loop including all slave nodes latency this data flag is received by the master node which then stops the counter. The final value of this counter gives the requested RTT evaluation in a similar way as [6], and the slot duration is given by a predefined tabulated memory that contains optimal values between ring occupancy and typical 10G-Ethernet (or else) frame length. This automatic initialization process allows the network to auto-discover itself, for example when an extra node is added. For our setup, we measured this auto-discovery procedure to last only 4.6 µs. After this initialization phase, the master node enters into the normal operation mode and sends a continuous slot flow. As stated before, a precise timing control of the slots along the ring is crucial for slot add/drop operations, for data and control slots synchronization or for any operations performed in the dedicated guard-band (Fig. 6) inserted between slots, like wavelength switching or slot erasing (as mentioned before but not used here). Fig. 7(left) shows the statistics of 420 iterations of the ring RTT measurement, expressed in 6.4 ns clock period unit. We found that the ring length is evaluated with an accuracy of less than one clock, which corresponds to less than 1.3 meter (i.e. 0.2% in our case w.r.t. 537m of fiber), and we verified that this is independent of the ring length. Also, Fig. 7(right) shows the histogram of 580 measurements of slot timing skew evaluated between ingoing and outgoing slots at mater node level. We show that the skew variation is lower than one clock, i.e. lower than 6.4 ns, which corresponds to 6.4% of the 100 ns guard-band. This guarantees that there is no overlapping between slots and switching operations performed in the guard band time. Note that the above timing accuracy values can be easily improved (at least by a factor of 2) by increasing the FPGA running frequency.

B. Node latency

In introduction, we mentioned the issue of high and unguaranteed latency in Ethernet switching technology used in current datacenter. Using the RTT evaluation procedure before and after by-passing a node, we measure a latency of one node as low as 204 ns, which is far below typical value of several µsec encountered with common Ethernet switches. Our value is mainly due to the 8b/10b 10Gbps serial-deserializer (serdes) embedded into the FPGA.

C. Auto-reconfiguration

The need of scalability in datacenter yields recurrent modification or addition of resources. Ring topology may appear less practical as adding a node need the ring to be open, contrary to Ethernet switches that may have available ports the new node can be connected to. But, if not, a complete switch must be changed or added. On the contrary, the ring can be automatically and quickly upgraded without any change of the existing equipment, as shown with our prototype: a ring opening is detected through a loss of frame signal by the master node in less than the RTT duration. When a frame is detected again after insertion of a new node and ring close up, the master node automatically restarts a fast auto-discovery (network length, number of node, …) and auto-reconfiguration (slot re-sizing), as described in Section III.A. For our prototype, this plug-and-play operation only lasts 4.6 µs, whereupon, the traffic automatically resumes in the ring. In parallel, the master node reports to the SDN controller the new network configuration including identification of the new node that this one sends via the control channel. Reporting to the SDN controller uses extended Open Flow protocol, and an ONOS graphic user interface displays the new network topology and characteristics (Fig. 2, left).

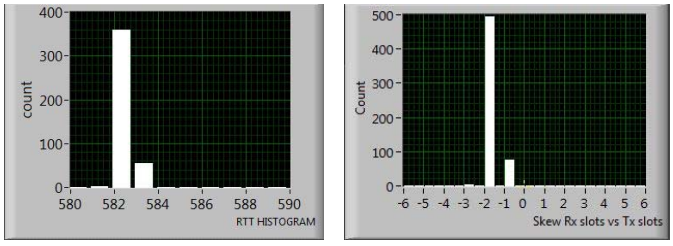

Fig. 7. left: round trip time measurement statistics, right: slot skew measurement between slot emission and reception at Master node.

D. SDN operations

Demonstration of scheduling and data channel capacity reconfiguration, plus data flow bandwidth measurement are done using the following scenario (Fig. 2): two data flows are established, a guaranteed capacity traffic flow sent from the Master and a best effort flow (no guarantee) from Slave 1. Both have Slave 2 for common destination. A remote Android application is used by the end-user to change on-demand and on-the-fly the capacity allocated to the guaranteed flow between 1 and 10 Gb/s. This information is sent to the SDN controller that converts the reserved capacity into slot reservations that are then transmitted to the master node, which automatically redistributes this scheduling information to all other nodes through the control channel. We measured a duration of 3.1 µs on our setup for local database of all nodes to be reconfigured.

