Dynamic Buffer Status Based Conflict Free Scheduling for a Fast Optical Switching Network

Fulong Yan[(1)], Chongjin Xie[(2)], Nicola Calabretta[(3)]

[(1)] Alibaba Cloud, Alibaba Group, Beijing, China, yanfulong.yfl@alibaba-inc.com
[(2)] Alibaba Cloud, Alibaba Group, Sunnyvale, California 94085, USA,
[(3)] Eindhoven University of Technology, Eindhoven, the Netherlands,

Abstract *We propose a dynamic buffer status matrix decomposition (BSMD) based conflict free scheduling for a fast optical switching network. The performance of BSMD outperforms both static and retransmission scheduling mechanisms, and BSMD achieves 10.1 μs latency and 98.8% throughput at load of 0.8.*

Introduction

With the booming of big data, cloud computing and internet of things, the data center (DC) traffic is increasing at an annual growth rate of 27%[1]. To meet the huge bandwidth requirement in DC, network interface card (NIC) and optical transceiver per lane supporting 100Gb/s are deployed. Meanwhile, with switch ASIC capacity approaching 100 Tb/s, it is challenging to increase its capacity further due to limited ball grid array density.

Alternatively, optical switching technologies are receiving broad interests from both the industry and academic community. Optical circuit switches (OCSes) are suitable for limited application scenarios due to the switching capability on the granularity of flows (from hundreds μs to tens of ms) rather than packets. To support the packet switching as achieved in electrical switches, fast optical switch (FOS) capable of switching on the order of nanoseconds must be considered.

FOS can be implemented based on various materials. Liquid crystals, semiconductor optical amplifier (SOA), waveguid bragg grating and Lithium niobate ($\mathrm{LiNbO_3}$) are all based on ultrafast electro-optic effect[2]. Currently, due to the lack of feasible optical buffer, the high performance scheduling algorithms for electrical switches can not be applied directly for the FOS. While the prevalent scheduling algorithm of FOS is failing and retransmission (FRT) mechanism, which unavoidably results in low network throughput under high load. Recently, a static scheduling algorithm considering the traffic rate matrix decomposition (TRMD) was proposed to improve the network performance of FRT[3]. However, TRMD requires the foreknowledge of network traffic rate matrix, which might be dynamic and inaccurate even if available in real DCs.

In this paper, we propose a dynamic conflict free scheduling mechanism considering the buffer status matrix decomposition (BSMD). The network latency and throughput of BSMD are investigated under realistic ON/OFF Pareto traffic with various mean burst lengths and frame lengths. Besides, we compare the performance of BSMD with that of TRMD and FRT to validate the effectiveness of BSMD. The obtained results show that BSMD outperforms TRMD and FRT.

BSMD implementation

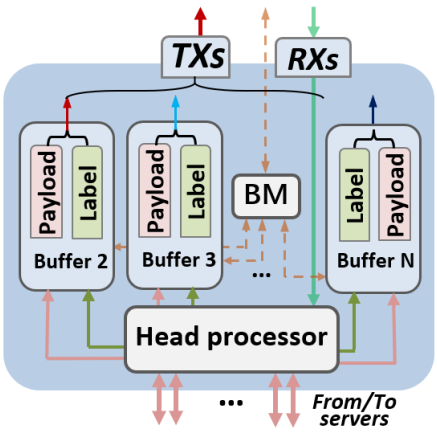

Fig. 1: The schematic of the ToR (BM: buffer management module; TX: transmitter; RX: Receiver).

The blocks of the ToR is illustrated in Fig. 1. There are $N-1$ logical buffer queues inside each ToR where the i-th queue buffers the packets destining to i-th ToR. The packet arriving from the servers are first processed by the head processor before sending to the dedicated queue. The buffers are segmented into cells with 64 bytes. The number of cells N_{cell} that a packet occupied is calculated by the following formula:

978-1-6654-3868-1/21/$31.00 ©2021 IEEE

$$N_{cell} = ceil(L_{packet}/64) \quad (1)$$

where L_{packet} is the length of the packet and $ceil$ is the function which return the minimal integer that no less than the input number. The queues update the occupation to the buffer management module (BM) as there is cell arriving and cell erasing.

