Noname manuscript No.
(will be inserted by the editor)

Hybrid and Optical Packet Switching Supporting Different Service Classes in Data Center Network

Artur Minakhmetov* · Cédric Ware · Luigi Iannone

Received: date / Accepted: date

Abstract Optical Packet Switching is a prominent technology proposing not only a reduction of the energy consumption by the elimination of numerous optical-electrical-optical conversions in electronic switches, but also a decrease of network latencies due to the cut-through nature of packet transmission. However, it is adversely affected by packet contention, preventing its deployment. Solutions have been proposed to tackle the problem: addition of shared electronic buffers to optical switches (then called hybrid opto-electronic switches), customization of TCP protocols, and use of different service classes of packets with distinct switching criteria.

In the context of data center networks we investigate a combination of said solutions and show that the hybrid switch, compared to the optical switch, boosts the performance of the data center network. Furthermore, we show that introducing a "Reliable" service class improves performance for this class not only in the case of the hybrid switch, but also brings the optical switch to performance levels comparable to that of the hybrid switch, all the while keeping other classes' performance on the same level.

Keywords Optical Packet Switching · Packet Switching · TCP Congestion Control · Optical Switches · Hybrid Switches · Classes of Service · Packet Preemption.

A. Minakhmetov*
LTCI, Télécom Paris, Institut Polytechnique de Paris, Palaiseau, F-91120
E-mail: artur.minakhmetov@telecom-paris.fr*

C. Ware
LTCI, Télécom Paris, Institut Polytechnique de Paris, Palaiseau, F-91120
E-mail: cedric.ware@telecom-paris.fr

L. Iannone
LTCI, Télécom Paris, Institut Polytechnique de Paris, Palaiseau, F-91120
E-mail: luigi.iannone@telecom-paris.fr

1 Introduction

The Optical Packet Switching (OPS) technology regained public interest in the mid-2000s [5] in the face of demand for high reconfigurability in networks, made possible through statistical multiplexing along with efficient capacity use and limiting the energy consumption of the switches [21]. However, with traffic being asynchronous and in the absence of technology that would make practical optical buffers in switches, the contention issue arises, leading to poor performance in terms of Packet Loss Ratio (PLR) [12], thus making the OPS concept impractical. To the present moment, several solutions have been proposed to bring the OPS technology to functional level, among which: adding a shared electronic buffer, thus making hybrid opto-electronic switches [28, 26, 24]; intelligent routing of packets of different priorities in the hypothesis that not all of them would need the same requirements for PLR [23]; and a network-level solution without changing the OPS hardware, introducing special TCP Congestion Control Algorithms (CCA) for packet transmission in order to increase overall network throughput, thus negating the still high PLR [7]. These three solutions are detailed below.

First, the hybrid switch consists in coupling an all-optical bufferless packet switch with an electronic buffer. Several implementations of the idea were already proposed in the last decade [28, 26, 24]. The concept of the hybrid switch considered in this study is: when contention occurs on two (or more) packets, i.e., when a packet requires using an output that is busy transmitting another packet, it is diverted to a shared electronic buffer through Optical-Electrical (OE) conversion. When the destination output is released, the buffered packet is emitted from the buffer, passing Electrical-Optical (EO) conversion. However, in the absence of contention, the hybrid switch works as an all-optical switch, without any wasteful OE and EO conversions. Adding a shared buffer with only a few input-output ports lets us considerably decrease PLR compared to an all-optical switch, and bring its performance up to the level of an electronic switch, but now with an important reduction in energy consumption, since one would save the OE/EO (OEO) conversions for most packets [23].

Second, highlighting an important question of the existence of classes of service in a network, Samoud et. al. [23] propose handling packets depending on their class: high priority packets can preempt low priority ones from being buffered or transmitted. It was shown that the demand for low PLR may be met for high priority packets and relaxed for others, achieving sustainable operation with a number of buffer input/output ports less than half that of optical links in a switch.

Third, Argibay-Losada et. al. [7] propose to use all-optical switches in OPS networks along with special TCP CCAs, in order to bring the OPS network throughput up to the same levels as in Electrical Packet Switching (EPS) networks with conventional electronic switches. Particularly noteworthy in protocol design is the Retransmission Timeout (RTO). This parameter controls how long to wait for the acknowledgment after sending a packet until the packet is considered lost and re-sent. When a transmission is successful and

without losses, RTO is set to a value close to the Round-Trip-Time (RTT), i.e., the time elapsed between the start of sending a packet and reception of its acknowledgment. By simple tweaking of initialization value of RTO and reducing it from conventional 1 s to 1 ms, it was shown that both custom and conventional TCP CCAs will boost the performance of the optical packet switched network.

In our previous works we analyzed the gain from use of the hybrid switch in a Data Center (DC) network by introducing Hybrid Optical Packet Switching (HOPS): we showed that HOPS with a custom-designed TCP can outperform OPS and EPS in throughput [17,16]. Furthermore, in [18] we have managed to show the possibility of a 4-fold reduction in DC energy consumption for data transport coming from OEO conversions while using HOPS compared to EPS. In this study we aim to investigate not only a combination of HOPS with custom design of TCP, but also the influence of the introduction of classes of service, i.e., switching and preemption rules for packets of different priorities.

