Investigation of Ethernet switches behavior in presence of contending flows at very high-speed

Sébastien Soudan*, Romaric Guillier*, Ludovic Hablot*, Yuetsu Kodama‡,
Tomohiro Kudoh‡, Fumihiro Okazaki‡, Ryousei Takano‡ and Pascale Vicat-blanc Primet*
*LIP, UMR INRIA-CNRS-ENS Lyon-UCB Lyon 5668
École Normale Supérieure de Lyon, France
Email: ssoudan@ens-lyon.fr
‡National Institute of Advanced Industrial Science and Technology (AIST), Japan

Abstract— **This paper examines the interactions between high speed Ethernet switches and TCP in high bandwidth delay product networks. First, the behavior of a range of Ethernet switches when two long lived connections compete to the same output port is investigated. Then we study the impact of these switches behaviors on TCP protocols in long and fast networks (LFNs). Several conditions in which scheduling mechanisms introduce heavy unfair bandwidth sharing and loss burst which impact TCP performance are shown.**

Keywords: Ethernet switches, queue management, high speed transport protocol, cross-layering

I. Introduction

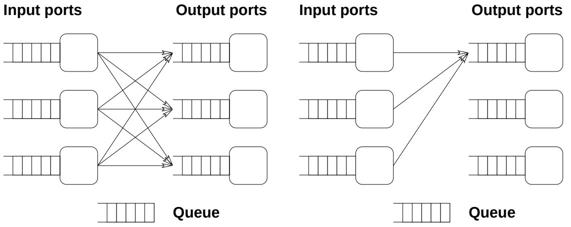
Most transport protocol designers addressing wired networks do not take link layer behaviors into account. They assume a complete transparency and determinist behavior (i.e. fairness) of this layer. However, Ethernet switches are store-and-forward equipments, which have limited buffering capacities to absorb congestions that can brutally occur, due to bursty nature of TCP sources [1]. Thus many Ethernet switches use contention algorithms to resolve access to a shared transmission channel [2]–[4]. These scheduling algorithms aim at limiting the amount of data that a subnet node may transmit per contention cycle. This helps in avoiding starvation for other nodes. The designers of these algorithms have to find a trade-off between global performance and fairness of these equipments in a range of traffic conditions [5]. Considering the case of grid environment where many huge data transfers may occur simultaneously, we explore the interactions between these layer 2 congestion control and scheduling mechanisms and TCP and try to understand how they interfere. The issue is to understand how the bandwidth and losses are distributed among flows by switches when traffic profiles correspond to huge data transfers in long distance high speed networks, and the implication on transport protocols design.

After a brief introduction on switches design and their arbitration algorithms, the second section of this paper details the experimental protocol adopted and the observed parameters. In the third part, we look, from a packet level point of view, at the steady-state behaviors of constant bit rate flows crossing several types of switches. In the fourth section, we study the interaction of such behavior with transport level protocols. The paper ends by a discussion on the problems of switch design in long distance Ethernet networks and datagrid context.

II. Ethernet Switches design and algorithms

Developers of Ethernet switching equipment face the big challenge of providing high performance and flexible equipments while driving equipment cost as low as possible. Most of current non-blocking high speed Ethernet switches are built around a crossbar switch using a fixed-size cell as a transfer unit. A crossbar switch is simple to implement and it allows multiple cells to be transferred across the fabric simultaneously, alleviating the congestion found on a conventional shared backplane.

