Real-Time Node Local Control for Ultra-Dynamic and Deterministic All-Optical Intra Data Center Networks

Mijail Szczerban, José Estarán, Nihel Benzaoui, Haïk Mardoyan, and Yvan Pointurier
Nokia Bell Labs, 7 Route de Villejust, 91620 Nozay, France, mijail.szczerban_gonzalez@nokia.com

Abstract: We enable ultra-dynamic features in scheduled optical data centers through a novel control mechanism local to each node. We experimentally show sub-µs resource allocation, at least halving distributed computing application completion time. © 2020 Nokia Bell Labs
OCIS codes: (060.1155) All-optical networks; (060.4250) Networks; (060.4510) Optical communications

1. Introduction

Today's internet is based on the best effort approach, where the network does the best it can to deliver a message, but nothing is guaranteed, thus, frames can be strongly delayed or even lost. Some applications cannot withstand this uncertainty: 5G front-haul, industry 4.0, high-frequency trading and telesurgery are examples of applications that can be affected by latency variability [1-3]. Most demanding time-critical applications typically expect tens to hundreds of µs latency, sub-µs latency standard deviation (jitter), and sub ms service turn-up. Deterministic and Dynamic Network (*DDN*) raises a future-proof solution guaranteeing performance while being dynamic enough to allow seamless service deployment and high efficiency [4], consisting in two main characteristics: (a) contentionless data plane allowing flow-granular resource allocation, and (b) real-time control plane. In [4], we have shown that *DDN* brings time-wise determinism thanks to per-flow end-to-end network slicing and jitter compensation mechanism, outperforming state-of-the-art technologies. However, deterministic performance was based in strict resource scheduling, thus, transmission was allowed only to flows reserving resources in advance, then limiting network's dynamicity and efficiency. In [5] we explored the dynamicity of *DDN* in a data center context, i.e. the ability to react rapidly, using real-time central network controller, allowing for networkwide responsiveness in tens of µs. Other publications have used non-real time software-defined network to control optical data center networks, but latency and reconfiguration time over 100s µs are well above our target [6-7]. This paper focuses on the maximum expression of dynamicity, using decision-making nodes and supporting opportunistic traffic in *DDN*. We propose, implement and evaluate for the first time a real-time node-local control with 10s ns responsiveness.

2. Real-time network control in DDN

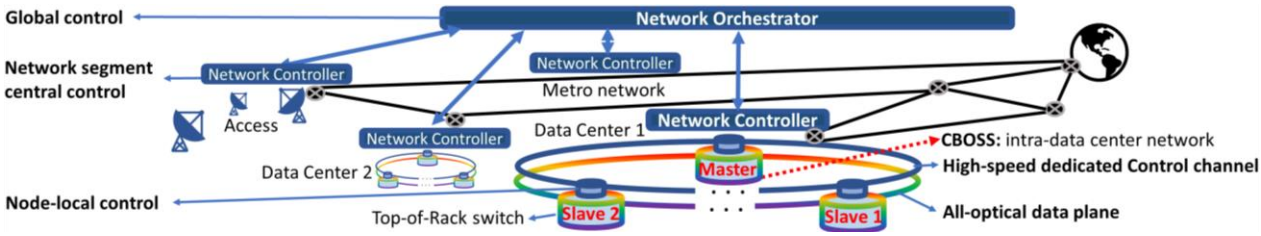

Fig. 1: Multi-segment network real-time control plane with three decision- making levels (global, network segment central and node-local). CBOSS ring network as intra data center network segment technology.