Considering the general interest in the scientific and industrial communities to implement different packets priorities in Data Centers (DCs), as well as the problem of traffic isolation for tenants in DC [19], we implement the idea presented by Samoud et. al. [23] and investigate the benefits of application of such technology in a DC network. We successfully show that one can considerably improve the performance of network consisting of hybrid switches with a small number of buffer inputs for high priority connections while keeping it on a good level for default connections. Additionally, we show that high priority connections in OPS network also can benefit from the introduction of classes of service, matching or even surpassing the performance of the network consisted of hybrid switches with a small number of buffer inputs without classes of service.

The paper is composed as follows: Sec. 2 presents the hybrid switch architecture and packets preemption policy, Sec. 3 outlines simulation conditions, Sec. 4 discusses the results obtained and, finally, Sec. 5 offers our main conclusions.

2 Hybrid Switch Architecture and Packets Preemption Policy

2.1 Hybrid Switch Architecture

The first concept of a hybrid switch was proposed in 2004 by R. Takahashi et al. [27], and the scientific community has kept its attention on the implementation of the idea since then [24]. In 2010 X. Ye et al. [28] presented a Datacenter Optical Switch (DOS), an optical packet switch, that could be seen as a prototype of a hybrid switch: switching was performed through a combination of Arrayed Waveguide Gratings (AWG) switching matrix (using wavelength-specific switch outputs) and Tunable Wavelength Converters (TWC) (converting signal to required wavelength for routing), contentions were managed through the shared electronic buffer, storing contending pack-

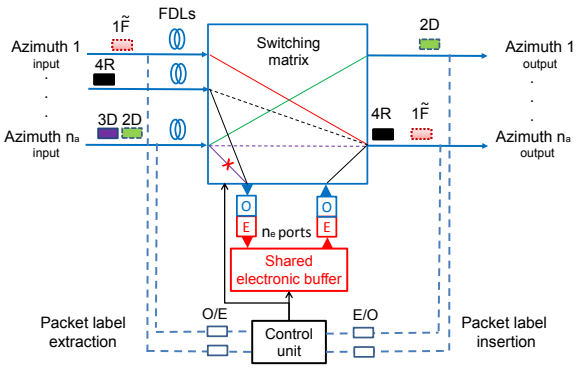

Fig. 1: General architecture of hybrid optical packet switch and class-specific switching rules demonstration

ets. In 2012 R. Takahashi et al. [26] presented a similar concept, called Hybrid Optoelectronic Packet Router (HOPR). DOS and HOPR, despite the name, are not quite what we call hybrid switches, as all the packets undergo OEO conversions by TWCs.

In 2016 T. Segawa et al. [24] proposed a switch that performs switching of optical packets through a Broadcast-and-Select (B&S) switching matrix and then re-amplification by Semiconductor Optical Amplifiers (SOAs). This switch splits the incoming optical packet into several ways corresponding to output ports, blocks those that don't match the packet's destination, and then re-amplifies the passed packet with a SOA. A shared electronic buffer is there to solve packet contention. The OEO conversion is made only for contending packets, unlike DOS or HOPR where all the packets undergo OEO conversions.

All of the presented solutions above have common main blocks, that we are emulating in our study in order to approach hybrid switch functions. The general structure of a hybrid switch is presented in Fig. 1 with the following main blocks: an optical switching matrix; a shared electronic buffer; and a control unit that configures the latter two according to the destination of the packets, carried by labels. The hybrid switch has n_a inputs and n_a outputs, representing non-wavelength-specific input and output channels, or Azimuths, thus making n_a channels for a switch. Another important parameter is n_e: n_e inputs and n_e outputs to the buffer. These are the channels through which a packet is routed/emitted to/from the buffer.

When a packet enters the switch, it carries along a label containing the destination address. Label management is generic so we didn't focus on label extraction, which isn't that easy, but will be required of any OPS/HOPS implementation so can be ignored when comparing them. Nevertheless, we propose and discuss several ways of label management. The labels can be extracted from the packet and processed without converting the packet itself to the electronic domain: the label may be extracted from the communication

channel through a splitter (usually 90:10) or a 1x2 MZI switch and then directed to the Control Unit (CU) where it undergoes O/E conversion (only the label, and not the whole packet carrying data); or by transmitting them out of band on dedicated wavelengths as in the OPS solution presented by Shacham et al. [25]. This solution allows label extraction via a tap coupler, requiring an OE conversion only for the label as well, and short Fiber Delay Lines at the inputs of the optical switch.

The Control Unit (CU), implemented electronically, controls the switching matrix. While CU analyzes the label, the packet is delayed in FDLs so as to give time to CU to adjust the switching matrix. Then, it would either route a packet to the desired output, or drop it. If CU decides to route the packet to desired output, it will generate new label, performing EO conversion, to add it to a packet on switch's output. This mechanism let us to stay out from OEO conversion of the whole packet.

