Understanding Flow Performance in the Wild

Muhammad Amir Mehmood[1], Nadi Sarrar[2], Steve Uhlig[3] and Anja Feldmann[2]
[1]Al-Khawarizmi Institute of Computer Science, UET, Lahore, Pakistan
[2]Technische Universität Berlin, T-Labs, Berlin, Germany
[3]Queen Marry University of London, London, UK
{amir, nadi, anja}@net.t-labs.tu-berlin.de, steve@eecs.qmul.ac.uk

***Abstract*—Recent Internet studies have reported on continued traffic growth and popularity of web-based applications. Any adverse impact that these observed trends may have on Internet traffic flows can result in sub par performance, which in turn results in unsatisfactory user experience.**

Leveraging data collected at a major content distribution network (CDN), we investigate flow-level performance in the wild. We observe that packet losses differ widely across flows of different sizes, and even for flows of similar size. To shed light on these observations, we rely on a controlled testbed setup with advanced instrumentation via NetFPGA cards. We highlight the key factors which can degrade flow-performance across different network loads and flow-size distributions.

We find that packet losses do not affect all flows similarly. Depending on the network load, some flows either suffer from significantly more drops (unhappy flows) or significantly less drops than the average loss rate (happy flows). Very few flows actually observe a loss rate similar to the average loss rate. Therefore, any single flow is very unlikely to observe the global packet loss process. Furthermore, we find that some flows are burstier than others as indicated by their average congestion window.

I. Introduction

The ubiquity of high speed Internet access, the proliferation of smart phones and the popularity of content-rich applications have enabled users to perform browsing, to stream videos, to play online games, and to share content through social networking platforms, anytime and anywhere. High speed Internet access has also changed user expectations. Customer studies from Google, Amazon, Yahoo, and Microsoft have demonstrated that few milliseconds difference in web performance impact business value [1]. As a result of changing trends in the Internet, performance studies are crucial for service providers, network operators and application designers.

Recent Internet traffic studies have revealed the growing popularity of applications based on the HTTP protocol [2], [3]. A large portion of this HTTP-based traffic is served by Content Distribution Networks (CDNs) whose servers are deployed around the world. The end users requesting this content are connected through heterogeneous access technologies that have widely different properties in terms of bandwidth, loss, and latency. As a consequence, understanding today's application performance is challenging.

In the past, performance has been mainly characterized by the overall packet loss rate, delay, and jitter measured at a vantage point. In practice, Internet traffic consists of a majority of short flows, e.g., news feeds, Facebook status updates, Twitter feeds, etc., along with small number of large flows, e.g., movie downloads that result in congestion. Despite many efforts, the dynamics of congestion is not well understood, and as a result its impact on small and large transfers of different applications.

Motivated by these challenges, in this paper, we look at application performance by focusing on individual flows. Our objective is to ascertain the impact of congestion on flows of different sizes. With the popularity of HTTP-based applications that rely on the TCP transport layer protocol, packet losses are inevitable. Standard variants of TCP use bandwidth probing to determine the available bandwidth on the network path. TCP increases its throughput until it receives a signal, a lost packet, that the network can no longer support the traffic load. Therefore, in this paper, we concentrate on *packet losses* as they are a key signal to better understand application performance.

Leveraging a large data set collected at a major content distribution network (CDN), we investigate the performance of flows in the Internet. Our work differs from prior research in that we consider performance metrics based on different classes of flows according to their size, hereafter referred to as *flow-classes*. We use TCP retransmissions as a key performance metric as any retransmission in a flow indicates either packet loss or high latency triggering a timeout event. Our aim is to highlight that flows can experience severe degradations that would be considered unacceptable by today's end-users.

Furthermore, to understand the key factors which lead to such flow-level performance degradations, we rely on tightly controlled experiments employing specific congestion levels and flow size distributions. Using advanced instrumentation across network layers, we track the loss process and its impact on each individual flow. In addition, we record TCP congestion window statistics for every flow. These statistics illuminate key factors that are responsible for performance degradation of flows in the wild.