Fig. 1 shows the schematic of real-time control plane network architecture which comprises three main layers. It aims at minimizing control and management delay. Therefore, a hierarchical split is performed to avoid unnecessary centralization and keeping control decisions local when possible [5]. The top layer is the global control layer, embodied by a network orchestrator, which oversees the deployment of inter-network segment services (i.e. inter data center). A real-time network central segment control layer present at each network segment is the most powerful control element in our control architecture; it monitors network segment's state and schedules transmission resources in real time through the dedicated optical control channel guaranteeing the transmission of monitoring data and control instructions to/from the real-time controller [5]. The reactivity of this control layer is on the 10's of µs for few km networks. The node local control layer is in charge of taking decisions on resources that are not reserved by the segment central network controller, requiring a responsiveness of ~10s ns. In this paper we present for the first time a node-local control able to take decisions autonomously through an ultra-fast decision-making mechanism and resource utilization flags delivered via the dedicated control channel. We leveraged Cloud-Burst Optical Slot Switching (CBOSS) [8], which is an all-optical slot switching intra-data center network to evaluate the real-time control concept. CBOSS relies on a transparent data plane avoiding any packet contention in intermediate nodes, and leverages wavelength (λ) and high-granular time-division multiplexing (~µs optical slots) which, in combination

with dedicated queues and interfaces, enable per flow network slicing. Concerning the control plane, CBOSS features an opaque optical control channel, which is systematically dropped, processed and retransmitted, see Fig. 2(a), providing a guaranteed path to transmit control and management information including routing instructions, monitoring data, network's synchronization and transmission scheduling. The schedule informs the flow (λ and queue) to be served at each time slot. The testbed used on the following experiments consists of a 3-node real-time CBOSS ring prototype [8]: one Master linked to the network controller and two Slaves. Each node equipped with a fixed-λ optical 10G receiver, and for the data plane a fast-tunable (ns scale) WDM 10G transmitter, using arrays of C-band DWDM SFP+ and SOAs as gates. The total ring length is ~3.5km of SMF, with equally split inter-node links. Fig. 2(a), shows the node scheme, including optical paths of control and data channels, the FPGA-node controller and the WDM transmitter. On top of Fig. 2(a) shows an example of the optical data plane at the output of the coupler, during a 10 time-slot periodic reservation. Note that some slots can be empty (no power) and others can have more than one λ inserted.

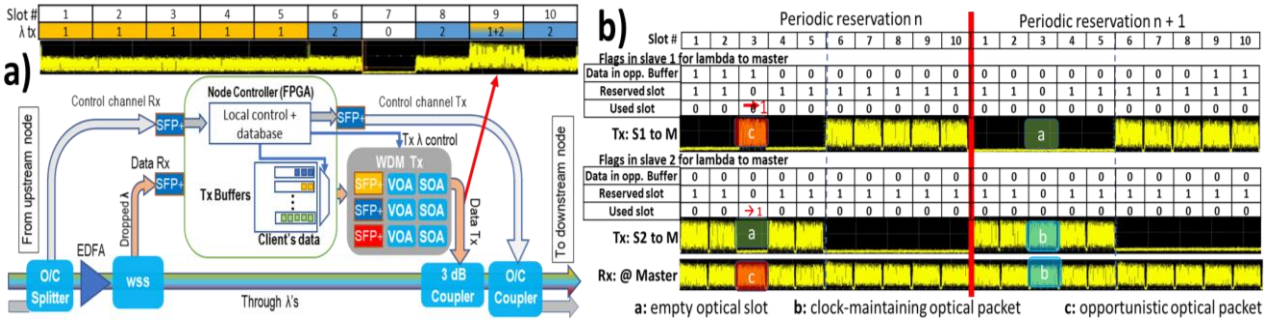

Fig. 2: a) CBOSS node architecture and scheduled optical transmission example, b) Local-control usage flags and transmission example

3. Novel local real-time control mechanism

Each node is composed of an FPGA-based node controller Fig. 2(a), that manages all node elements of both control and data plane. This includes optical switching, client interfacing, data queues, medium-access control as well as reception and retransmission of control channel. In previous implementations, the node controller was a functionally passive element that only received and implemented instructions coming from the network controller. To create the node local control layer, the node controller has become a decision-making element. The central controller still reserves resources network-wide, but idle resources are managed by the node local controller. To enable the real-time node-local decision mechanism, a signaling system, Fig. 2(b), has been devised whereby each λ usage state is informed through flags in the control channel at the beginning of each time-slot indicating whether the time-slot is reserved by the network controller and, if not, whether it has been used by another node to transmit opportunistically. The node also monitors client's data buffers to assess if there are frames to be transmitted. These flags and information are processed in few tens of nanoseconds since the decision on the λ and flow insertion at the incoming time-slot needs be made in negligible time with respect to the time slot duration (< 1.5 μs).