However, when several cells destined for the same output arrive in a time slot, at most one can actually leave the switch; the others must be buffered. There are many options for organizing the buffer pools. Buffers may be placed at the switch inputs, at the outputs, at both inputs and outputs, or at a centralized location. Output-queueing is a queueing technique in which all queues are placed at the outputs.There are no queues at the inputs. All arriving cells must be immediately delivered to their outputs which is a limiting factor at very high speed. When there are queues at the inputs the memory is only required to operate at twice the line rate, making input-queueing of interest for high-bandwidth switches. Unfortunately, it is known that an input-queued (IQ) switch with a single FIFO queue at each input performs poorly due to head-of-line (HOL) blocking limiting the achievable bandwidth to approximately 58.6% of the maximum. A viable solution - the virtual output queueing - has been introduced for HOL blocking elimination. However, the scheduling problem in VOQ switches is more complex than the one in single FIFO switches. VOQ switches maintain several queues at each input. VOQ switches require the use of a scheduler to configure the switch, deciding which input to connect to which output in each packet-time. In this case, the scheduler determines the performance of the switch: the throughput of the switch, the delay experienced by each packet and the number of packets lost due to buffer overflow. Among the proposed VOQ algorithms, Parallel Iterative matching (PIM) [3], iSLIP [4] and wave front arbiter (WFA) were demonstrated to be practical for high-bandwidth switches and shown to achieve 100% or close to 100% throughput when **the traffic is uniform**. PIM uses a random approach by choosing randomly packets among contending input ports whereas iSLIP uses a round-robin. Most of algorithm validation have considered only uniform traffic patterns (Figure 1(a)): a uniform distribution between all ports, all flows presenting an equal rate and Bernouilli i.i.d. (independent and identically distributed) arrivals. Nonhomogeneous systems where traffic intensity at each input varies and destination distribution is not uniform (Figure 1(b)) have been little considered because such traffic patterns are difficult to define exhaustively. From a global point of view, all algorithms attempt to achieve fairness among input port but from a local point of view results can be quite different [6, chapter 13]. In general, detailed design and algorithms adopted in switches are not revealed by the equipment provider. This study explores real switches behavior under imbalanced traffic pattern conditions corresponding to the realistic case of simultaneous bulk data transfers in datagrids using high speed TCP variants.



(a) With uniform traffic pattern (b) With non-uniform traffic pattern

Fig. 1. 3x3 ports switch

III. Experiment description

To observe the bandwidth sharing, loss and fairness patterns on a congested output port of a switch, a specific testbed and a restricted parametric space were used to explore 1 Gbps Ethernet switches behavior. Experiments described in this paper are available in the research report [7].

A. Testbed description

The experimental testbed consists of two sources connected to two ports and one common sink connected to a third port of a switch. GtrcNET-1[1] [8] is used to both generate traffic and monitor the output flows. GtrcNET-1 is an equipment made at the AIST which allows latency emulation, bandwidth limitation, and precise per-stream bandwidth measurements in GigE networks at wire speed. GtrcNET-1 has 4 GigE interfaces (channels). Tests were performed with several switches but most of the results presented here are based on a Foundry FastIron Edge X424 and a D-Link DGS1216-T. Firmware version of the Foundry switch is 02.0.00aTe1. “Flow control” is disabled on all used ports and priority level is set to 0. According to manufacturer’s documentation, D-Link has 512 KB and according to command line interface `show mem` command, Foundry switch has 128 MB of RAM but for both the way memory is shared among ports is not known.

B. Parametric space

The first set of experiments considers two contending flows at different constant rates. This permits to observe the switch behavior under different congestion levels, to highlights the differences between switches using mean and variance of per-flow output bandwidth. Fine-grained observations have been made using sequence numbers to observe per-packet switching behaviors in presence of two flows. We assume L2 equipments do not differentiate UDP and TCP packets. Tests that have been made corroborate this fact. Layer 2 experiments were then conducted with UDP flows as they can be generated easily by GtrcNET-1 and as they can be sent at a constant rate.

The following parameters were explored: flows’ rate, packet length and measure interval length. Different high levels of congestion using different flow’s rates were used : 800, 900, 950 and 1000 Mbps . These rates are transmission capacity (TC) used to generate UDP packets. Transmission capacity specifies the bitrate (including Inter Frame Gaps and preamble) of an emulated Ethernet link. Experiments were strictly included in the period of packet generation. IP packet length is set to 1500 bytes as high-speed connections use full size packets. In order to observe output flow bandwidths, packets were counted on intervals of 400 and 1000 μs (around 33 and 83 packets at 1 Gbps).

IV. Steady-state behaviors of long constant bit rate flows

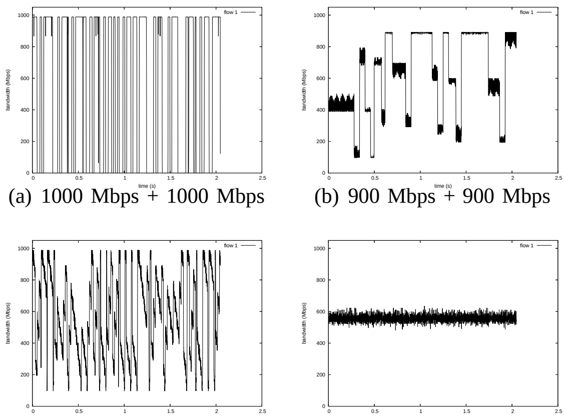
This section presents some of the bandwidth patterns observed on the output port of the Foundry Fast IronEdge X424 switch when two long CBR (Constant Bit Rate) flows are sent through this output port. We only concentrate on 1500 bytes packets as high throughput flows use such packets size (or over with jumbo frames). Only one of the two flows is represented. In all experiments, the sum was always constant at 986 Mbps. Measures are made using 1 ms intervals.

