The Local and Global Effects of Traffic Shaping

Massimiliano Marcon
Marcel Dischinger
Krishna Gummadi
Amin Vahdat

Technical Report
MPI–SWS–2008–001
October 2008

Abstract

The Internet is witnessing explosive growth in traffic due to bulk content transfers, such as multimedia and software downloads, and online sharing and backups of personal, commercial, and scientific data. Wide-area network bandwidth is expensive and this cost is forcing many ISPs to deploy middle boxes to contain bulk traffic. As a result, many Internet bottlenecks today are *economic* rather than *physical*. That is, for many links interconnecting distinct organizations, there is often plenty of available physical capacity. However, the cost of actually transmitting across these link is based on peak levels of utilization, for instance as measured by the 95% link utilization. Thus, there are incentives to perform traffic shaping across these links to limit peak levels of utilization.

In light of these trends, this paper makes the following contributions. We show that appropriate inter-ISP traffic shaping mechanisms can dramatically reduce peak levels of utilization with no impact on interactive applications and only minimal degradation of bulk data transfers. This suggests that in the future, it is in the self-interest of every ISP to perform such traffic shaping at the edges of its network. Unfortunately, we show that the local incentives to perform traffic shaping will result in dramatically negative global slowdown of bulk transfers, with the degradation growing as a function of the distance between the source and destination. Essentially, geographic time zone differences and the associated offsets in the local times of peak utilization mean that bulk transfers traveling sufficiently far will always be throttled by some ISP between the source and destination. Our findings suggest that once traffic shaping becomes predominant, alternative data transfer mechanisms will be needed to efficiently deliver bulk data across the Internet.

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 [4], distribution of large software and games [9, 38], online backups of personal and commercial data [3], and sharing of huge scientific data repositories [36]. Recent studies of Internet traffic in commercial and research backbones [5, 24, 34] as well as academic [11] and residential [12] access networks show that such bulk transfers account for a large and rapidly growing fraction of bytes transferred across the Internet.

The bulk data traffic in the Internet today represents just the tip of the iceberg. Tremendous amounts of digital data are being delivered outside of the Internet, for example using hard drives, optical media, or tapes [20, 21, 28], because it is cheaper and faster–though usually not more convenient or secure–than using the Internet. On an average day, Netflix, ships 1.6 million movie DVDs [28], or 6 petabytes of data. This is more than the estimated traffic exchanged between ISPs in the U.S. [30]. It is debatable whether the Internet can ever match the capacity of postal networks. However, the convenience of online transfers will likely drive the demand to deliver more bulk data over the Internet in the foreseeable future.

Internet bulk data transfers are expensive. A recent study [23] reported that ISPs (or CDNs) charge large content providers, such as YouTube and MSN Live, 0.1 to 1.0 cent per minute for a 200-400 kbps data stream. Higher bandwidth streams will cost more. The high cost of wide-area network traffic means that increasingly *economic* rather than *physical* constraints limit the performance of many Internet paths. That is, even when there is plenty of physical capacity available on a given link, ISP policies of charging customers based on peak bandwidth utilization (often measured by the 95%-ile over some time period) result in strong disincentives to approach the full physical capacity of inter-AS links.

While decades of research in congestion control shows how to manage transfers across physical bottlenecks, there is little understanding of how to manage transfers across economic bottlenecks. Instead, ISPs have developed a variety of ad hoc traffic shaping techniques to control bandwidth costs (see Section 1.1). This traffic shaping specifically targets bulk transfers because they consume the vast majority of bytes. Unfortunately, the policies themselves are often blunt and arbitrary, often shutting down entire applications without a sophisticated understanding of the resulting economic benefits. Targeting individual applications often leads to a "cat and mouse" game where applications or users attempt to obfuscate their behavior to prevent rate limiting while, simultaneously, researchers develop ever more sophisticated (and expensive) classification techniques.

Against this backdrop, this paper makes the following contributions. First, 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. Our proposed techniques limit the bandwidth consumed by bulk transfers during times of peak utilization, effectively smoothing bandwidth consumption over the course of the day. We propose and evaluate a composition of traffic shaping and queueing techniques that together achieve significant reductions in peak bandwidth, while minimally impacting completion times of individual bulk transfers. By contrast, we show that naive traffic shaping techniques can dramatically slow or even terminate many targeted flows.

