LPB: A Novel Load Balance Algorithm for OPSquare DCN under Real Application Traffics

Fulong Yan
Institute for Photonic Integration
Eindhoven University of Technology
Eindhoven, the Netherlands
f.yan@tue.nl

Xiaotao Guo
Institute for Photonic Integration
Eindhoven University of Technology
Eindhoven, the Netherlands
x.guo@tue.nl

Bitao Pan
Institute for Photonic Integration
Eindhoven University of Technology
Eindhoven, the Netherlands
b.pan@tue.nl

Xuwei Xue
Institute for Photonic Integration
Eindhoven University of Technology
Eindhoven, the Netherlands
x.xue.1@tue.nl

Shaojuan Zhang
Institute for Photonic Integration
Eindhoven University of Technology
Eindhoven, the Netherlands
s.zhang4@tue.nl

Nicola Calabretta
Institute for Photonic Integration
Eindhoven University of Technology
Eindhoven, the Netherlands
n.calabretta@tue.nl

***Abstract*— Network virtualization in today's data center (DC) creates new heterogeneous traffic patterns different from what so far have been reported in literatures. To properly evaluate the DC network performance, a virtualized DC traffic model is developed from the real application traffics. In this paper, we propose a novel lowest path buffer (LPB) algorithm and evaluate the network performance of LPB in OPSquare DC under the developed traffic model. Compared with Round-Robin and Localflow, LPB could achieve 23.7% and 32.1% less latency, respectively. Besides, LPB provides lower packet loss with respect to Round-Robin, Drill and Localflow.**

Keywords—Data center network, Traffic modeling, Load balancing.

I. INTRODUCTION

To support the large scalability and huge bandwidth needed in the next generation data center (DC), data center network (DCN) architectures adopting optical switches are proposed in recent years [1][2]. The fast optical switch (FOS) based OPSquare DCN, composed of two parallel intra-cluster and inter-cluster subnetworks, attracts lots of attention due to its high scalability, good network performance as well as cost and power consumption efficiency [2]. Moreover, OPSquare architecture naturally provides multiple paths between racks, which can be exploited, supported by a load balancing algorithm, to cope with the dynamic traffic to improve the bandwidth utilization and optimize the network performance.

Several load balancing algorithms have been investigated in the DCN, such as Equal-cost multi-path routing (ECMP), Localflow [3], Hedra and so on. The proposed DCN load balancing algorithms operating on the granularity of flow can be classified in two categories: the centralized and the distributed. ECMP is the most prevalently adopted distributed load balancing mechanism due to its simplicity. ECMP splits the flows without considering the size of the flow and the current network status, resulting in unsatisfied performance. Localflow also works distributedly, but it needs to monitor the status of the flow, which may not be accurate and could increase network operation overhead [3]. Centralized algorithms such as Hedra and MicoTE split the flows considering the flow size and the network status, but they react on a very large time scale (several seconds). Consequently, they are not suitable for the dynamic changing traffic (on the order of millisecond) in DC.

Besides the load balancing algorithms operating on the granularity of flow, there are also algorithms on the granularity of packet, such as Random, Round-Robin (RR) [4], and digital-reversal. The main problem of those packet level load balancing algorithms is the packet re-ordering in the receiver. All those generalized algorithms do not consider the peculiarity of the OPSquare DCN architecture with 2 shortest paths. Moreover, with the novel applications emerging in today's DC, the trend of network virtualization creates new heterogeneous traffic patterns different from what so far have been reported in literatures. In this paper, considering the dynamicity of the virtualized DC traffic and the peculiarity of the OPSquare network, a dedicated load balancing for OPSquare, , called lowest path buffer (LPB), is proposed to improve the network performance.

