Software-Defined Optical Intra-rack Network Architecture and MAC Protocol for Data Centers

Peristera Baziana
Department of Computer Science and Telecommunications
University of Thessaly
Lamia, Greece
baziana@uth.gr

***Abstract*—The optical interconnection and switching technology has recently gained ground in the data centers (DCs) globally. In this paper, we propose an optical network configuration to interconnect the servers of the same rack in a DC, based on passive optical functionalities. Thus, we focus in the intra-rack passive optical network and we propose the use of a number of dedicated wavelengths for the data communication among the servers through a passive optical coupler in the data plane. Meanwhile, we adopt a parallel control plane which is implemented in the optical domain too, in opposition to the majority of previous studies that performs the control communication over electrical links. Based on the software-defined networking (SDN) paradigm, we adopt at the Top-of-Rack (ToR) switch level, a rack controller which is responsible for the operational functionalities in the intra-rack network, such as medium access control (MAC) etc. Moreover, we introduce an intra-rack MAC protocol in conjunction with a bandwidth efficient allocation (BEA) algorithm to optimally and fairly assign to the servers the available intra-rack bandwidth for synchronous data transmission. According to the BEA algorithm, both the channels collisions and the destination conflicts are effectively avoided. The proposed intra-rack MAC protocol in conjunction with the BEA algorithm achieves high intra-rack network bandwidth exploitation that reaches almost 88%, while it manages high throughput and very low packet delay, as compared with previous relative study.**

Keywords—data center, rack controller, optical intra-rack network, software-defined networking.

I. Introduction

The recent big data industry explosion has appeared as one of the results of the great research interest about large scale data centers (DCs) globally. The internet applications popularity has dramatically increased, involving DC infrastructures to their service [1]. The diverse internet users applications most usually require immediate responses from DCs, fact that is of high research interest. On the same time, the traditional DC networks (DCNs) are based on electrical Ethernet interconnection and switching technology which causes significant power consumption problems [2].

On the other side, the optical switching technology arises as the most promising solution in order to reach high data rates in the contemporary DCNs. Especially, in [3-5] the optical circuit switching (OCS) technology has been proposed for the DCN infrastructure. Nevertheless, the OCS paradigm suffers from the disadvantages of the circuit switching implementations, such as the late response at possible network reconfigurations and for this reason it is not recommended for DCNs with high traffic variations. In addition, the optical packet switching (OPS) technology has been introduced to studies like [6-7]. However, these proposals are highly power consuming and for this reason they can hardly be followed in commercial solutions.

Moreover, the software-defined network (SDN) paradigm has been proposed for optical DCN environments [8-9]. The main research interest in this area is the substitution of the conventional electrical switching fabric in the inter-rack DCN level by optical switching fabric. Although the significant performance results for the inter-rack DCN, the intra-rack communication remains in the electrical domain. Since the power consumption in Top-of-Rack (ToR) electrical switches in DCNs is high enough [10], the optical interconnection in the rack level appears to be a key factor for the efficient DCNs solutions definition.

In [11], an optical intra-rack DCN infrastructure has been introduced which additionally occupies a rack controller to implement the SDN functionalities. In [11], and in the relative ones [12-13], a synchronous transmission software-defined MAC (SD-MAC) protocol is introduced which performs servers grouping to coordinate the transmissions and receptions during large cycle time periods, providing high complexitity. Although the intra-rack data communication is implemented in the optical domain, the control communication with the rack controller remains in the electrical domain and it is implemented over an Ethernet link. This fact causes serious synchronization problems between the electrical control and the optical data communication planes. In addition, the absence of any contention resolution mechanism or acknowledgment schemes, restricts the SD-MAC protocol reliability.

In this paper, we propose a software-defined optical intra-rack network architecture that implements both the intra-rack data communication and the control communication with the rack controller in the optical domain, ensuring the required control and data channels perfect synchronization. Also, we introduce an intra-rack MAC protocol in conjunction with a bandwidth efficient allocation (BEA) algorithm to fairly assign the intra-rack data wavelengths to the servers for synchronous transmission. The proposed intra-rack MAC protocol with the BEA algorithm provides reception acknowledgment for all the transmitted packets, while it guarantees a totally collision free communication scheme, efficiently avoiding both the collisions over the wavelengths and the destination conflicts. The proposed intra-rack network architecture is totally independent from the inter-rack one, providing high scalability and efficiency. Also the proposed intra-rack MAC protocol with the BEA algorithm performance is extensively

978-3-903176-33-1 © 2021 IFIP

compared via simulations with that of the SD-MAC protocol and it is proven that it provides higher average throughput and low average packet delay for all offered load conditions.