Our results indicate that it will be in the best interest of many ASs to perform variants of the traffic shaping techniques described in this paper. Unfortunately, we find that once a significant portion of ASs perform such *local* traffic shaping, the *global* system behavior degrades significantly. With increased adoption of traffic shaping, bulk transfer performance degrades as the end-to-end distance the transfer travels grows longer. Essentially, differences in the peak transfer times of ASs in different time zones means that the farther a flow travels (laterally), the higher the probability that at least one AS will throttle the flow at any given time. Even with moderate distances, we find that bulk transfers become constantly throttled to the point of delivering largely no utility.

Moving forward, we belive our results have significant implications. If indeed we enter a regime where many bulk transfers obtain poor performance as a result of economically-incentivized traffic shaping, satisfying the burgeoning demand for bulk transfers will require either different ISP pricing models or novel architectures for performing high-performance bulk transfers that respect existing ISP pricing incentives.

1.1 Motivating Examples

This paper presents a systematic analysis of the local and global effects of ISP traffic shaping. We motivate our study by presenting three real-world examples where popular and important bulk transfer applications suffered as a result of ad hoc ISP traffic shaping policies.

1. Rate-limiting applications to reduce the peak load and thereby, reduce bandwidth costs: In 2002, the University of Washington started limiting the bandwidth of incoming peer-to-peer file-sharing traffic to 20 Mbps because the traffic was costing it an estimated million dollars per year [6]. Published traces of the university's access link traffic from the same period show that the utilization of the access link was always below 70% [32]. This shows that the university was rate-limiting out of economic considerations rather than to avoid performance degradation for other applications. Further, this 20Mbps limit was in effect even during times of low overall utilization (e.g., overnight) despite the fact that ISP charging policy is typically based on 95th percentile utilization levels.

2. Blocking applications to reduce transit bandwidth costs: Some ISPs resort to blocking certain applications rather than rate limiting them. Comcast, the largest broadband ISP in the U.S., was recently caught blocking Bit-Torrent connections across its inter-AS links [15]. We conducted a simple experiment to check whether this blocking is due to capacity constraints or economic considerations. We conducted a BitTorrent transfer between a node in Comcast's network and a node outside it. Simultaneously, we ran a simple TCP data transfer between the same end hosts. Comcast broke up the BitTorrent transfer even as the TCP flow received normal throughput and observed near-zero packet loss. This suggests that BitTorrent was being blocked to lower Comcast's transit bandwidth costs.[1] Interestingly, we observed the same behavior independent of the time of the day, including early in the morning (5 A.M.) when network links are least utilized.

The above examples show that the *bottlenecks* constraining bulk data transfers in the Internet today are often economic rather than physical. While decades of congestion control research focused on managing transfers across physical bottlenecks, very few studies have focused on the problem of traffic shaping i.e., how to manage transfers across economic bottlenecks. As a result, ISPs often deploy egregiously sub-optimal ad hoc solutions.

. 3. Local traffic shaping can have unforeseen global consequences: Our next example is based on an attempt by one of the authors to transfer

[1] Our hypothesis is further supported by the fact that Comcast was not interfering with intra-AS BitTorrent traffic [15].