Real-time node-local control mechanism enables two features relevant for CBOSS network, requiring local high-speed control and not feasible with the fully scheduled (central control) version: first, opportunistic traffic insertion to use idle optical transmission resources (unused time slots) and, second, the insertion of clock-maintaining optical slots when a λ to be dropped in the next node is carrying unused or "empty time slots" in order to avoid loss of data at the receiver due to Clock and Data recovery (CDR) constraints which after long periods (few μs) without receiving optical data can lose track of the clock. Fig. 2(b), shows the node-local control mechanism in action. In this case, the third time slot of the 10-slot schedule window is not reserved and left for opportunistic traffic insertion. In the first reservation window (left side), opportunistic traffic insertion mechanism was used, local controller of Slave 1 detected opportunistic data stored in transmission buffer (local monitoring) and an un-used and non-reserved slot was in transit (control channel flags), thus, the third slot (indicated with the letter "c" in the figure) was used for opportunistic transmission. Slave 2 detects that this slot was used by a previous node (Slave 1) and did not make any insertion of this λ during this time slot (letter "a" in the figure). In the following reservation window, right side of Fig. 2(b), Slave 1 had no information at opportunistic queues, so the slot remains empty ("a"), nevertheless, Slave 2 detects that the time slot in the λ to be dropped at the next node is empty and inserts a clock-maintaining optical packet ("b"). Lower part of Fig. 2(b), shows the reception at Master from both Slaves using node-local control, no empty nor overlapped slots are observed at reception, if the clock-maintaining optical packet was not inserted, the receiver would have experienced absence of signal when there is no opportunistic data inserted (e.g. time slot "b") . Note that the central controller is not aware of these local decisions, thus, control plane communication delay is avoided. The decision on the insertion (λ and queue) is taken in 3 clock cycles (19.2 ns)

4. System evaluation and results

To test the novel node local control layer, we ran real-time distributed operations over the network: operands were sent from an FPGA-based client at Master node to FPGA servers at Slave nodes, see Fig. 3(a). These experiments evaluate flow establishment time, with only 1/10 un-reserved slot shared to be used opportunistically (opportunistic capacity) by both slaves to send results, while 9/10 of network's transmission resources were reserved for other deterministic flows. Once the client at Master retrieves operation results from Slaves, it sends the next operands. Figures 3(b) and 3(c), show the time taken by the network and computing servers to complete 2000 operations. Three scenarios were tested, (i) all transmissions were pre-reserved (static benchmark), (ii) dynamic reservation of resources (central network controller) and (iii) opportunistic use of slots (real-time node local control).