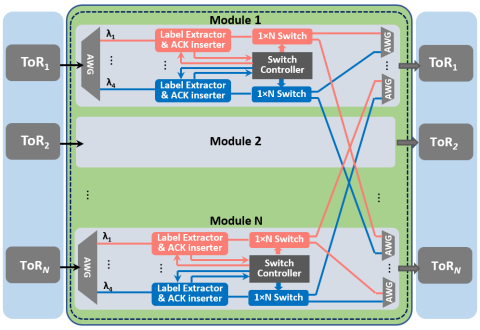

Fig. 2: The building blocks of the FOS.

The FOS monitors the dynamic traffic characteristics of the N ToRs through obtaining the buffer status matrix $\mathbf{B} = [b_{i,j}]_{N\times N}$ from the BM of ToRs in the FOS based network. $b_{i,j}$ is the buffer occupation of the queue j inside ToR_i ($i \neq j$). An optical packet is composed of c cells, and the length of the frame is set as T_f. Namely, there are T_f/T_s optical packets need to be scheduled in one frame where T_s is the length of optical packet slot .

The detailed operation of the BSMD is given in the following Algorithm 1. To start the scheduling in the first frame ($C_f = 1$), the ToRs simply adopt round robin mechanism under the identity matrix $\mathbf{I}_N$ and its $N-1$ permutation matrices $\mathbf{I}_N^i$ ($i \in [1, N-1]$). $\mathbf{I}_N^i$ is obtained by putting the j-th column of $\mathbf{I}_N$ as the $mod(j+i, N)$-th column of $\mathbf{I}_N^i$. Before the start of the subsequent frames ($C_f > 1$), the BMs sends buffer status to FOS at time t_1 where t_1 is resolved in Algorithm 1.

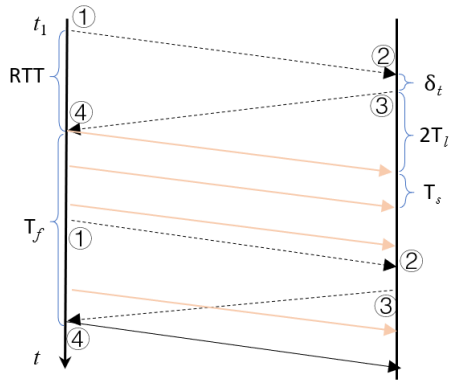

Fig. 3: The timestamps of BSMD (RTT:round trip time).

After a link delay of T_l, at timestamp $t_1 + T_l$ denoted by ② in Fig. 3, the buffer status of ToRs is provided to the FOS, and the FOS starts to decompose $\mathbf{B}$ under Birkhoff and von Neumann theorem (BvN) which is implemented for the scheduling of an input buffered crossbar switch[4]. While at time $t = t_1 + T_l + \delta_T$ of ③ in Fig. 3, the FOS completes the decomposition of $\mathbf{B}$ and begins to send α_k and P_k ($k \leq N^2 - 2N + 2$) to ToRs. After the link delay of T_l, the ToRs receive and update α_k and P_k at time $t = t_1 + RTT$ of ④.

At the starting of every frame, β_i is calculated by accumulating $\alpha_k (k \leq i)$ to determine the scheduling ratio of the matrix $\sum_{k=1}^{i} P_k$. Then the scheduling matrix P_k of current timestamp is obtained by solving the relation between current time t and $\beta_k \times T_f$ as shown in Algorithm 1. When the optical packets arrive at the FOS, the FOS is configured with the corresponding matching matrix $\mathbf{P}_k$ so that no contention happens.

