Switching Functions of a Data Center Top-of-Rack (ToR)

Ioannis Patronas, Angelos Kyriakos, Dionysios Reisis
Electronics Lab, Physics Dpt, National and Kapodistrian Univ. of Athens,
Panepistimiopolis, Physics Bld IV, V, 15784 Athens, Greece,
email: {johnpat, akyriakos, dreisis}@phys.uoa.gr

***Abstract*—In Data Centers each rack uses a Top-of-Rack (ToR) as a first level switch to connect servers to the aggregation switches. The current paper focuses on a hybrid electrical/optical ToR design, which first, adapts the servers Ethernet traffic to the optical TDMA operation of the core network for supporting optical switching in the Data Center's upper layer and second, it employs optical switching of traffic at the rack level. The proposed ToR architecture is based on an Ethernet switch and FPGA port extensions realizing the required functions to support 20 10Gbps connections, exploit the network routing resources and handle effectively virtual queues.**

I. Introduction

Data Centers constitute an integration point of information technology (IT) activities and the supporting devices for telecommunications, computing and data storage [1]–[3], [6]. Moreover, they offer an environment for private cloud computing and big data applications [1]–[3], [5], [6]. Architectures realizing data centers need to process large volumes of data in real-time, maintain high throughput communication among their subsystems and at the same time optimize blocking probability and latency. Design approaches include electrical or optical high throughput networks with hierarchical or distributed or even combined organizations scalable with respect to the number of servers and the bandwidth; switching subsystems improving bandwidth utilization, blocking probability and latency [1], [3], [7]; elements with low reconfiguration delay [9]; high-speed optical interconnects [10] and flexible control [5], [6].

Among the approaches for the design of data centers, which target the dynamic and efficient sharing of the optical resources as well as a collision-free network operation, are those operating in a slotted TDMA manner [1] and a software-defined-network (SDN) based control plane [5], [6]. In this case the role of the Top-of-Rack switches is upgraded because they are assigned with the tasks first, to convert the Ethernet traffic into TDMA traffic and second, to manage Virtual Output Queues (VOQs) for alleviating the head-of-line effect.

Aiming at a design supporting the above tasks the current paper presents a 20×20 10Gbps SFP+ ports ToR switch that operates in a data center with slotted hybrid electrical/optical interconnect which enables the dynamic and efficient temporal, spatial and wavelength allocation of resources. The data center network is divided into pods of racks, which contain the zones of disaggregated computing, storage and memory resources. The zones are connected to ToRs and through those, they can (all-optically or electro-optically) be connected to any other zone. The proposed ToR switch is advantageous for this type of network as it introduces the combination of the following features: a) in contrast with a typical $I \times I$ Ethernet switch, there is not a fixed association between output ports and MAC addresses a fact allowing the exploitation of multiple routing resources; b) it dynamically assigns outputs to frames based not only on the MAC addresses of destination ToRs but also the assigned plane/laser and slot; c) the architecture accomplishes the aforementioned functions by including a 16×16 Ethernet switch [7] with extensions at its $North$ side (network side) and $South$ side (servers side) that realize all the required functions as well as a 4×4 extension module connecting the servers directly to the optical switch; for the extensions we are using Xilinx NetFPGA SUME.

The paper is organized with Section II highlighting the network architecture. Section III presents the overall ToR architecture. Sections IV and V describe the south and north extensions respectively. Section VI shows the FPGA implementation reports and Section VII concludes the paper.

II. Data Center Network

The overall system topology of the network is depicted in Fig. 1. There are I parallel planes with each plane consisting of R unidirectional rings connecting P pods; in the current design $I = R = P = 20$. A pod includes R Wavelength Selective Switches (WSS) to connect to the R rings, $W = 80$ pod-switches (one pod-switch per ToR); moreover, it is connected to W ToR switches. Each ToR switch has I north ports. Each north port is directed to one pod-switch, which belongs to the ith pod of each plane (the ToR faces the ith pod of all the planes): each north port is connected to a different pod-switch with a tunable wavelength transmitter and a burst mode receiver. The south ports are connected to the servers through network interface cards (NICs).

