The Local and Global Effects of Traffic Shaping (Extended Version)

Massimiliano Marcon
Marcel Dischinger
Krishna Gummadi
Amin Vahdat

MPI–SWS–2010–004
September 2010

Abstract

The Internet is witnessing explosive growth in traffic, in large part due to bulk transfers. Delivering such traffic is expensive for ISPs because they pay other ISPs based on peak utilization. To limit costs, many ISPs are deploying ad-hoc traffic shaping policies that specifically target bulk flows. However, there is relatively little understanding today about the effectiveness of different shaping policies at reducing peak loads and what impact these policies have on the performance of bulk transfers.

In this paper, we compare several traffic shaping policies with respect to (1) the achieved reduction in peak network traffic and (2) the resulting performance loss for bulk transfers. We identified a practical policy that achieves peak traffic reductions of up to 50% with only limited performance loss for bulk transfers. However, we found that the same policy leads to large performance losses for bulk transfers when deployed by multiple ISPs along a networking path. Our analysis revealed that this is caused by certain TCP characteristics and differences in local peak utilization times. We present a data staging service that counteracts the deleterious end-to-end effects of local traffic shaping by delaying bulk data at appropriate points in the network to consume bandwidth when available. We show that the service is incrementally deployable and has reasonable storage requirements.

1 Introduction

The Internet is witnessing explosive growth in demand for bulk content. Examples of bulk content transfers include downloads of music and movie files [20], distribution of large software and games [18, 45], online backups of personal and commercial data [4], and sharing of huge scientific data repositories [44]. Recent studies of Internet traffic in commercial backbones [30] as well as academic [13] and residential [14] access networks show that such bulk transfers account for a large and rapidly growing fraction of bytes transferred across the Internet.

The bandwidth costs of delivering bulk data are substantial. A recent study [33] reported that average monthly wholesale prices for bandwidth vary from $30,000 per Gbps/month in Europe and North America to $90,000 in certain parts of Asia and Latin America. The high cost of wide-area network traffic means that increasingly *economic* rather than *physical* constraints limit the performance of many Internet paths. As charging is based on peak bandwidth utilization (typically the 95^{th} percentile over some time period), ISPs are incentivized to keep their bandwidth usage on inter-AS links much lower than the actual physical capacity.

To control their bandwidth costs, ISPs are deploying a variety of ad-hoc traffic shaping policies today. These policies target specifically bulk transfers, because they consume the vast majority of bytes [13, 35, 40]. However, these shaping policies are often blunt and arbitrary. For example, some ISPs limit the aggregate bandwidth consumed by bulk flows to a fixed value, independently of the current level of link utilization [24]. A few ISPs even resort to blocking entire applications [19]. So far, these policies are not supported by an understanding of their economic benefits relative to their negative impact on the performance of bulk transfers, and thus their negative impact on customer satisfaction.

Against this backdrop, this paper poses and answers three questions:

1. What reduction in peak utilization (and cost) can an ISP achieve by traffic shaping only bulk flows? How do policies that minimize

peak utilization affect the performance of bulk flows? Using traces from the access links of 35 universities, we show that diurnal patterns in bandwidth consumption offer a significant opportunity for intelligent traffic shaping that observes economic incentives and minimizes the peak levels of bandwidth consumption. We investigate techniques that limit the bandwidth consumed by bulk transfers during times of peak utilization, effectively smoothing bandwidth consumption over the course of the day. We find that a composition of traffic shaping and simple queuing techniques can achieve significant reductions in peak bandwidth, while minimally impacting completion times of individual bulk transfers. By contrast, we show that naive traffic shaping techniques either cannot achieve similar reductions in peak load or dramatically slow down many targeted flows.

2. As economic considerations drive all ISPs to adopt locally-optimal traffic shaping policies at their edges, how would bulk transfers that traverse multiple inter-ISP links be affected? Given the significant reduction in peak bandwidth usage (and thus in costs) that can be achieved with traffic shaping of only bulk flows, it is very likely that most ISPs would adopt such policies eventually. However, we found that even if ISPs deploy policies that are designed to minimize the *local* performance loss of bulk flows, the *global* performance loss of flows traversing multiple traffic shapers is substantial. In our analysis we found that this is caused by TCP characteristics and differences in local peak utilization times of ISPs.

3. Can ISPs avoid the deleterious global effects of local traffic shaping, without compromising their economic self-interest? Surprisingly, we find that there exists a rather simple answer to this problem. As flows are negatively affected only when they traverse more than one traffic shaper, the solution is to break long transfers into a number of subtransfers, ensuring that each subtransfer traverses only one traffic shaper. This preserves the local benefits of traffic shaping policies for ISPs and at the same time enables bulk transfer to efficiently exploit the bandwidth of the network path. The price to pay for this reconciliation of interests is storage, a relatively cheap resource whose price is steadily dropping.

The rest of this paper is structured as follows: Section 2 describes real-world traffic shaping policies in use today. Section 3 discusses the goals of an ideal traffic shaping policy. Section 4 compares different traffic shaping policies when traffic traverses only one traffic shaper, Section 5 analyzes the effects of multiple shapers active on a network path and Section 6 presents and evaluates our proposal to break end-to-end transfers. Finally, Section 7 discusses related work and Section 8 concludes the paper.

2 Traffic shaping policies in use today

ISPs today deploy a variety of traffic shaping policies. The main goal of these policies is to reduce network congestion and to distribute bandwidth fairly amongst customers [9]. This is typically achieved by reducing the peak network usage through traffic shaping applied either to single flows or to the aggregate traffic of a user. The reduction in peak network usage also has the side-effect that it reduces inter-AS traffic and thus bandwidth costs. At the same time, ISPs are also concerned to affect as few flows as possible to keep the effect on user traffic low [39].

To the best of our knowledge, there exists no previous study that analyzed the exact benefits of these policies and their impact on targeted flows when deployed in practice. In this section, we present three canonical examples of traffic shaping policies in use today. Most of these policies traffic shape bulk transfers [9, 24]. We investigate the benefits of these policies and compare them with more sophisticated policies in later sections.

1. Traffic shaping bulk applications on a per-flow basis. This policy shapes every flow belonging to bulk transfer applications to some fixed bandwidth. For example, Bell Canada revealed that it throttles traffic from P2P file-sharing applications in its broadband access networks to 256 Kbps per flow [9]. Traffic shaping applies to flows both in the downstream and in the upstream direction. Bell Canada chose to traffic shape only P2P file-sharing applications because it found that a small number of users of these applications were responsible for a disproportionate fraction of the total network traffic.

2. Traffic shaping aggregate traffic. Here, traffic shaping is applied to the aggregate traffic produced by multiple network flows. For example, Comcast handles congestion in its access network by throttling users who consume a large portion of their provisioned access bandwidth over a 5-minute time window [17]. All packets from these users are put in a lower

priority traffic class in order to be delayed or dropped before other users' traffic in case of network congestion. Another example of such a policy was deployed at the University of Washington in 2002 [24]. The university started limiting the aggregate bandwidth of all incoming peer-to-peer file-sharing traffic to 20 Mbps to reduce the estimated costs of one million dollars this type of traffic was causing per year.

3. Traffic shaping only at certain times of the day. This policy is orthogonal to the previous two policies and is typically used in combination with these. An ISP can decide to traffic shape throughout the day or restrict traffic shaping to specific time periods. For example, the University of Washington shapes P2P traffic during the entire day [24], while Bell Canada and Kabel Deutschland announced to only traffic shape during periods of "peak usage", i.e., between 4:30 pm and 2:00 am [9, 39]. Since many ISPs pay for transit bandwidth based on their peak load, shaping only during peak usage appears to be an effective way to reduce bandwidth costs.

While the above policies are simple to understand, they raise several questions:

1. How effective are the different traffic shaping policies at reducing network congestion and peak network usage?

2. What is the impact of traffic shaping policies on the performance of the targeted network flows?

3. Are there policies that achieve similar or better reduction in bandwidth costs, while penalizing traffic less?

To answer these questions, we first need to define the precise goals of traffic shaping, as well as the metrics with which we evaluate the impact of traffic shaping policies on network traffic.

3 Goals and potential of traffic shaping

In this section, we identify three goals for traffic shaping policies as deployed by ISPs: minimizing the peak network traffic, minimizing the number of flows targeted by traffic shaping, and minimizing the impact of traffic shaping on flows. We argue that traffic shaping policies should be designed around these goals, and quantify the potential of such policies through an analysis of real-world network traces.

3.1 Network traces

In our analysis of traffic shaping performance, we use publicly available Netflow records collected at the access links of 35 different universities and research institutions. The records contain incoming and outgoing traffic between these universities and the Abilene backbone [2]. Even though our traces come from a university environment, we confirmed that the relevant trace characteristics for our analysis (such as diurnal variations and skewness in flow size distribution) are consistent with those observed in several previous studies of commercial Internet traffic [3, 6].

The Netflow records were collected during a 1-week period starting on January 1st 2007, and contain durations and sizes of TCP flows. The Netflow data has two limitations: (1) long flows are broken down into shorter flows (with a maximum duration of 30 minutes), and (2) flows' packets are sampled with a 1% rate. To recover long flows from the Netflow data, we combine successive flows between the same TCP endpoints into longer flows using the technique employed in [31]. To account for the sampling rate, we multiply packet and byte counts by 100. While this approach is not reliable when applied to small flows, it was shown to be accurate for large bulk flows [41], which are the object of the traffic shaping policies considered in this paper.

3.2 Goals and potential

We identify the following three goals as the main practical objectives for an ISP that deploys traffic shaping.

Goal 1: Minimizing the peak network traffic. The main motivation for ISPs to deploy traffic shaping is often network congestion [9, 39]. With traffic shaping, ISPs can lower the risk of congestion by reducing the peak network usage. At the same time, lowering the peak network usage also reduces bandwidth costs for ISPs since they are often charged based on the near-peak utilization (e.g., 95^{th} percentile traffic load) of their links. This creates an incentive for ISPs to keep the peak network usage as low as possible to minimize bandwidth costs. Using our traces, we quantify the maximum peak reduction in network traffic ISPs can achieve with an optimal traffic shaping policy.

Figure 3.1 plots the network traffic in one of our traces (collected at the Ohio State university). The traffic exhibits strong diurnal variations, with traffic peaking around noon and dropping in the early morning. As a result of these variations, the daily traffic peak is considerably higher than the average daily traffic. Intuitively, the lower bound for any realistic peak reduction scheme is the average daily traffic, because this is the minimum traffic level that can assure that all traffic will eventually be delivered within the day[1].

Averaging across all access link traces, the daily peak is 2.6 times larger than the average traffic load. With respect to 95^{th} percentile, the peak is 1.7 times larger than the average traffic. These results suggest that traffic shaping has the potential to reduce ISPs' peak load by a factor of 2.

Goal 2: Minimizing the number of traffic shaped flows. While ISPs have an economic incentive to reduce the peak network usage as much as possible, they are also concerned with affecting as few flows as possible to keep the effect on user traffic low. As a consequence, most ISPs today target either users that are responsible for a disproportional large fraction of traffic (so-called "heavy-hitters"), or applications known to be bandwidth-hungry (e.g., file-sharing applications). Using our traces, we quantify the minimal fraction of bulk flows that need to be shaped to achieve a near-optimal reduction in peak load.