The switching matrix could be implemented by any of the aforementioned technologies: B&S switch + SOA, TWC + AWGs, or even assembled in Benes Architecture multiple MZIs. Fast switching matrices already exist, achieving fast switching speeds of few ns: switching matrices based on MZIs as in [10] or on SOAs [9]. The optical matrix has a negligible reconfiguration time, on the ns scale [9], included in total switching time (i.e., switching speed) of a switch, that we also consider negligible for our study.

The effect of taking into account actual switching speed in simulations would result in an additional delay to packet transmission time. For example, adding 10 ns of switching speed would result in a further delay of 10 ns per switch, or per link connected to a switch. Thus, adding 10 ns of switching speed to simulations would be equal to adding 2 meters of fiber to the link and keeping switching time at 0.

The influence of link length changes to network throughput was studied previously, but only for the case of agnostic switching rules. It was discovered that for DCTCP [14] and its basis TCP SACK [17], link lengths changes, in the range of 10m-100m don't influence throughput for HOPS, yielding the same results. When considering OPS, however, if the network is under very high load (10^9 req./s), the throughput gets better if one increases l_{link} from 10 m to 100 m. Such better performance may be explained by less appropriate reaction of TCP to congestion and contention in case of shorter links, when TCP may overestimate network state due to very low latency, and react to packet losses more poorly. When considering the case of class-specific switching rules presented here, we discover the same behavior of the network in reaction to link length changes as in agnostic switching rules. In the case of HOPS, the network performance (e.g., throughput and Flow Completion Time (FCT)) does not change if increasing l_{link} from 10 m to 100 m, and in the case of OPS, the performance is similar or even better.

In the current study, we thus choose to represent only the case of link length l_{link} of 10 m with negligible switching speed.

The switches presented in the study are considered to have a small port-count of 8. This is enabled through the use of n-ary Clos Fat-Tree topology [4],

as shown in Fig. 2. This small amount of I/O ports plays favorably into switching speed; however, if the port-count is increased, this could lead to a non-negligible increase of switching speed. In that case, one could suggest using a similar approach as the RotorNet [13] architecture for constructing a switch with some predefined fixed routes inside of the switch.

Nevertheless, before advocating for a change of switch architecture, we want to point out that n-Clos Fat-Tree topologies (the example considered here) [4] are destined exactly to solve the problem of switches port-count, using many low port-count switches, instead of a small amount of high-port count switches. This is why if the problem of scaling arrives, it could be successfully solved by using higher-order n-Clos Fat-Tree topology with moderate port-count switches, without an increase of switching speed.

To create a hybrid switch we are adding to the essential blocks of all-optical packet switch a shared electronic buffer. Same way as other blocks considered previously we assume that shared electronic buffer is generic and switching time is negligibly small. Shared electronic buffer is assumed to be implemented by burst receivers [22].

According to standards, the locking time of burst transceivers can build up to more than 100 ns [20], however, the late developments can lower this time towards few tens of ns [22]. Taking this solution developed by Rylyakov et al. [22] as an example with 31 ns of locking time, we can derive that such time in 10 Gb/s would be equal to an overhead of 39 B per packet in an OPS or HOPS network. In this work, as discussed in Sec. 3, we consider an overhead of 64 B per packet, and we consider that we are including in this overhead all necessary information for the establishment of a connection.

We are not considering any particular technology for the label management, control unit, and implement our simulations focusing on the assumed ideal optical switching matrix, and on a store-and-forward shared electronic buffer.

2.2 Packets Preemption Policy

The switching algorithm for a hybrid switch is adopted from [23] and implements different buffering and preemption rules for different packet classes. We consider three of them: Reliable (R), Fast (F) and Default packets (D). R packets are those that attempted to be saved by any means, even by preemption of F or D packets on their way to buffer or switch output. F packets could preempt only D packets on their way to the switch output. D packets cannot preempt other packets.

The priority distribution in the DC network is adopted from [23] and taken from the real study on core networks [1]. This may seem improper for DCs, however, we seek to study the performance of the hybrid switch in the known context. Also, it will be shown below that the distribution considered lets us organize a pool of premium users (10%) of R connections in DCs that could profit from the best performance, while other users almost wouldn't be influenced by performance loss. F packets can preempt D packets only on the

Algorithm 1 Preemption Policies in a Hybrid Switch

```
1: procedure SWITCH (PACKET p)
2:     prio ← p.priority_class
3:     switch_out ← get_destination_azimuth(p)
4:     if switch_out.is_free() then                          ▷ General switching rule of HOPS
5:         switch_out.receive(p)
6:     else if buffer_in.is_free() then
7:         buffer_in.receive(p)
8:     else if prio==R and buffer_in.receiving(D) then       ▷ Try to buffer R, preempt D
9:         buffer_in.preempt_last_packet(D)
10:        buffer_in.receive(p)
11:    else if prio==R and switch_out.receiving(D) then      ▷ Try to switch R, preempt D
12:        switch_out.preempt_last_packet(D)
13:        switch_out.receive(p)
14:    else if prio==R and buffer_in.receiving(F̃) then       ▷ Try to buffer R, preempt F̃
15:        buffer_input.preempt_last_packet(F̃)
16:        buffer_input.receive(p)
17:    else if prio==R and switch_out.receiving(F̃) then      ▷ Try to switch R, preempt F̃
18:        switch_out.preempt_last_packet(F̃)
19:        switch_out.receive(p)
20:    else if prio==F̃ and switch_out.receiving(D) then      ▷ Try to switch F̃, preempt D
21:        switch_out.preempt_last_packet(D)
22:        switch_out.receive(p)
23:    else
24:        drop(p)
```

