Performance evaluation of TCP over software-defined optical burst-switched data centre network

Muhammad Imran , Martin Collier , Pascal Landais , Kostas Katrinis

In this paper, we consider the performance of TCP when used in data centre networks (DCNs) featuring optical burst switching (OBS) using two-way reservation. The two-way reservation is not suitable in wide-area OBS networks due to high bandwidth-delay product (BDP). The burst loss using traditional methods of one-way reservation can be mistakenly interpreted by the TCP layer as congestion instead of contention in OBS network, leading to serious degradation of the TCP performance. The reduced BDP in DCNs allows the use of two-way reservation that results in zero burst loss. The modelled architecture features fast optical switches in a single hop topology. We apply different workloads with various burst assembly parameters to evaluate the performance of TCP. Our results show significant improvement in TCP performance as compared to traditional methods of OBS as well as to a conventional electronic packet switching DCN.

1. Introduction

Optical networks for data centres have gained significant attention over the last few years due to the potential and benefits of using optical components. Conventional electronic packet switch-ing DCNs are not power efficient due to power hungry transceivers and electronic switches. They are also unable to meet with higher bandwidth demands and are not scalable while on the other hands, optical interconnects are power efficient and can provide huge bandwidths. They are also scalable and can offer low latency and high throughput. The performance of optical network is directly related to the type of the optical switching technique used. These switching techniques are optical circuit switching (OCS), optical packet switching (OPS) and optical burst switching. In OCS, a connection is established before actual data transmission on a pre-established dedicated path from the source to the destination [1]. Long connection establishment time and bandwidth underutilizations in the case of low traffic load are the major limitations of the OCS. The microelectromechanical system (MEMS) optical cross connect (OXC) or OCS switch has been used in the backbone optical network for many years.

Hybrid designs for data centre networks that use OCS in conjunction with other technologies have been proposed [2–7]. Through [2,3] propose using OCS in conjunction with traditional electrical packet switching (EPS) while the LIGHTNESS project [4,5] employs OCS together with OPS. The Hydra, OSA and Reconfig-urable designs [6–8] augment OCS with a multi-hopping technique. Although these designs are cost effective but a major issue with these interconnects has been their slow reconfiguration time due to the limitation of 3D-MEMS technology. This reconfiguration time is influenced by two factors: (1) the switching time of the 3D-MEMS switch i.e. 10–100 ms, and (2) the software/control plane overhead required for the estimation of traffic demand and the calculation of a new OCS topology i.e. 100 ms to 1 s. Consequently, the control plane can only support applications that have high traffic stability, i.e. workloads that last several seconds [2].

In OPS, a packet consists of a data and a header portion which are in the optical domain. When the packet arrives at the node, the header is removed from the packet and is converted into the electrical domain for processing. During this processing time, the data in the packet has to be buffered in the node. Fibre delay lines (FDLs) are used for this purpose which can provide limited buffer-ing by routing the light to the FDLs ring. The packet is dropped if the switch is not configured within this time. The OPS drawbacks are lack of feasible optical buffer and packet loss due to output port contention. Speed of header processing should also be compatible with the data rate, otherwise packet loss occurs. This problem becomes significant at higher data rates because header processing speed might not be compatible with the higher data rates. The OPS for DCNs has been described recently in some studies [9–18]. The OPS can only be used with fast optical switching technologies that are now available [19,20,10,21]. The inherent drawbacks of OPS make these designs difficult to meet future requirements of DCNs.

OBS [22] is different from other techniques and is considered as a compromise between OCS and OPS. It has separate control and data planes similar to OCS. Packets are aggregated into bursts. A control packet is then transmitted on a dedicated control channel to reserve resources on all intermediate nodes from the source to the destination. The burst is sent at a particular time after sending the control packet which is called the offset time. During the off-set time, these bursts are temporarily stored at edge node before transmission. During this time, the switch controller at the core node processes the control information and sets up the switching matrix for the incoming burst. Burst loss due to output port con-tention is the major limitation of the OBS network. Output port contention can occur due to unavailability of a wavelength at the desired output port for the incoming burst. Several techniques exist in the literature to avoid contention such as FDLs, deflection rout-ing, wavelength conversion and segmentation based dropping but none of them can guarantee zero burst loss. OBS with two-way reservation protocol also known as tell and wait protocol ensures zero burst loss [22] in which a control packet reserves resources in all nodes from the source to the destination and is sent back to the source as an acknowledgement. The control packet has a high round trip time (RTT) for a large optical network.

In this paper, we extend our recent work by evaluating the performance of TCP over optical burst-switched data centre network using network-level simulation [23]. The performance of TCP over OBS network is degraded by the wrong interpretation of congestion in the network. The contention induced losses can be misinterpreted by the congestion induced losses. The contention refers to the burst loss due to unavailability of a wavelength even at the low network load. We implement OBS with a two-way reservation protocol to ensure zero burst loss. The two-way reservation is not suitable for long haul backbone optical networks due to the high RTT of the control packet and high bandwidth delay product but for our optical interconnect for the data centre network, this RTT is not high for several reasons: (1) the propagation delay is negligible; (2) faster optical switches are used at the core; (3) a fast optical control plane is used; (4) processing of the control packet is rapid and (5) a single hop topology is used [24]. The reduced RTT of the control packet results in lower bandwidth-delay product in DCNs. We use various

workloads with different burst assembly parameters to explore TCP performance with two-way reservation and compare its performance with conventional methods of one-way reservation of OBS networks. We also evaluate and compare the performance of TCP in the proposed scheme with the conven-tional electronic packet switching DCN. Our results show significant improvement of TCP performance in terms of throughput, time and packets loss as compared to the traditional methods of OBS. The proposed scheme also demonstrates efficient TCP performance than the conventional electronic packet switching DCN for all types of workloads. The remainder of this paper is structured as follows. Section 2 describes TCP over OBS. In Section 3, we present imple-mentation technique of OBS for data centre networks. We discuss performance evaluation in Section 4 and results in Section 5. We conclude in Section 6.

2. TCP over OBS

Burst loss and delay caused by the burst assembly and FDLs are the important features of the OBS that have great impact on the performance of the TCP. The burst loss can be misinterpreted as congestion in the network instead of contention. The timeout triggered by the contention is termed as False Time Out (FTO). After FTO, TCP sender starts with slow start mechanism which ultimately decreases network throughput. In most cases, several packets from different TCP sessions are included in a burst and the burst drop could result in loss of many packets per session, resulting in a network wide drop in throughput [25]. Another factor affecting performance is the delay that a packet experiences during the burst assembly process before its associated burst is transmitted and also in the FDLs ring if the burst is routed through in it during contention. If this accumulative delay is higher than retransmission timeout (RTO), then TCP senders start again with slow start resulting in decrease of throughput. TCP over OBS networks also suffers from a problem known as the high bandwidth delay product (BDP). The BDP determines the amount of data that can be sent over the network without being acknowledged. TCP throughput is bound by the BDP. If the TCP sender window is smaller than the BDP, there is a waste of the link capacity, and the TCP sender will be idle most of the time.