A. Two CBR flows with same rates

In figure 2(a), flows have the same input rate and it can be observed that they are strictly alternatively forwarded. There are many changeovers between the two flows but they appear to be completely random. The aggregated bandwidth is nearly constant and one flow can starve for more than 100 ms (for example: flow 1 between 1.6 s and 1.7 s). In the case of 64 bytes packets with 1 Gbps rates, we observed a similar behavior except that during the changeovers, the bandwidth is more fairly shared (600 Mbps-250 Mbps). The aggregated bandwidth is nearly constant but not optimal.

When both sending rates are less than the maximum (for example 900 Mbps with congestion level of 180% in figure 2(b)), flows do not starve but a real unfair sharing is observed for more than 300 ms. For example, from time 1.45 s to 1.75 s, one stream is running at more than 900 Mbps and the other at less than 50 Mbps.

[1] http://www.gtrc.aist.go.jp/gnet/



(a) 1000 Mbps + 1000 Mbps (b) 900 Mbps + 900 Mbps

(c) 1000 Mbps + 900 Mbps (d) 950 Mbps + 900 Mbps

Fig. 2. Output bandwidth of flow 1 when two flows share one output port (1500 bytes packets)

B. Two CBR flows with different rates

Figure 2(c), shows that when the two flows are sending at different rates (with one at wire speed), instant flow rate on the output port varies among a set of values. When none of the flows is at 1 Gbps speed and flows do not have the same rate (figure 2(d)) the sharing is closer to what would be expected (fair sharing). With 1 ms interval observation the aggregate throughput is nearly constant and optimal.

We conclude that sharing amongst input ports is really unfair on “short” time scale when sending rates are equal or when one is at wire speed with this specific nonblocking high speed switch. Each flow alternatively loss bursts of packets. Around 8300 packets of these size should have been lost at 1 Gbps speed within 100 ms. When none of the two rates is at wire rate, the sharing is much better. Note that in the other cases, the rate limitation is obtained by pacing packets. The following sections present some complementary measurements for a range of gigabit ethernet switches.

C. Quantitative measures for CBR flows with a range of switches

In this section, statistical metrics for two flows crossing different switches with different input rates are presented. Throughput measurements have been made using 100 ms intervals. It can be observed that table I shows a very high variance and minimum throughput of 0 Mbps when the input rates are equal to 1000 Mbps whereas with D-Link switch the throughput is always equal to 494 Mbps. Cisco 3750 and 4948 show similar behaviors as Foundry switch while Huawei S5648 and Dell 5224 are close to D-Link switch. The three first switches tend to make one of the flows starve for periods of time longer than 100 ms when the congestion is severe. These three switches also perform unfair sharing under high congestion whereas the three last always split the available bandwidth around 494 Mbps when the input rates are equals. With the D-Link switch, it also occurs when the input rates are different and the output port is congested.

Input rate		CH0 (Flow 1)				CH0 (Flow 2)			
CH2	CH3	ave	max	min	var	ave	max	min	var
1000	1000	800	988	666	19K	188	323	0	19K
800	800	197	197	197	0	792	791	792	0
500	500	494	494	494	0	494	494	494	0
800	600	569	574	566	4	419	422.52	414	3

TABLE I

TWO CBR FLOWS ON FOUNDRY FASTIRON EDGE X424

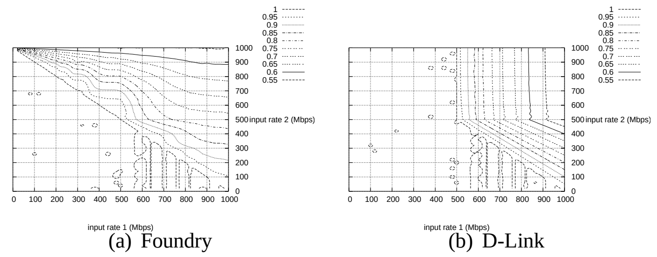
To conclude this section, it seems switches behavior divide in two different classes. In the first one, which corresponds to non-blocking switches, starvation can occur and high variance under severe congestion can be experienced. In the second one, which are much simpler switches, low variance and no starvation occurs. As TCP connections have no knowledge of which switches are in the network, it can be guessed that the behavior and performances of the connections can be highly and differently impacted (see section V).