The rest of the paper is carried out as follows: The software-defined intra-rack network is described in Section II. Section III presents the intra-rack MAC protocol. The performance evaluation is given in Section IV. Some conclusions are outlined in Section V.

II. Software-Defined Network Architecture

The proposed software-defined intra-rack network architecture is shown in Fig. 1. Each rack includes *M* servers. The communication among the servers of the same rack is performed locally without involving the ToR switch, to serve the intra-rack traffic. On the other hand, the communication between servers from different racks is performed through the ToR switch level to serve the inter-rack traffic. Therefore, we assume two discrete optical network architectures: the intra-rack and the inter-rack one. The inter-rack network uses the set of wavelengths $\lambda_i, \lambda_j \dots \lambda_z$ for the communication of the rack servers with the ToR switch, as Fig. 1 illustrates.

The interconnection among servers in the intra-rack network is based on passive optical functionalities. Meanwhile, the intra-rack network uses the separate set *S* of *W* wavelengths, $S=\{\lambda_1, \lambda_2 \dots \lambda_W\}$ for the data information exchange among servers in the same rack and they are called data channels. Also, a dedicated wavelength λ_0 is used for the control information exchange and it is called control channel.

Especially in the intra-rack network, we assume a passive optical coupler to implement the data communication over any of the wavelengths from the set *S*, as over a common shared access medium. For the communication in the intra-rack network, each server is connected to the passive coupler with an optical fiber that uses the set of wavelengths *S*.

In the ToR side, the software-defined network architecture includes a rack controller which acts as the control unit for the intra-rack network transmission coordination. Thus in the control plane, the rack controller is responsible for the proposed intra-rack MAC protocol execution in order to appropriately assign the intra-rack network bandwidth among the servers. It is worth mentioning that in the proposed software-defined intra-rack network architecture, the communication between any server and the rack controller is performed in the optical domain over the control channel λ_0.

Each server has two pairs of fast tunable optical transmitters and receivers. The first pair is exclusively used for the data communication over the intra-rack network and it can be tuned over the set of wavelengths *S*. The second pair is also exclusively used for the communication over the inter-rack network and it can be tuned over the wavelengths $\lambda_i, \lambda_j \dots \lambda_z$. Also, each server has a pair of fixed transmitter and receiver for the control information exchange with the rack controller over the control channel λ_0. In other words, each server has three different network interfaces: the intra-rack, the inter-rack and that for the communication with the rack controller, as Fig. 1 presents. Also, each server uses two pairs of multiplexers (MUX) and demultiplexers (DEMUX): one pair for the communication with the passive coupler in the intra-rack network, and another one for the communication in the inter-rack network.

In each server, the generated traffic is classified according to its destination server. Thus, the generated packets that are destined to any server in the intra-rack network are stored in a first-in, first-out way in the output electrical buffer for the intra-rack traffic, while the incoming packets from the intra-rack network are stored in the relative input electrical buffer for the intra-rack traffic, as Fig. 1 shows. Similar, each server accommodates an input and an output electrical buffer for the inter-rack traffic.