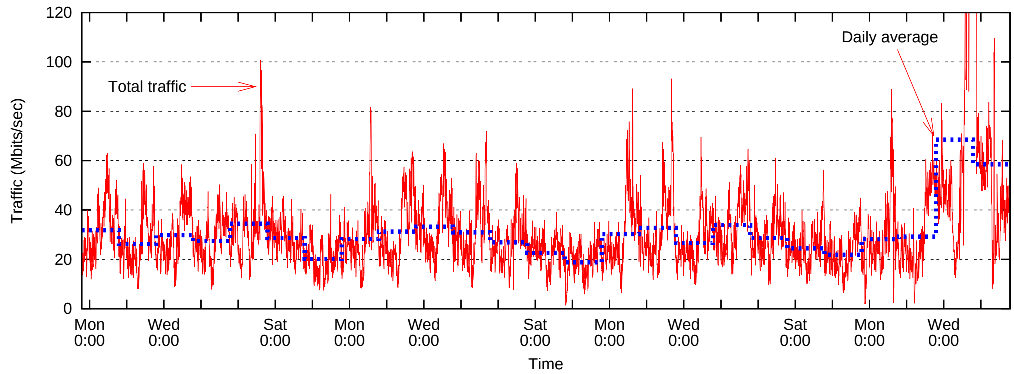


Figure 1.1: Variation in incoming traffic at Ohio State University over a month: The traffic exhibits recurring diurnal and weekly patterns.

a large scientific data set (> 1 Terabytes) between a node located on the west coast of the U.S. and another node in Europe. The European node was connected to an ISP that allowed free bandwidth between 12:00 AM and 6 AM local time. Ironically, this coincided with the peak utilization of the U.S. node, located 9 time zones away. We had to choose between forcing one organization to pay for the transit costs, taking months to transfer the data at some low bitrate, or resorting to out of band transmission using digital media. In this case, we happened to choose the last technique.

Taken together, our examples motivate the following questions. First, if the goal is to reduce bandwidth cost, what is the appropriate local traffic shaping policy for inter-AS links? Second, as economic considerations drive ISPs to adopt such traffic shaping policies, what is the global impact on end-to-end transfers across the Internet? We study these two questions in this paper.

2 Temporal Variation in Internet Traffic

ISPs employ traffic shaping to reduce the peak utilization of their access links and to lower transit costs. Existing traffic shaping techniques effectively act as blunt instruments; they rate limit particular classes of traffic even when there is no corresponding reduction in peak levels of utilization. One goal of this work is to demonstrate more effective traffic shaping techniques capable of shifting network load from periods of high utilization to periods of low utilization. Thus, the effectiveness of traffic shaping is limited by the skew in traffic distribution over time.

In this section, we study how network traffic varies over time by analyzing real-world network traces. Our goal is to develop an understanding of the potential opportunity for traffic shaping in the Internet.

Our characterization of temporal variation in network traffic is driven by three questions that play a critical role in our analysis of traffic shaping in later sections:

1. How large is the diurnal variation in network traffic? This variation is useful to estimate the optimal reduction in peak load one could achieve using traffic shaping.

2. How stable is network traffic across different days? As traffic shaping involves delaying some traffic for delivery at some future time, it is important to know whether recent history of network load can be used to plan future schedules. If traffic is stable across different days, ISPs can use the knowledge of network load on the previous day when traffic shaping.

3. What is the impact of bulk flows on network load? It is important to understand the contribution of bulk flows to network load, because (a) very large flows tend to be more tolerant to delays, and hence traffic shaping, than very short flows like Web traffic, and (b) traffic shaping a few very large flows is preferable to affecting a large number of very small flows.

2.1 Traces

We studied two types of traffic traces. First, we downloaded publicly available traces of incoming and outgoing traffic at the access links connecting over 40 different universities to the Abilene backbone [1]. Second, we collected a recent trace of incoming and outgoing traffic at the access link connecting ABC [1] university campus, with a population of over $10,000$ people, to the commercial Internet. Our Abilene traces were gathered over a period of several months, while our ABC trace was limited to one day.

Our ABC campus trace captured the start time, size, and duration for all TCP flows. Our Abilene traces contain Netflow [14] records of TCP flow durations and sizes. Abilene Netflow records are based on sampling every 100^{th} packet. To reproduce the original trace, we used commonly accepted techniques proposed elsewhere [13, 25].

2.2 Diurnal variation in network traffic

We first analyze a month-long trace of incoming traffic at the Ohio State University [2] for November 2007 and the day-long trace of incoming traffic at our ABC campus network. Analysis using Abilene traces of incoming and outgoing traffic at other universities yielded similar results.