II. THE PROPOSED LOAD BALANCING ALGORITHM

As shown in Fig. 1(a), there are two parallel subnetworks in OPSquare DCN network. The intra-cluster and the inter-cluster interconnection are accomplished by the intra-cluster

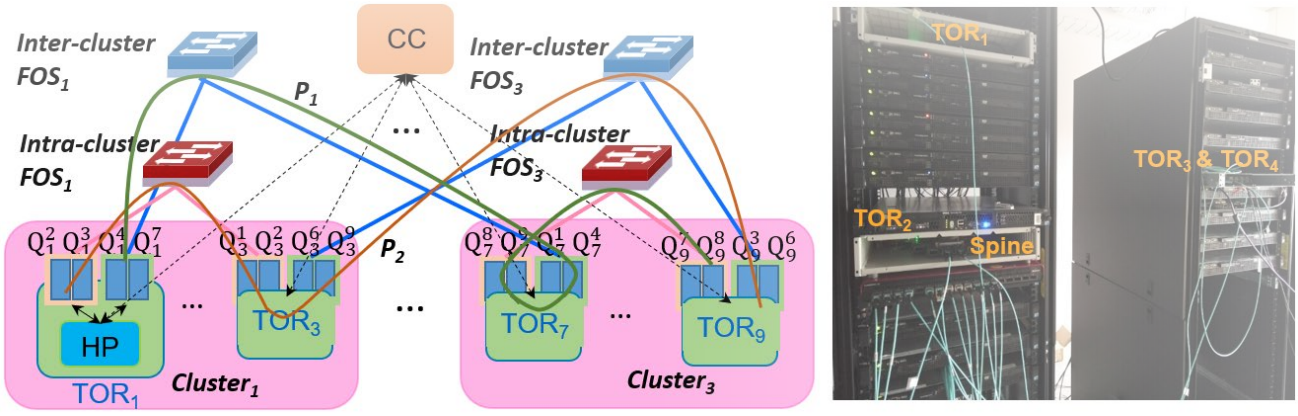

Fig. 1. (a) Downsize OPSquare DCN with 9 top of racks (TORs), (b) The ECO DCN setup.

978-3-903176-21-8 (c) 2020 IFIP

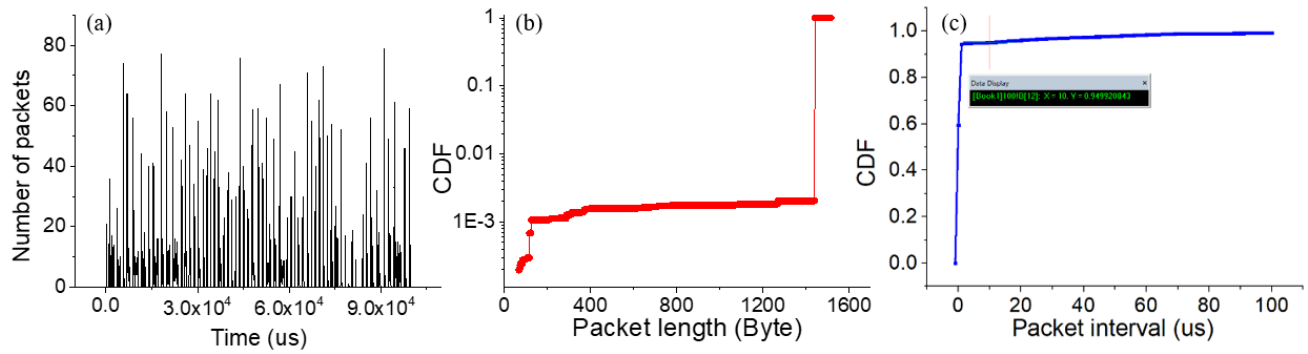

Fig. 2. (a) Traffic is binned by 100 μs, (b) CDF of the packet length, (c) CDF of the packet interval.

and inter-cluster FOS, respectively. For top of racks (TORs) in different clusters of OPSquare, either they are connected to the inter-FOS directly (TOR_1→inter_FOS_1→ TOR_7) if the TOR pairs are TOR_1 and TOR_7, or they are connected by the inter-FOS, intermediate TOR and intra-FOS resulting 3 hops (TOR_1→inter_FOS_1→ TOR_7→intra_FOS_3→ TOR_9) when the TOR pairs are TOR_1 and TOR_9. Considering an OPSquare DCN interconnects N^2 TORs with N-radix FOS, there would be N-1 logic queues for the buffer of the intra-cluster and inter-cluster TRXs.

