From Packets to XLFrames: Sand and Rocks for Transfer of Mice and Elephants

Dinil Mon Divakaran*, Eitan Altman†, Georg Post§, Ludovic Noirie§, Pascale Vicat-Blanc Primet*
* INRIA / Université de Lyon / ENS Lyon, {Dinil.Mon.Divakaran,Pascale.Primet}@ens-lyon.fr
†INRIA, Eitan.Altman@sophia.inria.fr
§Alcatel-Lucent Bell Labs, {Georg.Post,Ludovic.Noirie}@alcatel-lucent.fr

***Abstract*—Looking into the future, this paper presents the effects of having packets of large sizes, called XLFrames (XLFs), in a network. The analysis is motivated by the fact that the Internet is soon to witness stupendous amounts of traffic that have to be processed and switched at amplifying line rates; and this brings forth multiple challenges in the form of energy efficiency, network performance and end-host performance. Increasing the size of packets in the Internet has far-reaching incentives that otherwise appear hard to achieve. We foresee an Internet that multiplexes both packets (sand) and XLFs (rocks). As a first step, we analyse the effects of introducing XLFs in a network, and find the following: (i) the amount of packet-header processing is greatly reduced, (ii) while the fair multiplexing of XLFs with standard packets can be achieved using a careful queue management in routers.**

***Index Terms*—Future Internet, Packet size**

I. Introduction

The phenomenal growth of the Internet has been accompanied by the core of the Internet moving from capacities of tens of Gbits/s to hundreds and thousands of Gbits/s. It is just a matter of time before the access links will have matching capacities to flood the network with mammoth data. Such an unprecedented growth brings multiple challenges at end-hosts, switches and routers.

Though the changes in capacities have come along with changes in traffic trends (from web to P2P to video over Internet) [1]), what has remained relatively static across the history of the Internet is the standard packet size. Indeed, researchers in the past pointed out the necessity of raising the MTU, but the reasons posed then were mainly to improve the performance at the end-hosts [2]. On the other hand, we motivate the need for large packets with an aim to improve the efficiency of the network and the equipments (switches and routers), besides the end-hosts. This is elaborated in Section II.

We foresee a future Internet that multiplexes classical packets as well as large packets, called *XLFrames* (XLFs), on high-speed links. Some of the reasons for classical packets (sand) to exist along with XLFs (rocks) are: (i) Not all end-hosts will have the ability to send XLFs, say, due to hardware constraints, or restricted by BER of wireless channels (ii) Delay constraint applications, eg. voice, running on hosts connected with Mbits/s access rates might stick to packets to avoid intolerable round-trip delays. (iii) A TCP connection has short packets like control packets and ACKs (iv) Small flows (*mice*), e.g HTTP requests, are expected to use packets as there might not be sufficient data to transfer.

In this paper, we perform studies to analyse the effects of introducing XLFs in a network that carries classical packets. Specifically, we focus on the following questions: (i) What is the gain in terms of processing power? (ii) What is the effect in terms of throughput? (iii) With minimal changes to existing protocols (TCP/UDP, IP), how do XLFs integrate in the current scenario? These questions are addressed in Sections V and VI. We also sketch some of the issues with XLFs in Section VII.

This work was done in the framework of the INRIA and Alcatel-Lucent Bell Labs Joint Research Lab on Self Organized Networks.

II. Motivation

Huge amounts of traffic implies processing of large number of packets per unit time at terminals and network nodes. The increase in processing per unit of time also increases the power consumption at the equipments. The computing power required would increase too much unless some drastic measures are taken [3]. *Therefore minimizing power consumption at equipments is one important criteria that any research on Internet architecture should focus on* [4].

The growing line rate raises another major concern. The increase in processing power and memory speed is slower compared to the increase in transmission rates. Parallel, pipelined processing and the use of fast SRAM can keep up with reduced inter-packet time only at increased complexity and cost. As the bottleneck moves from transmission capacity to processing power and/or memory access time, the maximum throughput achievable by a flow becomes less than the line capacity. *Achieving maximum throughput is another goal.*

The cost of performing packet-level functions at line rate can be partitioned into two: per-byte processing cost and per-packet processing cost. Functions such as route lookup, classification, arbitration, scheduling etc. have per-packet costs. Besides, flow-level functions also add to per-packet cost. If the trend towards flow-aware networking is anything to go by [5], additional flow-level functions (such as flow table lookup, estimation of flow parameters, flow policing etc.) will add to per-packet cost. *Per-packet processing cost reduces if the number of packets that need to be processed per unit time is reduced.*