In order to overcome issues of TCP over OBS, several techniques have been presented in literature [26–37]. The authors in [26] evaluated the impact of burst assembly algorithms on different TCP implementations such as TCP Reno, New-Reno and SACK in OBS network and in other work [27], they proposed TCP implementation for OBS network called Burst TCP which tries to detect false time out and reacts properly. In [28], authors introduced burst retransmission scheme in which the bursts lost due to contention in the OBS network are retransmitted at the edge node. High Speed-TCP (HS-TCP) is a modification to TCP's window increase and decrease algorithm that allows it to run efficiently on networks with large BDP [33]. The authors in [29] evaluated the behaviour of high-speed TCP in OBS networks. Other technique [36] proposed modification in the burst assembly period at the edge node that aggregates packets from different TCP sessions into different bursts. In [30], authors presented TCP Vegas implementation using a threshold-based mechanism to identify network congestion under burst retransmission scheme. Source-ordering technique over a load-balanced OBS network is introduced by [31] to avoid false time out. In [35], authors presented predictive techniques in OBS that tries to improve TCP performance. TCP is very sensitive to the packet/burst losses and this has been shown in empirical study of different TCP implementations in OBS [34]. The authors in [32] presented a protocol level technique for estimating assembly time at the TCP end points and use this information to differentiate congestion induced loss or contention induced loss. The authors in [37] introduce a new layer between TCP and OBS layers for burst retransmission to mitigate the effect of burst loss due to contention on TCP performance. The new layer requires modification in ingress nodes to adapt the functionality of a new layer.

We observe through our literature survey that burst loss in TCP over OBS is still the major issue because techniques proposed so far either require modifications in protocol level or they are designed to implement in ingress nodes which is a difficult task. These techniques can help to improve TCP performance in OBS network but none of them can ensure zero burst loss due to contention.

3. OBS for data centre network

We employ OBS in the proposed data centre network architec-ture. The packets are aggregated to create bursts of short duration. A control packet is created to request the allocation of resources needed to transmit the burst from the controller by using a two-way reservation process similar to that proposed for optical burst switching networks [22]. Although such two-way reservation is not feasible in a long haul backbone network, in data centres it is suitable for the reasons presented earlier. The controller assigns resources and sends the control packet back to the originating node as an acknowledgement. The burst is then transmitted on the pre-established path configured by the controller.

The proposed architecture of OBS for data centre network is shown in Fig. 1. We use a two layer topology comprising Top of the Rack (ToR) switches at the edge and array of fast optical switches at the core similar to our hybrid design [38]. The ToR switches are electrical switches connecting servers in a rack using bidirectional fibre links. The ToR switches are also interfaced with the optical switches using unidirectional fibre links. Each ToR switch has X optical transceivers which are linked with the optical switches and is also termed as the degree of the ToR switch. If we consider N the total number of ToR switches in the network, then (N × N) is the minimum configuration for optical switches so that at least one port from all ToR switches connects to every (N × N) optical switch. A management network which consists of electrical switch/switches is used to connect every ToR switch to the controller using bidirec-tional fibre links. It also connects controller to fast optical switches using unidirectional fibre links.

The proposed design features separate control and data planes. The control plane comprises a centralized controller. The controller performs routing, scheduling and switch configuration functions. It receives connection setup requests from all ToR switches, finds routes, assigns timeslots to the connection requests, and config-ures optical switches with respect to the timeslots allocated. In order to perform these tasks, the controller keeps a record of the connection states of all optical switches. The data plane comprises optical switches that perform data forwarding on pre-configured lightpaths set up by the controller.

3.1. Scalability

Fast optical switch using semiconductor optical amplifiers (SOAs) as a switching fabric with 1024 ports has been proposed [10] while 512 ports using arrayed waveguide grating routers (AWGRs) as a switching fabric is also feasible [19]. Table 1 describes scalability analysis of the proposed topology using both AWGRs and SOAs as a fast optical switch. By considering SOAs as a switching fabric, the system size of 40,960 servers with 40 servers per rack and 81,920 servers with 80 servers per rack can be achieved with the proposed single stage topology without converting to multi-stage core topologies. Similarly, if we consider a pod switch instead of the ToR switch that has the capacity to integrate several ToR switches into a single unit and can aggregate a few hundreds to thousand servers [2], leads to the scalability up to 245,760 servers by considering 240 servers per pod which is ideal for future large scale data centres. The system size of 122,880 can also be achieved using AWGRs as a switching fabric by considering 240 servers per pod.

In our recent work [38], we present cost and power consumption analysis of the proposed design and its comparative analysis with electronic DCN. In [38], we have shown that the CAPEX cost of the proposed design is much higher than the conventional electronic DCN but this CAPEX cost is mitigated to some extent in the long run by its reduced OPEX cost due to its greater energy efficiency. In the foreseeable future, it is most likely that the CAPEX cost will fall driven by advances in the technology. For instance, we can mention the possibility of integrated photonics leading to fast switches being in mass production and their integration to complementary metal oxide semiconductor (CMOS) circuits.

3.2. ToR switch design

The ToR switch design is shown in Fig. 2a. The ToR switch has an electronic switch fabric which is connected to the servers in the rack to perform intra-rack (within rack) switching in the electrical domain. To perform inter-rack (between racks) switching, we employ (N − 1) virtual output queues (VOQs) where N is the number of ToR switches in the network. State of the art ToR switches support hundreds of VOQs. For example, the Cisco Nexus 5500 supports up to 384 VOQs, the Cisco 5548P supports up to 18,432 VOQs and the Cisco 5596 supports up to 37,728 VOQs [39,40]. There is a VOQ for each destination ToR switch in the DCN. Packets destined to the same ToR are aggregated into the same VOQ. The VOQ not only aggregates traffic to the same destination ToR switch but it also avoids head of line blocking (HOL). Each VOQ is configured for a destination network address. Each ToR switch maintains a VOQ table where entries comprise the destination rack network address and the VOQ number. The dispatcher module matches the destination network address of the packet with the entry in this table and forwards the packet on the required VOQ. The similar design of edge node for hybrid optical switching using multiple virtual queues for packets, short bursts, long bursts and circuits is also proposed in [41].

3.3. Control packet format

The format of the control packet is shown in Fig. 2b. The control packet is 440 bits long and contains two main fields, routing and reservation. The routing field contains IP address of the source ToR switch, IP address of the controller, and IDs of the source and destination ToR switches. IPv6 addresses take 128 bits, however this length can be reduced to 32 bits by using IPv4 addresses, that will reduce overall control packet length to 31 bytes. The reservation field is 96 bits long, and is divided into 3 sub-fields: (1) Burst length, (2) start time and (3) port number. The burst length field is filled by the ToR switches to request a timeslot from the controller. The controller fills rest of the two fields after processing the control packet. All of these three fields are 4 bytes long. The burst length field contains size of the burst expressed in bytes while the start time field contains time when the burst will be sent and the port number is the port of the ToR switch in which the burst is to be sent. IP address is used by the electrical switches to route the control packet from the ToR switch to the controller and back from the controller to the ToR switch in the management network. Whereas ID is used in the routing and scheduling algorithm which is explained in Section 3.5 Control packet processing and in Algorithm 1.

