

Analysis of the effects of XLFrames in a network

Dinil Mon Divakaran¹, Eitan Altman², Georg Post³, Ludovic Noirie³, and
Pascale Vicat-Blanc Primet¹

¹ INRIA / Université de Lyon / ENS Lyon,
{Dinil.Mon.Divakaran,Pascale.Primet}@ens-lyon.fr
Postal address: LIP, ENS Lyon, Lyon - 69007, France

² INRIA, Eitan.Altman@sophia.inria.fr

³ Alcatel-Lucent Bell Labs, {Georg.Post,Ludovic.Noirie}@alcatel-lucent.fr *

Abstract. The phenomenal increase in network capacity to hundreds and thousands of Gbits/s in the core as well as Gbits/s at the access, is soon to witness stupendous amounts of packets that have to be processed and switched at amplifying line rates. Looking into the future, we address the need for the integration of packets of larger size, called XLFrames (XLFs), into the future Internet framework. This paper analyses the effects of introducing XLFs in a network that has both packets and XLFs. We evaluate the gains in terms of processing power and throughput. As we observe that XLFs have an impact on loss rate and fairness, we study how, with minimal efforts at routers while keeping the existing protocols (TCP/UDP, IP), XLFs may integrate in the current scenario.

Keywords: Future Internet, Packet size, IP network

1 Introduction

The phenomenal growth of the Internet has been accompanied by the increase in network capacity at a remarkable rate. With 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 along multiple challenges at end-hosts, switches and routers.

In this paper, we explore the concept of large packets, called *XLFrames* (XLFs), challenged by the problems facing future Internet. We perform studies to analyse the effect of introducing XLFs in a network with standard packets. Indeed, we foresee a future Internet that has classical packets as well as XLFs. Some of the reasons for keeping standard packets along with XLFs are: (i) Not all end-hosts will have the ability to send XLFs, in particular those fixed and mobile devices whose access rates stay below 100 Mbps. (ii) A TCP connection has short packets (iii) Some delay-constraint applications (say voice) might stick

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

to packets (iv) Small flows, e.g HTTP requests, are expected to use packets as there might not be sufficient data to transfer.

We discuss our motivation and related works in Section 2. In Section 3, we analyse the gains in processing and throughput. The analyses on loss rates and unfairness are studied in Section 4. Therein, we present how existing standard mechanisms can be used to tackle these issues. The effect of having XLFs in wireless networks is studied in Section 5. We conclude in Section 6.

2 Motivations and related work

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 [1]. *Therefore minimizing power consumption at equipments is one important criteria that any research on Internet architecture should focus on [2].*

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. For example, arrival of consecutive minimum-sized packets (say, 64B) at a router with 40 Gbps line cards, puts an upper bound of 12.8 ns on processing time and memory access time. 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. Storing and retrieving packets to and from memory obviously involves per-byte cost (in fact, it depends on the word size). 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 [3], 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 [4]. According to a study [5], 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. 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*.

2.1 Related work

The concept of large packet sizes has been floating around for a decade [6, 7]. *Jumbo frames* was introduced to solve some of the end-host related problems. Now, we have more reasons, mainly from the network perspective (as described above), to pursue the idea of large packet size. The limiting factor of the MTU comes from the early Ethernet designs. But today, we have Gigabit Ethernet NICs that support packet sizes of 9000B and even larger [8–10]. Research networks such as Internet2 and GEANT also support XLFs. Wang *et al.* showed that in Ethernet-based storage area networks, the use of XLFs reduce CPU utilization and interrupts notably, while improving the throughput during data transfers [11]. The use of larger MTUs also contributes to high throughput achieved on iSCSI storage networks [12].

Researchers have also put forward the idea of aggregating packets in the network, at the edges. In [13], 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 [14]. The burst assembly process is known to have a significant impact on TCP performance [15]. Short assembly times hinder the congestion window growing pace, and long assembly time causes unacceptable delays. In addition, the burstification might also lead to synchronization of TCP flows, resulting in inefficient bandwidth utilization.

3 Benefits of introducing XLFs in a packet network

In this section, we analyse the benefits of XLF introduction and also provide an insight into the main drawback. We focus on the following questions: (i) What is the gain in terms of processing power? (ii) What is the gain in terms of throughput? We study these two questions first through simulations.