Typically, an ISP would use deep packet inspection to identify flows belonging to bandwidth-intensive applications. However, since our traces

[1] A higher peak reduction is only possible if traffic is dropped from the network, e.g., by blocking certain applications traffic. However, blocking is a very intrusive form of traffic shaping and ISPs that previously deployed it had to deal with very negative media coverage about this practice [43].

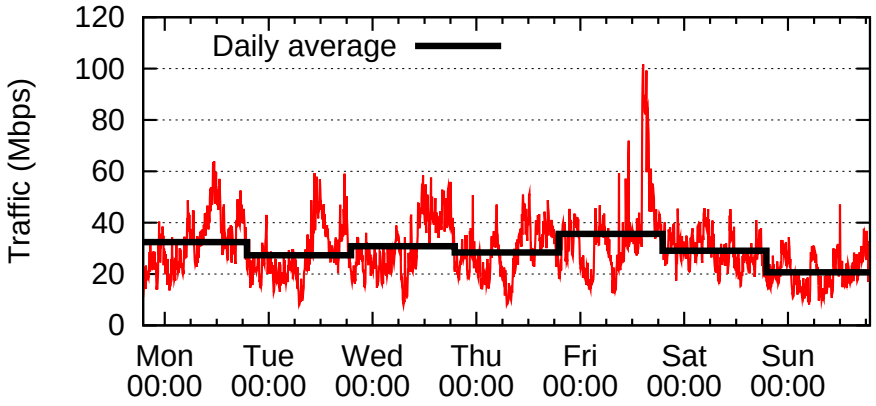


Figure 3.1: Downstream network traffic at Ohio State University: The traffic shows diurnal variations with large peak-to-average ratios.

do not contain information about application-level protocols, we identify bandwidth-intensive flows based on their size, i.e. the number of transferred bytes.

We sorted all flows in each of our trace by decreasing size. We then selected all flows larger than a certain size T for traffic shaping and computed the theoretical maximum peak reduction achievable. For this analysis, we assume that flows can be arbitrarily throttled, as long as they complete within the trace's time-frame of 1 week. We then repeated this for decreasing values of T, thus selecting more and more flows. Figure 3.2 plots the results for one of our traces. After selecting only 0.4% of the largest flows, the traffic peak reaches the average traffic load and no further reduction is possible (the "knee" in the figure). In this trace, this translates to flows that are larger than 10 MB. Across all traces, traffic shaping less than 4% of the flows is always sufficient to achieve the maximum peak reduction, and in 30 of our 35 traces traffic shaping less than 1% of the flows also suffices. This result suggests that ISPs can considerably reduce their peak while shaping a very small fraction of flows.

Goal 3: Minimizing the delay that traffic shaped flows incur. We found that ISPs have to shape only a small fraction of flows to achieve an optimal reduction in peak network usage. Note that this optimal reduction can be achieved without dropping any flows. Instead, in our analysis, we ensured that all shaped flows complete within the time-frame of the trace. However, even if only a small fraction of flows are affected by traffic shaping, ISPs should try to limit the delay incurred by these flows in order to minimally penalize the applications or users generating the bulk flows. With respect to this goal, focusing on bulk flows has the advantage that these

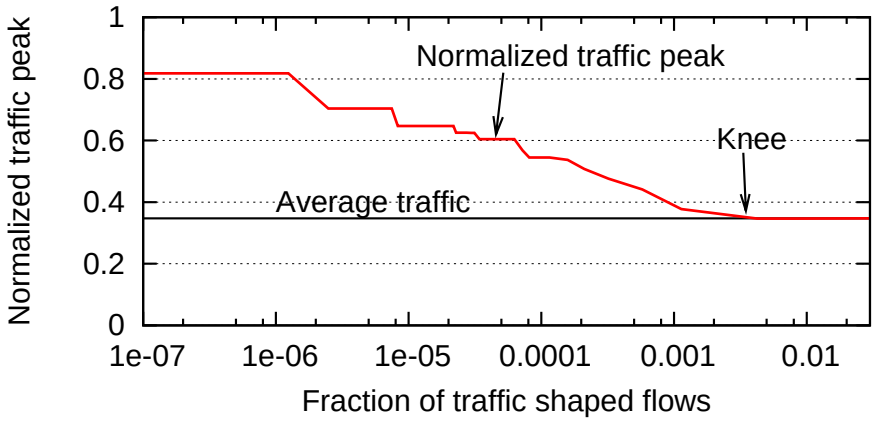


Figure 3.2: Tradeoff between maximum achievable peak reduction and fraction of traffic shaped flows: Intuitively, shaping more flows lowers the peak. However, the peak cannot be lower than the average traffic rate without dropping flows. At this point, shaping more flows has no further benefits.

flows, being large, have completion times on the order of minutes, hours or even days. Therefore, they can endure considerable absolute delays without severe damage to their performance. For example, the bulk flows in our trace take on average 3.5 minutes to complete when they are not traffic shaped, suggesting that they can be delayed by seconds without negative effects for applications.

In summary, we found that a traffic shaping policy should not only minimize the peak network traffic, but also affect as few flows as possible and minimize its impact on the shaped flows. In the next section, we compare how well different traffic shaping policies perform relative to these goals.

4 Local performance of traffic shaping policies

In this section we analyze how different traffic shaping policies perform based on the three metrics from Section 3: the peak reduction, the fraction of shaped flows, and the delay shaped flows incur. As we only consider a single traffic shaper in the network path here, we call this the local performance of traffic shaping policies. In Section 5, we analyze the effect of multiple traffic shapers in the networking path.

4.1 Selecting flows for traffic shaping

ISPs target only a subset of flows for traffic shaping, typically flows from bandwidth-intensive applications. Doing so, ISPs achieve very good peak reductions while keeping the number of affected flows low. In the following, we call flows that are subject to traffic shaping "low-priority traffic" and the remaining flows "best-effort traffic".

To identify flows from bandwidth-intensive applications, ISPs often employ deep packet inspection (DPI), which is widely available in routers [15] or provided by special DPI equipment [38]. Additionally, today's networking equipment allows ISPs to collect statistics on flow sizes, which can be used to mark large flows for traffic shaping [15, 29]. In practice, flow classification is implemented at ISPs' ingress routers. Flows are marked as low-priority or best-effort by setting the DSCP field in the IP header[1]. The traffic shaping equipment then selects the packets to traffic shape just based on the value of the DCSP field.

As our traces do not contain information to identify application protocols, we rely on flow sizes instead, i.e., flows that are larger than a certain "flow size threshold" are shaped. Picking the right flow size threshold is nontrivial,

[1]The DSCP field allows up to 64 different traffic classes.

because a higher threshold will affect fewer flows, but at the same time will give ISPs fewer bytes to traffic shape, and thus limit its ability to decrease peak usage. To select the right threshold for each trace, we use the analysis from Section 3.2 and pick the threshold that results in the maximum potential for peak reduction with the minimum fraction of flows being shaped.

In the traffic shaping policies in this section, unless explicitly stated otherwise, we keep a running counter of the bytes sent by each active network flow, and use its value to classify the flow. For example, if the flow size threshold is 10 MB, a 20 MB flow will send the first 10 MB as best-effort traffic. After that, the flow is classified as low-priority traffic and the remaining 10 MB of the flow are traffic shaped. This technique can also be used by ISPs to deploy a protocol-agnostic traffic shaping policy that targets all flows larger than certain flow size threshold. Note that modern traffic shaping equipment is capable of keeping such per-flow state even on high-speed links [15].

4.2 Selecting aggregate bandwidth limits

Some traffic shaping policies (e.g., as used by the University of Washington [24]) shape low-priority flows only when the traffic rate exceeds a certain "bandwidth limit". This limit can refer to the aggregate traffic (best-effort + low-priority traffic) or to the low-priority traffic only. For example, an ISP could traffic shape only when the total traffic rate exceeds 20 Mbps or when the low-priority traffic alone exceeds 20 Mbps.

The bandwidth limit determines the total reduction in traffic peak. As we showed in Section 3, the average traffic rate is the minimum value that enables delivery of all traffic. Therefore, in all policies that use a bandwidth limit, we set the bandwidth limit to the average traffic rate of the previous day plus 5% to account for small increases in demand. We found that this approach works well in practice because the average rate is quite stable across days. In fact, in our 35 1-week traces, we found only two days were this was not the case, i.e., the average traffic varied considerably from one day to the next. If there is a sudden increase in daily average traffic, too many low-priority flows may compete for too little bandwidth, thus incurring large delays or even starvation. To overcome this problem, ISPs can monitor the bandwidth of the low-priority flows and of the overall traffic in their network and increase the bandwidth limit if they detect a significant difference from the previous day.

4.3 Traffic shaping policies

We now describe the traffic shaping policies we evaluate. All of the traffic shaping policies described here can be implemented using well-known elements like token buckets, class-based rate limiting, and strict priority queuing, available in today's networking equipment [16, 34]. To design the traffic shaping policies we start from the real-world examples from Section 2 and develop more complex policies, which attempt to reduce the peak traffic while minimize the delay incurred by the traffic shaped flows. Note that all of the traffic shaping policies presented here shape only flows classified as low-priority; best-effort traffic is never shaped.

Per-flow bandwidth limit (PBL). With PBL, each low-priority flow is shaped to a fixed maximum bandwidth. Traffic shapers use a dedicated queue for each low-priority flow, and dequeue packets according to a token bucket algorithm. In our simulations, we limit the bandwidth consumed by each low-priority flow to 250 Kbps.

We also evaluate a variant of this policy called **PBL-PEAK**, where low-priority flows are shaped only between 9 am and 3 pm local time. This period corresponds to 6 hours centered around the peak utilization in our traces at about noon. Both PBL and PBL-PEAK require routers to allocate a new queue for each new low-priority flow, thus potentially limiting the practicality of these two policies.

Low-priority bandwidth limit (LBL). In this policy, the aggregate bandwidth consumed by all low-priority flows is bound by a bandwidth limit. Traffic shapers deploy two queues: one for best-effort traffic and one for low-priority traffic. A token bucket applied to the low-priority queue limits the low-priority traffic rate to the desired bandwidth limit. The bandwidth limit is determined based on the average bandwidth consumed by low-priority traffic on the previous day, as described before. No bandwidth limit is applied to the best-effort traffic. This policy can also be used to approximate PBL by using a dynamic bandwidth limit proportional to the number of low-priority flows.

Aggregate bandwidth limit (ABL). When the aggregate traffic (best-effort + low-priority traffic) approaches the bandwidth limit, low-priority flows are shaped to keep the aggregate traffic below the limit. Note that best-effort traffic is never shaped. Therefore, if the best-effort traffic exceeds the bandwidth limit, this policy cannot guarantee that the aggregate traffic stays below the bandwidth limit. However, in such cases the traffic shaper throttles the low-priority traffic to zero bandwidth until the best-effort traffic falls below the bandwidth limit.

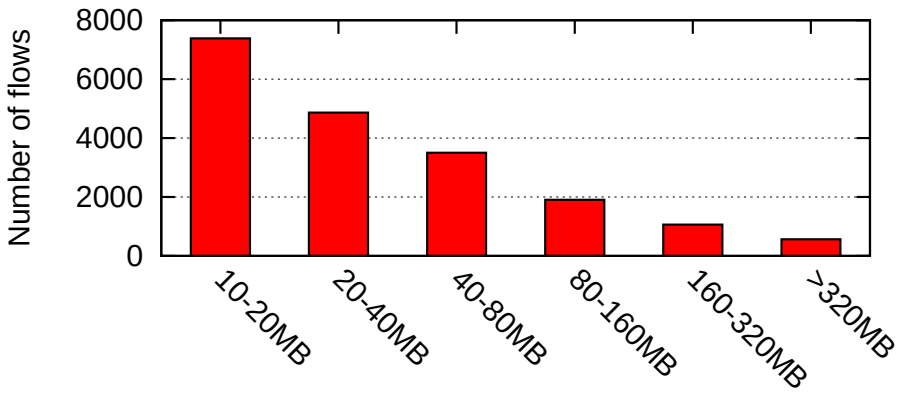


Figure 4.1: Number of flows >10 MB per flow size range for the Ohio State trace.