Our main insight relates to **flow happiness**. We find that packet losses are not evenly distributed among flows of different sizes. Large flows are positively discriminated. We call such flows *happy flows*. On the other hand, small flows are negatively discriminated. We call them *unhappy flows*. The losses observed by individual flows differ across flow sizes as well as within flow sizes. Moreover, any single flow is very unlikely to observe the overall packet loss process, i.e., average

TABLE I
SUMMARY OF PROPERTIES OF ANONYMIZED TRACE.

Name	Type	Time	Size	Duration
CDN	conn. logs (sampled)	Mar 2010	50GB	2 weeks

packet losses on a link are a misleading metric to understand flow performance.

The remainder of this paper is structured as follows. In Section II, we present our data set that we use. We explain our experimental methodology in Section III. In Section IV, we present our results on per flow basis. We describe related work in Section V and summarize our work in Section VI.

II. DATASET

In this section, we describe general properties of the dataset used. We also provide an overview of the data collection methodology. Table I summarises the characteristics of our dataset. Our data set consists of two weeks of connection level logs from the servers of a major content distribution network (CDN). We specifically select the CDN servers which are serving customers, both DSL and mobile, of a large European ISP.

The logs are obtained from kernel level monitoring on the CDN servers. They include low level statistics such as total packets, bytes, retransmitted packets and bytes, RTTs, and durations for each TCP connection. Due to the sheer data volume, these logs are only generated for sampled connections. Statistics for all flows are maintained in the kernel, and once the sampling mechanism is triggered, statistics for that flow are recorded to disk. As most of the CDN traffic are downloads from end-users, this data provides statistics about the connection for a single side, i.e., it captures only the HTTP traffic in the direction from the CDN servers to the customer.

When sampling is triggered, the statistics of the connection until that time are recorded and captured on disk. The sampling process is flow based, and therefore will provide a representative view of all flows. Packet sampling on the other hand is biased towards large flow sizes, missing short flows. In total, we select 57M connections requested from various different access technologies for our analysis.

III. METHODOLOGY

In this section, we first describe our methodology to group flows into *flow-classes* based on their size. Next, we present our experimental setup and discuss its design.

A. Flow classes

Internet flow sizes are consistent with heavy-tailed distributions [4]. Therefore, we use logarithmic classes, referred to as *flow-classes*, for binning flows based on their payload bytes. We define flow-class i as all flows such that $2^i <$ payload bytes $\leq 2^{i+1}$, for $i = 0, 1, 2..n$. The largest flow-class also contains flows larger than 2^n. By analyzing flows separately for each size-based class, we can compare flow performance across flow-classes. We typically start with a flow-class of 1KB and go up to a flow-class of 1GB.

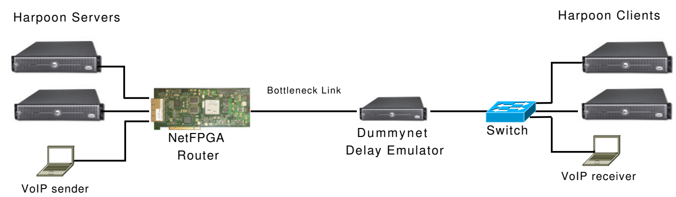


Fig. 1. Experimental Setup

As different flow-classes may be dominated by different types of flows, we believe that it is necessary to study flow performance across flow-classes. For example, some types of video objects have median sizes of 265KB, 802KB, and 1743KB [5]. Likewise, most of the Google products have flows in the range of 4-16KB [6]. Most recent studies about the changing nature of website complexity [7] have shown that overall median web-page sizes for short, medium, and long web pages has grown to 40KB, 122KB, and 286KB respectively. Moreover, when a user observes good application performance, he may be tempted to access larger objects and thus generate larger flows. If the performance is impaired on the other hand, e.g., due to retransmissions, he may restrict himself to smaller objects, e.g., a lesser quality image, which corresponds to smaller flows.