3.1 Simulation setup

We simulate the multiplexing of TCP flows with small and large packets using NS-2; for all the simulations, we consider a dumbbell topology, connecting n *src-dst* pairs (see Fig. 1). Packet size is 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,

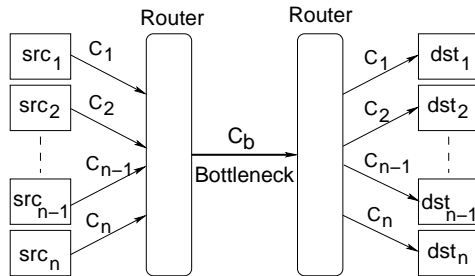


Fig. 1. Topology considered for simulations

and from the bottleneck to *dst* nodes, are set depending on the level of congestion desired for the scenario. The congestion factor $\rho = \sum_{i=1}^n C_i/C_b$, C_i being the capacity of link i .

The size of the bottleneck queue is set in bytes, as the bandwidth delay product (*BDP*) for 100 ms. The buffer size at 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, we set $C_b = 1$ 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 are throughput and drop rate of individual and aggregate flows, all measurements being made at the bottleneck link. 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.

3.2 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 [16]. Since it is also well-known that TCP contributes more than 90% in bytes as well as packets seen in the Internet traffic, it can be concluded that a majority of the small-sized packets are TCP ACKs (around 40% of IP packets). 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 (with $\alpha = 1.2$) 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*, we assume flows of size greater than 50kB to be switched in packets of size 19500B ($F = 13$) and

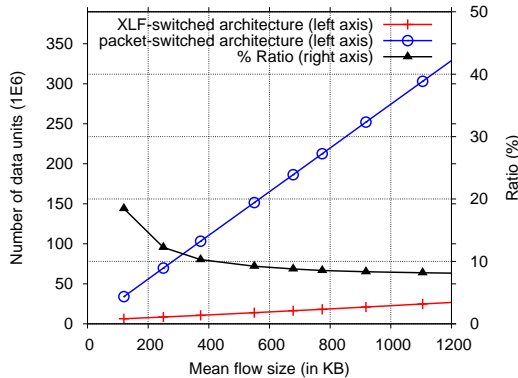


Fig. 2. Comparison of packet counts

the rest 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, but 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 reduction in the number of packets also reduces the number of arbitration decisions required to switch packets from the inputs to the outputs. Though this is not possible in traditional switches using cell-based switching (where variable length packets are segmented into fixed size cells), it should be noted that with the feasibility of adding buffers in a crossbar switch, recent research works have explored asynchronous buffered crossbar architecture, proposing it as the next level of crossbar switches that can scale to hundreds of ports [17,18]. As the units being switched in such architectures are variable-size packets, the gain due to XLFs is directly proportional to the reduction in the number of packets.

3.3 Throughput gains

To estimate the gain in throughput when a flow is switched in XLFs, consider packet processing time in high-capacity switches. For example, processing 64B packets at 100 Gbps represents about 200M packets/s, whereas with 1500B packet size, the required speed decreases to 8.3M packets/s, and with even larger packets of, say 19500B, the required speed is only 640k packets/s. Of course, real traffic is a mix of different packet sizes, but the mean time between packet operations will increase with average size. With appropriate implementation of electronics, significant saving of electrical power can be achieved in the packet-header processing parts of an equipment.