The network uses WDM technology. Each of the R fiber rings of the I planes carries WDM traffic comprising W wavelengths propagating simultaneously in each ring in the same direction (unidirectional). In the optical links from a ToR to a pod-switch each fiber carries a single wavelength at each time instance. The wavelengths can change at different time instances and the traffic is multiplexed in the time domain using TDMA slots. In the ToR to pod-switch links, the wavelength assignment is performed dynamically per TDMA slot in the

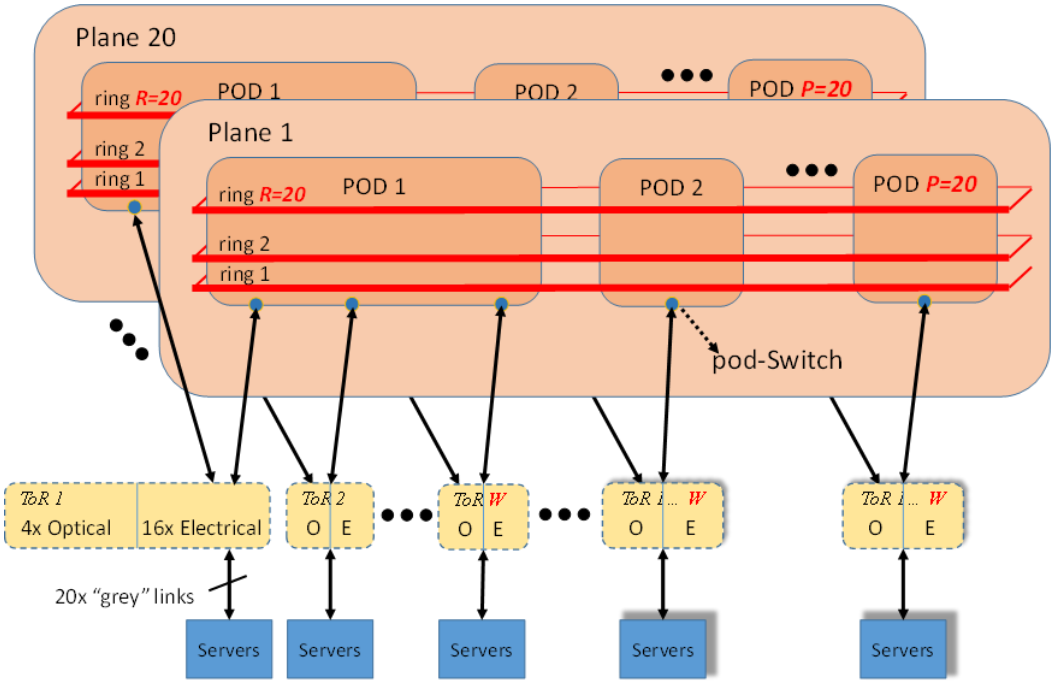


Fig. 1. Data Center Network Architecture

upstream direction depending on the system's decisions. At the lower network level, the connections between the servers and the ToR are independent of wavelength and bidirectional. The routing information is encoded in the wavelength domain.

Among the novelties of the network is the communication scheme. There are two major scenarios for a ToR to ToR communication. In the first case the source and destination ToRs belong to the same pod (intra-pod communication) and in the second to different pods (inter-pod). In the first case, the control plane informs the source ToR regrading the wavelength used by the destination ToR receiver and the timeslot for transmission. The optical signal transmitted in that slot reaches the specific plane pod-switch, which through a fast space switch dedicated to each ToR and an Arrayed Waveguide Grating (AWG) forwards the signal to the destination ToR (belonging to the same pod). In the second case, the control instructs the source ToR what timeslot and wavelength to use for transmitting to the specific destination ToR. The optical signal reaches the specified (by the control) plane pod-switch, at which the fast space switch (dedicated to the source ToR) forwards the optical signal to the cyclic $W \times R$ AWG; based on the incoming port and the wavelength the signal reaches a specific fiber ring to travel through pod-switches. Pod-switches use the 1×2 Wavelength Selective Switch (WSS) for each incoming ring and according to wavelength they either forward the signal on the ring or drop it to the ToRs; the WSS of the pod that includes the destination ToR drops the signal and through an $R \times 1$ combiner and the AWG, based on the wavelength, routes the signal to the destination ToR.