B. Experimental Setup

To study the key factors that degrade performance of the flows in the wild, we rely on a configurable and flexible testbed [8] that allows tightly controlled experiments. We now outline the salient features of our experimental setup, as shown in Figure 1.

Realistic Traffic Generation: To generate Internet like traffic, we rely on multiple PCs. We select Harpoon [9] for its ability to reproduce flow-level behavior consistent with Internet traffic. The two main parameters used for customizing Harpoon are the flow-size distribution and the flow inter-arrival time distribution. Most flows in the Internet rely on closed-loop feedback [10]. Therefore, we use TCP flows for most of the traffic. We also add some UDP flows using a VoIP client PJSIP. No other traffic was present on the network during the experiments.

Harpoon is configured to choose file sizes according to Pareto distributions with $\alpha = \{1.2, 1.5, 2.0\}$ and a mean of $\mu = 110$KB. These choices for the Pareto distribution ensure a finite mean while ensuring that the generated traffic exhibits variability and scaling behavior. To limit our parameter space, we choose an exponential distribution with mean $\mu = 1$ second for the inter-connection times, i.e., the user waiting times between different web requests.

Topology Emulation: The network topology we use is the classical *dumbbell* as shown in Figure 1. All network interfaces are 1 Gigabit Ethernet cards. The configurable network bottleneck is located between the NetFPGA router and the Dummynet delay emulator. Harpoon clients sent Web requests to the Harpoon servers. Using Dummynet [11] we add a delay

TABLE II
TRAFFIC GENERATION PARAMETERS.

Load	Low	High	Very high
No. of Harpoon sessions	80	200	360
Offered load (%)	50	96	170
Average no. of concurrent TCP flows	140	1250	1700

of 150ms to every ACK packet from the Harpoon clients to the Harpoon servers. This additional delay along with the queuing delay due to cross traffic enables us to emulate round-trip-times as they occur in WAN environments [2]. We explicitly chose to focus on relatively large RTTs to better observe the impact of the congestion and the delay imposed by TCP's feedback mechanism. With our bottleneck capacity of 242Mbps and mean round-trip time around 150ms, we chose 128 and 256 packets buffer.

Monitoring: One of the challenges in the testbed environment is to monitor buffer statistics at the router buffers. Since commercial routers do not provide fine time scale statistics about their buffer occupancy, we opt for the NetFPGA [12] as a router. It allows to gather highly accurate buffer statistics. Moreover, we monitor the internal behavior of the TCP stack at Harpoon servers using the *tcphook* [13] Linux kernel module. This approach allows us to correlate TCP congestion dynamics with different congestion levels across flow-classes.

Network Bottleneck: To ensure that the only bottleneck in our setup is the router buffer of the NetFPGA card, see Figure 1, we increase the maximum TCP receive window size to 20MB. This ensures that the transfers are not TCP receiver-window limited. All experiments use TCP New Reno to control the size of the TCP congestion window.

Data capture: We capture packet level traces at both the ingress and egress ports of the NetFPGA router. By comparing both traces, we are able to pinpoint missing packets along with transport layer information, e.g., TCP sequence numbers, as well as timing information about when the drop occurred. In addition, we can observe all generated flows from the ingress port trace. Thus we can study the per-flow loss process. We run each experiment for 30 minutes. This duration allows each individual experiment to stabilize. The resulting traces, despite their size, can be analyzed within a reasonable time.