D. Steady-state switch’s characteristic for CBR flows

While previous section showed metrics in a small number of situations for several switches, this section presents some metrics for two switches for a range of input rates (from 0 to 1 Gbps by 20 Mbps). In order to characterize switching behaviors, the ratio of output bandwidth divided by input bandwidth were measured for each input rates (from 0 to 1 Gbps by 20 Mbps). Standard deviation of the rates were also measured. Each experiment lasts 12 seconds, measurements have been done on 1 ms intervals and have been repeated 3 times.

Figure 3(a) shows the isoline of the ratio of output bandwidth divided by input bandwidth of flow 1 on Foundry switch (figures for flow 2 are similar but symmetrical). X axis is the input rate of the first flow and Y axis the one of the second flow. It can be observed that the isolines tends to join at one point. When there is no congestion (below the line joining (0, 1000) and (1000, 0)), the ratio is equal to 1, in other words there is no drop. Figure 3(b) shows the D-link switch’s behavior is completely different and probably related to the switches design. If the input rate of the flow 2 is less or equal to 500 Mbps, its output rate is always equal to the input rate regardless of the input rate of flow 1 as we can observe on the left side of the figure. This switch as probably no input queues and manages contention with a simple round robin mechanism.

Figure 4 shows the standard deviation of the output bandwidth for the flow 1 with the Foundry switch (with 1 ms measurements’ interval). The standard deviation is quite low in the usual case. But when the input rates are the same or when one of them is at the maximum, more deviation can be observed. The highest standard deviation is obtained when the two flows are at 1 Gbps. That is when alternate complete starvation of one of the flows was observed.



(a) Foundry (b) D-Link

Fig. 3. Isoline of output bandwidth over input bandwidth of flow 1 on Foundry switch

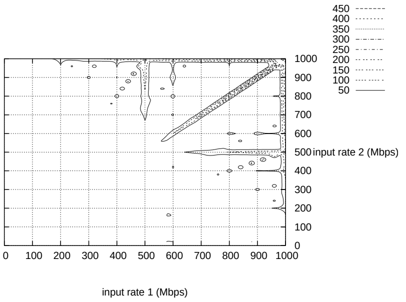
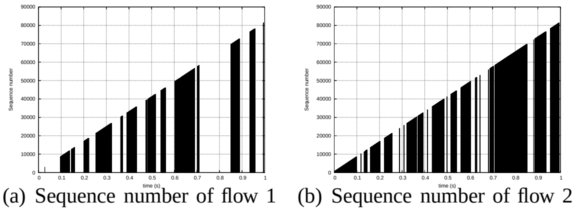


Fig. 4. Isoline of standard deviation of output bandwidth of flow 1 on Foundry switch

E. Sequence number analysis

In this section, instead of monitoring the output bandwidth, the sequence numbers of forwarded packets are monitored. Figure 5 shows the situation with two 1000 Mbps flows and figure 6 with two 400 Mbps flows on Foundry switch. In these figures, the sequence number of a packet at the date it was observed on the output port is represented by an impulse. It can be noticed that when the two flows are at max speed, (figure 5), only one flow is forwarded at a time most of the time. When flows sending rate are less than half of the capacity, output packets are picked alternatively from the two flows (figure 6 is a zoom-in of a short interval).



(a) Sequence number of flow 1 (b) Sequence number of flow 2

Fig. 5. Evolution of sequence number of packets on the output port (1000 Mbps + 1000 Mbps (1500 bytes) on Foundry switch)

On D-Link switch, even when the two input rates are 1000 Mbps, packets are forwarded alternatively from the two ports while others are alternatively dropped. Sequence numbers of forwarded packets are growing by from 1 to 3 as there is only 1000 Mbps of bandwidth on output port and some of the input's packets have to be dropped. This is different from the Foundry switches where packets are dropped by burst. The case with 400 Mbps flows and D-link switch is similar to Foundry one. This confirms the two tested switches have likely different design and queue management strategies. Next section examines how these Ethernet switches performance impact TCP performance.