III. ToR Architecture

The overall 20×20 ToR architecture is depicted in Fig. 2. It includes a legacy 16×16 Ethernet switch with a north extender component with 16 ports towards the pod-switches,

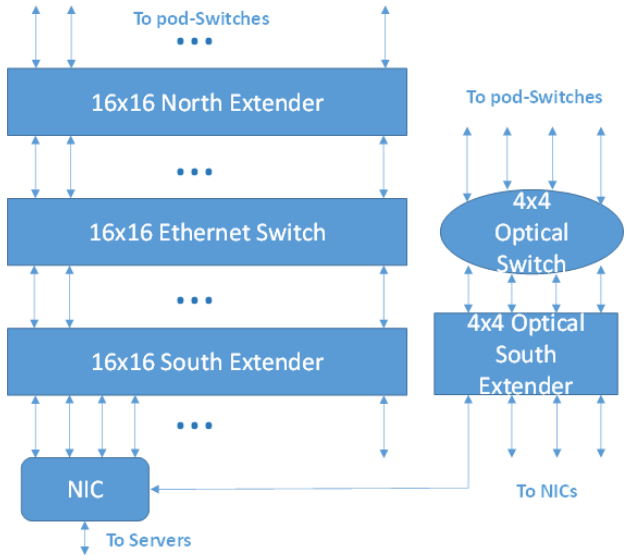


Fig. 2. ToR Switch

a south extender with 16 Ethernet ports towards the servers (NICs). An additional extender module has 4 south ports to the NICs and 4 north ports to the optical switches to connect the NICs directly to the optical switches and consequently, it is called the optical south extender. The 20 south interfaces to the NICs are standard (legacy) 10GbE Ethernet ports while the 20 north are all-optical TDMA ports. The main task of the ToR is to organize a queue per destination ToR, that is for the proposed network the number of possible queues is 1600. The majority of the required functions is related to the upstream flow and it is included in the the south extenders. The following section describes the south extenders functionality and the corresponding architectures.

IV. South extension

The south extender realizes a set of functions that belong to two different and independent data flows: the upstream and the downstream. The upstream flow major functions are a) the reception of the Ethernet frames sent from the NICs and their formation into payload of TDMA frames, b) the implementation of the scheduling commands when it transmits in the upstream (north) direction and c) the realization of the Virtual Output Queues (VOQs). The downstream flow has to receive the TDMA frames, extract the Ethernet frames and forward these to the corresponding NICs.

A. Upstream Data Flow Architecture

The architecture of the south extender's upstream data flow is depicted in Fig. 3. The design involves a memory (buffer) shared by all the 16 upstream flows, a *write FSM*, a *read FSM*, a queue management based on a memory map organization and a scheduling command interpreter. First, we describe the memory organization along with the queue management organization and then, starting from the south input to the north output the blocks that accomplish the major tasks of the upstream direction in the south extender.

1) Shared Memory & Queue management: The major memory modules in the upstream flow architecture are three. First is the shared memory that stores the destination queues. It is divided into pages of 228 KB, with each page containing tha data volume of each TDMA frame's payload. Each page is filled with Ethernet frames. A queue is realized as a linked list of pages, with the *next pointer* (pointer to the next page) located in the page. The current implementation of the shared memory allows for distinct burst write and burst read operations. Note here that, we have designed the shared memory so that in the time that equals to the TDMA slot, it is able to accomplish a burst write operation of a page for every south input port and a burst read operation of a page for every north output port.

The second memory module is a SRAM, called *memory map*, which stores the information that is necessary for the queue manager operations. It has entries equal to the maximum number of queues. Each entry stores: a) the initial address (first page or head pointer) of the first page of the queue, b) the starting address of the last page (tail pointer) and c) a pointer showing the location that the next Ethernet frame (*next Ethernet frame pointer*) will be written within the last page.

The third memory is an *unused pages FIFO* that stores the addresses of the pages that are unused. Each time a page is transmitted in the upstream direction, we free its memory space, which will be used for gathering another flow of Ethernet frames and consequently, the starting address of the free space is forwarded and written into this FIFO.

The following paragraphs give the description and functions of the upstream blocks that perform operations.