In TOR *i*, $\mathbf{Q}_j$ is denoted as the logic queue buffering packets for *j*-th TOR, while the occupation of $\mathbf{Q}_j$ is marked as L_j. Our proposal of the load balancing is similar to the Drill [5] which is also based on the buffer occupation and operates on the granularity of packet. However, Drill only counts the buffer occupation of the current hop, resulting in >10 μs latency at busy server ratio of 0.5, which is unsatisfied. Based on the occupation of the TOR buffer size in the whole path, we propose a novel dedicated lowest path buffer (LPB) load balancing algorithm for OPSquare DCN. In the operation of LPB, the central controller (CC) monitors the intra-cluster and inter-cluster buffer occupation. The head processor (HP) of the TOR updates the status of the buffer occupation of all the TORs from the CC. Then the buffer occupation of the two paths for the received packet are calculated. Take as an example the downsize OPSquare DCN architecture shown in Fig. 1 (a), assuming TOR_1 as the source TOR, there are two available paths when TOR_9 acts as the destination TOR. The first path (P_1) is TOR_1→ inter-FOS_1→ TOR_7→ intra-FOS_3→ TOR_9, while the second path (P_2) would be TOR_1→ intra-FOS_1→ TOR_3→ inter-FOS_3→ TOR_9. The packets on P_1 will pass through Q_1^7 and Q_7^9, while the packets on P_2 pass through Q_1^3 and Q_3^9. Therefore, the path buffer occupation for P_1 and P_2 are $L_1^7 + L_7^9$ and $L_1^3 + L_3^9$, respectively. Under the case of $L_1^7+L_7^9 < L_1^3 + L_3^9$, P_1 will be chosen as the transmission path. Therefore, the packet will be queued in the inter-cluster buffer $\mathbf{Q}_1^7$. Otherwise, P_2 will be chosen as the transmission path, and the the packet will be queued in the corresponding intra-cluster buffer $\mathbf{Q}_1^3$. Benefiting from the characters of the OPSquare network architecture, the proposed LPB algorithm considers the occupation of the TOR buffer size in the whole path rather than the TOR buffer occupation of the current hop. Therefore, LPB can achieve load balancing efficiently under dynamic changing DC traffic.

In the simulations, the link distance between the ToR and the CC is 100 m. The LPB messages generated by the CC are sent back to the ToRs by separate links using out of band signals. We do not consider the case that the LPB message is lost. The time serial procedures on the operation of LPB is implemented as follows. At time t, all ToRs send their buffer status to the CC. After receiving the buffer status from ToRs, the CC starts to process all the message at t+0.5 μs. The processing time of CC is taken as 99 μs, that is to say, the packet will be sent back to the ToR at time t+99.5 μs. Therefore, the ToRs receive message from CC at t+100 μs, and the HPs update the buffer status of the network. Meanwhile, all ToRs send buffer status again to the CC to start the next period operation of LPB. Therefore, the update period of the buffers status is 100 μs. Additional patch is needed to export the buffer status of the TOR from HP to the CC.

Benefiting from the characters of multiple paths in the OPSquare network architecture, the proposed LPB algorithm considers the occupation of the ToR buffer size in the whole path rather than the ToR buffer occupation of the current hop. Therefore, LPB is expected to achieve load balancing efficiently under various traffic patterns. The intrinsic improvement of LPB is the complete balanced load on the whole network rather than part of the links. The associated cost is to exchange the ToR buffer status between ToR and CC. The ToR needs to maintain the buffer status of the whole network based on the message sent by CC. The ToR updates its buffers status when head processor receives new message from CC.