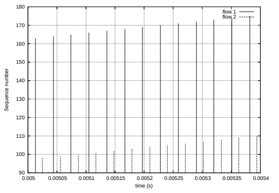


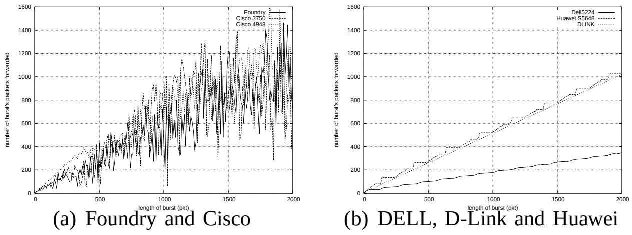
Fig. 6. Evolution of sequence number of packets on the output port (400 Mbps + 400 Mbps (1500 bytes) on Foundry switch)

V. IMPACT ON TCP

As TCP uses a congestion avoidance mechanism, one can assume this prevents such high congestion level on switches' output ports to occur. However in the slow start phase as the congestion window is doubling at each RTT and during aggressive congestion window increase phases (as in BIC [9]), flows can send very long trains of back-to-back packets and face severe congestions in Ethernet equipment.

A. Slow start experiment

In this section, we study the impact of L2 packet scheduling algorithms on already established flows when a new connection starts. The testbed used is similar to the one presented before but the first CBR flow was replaced by a burst of variable length. In this experiment, the bandwidths obtained by the CBR flow and the burst were measured. We assume that the amount of CBR flow's lost bandwidth corresponds to a number of packets lost as in a long run situation, the switch can't buffer all the packets. Figure 7 represents the estimated number of loss that the first flow experienced as a function of the length of the burst with different switches. It can be seen that generally the burst gets most of his packets going through the output port, which causes a large dent on the CBR flow. But again two different behaviors can be observed. Figure 7(b) shows very regular lines for the DELL, D-Link and Huawei switches whereas they are very noisy for the Cisco and Foundry switches (figure 7(a)) which might indicate these switches use more sophisticated algorithms.



(a) Foundry and Cisco (b) DELL, D-Link and Huawei

Fig. 7. 1000 Mbps CBR flow's losses due to a burst as a function of its length

As switches induce different packets drop patterns in congested situation, the next section will explore how different TCP variants under various latencies, with and without SACK on blocking (D-Link) and nonblocking (Foundry) switches, adapt to these behaviors. Here, due space limit, we concentrate on BIC and Reno and two latency 0 ms RTT and 50 ms RTT latencies results.

B. Comparison of TCP variants behavior on different switches

Experiments use four hosts: two senders and two receivers, all running `iperf` on 2.6.17 linux kernel. The two flows involved share a 1 GbE link of configurable RTT: 0 ms or 50 ms. Bottleneck takes place in the switch before this link. We observe the two flows on this link using the GtrcNET-1 box. All the experiments share the same experimental protocol: first flow is started for 400 s, 20 s later second flow is started for 380 s. In these experiments, TCP buffers were set to 25 Mbytes and `txqueuelen` to 5000 packets to avoid software limitation on end hosts.

Figures presented in this section represent flows' throughputs on 0 ms and 50 ms RTT GigE links.

Comparison between figures 8 and 9 which differ only by the switch used, shows that even when important buffers are not needed because the latency is small and so is the congestion window, packets scheduling algorithms can impact TCP behaviors. We can observe a higher variability of throughput and period of starvation on figure 9.

When the latency is more important, it become more difficult to charge buffer size or scheduling algorithm but we can also observe on figures 10 and 11 that protocols are impacted differently. On figure 11 we can observe the consequence of an important buffering inside the switch which artificially increases the RTT from 52 ms at 75 s to 80 ms just before time 150 ms.