To implement this policy, traffic shapers deploy two queues: a high-priority queue for the best-effort traffic and a low priority queue for the low-priority traffic. Both queues share a single token bucket, which generates tokens at a rate corresponding to the aggregate bandwidth limit. Each time a packet from either queue is forwarded, tokens are consumed. However, best-effort packets are always granted access to the link, even if there are not enough tokens left. This is unlike an ordinary token bucket and can cause the token count to occasionally become negative, thus precluding low-priority packets from using the link. As long as the total traffic rate is below the bandwidth limit, there are always enough tokens to forward both best-effort and low-priority traffic. But, as the total traffic level exceeds the bandwidth limit, low-priority flows are shaped.

Aggregate bandwidth limit with shortest-flow first scheduling (ABL-SFF). This policy is as ABL, but additionally optimizes the usage of the bandwidth available to the low-priority flows. Unlike PBL or LBL, in ABL low-priority traffic is not guaranteed a minimum bandwidth allocation, but all low-priority flows compete for the bandwidth the best-effort traffic is not using. Thus, when the total traffic reaches the bandwidth limit, the bandwidth available to low-priority flows becomes so low that some of these flows get substantially delayed or even stalled.

We gained an insight on how to lessen this problem by looking at the flow-size distribution in our traces. Figure 4.1 shows the number of low-priority flows that fall into different size ranges in one of our traces. The distribution of flow sizes is heavily skewed with roughly 85% of low-priority flows having size between 10 MB and 100 MB. Such a flow size distribution is quite common in Internet traffic. [40]. Under such skewed distributions, it is

well-known that giving priority to small flows reduces the mean completion time [27,36]. Therefore, in the ABL-SFF policy, when selecting a low-priority packet to send over the link, the traffic shaper always chooses the packet from the flow with the smallest size. This effectively replaces the usual FIFO queueing with shortest-flow-first queueing. To implement this policy, the traffic shaper needs to allocate a separate queue for each low-priority flow. Also, the shaper needs a priori knowledge of the size of each flow to select the next low priority packet. This makes this policy not directly applicable to general network flows, whose size cannot be known, but gives an useful lower-bound on the minimum delay that low-priority flows incur with the ABL policy.

Aggregate bandwidth limit with strict priority queuing (ABL-PQ). This policy is a practical version of ABL-SFF and can be implemented by ISPs with today's equipment. It approximates the shortest flow first scheduling of ABL-SFF as follows. First, unlike ABL-SFF, it does not assume a priory knowledge of flow sizes, but instead keeps a running count of the bytes sent by each active network flow and uses this value as an estimate of the flow size. Second, ABL-PQ does not use a separate queue for each low-priority flow, but instead uses a fixed, small number of low-priority packet queues. Each queue accommodates packets of low-priority flows whose size fall in a given range. When the traffic shaper has bandwidth to send low-priority traffic, it schedules the low-priority queues giving *strict priority* to the queues that accommodate smaller flows.

To balance the load of the low-priority queues, we selected contiguous ranges of exponentially increasing width. This is motivated by the typical skewness of the flow size distribution in the Internet. For our traces, where flows larger than 10 MB are classified as low-priority traffic, the first low-priority queue contains packets of flows that have transferred between 10 MB and 20 MB, the second queue contains packets of flows that have transferred between 20 MB and 40 MB, and so on. As opposed to ABL-SFF, this policy uses a limited number of queues (we use 6 in our experiments) and can be easily supported by today's networking equipment. Remember that ISPs typically deploy flow classification at their ingress points and that network equipment is capable of keeping per-flow state [15, 29].

4.4 Comparison methodology

We used trace-driven simulations to study the behavior of flows under various traffic shaping mechanisms. We conducted our analysis using the ns-2

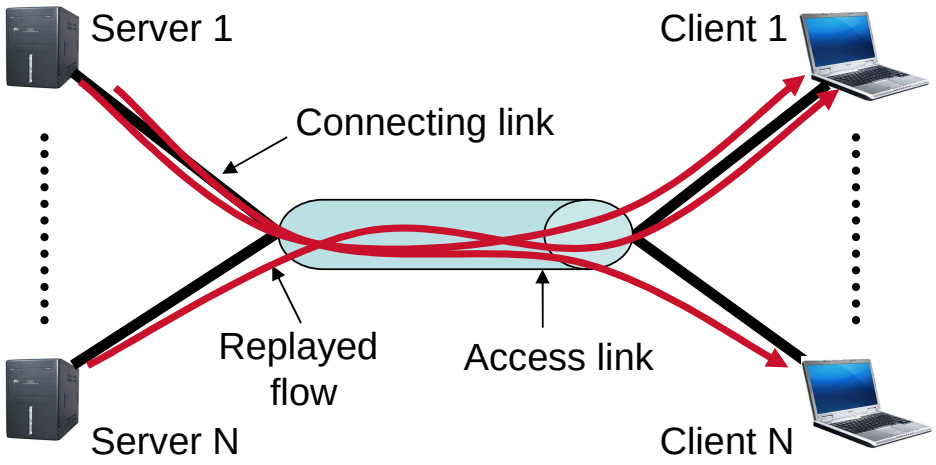


Figure 4.2: Simulation topology: All replayed TCP flows cross a shared access link where traffic shaping takes place.

simulator and the traces from Section 3. During the simulation, we replayed all TCP flows in each trace using the ns-2 implementation of TCP-Reno.

We used the simulation topology shown in Figure 4.2 to analyze traffic shaping over an access link. We faced an interesting challenge while replaying the TCP flows: our traces included information about flow arrival times, sizes, and durations, but we lacked information about flow round-trip times (RTTs) and loss rates. To simulate packet losses, we set the capacity of the link connecting the server node for each flow to match the average bandwidth of the flow (see Figure 4.2). This ensures that the simulated flows complete in similar durations as the original flows in the trace. Furthermore, we picked the RTT of a flow choosing from a distribution of latency measurements using the King tool [26]. We found that the aggregate bandwidth of the simulated flows match the one of the original flows from the traces very well.

To compare different traffic shaping policies, we focused on the three metrics from Section 3: the achieved peak reduction, the fraction of shaped flows, and the delay shaped flows incur.

4.5 Results

We now present the results of the comparison of the different traffic shaping policies.

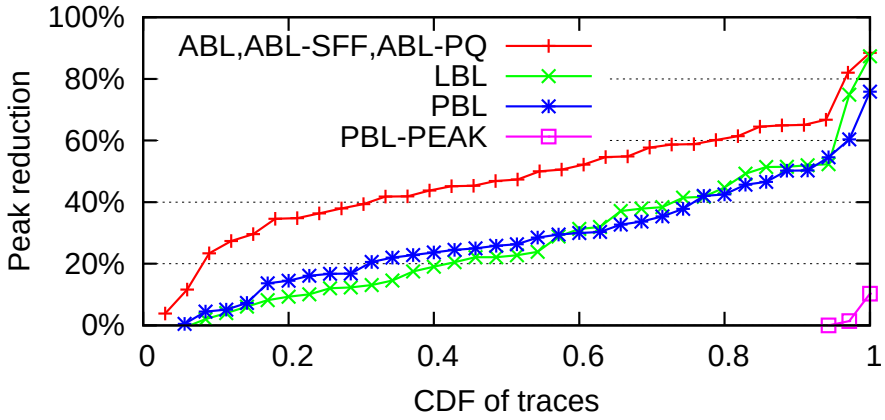


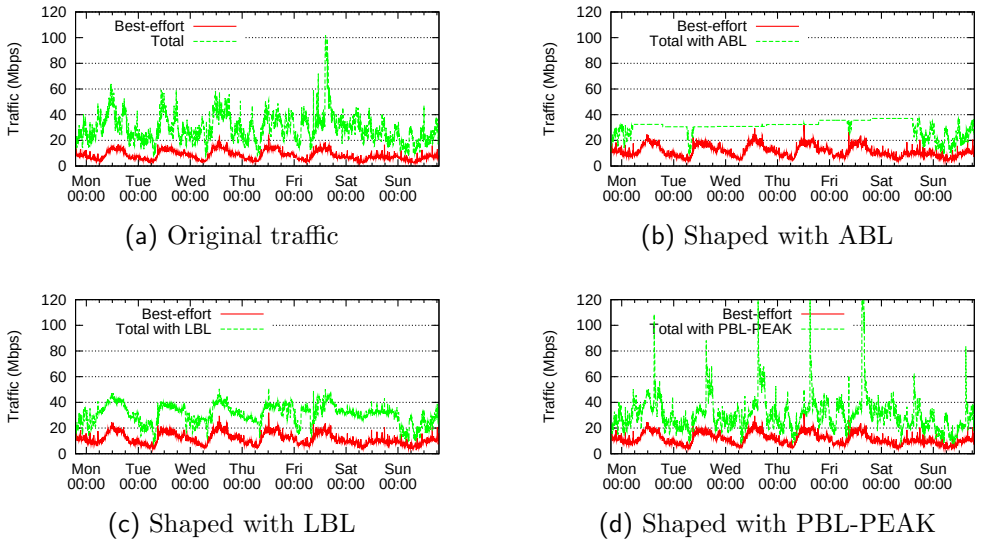
Figure 4.3: Reduction in peak with different traffic shaping policies: Traffic shaping policies based on aggregate bandwidth limits (ABL) achieve considerable reductions in peak traffic.

4.5.1 Peak reduction

We start by presenting the overall peak reductions attained by the different policies across all our traces, shown in Figure 4.3. Since ABL, ABL-SFF and ABL-PQ all cap the traffic at the same limit, we report only one line for all of them. The ABL policies achieve a considerably higher peak reduction than LBL. This is because LBL does not take into account the level of best-effort traffic when computing the low-priority traffic cap. PBL performs similarly to LBL, while PBL-PEAK is by far the worst-performing policy, causing in 90% of the cases an *increase* in traffic peak (these correspond to points that lie on the negative side of the y-axis in the figure, and are not shown).

To better understand the differences in peak reduction among the different policies, we show in Figure 4.4 time plots of the traffic in an example trace. Flows smaller than 10 MB are marked as best-effort traffic. Figure 4.4(a) shows the original traffic trace without traffic shaping. Compared to the original trace, the ABL policies (Figure 4.4(b)) considerably reduce peak bandwidth (-64%). LBL (Figure 4.4(c)) achieves lower, but still substantial reductions (-51%).

Comparing LBL and ABL, we observe that ABL achieves a much smoother peak as the total amount of traffic is capped to a constant daily threshold (note that best-effort traffic can still occasionally exceed the threshold). The advantage of LBL is that it guarantees a minimum amount of bandwidth to low-priority traffic, and thus avoids stalling low-priority flows. However, the total traffic still shows diurnal patterns and the peak reduction is thus not as a large as with ABL.