Load: To create different network conditions we rely on three different offered load levels by changing the number of parallel Harpoon sessions on our clients. Note, increasing the offered load can lead to different link utilizations. We distinguish three load levels: *low*, *high*, and *very high*. To determine the necessary number of Harpoon sessions, we run the experiments without link capacity limitations. The lowest load, called *low load*, corresponds to a mean link utilization around 50% which should not impose too much congestion. However, once the load exceeds 50% one can expect degradations in the quality of service, e.g., increased delay and packet loss. Therefore we choose the *high load* scenario in such a way that the resulting utilization will be close to the link capacity. In the *very high load* scenario we intentionally overload the bottleneck link by letting the Harpoon servers generate about 1.7 times the capacity of the bottleneck link. The resulting number of Harpoon sessions and the average number of concurrent TCP flows are shown in Table II. A Harpoon session is equivalent to flows generated by an Internet user.

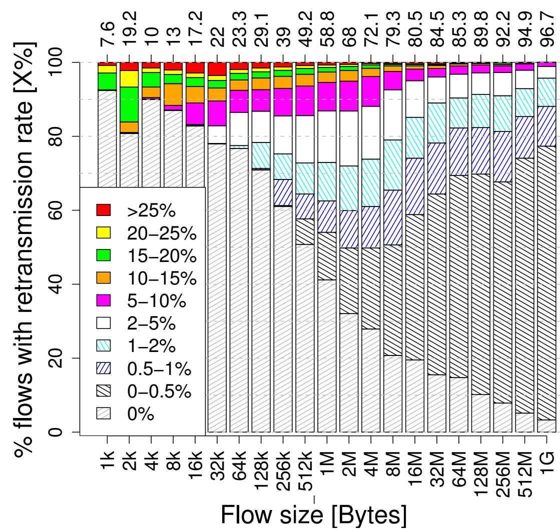


Fig. 2. CDN flows: Retransmission rate vs. flow size

IV. RESULTS

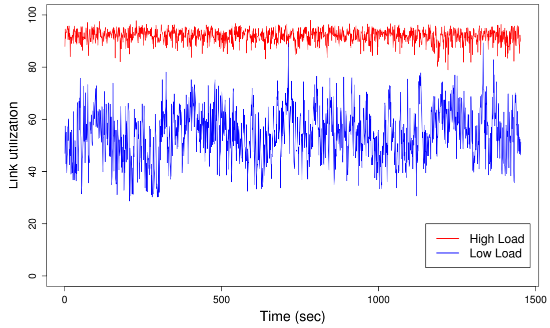
In this section, we first present flow performance as observed from the CDN dataset. Next, we explore the potential causes of our observations in the data set with the help of testbed experimentation.

A. *Happy flows: Myth or reality*

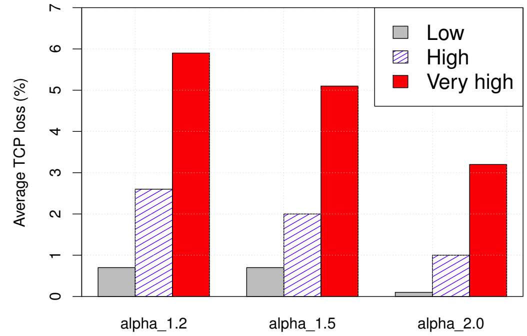
We start by analyzing CDN logs as described in Section II. The average packet retransmission rate across the dataset is 1.5%. This rate is comparable to previous studies [14], [6]. In total, 16.9% of connections were found with retransmission packets. A large number of connections therefore do not see any packet retransmission. This implies that some of the connections see more retransmissions. Indeed, 3.5% of the connections have a retransmission rate higher than 20%.

Traditionally, the performance received by bulk flows is considered as the overall performance. While bulk flows contain the majority of the bytes, most of the flows are short [4]. In comparison to the attention that bulk flows have received, short flows have received almost none. Yet, the performance short flows receive can be crucial for the experience of the user.