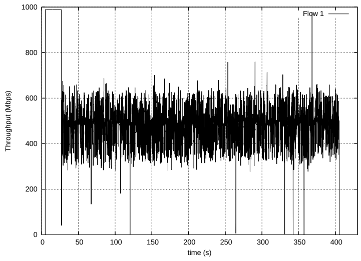


Fig. 8. Throughput evolution of one of the two BIC flows with SACK (0 ms RTT) on D-Link switch

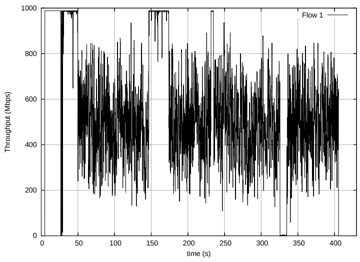


Fig. 9. Throughput evolution of one of the two BIC flows with SACK (0 ms RTT) on Foundry switch

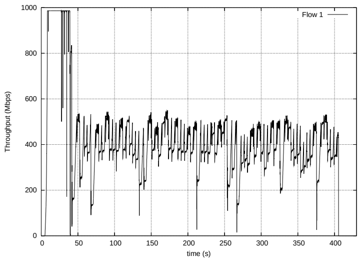


Fig. 10. Throughput evolution of one of the two BIC flows with SACK (50 ms RTT) on Foundry switch

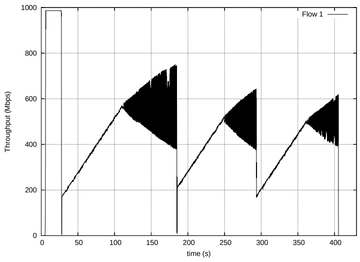


Fig. 11. Throughput evolution of one of the two Reno flows with SACK (50 ms RTT) on Foundry switch

On table II, we can observe that mean goodput is higher when using Foundry switch than D-Link one. Table III highlights the fact that there is more retransmission with D-Link switch than with Foundry with 0 ms RTT. In this case, both flows tend to send packet back to back which is the worse case in term of contention put to the switch.

Figures and tables of this sections have shown that the behaviors and performances of TCP variants on different switches can dramatically vary. BIC protocol, which is more aggressive than Reno tends to lose more packets but is able to react more quickly.

TCP variant	Sack?	Switch	Mean goodputs	
			0 ms RTT	50 ms RTT
BIC	Yes	Foundry	524 & 417	394 & 372
		D-Link	469 & 468	208 & 168
	No	Foundry	381 & 471	224 & 192
		D-Link	366 & 361	146 & 118
Reno	Yes	Foundry	480 & 461	412 & 345
		D-Link	436 & 434	180 & 141
	No	Foundry	452 & 433	516 & 254
		D-Link	379 & 364	177 & 154

TABLE II

MEAN GOODPUTS OF 2 FLOWS SHARING ONE PORT FOR 380 S (MBPS)

VI. DISCUSSION

This work on interaction between transport protocols and layer two equipments in the context of high speed wired networks highlights different behaviors and level of performance of these protocols in specific situations. Switches have been evaluated in an extreme situation which is not likely to be

TCP variant	Sack?	Switch	Retransmission per seconds	
			0 ms RTT	50 ms RTT
BIC	Yes	Foundry	63.2 & 58.1	15.0 & 15.8
		D-Link	217.6 & 227.2	4.4 & 2.8
	No	Foundry	43.3 & 53.4	6.6 & 7.7
		D-Link	471.2 & 493.2	4.5 & 3.9
Reno	Yes	Foundry	50.2 & 50.0	3.4 & 0.3
		D-Link	193.5 & 192.3	0.2 & 0.6
	No	Foundry	47.5 & 46.8	0.4 & 0.5
		D-Link	94.0 & 100.6	0.1 & 0.1

TABLE III

NUMBER OF RETRANSMISSIONS PER SECONDS FOR 2 FLOWS SHARING ONE PORT FOR 380 S (PKT/S)

the one for which switches algorithms were optimized. We considered non-uniform traffic: only three ports (two input ports and one output port) among more than 24 are used and put to a high congestion level. However, this situation is not so uncommon in a datagrid context where large amount of data can be moved between nodes with a low multiplexing level and a non-uniform distribution of sources and destinations. For example, grids often use Ethernet over DWDM as long distance clusters interconnection. Congestions between flows generally take place in Ethernet switches and occur on a long latency link or a local link depending on nodes involved.

We have seen that sophisticated switching algorithms of nonblocking switches do not handle very predictably such stressing conditions. Further investigations are then needed to understand what really happen in switches and how to improve protocol in this particular context. For example, are "uplink" ports managed differently? How is the memory managed? What is the buffer length of a port? Is there different switching strategies applied depending on inputs' "load"? When isback-pressure through Ethernet PAUSE packets triggered?

To better design high speed transport protocols over next-generation carrier Ethernet networks, we argue it would be useful switches vendors to publish a precise description of their products. We also think protocols developers should take into account a large spectrum of loss pattern and be very careful in their testbed design when validating their proposals.