(a) Original traffic

(b) Shaped with ABL

(c) Shaped with LBL

(d) Shaped with PBL-PEAK

Figure 4.4: Traffic in the Ohio State trace with different traffic shaping policies: Each plot shows best-effort traffic as well as the total amount of traffic (best-effort + low-priority traffic).

Finally, Figure 4.4(d) shows that PBL-PEAK is largely ineffective at reducing traffic peak. In fact, PBL-PEAK increases the traffic peak by 11% in this case. To understand this counterintuitive result, consider the following example. During the traffic shaping period (9 am to 3 pm), each low-priority flow is throttled to 250 Kbps. This small per-flow bandwidth makes it hard for low-priority flows to complete. As a result, the number of active low-priority flows increases during the traffic shaping period. At the end of the traffic shaping period all these flows are given full bandwidth again, which they promptly consume. This causes the traffic spikes that are visible in Figure 4.4(d) on each day at 3 pm, i.e., the end of the traffic shaping period. These spikes can be considerably higher than the original traffic peak. This phenomenon does not occur with PBL because traffic shaping occurs throughout the day (not shown).

4.5.2 Number of delayed low-priority flows

Since in our analysis all traffic shaping policies use the same flow size threshold, the flows that are treated as low-priority by each traffic shaping policy are the same. However, depending on the policy, some of these flows may incur only moderate delay. We regard a low-priority flow as delayed if its completion time increases by more than 5% compared to when no traffic shaping is

Policy	Flows delayed by >5%	Average peak reduction
ABL	80%	48%
PBL	71%	29%
LBL	61%	28%
ABL-PQ	51%	48%
ABL-SFF	32%	48%
PBL-PEAK	24%	-87%

Table 4.1: Fraction of low-priority flows delayed by more than 5% and average peak reduction: Among the practical policies that maximize peak reduction, ABL-PQ delays the fewest flows.

in place. Table 4.1 reports, across all traces, the fraction of low-priority flows that are delayed by more than 5% with each traffic shaping policy and the achieved average peak reduction. ABL affects the most flows, followed by PBL, which only gives 250 Kbps to each flow. Compared to ABL, ABL-SFF and ABL-PQ greatly reduce the number of delayed flows. PBL-PEAK delays very few flows because it only rate limits for 6 hours a day, but it also significantly increases the peak usage as pointed out above. Interestingly, although LBL always allocates a minimum amount of bandwidth to low-priority flows, it delays more flows than ABL-PQ and ABL-SFF, which do not provide such a guarantee. The reason is that both ABL-PQ and ABL-SFF give priority to smaller flows, thus shifting the bulk of the delay to a few large flows.

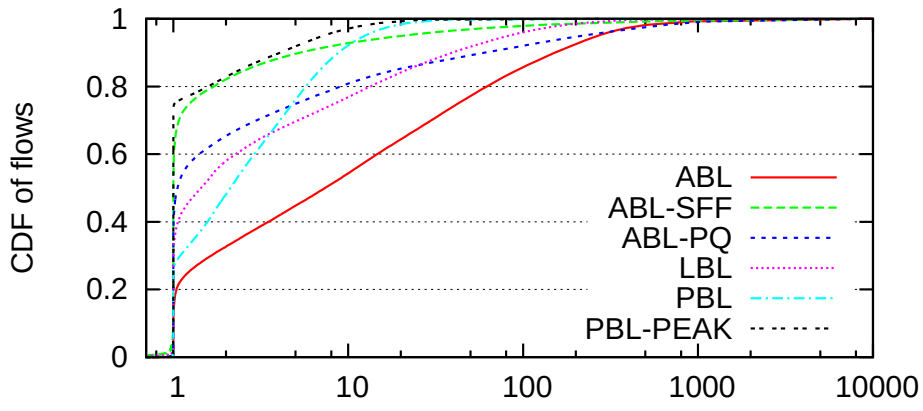
4.5.3 Delay of low-priority flows

Figure 4.5 plots the CDFs of relative and absolute delays of low-priority flows for different policies across all our experiments. ABL causes the largest delays while both ABL-SFF and PBL-PEAK lead to very low delays. However, as mentioned above, PBL-PEAK also significantly increases peak usage and has therefore little utility. With ABL, about half of low-priority flows take 10 times longer or more to complete compared to when they are not traffic shaped. With ABL-PQ, only 20% of low-priority flows take 10 times longer or more to complete. Regarding the absolute delay of flows (Figure 4.5(b)), we observed that at most 20% of low-priority flows are delayed by more than 1 hour for all policies, and almost no flow is delayed by more than 12 hours.

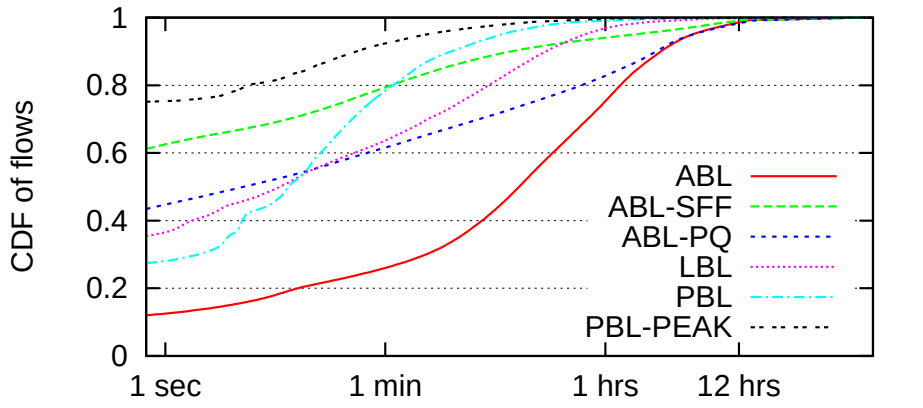
4.6 Summary

We compared the performance of 5 traffic shaping policies with respect to our goals of peak reduction, minimum number of delayed flows, and minimum

increase in completion time. We found that the ABL policies result in the best peak reduction (almost 50% in half of our traces). In addition, ABL-SFF keeps the delay incurred by low-priority flows to a minimum. However, it might not be possible to implemented ABL-SFF in practice as it requires a distinct router queue for each low-priority flow. A more practical alternative to ABL-SFF is ABL-PQ, which achieves both high peak reduction and moderate delay of low-priority flows.



(a) Relative delay



(b) Absolute delay

Figure 4.5: CDFs of relative and absolute delays for low-priority flows across all our experiments: The relative delay is the ratio of the completion time of the traffic shaped flow to its completion time with no traffic shaping. With the exception of ABL and ABL-PQ, few low-priority flows get delayed by more than 1 hours, and almost none is delayed by more than 12 hours.

5 The Global Impact of Local Traffic Shaping

In this section, we focus on the impact wide-spread deployment of traffic shaping has on the end-to-end performance of bulk flows in the Internet. As economic incentives are likely to drive ISPs to deploy traffic shapers at the boundaries of their networks, long flows may be subject to traffic shaping at multiple inter-AS links (see Figure 5.1).

Our goal is to understand how bulk transfers are affected by multiple independent traffic shapers along their paths. This is in contrast to our analysis in the previous section that analyzed the behavior of flows passing through a single traffic shaper.

For the analysis, we assume that each traffic shaper implements the ABL-PQ policy from the previous section, as this policy enables maximum peak reduction with low impact on network flows.

5.1 Analysis methodology

Our analysis is based on trace-driven simulation experiments conducted using ns-2. Figure 5.2 shows the topology we used in our analysis; it consists of two traffic shaped links connected to each other. We used our university traces to simulate the local traffic traversing each of the shapers, according to the methodology we described in Section 4.4). using the same methodology as in the previous section. In addition to the flows from the traces, we simulated a week-long bulk TCP flow that traverses both traffic shaped links. We analyzed the performance of this week-long bulk flow to understand the impact of multiple traffic shapers. We focused on a single long-running bulk flow because small flows are left largely unaffected by the ABL-PQ policy.

We also ran simulation experiments with the long running bulk flow traversing each of the traffic shaped links separately. This allows us to com-

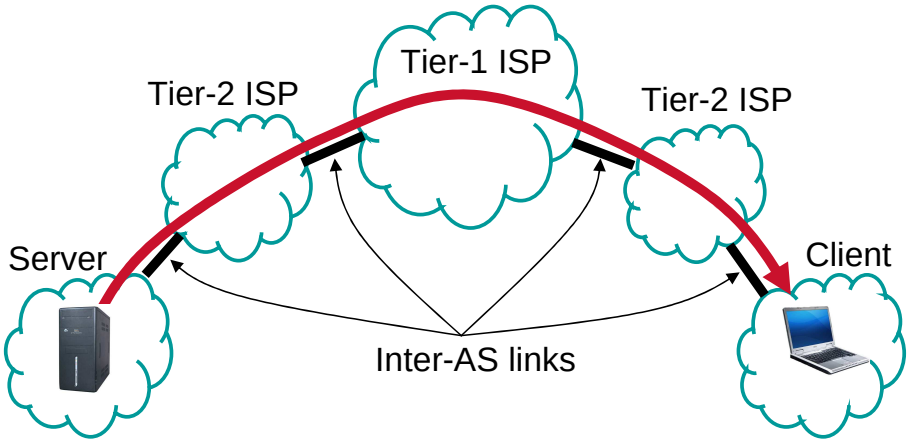


Figure 5.1: Flow traversing multiple ISPs: It is likely that a transfer between a server and a client traverses multiple traffic-shaping ISPs.

pare the flow's performance when it is crossing a pair of traffic shapers with the flow's performance when it is crossing each traffic shaper separately.

Although our long-running flow is active throughout the simulated week, we focus solely on the performance achieved from Tuesday to Thursday. The reason is that there is often sufficient available bandwidth to serve all traffic around weekends, and as a consequence our traffic shapers are mostly active during the central days of the week.

As a measure of a bulk flow's performance, we count the number of bytes the bulk flow was able to send from Tuesday to Thursday. To quantify the impact of multiple traffic shapers on a flow, we define a metric called *end-to-end performance loss*. End-to-end performance loss is defined as the relative decrease in performance of a bulk flow traversing multiple traffic shapers compared to the minimum performance the bulk flow achieves when it traverses either of the two traffic shapers separately. More formally, consider a flow that transfers B_1 and B_2 bytes when it separately traverses traffic shapers S_1 and S_2, respectively. If the same flow transfers G bytes when it simultaneously traverses S_1 and S_2, the end-to-end performance loss of the flow is: $(min(B_1, B_2) - G)/min(B_1, B_2)$.

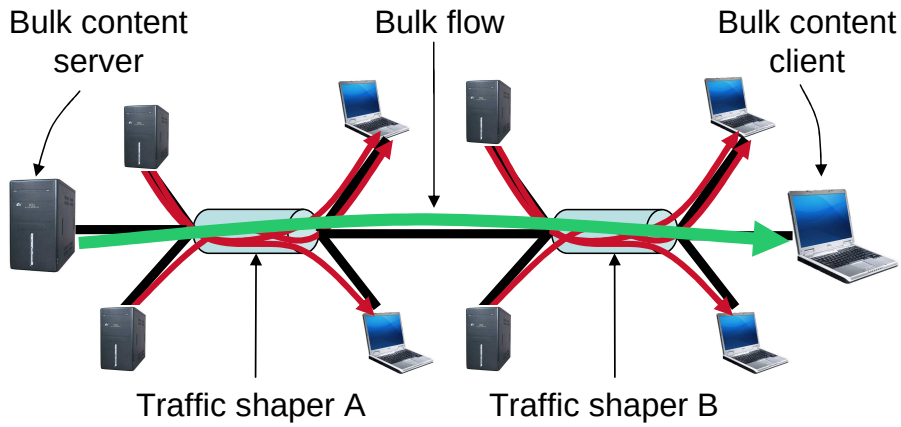


Figure 5.2: Simulation topology for analyzing the performance of a flow passing two traffic shapers: A long-running bulk TCP flow transfers data from the server to the client traversing two traffic shapers that act independently.