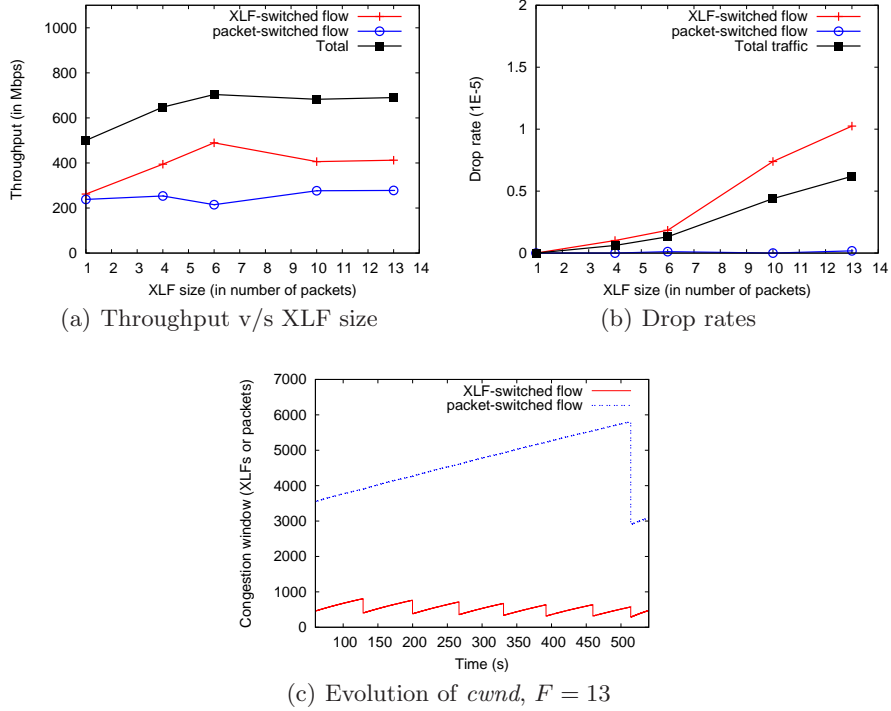


Fig. 3. Flows through Droptail queue, $\rho = 1.2$

We multiplex two TCP flows, one in packets and the other in XLFs, using NS-2 for a processing delay equal to the transmission delay. $C_b = 1$ Gbps, and $\rho = 1.2$; that is $C_1 = C_2 = 600$ Mbps. The 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 larger XLFs. In fact, even the packet-switched flow achieves higher throughput when the other flow is switched in XLFs. The drop rates are as seen in Fig 3(b). As expected, drop rates of XLFs is higher than that of packets. The evolutions of TCP congestion window ($cwnd$) for both flows are plotted in Fig. 3(c). This gives insights into the higher drop rates experienced by XLFs. XLFs of size F arriving at 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 more space in the queue, resulting in the increase in $cwnd$ of packet-switched flow.

4 Adapting the protocols and mechanisms to XLFs

In the previous section, simulation results highlighted the impact of packet size on the loss rate and the resulting unfairness. In this section, we explore how,

without changing the existing protocols (TCP/UDP, IP), XLFs integrate in the current scenario.

4.1 XLF and loss probability

To understand the loss rate issue, we first analyse the losses at a queue that has arrivals in packets as well as XLFs using a Markov model. λ_p is the packet arrival rate, and λ_f is the XLF arrival rate; both arrival processes are assumed to be Poisson in nature. $\lambda = \lambda_p + \lambda_f$. The service time of packets is exponentially distributed with mean rate μ . The Markov model is as given in Fig. 4.

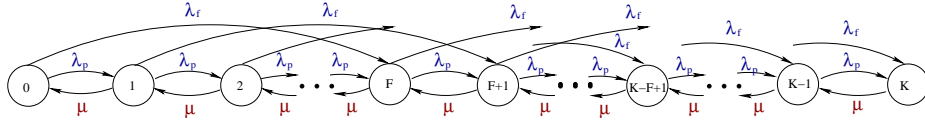


Fig. 4. Markovian queue with packet and XLF arrivals

The queue is measured in packets. An XLF arrival brings F packets. Hence an arrival of an XLF when the state is i , results in a transition to state $i + F$, provided $i + F < K$, where K is the queue size in packets. Similarly, it takes F stages of services to completely serve an XLF. If π_i is the probability for having i packets in the queue, the steady-state equations are given by,

$$\begin{aligned}
 \pi_0 \lambda &= \pi_1 \mu \\
 \pi_i (\lambda + \mu) &= \pi_{i-1} \lambda_p + \pi_{i+1} \mu \quad 1 \leq i \leq F-1 \\
 \pi_i (\lambda + \mu) &= \pi_{i-F} \lambda_f + \pi_{i-1} \lambda_p + \pi_{i+1} \mu \quad F \leq i \leq K-F \\
 \pi_i (\lambda_p + \mu) &= \pi_{i-F} \lambda_f + \pi_{i-1} \lambda_p + \pi_{i+1} \mu \quad K-F+1 \leq i \leq K-1 \\
 \pi_K \mu &= \pi_{K-1} \lambda_p + \pi_{K-F} \lambda_f
 \end{aligned} \tag{1}$$