CRC and flags are optional fields that are set by the IP/network layer in ToR switches/controller. Since we used OMNeT++ simulation framework, we did not set these fields. These are automatically handled by the network layer. The proposed algorithms in ToR switches and in the controller do not care about these fields.

3.4. Traffic aggregation

Burst assembly can be timer based, length based or a mix of both [22,42]. We consider the mixed approach in which either a timer expires or the burst length exceeds a threshold. The timer starts when a packet arrives at the empty VOQ. If the VOQ is not empty when the packet arrives, it joins other packets in the VOQ. The control packet is generated after the timer expires or the burst length exceeds a threshold and is sent to the controller using transceiver dedicated for the control plane. The controller processes the control packet and sends it back to the source ToR switch. When the control packet arrives at the ToR switch, the scheduler module of the ToR switch generates a burst according to size of the burst length specified in the control packet. The generated burst is then sent to the queue of the allocated port of the ToR switch. The scheduler module also initiates a new timer if the VOQ is not empty after burst generation because new packets might have arrived during the round trip time of the control packet.

The ToR switch also has burst disassembler and packet extractor module to disassemble the bursts received through the receivers. The receivers perform O–E conversion and send bursts to the dis-assembler module where packets are extracted from them and are sent to the electronic switch fabric and finally to the destination servers using electrical switching.

Burst assembly cycle is shown in Fig. 3. Packets are aggregated to make a burst. Burst assembly time is represented by Ta. The control packet is sent by the ToR switch to the controller for resource reservation by using a management network. The time control packet takes to reach the controller is called overhead time and is represented by Toh. The Toh includes its propagation delay, O–E–O conversion delay, processing and queuing delay at electrical switch and its transmission delay. The controller processes the control packet and assigns a timeslot on an optical switch path. The time controller takes to process the control packet is denoted by Tproc. The control packet is sent back to the ToR switch and it again takes Toh to arrive at the ToR switch. The time difference when it arrives back at the ToR switch and when it was departed from the ToR switch is called round trip time (RTT). After process-ing the control packet, a configuration message is also generated by the controller to configure the optical switch. The configuration message also takes Toh to reach at the optical switch. The Tsw is the time that switch takes to configure an optical switch path and is called switch configuration time. In the end, the burst is transmitted at the assigned timeslot on an optical switch path. The time burst takes for the transmission is denoted by Ttran. The length of the Ttran depends upon the size of the burst and data rate of the channel. As the data rate increases, the length of the Ttran decreases for the same size of burst. As soon as the control packet is sent by the ToR switch, subsequent burst assembly process is also started and this cycle repeats as long as there is traffic.

3.5. Control packet processing

We consider horizon scheduling that was proposed for OBS net-work [22]. The term horizon refers to the latest available time when the channel will be free. There are also some other scheduling techniques in OBS networks but we consider this technique due to its efficiency and implementation simplicity. Our goal is to use a technique that fulfils our objective of the fast

optical control plane. The horizon scheduling is explained with the help of Fig. 4. Suppose we have 5 channels on which an incoming burst can be scheduled. Fig. 4a shows the states of the channels when a control packet arrives at the controller. The horizon scheduling uses a minimum value method to find a latest available channel. Fig. 4b shows the states of the channels after allocating a timeslot for the incoming burst.

The controller keeps a record of the connections of all optical switches. It performs routing, scheduling and switch configuration operations. These operations are depicted in Algorithm 1. There are two data structures which are used to maintain record of horizons of input and output ports (lines 1–2). The controller gets source and destination IDs of ToR switches from the control packet that arrives at the controller (lines 3–5). The controller performs routing operation by using a technique to find a minimum value method in input and output horizons (lines 6–30). Two values of horizons one each for input and output ports are initialized to a maximum value (lines 6–7). There are two inner loops that calculates the input and output horizons i.e. lines 11–16 and lines 17–22. This is a simple operation to get a minimum value. After calculating input/output horizons, maximum value from them is selected and is assigned to input and output horizons (lines 23–28). This procedure continues until all optical switches are traversed i.e. outer loop in line 8. The routing operation results in finding optimal input/output ports and their relevant horizons. Scheduling is the next operation that assigns a timeslot on the selected input/output ports (lines 32–46). The length of the times-lot is calculated from the burst length field in the control packet (lines 35–36). The Tstart and Tend represent the start and end time of the timeslot (lines 37–38). The Tsw is the switching time of the optical switch, Tproc is the processing time of the control packet at the controller, Toh is the aggregate time that a control packet spends in control plane as discussed earlier. We also consider a guard time (Tguard) in the timeslot to avoid synchronization problems. The horizons on the selected input and output ports are updated with a new
time (lines 39–40). The controller updates the control packet by assining the start time and the port number in it (lines 41–42). It then swaps the source and destination IP addresses in the control packet and sends it back to the source ToR switch (lines 43–46).

Switch configuration is the final task of the controller. After pro-cessing the control packet, a configuration message is generated (line 47). The controller sets fields such as input port, output port and the time at which a switch will be configured. It also fills source IP address of the controller and destination IP address of the optical switch. In the end, the configuration message is sent to the switch controller for optical switch configuration. The switch controller configures the optical switch according to the instructions in the configuration message.

4. Performance analysis

To assess the performance of TCP over OBS for data centre network, we developed simulations models in the OMNeT++ simulation framework [43]. We use inet models of OMNeT++ to simulate behaviour of TCP, servers, electrical switches while we develop models for the ToR switches, the controller and the optical switches. Important simulation parameters are presented in Table 2. Our simulation model consists of 24 ToR switches. Each ToR switch has 40 servers connected to it. The controller and ToR switches are connected to the management network via an electrical switch. We use one fast switch which is interfaced to the management network.

We consider a bijective traffic model in which the number of TCP flows a server generates is equal to the number of TCP flows the server receives. We use 40 TCP flows per server equal to the size of number of servers in a rack. Each TCP flow in a server sends 25 MB data to another server, so each server sends total 25 × 40 = 1 GB data to other servers. We consider three cases of topological degree of communication (TDC) to investigate the traffic diversity work-load. The TDC represents rack level flows i.e. TDC = 1 means that servers in a rack send data traffic to servers in only 1 destination rack while TDC = 4 reveals that each server in a rack send data traffic to 10 servers each in four racks (total 40 servers in 4 racks). Similar method is used for TDC = 8 (i.e. each server in a rack sends 5 TCP flows to 5 servers each in eight racks). The TDC = 1 shows low diversity workloads, TDC = 4 shows medium diversity workloads while TDC = 8 shows high diversity workloads. We consider asymmetric traffic, which means that the servers in rack A send data to the servers in rack B, the servers in rack B send data to the servers in rack C while the servers in rack B and C only send ACKs to the servers in rack A and B respectively. We use TCPReno models for TCP implementation available in OMNeT++ inet models [43].

We use a value of 1 μs for the switching time of the optical switches because this is a conservative choice, although in some types of fast optical switch this value can be as low as few nanoseconds [11,10]. The RTT of the control packet includes its processing time at the controller (Tproc) and twice of the overhead time (Toh). The aggregate value of Toh is conservatively set

to 1 s although all these delays are negligible (at most a few nanoseconds [7]). We choose a value of 1 s for Tproc. The value of Tproc is compatible with its actual value that we measure in Section 5.1.