5.2 The impact of multiple traffic shapers on end-to-end performance

To study the effects of multiple traffic shapers, we used traces from 15 of our 35 university access links. We simulated traffic shaping on these links and analyzed the performance of bulk flows over the all possible (105) pairings of the 15 traffic shaped links. The universities are spread across four time zones in the US. When replaying the traces in simulation we adjusted for the differences in local time by time shifting all traces to the Eastern Standard Time. We discuss the impact of the differences in local time zones of traffic shapers in the next section.

Figure 5.4(a) shows the end-to-end performance loss experienced by flows traversing pairs of traffic shaped links relative to their performance when they cross each of the traffic shaped links individually. Even when the traffic shapers are in the same time zone, the loss in end-to-end performance is significant. In almost 80% of the cases, flows crossing two shapers sent 40% less data than what they would have sent over either of the shapers independently. In 50% of the pairings, the loss in performance is larger than 60%. While we do not show the data here, the performance continues to slide dramatically for each additional traffic shaper.

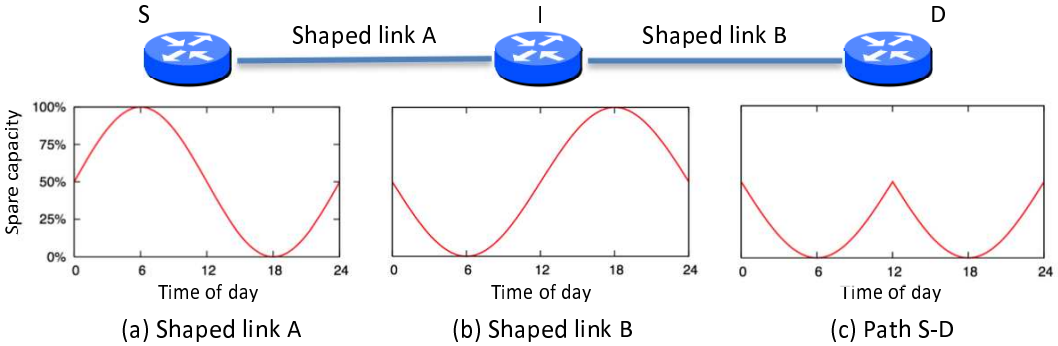


Figure 5.3: Transfer from a source S to a destination D through two shaped links: at any time, transfers are limited to using the end-to-end bandwidth available along the $S - D$ path, that is, the minimum of bandwidths available on links A and B

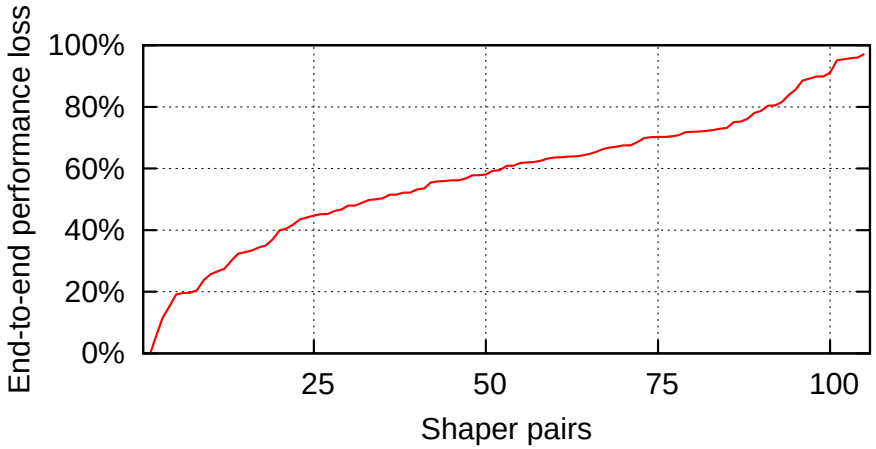
5.2.1 Factors affecting end-to-end performance

The considerable performance loss for flows traversing multiple traffic shapers can be mainly attributed to two factors. First, at any time t of the simulation, a flow traversing two traffic shapers S_1 and S_2 is limited to using only the *minimum bandwidth available at either traffic shaper at time t*. However, the sum of these minima over the entire simulation time can be lower than the total bandwidth available at either of the two traffic shapers during the same time. More formally, if T is the total simulation time and B_i^t is the bandwidth available at time t at traffic shaper S_i, then $\sum_{t=1}^{T} B_i^t$ is the total bandwidth available at each traffic shaper S_i during T. The first limiting factor can therefore be written as:

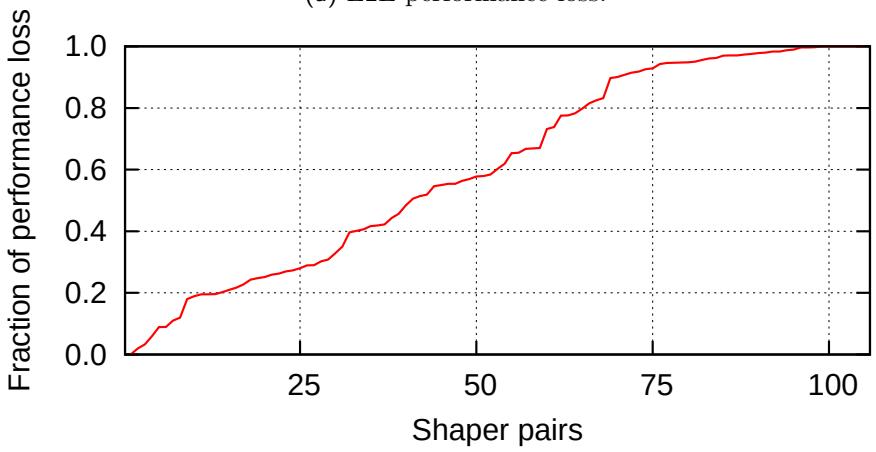
$$\sum_{t=1}^{T} min(B_1^t, B_2^t) \leq min(\sum_{t=1}^{T} B_1^t, \sum_{t=1}^{T} B_2^t) \tag{5.1}$$

We refer to the loss in performance due to this factor as the *loss due to offsets in shapers' available bandwidth*. This loss is visually explained in Figure 5.3, which shows a network path that traverses two shaped links. Figures 5.3(a) and (b) show how the bandwidth available at each shaper varies over time. A flow that traverses both shapers is limited by the minimum bandwidth available at either shaper, which is shown in Figure 5.3(c). As a consequence, the flow can use only a fraction of the bandwidth available when it traverses either traffic shaper in isolation.

The second limiting factor is that TCP congestion control may prevent the flow from fully using the bandwidth available at any time t. Because each traffic shaper is throttling bulk flows independent of other shapers, multiple



(a) E2E performance loss.



(b) Fraction of loss due to offsets in the shapers' available bandwidth.

Figure 5.4: The impact of two traffic shapers on the path: The E2E performance loss is significant in most cases (a). In only a few cases can the loss be entirely attributed to offsets in the shapers' available bandwidth (b).

traffic shapers can lead to multiple congested (lossy) links along an Internet path. A long TCP flow traversing two or more congested shapers would be at a serious disadvantage when it competes for bandwidth against shorter flows traversing only a single shaper. Prior studies [23] have shown that multiple congested gateways can lead to an additional drop in the end-to-end performance of a TCP flow. We refer to this performance loss as the *loss due to TCP behavior*.

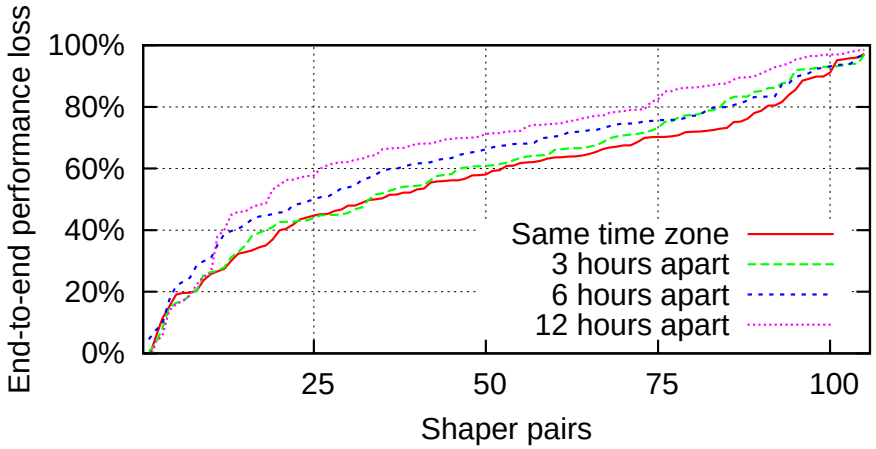
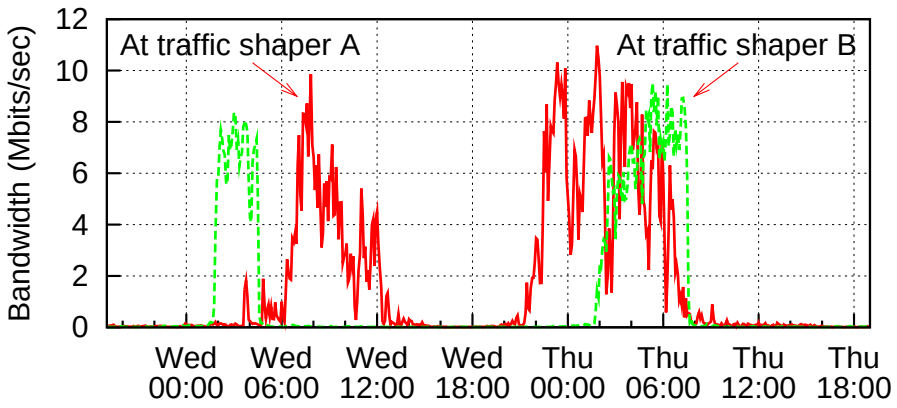


Figure 5.5: Impact of time zone offset on performance: Compared to two traffic shapers located in the same time zone, topologies with 3, 6, and 12 hours time difference in the location of the shapers loose performance only moderately.

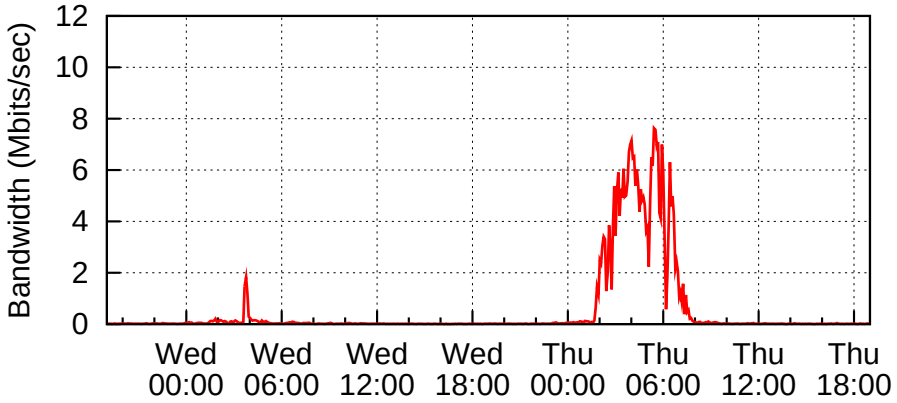
5.2.2 Estimating the impact of the factors

Next we investigate the role of the two factors namely, *loss due to offsets in shapers' available bandwidth* and *loss due to TCP behavior*, in the drop in end-to-end performance of bulk flows. To estimate the performance drop due to offsets in available bandwidth, we do the following: for each pair of traffic shapers, we compute the number of bytes that could have been transferred by a hypothetical flow that fully uses the minimum bandwidth available at either of the shapers at all times during the simulation. In other words, the performance of the hypothetical flow is not affected by the second factor, i.e., TCP behavior when crossing multiple congested shapers.