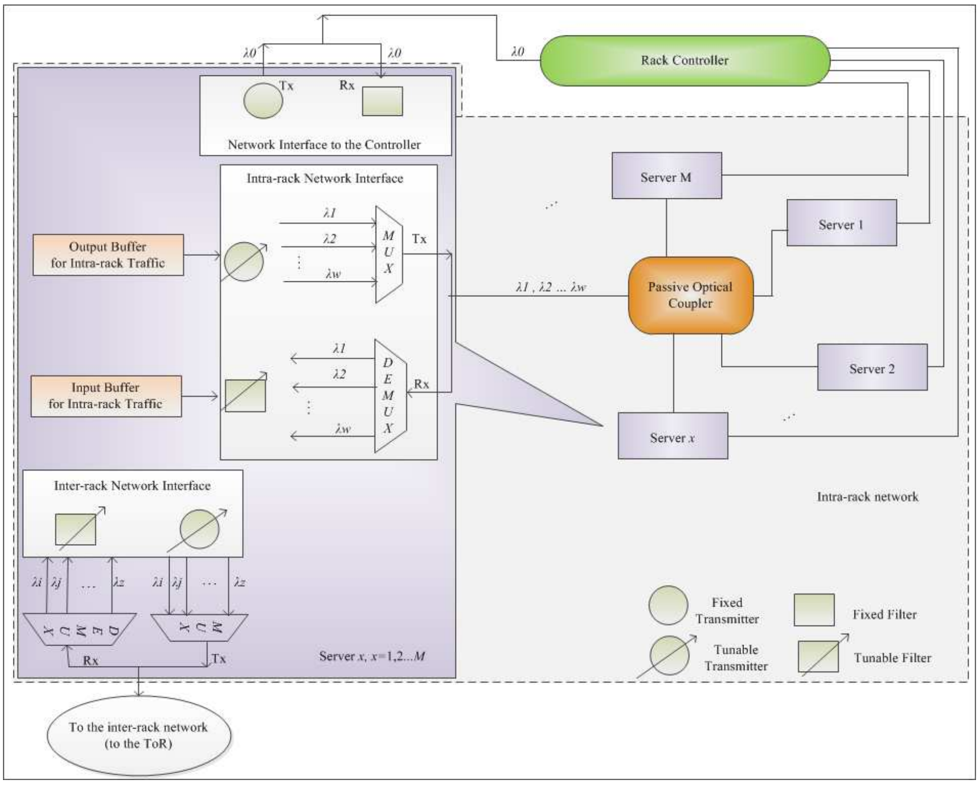

Fig. 1 The software-defined intra-rack network architecture.

III. Intra-Rack Communication and MAC Protocol

In the followings, we focus on the intra-rack network and we leave the inter-rack one for further investigation.

Let's denote by L the fiber length in meters between a server and the passive coupler in an intra-rack network. Also, let's denote by v the light velocity in the fiber in m/s. The propagation delay T_p in seconds between any sever and the passive coupler is:

$$T_p = \frac{L}{v} \quad (1)$$

We assume a common clock in the intra-rack network. All servers are synchronized to it for the control and the data packets transmission and reception. The time axis is divided into contiguous cycles with duration T. The cycle time duration consists of two time periods, as Fig. 2 shows: the first time period is denoted by T_t and is equal to the transmission time of the largest data packet in the intra-rack network, while the second time period is denoted by T_s and is equal to the maximum of the tuning times of the tunable transmitter T_{st} and receiver T_{sr} respectively, i.e.:

$$T_s = \max\{T_{st},\ T_{sr}\} \quad (2)$$

Software-defined control plane

According to the proposed access algorithm, for the data packets transmission during the cycle (i+1), the relative control information must be exchanged between the intra-rack servers and the rack controller during the previous cycle i. Especially, at the beginning of the cycle i, let's say the time instant t, all the servers send a control packet to the rack controller providing the suitable control information in order for the rack controller to run the proposed BEA algorithm for wavelength assignment. Thus as Fig 2 illustrates, a control packet consists of two fields: the source server id and the destination server id of the relative data packet that is accommodated at the first position of the source server's output buffer of intra-rack traffic.

Thus, the control packet size L_c in bits is:

$$L_c = 2 \times K \quad \text{bits} \quad (3)$$

where: K is the number of the required bits for a server identification in binary mode.

If we denote by R the transmission rate of the control channel as well as of each data channel of the set S in b/s, the control packet transmission time T_c is:

$$T_c = \frac{L_c}{R} \quad (4)$$

Starting from the beginning of the cycle i at the time instant let's say t, we define M time slots, where the x-th (x=1, 2,…M) time slot corresponds to the x-th server for the control packet transmission. The duration T_{ts} of each time slot is equal to T_c plus a guard band time T_g, as Fig. 2 shows.

