Amélioration du débit des réseaux optiques via TCP Stop-and-Wait sur les commutateurs hybrides

Artur Minakhmetov, Cédric Ware, Luigi Iannone

Amélioration du débit des réseaux optiques via TCP Stop-and-Wait sur les commutateurs hybrides

Artur Minakhmetov[1] et Cédric Ware[1] et Luigi Iannone[1]

[1]*LTCI, Télécom ParisTech, Université Paris-Saclay, 75013, Paris, France*

Nous démontrons une augmentation possible de 50% du débit dans les réseaux OPS (Optical Packet Switching) de data centers en remplaçant les commutateurs tout optique par des commutateurs optiques à buffers électroniques partagés et en implémentant simultanément les algorithmes TCP Stop-and-Wait.

Mots-clefs : Optical Packet Switching, Packet Switching, Congestion control, TCP Congestion Control design, CCA, Optical Switches, Hybrid Switches, Commutateur Optique, Commutateur Hybride, Commutation Optique des Packets

1 Introduction

The concept of Optical Packet Switching (OPS) regained momentum in the mid-2000s [dAASBSP17] as a response to the need for highly reconfigurable networks, for both efficient capacity usage via statistical multiplexing as well as curbing the unsustainable growth of energy consumption in switches [RX05]. However, when traffic is asynchronous, OPS is vulnerable to contention due to the absence of optical buffers, leading to a high packet loss ratio (PLR) [KOBT06, WSGL16]. Thus far, a number of solutions have been proposed that may help make OPS practical [WSGL16], among which combining an all-optical switch with a shared electronic buffer, creating a hybrid switch [IT17], as well as careful TCP-based congestion control algorithm (CCA) design for networks that consist only of all-optical switches [ALSNQ16].

The concept of hybrid switches brings together all-optical and electronic switches – it acts as an all-optical switch in general, but when two packets contend for the same output port then the electronic part comes into play. The second packet that requested the output port is put into an electronic buffer with Optical-Electrical (OE) conversion. Once the requested resource is released, the buffered packet is emitted from buffer. Such an approach helps keep PLR low.

Another approach to bringing OPS bufferless switches to the level of electronics ones is the proper design of TCP CCAs. One of the major factors in TCP is the Retransmission Time-Out (RTO) – the time after which a packet is considered lost, and hence retransmitted, unless the corresponding acknowledgement was received. The RTO is usually set as close as possible to the Round-Trip-Time (RTT), i.e. the time elapsed between the beginning of the transmission of the data packet and the reception of the corresponding acknowledgment. Also important is to decide how many packets to send at a time. Argibay-Losada et al. [ALSNQ16] propose the use of Stop-And-Wait (SAW), which consists in sending one data packet at a time, then waiting for the corresponding acknowledgement till RTO expires. A new data packet is sent only after the previous one has been successfully acknowledged. Considering the fact that in data-center networks the transmission duration of an optical packet is usually longer than the propagation delay between two nodes [ALSNQ16], such a SAW protocol proves to be efficient.

In this paper we bring the hybrid switches and the TCP CCAs concepts together – adapting SAW so as to take advantage from the potential of electronic buffers in data-center networks. We compare by simulation the performance of the original SAW algorithm with bufferless switches and with hybrid ones. Then, an adaptation of SAW is proposed, allowing to achieve a 50% throughput improvement. The paper is composed as follows – Sec. 2 describes how simulations were carried out, while Sec. 3 discusses the results obtained. Finally, Sec. 4 offers our main conclusions.

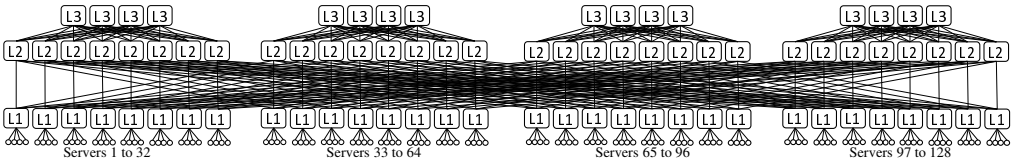

FIGURE 1: Fat-tree topology network, interconnecting 128 servers with three layers of switches.

2 Experimental Simulation Setup

We simulate the communications of data-center servers by means of optical packets, for two scenarios : *i)* when the network is composed of only all-optical switches and *ii)* when it is composed only of hybrid switches. Communications consist of transmitting files between server pairs through TCP connections. The files' size is random, following a log-normal distribution [ABDL07]. File transmission is done by data packets of Maximum Transmission Unit (MTU) size, i.e., 9 kB with a duration τ dependent on the bit-rate. (The last packet may be smaller since file size is not an exact multiple of the MTU.) The actual transmission of each data packet is regulated by the TCP CCA, which decides whether to send the next packet or to retransmit a not acknowledged one. To be realistic, the initial 3-way handshake and 3-way connection termination are also simulated, as well as 64-byte SYN, FIN, and ACK signaling packets. The network is characterized by the throughput (in Gb/s) as a function of the arrival rate of new connections.