No more XLFs can be accepted if the queue is any state k such that $k \geq K - F + 1$. Packet loss probability is π_K . Fig. 5 shows the loss probabilities for packets and XLFs for increasing XLF size. The load brought by the packets was set equal to that brought by XLFs, and they were kept constant so as to have a constant load for varying XLF size. The buffer size was set to 200 packets. The losses are seen to increase; and more importantly, the losses experienced by XLFs are increasing by orders of magnitude.

4.2 XLFs and unfairness

As the introduction of XLFs in packet networks increases the losses of both packets and XLFs, we analyse the effects on individual and aggregated throughput when classical queuing and scheduling mechanisms are used. We simulate the basic scenario using Droptail queues and FCFS discipline. We assume that

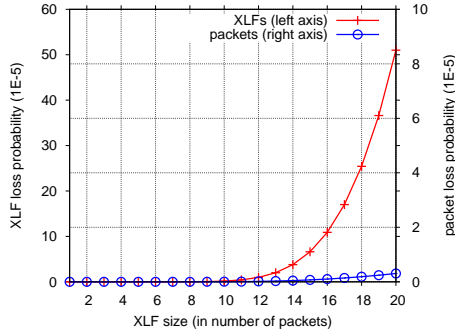


Fig. 5. Loss probabilities as a function of XLF size

the effect of processing delay (on throughput) is negligible due to pipelining, as is the case today. We consider 10 parallel flows, each being switched either in packets or in XLFs. $\rho = 1.2$, $C_b = 1 \text{ Gbps}$. The aggregate throughput achieved is always more than 999 Mbps, and hence not shown.

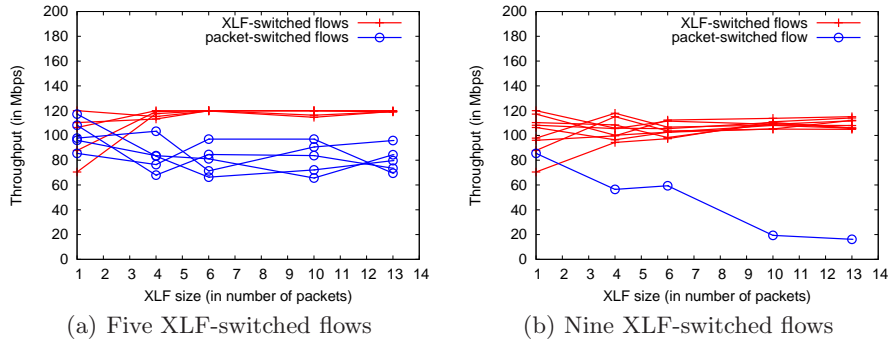


Fig. 6. Throughput of flows through Droptail queue, $\rho = 1.2$

Fig. 6 displays the throughputs of 10 TCP flows. In Fig. 6(a), five of them are switched in XLFs and the rest in packets, whereas nine of the 10 are XLF-switched flows in Fig. 6(b). The drop rates for individual flows and aggregate traffic for the scenario corresponding to Fig. 6(b) are as seen in Fig. 7. It is to be noted that packet-switched flow gets lesser bandwidth as the number of XLF-switched flow increases, causing unfairness. Besides, XLFs experience higher drop rates.

4.3 Using Deficit Round Robin (DRR) for achieving fairness

To achieve fairness (in terms of throughput) among competing flows of very different packet sizes, we propose to use DRR (instead of FCFS) for scheduling

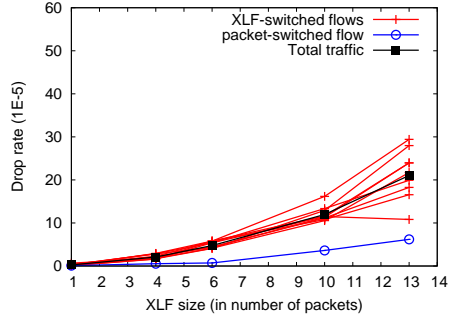


Fig. 7. Drop rates. Nine XLF-switched flows

the data units. 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. 8. We observe from Fig. 8(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).