Figure 1.1 shows the bandwidth consumed by the incoming traffic at Ohio State during the course of the month. The bandwidth is averaged over an interval of 5 minutes. We chose 5 minutes because it is widely believed to be the duration over which ISPs average bandwidth for billing their customers. The network traffic exhibits recurring diurnal and weekly patterns; we see considerable variation in traffic during the course of a single day, and low bandwidth usage on weekends.

Figure 2.1 focuses on the diurnal variation in incoming network traffic at the ABC campus network. The traffic shows a clear diurnal behavior similar to the one we observed in our Abilene trace. The traffic is significantly lower during the early morning compared to the rest of the day; in fact, the peak bandwidth is 8.3 times higher than the trough bandwidth. The considerable diurnal variation in network load incentivizes ISPs to deploy traffic shaping. By shifting some load from the peak hours to the times when the network is less utilized, ISPs can hope to reduce peak traffic and thereby, their bandwidth costs.

[1] Name obscured for anonymity.

[2] We chose Ohio State University as it is one of the largest universities connecting to the Abilene network.

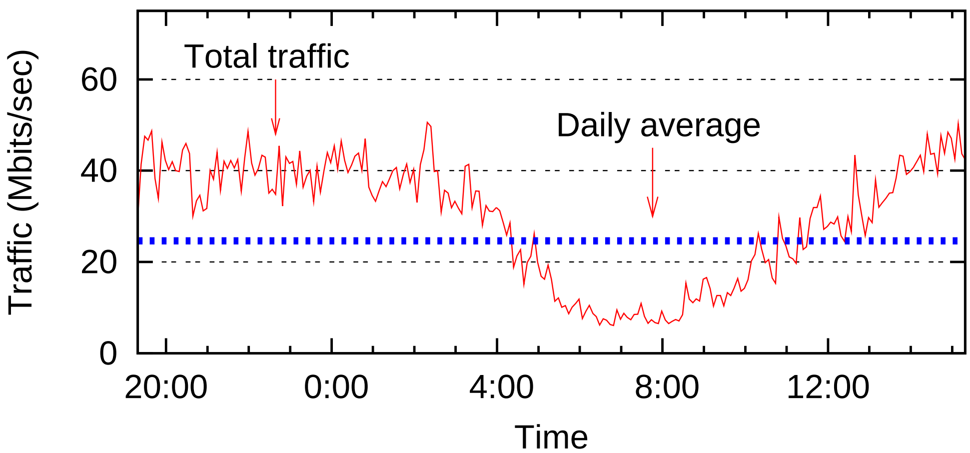


Figure 2.1: Variation in incoming traffic at ABC University: The traffic exhibits a diurnal pattern with a trough in the early morning.

Average bandwidth represents the best reduction in peak bandwidth any traffic shaping algorithm can hope to achieve. Attempts to push the peak below the average would lead to some flows being starved. To quantify the optimal reduction in peak load with traffic shaping, we computed the ratio of the daily peak bandwidth to the daily average bandwidth in our traces. The diurnal average bandwidth is 40% to 60% lower than the diurnal peak bandwidth in all our traces (shown in Figures 1.1 and 2.1). Thus, with intelligent traffic shaping, ISPs can hope to reduce their peak utilization by half.

Finally, since ISPs charge network traffic based on 95^{th} percentile usage, we compared daily peak utilization to daily 95^{th} percentile utilization. On most days, the difference was fairly small (26%), though occasionally the difference could be as high as 61%. The relative proximity of 95^{th} percentile and peak utilization suggests that traffic shaping can lead to similar reductions in 95^{th} percentile utilization.

2.3 Daily stability of network traffic

Figure 2.2 shows how the daily average, daily peak, and daily 95^{th} percentile traffic change during our month-long trace. It shows that while daily average traffic remains relatively stable (with slightly less traffic on the weekends than on weekdays), the daily 95^{th} percentile and peak traffic change significantly. This suggests that while ISPs may not be able to predict the peak or 95^{th} percentile traffic, they have a good estimate of the average demand. This