At time instant $T_1 = M \times T_{ts}$ after the cycle beginning, all the control packets have been transmitted by all servers, without experiencing any collision over the control channel. Then, they propagate over the control channel λ_0 and they are received by the rack controller at the time instant ($t + M \times T_{ts} + T_p$). By that time instant, the rack controller has received all the required control information in order to run the intra-rack MAC protocol and especially the BEA algorithm to determine the servers data transmissions and receptions during the next cycle (i+1). The control processing time by the rack controller is denoted by T_{proc}. The result of the BEA algorithm is being announced by the rack controller to the rack servers by sending a software-defined controller response (SDC-R) message over the control channel λ_0. This response consists of W fields, where the y-th (y=1, 2,…W) field corresponds to the y-th data channel of the intra-rack network. In each field, the rack controller writes the source server id to which the BEA algorithm has been assigned the relative data channel for transmission during the next cycle (i+1), as well as the destination server id of the relative data packet.

The SDC-R message size L_m in bits is:

$$L_m = N \times 2 \times K \quad \text{bits} \quad (5)$$

Also, the SDC-R message transmission time T_2 is:

$$T_2 = \frac{L_m}{R} \quad (6)$$

The SDC-R message propagates over the control channel λ_0 and it is received by all rack servers at time instant ($t + M \times T_{ts} + 2 \times T_p + T_{proc} + T_2$). By that time, all the servers are informed about the servers that have won access over the data channels of the intra-rack networks during the next cycle (i+1), according to the intra-rack MAC protocol. If a server recognizes its id at the source id sub-field of the y-th (y=1, 2,…W) field of the SDC-R message, it will transmit a data packet over the y-th data channel during the next cycle (i+1). Similar, if a server recognizes its id at the destination id sub-field of the y-th (y=1, 2,…W) field of the SDC-R message, it will receive a data packet from the y-th data channel during the next cycle (i+1).

Based on this information, at the time instant ($t + T_t$) after the beginning of the cycle i, the servers start tuning their tunable transmitter and receiver to the appropriate data channel for the transmission and reception over the intra-rack network during the next cycle (i+1). The transceivers tuning will have been concluded by the time instant ($t + T_t + T_s$) when the next cycle (i+1) begins.

Intra-rack MAC protocol

According to the intra-rack MAC protocol, all the servers at the beginning of the cycle i, follow the next actions to determine the data packets transmission and reception during the next cycle (i+1).

Especially, the x-th (x=1, 2,…M) server:

- transmits a control packet during the x-th time slot of the cycle i over the control channel λ_0. The control packet is received by the rack controller along with the relative control packets from all the rack servers.
- receives at time instant ($t + M \times T_{ts} + 2 \times T_p + T_{proc} + T_2$) from the control channel λ_0 the SDC-R message sent by the rack controller.