way to switch output, while R packets first would consider preemption of D packet being buffered. Thus F packets had lower delay than R packets [23]. However, further it will be shown that this device-level gain doesn't translate to network-level gain in a DC network in terms of Flow Completion Time (FCT), and R connections perform better than F. That's why here we refer to Fast (F) as Not-So-Fast ($\tilde{\text{F}}$) packets and connections. Eventually, in this study we consider, that 10% of connections have R priority, 40% of connections have $\tilde{\text{F}}$ priority, 50% of connections have D priority.

When a packet enters the switch it checks if required Azimuth output (i.e., switch output) is available. If yes, the packet occupies it. Otherwise, the packet checks if any of buffer inputs are available. If yes, it occupies one and starts buffering. If none of the buffer inputs are available, in the case of absence of preemption policy in a switch the packet would be simply dropped. Here, we consider a switch with preemption policy that would follow the steps of algorithm presented in Alg. 1. If a packet of any type is buffered, it is re-emitted FIFO, as soon as required switch output is available.

2.3 Manageability of Packet Loss

On a Fig. 1 we demonstrate the Alg. 1 on a hybrid switch with only one buffer input/output $n_e = 1$ for simplicity. We have packets: 1 $\tilde{\text{F}}$ – a Not-So-Fast packet, 2D and 3D – two Default packets, and 4R – a Reliable packet. Packet 2D requires an available output of the switch and is transmitted without any

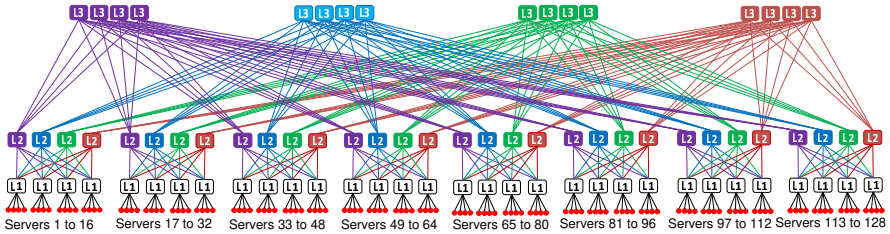

Fig. 2: Fat-tree topology network, interconnecting 128 servers with three layers of switches.

OEO conversions directly. Then, packet 1 $\tilde{\text{F}}$ arrive at the switch and is directed to the required output of the switch. After, a Default 3D packet arrives at the switch, and is redirected to the buffer, as the required Azimuth is occupied by the packet 1 $\tilde{\text{F}}$. Further, a Reliable packet 4R arrives and requires the same output as packets 1 $\tilde{\text{F}}$ and 3D, 1 $\tilde{\text{F}}$ is still occupying the switch output and 2D occupying sole buffer input. In an agnostic switching rules case the packet 4R would be lost, and packets 1 $\tilde{\text{F}}$ and 3D would be transmitted, but in the current case it is a Default packet is preempted and lost, while 4R packet is put in the buffer and then transmitted to the output of the switch.

We bring attention of the reader to the fact, that 4R packet is delayed in comparison to 1 $\tilde{\text{F}}$ and 2D. 4R packet was switched through the electronic buffer, i.e., in a store-and-forward mode, whereas packets 1 $\tilde{\text{F}}$ and 2D were switched optically, i.e., in a cut-through mode.

In this context class specific switching rules help us to control, what packet would be lost, 4R or 3D. In other words this translates to the general rule: we can decide what packets would be lost, while keeping more or less same overall PLR. Class specific switching rules allow us to take control and making packets drop more manageable.

3 Study Methodology

As in our previous work [16,17], we simulate the communications of DC servers by means of optical packets. We study DC network performance for two groups of scenarios: DC with classes of service using preemption policy outlined in Sec. 2.2, and DC with switches that don't have any preemption rules. For each scenario we consider OPS and HOPS case.

Communications consist of transmitting files between server pairs through TCP connections. The files' size is random, following a lognormal-like distribution [3], which has two modes around 10 MB and 1 GB. We simulate transmission of 1024 random files (on the same order as 1000 in [7]), i.e., 8 connections per server. File transmission is done by data packets using jumbo frames with a size of 9 kB. This value defines the packet's payload and corresponds to Jumbo Ethernet frame's payload.

In our study we also use SYN, FIN, and ACK signaling packets. We choose for them to have the minimal size of the Ethernet frame of 64 B [2]. We assume that this minimal size would contain only the relevant information about Ethernet, TCP/IP layers. As we still need to attach to the jumbo frames all the information of these layers, for simplicity, we just attach to it a header of 64 B discussed previously. Thus we construct a packet of maximum size 9064 B to be used in our simulations, with a duration τ dependent on the bit-rate. Servers have network interface cards of 10 Gb/s bit-rate. Buffer inputs and outputs used by a hybrid switch support the same bit-rate.