We consider four cases for traffic aggregation by using values of aggregation drawn from the set {{50 μs,50 KB}, {50 μs, 100 KB}, {100 μs, 50 KB}, {100 μs, 100 KB}}. We conservatively consider a buffer size of 1000 packets (i.e. 1.5 MB) per electronic port/VOQ in ToR switches and switches in electronic DCN while state of the art switches can support a much higher buffer size [39,40]. The minimum value of buffer size should be greater than the maximum burst size (i.e. 100 KB at a 10 Gbps data rate).

5. Results and discussion

We examine the performance of TCP by measuring through-put, completion time, packets loss and round trip time of TCP segments. We compare TCP performance of our design using OBS with two-way reservation with OBS by using traditional methods of one-way reservation. We also evaluate the TCP performance of a conventional DCN based on the electronic packet switching using simulation and compare results with the proposed OBS scheme. We consider a two layer leaf spine topology for the electronic packet switching DCN. The detailed description of this topology is avail-able in our recent work [38]. The simulation results obtained are shown in Figs. 5–10.

Fig. 5 shows throughput performance achieved for three values of TDC across a range of burst assembly parameters in proposed design of OBS with two-way reservation schemes while Fig. 6 shows throughput performance using OBS with traditional methods of one-way reservation. Four curves in each plot of Figs. 5 and 6 rep-resent average throughput with respect to the time for different values of the burst aggregations. The burst loss of OBS with two-way reservation is zero for all values of TDC that leads to high throughput as shown in all plots of Fig. 5. But in the case of OBS with traditional one-way reservation, the performance is good only when TDC = 1. The performance is degraded with the increase of traffic diversity e.g. with TDC = 4 and TDC = 8 as shown in Fig. 6b and c. This is because burst loss in OBS with one-way reservation is negligible when TDC = 1 as all bursts are going to only one rack and the requests for burst reservations comes in an order. But in case of higher TDC values, burst loss increases with the increase of TDC because of a large number of overlapping control packet requests come. The network observes some throughput initially as shown in Fig. 6b but after half of a second, throughput drops and servers go into slow start. In the case of TDC = 8 as shown in Fig. 6c, the throughput is minimum due to the large number of bursts loss. Similar trend of decrease in the throughput with the increase of TDC values is observed in the electronic packet switching DCN as shown in Fig. 7a. The packets are lost in the electronic network due to buffer overflow both at edge and core nodes which results in decrease of throughput. The throughput achieved in the proposed scheme is better than traditional methods of OBS and electronic DCN.

Fig. 8 shows the time taken by all servers to send 1 GB data traffic to other servers for three values of TDC across a range of burst assembly parameters in both designs. It can be observed that servers only take couple of seconds to complete data trans-fer in our design but in case of traditional methods, they take tens of seconds (greater than 20 s) to complete data transfer. This is because when packets are lost, servers go into slow start after retransmission timeout. This process continues as long as there are packets loss. Due to going down to slow start phase again and again, the servers are not able to utilize their full capacity. So they take longer time to send similar amount of traffic as compared to the time they take using two-way reservation. While in the elec-tronic network, the servers take 5–8 s to complete the data transfer as shown in Fig. 7a which is better than using traditional meth-ods of OBS but almost three times as compared to the proposed scheme.

Fig. 9 shows the packets loss for three values of TDC across a range of burst assembly parameters in both designs. Packets are also lost in our design due to the buffer overflow of network interface card (NIC). We consider NIC size equal to 500 packets. In OBS with traditional methods, packets are not only lost when the burst is dropped but also due to the buffer overflow. In all three values of TDC, packet loss is negligible in our design, while in traditional methods, packet loss increases with the increase of TDC as shown in Fig. 9b and c. This is due to the higher number of bursts loss with the high diversity traffic workload. In the electronic DCN, the packet losses are better than traditional methods of OBS but are slightly higher than the proposed scheme for all types of workloads as shown in Fig. 7b.

Fig. 10 shows the average round trip time (RTT) of TCP segments for three values of TDC across a range of burst assembly parameters in both designs. The round trip time is measured when a TCP segment is sent by the TCP application in a source server and its acknowledgement is received at the source server. The RTT in all three cases of TDC across different burst aggregation schemes in our design is around 500 s. It can be noticed that the aggregation time that we consider is only 50 s and

100 s while the RTT is around 500 s. This is because, TCP segments also spend some time in the queue of NIC in server and ToR switches. When there is high traffic, these segments have to wait in the queue to get transmitted. This is not the case in traditional methods of OBS using one-way reservation. For example, with TDC = 4 and TDC = 8 as shown in Fig. 10b and c, the RTT in most of the cases is around 200 s because there is not much traffic to send due to lower throughput observed in traditional mechanisms of OBS. So, TCP segments find it easy to go through the queue of NIC in low traffic that results in lower RTT. The RTT in the electronic network is higher than traditional and the proposed scheme as shown in Fig. 7b. This is because of additional delay which is occurred at the core node while in traditional and proposed methods of OBS delay is occurred only at the edge node.

5.1. Performance of the control plane

In order to assess the performance of the control plane, we run our algorithm on an Intel host with a Core i7, 2.17 GHz processor and 16 GB RAM. The results were obtained for several combinations of parameters. To ensure statistical significance, we averaged the results of 1,000,000 runs and the results are shown in Table 3. When a control packet arrives at the controller, the controller performs the routing, scheduling and switch configuration operations described in Algorithm 1. The complexity of the routing and scheduling algorithm is O(2X +), where X is the degree of ToR switches and represents the sum of processing time of all other instructions. The complexity of the switch configuration operations is O(P +) where P is the total number of optical switches. The is assumed to be a constant of negligibly low value. We measure the execution time in fully subscribed, 2:1 oversubscribed and 4:1 oversubscribed networks using 40 servers per rack as shown in Table 3. It can be noticed in Table 3 that the execution time of rout-ing and scheduling operations is in nanoseconds scale for all types of networks. It is minimum in 4:1 oversubscribed network but it increases slightly as we decrease network oversubscription. Simi-larly, the execution time of the switch configuration operations is minimum when P is minimum and it increases slightly with the increase of the number of optical switches. The overall execution time of switch configuration operations is negligible (at most a few nanoseconds). So our algorithms in the control plane demonstrate efficient performance for all types of network oversubscriptions.

6. Conclusion

We investigated the performance of TCP over optical burst-switched data centre network by using network-level simulation. Burst loss is the major limitation of traditional OBS that degrades the performance of TCP by misinterpretation of network congestion. We implement OBS with two-way reservation to get zero burst loss. Two-way reservation is not appropriate for traditional back-bone optical networks due to the high RTT of the control packet and high bandwidth delay product but in a data centre network, this RTT is not high. We examine different burst assembly parameters with differ-ent traffic workloads to evaluate the performance of TCP. Our results reveal significant improvement in TCP performance in terms of throughput, time and packets loss as compared to traditional methods of OBS for all types of workloads. The proposed scheme also demonstrates efficient TCP performance than the conventional electronic packet switching network for all types of workloads.