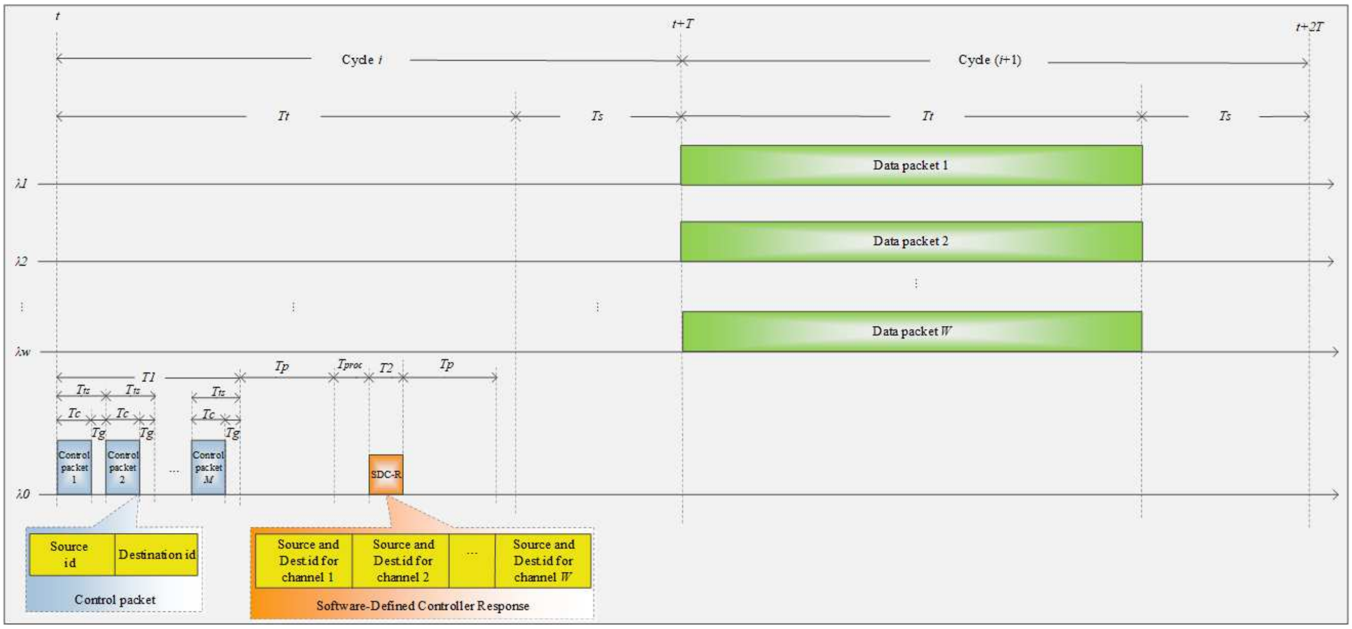

Fig. 2. Cycle structure for the intra-rack communication: The control information is transmitted to the SDN rack controller at the beginning of the i-th cycle, over the control channel λ_0. Afterwards, the SDN controller responses with a SDC-R control message during the i-th cycle, over control channel λ_0. The SDC-R message informs about the wavelength assignment by the rack controller. The tunable trasnceivers tuning time Ts follows till the cycle completion. The relative data packets are transmitted by the servers during the $(i+1)$-th cycle over the assigned data channels. In the figure, it is assumed that the servers transmit long data packets of 1500 Bytes over data channels.

- Based on the SDC-R message, at time instant $(t+T_t)$: it tunes its tunable intra-rack transceivers to the appropriate data channels for transmission and reception during the next cycle $(i+1)$.

Also according to the intra-rack MAC protocol, the rack controller runs the BEA algorithm to assign the data channels bandwidth to the rack servers. Thus, the rack controller by the time instant $(t+M\times T_{ts}+T_p)$ when it has received the control packets from all rack servers, it follows the next steps:

- In order to avoid any data packet collision due to the concurrent transmission over the same data channel, it selects randomly W servers that attempt transmission during the next cycle $(i+1)$ and it assigns to each of them one available data channel from the set S.
- In order to avoid any destination conflict at the receiver side, it checks if any of the selected source servers have the same destination. In this case, it selects randomly one of these source servers and gives to this server the right to transmit to the common destination server, while it rejects the transmission right to the other collided at destination servers. If there are more servers attempting for transmission, it selects randomly as many servers as the rejected servers are, and runs this step again till no other servers attempting transmission are available or the number of selected servers for transmission is higher than the number W of data channels.
- Then it writes the selected servers ids in the source id sub-field of the SDC-R message fields that correspond to the assigned data channels, while it writes in the destination id sub-field the destination server id of the relative data packets.
- Finally, it sends the SDC-R message over the control channel λ_0, in order to inform all racks servers about their access and reception rights during the next cycle $(i+1)$.

It is worth mentioning that the proposed intra-rack MAC protocol along with the BEA algorithm totally avoids the packets collisions over the control and the data channels in the intra-rack network, while it effectively faces the destination conflicts, efficiently exploiting the available data channels bandwidth. Also, the proposed intra-rack MAC protocol along with the BEA algorithm provides fair access among all servers over the intra-rack network, since the transmission access right is granted based on random selection procedures.

It is evident that the maximum throughput Th_{max} that can be achieved by the proposed intra-rack MAC protocol is:

$$Th_{\max} = \frac{T_t}{T_t + T_s} \tag{7}$$

IV. PERFORMANCE EVALUATION

In order to evaluate the proposed intra-rack MAC protocol performance along with the proposed BEA algorithm, we implemented a discrete time network simulator based on the C programming language. The network simulator simulates the servers' actions in the intra-rack network as well as the rack controller algorithmic steps for the BEA algorithm, while it provides predictions for the intra-network average throughput and average packet delay performance measures.

For studying comparative results, we choose to compare the performance of the proposed intra-rack MAC protocol and the BEA algorithm with that of a relative SD-MAC one presented in [11]. In order to have a similar base for comparison, we follow the assumptions of SD-MAC protocol. Thus, we assume that the intra-rack network consists of W=8 data channels, while it interconnects M=64 servers. The fiber length in the intra-rack network is L=10 m, which means that the propagation delay in the intra-rack fiber is T_p=50 ns. The tunable transceivers tuning time T_s is assumed to be 200 ns. The control and each data channel data rate is assumed to be R_w=10 Gbps. The traffic model considered is similar to that of [11], while the ratio of intra-rack to inter-rack traffic is equal to 80%.