III. Virtualized DC Traffic Modeling

To optimize the network performance, the traffic model of the virtualized application should be developed firstly. We build our Leaf-Spine electrical & optical communication (ECO) DC as shown in Figure 1(b). In the ECO DC, each TOR connects 4 servers, and there are in total 16 servers in the network connected by 4 TORs. Then the traffic traces of the running applications are captured and analyzed. The developed traffic model acts as the traffic generator in the section 4 for the network performance evaluations.

We build a Kubernetes cluster on the ECO DCN for orchestrating the application Docker containers automatically. Then we deploy the containerized web serving, media streaming and the Spark computing applications in our ECO DCN. By capturing the traffic in the ECO DCN of the 16 servers for 5 minutes, we analyze the captured traces to develop the traffic model. As shown in Fig. 2 (a), when the captured traffic is binned by 100 μs, it demonstrates a clear ON-OFF characteristics. In Fig. 2(b), we can observe the bimodal distribution of the packets in the captured traces. One is around 100-byte, while the other is around 1500-byte. It's similar to the results shown in [6]. Figure 2 (c) describes cumulative distribution function (CDF) of the packet interval, and 95% of the packet intervals are less than 10 μs.

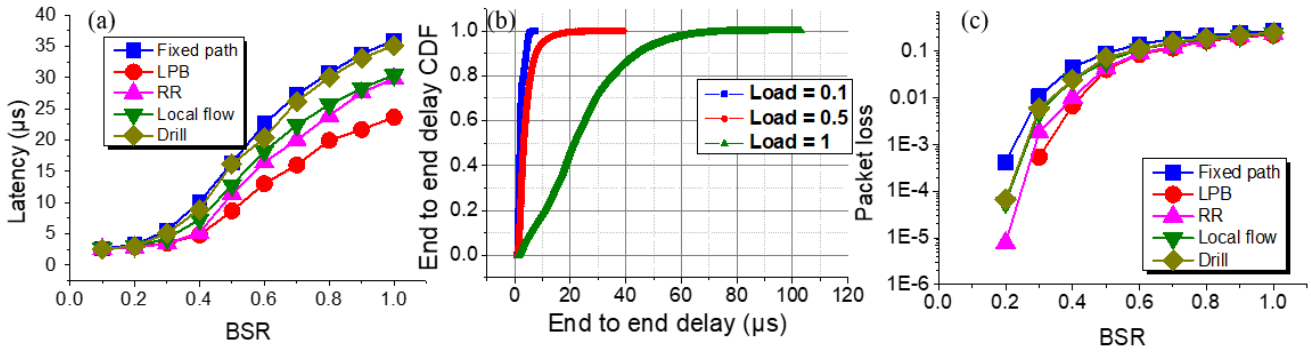

Fig. 3. (a) Latency loss of various algorithms, (b) latency CDF of LPB, (c) Packet loss of various algorithms.

IV. OPSQUARE NETWORK PERFORMANCE OPTIMIZATION: RESULTS AND DISCUSSION

For an OPSquare DCN connecting 2560 servers (FOS radix is 8), each of which generates packets based on the traffic model we build upon the collected traffic of real DC applications running in ECO DC shown in section 2. OMNeT++ is used to build the OPSquare DCN for running the simulations. For the simulation setup, the OPSquare DCN is equally divided into two subnetworks. The first subnetwork comprises of the server with index in the range of [1, 1280], while the servers with index in the range of [1281, 2560] construct the second subnetwork. To emulate the imbalance load of the OPSquare DCN, the servers in the second subnetwork do not create inter-cluster traffic. Considering the fact that a large amount of the DC traffic is exchanged within the TOR, we set 40% intra-TOR, 40% inter-cluster traffic, and 20% intra-cluster traffic in the first subnetwork.

In OPSquare DCN, there is no need for load balancing for the intra-cluster traffic since the intra-cluster traffic only pass one hop. Therefore, we focus on the load balancing of the inter-cluster traffic and investigate the network performance of the inter-cluster traffic. In the simulations, the buffer size is fixed as 50 KB. We vary the ratio of the servers that are busy generating packets with load from 0.1 to 1. The busy server ratio (BSR) is in the range of [0.1, 1], and idle servers do not generate packets. We compare the performance of LPB with the fixed path (none load-balancing), Localflow, and RR.