References

[1] K.J. Barker, A. Benner, R. Hoare, A. Hoisie, A.K. Jones, D.K. Kerbyson, D. Li, R. Melhem, R. Rajamony, E. Schenfeld, et al., On the feasibility of optical circuit switching for high performance computing systems, in: Proceedings of the 2005 ACM/IEEE Conference on Supercomputing, IEEE Computer Society, 2005, p. 16.

[2] N. Farrington, G. Porter, S. Radhakrishnan, H.H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, A. Vahdat, Helios: a hybrid electrical/optical switch architecture for modular data centers, ACM SIGCOMM Comput. Commun. Rev. 41 (4) (2011) 339–350.

[3] G. Wang, D.G. Andersen, M. Kaminsky, K. Papagiannaki, T.S. Ng, M. Kozuch, M. Ryan, c-Through: part-time optics in data centers, ACM SIGCOMM Comput. Commun. Rev. 40 (2010) 327–338.

[4] W. Miao, F. Agraz, S. Peng, S. Spadaro, G. Bernini, J. Perelló, G. Zervas, R. Nejabati, N. Ciulli, D. Simeonidou, et al., SDN-enabled OPS with QoS guarantee for reconfigurable virtual data center networks, J. Opt. Commun. Netw. 7 (7) (2015) 634–643.

[5] S. Peng, B. Guo, C. Jackson, R. Nejabati, F. Agraz, S. Spadaro, G. Bernini, N. Ciulli, D. Simeonidou, Multi-tenant software-defined hybrid optical switched data centre, J. Lightwave Technol. 33 (15) (2015) 3224–3233.

[6] K. Christodoulopoulos, D. Lugones, K. Katrinis, M. Ruffini, D. O'Mahony, Performance evaluation of a hybrid optical/electrical interconnect, IEEE/OSA J. Opt. Commun. Netw. 7 (3) (2015) 193–204.

[7] K. Chen, A. Singla, A. Singh, K. Ramachandran, L. Xu, Y. Zhang, X. Wen, Y. Chen, OSA: an optical switching architecture for data center networks with unprecedented flexibility, IEEE/ACM Trans. Netw. 22 (2) (2014) 498–511, doi:10.1109/TNET.2013.2253120.

[8] D. Lugones, K. Katrinis, G. Theodoropoulos, M. Collier, A reconfigurable, regular-topology cluster/datacenter network using commodity optical switches, Future Gener. Comput. Syst. 30 (2014) 78–89.

[9] O. Liboiron-Ladouceur, P.G. Raponi, N. Andriolli, I. Cerutti, M.S. Hai, P. Castoldi, A scalable space-time multi-plane optical interconnection network using energy-efficient enabling technologies [invited], IEEE/OSA J. Opt. Commun. Netw. 3 (8) (2011) A1–A11.

[10] O. Liboiron-Ladouceur, I. Cerutti, P.G. Raponi, N. Andriolli, P. Castoldi, Energy-efficient design of a scalable optical multiplane interconnection architecture, IEEE J. Sel. Top. Quantum Electron. 17 (2) (2011) 377–383.

[11] Y. Yin, R. Proietti, X. Ye, C.J. Nitta, V. Akella, S. Yoo, LIONS: an AWGR-based low-latency optical switch for high-performance computing and data centers, IEEE J. Sel. Top. Quantum Electron. 19 (2) (2013) 3600409.

[12] G. Wu, H. Gu, K. Wang, X. Yu, Y. Guo, A scalable AWG-based data center network for cloud computing, Opt. Switch. Netw. 16 (2015) 46–51.

[13] P.N. Ji, D. Qian, K. Kanonakis, C. Kachris, I. Tomkos, Design and evaluation of a flexible-bandwidth OFDM-based intra-data center interconnect, IEEE J. Sel. Top. Quantum Electron. 19 (2) (2013), 3700310–3700310.

[14] G. Porter, R. Strong, N. Farrington, A. Forencich, P. Chen-Sun, T. Rosing, Y. Fainman, G. Papen, A. Vahdat, Integrating Microsecond Circuit Switching into the Data Center, vol. 43, ACM, 2013.

[15] O. Liboiron-Ladouceur, A. Shacham, B.A. Small, B.G. Lee, H. Wang, C.P. Lai, A. Biberman, K. Bergman, The data vortex optical packet switched interconnection network, J. Lightwave Technol. 26 (13) (2008) 1777–1789.

[16] Q. Yang, Latency-optimized high performance data vortex optical switching network, Opt. Switch. Netw. 18 (2015) 1–10.

[17] J. Wang, C. McArdle, L.P. Barry, Optical packet switch with energy-efficient hybrid optical/electronic buffering for data center and HPC networks, Photonic Netw. Commun. (2015) 1–15.

[18] B. Rahimzadeh Rofoee, G. Zervas, Y. Yan, D. Simeonidou, Griffin: programmable optical datacenter with SDN enabled function planning and virtualisation, J. Lightwave Technol. 33 (24) (2015) 5164–5177.

[19] K. Takada, M. Abe, M. Shibata, M. Ishii, K. Okamoto, Low-crosstalk
10-GHz-spaced 512-channel arrayed-waveguide grating multi/demultiplexer fabricated on a 4-in wafer, IEEE Photonics Technol. Lett. 13 (11) (2001) 1182–1184.

[20] S. Aleksic, Analysis of power consumption in future high-capacity network nodes, IEEE/OSA J. Opt. Commun. Netw. 1 (3) (2009) 245–258.

[21] K. Nashimoto, D. Kudzuma, H. Han, High-speed switching and filtering using PLZT waveguide devices, in: 2010 15th OptoeElectronics and Communications Conference (OECC), IEEE, 2010, pp. 540–542.

[22] Y. Chen, C. Qiao, X. Yu, Optical burst switching: a new area in optical networking research, IEEE Netw. 18 (3) (2004) 16–23.

[23] M. Imran, M. Collier, P. Landais, K. Katrinis, Performance evaluation of TCP over optical burst-switched data center network, in: Proceedings of the 18th IEEE International Conference on Computational Science and Engineering, Porto, Portugal IEEE, 2015.

[24] M. Imran, M. Collier, P. Landais, K. Katrinis, Software-defined optical burst switching for HPC and cloud computing data centers, J. Opt. Commun. Netw. 8
(8) (2016) 610–620.

[25] S. Gowda, R.K. Shenai, K.M. Sivalingam, H.C. Cankaya, Performance evaluation of TCP over optical burst-switched (OBS) WDM networks, in: IEEE International Conference on Communications, 2003, ICC'03, vol. 2, IEEE, 2003, pp. 1433–1437.

[26] X. Yu, J. Li, X. Cao, Y. Chen, C. Qiao, Traffic statistics and performance evaluation in optical burst switched networks, J. Lightwave Technol. 22 (12) (2004) 2722–2738.

[27] X. Yu, C. Qiao, Y. Liu, TCP implementations and false time out detection in OBS networks, in: INFOCOM 2004, Twenty-Third Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 2, IEEE, 2004, pp. 774–784.