The actual transmission of each data packet is regulated by the DCTCP CCA [8], a TCP variant developed for DCs, which decides whether to send the next packet or to retransmit a not-acknowledged one. CCA uses next constants: $DCTCP_{threshold}$ = 27192 B, $DCTCP_{acks/pckt}$ = 1, $DCTCP_g$ = 0.06, as favorable for HOPS. We apply the crucial reduction of the initialization value of RTO towards 1 ms, as advised in [7]. To be realistic, the initial 3-way handshake and 3-way connection termination are also simulated.

We developed a discrete-event network simulator based on an earlier hybrid switch simulator [23], extended so as to handle whole networks and include TCP emulation. The simulated network consists of hybrid switches with the following architecture: each has n_a azimuths, representing the number of input/output optical ports, and n_e input/output ports to the electronic buffer, as shown in Fig. 1. The case of bufferless all-optical switch (OPS) corresponds to $n_e = 0$, for the case of the hybrid switch (HOPS) we consider $n_e = 2$.

We study the DC fat-tree topology, interconnecting 128 servers by means of 80 identical switches with $n_a = 8$ azimuths, presented in Fig. 2, a sub-case of a topology deployed in a Facebook's DCs [6]. All links are bidirectional and of the same length l_{link} = 10 m as typical link lengths for DC. The link plays the role of device-to-device connection, i.e., server-to-switch, switch-to-server or switch-to-switch. The link is assumed to represent a non-wavelength-specific channel. Paths between servers are calculated as a minimum number of hops, which offers multiple equal paths for packet transmission allowing load-balancing and thus lowering the PLR.

The network is characterized by the network throughput (in Gb/s) and average FCT (in µs) for each class of connections, and for all classes of connections combined (i.e., "General" performance) as a function of the arrival rate of new connections, represented by the Poissonian process.

Connection demands arrive following the Poisson distribution with a given mean number of file transmission requests per second, which defines the load on the network. A class, R, $\tilde{\text{F}}$ or D, is assigned to a connection according to the distribution we adopted earlier, provided in Sec. 2.2. We assume that each server gets at least 8 connection demands during one instance of simulation for a defined set of network parameters, thus making a total of 1024 connections to establish among 128 servers considered in this study.

The network performance with different switch types (OPS or HOPS) and switching rules is studied under progressively increasing load. In addition to

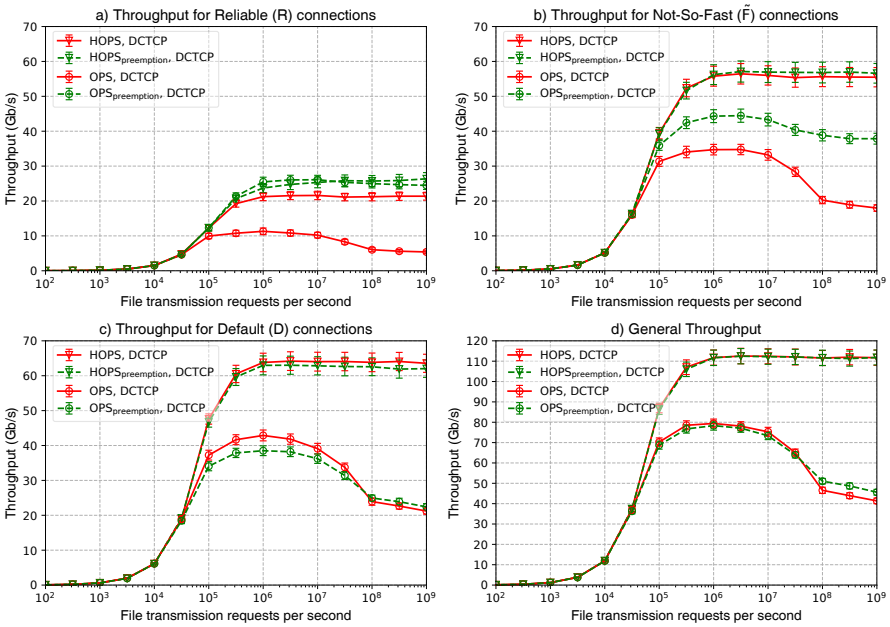

Fig. 3: DC network's throughput for connections: a) Reliable (R) connections, b) Not-So-Fast ($\tilde{F}$) connections, c) Default (D) connections, d) Overall Network Performance

network throughput measurement, we also choose to measure FCT, as a metric considered to be the most important for network state characterization [11].

4 Evaluation Results

We present here the results of our study and their analysis. To reduce statistical fluctuations, we repeated every simulation a hundred times with different random seeds for $n_e = 0$ (OPS) and $n_e = 2$ (HOPS). The mean throughput and mean FCT are represented in Fig. 3 and in Fig. 4 with 95% t-Student confidence intervals, for three types of connections: R, $\tilde{F}$ and D connections. We take as a reference results from the network without packet preemption policy: the division of connections to classes is artificial and just represent corresponding to classes' percentage of connections in the network. We define high load as more than 10^5 connections per second.

While comparing just OPS and HOPS, it is seen that in general HOPS outperforms or has the same performance as OPS, but with the cost of only $n_e = 2$ buffer inputs.