To explore the impact of retransmission packets on individual flows, we use logarithmic binning according to the flow size (see Section III-A). For each bin, we compute, for all flows within the bin, the percentage of retransmitted packets. We then use another binning to show what percentage of flows within a given size bin, have a percentage of retransmitted packets that falls within the bin range. This data is then plotted as a stacked barplot with a separate bar per flow size class. Within this bar, we show the fraction of flows with retransmission rate larger than 25% at the top and the fraction of flows with no retransmission packets at the bottom, i.e,



(a) Load variability



(b) TCP packet loss (256 packet buffer)

Fig. 3. Impact of load on traffic variability and packet loss

happy flows. Thus, the y-axis shows, for each flow size bin, the cumulative percentage of flows with retransmission rate of at least y. In addition, the numbers on top of the bins indicate the overall percentage of flows within the bin that have retransmitted packets.

Figure 2 shows the stacked plot for the whole CDN data. The stacked plot illuminates the differences of flow performance across different flow sizes. In particular, we found that 7.6-29.1% of flows smaller than 128KB experience retransmissions. Such flows most likely represent interactive web browsing. Any loss that occurs in such flows is unlikely to be recovered by the TCP fast retransmission mechanism, because of the limited number of packets in flight. Timeouts are therefore necessary to recover for losses, slowing down the data transfer. Another unexpected observation in Figure 2 relates to the large flows. Surprisingly, despite their duration, large flows (up to 1GB) can survive without suffering any retransmissions. Finally, middle-sized flows, in the range of 512KB-8MB, have a higher percentage of flows with higher retransmission rates.

The observations from the CDN flows display a wide diversity in flow performance across different flow sizes. However, these observations do not reveal the underlying causes. To investigate potential factors that may explain flow performance, we next proceed to our experimental evaluation of flow performance with in controlled testbed setup.

B. Experimental results

Next, we describe the results of our testbed experimentation. We start by exploring the effects of traffic variability and burstiness on the overall link utilization and packet loss. We continue with the analysis of flow-level packet loss and finally we present congestion window dynamics for different flow-classes.

1) Impact of load and burstiness: To better understand the properties of generated traffic in our testbed as explained in Section III-B, we first concentrate on the overall statistics such as the link utilization and traffic variability for different loads. Although we control the average link utilization, on its own it does not tell the whole story.

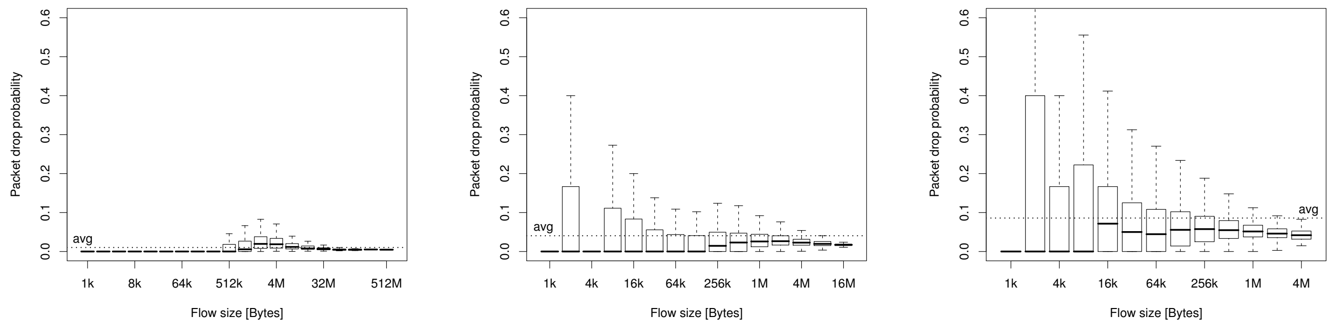
Figure 3(a) shows the link utilization across time for 1s time bins for two experiments: one with low offered load and one with high offered load. In contrast to the high load scenario, the low load offers more variations in the link utilization. Note that link utilization is a consequence of traffic contribution by all flows that share the bottleneck link and competing for available resources.

Next, we examine the impact of burstiness on the overall packet loss. In principle, traffic burstiness is not a problem if buffers can accommodate the bursts. However, when traffic burstiness leads to packet losses, it may impede flow performance. However, losses are inevitable with TCP because of the way it estimates the available path capacity: by generating losses and backing off once it detects a loss. We thus study the average packet loss under different offered loads and flow size distributions.