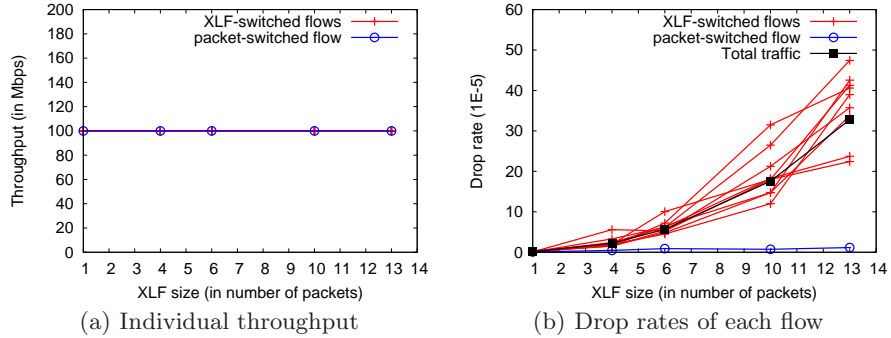


Fig. 8. Throughput and drop rates using DRR, $\rho = 1.2$

Observing the ratio of XLF drop rate to packet drop rate, say β , we notice that β has higher values in simulations using DRR compared to those without DRR. In the standard approximation of throughput [19],

$$\phi \approx \frac{C \times S}{RTT \times l^k} \quad (2)$$

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 (drop) probability increases, the probability that the losses are in a single RTT is higher for small 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 the square of XLF size, a trend well observed in Fig. 8(b).

For the 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, $\phi \propto S/\text{sqrt}(S) = \text{sqrt}(S)$, explaining the observed unfairness of the Droptail model: flows receive throughput in proportion to the square root of their packet sizes. A simulation with 50 flows was performed. The dumbbell topology had 25 branches. There were flows in both directions, and packet sizes were linearly spaced from 1000B to 20000B. Each source had 1 Gbps capacity, and so did the bottleneck. Fig. 9 shows the throughput and loss behaviour for flows with different packet sizes.

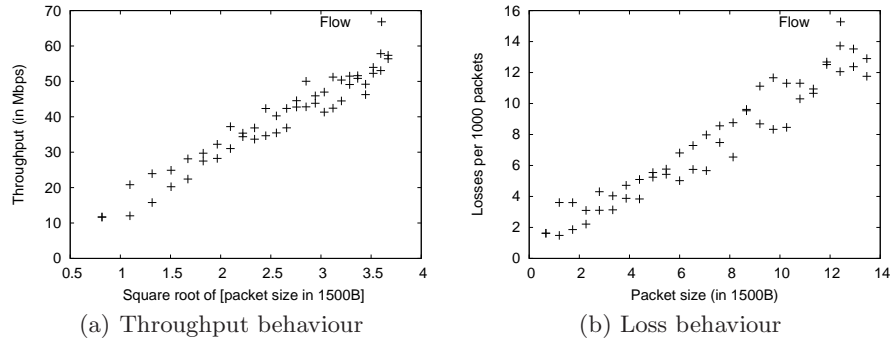


Fig. 9. Analysis of 50 flows through a Droptail queue

4.4 Loss reduction using ECN (Explicit Congestion Notification)

This section motivates the use of ECN, an IETF standard option for packet marking in routers, to reduce the high drop rates experienced by XLFs. At ECN-enabled routers, instead of being dropped, packets are marked before the queue starts to overflow. This makes it necessary to estimate the queue length, and accordingly mark the packets with some probability. At the same time, we also use DRR scheduling to ensure fairness among flows with varying packet lengths. To simulate this setup, we incorporate code in NS-2 to probabilistically mark the packets, depending on the length of the common queue⁴. Fig. 10 compares the

⁴ Even though NS-2 has RED implementation, its design rules out the use of RED and DRR scheduling at the same time.

drop rates experienced by flows switched in XLFs and packets, when nine out of ten flows were XLF-switched flows. Evidently, the drop rates have gone down very low. Obviously, due to the DRR scheduling there was also no unfairness among flows.