2) Input: Each south port uses a SFP+ transceiver to connect the ToR to the corresponding NIC. A 10GbE MAC manages the reception of the upstream Ethernet frames and stores them in a queue (FIFO); a LUT provides an identification key called the *ToR-tag* to be used instead of their destination MAC address within the ToR. All the input FIFOs inform the following block upon an Ethernet frame reception.

3) Write FSM: The FSM polls the input FIFOs and gets an Ethernet frame from each input port. It uses the frame's *ToR-tag* to identify the page that the frame must be written. It computes the remaining free space in the page by using the *next Ethernet frame pointer* address and the page's final address and checks if the current frame fits into that free space. If it does not fit it will consider the page filled and it will create a new page along with the necessary pointer operations.

4) Read FSM: The Read FSM operates according to the scheduler commands that describe: a) the queue's *ToR-tag* that will send a page to the north extender and b) the exact time for this transmission. The Read FSM identifies the first page in the queue of the specified *ToR-tag*, transmits the page, updates the head pointer of that queue in the *memory map* and finally, writes the address of the freed page in the *unused pages FIFO*.

5) Memory Lock Mechanism: The shared memory organization optimizes its data transfer performance when it operates either in burst read or burst write modes: a *lock mechanism* allows alternative read and write burst operations if there are such requests by the read and write FSMs. The write requests made by the south ports to the write FSM are serviced in a round robin fashion; the same holds for the read requests made by the scheduler's command interpreter for each north port.

6) Virtual Output Queues Realization: The VOQ mechanisms of the ToR are implemented entirely in the south extender. The shared memory organization and the dynamic queue configuration allow the VOQ effective realization under all circumstances. Moreover, in the case that speed-up is required, that is more than one pages from the same queue have to be transmitted in the same slot, the read FSM will read contiguous pages from the respective queue and it will forward these pages to the outputs specified by the scheduler.

B. Downstream Data Flow

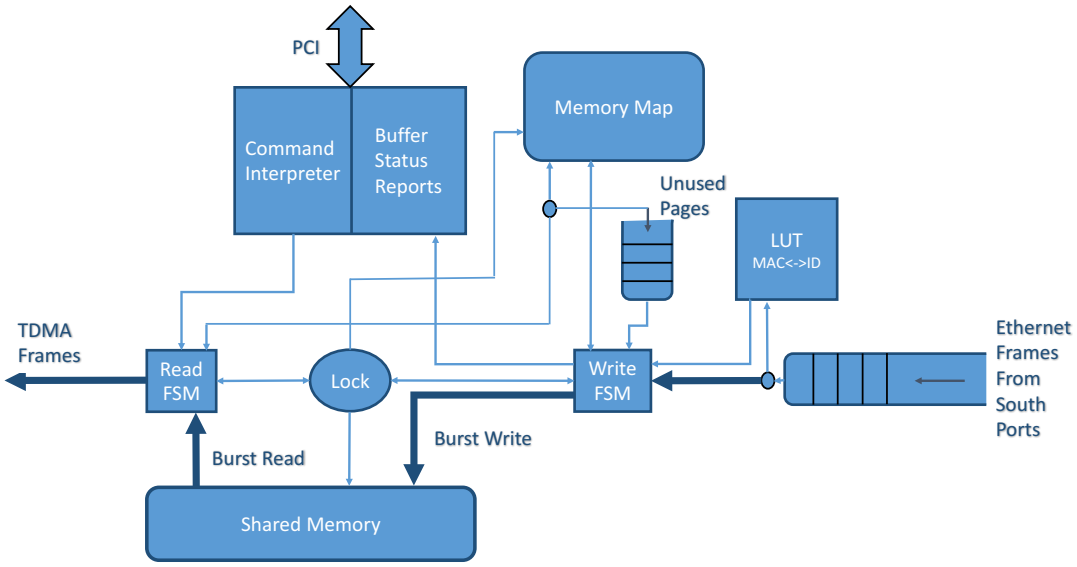
The downstream design is straightforward: each north input port receives a burst of Ethernet frames from the switch, which were extracted from a TDMA frame and they are directly forwarded to the south output port that corresponds to the north input.

C. Optical South Extender

The optical south extender module has four 10GbE south ports to interface the NICs and four optical TDMA ports to the optical switches. The functionality and the architecture of this module is the same as the south extender with the addition of the functionality of the TDMA ports that are the same as those of the north extender, the description of which is presented in the next section.