We developed a discrete-event network simulator based on an earlier hybrid switch simulator [SWL14], extended so as to handle whole data-center 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 as well as output optical ports, and n_e input/output ports to the electronic buffer. The case of the bufferless all-optical switch corresponds to $n_e = 0$. When a packet is switched to an available azimuth, it occupies it. If the azimuth is busy, then the packet is redirected to the electronic buffer through an electronic port. The packet will then be re-emitted when the output azimuth it needs is released. The re-emission queuing strategy of the buffer is First-In-First-Out (FIFO) for a given azimuth.

The TCP CCA used in our study is SAW [ALSNQ16] and modifications of such. SAW is efficient when the transmission of a packet takes longer than the propagation delay between servers. Here an RTO T_1 of 1 ms is taken as the initial value. If the corresponding acknowledgement packet is not received within this time, for retransmission the RTO is now multiplied by a constant factor $\alpha > 1$: $T_i = \alpha \cdot T_{i-1}$ up to a maximum value of $T_i = 60$ s. When the acknowledgment is received, the RTO is updated to a weighted average of the current value and measured RTT γ : $T_i = \beta \cdot \gamma + (1-\beta) \cdot T_{i-1}$, with $\beta \in (0,1)$. We opt for $\alpha = 1.1$ and $\beta = 0.5$ as for the more suitable parameters found in [ALSNQ16].

The proposed setup works efficiently for all-optical switches, but does not allow to take advantage of the buffers. Indeed, even putting and extracting a packet into/from the buffer adds up to the RTT, becoming longer than the RTO estimated for the previous non-buffered packet, and thus the server will consider such a packet as lost. To overcome this limitation, we propose a modification of the SAW algorithm, so that the RTO is increased by a multiple p of packet duration τ, so as to give a chance to packets having traversed up to p buffers to arrive before the RTO. Hence, instead of the T_i shown before, we take as an update for RTO : $T'_i = T_i + p \cdot \tau$. Our simulations consider $p \in \{0,1,4\}$. The essential modification in SAW algorithm is to wait slightly longer, so we refer to it as Stop-And-Wait-Longer – SAWL.

In order to evaluate the performance of these algorithms we consider the fat-tree topology of a data-center shown in Fig. 1, that enables efficient use of distribution frameworks (e.g. MapReduce) [ALSNQ16].

Such a network interconnects 128 servers, that have network interface cards of 10 Gb/s bit rate, by means of 80 identical switches, that have $n_a = 8$ azimuths and a variable number of $n_e \in \{0,2,5\}$ of the same bit rate. All links are bidirectional and of the same length $l_{link} \in \{10,100,1000,10000\}$ m : 10 m and 100 m are for data-centers, 1000 m and 10000 m are considered to check the evolution of our results with increasing range. Paths between servers are calculated as minimum number of hops, which offers multiple equal paths for packet transmission ; that is very beneficial for OPS, allowing lowering the PLR thanks to load-balancing. This means that a packet has an equal probability to use each of the available paths (the

same rule applies for buffered packets). Reordering issues do not arise since SAW only allows one packet per connection in flight.

In our case we follow poissonian process of arrivals of new connection demands (file transmission requests per second) between all of the servers, so as to study the performance of a network with different switches and protocols under progressively increasing load. As in [ALSNQ16] high load could be related to a MapReduce-like model of load distribution in a data-center, supporting a search engine, where network load spikes occur when many servers must rapidly exchange information to form a response, passing through a "shuffling phase".

3 Evaluation Results

To reduce statistical fluctuations, we repeated every simulation with different random seeds for each pair of n_e and p values, a hundred times for $l_{link} \in \{10, 100\}$ m and ten times for $l_{link} \in \{1000, 10000\}$ m. The mean throughput obtained is represented in Fig. 2. Its standard deviation is $\sigma \in (0.144, 0.196)$ times the

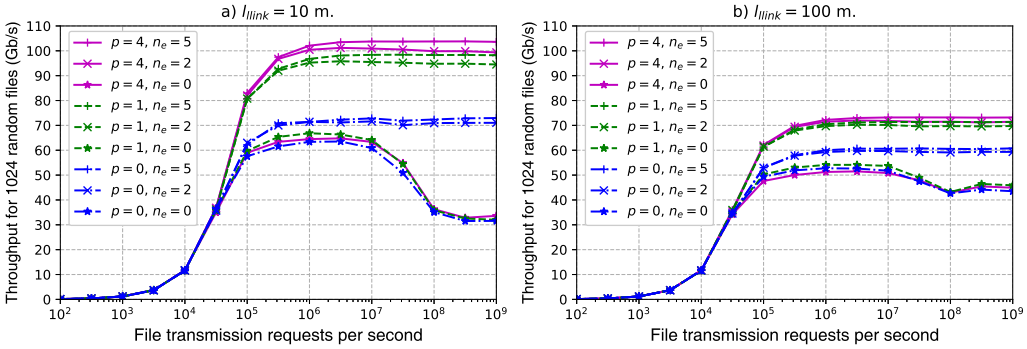

FIGURE 2: Network throughput dependence on TCP SAWL parameter p and number of buffer I/O ports n_e for : a) $l_{link} = 10$ m, b) $l_{link} = 100$ m.