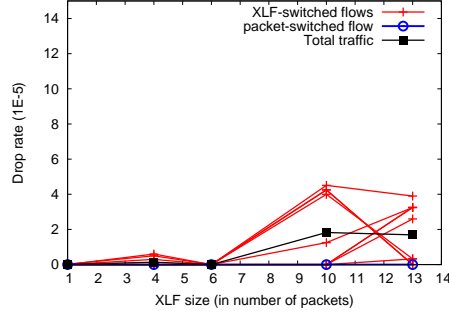


Fig. 10. Drop rates using ECN with DRR. Nine XLF-switched flows

5 XLFs in wireless networks

Nowadays, access to the network is often 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 is

$$P_l(N) = 1 - (1 - p)^N \quad (3)$$

Throughput in packets/s is proportional to $1/\sqrt{P_l(N)}$, and the goodput is thus proportional to $(1 - P_l(N))/\sqrt{P_l(N)}$; For small p ,

$$P_l(N) = 1 - [(1 - p)^{(N/p)}]^p \sim 1 - \exp(-Np) \quad (4)$$

To obtain the best packet size N we need to maximize

$$Goodput(N) = \frac{N \exp(-Np)}{\sqrt{1 - \exp(-Np)}} c \quad (5)$$

where c is a constant. We write

$$Goodput = \frac{\theta \exp(-\theta)}{p \sqrt{1 - \exp(-\theta)}} c \quad (6)$$

where $\theta = Np$. Fig. 11 shows the goodput in units of length per second, where we take a unit to be c/p , as a function of the normalized packet length θ . The function has a unique maximum at $\theta = 0.6438$ obtained by differentiating the Goodput with respect to θ and equating to zero. We obtain,

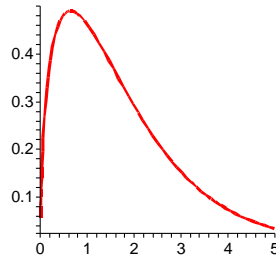


Fig. 11. Normalized goodput as a function of θ

Lemma 51 *The optimal throughput is obtained for the packet size $N = 0.6438/p$, 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 the lower layers should include coding or ARQ so as to decrease the bit error rate.

5.1 Using block FEC

Block FEC can be used to reduce the loss rates in wireless networks. We use the Gilbert loss model and the FEC model described in [20]. A link-level packet is sent as multiple smaller units called transmission units (TUs in short). Let p be the probability of losing a TU, and let N be the packet size in TUs. $N = K + R$, where R is the number of redundant TUs and K is the number of *useful* TUs. In block FEC, such a packet of N TUs is lost, only if more than R TUs are lost. The probability of losing more than R TUs is,

$$P_{(R,N)} = 1 - \sum_{i=0}^R \binom{N}{i} p^i (1-p)^{N-i}$$

For the throughput of a TCP flow, we use the formula, $\phi = \frac{S}{RTT} \sqrt{\frac{3}{2P}}$, where P is the probability of losing a packet. The throughput under the FEC scheme with parameter R is,

$$\phi_{(R,N)} = \min \left(\frac{S}{RTT} \sqrt{\frac{3}{2P_{(R,N)}}}, \frac{K}{N} C \right)$$

where C is the capacity of the (bottlenecked) wireless link. Fig. 12 shows the throughput as a function of the ratio of K/N . The transmission unit is assumed to be 100 bytes. N is then set to 15 (TUs) for packets, and to 200 for XLFs. The capacity C was set to 1 Gbps. We see that, it is possible to achieve higher throughput with considerably larger K/N ratio for XLFs.