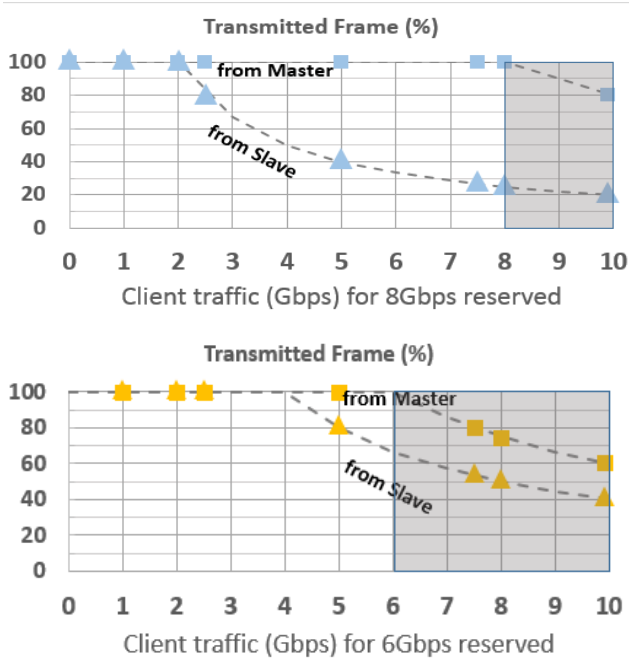

Fig. 8. Data flow bandwidth measurement for two cases of allocation for the Master priority node: (top) 8Gbps and (down) 6Gbps. Dash lines: expected variations.

Then, during traffic transmission, the master node indexes outgoing control slots cyclically with a 10 slots window and writes this index onto dedicated bit field. Every slave node controller compares this slot index with its local database and controls add/drop operations as explained in Section II.B. Using an FPGA-embedded Integrated Logic Analyzer (ILA), we verify at each node that frames of the two traffic flows only use their own reserved dedicated slots (Fig. 7). Data bandwidth utilization measurement is performed using a 10Gb traffic analyzer that generates the two flows using 1534 Bytes length of 10Gbps frames that are regularly spaced to generate a client traffic request of multiples of 1Gbps from 1 to 10 Gbps. The slot size is adjusted to fit the 1534 Bytes encoded frames to prevent bandwidth loss, and the 10Gbps net data rate is assured by using 35% overrating to account for the 25% overhead of 8b10b encoding, plus the data framing and 100ns guard band between slots. These data flows are inserted at the client side of the Master and Slave_1 nodes and both received at the Slave_2 that retransmit them to the traffic analyzer that performs the measurements on received frames. Fig. 8 shows for two cases of reserved and guaranteed bandwidth allocated for the Master node transmission (8 and 6 Gbps) the measured bandwidth expressed in terms of frame percentage that are correctly received at Slave_2 destination. Compared with expected results (dashed lines) given by 1-N/T, where T is the client traffic request and N the number of reserved slots for this traffic, we show that the capacity allocation downloaded from the SDN controller is correctly considered inside the network, with less than 0.5% between the corresponding expected data bandwidth and the measured one. This low error rate may be due to the above-mentioned overrating set without margin. Taking this into account, we see for both cases that frames sent by the Master node reserved and guaranteed capacity are received without loss at Slave_2 destination (as long as the client request does not exceed this reserved capacity).

IV. Conclusion

Data centers will have to provide dynamic reconfiguration capacity, scalability, latency and QoS guarantees in order to support emerging services. To meet these requirements, we presented operation and key functions of an SDN based optical slotted ring topology. In particular, we showed for the first time through results from an experimental setup, that latency can be controlled within less than a few of ns (6.4 ns here) and that each node adds a latency as low as 204 ns. We demonstrated the network update with a node addition and ring reconfiguration as fast as several µs (4.6 µs here). Also, we presented an implementation of the interaction between the SDN controller and the network nodes through the control plane, and demonstrated that SDN control functions can be reprogrammed in less than several µs (4.6 µs here). These experiments consolidate the feasibility of the SDN operation in datacenter using slotted optical packet switching technology.

[1] M. A. Taubenblatt, “Optical interconnect for High-Performance Computing”, J. of Ligthwave Tech., 30-4, Feb. 2012.

[2] F. Po Tso, S. Jouet, D. Pezaros, “Network and server resource management strategies for data centre infrastructures: A survey”, Computer Networks, vol. 106, pp.209-225, 2016.

[3] D. Chiaroni et al, “Packet OADMs for the next generation of ring networks”, Bell Labs Tech. J., vol. 14, no. 4, pp. 265–283, Winter 2010.

[4] V. Kamchevska et al, “Experimental demonstration of multidimensional switching nodes for all-optical data center networks”, in European Conf. on Optical Communication (ECOC), Cannes, France, 2015.

[5] Y. Pointurier et al, ‘Green optical slot switching torus for mega-datacenters’, ECOC 2015.

[6] V. Kamchevska et al, “Synchronization in a random length ring network for SDN-controlled optical TDM switching”, J. Opt. Commun. Netw., vol. 9-1, 2017.

[7] ONOS consortium, http://onosproject.org