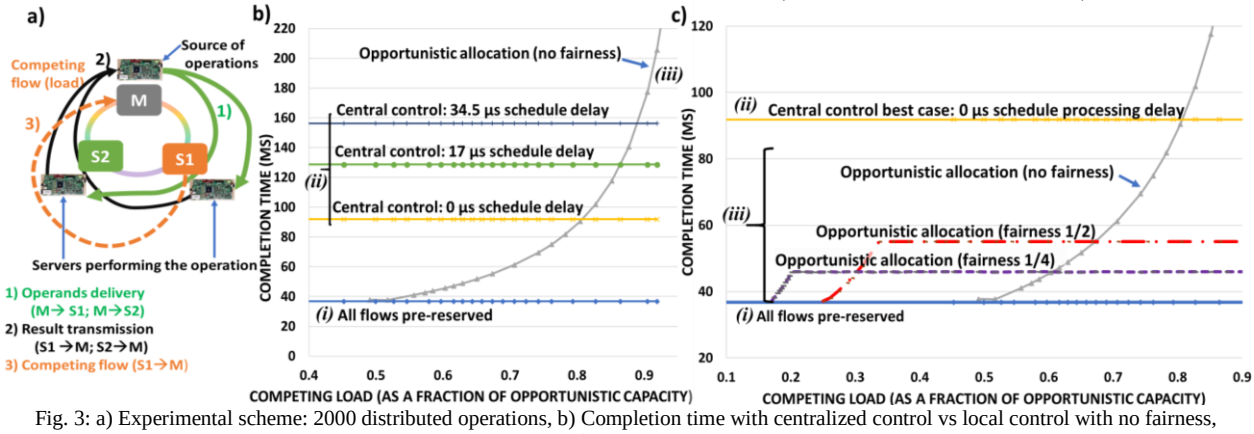

Fig. 3: a) Experimental scheme: 2000 distributed operations, b) Completion time with centralized control vs local control with no fairness, c) Completion time local control with fairness mechanism vs other control mechanisms.

In case (i) and (ii), Fig. 3(b), the completion time is constant when competing load varies since these approaches reserve slots for the transmission, performance was guaranteed. Case (ii) had larger latency than (i) since it required systematic delivery of monitoring data from nodes, central decision and instruction distribution to establish communication. Performance in case (ii) is also dependent on the schedule computation complexity, Fig. 3(b), shows an increase on completion time when schedule processing time increases, we emulated schedule processing time through a fixed delay before the controller delivers new resource allocation. Opportunistic resource allocation case (iii), enabled by our node local control layer, outperforms central control at low load since it allows to insert frames containing results as soon as an unused slot is available (which is likely in low load scenario). However, when there are competing opportunistic flows, and no starvation control or fairness mechanism is applied, the system takes longer to complete the operations as competing load increases since opportunistic slots are taken by the competing flow, leaving fewer opportunities for the intermediate node to transmit results. Fig. 3(c) shows in dashed lines the performance when enforcing fairness to opportunistic insertion, limiting to 1 opportunistic slot every 2 (or 4) available to be used by any opportunistic flow. This simple approach allows to guarantee bounded and low latency even when using node local control approach. In *DDN* context this is critical since we show that node local control mechanism not only supports best effort traffic but can also be used for time-sensitive traffic.

5. Conclusions

We proved that determinism and high-speed dynamism can coexist in a network by implementing the node local control layer in a functional intra-data center optical slotted network. Node local control increases overall network dynamicity when comparing with centralized network control, it allowed the opportunistic transmission of optical packets in the network with bounded latency and, with the insertion of clock-maintaining optical packets in idle time slots, it proved how a highly-reactive control plane can help to reduce physical layer's requirements. Node local decisions were made and implemented in less than 50 ns without intervention of the central network controller.

6. References

[1] Y. Pointurier, et al.: "End-to-end Time Sensitive Optical Networking: Challenges and Solutions", IEEE/OSA JOCN 2019
[2] W. A. Khan, et al. "Analysis of the requirements for offering industry 4.0 applications as a cloud service," in PISIE, 2017, pp. 1181–1188.
[3] A. Nasrallah, et al.:'Ultra-Low Latency (ULL) Networks: The IEEE TSN and IETF DetNet Standards and Related 5G ULL Research',2018,
[4] N. Benzaoui, et al.: 'DDN: Deterministic Dynamic Network', 2018 ECOC, Rome, 2018, pp. 1-3. DOI: 10.1109/ECOC.2018.8535191
[5] M. Szczerban, et al., "Real-time control for Deterministic Dynamic Networks" ECOC 2019, Dublin, pp. 1-4.
[6] P. Bakopoulos et al., "NEPHELE: An End-to-End Scalable and Dynamically Reconfigurable Optical Architecture for Application-Aware SDN Cloud Data Centers," in IEEE Communications Magazine, vol. 56, no. 2, pp. 178-188, Feb. 2018. DOI: 10.1109/MCOM.2018.1600804
[7] K. Kontodimas, et al. : "Resource allocation in slotted optical data center networks," 2018 International Conference on Optical Network Design and Modeling (ONDM), Dublin, 2018, pp. 248-253. DOI: 10.23919/ONDM.2018.8396140
[8] N. Benzaoui, et al.: "CBOSS: bringing traffic engineering inside data center networks," in JOCN, vol. 10, no. 7, pp.117-125, July 2018.