Figure 5.4(b) shows the fraction of end-to-end performance loss that can be attributed to offsets in the shapers' available bandwidth. The bandwidth offset between shapers accounts for the entire performance loss only in a few cases, while in many cases it accounts for less than half of the performance loss. We attribute the rest to the penalty TCP flows suffer when traversing multiple bottlenecks. Simultaneously competing with other flows at multiple congested links takes a heavy toll on the overall performance of a TCP flow, and traversing multiple traffic shapers makes this scenario very likely.

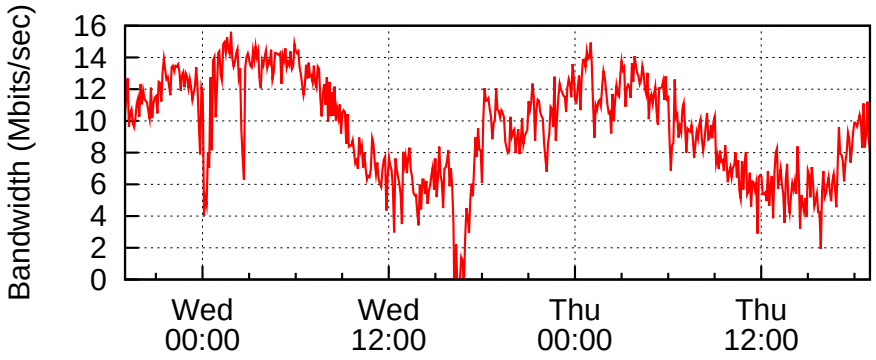


(a) Bandwidth achieved by a bulk flow when it traverses A and B separately.

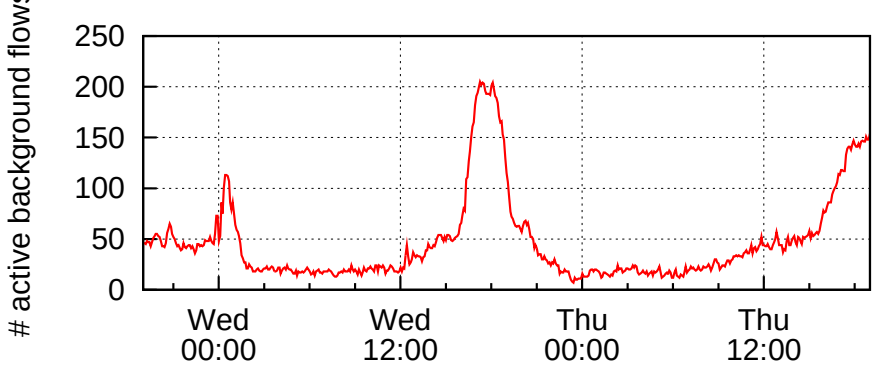


(b) Available bandwidth when the bulk flow traverses both A and B.

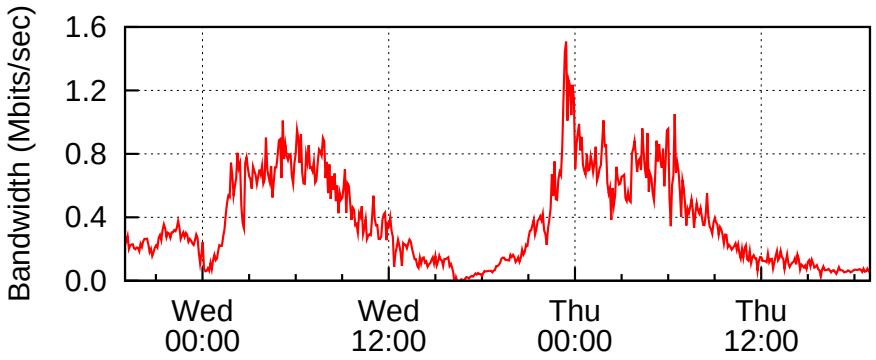
Figure 5.6: TCP transfer is extremely bursty: When a flow traverses one traffic shaper, most bytes are transferred within a small amount of time, i.e., within the peak periods in (a). When a flow traverses multiple shapers, and the peak periods across these shapers do not overlap nicely, the flow incurs a high loss in E2E performance (b).



(a) Bandwidth available to all low-priority flows.



(b) Number of active low-priority flows.



(c) Bandwidth available per flow.

Figure 5.7: Traffic shaping can lead to very bursty transfers: The drops in available bandwidth to low-priority flows over time (a) cause the number of active TCP flows to increase when bandwidth is scarce (b), leading to much sharper peak to trough ratios in the per-flow available bandwidth (c).

Time zone difference	Median total perf. loss	Avg. total perf. loss
0 hrs	60%	58%
3 hrs	62%	60%
6 hrs	68%	64%
12 hrs	72%	69%

Table 5.1: Median and average loss in performance for different time zone offsets: Compared to the loss in performance caused by adding a second traffic shaper to the path, increasing the time difference of this shapers increases the loss only moderately.

5.3 Traffic shaping impact across time zones

If multiple traffic shapers are in the same time zone they also share similar night and day cycles. However, if they are many time zones apart from each other, their night and day time cycles will be out of phase. This can cause a severe decrease in the end-to-end performance of passing bulk flows. In the previous section, we found that the end-to-end performance drops significantly even when the shapers are in the same time zone. Next, we investigate whether we will see additional loss in performance when these traffic shapers are separated by additional time zones. We repeat the simulations with two traffic shapers, but vary the time zone difference between the two shapers by time-shifting the traces before they are replayed.

In Figure 5.5, we plot the end-to-end performance loss for simulations where the two shapers are in the same time zone and for simulations where the two shapers are 3, 6, and 12 hours apart. We show the average performance loss in Table 5.1. Strikingly, while the performance loss increases with the time zone difference between traffic shapers, the additional loss is rather small compared to the performance loss incurred when the two shapers are in the same time zone. While two shapers in the same time zone result on average in 58% loss in performance, spacing them by 3 hours decreases performance only by an additional 2%. A time shift of 12 hours results in an average total performance loss of 69%.

To understand why most performance loss is suffered when two traffic shapers are in the same time zone, we took a closer look at the bandwidth achieved by low-priority flows when they traverse only one traffic shaper. Figure 5.6(a) plots the bandwidth achieved by a long bulk flow over the course of two days when it traverses two shapers A and B in isolation. Interestingly, the flow exhibits diurnal patterns with a very high peak-to-trough ratio; it reaches a very high peak throughput, but only for a short time be-

tween midnight and early morning. In fact, we found that more than 90% of all bytes are transferred during less than 10% of the flow's total duration. Thus, when a flow crosses both traffic shapers A and B, even a marginal misalignment in the peak throughput periods of A and B can lead to a dramatic drop in end-to-end throughput. We show this in Figure 5.6(b), which plots the bandwidth available on a path traversing both shapers A and B. Such small misalignments in the peak throughput periods can occur even when shapers are in the same time zone. This explains why time zone differences between traffic shapers result in a relatively small additional loss in end-to-end performance.

The extreme diurnal patterns exhibited by a single traffic shaped bulk flow stands in contrast to the more gentle diurnal patterns exhibited by the aggregate bandwidth available to all bulk flows. We explain the reasons for the difference in Figure 5.7. Figure 5.7(a) plots the total bandwidth available to all low-priority flows at traffic shaper A over time. The peak-to-average ratio in available bandwidth is approximately two, consistent with our observations in Section 3.2. Figure 5.7(b) plots the number of active low-priority flows that compete for this bandwidth. The number of active flows increases sharply at times when the available bandwidth is low, because new flows arrive at the traffic shaper and existing ones don't complete due to lack of bandwidth. The number of active flows decreases sharply again when more bandwidth becomes available, resulting in very pronounced diurnal patterns in the number of active flows. Figure 5.7(c) plots the fair share of bandwidth for each bulk flow, which is obtained by dividing the aggregate available bandwidth by the number of active flows at any point in time. The per-flow bandwidth exhibits considerably sharper diurnal patterns than the aggregate bandwidth due to the variation in number of active flows over time. This explains why traffic shaped flows transfer most of their bytes during a short window of time in which they achieve their peak throughput.

5.4 Summary

In this section, we identified two main factors that affect the performance of long bulk flows traversing multiple traffic shapers: *loss in end-to-end bandwidth* and *bias of TCP against connections traversing multiple congested links*. First, we found that a long bulk flow traversing two traffic shapers suffers a considerable loss in performance, and in many cases the expected loss in available end-to-end bandwidth is not enough to warrant such a high loss. In these cases, most of the loss in performance comes from the unfairness suffered by a TCP flow when it traverses multiple congested links.

Second, we found that there is no large additional loss when the two traffic shapers are located in different time zones. The reason for this is that a long bulk flow traversing a single traffic shaper transfers *most of its data in a short time window*. Thus, when a flow traverses multiple traffic shapers, its end-to-end performance depends on how well the time windows at each shaper overlap. However, because these time windows are short, there is a high chance that they poorly overlap even when the traffic shapers are in the same time zone.

6 Improving the Performance of Bulk Transfers with Staging

In the previous sections, we showed that ISPs have an incentive to selfishly traffic shape interdomain bulk transfers to reduce their transit costs. However, as more and more ISPs do that, the end-to-end performace of bulk transfers is likely to decrease dramatically, directly affecting the ISPs' customers. In this section, we investigate whether it is possible to avoid such a tragedy of the commons, without restricting ISPs from deploying locally-optimal traffic shaping policies.

6.1 Isolating the effects of local traffic shaping with staging

The root cause of the global slowdown of bulk transfers is the harmful interaction between traffic shapers local to different ISPs. Individually, each local traffic shaper affects bulk flows only minimally. But, taken together, multiple traffic shapers along an end-to-end path inflict substantial performance penalty.

To prevent traffic shapers at different links along a path from interfering with one another, we propose to *stage* bulk transfers. By staging we refer to breaking up an end-to-end transfer along a path into a series of sub-transfers along segments of the path, where each path segment contains only one traffic shaper (see Figure 6.1). The end points of each sub-transfer maintain their own congestion control state, thus isolating traffic shaping within one path segment from affecting the transfers in other segments.

When transfers along different segments are decoupled, data might arrive at a router connecting two successive segments faster along the upstream segment than it can be sent along the downstream segment. In such cases, we need to temporarily store data at the intermediate router connecting the two

segments. The buffered data would be drained at a later point in time when there is more available bandwidth on the downstream segment than upstream segment (see Figure 5.3). Thus, staging needs storage at intermediate points to exploit the bandwidth available on each of the upstream and downstream path segments separately and efficiently. In contrast, end-to-end transfers are limited to the minimum bandwidth available across all the path segments at all times.