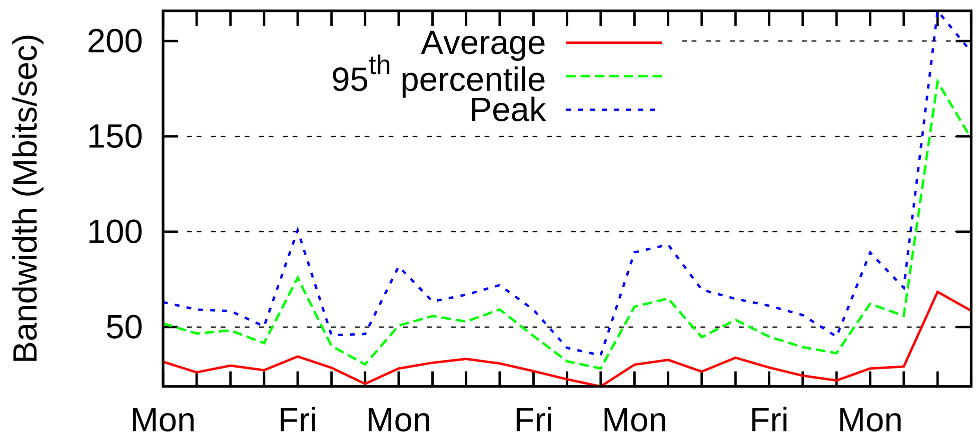


Figure 2.2: Variation in daily average, 95^{th} percentile, and peak bandwidths over a month: While the average bandwidth utilization is rather stable across days, 95^{th} percentile and peak bandwidth vary significantly.

Ratio	Duration		
	Day	Week	Month
peak to average	2	3.6	7
95^{th} percentile to average	1.7	1.7	1.9

Table 2.1: Although the ratio of peak to average at different time scales varies considerably, the relative difference between 95^{th} percentile and average is rather constant.

provides yet another incentive for ISPs to deploy traffic shaping, i.e., to tame the unpredictable peak load to the more predictable and manageable levels of average load.

One hypothesis for why ISPs prefer 95^{th} percentile utilization as the charging model is that it is a more stable metric when computed across multiple days. Table 2.1 shows how the different metrics change when computed over the first day, first week, and entire month of the Ohio trace shown in Figure 1.1. The table shows that when we go from a day to a month, the ratio of peak to average bandwidth increases from a factor of 2 to 7. Surprisingly, the ratio of 95^{th} percentile to average remains stable around 2. This not only confirms our hypothesis, but it also shows that ISPs can reduce their 95^{th} percentile utilization by a factor of 2 with traffic shaping.

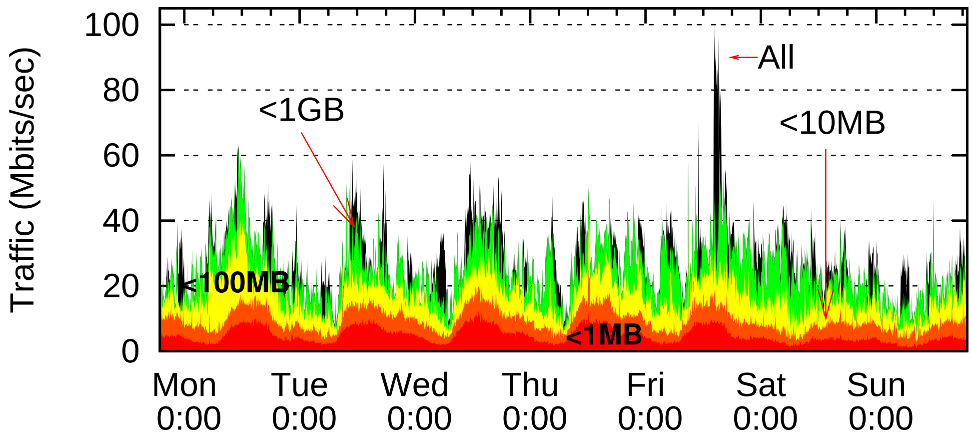


Figure 2.3: Incoming traffic by flow size: Although few in number, large flows contribute significantly to peak bandwidth.

Flow size	Normalized Peak
$<$ 1MB	0.06
$<$ 10MB	0.13
$<$ 100MB	0.21
$<$ 1GB	0.70
All Flows	1.0

Table 2.2: Contribution of bulk flows to traffic peak: Removing all flows bigger than 10MB results in a peak reduction of 87%.