mean value, with a Pearson correlation coefficient $\rho \in (0.973, 0.998)$. The results for SAW in bufferless networks ($p = 0$, $n_e = 0$) differ a little from the results obtained by Argibay-Losada et al. [ALSNQ16], but coincides for high load (more than 10^8 requests/s). This could be explained by the difference in the file size distribution implemented in the simulator, which has a lot of arbitrary parameters, as well as by possible differences in the way of load implementation.

When we consider links of 10 m and 100 m for the case of SAW $p = 0$ and different n_e values, we see that the results do not differ much, except at high load, where the hybrid switch performs better, reaching double the throughput for $l_{link} = 10$ m. This happens due to longer queuing times, which means a higher RTT on average, that lets us wait for acknowledgments of buffered packets. The hybrid switch is a robust solution for heavily-loaded networks, so they could support more traffic, in our case after $load = 10^7$ file requests per second.

If we consider SAWL ($p > 1$), we see that for data-center networks composed of hybrid switches we have a gain of more than 50% with respect to the bufferless networks using vanilla SAW. The bigger n_e, the better the throughput. During our simulations, we have found that increasing p to more than 4 does not provide better results, meaning that a packet could be buffered four times before RTO. At high load (more than 10^7 requests/s) the throughput is increased by a factor of 3 in the case of $l_{link} = 10$ m.

We note that the throughput is almost the same for any switch for link lengths of 1000 m (or greater), and mean throughput does not go above 16.6 Gb/s for $l_{link} = 1000$ m, and 1.9 Gb/s for $l_{link} = 10000$ m (not shown here due to space constraints). No gain is observed due to buffers nor algorithm design. This is explained by the fact that for each connection there is only one packet in flight, limiting the throughput, while individual switches' load never rise high enough to cause significant contention.

4 Conclusion

The data-center networks could benefit from hybrid switches that have a lower energy consumption than electric ones, and a higher throughput and robustness than all-optical ones, using just a few electric ports and introducing the specially-designed TCP protocols. This could be applied to any type of networks and load by changing the number of the electric ports and carefully adjusting the TCP algorithm. In this paper we showed how by introducing a small modification to the SAW algorithm allows to take advantage of hybrid switches and gain at least 50% in throughput, and even more in some cases.

Yet, we still see that in long distances a hybrid switch does not show any different result compared to an all-optical one. Nevertheless, this could be improved – for the networks with the propagation time greater than packet duration a more suited TCP CCA is mAIMD [ALSNQ16], which is a subject of our future work.

Références

[ABDL07] N. Agrawal, W. Bolosky, J. Douceur, and J. Lorch. A five-year study of file-system metadata. *ACM Trans. Storage*, 3(3), 2007.

[ALSNQ16] P. J. Argibay-Losada, G. Sahin, K. Nozhnina, and C. Qiao. Transport-layer control to increase throughput in bufferless optical packet-switching networks. *IEEE J. Opt. Commun. Netw.*, 8(12) :947–961, December 2016.

[dAASBSP17] José Roberto de Almeida Amazonas, Germán Santos-Boada, and Josep Solé-Pareta. Who shot optical packet switching ? In *Int. Conference on Transparent Optical Networks (ICTON)*, number Th.B3.3, July 2017.

[IT17] Salah Ibrahim and Ryo Takahashi. Hybrid optoelectronic router for future optical packet-switched networks. In Sergei L. Pyshkin and John Ballato, editors, *Optoelectronics - Advanced Device Structures*, chapter 04. InTech, Rijeka, 2017.

[KOBT06] A. Kimsas, H. Øverby, S. Bjornstad, and V. L. Tuft. A cross layer study of packet loss in all-optical networks. In *Proceedings of AICT/ICIW*, 2006.

[RX05] George N. Rouskas and Lisong Xu. *Optical Packet Switching*, pages 111–127. Springer US, Boston, MA, 2005.

[SWL14] W. Samoud, C. Ware, and M. Lourdiane. Investigation of a hybrid optical-electronic switch supporting different service classes. In *Photonics North*, volume 9288, pages 928809,1–6, Montreal, Canada, May 2014.

[WSGL16] C. Ware, W. Samoud, Ph. Gravey, and M. Lourdiane. Recent advances in optical and hybrid packet switching. In *Int. Conference on Transparent Optical Networks (ICTON)*, number Tu.D3.4, Trento, Italia, July 2016.