The amount of storage available at intermediate points crucially determines how effectively staging works. If there is too little storage, it would be hard to overcome large offsets in the times when bandwidth is available across different path segments. Once a router runs out of storage, the transfer on its upstream path segment gets throttled down to the bandwidth available on its downstream path segment, and staging yields no further benefits. On the other hand, adding more storage beyond a certain limit is wasteful and does not improve the performance of the transfers. Our evaluation in section 6.3 quantifies the benefits of staging as a function of storage deployed.

6.2 Design alternatives

The basic idea behind staging – splitting an end-to-end transport connection into a series of cascading transport connections – has been previously used in other contexts, such as caching web content with proxy servers [5, 22, 42], and improving TCP performance over paths with wireless or satellite links as their last hop [7, 12, 28]. Tremendous research and development have gone into addressing transport layer issues (e.g., end-to-end reliability) that arise when implementing staged transfers [32]. Rather than reinvent the wheel here, we present a high-level overview of the design alternatives for staging and cite prior work for the details of the design. However, we do discuss the tradeoffs between the designs in terms of their applicability to bulk transfers, their deployment barriers and their deployment incentives.

6.2.1 Proxy server based staging

Our first design involves using popular HTTP proxies [5, 42] for staging bulk content transfers. When a client wants to download bulk content from a server, it simply establishes a transport connection (e.g., TCP) to a proxy and requests content from it. The proxy in turn connects to the server, fetches the content, and forwards it to the client. Thus, the data transfer is staged at the proxy. Note that proxy server itself can connect to another upstream proxy server and establish a chain of inter-proxy transport connections before

eventually connecting to the content server. In this case, the bulk transfer would be staged at each of the multiple proxy servers along the path.

For the design to work efficiently, one would have to both deploy proxies at key locations in the Internet and select the proxies such that each segment of the staged transfer contains only one traffic shaper. The proxies could be deployed by the ISPs themselves or by content delivery networks like Akamai [1]. Transit ISPs might be incentivized to deploy such proxies because they are inexpensive and offer significant performance benefits to their customers (see Section 6.3). Further, there are incremental benefits even in a partial deployment scenario; when a large transit ISP deploys one or more proxies within its backbone network, it immediately benefits transfers between any two of its traffic shaping customers. On the other hand, CDNs like Akamai might be able to leverage their distributed caches world-wide to offer staging service for end users wishing to speed their bulk downloads at a price.

One disadvantage with the proxy-based approach is that it is non-transparent; clients need to be configured with the address of their upstream proxy. Another potential problem is that only bulk transfers conducted using the HTTP protocol can be staged. This might not be a serious limitation in the Internet today as a majority of content transfers work over HTTP [21,25].

6.2.2 Split-TCP based staging

Our second design is inspired by the Split-TCP designs that are deployed by satellite or cellular broadband ISPs on their last hop [7, 12]. To implement Split-TCP, ISPs deploy boxes (sometimes referred to as Performance Enhancing Proxies or PEPs [37]) that split TCP connections along a path by intercepting data packets from the sender and impersonating the receiver by ACKing the receipt of the packets even before forwarding the packets to the receiver. Simultaneously, the boxes impersonate the sender by forwarding the data packets to the receiver with spoofed source address. Effectively, the bulk transfer is staged at the Split-TCP box.

To stage transfers with Split-TCP, ISPs need to deploy Split-TCP boxes at some intermediate point along the paths taken by the transfers. A transit ISP could deploy such boxes at its customers' access routers, where they can intercept and split all bulk flows to and from its customers.

Compared to the HTTP proxy based approach, Split TCP has two primary advantages. First, it is transparent to end hosts; clients do not need to be configured with addresses of Split-TCP boxes as they are deployed along the network paths by ISPs and intercept the packets automatically. Second, because Split-TCP operates at the transport layer, it works with

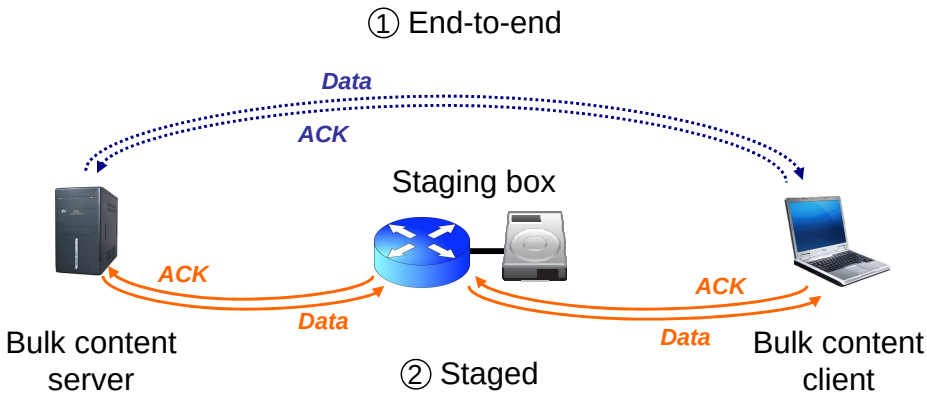


Figure 6.1: Simulation topology used to evaluate the staging service: A staging box is deployed on the network path between the 2 traffic shapers, breaking the bulk transfer into two independent subtransfers.

a wider variety of bulk flows, including those that do not use the HTTP protocol. However, Split-TCP is known to break the end-to-end semantics of TCP (e.g., end-to-end reliability), and there has been significant work and several RFCs devoted to analyzing the resulting risks and potential fixes [8, 11, 37]. More recently, a number of research efforts have developed variants of Split-TCP that maintain end-to-end semantics [32]. However, they require modifications to the TCP implementations at the end hosts.

6.2.3 Summary

Our discussion above suggests that there are many alternative staging designs that could be deployed. Some of the designs are transparent, while others are not. Some can be deployed only by ISPs, while others could be deployed by CDNs like Akamai as well. However, two key questions remain unanswered about all of these designs. First, how effective is staging at restoring the performance of bulk transfers? Second, how much storage does staging need? We answer these questions below.

6.3 Evaluation

To understand the performance benefits from staging transfers and to estimate the storage staged transfers would need, we implemented and analyzed a simple proxy based staging service design in the ns-2 simulator. We evaluate the staging service using the network topology and methodology from Section 5. We still focus on the performance of a single bulk flow traversing

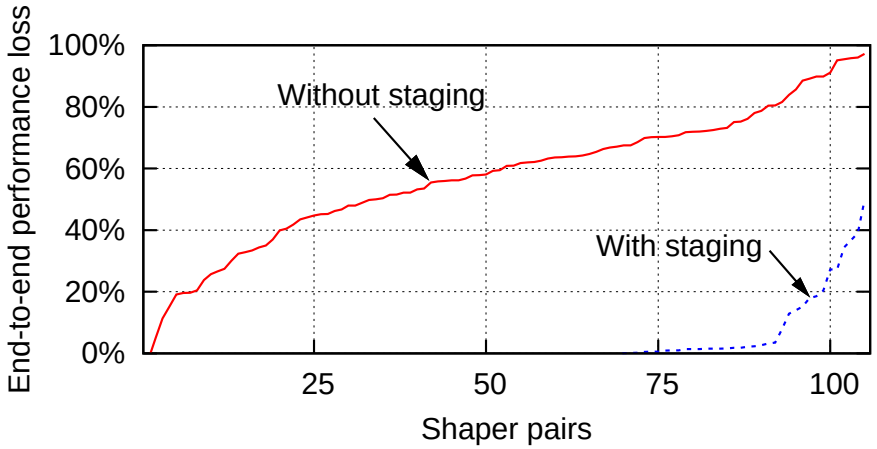


Figure 6.2: Performance loss with and without the staging service: In roughly 90% of the cases, deploying the staging service recovers the full bulk transfer performance.

a pair of traffic shapers. However, we now place a staging proxy on the network path between the two traffic shapers, as shown in Figure 6.1. The proxy breaks the bulk transfer into two independent subtransfers by fetching data traversing the first traffic shaper into a local store before sending it across the second traffic shaper.

6.3.1 Does staging improve performace of end-to-end transfers?

We first analyze simulation results from the idealized scenario when there is unlimited amount of storage at the staging proxy. The performance of transfers under this scenario represents an upper bound on the potential benefits from staging. Figure 6.2 compares the loss in performance suffered by our long-running flow traversing 2 traffic shapers with and without the staging service. The performance loss is computed relative to the minimum performance of the flow when it is traversing either of the two traffic shapers individually and it is computed in terms of the number of bytes the flow transfers during our simulation. The figure shows that staged transfers perform significantly better than end-to-end transfers without staging. In fact, with staging, we see no performance loss when traversing multiple traffic shapers in almost 90% of the cases. This suggests that a staging service could be very effective in counteracting the harmful global slowdown in the performance of bulk flows.

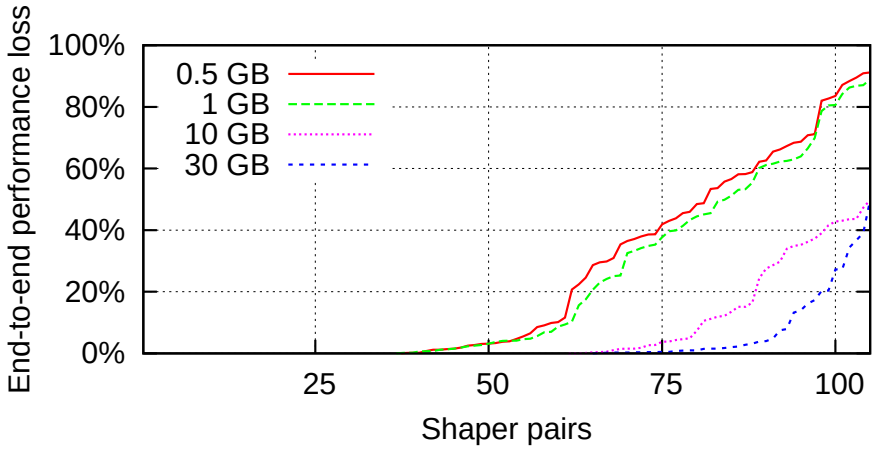


Figure 6.3: Performance loss for the staging service with different maximum amounts of per-flow storage: Reducing the available storage causes the staging service to become less effective.

6.3.2 How much storage is needed?

In practice, only a limited amount of storage will be available at a staging proxy. So we repeated the experiments limiting the amount of data our bulk flow can buffer at the staging box. Figure 6.3 shows how our long-running bulk flow's performance varies with different storage limits. As expected, the transfer performance improves when more storage is available. However, the performance benefits are considerable, even when only a small amount of storage is available for staging. When we allocate 30GB to the bulk flow, the performance approaches the optimal performance we observed with unlimited storage (see Figure 6.2).