2.4 Impact of bulk flows on network traffic

Table 2.3 shows the fraction of flows and bytes contributed by flows of different sizes in the Ohio trace. A vast majority ($> 99.5\%$) of all flows are less than 10MB in size. However, the small percent of flows ($< 0.5\%$) larger than 10MB account for almost 70% of all bytes transferred. Flows larger than 10MB account for 0.07% of all flows and 47% of all bytes in our ABC trace which suggests that our observations are not unique to the Ohio State trace. Thus, bulk flows are very few in number but they account for most of the bytes in the traffic. This means that traffic shaping of a small number of large bulk flows can have a dramatic impact on network traffic.

Bulk flows not only account for a large fraction of bytes, but they also contribute significantly to the peak network load. Figure 2.3 shows the breakdown of network traffic in the Ohio State trace based on the size of the flows. Bulk flows follow the same diurnal patterns as the rest of the traffic, and

Flow size	Perc. of flows	Perc. of bytes
< 1 MB	97.0%	16%
< 10 MB	99.5%	32%
< 100 MB	99.9%	54%
< 1 GB	99.99%	80%

Table 2.3: Distribution of flows and bytes: Large flows are very few in number, but they account for a significant fraction of the total traffic.

consequently, they cause a substantial rise in peak traffic. Table 2.2 quantifies the contribution of bulk flows to the peak traffic. If we remove bulk flows larger than 10MB, the peak utilization drops by 87%. This shows that traffic shaping bulk flows holds a tremendous potential for reducing peak utilization.

2.5 Summary

We studied temporal variation in network traffic. We found significant diurnal variation in bandwidth consumption. In the limit, an appropriate traffic shaping mechanism that can offload traffic during times of high utilization to times of lower utilization can reduce peak and 95^{th} percentile utilization by a factor of 2. Average bandwidth is surprisingly predictable, which helps a potential traffic shaper to plan future schedules. Finally, targeting a small number of bulk flows for traffic shaping can lead to a significant decrease in peak load.

3 Inter-AS Traffic Shaping

In this section, we discuss issues involved in shaping traffic at the access links of an ISP. The ISP's goal is to reduce peak link utilization, while leaving interactive flows unaffected and causing only minimal degradation in the completion times of bulk flows. To this end, the ISP's access routers classify flows into two broad traffic classes and schedule their packets for transmission into separate queues: a foreground queue for interactive traffic, and one or more background queues for bulk traffic.

3.1 Analysis methodology

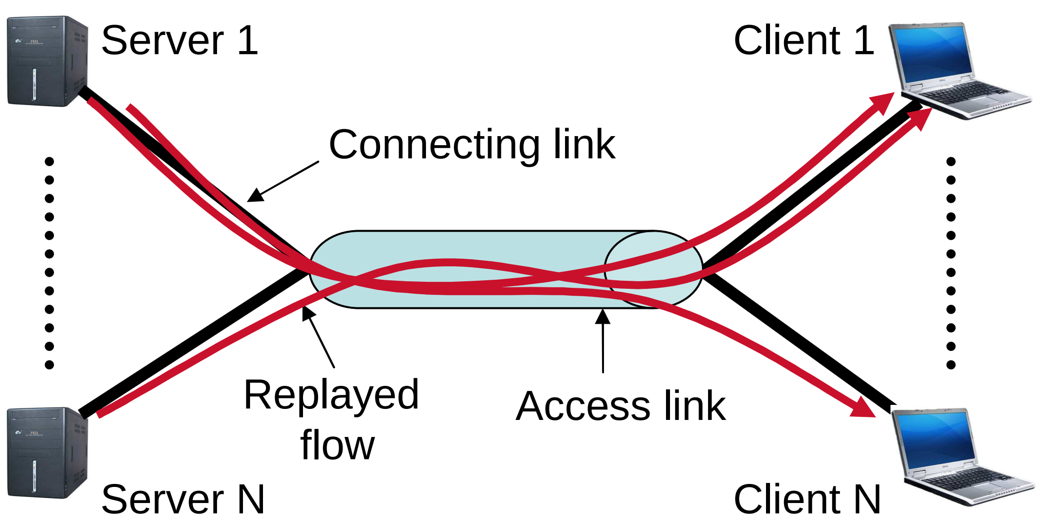


Figure 3.1: Simulation topology: Each flow in the trace is simulated by setting the capacity of the server's connecting link to be equal to the flow's average bandwidth.