[28] Q. Zhang, V.M. Vokkarane, Y. Wang, J.P. Jue, Analysis of TCP over optical burst-switched networks with burst retransmission, in: IEEE Global Telecommunications Conference, 2005, GLOBECOM'05, vol. 4, IEEE, 2005, p. 6.

[29] L. Zhu, N. Ansari, J. Liu, Throughput of high-speed TCP in optical burst switching networks, IEE Proc. Commun. 152 (3) (2005) 349–352.

[30] B. Shihada, Q. Zhang, P.-H. Ho, J.P. Jue, A novel implementation of TCP Vegas for optical burst switched networks, Opt. Switch. Netw. 7 (3) (2010) 115–126.

[31] B. Komatireddy, N. Charbonneau, V.M. Vokkarane, Source-ordering for improved TCP performance over load-balanced optical burst-switched (OBS) networks, Photonic Netw. Commun. 19 (1) (2010) 1–8.

[32] S. Datta, A. Dutta, S. Choudhury, Design and analysis of a modified TCP for optical burst switched networks, in: 2014 2nd International Conference on Business and Information Management (ICBIM), IEEE, 2014, pp. 7–10.

[33] S. Floyd, 2003. Highspeed TCP for Large Congestion Windows, RFC 3649, The Internet Society.

[34] L. Liu, H. Guo, T. Tsuritani, Y. Yin, J. Wu, X. Hong, J. Lin, M. Suzuki, Dynamic provisioning of self-organized consumer grid services over integrated OBS/WSON networks, J. Lightwave Technol. 30 (5) (2012) 734–753.

[35] K. Ramantas, K. Vlachos, A TCP-specific traffic profiling and prediction scheme for performance optimization in OBS networks, J. Opt. Commun. Netw. 3 (12) (2011) 924–936.

[36] S. Peng, Z. Li, Y. He, A. Xu, TCP window-based flow-oriented dynamic assembly algorithm for OBS networks, J. Lightwave Technol. 27 (6) (2009) 670–678.

[37] N. Sreenath, N. Srinath, J.A. Suren, K. Kumar, Reducing the impact of false time out on TCP performance in TCP over OBS networks, Photonic Netw. Commun. 27 (1) (2014) 47–56.

[38] M. Imran, M. Collier, P. Landais, K. Katrinis, Performance evaluation of hybrid optical switch architecture for data center networks, Opt. Switch. Netw. 21 (2016) 1–15.

[39] Cisco nexus 5596, http://www.cisco.com/c/en/us/products/collateral/ switches/nexus-5548p-switch/white paper c11-622479.html (visited on 05.10.16).

[40] Cisco nexus 5548P, 5548UP, 5596UP, and 5596T switches data sheet, http:// www.cisco.com/c/en/us/products/collateral/switches/nexus-5000-series-switches/data sheet c78-618603.html (visited on 05.10.16).

[41] M. Fiorani, M. Casoni, S. Aleksic, Large data center interconnects employing hybrid optical switching, in: Network and Optical Communications (NOC), 2013 18th European Conference on and Optical Cabling and Infrastructure (OC&i), 2013 8th Conference on, IEEE, 2013, pp. 61–68.

[42] X. Cao, J. Li, Y. Chen, C. Qiao, Assembling TCP/IP packets in optical burst switched networks, in: IEEE Global Telecommunications Conference, 2002, GLOBECOM'02, vol. 3, IEEE, 2002, pp. 2808–2812.

[43] OMNeT++ simulation framework, http://omnetpp.org/ (visited on 05.10.16).

Figures

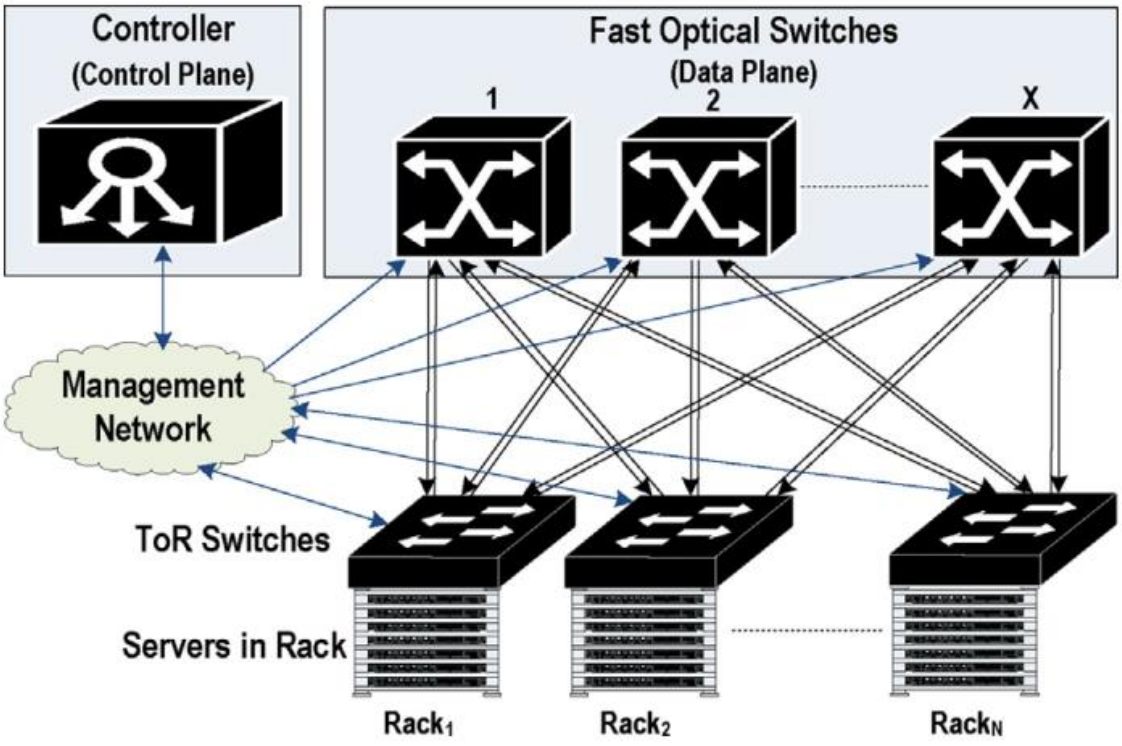
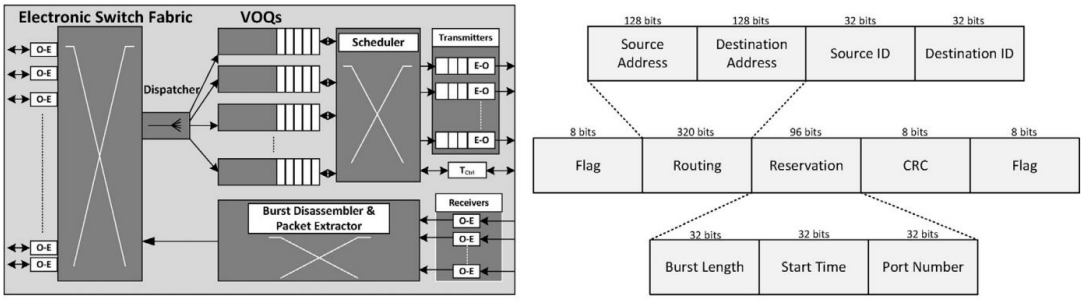


Fig. 1. Topology diagram for data centre network.