At an end-host, the costs in terms of protocol processing and interrupt handling for packets make it challenging to achieve 100% throughput at high line rates [6]. The rule of thumb for the bit rate to processing ratio is $1\ bps$ / $1\ Hz$, indicating the difficulty in utilizing line rates of 10 Gbps and higher. But, according to a study [7], it is hardly possible to achieve $1\ bps$ for every $1\ Hz$. The authors also note that packet transmission/reception form a substantial part (28 - 40%) of commercial server workloads. Indeed, the problems faced at the end-hosts are scaled-up versions of what are seen at the equipments. Though methods like TOE (TCP Offload Engine) and packet coalescence have been proposed, it is not clear if these are long term solutions. To achieve high throughput, and at the same time, to save CPU cycles for other applications, *it is necessary to minimize the processing cost involved in protocol processing and interrupt handling*.

We argue that all the above points are sound motivating factors to break the barrier of traditional packet size; and look into a future where data is transported in *XLFs*.

III. Related work

The concept of large packet sizes has been floating around for a decade [2], [8]. *Jumbo frames* was introduced to solve some of the end-host related problems. The limiting factor of the MTU comes from the early Ethernet designs. Today's Gigabit Ethernet NICs support packet sizes of 9000B and even larger [9], [10], [11]. Large packets are also supported in research networks such as Internet2 and GEANT. Wang *et al.* showed that in Ethernet-based storage area networks, the use of large packets reduce CPU utilization and interrupts notably, while improving the throughput during data transfers [12]. The use of larger MTUs also contributes to high throughput achieved on iSCSI storage networks [13].

Researchers have also put forward the idea of aggregating packets in the network, at the edges. In [14], the authors propose to dynamically encapsulate packets into large packet at the ingress of a domain, and sent out to the network with an additional new header. At the egress of the domain the original packets are decapsulated from the large packet and transmitted to the destination. A timer is used to decide on the number of packets that will be encapsulated. This is similar to the burst assembly process in OBS (Optical Burst Switching), one of the paradigms in optical networks [15]. The burst assembly process is known to have a significant impact on TCP performance [16], for example flow synchronization.

A. Our focus

XLFs are more relevant today than in the past mainly to improve the network performance and energy efficiency as discussed above. We attempt to contribute to, (i) a rigorous study and analysis on how XLFs fit into today's Internet architecture, and (ii) based on the analysis, make necessary modifications to the present queueing and scheduling policies, and even to the protocols (like TCP) if need be, for incorporating XLFs.

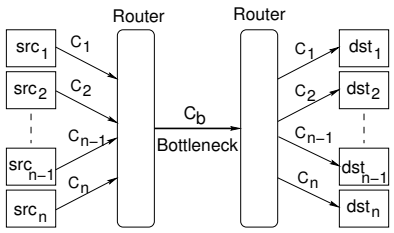


Fig. 1. Topology considered for simulations

IV. Simulation setup

We simulate the multiplexing of TCP flows with large and small packets using NS-2; for all the simulations, we consider a dumbbell topology, connecting n *src-dst* pairs (see Fig. 1). Packet size is taken as 1500B. XLF size is in number of packets; i.e., an XLF of size F has $F \times 1500$B in it. We use *data units* to refer to a transporting unit independent of its size (a data unit can be a packet or an XLF). Fixing the bottleneck capacity as C_b, the capacities of links from *src* nodes to the bottleneck, and from the bottleneck to *dst* nodes, are set depending on the level of congestion desired for the scenario. The congestion factor is defined here as $\rho = \frac{\sum_{i=1}^{n} C_i}{C_b}$, C_i being the capacity of link i.

The bottleneck queue size is set in bytes, as the bandwidth delay product (BDP) for 100 ms. The queue size of each source is $1.2 \times BDP$. TCP window size is set high enough so as to be limited only by the network (and not by the end-hosts). TCP Sack is used in all scenarios, with no delayed ACKs. Since simulating events using links with 10 Gbps or higher has practical difficulties, $C_b = 1\ Gbps$ for simulations. Simulations are run for 540s. For all measurements, we ignore the first 60s of the simulation to avoid transient states. The metrics measured, at the bottleneck link, are throughput and drop rate of individual and aggregate flows. For a single flow, the throughput is measured as the number of bytes transmitted during an interval; whereas the drop rate equals the number of data units dropped (due to the queue being full) over the total number of data units sent.