We define the following performance measures. First, we define the normalized load as the ratio of the average throughput obtained over all data channels to the data channel data rate. Also, we define the average throughput in b/s of the intra-rack network, as the ratio of the total size of all data packets sent by all servers in the intra-rack network during the simulation time to the simulation time. Also, the average packet delay is defined as the mean time in seconds between a data packet generation and its reception by the destination server, during the simulation time. It has to be noticed that the average packet delay counts in the packet queuing delay at the output buffer of the intra-rack traffic, the transmission delay and the round trip propagation delay time ($2 \times T_p$).

Fig. 3 shows the average throughput vs the normalized load for the proposed intra-rack MAC protocol and the BEA algorithm as compared to the SD-MAC protocol, for M=64 servers, W=8 data channels and R_w=10 Gbps. As it is shown, the proposed intra-rack MAC protocol and the BEA algorithm achieves much higher throughput than the SD-MAC one for the entire normalized load range. For example, the throughput achieved is higher: 28% for load=0.4, 29% for load=0.6 and 30% for load=0.8. The higher throughput level is explained by the fact that the proposed intra-rack MAC protocol in conjunction with the BEA algorithm effectively allocates the data channels bandwidth to the rack servers, totally avoiding both the channel collisions and the destination conflicts. This is achieved by simply assigning the available data channels to the servers and allowing the transmission of only one data packet of those which would collide at the destination, rejecting the rest of them. This is a really simple access algorithm with high performance. On the other hand, the SD-MAC protocol adopts a more complex access scheme by organizing the rack servers into transmitting and receiving groups, aiming to reduce the data channels collisions and the destination conflicts, without managing to obtain very high bandwidth exploitation. It is worth mentioning that the proposed intra-rack MAC protocol and the BEA algorithm for load=0.88 achieves almost 88% total data bandwidth exploitation that conforms to the results of (7), while the SD-MAC one reaches only 51.5% for the same load conditions.

Moreover, Fig. 4 presents the average packet delay vs the normalized load for the proposed intra-rack MAC protocol and the BEA algorithm as compared to the SD-MAC protocol, for M=64 servers, W=8 data channels and R_w=10 Gbps. As it is illustrated, the proposed intra-rack MAC protocol in conjunction with the BEA algorithm achieves much lower average packet delay values. For example, the delay achieved is 3.36 μs for load=0.4, 3.45 μs for load=0.6 and 4.3 μs for load=0.8. These values are almost 98% lower

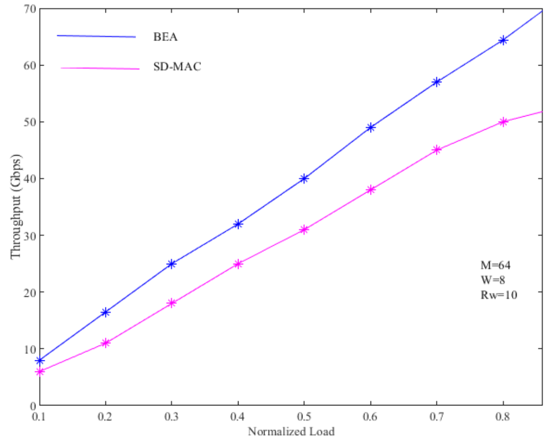

Fig. 3. Average throughput vs normalized load, for M=64, W=8 and R_w=10 Gbps, for the proposed BEA and the SD-MAC protocol.

than the relative delay values of the SD-MAC protocol. This SD-MAC protocol performance is due to the fact that it assumes long time cycles, which are additionally subdivided into sub-periods, to organize the servers into transmission and reception groups and to coordinate the data packets communication during the whole cycle period, causing long delays. Also, the adoption that the control communication between any server and the rack controller is performed in the electrical domain over a dedicated Ethernet link, burdens the time framing for synchronization purposes, causing high delays. On the other hand, the proposed intra-rack MAC protocol in conjunction with the BEA algorithm eliminates the total transceivers tuning time, by properly defining a specific time period for transceivers tuning during the last part of the previous cycle, after the SDC-R message reception by all servers. In addition, the proposed intra-rack network configuration adopts a separate wavelength for the control communication with the rack controller, exclusively in the optical domain, managing to ensure synchronization preconditions and achieving low delay values.