(a) ToR Switch Design

(b) Control Packet Format

Fig. 2. Design of ToR switches and format of the control packet.

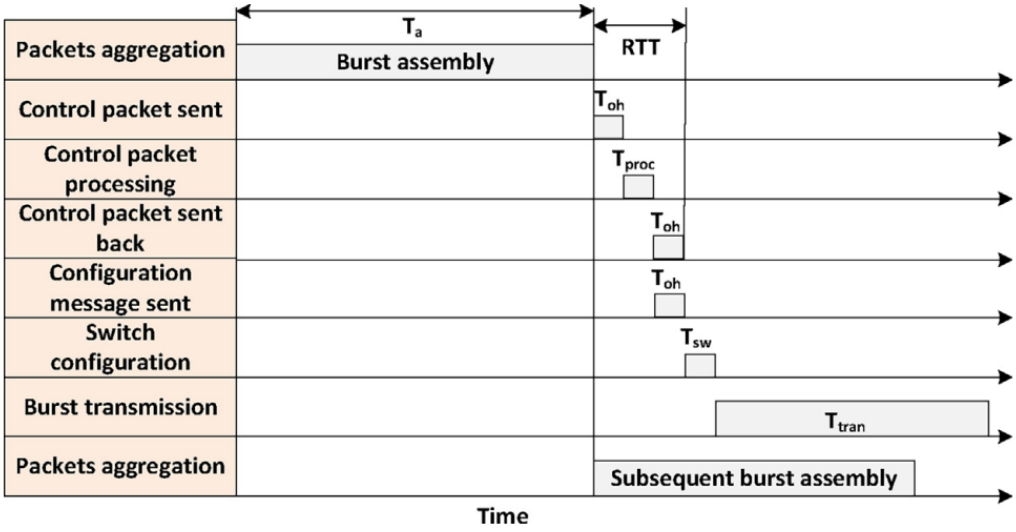
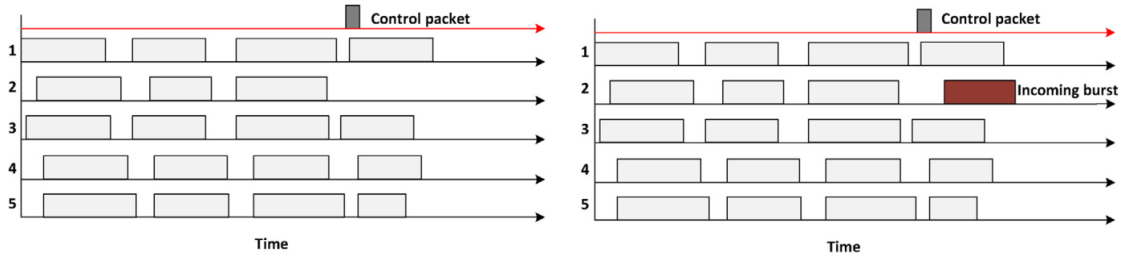


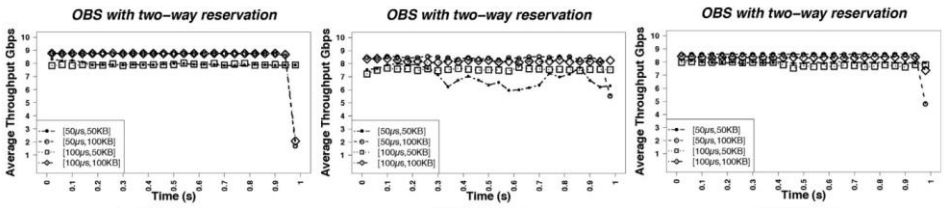
Fig. 3. Burst assembly cycle.



(a) Channels states before timeslot allocation

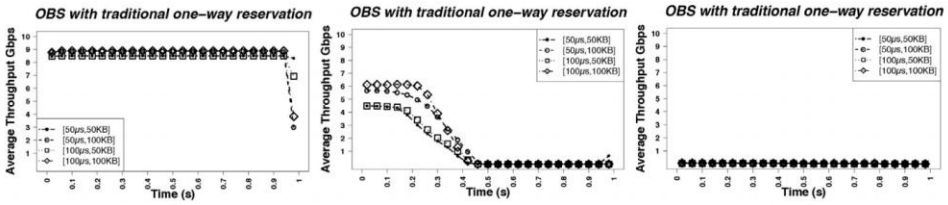
(b) Channels states after timeslot allocation

Fig. 4. Horizon scheduling.



(a) TDC=1 (b) TDC=4 (c) TDC=8

Fig. 5. Average throughput of proposed design using OBS with two-way reservation by considering different burst aggregation parameters and with respect to different TDC values.



(a) TDC=1 (b) TDC=4 (c) TDC=8

Fig. 6. Average throughput of OBS with traditional methods of one-way reservation by considering different burst aggregation parameters and with respect to different TDC values.

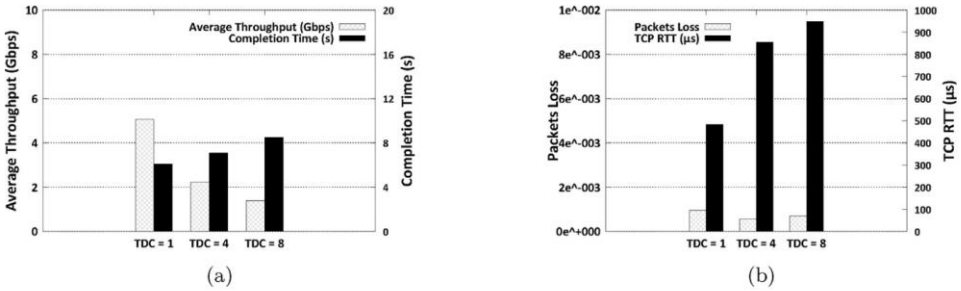
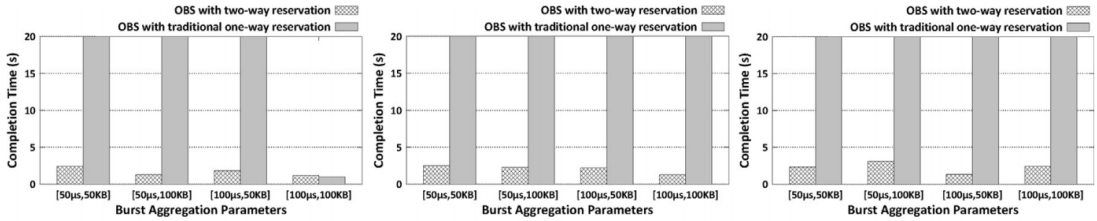
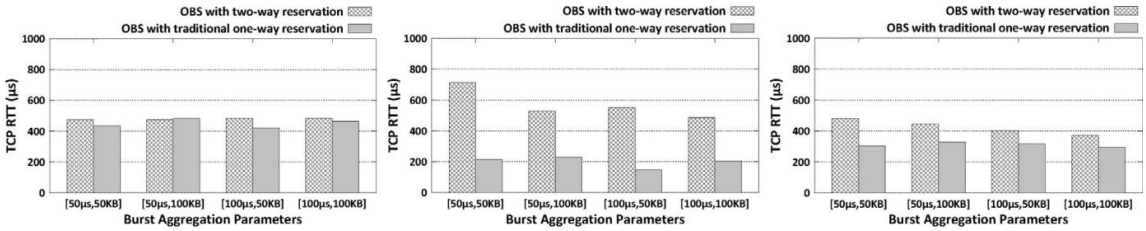


Fig. 7. Performance analysis of TCP over conventional electronic packet switching DCN with different values of TDC: (a) average throughput achieved during first second of simulation time and completion time to transfer 1 GB data from each server and (b) packets loss and average round trip time of TCP segments.