V. Gains

A. Processing gain

The reduction in the number of data units at an equipment results in processing gain. A recent study shows, the packet size distribution is no more trimodal, but rather bimodal, with nearly 50% of packet lengths between 40 and 100B, and around 40% between 1400 and 1500 bytes [17]. Since it is well-known that TCP contributes more than 90% in bytes as well as packets seen in the Internet traffic, it can be concluded from [17] that a majority of the small-sized packets are TCP ACKs. The use of XLFs for large flows reduces the number of data units as well as ACKs.

We illustrate this using numerical analysis. TCP flow sizes were generated using Pareto distribution. A heavy tail distribution for traffic was revealed some time back, and it still continues to exist [18]. Flow sizes were generated with the

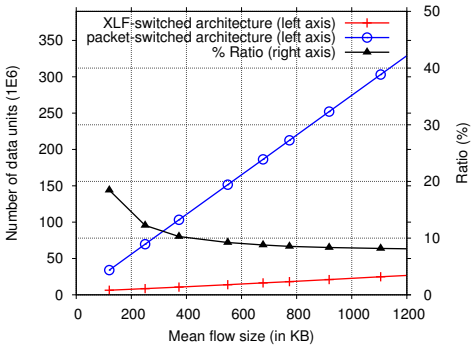
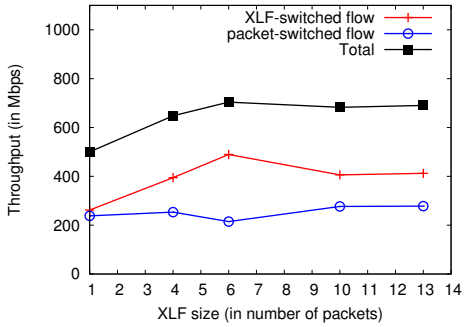


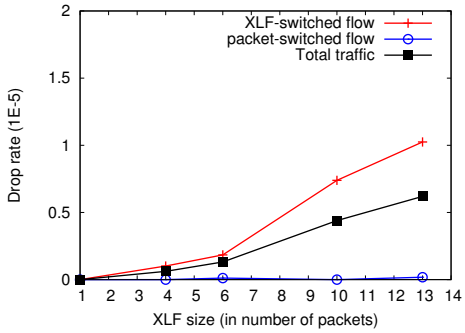
Fig. 2. Comparing counts of data units

shape parameter, $\alpha = 1.2$, and for varying mean flow sizes. Number of data units required to transfer each flow was estimated for a given flow size (this included TCP control packets too). In packet-switched architecture, all flows are switched in packets of size 1500B. In order to estimate the number of data units required to transfer flows in an XLF-switched architecture[1], we assume flows of size greater than 50kB (*elephants*) to be switched in XLFs of size 19500B ($F = 13$) and all other flows (*mice*) in 1500B packets. The volume of bytes contained in flows of size greater than 50kB is 90% (of a total of 20GB) when the mean flow size is 100kB, and more than 99% (of a total of around 200GB) when the mean flow size is 1MB. Fig. 2 compares the number of data units in XLF-switched and packet-switched architectures. The ratio of data units in XLF-switched architecture to that of packet-switched architecture is also shown in the same figure, on the right axis. For a distribution of flow sizes with a mean of 1MB, the total number of data units in XLF-switched architecture is just around 8% of that required in packet-switching architecture.

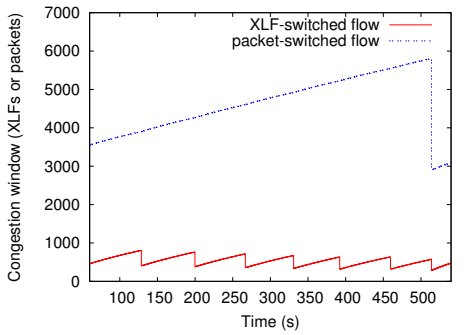
The savings becomes clear if we observe how various functions are executed in a router. TCAM (Ternary content addressable memories), for example, are powerful solutions for route lookup, and for many other classification tasks that a network node handles once per packet header. Their main drawback is the high power consumption that adds more than 10 Watts to the power budget of a line-card. A dominating part of this power is proportional to the activity, i.e. the number of search operations per second [19], [20]. With the optimistic scenario of XLFrame traffic having just around 10% of packet headers compared to the reference traffic, this translates to considerable power savings per card. Even if the peak operation speed will be unchanged; in a mix of XLFs and sequences of minimal-size packets, the latter will trigger bursts of activity interspersed with relatively long pauses: the TCAM can *cool down* during XLF forwarding. If the TCAMs are optimized for burst access [19], the mix of XLFs and packets is well adapted.