The R connections benefit the most from the introduction of the classes of service and preemption policy as it seen on Fig. 3a) and Fig. 4a) both in the cases of OPS and HOPS. Throughput for R connections in HOPS network rises by around 25% (Fig. 3a), while in OPS case it rises by a factor 2.5 at

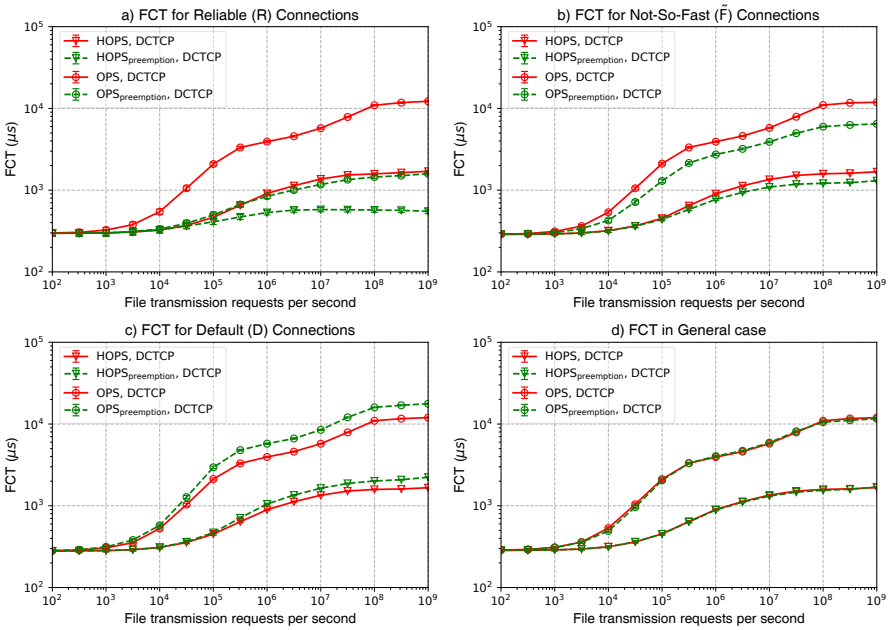

Fig. 4: DC network's Flow Completion Time for connections: a) Reliable (R) connections, b) Not-So-Fast ($\tilde{F}$) connections, c) Default (D) connections, d) Overall Network Performance

least on high load, matching the performance of HOPS network. We would like to bring readers attention on the fact that it seems to be low throughput, compared to other classes of service, but this is the mere effect of the fact that in the network only 10% of connections are of type R. However, if one considers the FCT, which is comparable with other types of classes and lowest among them, then the preemption policy's benefits are more evident: on the highest considered load OPS reduces its FCT almost by a factor of 8, while HOPS reduces it by at least a factor of 2, keeping it on the level of tens of µs. Even if OPS's FCT doesn't match FCT in the case of HOPS while considering classes of service, it does match the FCT in the case of HOPS without classes of service. While applying preemption policy, connections are indeed Reliable: in Fig. 5 we can see that PLR (ratio of packets lost due to preemption or dropping to packets emitted by servers) decreases by around factor of 10.

The $\tilde{F}$ traffic benefits less than R traffic from introduction of classes of service, but the gain is still there. For OPS we managed to boost the throughput by almost 30-100% on the high load, while for HOPS the gain is less evident. However, when we consider FCT on Fig. 4b) we can see that OPS decreases its FCT by almost a factor of 2 for high load, and HOPS around 25%. HOPS FCT for $\tilde{F}$ packets is bigger than for those of reliable (R), contrary to what may be induced from [23], where they are labeled as Fast (F). This may be explained by the fact that the delay benefits for F packets are on the order of a µs, while here FCT is of an order of tens and hundreds of µs, and is defined

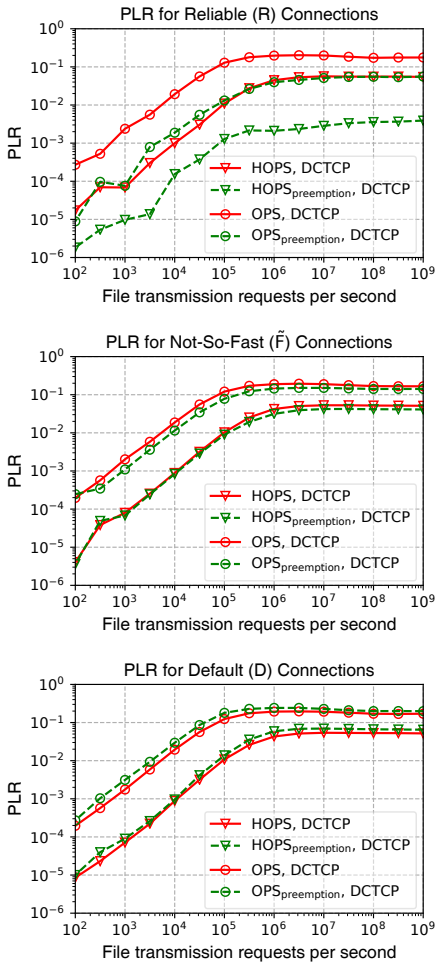

Fig. 5: Mean PLR of Reliable (R), Not-So-Fast ($\tilde{F}$) and Default (D) Connections

mostly by TCP CCAs when contention problem is solved. While considering PLR of $\tilde{F}$ connections on Fig. 5, we can see that there is a marginal gain of around 10%, if to introduce class-specific switching rules, making it the second class to R that would benefit from them.