(a) TDC=1 (b) TDC=4 (c) TDC=8

Fig. 8. Completion time to transfer 1 GB data from each server with different burst aggregation parameters and with respect to different TDC values.



(a) TDC=1 (b) TDC=4 (c) TDC=8

Fig. 10. Average round trip time of TCP segments with different burst aggregation parameters and with respect to different TDC values.

Algorithms

Algorithm 1. Routing, scheduling and switch configurations.

1: *horizoninput*[*totalracks* × *ToRDegree*]
2: *horizonoutput*[*totalracks* × *ToRDegree*]
{**Above lines represent data structures of horizons for all inputs and outputs in optical switch paths.**}
3: *controlpacket* ⟵ *control packet arrives at the controller*
4: *srcID* ⟵ *controlpacket*. *getSrcId*()
5: *destID* ⟵ *controlpacket*. *getDestId*()
{**Above lines get source and destination IDs of ToR switches from the control packet arrives at the controller.**}
6: *minInputHorizon* ⟵ *maxValue*
7: *minOutputHorizon* ⟵ *maxValue*
8: **for** $i = 0$ to $P - 1$ **do**
9: *min*1 ⟵ *maxValue*
10: *min*2 ⟵ *maxValue*
11: **for** $j = i + srcID \times ToRDegree$ to $(i + srcID \times ToRDegree + Q - 1)$ **do**
12: **if** *horizoninput*[*j*] < *min*1 **thne**
13: *min*1 ⟵ *horizoninput*[*j*]
14: *port*1 ⟵ *j*
15: **end if**
16: **end for**
17: **for** $k = i + destID \times ToRDegree$ to $(i + destID \times ToRDegree + Q - 1)$ **do**
18: **if** *horizonoutput*[*k*] < *min*2 **then**
19: *min*2 ⟵ *horizonoutput*[*k*]
20: *port*2 ⟵ *k*
21: **end if**
22: **end for**
23: *min*3 ⟵ *getMax*(*min*1, *min*2)
24: **if** *min*3 < *minInputHorizon* **and** *min*3 < *minOutputHorizon* **then**
25: *minInputHorizon* ⟵ *min*1
26: *minOutputHorizon* ⟵ *min*2
27: *inputport* ⟵ *port*1
28: *outputport* ⟵ *port*2
29: **end if**
30: **end for**
{**Above blocks of code select optimal input and output ports and their horizon in optical switch path.** *P* **is the number of total optical switches.** *Q* **is the number of ports of each ToR switch that are connected with each optical switch. Routing operation finishes here. Scheduling is the next operation.**}
31: T_{start} ⟵ *getMax*(*minInputHorizon*, *minOutputHorizon*) {**It gets maximum of two horizons and assigns it to the start time**}
32: **if** T_{start} < *getCurrentTime*() **then**
33: T_{start} ⟵ *getCurrentTime*()
34: **end if**
{**Current time is assigned to the start time if horizons are less than current time.**}
35: *burstlength* ⟵ *controlpacket*. *getBurstLength*()
36: T_{RL} ⟵ *burstlength* * 8/*datarate*
{**Requested timeslot** T_{RL} **is calculated from the burst length (BL) in the control packet.**}
37: $T_{start} \longleftarrow T_{start} + T_{sw} + T_{proc} + T_{oh}$
38: $T_{end} \longleftarrow T_{start} + T_{RL} + T_{guard}$
{**Above lines represent start and end time of a timeslot in an optical switch path.**}
39: *horizoninput*[*inputport*] ⟵ T_{end}
40 *horizonoutput*[*outputport*] ⟵ T_{end}
{**Horizons are updated with new time.**}
41: *controlpacket*. *setstarttime*(T_{start})
42: *controlpacket*. *setport*(*inputport* **mod** *ToRDegree*)
{**Control packet is updated with start time and port number of the ToR switch.**}
43: *destadd* ⟵ *controlpacket*. *getsourceadd*())
44: *controlpacket*. *setdestadd*(*controlpacket*. *getsourceadd*())
45: *controlpacket*. *setsourceadd*(*destadd*)
46: *sendAt*(*cp*, $T_{curr} + T_{proc}$)
{**Source and destination addresses in the control packet are swapped and the control packet is sent back to the source ToR. It also completes the scheduling operation. Switch configuration is the next task of the controller.**}
47: *confmsg* ⟵ *createConfMsg*()
48: *confmsg*. *settime*($T_{start} - T_{sw}$)
49: *confmsg*. *setinputport*((*inputport* **mod** *Q*) + (*srcID* × *Q*))
50: *confmsg*. *setoutputport*((*outputport* **mod** *Q*) + (*destID* × *Q*))
51: $L \longleftarrow \dfrac{inputport \ \ \textbf{mod} \ \ ToRDegree}{Q}$
52: *confmsg*. *setdestadd*(*getOpticalSwitchAdd*($\lfloor L \rfloor$))
53: *confmsg*. *setsourceadd*(*getControllerAdd*())
{**Above lines of code perform switch configuration operation.**}

Tables

Scalability analysis.

Optical switch	Switch size	Servers per rack	Total racks	Servers
SOA	[1024 × 1024]	40	1024	40,960
		80	2048	81,920
		240	6144	245,760
AWGR	[512 × 512]	40	512	20,480
		80	1024	40,960
		240	3072	122,880

Table 2
Simulation parameters.

Parameter name	Symbol	Value
Racks/ToR switches	N	24
Servers per rack	S_{RK}	40
Degree of ToR switches	X	40
TCP flows per server		40
Data in each TCP flow		25 MB
Topological degree of communication	TDC	{1,4,8} racks
Data rate	R	10 Gbps
TCP window size		64 KB
Control packet processing time	T_{proc}	1 μs
Switching time of fast switch	T_{sw}	1 μs
Overhead	T_{oh}	1 μs
Burst aggregation	T_a	{{50 μs, 50 KB}, {50 μs, 100 KB}}, {100 μs, 50 KB}, {100 μs, 100 KB}}
Buffer size per electronic port/VOQ		1000 packets

Table 3
Performance of the control plane.

Algorithm	Oversubscription	Optical switch (P)	Degree of ToR (X)	Exec. T
Routing and scheduling	4:1	$\forall P$	10	<0.15 μs
	2:1		20	<0.3 μs
	1:1		40	<1 μs
Switch configuration	4:1	10	10	≈0.029 μs
	2:1	20	20	≈0.031 μs
	1:1	40	40	≈0.033 μs