(a) Throughput: two persistent flows and their aggregate



(b) Drop rates



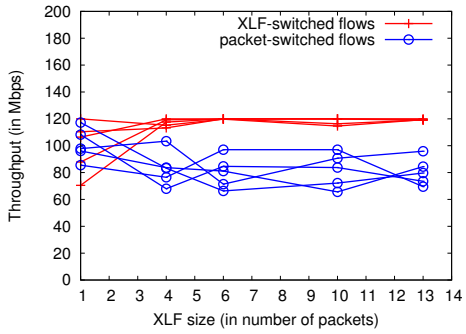
(c) Evolution of *cwnd*, $F = 13$

Fig. 3. Flows through Droptail queue, $\rho = 1.2$

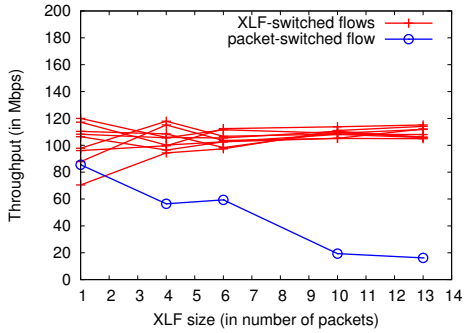
B. Throughput gains

We simulate two TCP flows in NS-2 for an unmasked processing delay (which is over and above that cut down by pipelining) set equal to transmission delay at the router connected to the sources. $C_b = 1$ Gbps, $\rho = 1.2$; i.e., $C_1 = C_2 = 600$ Mbps. Results are displayed in Fig. 3. Observe that the aggregate throughput achieved is minimum when $F = 1$ (packet size = XLF size = 1500B). Higher link utilization is achieved with XLFs as the processing delay per byte is lesser. In fact, even the packet-switched flow achieves higher throughput when the other flow is switched in XLFs. Drop rates of XLFs is higher than that of packets (ref. 3(b)). The evolutions of congestion window (*cwnd*) of both flows seen in Fig. 3(c) give insights into the higher drop

[1]We use XLF-switched architecture that has packets as well as XLFs, to distinguish it from the traditional packet-switched architecture.



(a) Five XLF flows, five packet flows



(b) Nine XLF flows, one packet flow

Fig. 4. Throughput of flows through Droptail queue, $\rho = 1.2$

rates experienced by XLFs. XLFs of size F approaching a queue with space for just $F-1$ packets get rejected, whereas packets are still accepted. Additionally, the slowing down of the XLF-switched flow makes space in the queue, resulting in the increase in *cwnd* of packet-switched flow.

VI. XLF - PACKET MULTIPLEXING

In this section, we analyse the effects of integrating XLFs in current networks, without any modification to the protocols. First we simulate a naive scenario using Droptail queues. To avoid unfairness to packet-switched flows, Deficit Round Robin (DRR) scheduling is used in the second scenario. We assume that the effect of processing delay (on throughput) is negligible, as is the case today. We consider 10 parallel flows, each being switched in either packets or XLFs. $\rho = 1.2$, $C_b = 1$ *Gbps*. The aggregate throughput achieved is always more than 999 Mbps, and hence not shown.

A. Using Droptail

Fig. 4 displays the throughputs of 10 TCP flows. In Fig. 4(a), five of them are switched in XLFs and the rest in packets, whereas nine of the 10 are XLF-switched flows in Fig. 4(b). The drop rates for individual flows and aggregate traffic for the scenario corresponding to Fig. 4(b) are as seen in Fig. 5. It is to be noted that packet-switched flow gets lesser bandwidth as the number of competing XLF flows increases, causing unfairness. Besides, XLFs experience higher drop rates.