We used trace-driven simulations to study the behavior of flows under various traffic shaping mechanisms. Our analysis was conducted using ns-2 and our traces were collected at the border routers of university networks (as described in Section 2.1). While the general representativeness of university traffic is a concern, our analysis is primarily dependent on a few characteristics of the trace, such as the distribution of flow sizes and diurnal variations in flow arrival times. In Section 2, we showed that these characteristics are representative of broader Internet traffic.

We used the simulation topology shown in Figure 3.1 to analyze traffic shaping over the access link. We faced an interesting challenge when we tried to replay 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 3.1). This ensures that the simulated flows complete in similar durations as the original flows in the trace. We pick the RTT of a flow choosing from a distribution of latency measurements using the King tool [22].

3.1.1 Validation

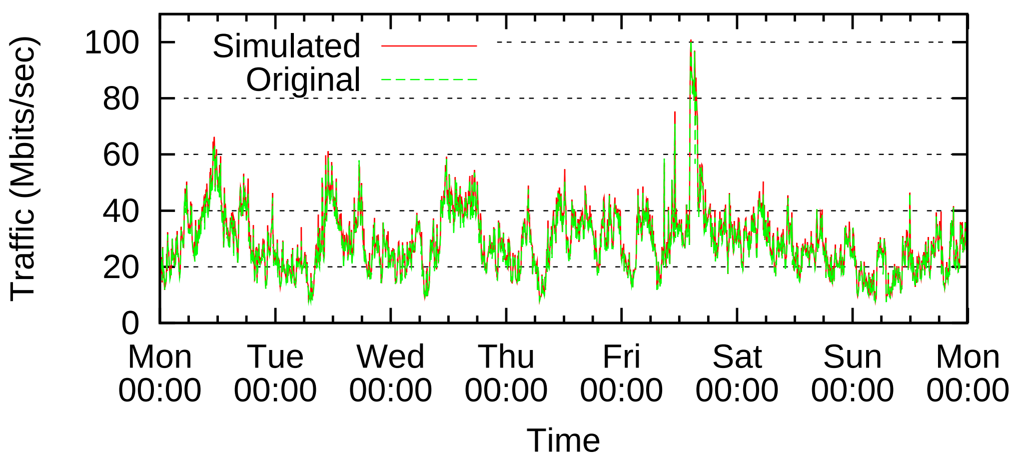


Figure 3.2: Comparing original and replayed traces: The aggregate bandwidth of the original trace and our simulation match very well.

We examined how well our replayed trace matches the original trace. As flow start times, sizes and durations are inputs to the replayed trace, we only compared the aggregate bandwidth consumed by flows at the access links in both traces. Figure 3.2 shows that the aggregate bandwidth over time looks very similar. The plots match very well even when we restrict the flows to less than a certain size. Our results suggest that our simulated trace faithfully replays the original trace from the perspective of properties that are important for traffic shaping.

3.2 Differentiating traffic classes

To implement traffic shaping, ISP's access routers need to distinguish between packets belonging to interactive flows and bulk flows. The former are sent to a higher-priority foreground queue, while the latter are scheduled in a lower-priority background queue. At a high level, routers can identify background flows in one of three ways. First, ISPs can rely on end hosts or applications to mark packets that can or cannot be delayed in background queues. ISPs can incentivize applications to do so through a tiered pricing model [17, 19, 39]. Second, ISPs can use traffic analysis to identify packets belonging to different applications and prioritize traffic on a per-application basis. Application-level traffic shaping based on the analysis of packet headers or content is widely deployed in the Internet today [31]. Third, ISPs can differentiate between flows based on their size, giving lower priority to flows larger than a certain size. Many residential ISPs are known to limit the rates of flows that are very large or customers that use their links heavily [16].