In order to explore the proposed intra-rack MAC protocol and the BEA algorithm performance for diverse intra-rack network configurations, we study the throughput and the packet delay when the number W of data channels varies. Especially, Fig. 5 shows the average throughput vs the

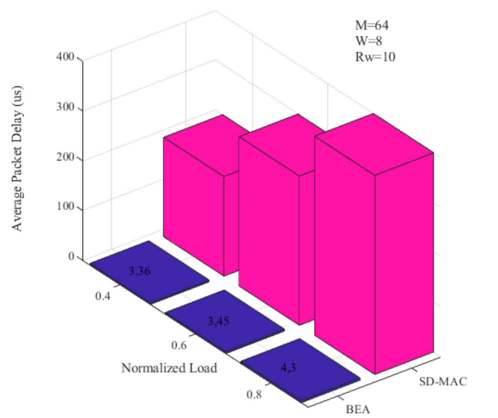

Fig. 4. Average packet delay vs normalized load, for M=64, W=8 and R_w=10 Gbps, for the proposed BEA and the SD-MAC protocol.

normalized load for M=64 servers, R_w=10 Gbps and W=4, 8, 12 data channels. As it is observed, the throughput achieved is an increasing function of W. For example for load=0.8, the throughput is: 32.5 Gbps for W=4, 64.4 Gbps for W =8 and 96 Gbps for W =12. This behavior is explained by the fact that as the number of data channels increases, the available data bandwidth increases too. Therefore, the BEA algorithm assigns more data channels for transmission, reducing the probability of a data packet collision. This provides higher throughput values.

The above results are validated in Fig. 6 that illustrates the average packet delay vs the average throughput for M=64 servers, R_w=10 Gbps and W=4, 8, 12 data channels. Indeed, the intra-rack network configuration for load=0.8 achieves throughput improvement about: 98% when increasing the number W of data channels from W=4 to W=8, and 50% when it increases from W=8 to W=12. The relative average packet delay keeps very low values, lower than 5 μs, for normalized load conditions up to 0.87. For loads higher than this threshold, the intra-rack network reaches saturation and the delay dramatically increases, as Fig. 6 illustrates.

V. CONCLUSIONS

In this paper we focus on the performance improvement conditions inside the software-defined passive optical intra-rack network of a data center. We propose an intra-rack network configuration, based on passive optical functionalities, that occupies several wavelengths to guarantee the data traffic communication among servers in the same rack. The software-defined paradigm is employed by a rack controller that performs the access coordination over the intra-rack network. In opposition to other intra-rack networks, in our study the communication between any server and the rack controller is performed in the optical domain over a dedicated wavelength, ensuring the preconditions for synchronization among the transmissions.

Moreover, we propose an intra-rack MAC protocol in conjunction with the BEA algorithm, to schedule the data packets transmission and reception over the intra-rack network in a per cycle base. Extensive simulation results prove that the proposed intra-rack MAC protocol and the BEA algorithm, as compared to the relative protocol of [11], achieves almost 30% higher throughput and 98% lower packet delay, grace to the effective data channels allocation algorithm followed which totally avoids both the data channels collisions and the destination conflicts. This study could be extended to topics like the inter-rack network access coordination and the power consumption restrictions.

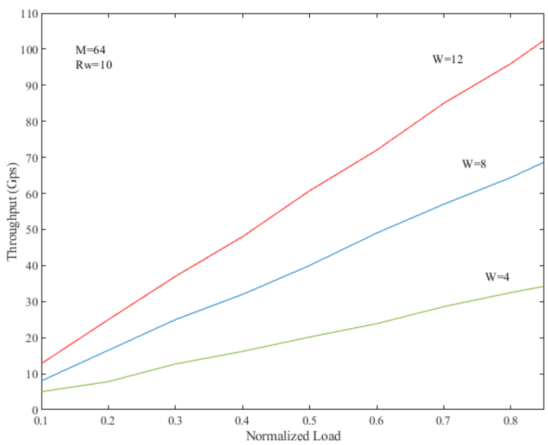

Fig. 5. Average throughput vs normalized load, for M=64, W=4,8,12 and R_w=10 Gbps, for the proposed BEA protocol.