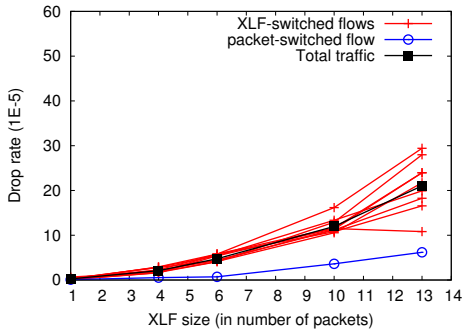


Fig. 5. Drop rates: Nine XLF flows, one packet flow

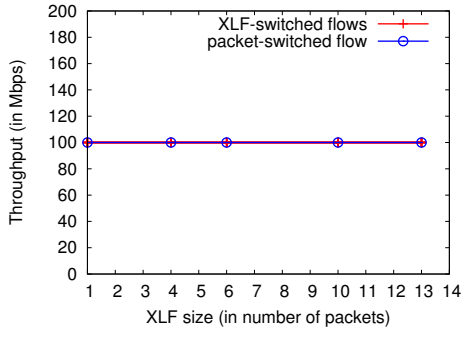
B. Using DRR

To achieve fairness (in terms of equal throughput) among competing flows, we simulate the same scenarios (as in the previous section) using DRR for scheduling packets and XLFs. The results for nine XLF-switched flows and one packet-switched flow are plotted in Fig. 6. We observe from Fig. 6(a) that each flow gets equal share of the bottleneck bandwidth. The results are similar for varying number of XLF-switched flows, though not plotted here. The drop rates of XLFs are seen to be high. Packet drop rate remains more or less a constant as expected from the standard approximation (see discussion below). With DRR forwarding, the queue has per-flow buffer limits, so that the drop rates are not correlated like in Droptail's FIFO queue.

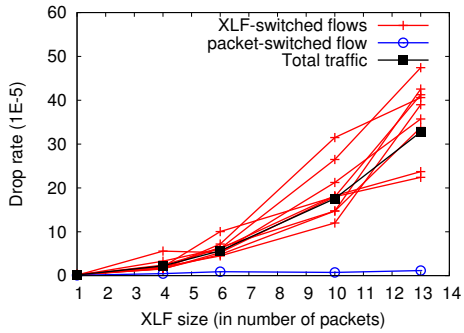
C. Discussion

Observing the ratio of XLF to packet drop rates, say β, we notice that β has higher values in simulations using DRR compared to those without DRR. In the standard approximation of throughput of TCP flows with packet size S [21], $\phi \approx \frac{C \times S}{RTT \times l^k}$, where l is the packet loss event rate, and C a constant. k is usually around 0.5. Using subscripts f and p for XLF and packet respectively, and β' to represent the ratio of XLF to packet loss event rates $(l_f : l_p)$, $\beta' = (\frac{F}{\phi_f/\phi_p})^2$. Note that the throughput formula uses loss event rate, where a loss event corresponds to one or more packet losses within a single RTT. As the loss probability increases, the probability that the losses are in a single RTT is higher for smaller packets. Hence $\beta' \geq \beta$. Nevertheless, this explains the high values for β with DRR scheduling: as $\phi_p = \phi_f$, the ratio grows with square of XLF size, a trend well observed in Fig. 6(b).

In the case of Droptail, assume that the drop probability per arriving bit is constant and so the packet loss rate is proportional to the size S of a packet. From the above throughput approximation, it follows immediately that, $\phi \propto S/\sqrt{S} = \sqrt{S}$. This indeed explains the observed unfairness of the Droptail model: large packets receive more throughput in proportion to the square root of their size. Fig. 7 shows the throughput and loss behaviour for flows as a function of packet sizes, confirming the above analysis. The simulated scenario had traffic from 50 flows (with different packet sizes) in both directions over a dumbbell topology with 25 branches. Packet

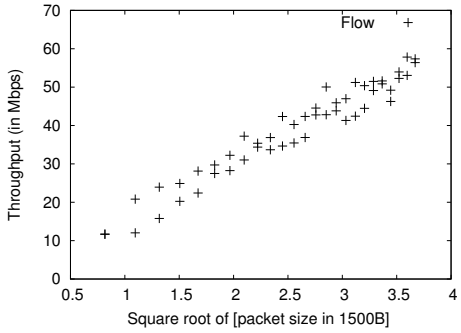


(a) Per-flow throughput (9 + 1 flows)

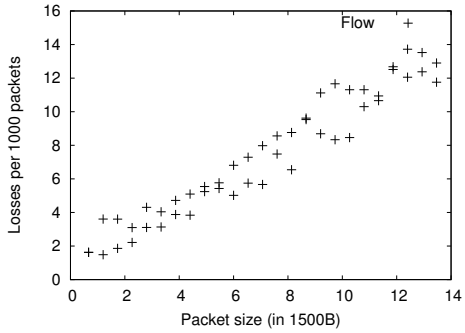


(b) Drop rates of each flow

Fig. 6. Throughput and drop rates using DRR, $\rho = 1.2$



(a) Per-flow throughput



(b) Per-flow packet loss

Fig. 7. Aggregate of 50 flows through a Droptail queue

sizes were linearly spaced from 1000B to 20000B. Each source had 1 Gbps capacity, and so did the bottleneck.