On an other hand, designing switching algorithms and Ethernet equipments taking into account future contending large data movements could be of importance if such applications spread out as it is envisioned. Observed performance with such non-uniform traffic are not optimal (the throughput of the output ports is not maximized), the sharing is not fair and performance are not predictable. Developing switches optimized for grid application certainly would have advantage. Nevertheless, switches designers should be aware that Grid applications need features that are different from those required by traditional best effort (e-mail) and real-time services (VoIP).

VII. CONCLUSION

Packet scheduling algorithms for Ethernet equipments have been designed for heterogeneous traffics and highly multiplexed environments. Nowadays Ethernet switches are also used in situations where these assumptions can be incorrect such as grids. This paper reveals several conditions in which scheduling mechanisms introduced in non-blocking switches introduce heavy unfairness (or starvation) on large intervals (300 ms) and loss bursts which impact TCP performance. These conditions correspond to situations where several huge data movements occur simultaneously. It also shows that behaviors are different from switch to switch and not easily predictable. These observations offer some tracks to better understand layer interactions. They may explain some congestion collapse situations observed in real experiments and why and how parallel transfers mixing and pacing packets of different connections take advantages over single stream transfers.

We plan to pursue this investigation of layer 2-layer 4 interactions and explore how to model it and better adapt control algorithms to fit the new applications requirements. We also plan to do the same precise measurements with flow control (802.3x) and sender-based software pacing [10] which both tend to avoid queue overflows and modify packets interarrival.

VIII. ACKNOWLEDGMENTS

This work has been funded by the French ministry of Education and Research, INRIA, and CNRS, via ACI GRID's Grid5000 project and ACI MD's Data Grid Explorer project, the IGTMD ANR grant, NEGST CNRS-JSP project, and the INRIA GridNet FJ team. A part of this research was supported by a grant from the Ministry of Education, Sports, Culture, Science and Technology (MEXT) of Japan through the NAREGI (National Research Grid Initiative) Project and the PAI SAKURA 100000SF with AIST-GTRC.

REFERENCES

[1] H. Jiang and C. Dovrolis, "The origin of TCP traffic burstiness in short time scales," in *IEEE INFOCOM*, 2005.

[2] M. Fayyazi, D. Kaeli, and W. Meleis, "Parallel maximum weight bipartite matching algorithms for scheduling in input-queued switches," *IPDPS*, vol. 01, p. 4b, 2004.

[3] T. E. Anderson, S. S. Owicki, J. B. Saxe, and C. P. Thacker, "High-speed switch scheduling for local-area networks," *ACM Trans. Comput. Syst.*, vol. 11, no. 4, pp. 319–352, 1993.

[4] N. McKeown, "The iSLIP scheduling algorithm for input-queued switches," *IEEE/ACM Trans. Netw.*, vol. 7, no. 2, pp. 188–201, 1999.

[5] N. McKeown and T. E. Anderson, "A quantitative comparison of iterative scheduling algorithms for input-queued switches," *Computer Networks and ISDN Systems*, vol. 30, no. 24, pp. 2309–2326, December 1998.

[6] G. Varghese, *Network Algorithmics: an interdisciplinary approach to designing fast networked devices*. Elsevier, 2005.

[7] S. Soudan, R. Guillier, L. Hablot, Y. Kodama, T. Kudoh, F. Okazaki, P. Primet, and R. Takano, "Investigation of ethernet switches behavior in presence of contending flows at very high-speed," INRIA, Research Report 6031, 11 2006. [Online]. Available: https://hal.inria.fr/inria-00115893

[8] Y. Kodama, T. Kudoh, T. Takano, H. Sato, O. Tatebe, and S. Sekiguchi, "GNET-1: Gigabit ethernet network testbed," in *Proceedings of the IEEE International Conference Cluster 2004*, San Diego, California, USA, Sept. 20-23 2003.

[9] L. Xu, K. Harfoush, and I. Rhee, "Binary increase congestion control for fast long-distance networks," in *IEEE INFOCOM*, 2004.

[10] R. Takano, T. Kudoh, Y. Kodama, M. Matsuda, H. Tezuka, and Y. Ishikawa, "Design and evaluation of precise software pacing mechanisms for fast long-distance networks," in *PFLDnet*, Lyon, France, 2005.