Figure 3(a) shows the server-to-server latency of the inter-cluster traffic in the first subnetwork. The server-to-server latency is defined as the difference of the time stamp of packet received at the destination server and time stamp of packet sent out by the source server. With the increase of the BSR, the server-to-server average latency increases for all the load balancing algorithms. When the BSR is < 0.3, the server-to-server latency of all the algorithms is < 5 µs, and there are only marginal benefits of the load balancing algorithms compared with fixed path (none load balancing). However, when the BSR is > 0.4, the latency of fixed path increases rapidly, and the latency performance improvement between fixed path and LPB becomes increasingly noticeable. At BSR of 0.5, the latencies of fixed path, RR, Localflow, Drill and LPB are 16.3, 11.4, 12.8, 16.2 and 8.7 µs, respectively. The latency of LPB is 23.7% less than that of RR.

Figure 3(b) shows the CDF of the server-to-server latency of LPB. As the load increases, the maximum latency increases due to the high contention probability. When the load is 0.1, 99% of the latency is lower than 5.5 µs. Moreover, 90% of the latency is lower than 7.7 µs at load of 0.5. The maximum latency is 103 µs at load of 1. Limiting the maximum number of retransmissions results in lower server-to-server latency, but at the expense of high packet loss.

The packet loss performance is shown in Fig. 3(c). For Localflow and RR, a packet loss less than 10^{-4} is achieved when the BSR is 0.2. A packet loss less than 10^{-3} for BSR of 0.3 is achieved for LPB. The performance of LPB is better than Localflow and RR for all different BSRs. When the BSR reach 0.5, the packet loss is unavoidable (>10^{-2}) due to the high probability of contention.

V. CONCLUSION

To improve the bandwidth utilization and optimize the network performance in OPSquare DCN, we proposed a novel load balancing algorithms LPB which considers the characters of OPSquare architecture. Based on the traffic traces captured in our ECO DCN with virtualized representative DC applications running inside, the real traffic model is developed to optimize the OPSquare network performance. The latency of LPB is 23.7% less than that of RR at BSR of 0.5. And the packet loss ratio of LPB is lower than Localflow and RR. The results validate that LPB could achieve optimal network performance in OPSquare DCN compared with the existing load balancing algorithms.

ACKNOWLEDGMENT

The authors would like to thank the European Union's Horizon 2020 research and innovation programme under grant agreement PASSION No 780326 and the EU H2020 QAMELEON project (n° 780354) for supporting this work.

REFERENCES

[1] N. Farrington *et al.*, "Helios: a hybrid electrical/optical switch architecture for modular data centers," *Proc. ACM SIGCOMM 2010 Conf. SIGCOMM - SIGCOMM '10*, p. 339, 2010.

[2] F. Yan, W. Miao, O. Raz, and N. Calabretta, "OPSquare: A Flat DCN Architecture Based on Flow-Controlled Optical Packet Switches," *J. Opt. Commun. Netw.*, 2017.

[3] S. Sen, D. Shue, S. Ihm, and M. J. Freedman, "Scalable, optimal flow routing in datacenters via local link balancing," in *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies - CoNEXT '13*, 2013.

[4] P. Samal and P. Mishra, "Analysis of Variants in Round Robin Algorithms for Load Balancing in Cloud Computing," *Int. J. Comput. Sci. Inf. Technol.*, 2013.

[5] S. Ghorbani, Z. Yang, P. B. Godfrey, Y. Ganjali, and A. Firoozshahian, "DRILL: Micro Load Balancing for Low-latency Data Center Networks," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication - SIGCOMM '17*, 2017.

[6] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The Nature of Datacenter Traffic: Measurements & Analysis," in *IMC '09 Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, 2009.