The D traffic does not benefit from the introduction of classes of service, and it is on its account the gains for R and $\tilde{F}$ traffic exists. However, while considering the performance reductions, we notice almost unchanged throughput for HOPS case, and for OPS the drop of only 10% at most, which could be seen as a beneficial trade-off in R and $\tilde{F}$ traffic favor with their boost of performance both in throughput and FCT. The same drop of performance of 10% are observed for PLR, as shown on Fig. 5. But, by paying that cost

one can get same 10% of improvements for $\tilde{F}$ connections, and 100% for R connections.

The network as a whole, regardless of the presence of classes of service, performs the same, which is expected, as connections occupy limited network resources. We can observe that the gain due to introduction of classes of service for R and $\tilde{F}$ traffic decreases with the increase of number of buffer inputs/outputs (i.e., from $n_e = 0$ towards $n_e = 2$), and for fully-buffered switch ($n_e = n_a = 8$) the gain would be 0, because no packet would ever require preemption, only buffering. However, there are technological benefits to use small number of buffer input/outputs as it directly means simplification of switching matrix ($n_a = 8$, $n_e = 2$ means 10x10, $n_a = n_e = 8$ means 16x16 matrix) and reduction of number of burst receivers (inputs) and transmitters (outputs) for buffers. In the case of EPS, the gain would be also 0, but in general EPS entails an increase in energy consumption for OEO conversions compared to HOPS by a factor of 2 to 4 [18] on high load.

While observing the network performance overall, it's seen that introduction of classes of service both in OPS and HOPS helps to boost the performance for the R and $\tilde{F}$ connections, while keeping the performance of D connections relatively on the same level. This fact could lead to economic benefits in a Data Center: charge more priority clients for extra performance, almost without loss of it for others. Furthermore, using pure OPS instead of HOPS in DCs may be economically viable, as OPS delivers the best possible performance to R connections, on the level of HOPS performance for $\tilde{F}$ connections, and relatively low performance for D connections, since high performance may be not needed for D connections.

5 Conclusions

In this study we enhanced the analysis of HOPS and OPS DC networks by applying classes of service in terms of preemption policy for packets in optical and hybrid switches, while solving the contention problem. In the case of HOPS we demonstrated that with custom packet preemption rules, one can improve the performance for Reliable and Not-So-Fast class connections, almost without losing it for Default connections. Furthermore, we showed that classes of service can boost the performance of OPS for Reliable and Not-So-Fast class connections, match or bring it to the level of those in HOPS. This proves that OPS could be used in DCs, delivering high performance for certain connections, while Default class connections are still served on an adequate level.

It is worth noting that HOPS and OPS networks are enabled not only by specific types of switches but also by server-side management of TCP connections, regardless of its classes. For the presentation in this paper, we chose the DCTCP protocol, but one may use other ones. If a network operator aims at obtaining lower latencies (e.g., Round Trip Time) and is ready to trade-off throughput, we would recommend using TCP SAWL [17,15]. It can lower la-

tencies compared to DCTCP, if applied to small link lengths DC network, both in cases of agnostic switching rules and class-specific switching rules. However, this gain comes at the price of a lower throughput in all connection classes.

It remains to be seen whether these results remain with a different service class distribution; and whether an actual low-latency service class can be implemented (e.g., using another protocol than TCP). Furthermore, the subject of future studies may include consideration of high-speed transceivers (higher than 10 Gb/s), as well as the application of smaller than jumbo frames of 9 KB, involving shorter packet transmission times, in order to understand how it would influence network performance.