Algorithm 1 The BSMD scheduling procedures

Initialization: $t = 0$, $\beta_0 = 0$, $C_f = 0$, $\alpha_k = 0 \& P_k = \mathbf{0}$ ($\forall k \in [1, N^2 - 2N + 2]$);
while $t \leq t_{end}$ **do**
  if $C_f == 0$ **then**
    $\alpha_i = 1/N (\forall i \in [1, N])$;
    $P_i = \mathbf{I}_N^i$;
    $C_f = C_f + 1$;
  if $mod(t, T_f) + RTT \leq T_f$ && $mod(t + T_s, T_f) + RTT > T_f$ **then**
    $t_0 = t$; ①
    The BMs send buffer status to FOS;
  if $t = t_0 + T_l$ **then**
    The FOS starts to decompose $\mathbf{B}$; ②
  if $t = t_0 + T_l + \delta_T$ **then**
    The FOS sends α_k and P_k to ToRs; ③
  if $t = t_0 + RTT$ **then**
    Update α_k and P_k in each ToR; ④
    $C_f = C_f + 1$;
  if C_f *is updated* **then**
    for $1 \leq i \leq K$ **do**
      $\beta_i = \beta_{i-1} + \alpha_i$;
      i++;
    $k = 0$;
    while $mod(t, T_f) > \beta_k \times T_f$ **do**
      k++;
  ToRs send the packets based on P_k.
  $t = t + T_S$

To facilitate the understanding of BSMD operation, we give an example of a cluster with 4 ToRs where each ToR is equipped with 1 TRX. Suppose at time of t_1, we have $\mathbf{B} = \begin{bmatrix} 0 & 15 & 10 & 20 \\ 15 & 0 & 20 & 10 \\ 25 & 10 & 0 & 10 \\ 5 & 20 & 15 & 0 \end{bmatrix}$ ($\mathbf{P}_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$, $\mathbf{P}_2 =$

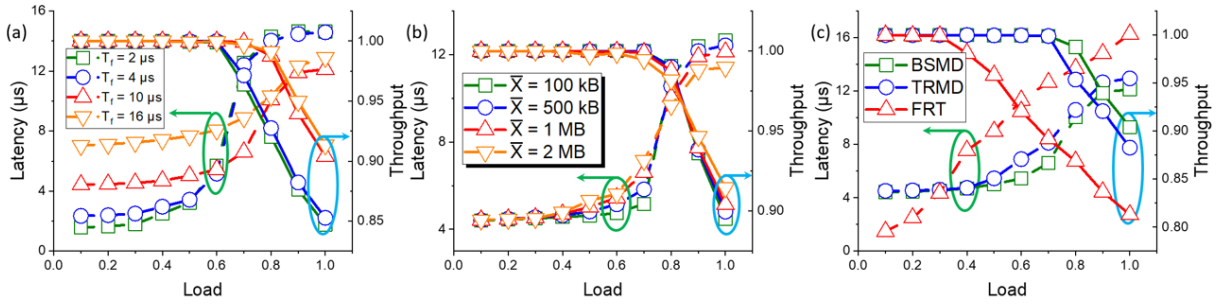

Fig. 4: The results of BSMD under (a) different frame lengths (b) various mean burst lengths (c) comparison with TRMD and FRT.

$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$, $\mathbf{P}_3 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$), and T_f/T_s is 50. Then we have $\mathbf{B} = 20\mathbf{P}_1 + 12\mathbf{P}_2 + 8\mathbf{P}_3$.

The time complexity of the BSMD is determined by BvN matrix decomposition ($O(N^{4.5})$). However, when each of the p TRXs is only responsible for a group of N/p ToRs, namely, the optical network is split into p groups. There will be no contention happening among different groups. The time complexity of BSMD algorithm could be decreased to $O((N/p)^{4.5})$. Given a linear increase of p with respect to N, a time complexity of $O(1)$ can be achieved for BSMD.

