Optical Switching Impact on TCP Throughput Limited by TCP Buffers

Giovane C.M. Moura, Tiago Fioreze, Pieter-Tjerk de Boer, and Aiko Pras

University of Twente
Centre for Telematics and Information Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Design and Analysis of Communications Systems (DACS)
Enschede, The Netherlands
`{g.c.m.moura,t.fioreze,p.t.deboer,a.pras}@utwente.nl`

Abstract. In this paper, we observe the performance of TCP throughput when self-management is employed to automatically move flows from the IP level to established connections at the optical level. This move can result in many packets arriving out of order at the receiver and even being discarded, since some of these packets would be transferred more quickly over an optical connection than the other packets transferred over an IP path. To the best of our knowledge, so far there is no work in the literature that evaluates the TCP throughput when flows undergo such conditions. Within this context, this paper presents an analysis of the impact of optical switching on the TCP CUBIC throughput when the throughput itself is limited by the TCP buffer sizes.

1 Introduction

The *Transmission Control Protocol* (TCP) is the main protocol used for end-to-end communications, representing approximately 90% of all traffic across the public Internet currently [1]. Given its major role, knowledge about TCP performance is important to understand the overall picture of service performance for IP networks. Due to that, its performance has been widely investigated in the networking community [2] [3], and most of the research works identify packet loss as one of the major impacting factors on TCP throughput [4].

Nowadays, higher transmission rates can be achieved by the advent of hybrid communications. One example is optical switching, in which network flows can be forwarded at the optical level instead of relying solely on traditional IP routing. Approaches currently employed to manage optical switched networks rely on human operators to: (*i*) select flows to be moved to the optical level, and (*ii*) create and release the necessary optical connections for such flows.

However, self-management approaches have been investigated with the goal to move automatically *on-the-fly* IP flows to optical connections [5] [6]. By avoiding the IP routing, it is expected that packets switched to the optical level would be delivered more quickly than the remaining others still being forwarded at the IP level. This could cause packets belonging to the same flow to arrive out of order

G. Nunzi, C. Scoglio, and X. Li (Eds.): IPOM 2009, LNCS 5843, pp. 161–166, 2009.

at their destination or even being discarded, which could result in performance problems with the TCP throughput.

Based on the foregoing, the question one could pose is: *what is the impact on the TCP throughput when flows are moved on-the-fly to optical connections?* It is known that TCP performs packet reordering, but little is known about its limits under such circumstances. Thus, the goal of this paper is to present a study on the impact of this *on-the-fly* move on TCP CUBIC throughput, when the throughput itself is limited by the TCP buffer sizes of both sender and receiver.

To address this question, we conducted a series of simulations using the ns-2 network simulator [7], version 2.33. The first step was to review the literature about the different versions of TCP used. Out of many TCP flavors, we chose TCP CUBIC (version 2.1) [8] because it was specially designed for high-speed networks and it is the default version employed in recent Linux kernels. Next, we defined the simulation scenario and finally the simulations were conducted.

The rest of paper is structured as follows. Section 2 introduces the simulation setup used in our simulations. Following that, Sections 3 presents the results obtained and Section 4 describes our conclusions and proposes future work.

2 Simulation Setup

Figure 1 shows the topology used in our simulations. It consists of three routers (`r1`, `r2`, and `r3`) and two nodes (`Sender` and `Receiver`) connected by two different paths: the IP path (`r1-r2-r3`) and the optical path (`r1-r3`).

The simulation starts with the sender opening a single TCP connection with the receiver forwarding data only via the IP path. After reaching pre-defined throughput values, the router `r1` performs the transition of the TCP flow to the optical level, and starts to forward all the data to the receiver via the optical path. It is worth observing that the transition affects only one direction of the flow (`sender` → `receiver`), and that the acknowledgment packets (ACKs) in the reverse direction (`receiver` → `sender`) continue to use the IP path. For a short period of time after the switch – the *transient phase* – there will be data packets on both the IP and optical paths. After the transient phase, the simulation reaches a new phase, in which there are only data packets on the optical path and only ACKs on the IP path. The simulation finishes soon afterwards.

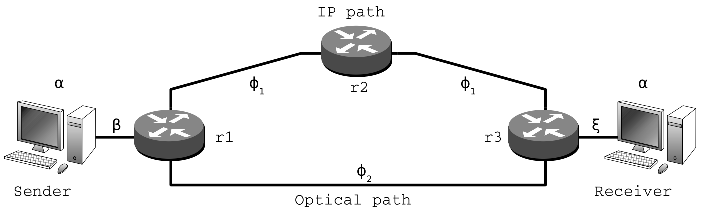


Fig. 1. Topology used in the simulations and limiting factors (Greek letters)

In order to evaluate how the TCP flow is affected by its transition from the network level to the optical level, we analyzed the trace files generated by the ns-2 to compute the throughput perceived by the receiver machine. We are interested not only in observing the maximum throughput, but also in determining *how the throughput changes* after the moment the transition is triggered.