One of the contributors to Internet traffic variability is the heavy-tailed nature of flow size distributions [4]. This variability can be characterized by the shape parameter (α) of a Pareto distribution. We expect to see an impact of the degree of the heavy-tailedness of flow size distributions on the loss process.

Figure 3(b) shows the average loss observed by TCP flows for different loads and flow size distributions (α = 1.2, 1.5, 2.0). The impact of the heavy-tailedness of the flow size distribution is visible. The lower the value of α, the heavier the tail of the flow size distribution, and the higher the packet losses due to a larger number of small flows and the few large flows. When traffic load is high the impact of heavy-tails on packet loss is limited by the way TCP is restricted in its burstiness. In the rest of the paper, we present results with α = 1.2, which matches real Internet traffic.

2) Flow-level packet loss: So far, we have considered overall packet loss statistics, i.e., one that takes places across all flows that share the bottleneck link. However, we already observed in Section IV-A that flows have their own view of the loss process. Next, we study the loss process of flows individually across different flows sizes. We start by studying how different flow sizes are impacted by losses for different offered loads. For each individual flow, the relevant information is not the overall loss rate but the fraction of its packets that have been dropped. Therefore, we compute packet



(a) Low load

(b) High load

(c) Very high load

Fig. 4. Per flow-size packet loss probability for different loads (128 packet buffer).

loss rate for each flow as the fraction of packets that were dropped divided by the total number of packets sent by the sender.

Figure 4 shows the per-flow packet drop probability (y-axis, using box-plots) across different TCP flow sizes (x-axis) and for low, high, and very high offered load. Box-plots show the minimum, the percentiles 25, 50, 75, and the maximum. Flow sizes are binned into logarithmic sizes. The total packet loss probability in this scenario is rather small with 1%. However, the loss is not distributed evenly across all flows.

Under low load (Figure 4(a)), we observe that flows with sizes from 512KB to 32MB suffer from higher loss rates compared to other flow sizes. When the link utilization is high (Figure 4(b)), a subset of the small flows suffer from larger packet loss probability than the set of larger flows. Note, most of the small flows still have a very small packet loss probability. Only some unlucky flows see more losses than the rest of the flows, for a given flow size.

Under high load, the average packet loss probability across all flows sizes increases. Small flows tend to have a few unlucky flows that suffer from very high loss probabilities. A higher fraction of larger flows (larger than 16KB) experience packet loss rates of roughly the same rate as the total packet loss rate. Even under very high offered load some happy flows do not observe significant losses.

We conclude that irrespective of the offered load, the observed packet loss probability of a single flow is unlikely to be representative of the total packet loss rate. Even very large flows, which one may expect to experience losses close to the overall loss rate, can observe packet loss probabilities that differ significantly from the overall one. In general, most flows will not observe many packet losses making them *happy flows*. However, some specific flows might observe unusually high packet loss probabilities only when subjected to severe congestion—just as some of our CDN's *unhappy flows* from Section IV-A.

3) Flow performance and congestion window: Finally, we investigate the bursty behavior of individual flows to correlate flow performance and flow size. Given that TCP state is only maintained accurately in the end-host operating system, we collect TCP congestion window data on the servers in our testbed setup (See Section III-B).

Since, most of the flows in the Internet are small, we cannot expect that TCP reaches to the congestion avoidance phase for small flows. Conversely, large flows are expected to take advantage of the full network capacity. To confirm this intuition, we plot Figure 5. The x-axis shows the TCP flow sizes while the y-axis shows, using a box-plot, the distribution of the average TCP congestion window size over the flow lifetime. This plot corresponds to a low load scenario with buffer size of 128 packets.