VII. Issues

Since an XLF-switched flow increases its *cwnd* in XLFs, it grows faster in the slow-start phase than a packet-switched flow. This can lead to bursty losses that need to be analysed.

Delay constraint flows might experience blocking due to the ongoing transmission of an XLF. For example, it takes 12 μs to transmit a 15000B XLF over a 10 Gbps link. A packet arriving at a node when an XLF transmission has just begun, is blocked for at least 12 μs. But, 12 μs is negligible compared to 1 ms delay required to propagate 200 km of fiber.

A. XLFs in wireless networks

Assume that access to the network is through some wireless channels that introduce random drop of packets. We show below that if TCP is used then its throughput may be very sensitive to the choice of packet size, making the access point inefficient.

Assume that the probability for an erroneous bit is p. The probability to lose a packet of size N bits, $P(N) = 1-(1-p)^N$. Throughput in packets/s is proportional to $1/\sqrt{(P(N))}$, and the goodput is thus proportional to $(1-P(N))/\sqrt{(P(N))}$. For small p,

$$P(N) = 1 - [(1-p)^{(N/p)}]^p \sim 1 - \exp(-Np) \qquad (1)$$

To obtain the best packet size N we need to maximize

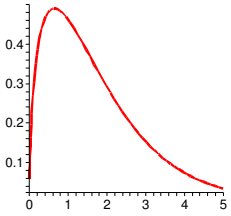


Fig. 8. Normalized goodput as a function of θ

$$Goodput(N) = \frac{N\exp(-Np)}{\sqrt{1-\exp(-Np)}}c \qquad (2)$$

where c is a constant, and $Goodput$ is in bits/s. We get

$$Goodput = \frac{\theta\exp(-\theta)}{p\sqrt{1-\exp(-\theta)}}c \qquad (3)$$

where $\theta = Np$. Fig. 8 shows the normalized goodput, where we take a unit to be c/p (in bits/s), as a function of the normalized packet length θ. The function has a unique maximum at $\theta = 0.6438$, obtained by differentiating $Goodput$ with respect to θ and equating to zero. We obtain,

Lemma 7.1: Optimal throughput is obtained for $N = 0.6438/p$ bits, where p is the bit-error rate.

Hence we propose, packets be put into XLFs after the access which suggests using Split TCP, or that lower layers include coding or ARQ so as to decrease BER.

VIII. Conclusions

Through arguments, analyses and simulations, this article reasoned the advantages of XLFs in future Internet. Private data networks and high speed LANs can be expected to be the first to support XLFs on a large scale, whereas public Internet might not introduce them completely before the access links reach Gbits/s rates. Numerical analysis showed that the number of data units at an equipment can be brought down to as low as 8% by the use of XLFs along with packets, thus bringing savings in processing resources and electrical power. When packet-header processing becomes the bottleneck, achievable throughput is close to the line rate for flows in XLFs. Increase in packet size by an order of magnitude also decreases the speed required to process packets by an order. Simple round robin mechanism like DRR is illustrated as solving the unfairness problem. This scheme is scalable as the number flows that need to be scheduled is not more than a few hundreds [22].

In this work, the effect of XLF-switched flows on short packet-switched flows has not been analysed. The introduction of XLFs increases the loss rates of not only XLFs but also packets. A short packet-switched flow that experiences a packet loss, faces a degraded response time, due to (possible) timeout and retransmissions. This performance degradation has to be studied, and possible solutions, such as priority scheduling that differentiates between short and large flows [23], have to be explored.

The high drop rates experienced by XLFs can be brought down by using ECN (Explicit Congestion Notification). Further, we plan to look at hybrid solutions to interface wireless network with XLF-switched architecture. It is also interesting to explore XLF integration in OBS networks. With the introduction of XLFs, fewer packets will be assembled into a burst, thus reducing the synchronization effects.