References

1. 100Gb/s Réseau Internet Adaptative (100GRIA) FUI9 project. Tech. rep. (2012)
2. IEEE standard for ethernet. IEEE Std 802.3-2015 (Revision of IEEE Std 802.3-2012) pp. 1–4017 (2016)
3. Agrawal, N., Bolosky, W., Douceur, J., Lorch, J.: A five-year study of file-system metadata. ACM Trans. Storage **3**(3) (2007)
4. Al-Fares, M., Loukissas, A., Vahdat, A.: A scalable, commodity data center network architecture. In: Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication, SIGCOMM '08, pp. 63–74. ACM, New York, NY, USA (2008). DOI 10.1145/1402958.1402967
5. de Almeida Amazonas, J.R., Santos-Boada, G., Solé-Pareta, J.: Who shot optical packet switching? In: Int. Conference on Transparent Optical Networks (ICTON), Th.B3.3 (2017)
6. Andreyev, A.: Introducing data center fabric, the next-generation facebook data center network. Online: https://code.fb.com/production-engineering/introducing-data-center-fabric-the-next-generation-facebook-data-center-network/ (2014). Accessed: 2018-07-17
7. Argibay-Losada, P.J., Sahin, G., Nozhnina, K., Qiao, C.: Transport-layer control to increase throughput in bufferless optical packet-switching networks. IEEE J. Opt. Commun. Netw. **8**(12), 947–961 (2016)
8. Bensley, S., Thaler, D., Balasubramanian, P., Eggert, L., Judd, G.: Data Center TCP (DCTCP): TCP Congestion Control for Data Centers. RFC 8257, RFC Editor (2017)
9. Cheng, Q., Wonfor, A., Wei, J.L., Penty, R.V., White, I.H.: Low-energy, high-performance lossless 8x8 soa switch. In: Optical Fiber Communication Conference, p. Th4E.6. Optical Society of America (2015)
10. Chu, T., Qiao, L., Tang, W., Guo, D., Wu, W.: Fast, high-radix silicon photonic switches. In: Optical Fiber Communication Conference, p. Th1J.4. Optical Society of America (2018). DOI 10.1364/OFC.2018.Th1J.4
11. Dukkipati, N., McKeown, N.: Why flow-completion time is the right metric for congestion control. SIGCOMM Comput. Commun. Rev. **36**(1), 59–62 (2006)
12. Kimsas, A., Øverby, H., Bjornstad, S., Tuft, V.L.: A cross layer study of packet loss in all-optical networks. In: Proceedings of AICT/ICIW (2006)
13. Mellette, W.M.: A practical approach to optical switching in data centers. In: 2019 Optical Fiber Communications Conference and Exhibition (OFC), pp. 1–3 (2019)
14. Minakhmetov, A.: Cross-layer hybrid and optical packet switching. Theses, Institut Polytechnique de Paris (2019). URL https://pastel.archives-ouvertes.fr/tel-02481270
15. Minakhmetov, A., Nagarajan, A., Iannone, L., Ware, C.: On the Latencies in a Hybrid Optical Packet Switching Network in Data Center. In: 2019 Optical Fiber Communications Conference and Exhibition (OFC), pp. 1–3 (2019)
16. Minakhmetov, A., Ware, C., Iannone, L.: Optical Networks Throughput Enhancement via TCP Stop-and-Wait on Hybrid Switches. In: Optical Fiber Communication Conference, p. W4I.4. Optical Society of America (2018)

17. Minakhmetov, A., Ware, C., Iannone, L.: TCP Congestion Control in Datacenter Optical Packet Networks on Hybrid Switches. IEEE J. Opt. Commun. Netw. **10**(7), B71–B81 (2018)
18. Minakhmetov, A., Ware, C., Iannone, L.: Data Center's Energy Savings for Data Transport via TCP on Hybrid Optoelectronic Switches. IEEE Photonics Technology Letters **31**(8), 631–634 (2019). DOI 10.1109/LPT.2019.2902980
19. Noormohammadpour, M., Raghavendra, C.S.: Datacenter traffic control: Understanding techniques and tradeoffs. IEEE Communications Surveys Tutorials **20**(2), 1492–1525 (2018)
20. Qiu, X.: [ofc 2013 tutorial ow3g.4] burst-mode receiver technology for short synchronization. In: 2013 Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference (OFC/NFOEC), pp. 1–28 (2013)
21. Rouskas, G.N., Xu, L.: Optical Packet Switching, pp. 111–127. Springer US, Boston, MA (2005)
22. Rylyakov, A., Proesel, J., Rylov, S., Lee, B., Bulzacchelli, J., Ardey, A., Schow, C., Meghelli, M.: A 25 gb/s burst-mode receiver for low latency photonic switch networks. In: Optical Fiber Communication Conference, p. W3D.2. Optical Society of America (2015). DOI 10.1364/OFC.2015.W3D.2. URL http://www.osapublishing.org/abstract.cfm?URI=OFC-2015-W3D.2
23. Samoud, W., Ware, C., Lourdiane, M.: Performance analysis of a hybrid optical-electronic packet switch supporting different service classes. IEEE J. Opt. Commun. Netw. **7**(9), 952–959 (2015)
24. Segawa, T., Ibrahim, S., Nakahara, T., Muranaka, Y., Takahashi, R.: Low-power optical packet switching for 100-Gb/s burst optical packets with a label processor and 8 x 8 optical switch. J. Lightw. Technol. **34**(8), 1844–1850 (2016)
25. Shacham, A., Small, B.A., Liboiron-Ladouceur, O., Bergman, K.: A fully implemented 12x12 data vortex optical packet switching interconnection network. J. Lightwave Technol. **23**(10), 3066 (2005)
26. Takahashi, R., Nakahara, T., Suzaki, Y., Segawa, T., Ishikawa, H., Ibrahim, S.: Recent progress on the hybrid optoelectronic router. In: 2012 International Conference on Photonics in Switching (PS), pp. 1–3 (2012)
27. Takahashi, R., Nakahara, T., Takahata, K., Takenouchi, H., Yasui, T., Kondo, N., Suzuki, H.: Ultrafast optoelectronic packet processing for asynchronous,optical-packet-switched networks, Invited. J. Opt. Netw. **3**(12), 914–930 (2004)
28. Ye, X., Mejia, P., Yin, Y., Proietti, R., Yoo, S.J.B., Akella, V.: DOS - a scalable optical switch for datacenters. In: 2010 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS), pp. 1–12 (2010)