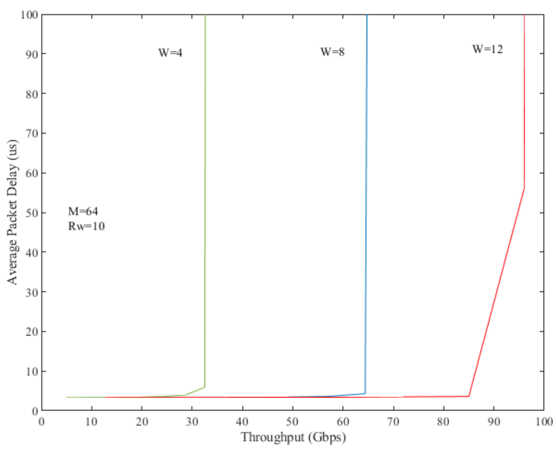

Fig. 6. Average packet delay vs average throughput, for M=64, W=4,8,12 and R_w=10 Gbps, for the proposed BEA protocol.

REFERENCES

[1] Urata R, Liu H, Zhou X, and Vahdat A., "Datacenter interconnect and networking: from evolution to holistic revolution", *in Proc. OSA/IEEE Optical Fiber Communications Conference and Exhibition*, 2017, Los Angeles, CA.

[2] N. Binkert, A. Davis, N. P. Jouppi, M. McLaren, N. Muralimanohar, R. Schreiver, and J. H. Ahn, "The role of optics in future high radix switch design," *in Proc. 38th Annu. Int. Symp. on Comp. Archit.(ISCA)*, 2011, pp. 437–447.

[3] Farrington N, Porter G, Radhakrishnan S, et al., "Helios: a hybrid electrical/optical switch architecture for modular data centers", *in Proc. ACM SIGCOMM Conference* 2010, 2010,New Delhi, India.

[4] Chen K, Singla A, SinghA, et al., "AnOptical switching architecture for data center networks with unprecedented flexibility", *in Proc. USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2012, San Jose, CA.

[5] Tang Y, GuoH, Zhu Y, et al., "Effectively reconfigure the optical circuit switching layer topology in data center network by OCBridge",. J Light Technol., vol. 37, no. 3, pp. 897-908, 2019.

[6] C. Kachris and I. Tomkos, "A survey on optical interconnects for data centers," IEEE Commun. Surveys Tuts., vol. 14, no. 4, pp. 1021–1036, Fourth quarter 2012.

[7] C. Kachris, K. Kanonakis, and I. Tomkos, "Optical interconnection networks in data centers: Recent trends and future challenges," IEEE Commun., Mag., vol. 51, no. 9, pp. 39–45, Sep. 2013.

[8] Perello J, Spadaro S, Ricciardi S, et al., "All-optical packet/circuit switching-based data center network for enhanced scalability, latency, and throughput", IEEE Network., vol. 27, no 6, pp. 14-22, 2013.

[9] Yu Y, Zhang J, Zhao Y, et al., "Field demonstration of multi-domain software-defined transport networking with multi-controller collaboration for data center interconnection", IEEE/OSA J Opt Commun Netw. 2015;7(2):A301-A308.

[10] Pries R, Jarschel M, Schlosser D, Klopf M, Tran-Gia P., "Power consumption analysis of data center architectures", *in Proc. International Conference on Green Communications and Networking*, 2011, Colmar, France.

[11] Y. P. Cai, Z. Yao, T. Li, S. Luo, and L. Zhou, "SD-MAC: Design and evaluation of a software-defined passive optical intrarack network in data centers", Trans. Emerging Telecom. Techn., e3764, 2019.

[12] Y. P. Cai, S. Luo, and L. Zhou, "Software defined passive optical intra-rack networks in data centers," *in Proc. IEEE Globecom*, 2016, pp. 1–6.

[13] Y. P. Cai, S. Luo, L. Zhou, Z. Yao, and T. Li, "Design and Evaluation of a Software Defined Passive Optical Intra-Rack Network in Data Centers", *in Proc. IEEE INFOCOM 2019 - ICCN Workshops,* 2019, pp.56-61.