In general, the TCP throughput can be delimited by several factors, such as the kind of application utilized, the TCP buffers sizes employed, and the link capacity [9]. In this work we focus however on the impact of moving flows *on-the-fly* on TCP throughput when flows are limited by TCP buffer sizes.

2.1 Simulation Scenario

Before executing the simulations, we had to calculate the TCP buffer sizes to be used. This value can be obtained by multiplying the flow data rate by the RTT value before the move ($flowRate \times RTT$), and indicates the maximum amount of data on the network at any given time. This value is also the minimum size that TCP buffers must have to handle the specified rates.

Due to that, we had to decide *the moment* at which a flow should be moved to the optical level ($flowRate$) and the RTT values (RTT) to be used. For the evaluated scenario, we have configured the TCP buffers to restrict flows to three data rates (100 Mbps, 1 Gbps, and 10 Gbps) and then we performed the flow transition at each one of these data rates. RTT, in turn, is usually a function of the physical distance between the sender and the receiver. In this study, we have simulated three different situations in which sender and receiver are located in hypothetical cities: (*i*) close to each other, (*ii*) relatively far from each other, and (*iii*) very far from each other. In each case, we assumed that the RTT value after the move would be smaller than before, since packets could avoid the routing process at the IP level. We then have used the RTT values described in Table 1.

After that, we could finally determine the required buffer sizes. Table 2 summarizes the values used in our simulations, where each line represents a single simulation. For each simulation, we have calculated the values for the buffer sizes (αs of Figure 1). Due to space constraints, we present in this table only the cases where RTT is equal to 10 ms, even though an equal number of simulations was conducted for 100 ms and 1000 ms RTT cases. Since we were interested in having TCP buffer sizes as the limiting factor, we over-provisioned the other limiting factors (link speed: β, ϕ, and ξ in Figure 1) to allow the network to handle all the data generated by the sender. These values are presented in Table 2.

The next section shows the results obtained from our simulations.

Table 1. RTT values employed in the simulations

Distance	RTT before moving (ms)	RTT after moving (ms)
Close	10	6
Relatively Far	100	60
Very far	1000	600

Table 2. Values used in simulations for RTT equal to 10 ms

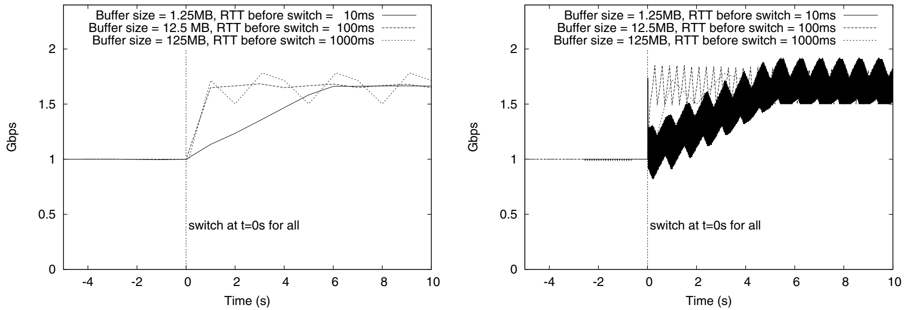
Limiting rate	Buffer size (α) with RTT=10ms	Link speed ($\beta = \phi_1 = \phi_2 = \xi$)
100 Mbps	0.125 MB	622.08 Mbps
1 Gbps	1.25 MB	2.488 Gbps
10 Gbps	12.5MB	39.813 Gbps

3 Simulation Results

This section shows the results of our simulations when the data rate is limited by the TCP buffers. In these simulations, the flow transition was performed at the time $t = 0$s, denoted by a vertical line. One important factor when computing throughput is the granularity (*i.e.*, the time interval used to average the data rate transmitted). If the granularity is very coarse, it may mask some effects that might occur in the network. If it is too fine, it may present a too detailed view, which makes the acquisition of a general view of the throughput more complex. Figure 2 shows the results when the TCP buffers limit the data rate to 1 Gbps.

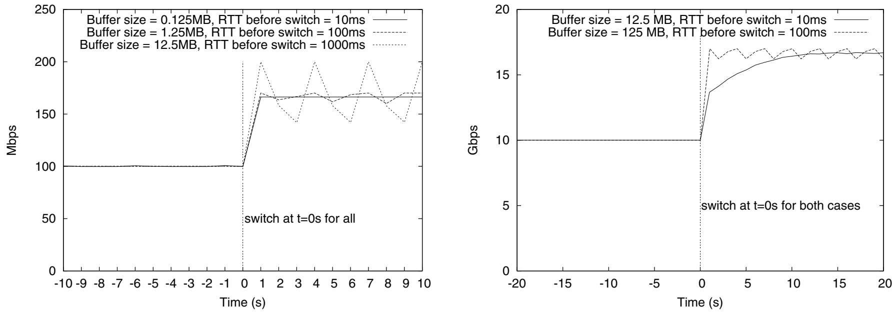
As shown in Figure 2(a), the flows present a stable rate equal to 1 Gbps before the flow transition. However, after the transition ($t = 0$s), it is possible to note that no throughput reduction is observed. In fact, we can see that CUBIC reacts very well to the new configuration, and the throughput increases quickly to the expected new theoretical value of 1,667 Gbps (obtained by dividing the buffer size by the RTT).