Our results suggest that the amount of storage that we would have to allocate for each bulk flow, while not extremely large, is considerable. However, one key question remains: how much aggregate storage does one have to deploy for all bulk flows crossing an access link? It is not possible to answer this question directly from our simulations as we have only one staged bulk flow traversing both the traffic shapers simultaneously. However, we can derive an estimate of the aggregate storage required for all flows as follows: we first multiply the storage consumed by our bulk flow with the number of bulk flows that are actively traffic shaped at different times of the day and then compute the maximum storage that would be required at any time during the course of the day. Figure 6.4 plots the results of this estimate. In 50% percent of the cases, the aggregate storage required for staging all bulk flows at university access links is less than 1TB. In 97% of the cases, the stor-

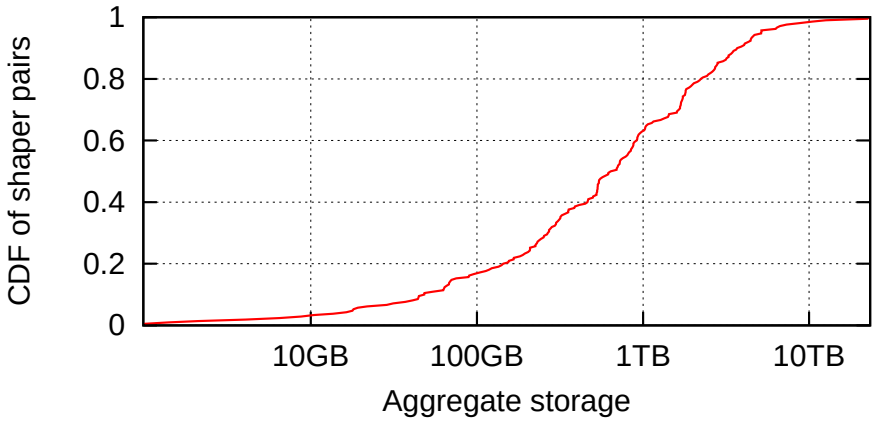


Figure 6.4: Estimate of the aggregate storage required for the staging service: The total amount of storage required very rarely exceeds 10TB.

age required is less than 10TB. In general, we believe that the performance benefits available from staging justify the requisite storage.

7 Related work

Traffic shaping already plays an important role in ISPs strategies to reduce network congestion and bandwidth costs. For example, some major ISPs were found to block BitTorrent traffic in their networks [19], and other ISPs are known to throttle bandwidth-intensive applications or users that consume a disproportional large fraction of bandwidth [9, 17, 24].

Today's networking equipment enables ISPs to deploy even complex traffic shaping policies at line speeds [15, 38]. This equipment typically supports deep packet inspection, which allows ISPs to identify the traffic of particular applications, and a wide range of traffic shaping and queue management techniques, including token buckets and priority queueing. The queueing and shaping techniques provided by this equipment – and also used in this paper – were introduced in the context of quality-of-service and DiffServ [10]. They were originally developed to provide flow delay guarantees that are better than best-effort. Today, they are still used to give higher precedence to some traffic (e.g., for voice-over-IP traffic), but also to throttle the bandwidth usage of some applications (e.g., file sharing applications).

There is a large body of literature on splitting TCP connections. Typically, these papers focus on improving the performance of TCP connections over wireless (including ad-hoc and cellular) links [7, 12, 32] or over high-latency connections such as satellite links [28]. Unlike our approach, they do not stage large amounts of data in the network, but buffer a few packets to gracefully recover from occasional packet loss. Another very popular example of splitting end-to-end connections are the widely used web proxy caches [42] aiming at faster download of web content.

To the best of our knowledge, our paper is the first study that characterizes the local and global effects of traffic shaping in the Internet. The only related work we are aware of is from Laoutaris et al. [33], who quantified how much additional "delay tolerant" data (i.e., data that can tolerate delivery delays of hours or days) ISPs could send for free by exploiting 95^{th} percentile billing and diurnal patterns in today's Internet traffic. To achieve

this, they present and evaluate simple end-to-end scheduling policies as well as "store-and-forward" techniques that use storage deployed in the network. They show that it is possible to transfer multiple TBytes during off-peak times with no additional costs.

There are three main differences between our work and theirs. First, while [33] aims to send additional (delay-tolerant) data without increasing bandwidth costs for ISPs[1], our work reduces the peak bandwidth usage of ISPs for today's traffic with only moderate impact (i.e., delay) on shaped flows.

Second, the approach presented by Laoutaris et al. requires fine-grained and real-time information about the load of the network for scheduling decisions, and a transport layer that is capable of instantaneously using all available spare bandwidth for the delay tolerant traffic. On the contrary, our traffic shaping policies can be deployed on today's networking equipment.

Third, while the analysis in [33] uses data that comprises only aggregate network loads, we use flow-level NetFlow traces that enable us to study the behavior of single TCP flows and perform a more detailed and realistic analysis. Thanks to this detailed analysis we could identify global effects of traffic shaping that are related to TCP characteristics and would have escaped an analysis based on traffic aggregates only.

[1]By increasing the average bandwidth usage to nearly the peak usage.

8 Conclusions

We conducted a systematic analysis of traffic shaping. Even though traffic shaping is widely deployed today, most deployed techniques are ad-hoc, without a clear understanding of their effect on network traffic.

We compared different traffic shaping policies inspired by real-world examples. We found a local traffic shaping policy that greatly reduce peak network traffic while minimizing the impact on the performance of the shaped flows. However, we also found that multiple of these traffic shapers in the path of a bulk flow can have a significant impact on its performance. To counteract this negative global effect, we propose staging, i.e., breaking end-to-end connections at multiple points in the network. The staging points need storage to temporarily buffer the data of bulk flows in transit. Our evaluation shows that staging is effective at restoring the performance of traffic shaped bulk transfers and requires only a reasonable amount of storage.

Bibliography

[1] Akamai technologies. http://www.akamai.com/.

[2] Abilene Backbone Network. http://abilene.internet2.edu.

[3] S. Agarwal, A. Nucci, and S. Bhattacharrya. Measuring the Shared Fate of IGP Engineering and Interdomain Traffic. In *Proc. of IEEE ICNP*, 2005.

[4] Amazon Simple Storage Service. http://aws.amazon.com/s3/.

[5] Apache http server. http://httpd.apache.org/.

[6] N. B. Azzouna and F. Guillemin. Analysis of ADSL Traffic on an IP Backbone Link. In *Proc. of IEEE Global Telecommunications Conference*, 2003.

[7] A. Bakre and B. R. Badrinath. I-tcp: Indirect tcp for mobile hosts. In *Proc. of International Conference on Distributed Computing Systems*, 1995.

[8] H. Balakrishnan, V. N. Padmanabhan, and G. F. et al. TCP Performance Implications of Network Path Asymmetry, 2002.

[9] Comments of Bell Aliant Regional Communications, Limited Partnership and Bell Canada. http://www.crtc.gc.ca/PartVII/eng/2008/8646/c12_200815400.htm#2b.

[10] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Service. RFC 2475 (Informational), Dec. 1998. Updated by RFC 3260.

[11] J. Border, M. Kojo, and J. G. et al. Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations, 2001.

[12] K. Brown and S. Singh. M-tcp: Tcp for mobile cellular networks. *SIGCOMM Comput. Commun. Rev.*, 27(5):19–43, 1997.

[13] N. Brownlee and kc claffy. Understanding Internet Traffic Streams: Dragonflies and Tortoises. *IEEE Communications Magazine*, Oct 2002.

[14] K. Cho, K. Fukuda, H. Esaki, and A. Kato. The Impact and Implications of the Growth in Residential User-to-user Traffic. In *Proc. of SIGCOMM'06*, 2006.

[15] Cisco IOS Classification. http://www.cisco.com/en/US/docs/ios/12_2/qos/configuration/guide/qcfclass.html.

[16] Cisco IOS Congestion Management. http://www.cisco.com/en/US/docs/ios/12_2/qos/configuration/guide/qcfconmg_ps1835_TSD_Products_Configuration_Guide_Chapter.html.

[17] Comcast: Description of planned network management practices. http://downloads.comcast.net/docs/Attachment_B_Future_Practices.pdf.

[18] Deutsche Telekom AG. Softwareload. http://www.softwareload.com.

[19] M. Dischinger, A. Mislove, A. Haeberlen, and K. P. Gummadi. Detecting BitTorrent Blocking. In *Proc. of IMC*, 2008.

[20] P. Elmer-DeWitt. iTunes store: 5 billion songs; 50,000 movies per day, June 19th, 2008. http://tech.fortune.cnn.com/2008/06/19/itunes-store-5-billion-songs-50000-movies-per-day.

[21] Fasttrack p2p network.

[22] R. Fielding, J. Gettys, and J. M. et al. Hypertext Transfer Protocol – HTTP/1.1, 1999.

[23] S. Floyd. Connections with multiple congested gateways in packet-switched networks part 1: One-way traffic. *ACM Computer Communication Review*, 21:30–47, 1991.

[24] S. Friederich. Bandwidth restrictions save almost $1 million, Oct 22nd, 2002. http://thedaily.washington.edu/2002/10/22/bandwidth-restrictions-save-almost-1-million/.

[25] Gnutella p2p network.

[26] K. P. Gummadi, S. Saroiu, and S. D. Gribble. King: estimating latency between arbitrary internet end hosts. In *Proc. of IMW '02*, Marseille, France, 2002.

[27] L. Guo and I. Mitta. Scheduling flows with unknown sizes: An approximate analysis. In *Proc. of SIGMETRICS*, 2002.

[28] T. R. Henderson and R. H. Katz. Transport Protocols for Internet-Compatible Satellite Networks. *IEEE Journal on Selected Areas in Communications*, 17:326–344, 1999.

[29] ipoque GmbH. OpenDPI, 2009. http://www.opendpi.org.

[30] T. Karagiannis, A. Broido, N. Brownlee, K. Claffy, and M. Faloutsos. Is P2P Dying or just Hiding? In *Proc. of IEEE Globecom*, 2004.

[31] T. Karagiannis, A. Broido, M. Faloutsos, and K. claffy. Transport Layer Identification of P2P Traffic. In *Proc. of IMC*, 2004.

[32] S. Kopparty, S. V. Krishnamurthy, M. Faloutsos, and S. K. Tripathi. Split tcp for mobile ad-hoc networks. In *Proc. of GLOBECOM*, 2002.

[33] N. Laoutaris, G. Smaragdakis, P. Rodriguez, and R. Sundaram. Delay Tolerant Bulk Data Transfers in the Internet. In *Proc. of SIGMETRICS*, 2009.

[34] Linux Advanced Routing & Traffic Control HOWTO. http://lartc.org/lartc.html.

[35] G. Maier, A. Feldmann, V. Paxson, and M. Allman. On dominant characteristics of residential broadband internet traffic. In *Proc. of IMC*, 2009.

[36] I. Matta and L. Guo. Differentiated predictive fair service for tcp flows. In *Proc. of ICNP*, 2000.

[37] G. Montenegro, S. Dawkins, and M. K. et al. Long Thin Networks, 2000.

[38] Blue Coat PacketShaper. `http://bluecoat.com/products/packetshaper`.

[39] J. Röttgers. Internetanbieter bremst Tauschbörsen aus. Focus Online, Mar 6th, 2008. `http://www.focus.de/digital/internet/kabel-deutschland_aid_264070.html`.

[40] A. Shaikh, J. Rexford, and K. G. Shin. Load-Sensitive Routing of Long-Lived IP Flows. In *Proc. of SIGCOMM*, 1999.

[41] S. Shalunov and B. Teitelbaum. TCP Use and Performance on Internet2. In *Proc. of ACM SIGCOMM Internet Measurement Workshop*, 2001.

[42] Squid proxy cache. `http://www.squid-cache.org`.

[43] R. Topolski. Comcast is using Sandvine to manage P2P connections, May 2007. `http://www.dslreports.com/forum/r18323368-Comcast-is-using-Sandvine-to-manage-P2P-Connections`.

[44] Universities Prepare for Data Deluge from CERN Collider, May 2007. `http://www.hpcwire.com/hpc/1572567.html`.

[45] Valve Corp. Steam. `http://store.steampowered.com`.