Figure 5 reveals a non-linear trend and indicate that only large flows manage to reach an average window size of the same order of magnitude as the buffer size. Except for very small flows and very large ones, the average window size grows with the flow size until it reaches values in the order of the buffer size. Note, the congestion window can take values as large as twice the buffer size before TCP will be signaled that congestion occurred at the buffer.

Interestingly, those very flows in the range of 1MB-16MB for which the TCP congestion window grows beyond the buffer size are those who observe the unusually high packet loss. This means that the flows of such sizes are burstier than the large flows whereas small flows finish before they could reach the maximum buffer size. Note that 15% of the flows in the 256KB-4MB range in our CDN data show retransmission rates greater than 5%. The main cause of such a high retransmission rate is the very bursty nature of these flows as indicated by the average congestion window size.

Large flows ($>$ 16MB) show a steady behavior with not so high average congestion window and packet loss rate. Given that the throughput achieved by a TCP flow depends highly on the congestion window, large flows do not fully utilize the available bandwidth as indicated by their average window size. In that case, the actual throughput that can be achieved by a TCP flow is much lower than what one might expect, affecting flow performance. This phenomenon has been observed in residential traffic [2].

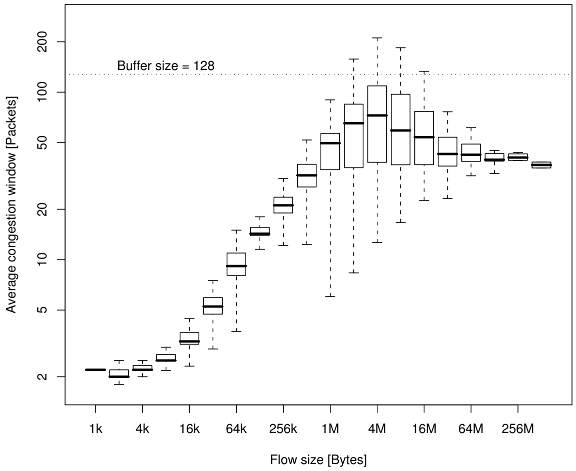


Fig. 5. TCP congestion window against flow size (low load).

V. Related Work

In the past many researchers have studied the correlation between packet losses on Internet paths, e.g., [15], [16], [17]. These approaches relied on active measurements for sampling the path properties of the data plane, e.g., send probes every tens of milliseconds. Due to this sampling, the actual loss process has to be inferred from observed losses experienced by the probes. While such an inference process might accurately estimate the average path loss observed by a flow on a given path, it does not provide an accurate view about how the losses are distributed among the flows.

Sun et al. [18] studied the performance bottlenecks of CoralCDN in PlanetLab. Their findings suggest that 10% of the connections are server-limited and connections with no packet loss can be congestion window limited. Heikkinen et al. [19] compared the web characteristics of fixed and mobile users in terms of bytes-per-connection and packet loss. Similarly, Lee et al. [20] studied the performance of a congested academic network. Zhang et al. [21] have found that flow size and flow rate are two highly correlated metrics.

In the context of applications and losses, Alcock et al. [22] have exposed the problem of YouTube block send, that causes unexpected losses. Izal et al. [23] have studied the behavior and performance of Bittorrent over a period of multiple months.

VI. Summary

The popularity of web-based applications have changed the Internet traffic landscape all together. In this paper, we have brought a new perspective on the performance of flows in the Internet, through *flow-classes* based on flow sizes.

Leveraging a data set collected at a major content distribution network, we investigated flow performance, across flow sizes. Surprisingly, we found that packet losses do not affect all flows similarly. To find out the causes behind our observations in the data, we rely on a controlled testbed setup using advanced instrumentation via NetFPGA. We essentially target the question of how packet losses impact individual flows, depending on specific network loads and flow-size distributions.

We found that, depending upon the network load, there are few unhappy flows, especially small ones. On the other hand, most flows, especially large ones, are happy and do not observe high losses compared to the overall loss rate. Furthermore, very few flows actually observe a loss rate similar to the average loss rate. Therefore, any single flow is very unlikely to observe the overall packet loss process.