Simulation stup

The OMNeT++ platform is utilized to carry out simulations in a FOS based DCN supporting 256 servers equipped with 100Gb/s NIC (FOS radix of 8). Each server generates 10^5 packets independently based on ON/OFF Pareto distribution model[5]. The mean burst length of the ON period is set among 100KB and 2MB. Half traffic of 1600Gb/s resides inside the ToR, while the rest traffic destine to servers in other 7 ToRs with same probability.

There are 4 TRXs in each ToR, each TRX operates at 400 Gb/s. We take the optical packet size as 9600 Bytes comprising of 150 cells with the same destination ($c = 150$). A guardband of 8 ns including switching and optical packet preamble is considered resulting in an optical packet time slot T_s of 200 ns[6]. The link length between ToR and FOS is 50m resulting in link delay T_l of 250 ns. The δ_t is set as 10 ns.

Results and discussion

First, we investigate the performance of BSMD under different frame lengths T_f. We take T_f in the range of [2, 16] μs. The mean burst length $\overline{X}$ is set as 1MB. Figure 4(a) shows that the ToR to ToR latency of BSMD increases as the frame length increases when the load is $\leq$ 0.6. The reason is that the packets may wait a whole frame length if it is not scheduled in current frame. As the load is ≥ 0.7, the throughput decreases dramatically, the amount of packets starts to saturate the network load. Therefore, increasing the frame length improves the scheduling performance.

Secondly, Figure 4 (b) shows that the ToR to ToR latency of BSMD under various $\overline{X}$ ($T_f = 10\mu s$). Similar to the results shown in Fig. 4(a), as the load is $\leq$ 0.7, the latency of BSMD increases as $\overline{X}$ increases. At load of 0.6, the latency is 4.7 μs and 5.6 μs as $\overline{X}$ equals 100KB and 2MB, respectively. When the load approaches 1, the OFF periods of traffic shrink rapidly as the network becomes saturated.

The scheduling performance of BSMD is also compared with TRMD and FRT under T_f of 10 μs and $\overline{X}$ of 1MB. As shown in Fig. 4(c), as the load ≤ 0.3, the latency of FRT is the lowest due to the low contention and direct transmission of the arriving packets. However, as the load increases, the increasing contentions of FRT results in large amount of retransmissions. The performance of FRT deteriorates greatly, while the BSMD and TRMD can handle the traffic even with load $>$ 0.3. Moreover, the performance of BSMD outperforms TRMD due to the dynamic adaption to the traffic.

Conclusions

We proposed a novel scheduling mechanism BSMD for a FOS network based on the dynamic buffer status of ToRs. We demonstrate the operation of BSMD in the FOS network and also investigate the performance of BSMD under various mean burst lengths and frame sizes. Besides, comparing the performance of BSMD with TRMD and FRT, the results show that BSMD outperforms TRMD and FRT, and achieves latency of 10.1 μs and 98.8% throughput at load of 0.8.

Acknowledgements

The authors thank Alibaba Group through the Alibaba Innovative Research Program for partially supporting this work.

References

[1] Cisco, *Cisco annual internet report (2018–2023) white paper*, 2020.

[2] F. Testa and L. Pavesi, *Optical switching in next generation data centers*. Springer, 2017.

[3] F. Yan, C. Xie, and N. Calabretta, "Traffic rate matrix decomposition based conflict free scheduling for a fast optical switching network", in *OFC Conference*, 2021.

[4] C.-S. Chang *et al.*, "Birkhoff-von neumann input buffered crossbar switches", in *Proceedings IEEE INFOCOM 2000.*, IEEE, vol. 3, 2000, pp. 1614–1623.

[5] F. Yan, W. Miao, *et al.*, "Opsquare: A flat dcn architecture based on flow-controlled optical packet switches", *IEEE/OSA Journal of Optical Communications and Networking*, vol. 9, no. 4, pp. 291–303, 2017.

[6] H. Ballani, P. Costa, *et al.*, "Sirius: A flat datacenter network with nanosecond optical switching", in *ACM SIGCOMM*, 2020, pp. 782–797.