References

[1] "Cisco Visual Networking Index - Forecast and Methodology, 2007-2012," Cisco White Papers, Jun 2008.

[2] P. Dykstra, "Gigabit ethernet jumbo frames and why you should care," Dec. 1999, http://sd.wareonearth.com/ phil/jumbo.html.

[3] T. T. Ye, G. D. Micheli, and L. Benini, "Analysis of power consumption on switch fabrics in network routers," in *Proc. DAC '02*, 2002, pp. 524–529.

[4] "Telcos demand greener network equipment," 2008, http://www.reuters.com/article/idUSN1847837420080619.

[5] T. Bonald, S. Oueslati-Boulahia, and J. Roberts, "IP traffic and QoS control: the need for a flow-aware architecture," in *World Telecommunications Congress*, Sep. 2002.

[6] W. Feng, P. Balaji, C. Baron, L. N. Bhuyan, and D. K. Panda, "Performance Characterization of a 10-Gigabit Ethernet TOE," in *Proc. HOTI '05*, 2005, pp. 58–63.

[7] S. Makineni and R. Iyer, "Architectural Characterization of TCP/IP Packet Processing on the Pentium®M Microprocessor," in *HPCA '04: Proc. of the 10th Int'l Symposium on High Perf. Computer Architecture*, 2004, pp. 152–161.

[8] M. Mathis, "Pushing up the Internet MTU," *Internet2/NLANR Joint Techs Meeting, Miami, Florida*, Mar. 2003.

[9] http://www.intel.com/support/network/sb/cs-001911.htm.

[10] "Myri-10G: Myrinet Converges with Ethernet," www.myri.com/news/051121/Myricom_SC05_Myri-10G.pdf.

[11] http://kbserver.netgear.com/kb_web_files/n101539.asp.

[12] W. Y. H. Wang, H. N. Yeo, Y. L. Zhu, T. C. Chong, T. Y. Chai, L. Zhou, and J. Bitwas, "Design and development of Ethernet-based storage area network protocol," *Computer Communications*, vol. 29, no. 9, pp. 1271–1283, 2006.

[13] H. Simitci, C. Malakapalli, and V. Gunturu, "Evaluation of SCSI over TCP/IP and SCSI over fibre channel connections," in *Hot Interconnects 9, '01*, Aug. 2001, pp. 87–91.

[14] D. Salyers, Y. Jiang, A. Striegel, and C. P. labauer, "JumboGen: dynamic jumbo frame generation for network performance scalability," *SIGCOMM CCR*, vol. 37, no. 5, pp. 53–64, 2007.

[15] C. Qiao and M. Yoo, "Optical burst switching (OBS) - a new paradigm for an optical Internet," *J. High Speed Netw.*, vol. 8, no. 1, pp. 69–84, 1999.

[16] K. Vlachos, "Burstification effect on the TCP synchronization and congestion window mechanism," in *BROADNETS*, 2007, pp. 24–28.

[17] W. John and S. Tafvelin, "Analysis of Internet backbone traffic and header anomalies observed," in *IMC '07*, pp. 111–116.

[18] D. Collange and J.-L. Costeux, "Passive Estimation of Quality of Experience," *Journal of Universal Computer Science*, vol. 14, no. 5, pp. 625–641, Mar. 2008.

[19] W. Wu, J. Shi, L. Zuo, and B. Shi, "Power-Efficient TCAMS for Bursty Access Patterns," *IEEE Micro*, vol. 25, no. 4, pp. 64–72, 2005.

[20] B. Agrawal and T. Sherwood, "Modeling TCAM power for next generation network devices." in *ISPASS*, 2006, pp. 120–129.

[21] M. Mathis, J. Semke, and J. Mahdavi, "The macroscopic behavior of the TCP congestion avoidance algorithm," *SIGCOMM CCR*, vol. 27, no. 3, pp. 67–82, 1997.

[22] A. Kortebi, L. Muscariello, S. Oueslati, and J. Roberts, "Evaluating the number of active flows in a scheduler realizing fair statistical bandwidth sharing," *SIGMETRICS Perform. Eval. Rev.*, vol. 33, no. 1, 2005.

[23] K. Avrachenkov, U. Ayesta, P. Brown, and E. Nyberg, "Differentiation Between Short and Long TCP Flows: Predictability of the Response Time," in *Proc. IEEE INFOCOM*, 2004.