V. North extension

The north extender operation in the upstream direction includes first, the reception of the pages transmitted by the



+

Fig. 3. South Extender Upstream Flow Architecture

TABLE I
FPGA UTILIZATION IN THE XILINX NETFPGA

Implementation Results	Slice LUTs	Slices	Slice Registers	Operating Frequency MHz	RAM Blocks
Ethernet Interfaces	19624	7168	21584	156.25	72
Memory Organization	16563	6706	22908	200	17
PCIBlock	8679	3264	10973	250	22

south extender through the switch. Second, the encapsulation of these pages, which contain Ethernet frames, into TDMA frames with the addition of the required: preamble, which for the experiments was set to 20 μsec and the delimiter of 64 bits. Third, is the transmission of each TDMA frame by using the wavelength mandated by the scheduler.

In the downstream direction it simply extracts the Ethernet frames from the TDMA frame and it forwards these to the south extender through the Ethernet switch [7].

VI. FPGA IMPLEMENTATION

An example implementation for validating the design involves a 2×2 version of the south extender and a simple cut-through north extender. The design implementation targets the Xilinx NetFPGA SUME evaluation board because it includes a DRAM that we use as a shared buffer, an SRAM for the memory map and four 10Gbps transceivers that are used for the 2×2 configuration. The results for the south extender FPGA utilization are shown in the Table I. For the transceivers we use the Xilinx GTH connected to SFP+ optical transceivers to achieve the 10 Gbps for both sides of each extender. For the Ethernet connections we use the Xilnx 10GbE Subsystem which includes the MAC and the 10GbE PCS/PMA. The PCIe is implemented with the PCIe Xilinx core and the Riffa [8].

VII. CONCLUSION

The present paper introduced an approach for the design of a ToR Switch for data center with TDMA optical communication and SDN control plane. The approach results in a modular architecture based on an Ethernet switch and extenders that realize all the required additional functionality for the TDMA optical switching and to allow the exploitation of the network's multiple routing resources and the effective handling of VOQs.

ACKNOWLEDGEMENT

This work has been funded by the European Unions Horizon 2020 research and innovation programme under grant agreement No 645212 (NEPHELE).

REFERENCES

[1] B. C. Vattikonda, G. Porter, A. Vahdat, and A. C. Snoeren, "Practical tdma for datacenter ethernet," in *Proceedings of the 7th ACM European Conference on Computer Systems*, ser. EuroSys '12, 2012, pp. 225–238.
[2] N. Farrington, E. Rubow, and A. Vahdat, "Data center switch architecture in the age of merchant silicon," in *Proceedings of the 2009 17th IEEE Symposium on High Performance Interconnects*, ser. HOTI '09, 2009, pp. 93–102.
[3] A. Vahdat, M. Al-Fares, N. Farrington, R. N. Mysore, G. Porter, and S. Radhakrishnan, "Scale-out networking in the data center," *IEEE Micro*, vol. 30, no. 4, pp. 29–41, July 2010.
[4] N. Zilberman, Y. Audzevich, G. A. Covington, and A. W. Moore, "Netfpga sume: Toward 100 gbps as research commodity," *IEEE Micro*, vol. 34, no. 5, pp. 32–41, Sept 2014.
[5] B. Hindman, A. Konwinski, M. Zaharia, A. Ghodsi, A. D. Joseph, R. Katz, S. Shenker, and I. Stoica, "Mesos: A platform for fine-grained resource sharing in the data center," in *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI'11, 2011, pp. 295–308.
[6] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, "Data center tcp (dctcp)," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. –, Aug. 2010.
[7] Mellanox. Switch system pb sx1024.
[8] Reusable integration framework for fpga accelerators. [Online]. Available: riffa.ucsd.edu
[9] K. Numata, J. R. Chen, and S. T. Wu, "Precision and fast wavelength tuning of a dynamically phase-locked widely-tunable laser," *Opt. Express*, vol. 20, no. 13, pp. 14 234–14 243, Jun 2012.
[10] G. Porter, R. Strong, N. Farrington, A. Forencich, P. Chen-Sun, T. Rosing, Y. Fainman, G. Papen, and A. Vahdat, "Integrating microsecond circuit switching into the data center," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 447–458, Aug. 2013.