Figure 2(b), in turn, presents the throughput results obtained from the same simulation of Figure 2(a), but with different granularity values (equal to the RTT of each case before the switch). With these granularity values, it is possible to see that, in fact, flows experience a reduction on the throughput just after the transition, but they quickly recover from that. What happens is that many packets arrive out-of-order at the receiver, causing many duplicate ACKs to be



(a) Granularity = 1000 ms (b) Granularity = RTT (before switch)

Fig. 2. Throughput of TCP flows in Scenario A (1 Gbps limiting rate)



(a) Limiting rate = 100 Mbps (b) Limiting rate = 10 Gbps

Fig. 3. Throughput of TCP flows in Scenario A

transmitted to the sender (for example, 1372 duplicate ACKs for the 10ms RTT case). Before this simulation, it was not clear how TCP CUBIC would reduce its congestion window (and, ultimately, cause throughput reduction) in face of many packets arriving out-of-order. This simulation however shows that TCP CUBIC performs well and recovers quickly from the reordering conditions imposed by the flow transition.

Figures 3(a) and 3(b) show the throughput when TCP buffers limit the data rate to 100 Mbps and 10 Gbps, respectively, using a granularity of 1000ms. Similar to 1 Gbps case, the throughput also quickly increases after the switch. In Figure 3(b), we have not included the throughput when the RTT is 1000ms because the TCP throughput does not reach the rate selected for comparison (10Gbps) before the transition.

Despite the granularity used, we can conclude that when the buffer size is the limiting factor, the users will experience a brief reduction on the throughput followed by quick increase when a flow transition occurs.

4 Conclusions and Future Work

In this paper, we presented an analysis of the impact of moving flows *on-the-fly* on TCP throughput when TCP buffer sizes act as the limiting factor for the throughput. TCP CUBIC was the TCP flavor employed in our analysis due to its design for high-speed networks and its use in recent versions of Linux kernels. The analysis shown in this paper was performed by using the ns-2 simulator.

When the size of the TCP buffers in the sender and receiver is the limiting factor, we observed that the TCP throughput is slightly reduced just after the flow transition (*i.e.*, the transient phase). However, we observed that the throughput significantly increases when the transient phase is over, due to the smaller RTT values existing in the optical path. The smaller RTT value allowed the sender to have a higher transmission rate in our simulation. Another important observation from our simulations is that TCP CUBIC quickly reacts to new the network conditions available after the flow transition and does not suffer from the massive reordering.

As future work, we plan to evaluate the impact of moving flows *on-the-fly* to the optical level when the limiting factors for the TCP throughput are the link capacities.

Acknowledgments. This research work has been supported by the EC IST-EMANICS Network of Excellence (#26854).

References

1. Kim, M.-S., Won, Y.J., Hong, J.W.: Characteristic Analysis of Internet Traffic from the Perspective of Flows. Computer Communications 29(10), 1639–1652 (2006)
2. Grieco, L.A., Mascolo, S.: Performance evaluation and comparison of Westwood+, New Reno, and Vegas TCP congestion control. SIGCOMM Comput. Commun. Rev. 34(2), 25–38 (2004)
3. Koutsonikolas, D., Dyaberi, J.M., Garimella, P., Fahmy, S., Hu, Y.C.: On TCP throughput and window size in a multihop wireless network testbed. In: WINTECH, pp. 51–58 (2007)
4. Chandran, K., Raghunathan, S., Venkatesan, S., Prakash, R.: A feedback-based scheme for improving TCP performance in ad hoc wireless networks. IEEE Personal Communications 8(1), 34–39 (2001)
5. Fioreze, T., Pras, A.: Self-management of lambda-connections in optical networks. In: Bandara, A.K., Burgess, M. (eds.) AIMS 2007. LNCS, vol. 4543, pp. 212–215. Springer, Heidelberg (2007)
6. Fioreze, T., van de Meent, R., Pras, A.: An architecture for the self-management of lambda-connections in hybrid networks. In: Pras, A., van Sinderen, M. (eds.) EUNICE 2007. LNCS, vol. 4606, pp. 141–148. Springer, Heidelberg (2007)
7. ns2. The Network Simulator NS-2, `http://www.isi.edu/nsnam/ns/`
8. Ha, S., Rhee, I., Xu, L.: CUBIC: a new TCP-friendly high-speed TCP variant. SIGOPS Oper. Syst. Rev. 42(5), 64–74 (2008)
9. Timmer, M., de Boer, P.-T., Pras, A.: How to identify the speed limiting factor of a tcp flow. In: Proceedings, 4th IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services, April 2006, pp. 17–24 (2006)