In future work, we will further investigate the relationship between different factors that affect flow performance, such as the interactions between specific applications, the network conditions, and user experience.

References

[1] "Strangeloop Site Optimizer," http://www.strangeloopnetworks.com/assets/Uploads/SO-Datasheet.pdf.

[2] G. Maier, A. Feldmann, V. Paxson, and M. Allman, "On dominant characteristics of residential broadband internet traffic," in *ACM IMC*, 2009.

[3] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, "Internet inter-domain traffic," in *Proceedings of the ACM SIGCOMM*, 2010.

[4] M. E. Crovella and A. Bestavros, "Self-similarity in World Wide Web traffic: Evidence and possible causes," *IEEE/ACM ToN*, 1997.

[5] S. Ihm and V. S. Pai, "Towards understanding modern web traffic," in *ACM IMC*, 2011.

[6] N. Dukkipati, M. Mathis, Y. Cheng, and M. Ghobadi, "Proportional rate reduction for TCP," in *ACM IMC*, 2011.

[7] M. Butkiewicz, H. Madhyastha, and V. Sekar, "Understanding website complexity: measurements, metrics, and implications," in *ACM IMC*, 2011.

[8] A. Mehmood, A. Wundsam, S. Uhlig, D. Levin, N. Sarrar, and A. Feldmann, "QoE-Lab: Towards evaluating Quality of Experience for Future Internet Conditions," in *Proceedings of TRIDENTCOM*, April 2011.

[9] J. Sommers and P. Barford, "Self-configuring network traffic generation," in *ACM IMC*, 2004.

[10] R. Prasad and C. Dovrolis, "Measuring the congestion responsiveness of Internet traffic," in *PAM*, 2007.

[11] L. Rizzo, "Dummynet: a simple approach to the evaluation of network protocols," *ACM CCR*, 1997.

[12] "NetFPGA Project," http://yuba.stanford.edu/NetFPGA/.

[13] J. Yoo, T. Huhn, and J. Kim, "Active capture of wireless traces: overcome the lack in protocol analysis," in *Proceedings of WiNTECH*, 2008.

[14] K. Pentikousis, H. Badr, and A. Andrade, "A comparative study of aggregate TCP retransmission rates," 2011, http://arxiv.org/abs/1112.2292.

[15] J.-C. Bolot, "End-to-end packet delay and loss behavior in the Internet," in *Proceedings of the ACM SIGCOMM*, 1993.

[16] H. Nguyen and M. Roughan, "On the correlation of internet packet losses," in *Proceedings of ATNAC*, 2008.

[17] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and modelling of the temporal dependence in packet loss," in *Proceedings of the IEEE INFOCOM*, 1999.

[18] P. Sun, M. Yu, M. Feedman, and J. Rexfor, "Identifying performance bottlencks in CDNs through TCP-level monitoring," in *Proceedings of the ACM SIGCOMM workshop W-MUST*, 2011.

[19] M. Heikkinen and A. Berger, "Comparison of user traffic characteristics on mobile-access versus fixed-access networks," in *PAM*, 2012.

[20] C. Lee, D. K. Lee, Y. Yi, and S. Moon, "Operating a network link at 100%," in *PAM*, 2011.

[21] Y. Zhang, L. Breslau, V. Paxson, and S. Shenker, "On the characteristics and origins of internet flow rates," in *Proceedings of the ACM SIGCOMM*, 2002.

[22] S. Alcock and R. Nelson, "Application flow control in YouTube video streams," *ACM SIGCOMM Computer Communication Review (CCR)*, 2011.

[23] M. Izal, G. Urvoy-Keller, E. Biersack, P. Felber, A. Hamra, and L. Garcés-Erice, "Dissecting bittorrent: Five months in a torrent's lifetime," in *Proceedings of the Passive and Active Measurement Conference (PAM)*, 2004.