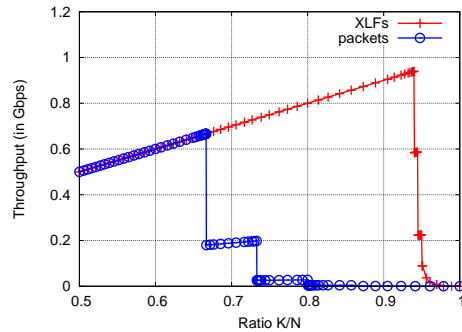


Fig. 12. Throughput as a function of K/N

6 Conclusions

Through arguments, analyses and simulations, this article reasoned the need for having XLFs in future Internet. 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. In the case where processing is the bottleneck, achievable throughput is close to the line rate when XLFs are used. Increase in packet size by an order also decreases the speed required to process packets by an order. Though XLFs introduce unfairness without any fair queueing algorithm, simple round robin mechanisms like DRR has been illustrated as solving this issue. The high drop rates experienced by XLFs is brought down by using ECN feature. Finally, we analyse the redundancy that has to be added in the wireless access network integrating XLFs. These analyses have shown that XLFs will bring a lot of benefits and can be integrated with light adaptation of current mechanisms and no changes to the standard protocols.

References

1. Ye, T.T., Micheli, G.D., Benini, L.: Analysis of power consumption on switch fabrics in network routers. In: DAC '02: Proc. of the 39th Conf. on Design automation. (2002) 524–529
2. : Telcos demand greener network equipment. <http://www.reuters.com/article/idUSN1847837420080619> (2008)
3. Bonald, T., Oueslati-Boulaiah, S., Roberts, J.: IP traffic and QoS control: the need for a flow-aware architecture. In: World Telecommunications Congress. (September 2002)
4. Feng, W., Balaji, P., Baron, C., Bhuyan, L.N., Panda, D.K.: Performance Characterization of a 10-Gigabit Ethernet TOE. In: HOTI '05: Proc. of the 13th Symposium on High Perf. Interconnects. (2005) 58–63
5. Makineni, S., Iyer, R.: Architectural Characterization of TCP/IP Packet Processing on the Pentium®M Microprocessor. In: HPCA '04. (2004) 152–161

6. Dykstra, P.: Gigabit ethernet jumbo frames and why you should care. <http://sd.wareonearth.com/phil/jumbo.html> (December 1999)
7. Mathis, M.: Pushing up the Internet MTU. Internet2/NLANR Joint Techs Meeting, Miami, Florida (March 2003)
8. <http://www.intel.com/support/network/sb/cs-001911.htm>
9. : Myri-10G: Myrinet Converges with Ethernet. www.myri.com/news/051121/Myricom_SC05_Myri-10G.pdf
10. http://kbserver.netgear.com/kb_web_files/n101539.asp
11. Wang, W.Y.H., Yeo, H.N., Zhu, Y.L., Chong, T.C., Chai, T.Y., Zhou, L., Bitwas, J.: Design and development of Ethernet-based storage area network protocol. *Computer Communications* **29**(9) (2006) 1271–1283
12. Simitci, H., Malakapalli, C., Gunturu, V.: Evaluation of SCSI over TCP/IP and SCSI over fibre channel connections. In: *Hot Interconnects 9, '01*. (August 2001) 87–91
13. Salyers, D., Jiang, Y., Striegel, A., Poellabauer, C.: JumboGen: dynamic jumbo frame generation for network performance scalability. *SIGCOMM CCR* **37**(5) (2007) 53–64
14. Qiao, C., Yoo, M.: Optical burst switching (OBS) - a new paradigm for an optical Internet. *J. High Speed Netw.* **8**(1) (1999) 69–84
15. Vlachos, K.: Burstification effect on the TCP synchronization and congestion window mechanism. In: *BROADNETS*. (2007) 24–28
16. John, W., Tafvelin, S.: Analysis of Internet backbone traffic and header anomalies observed. In: *IMC '07*. (2007) 111–116
17. Katevenis, M., Passas, G., Simos, D., Papaefstathiou, I., Chrysos, N.: Variable packet size buffered crossbar (CICQ) switches. In: *ICC '04*. Volume 2. (Jun 2004) 1090–1096
18. Chrysos, N., Katevenis, M.: Scheduling in non-blocking buffered three-stage switching fabrics. In: *IEEE INFOCOM*. (April 2006)
19. Mathis, M., Semke, J., Mahdavi, J.: The macroscopic behavior of the TCP congestion avoidance algorithm. *SIGCOMM CCR* **27**(3) (1997) 67–82
20. Barakat, C., Altman, E.: Bandwidth tradeoff between tcp and link-level fec. *Comput. Netw.* **39**(2) (2002) 133–150