In our analysis of traffic shaping, we differentiate between foreground and background flows based on their size. Our choice was driven primarily by the limitations of our trace – we have accurate information about flow sizes but little else. In practice, an ISP might use one or more of the three techniques we discussed above to correctly identify background flows. The relative merits of these approaches is orthogonal to our work.

3.2.1 Optimal threshold size for background flows

We categorize all flows larger than a certain threshold size as background flows. Determining this threshold presents a tradeoff between the number of flows affected by traffic shaping and their potential to reduce the peak utilization. Since no sustainable traffic shaping can push the peak utilization below the average utilization, the ideal threshold size is the one that affects the fewest number of flows that still allows the peak to be lowered to the average.

Figure 3.3 illustrate this tradeoff. For each threshold value (plotted along the X-axis), we computed (a) the fraction of affected flows (shown on the right Y-axis) and (b) the maximum achievable peak reduction (shown on the

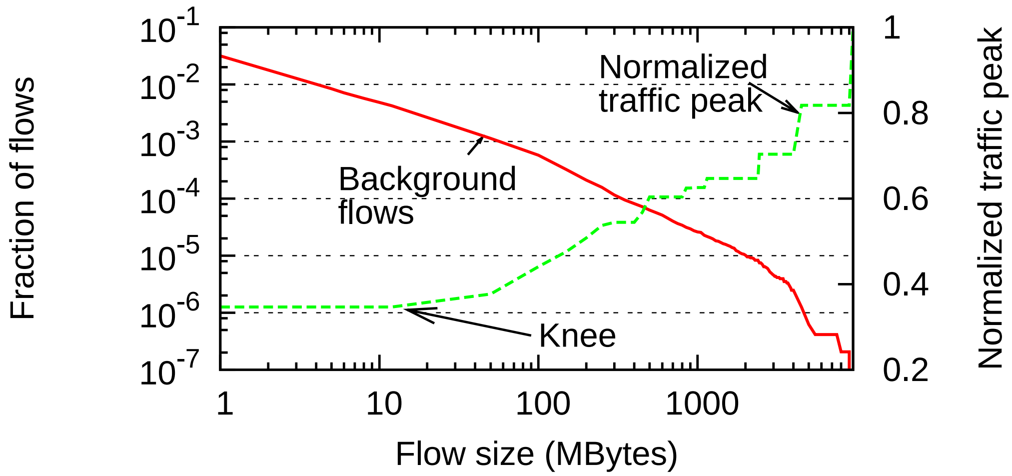


Figure 3.3: Selecting the boundary between foreground and background flows: At about 10MB the tradeoff between low peak utilization and the number of flows affected by traffic shaping is optimal.

left Y-axis). As expected, selecting a lower threshold causes more flows to be traffic shaped and increases the magnitude of the peak reduction. However, beyond a certain point, decreasing the threshold does not reduce the peak further; it only affects more flows needlessly. The resulting knee in the curve marks the optimal threshold size, approximately 10MB in this trace. Analysis using traces of other universities showed that a majority have an optimal threshold of around 10MB as well.

3.3 Single queue traffic shaping

We begin with an analysis of a simple traffic shaper that uses one foreground and one background queue to separate flows into two traffic classes. The traffic in the foreground queue is left untouched, and sent in the conventional best-effort manner. The packets in the background are sent only when (a) the foreground queue is empty, and (b) the link utilization is below the bandwidth limit being imposed. This policy gives absolute priority to foreground flows over background flows.

To enforce the aggregate bandwidth limit, we use a modified token bucket algorithm. As in a traditional token bucket, tokens are generated at a rate equal to the bandwidth limit to be enforced. Sending a packet consumes a number of tokens equal to the packet's size. However, unlike a traditional token bucket, foreground packets may be sent even when tokens are unavailable. This means that the number of tokens can become negative and that the overall traffic would exceed the bandwidth limit when the foreground traffic exceeds the limit. To reflect the current bandwidth metering models, which average bandwidth over 5-minute time intervals, we reset the number of tokens every 5 minutes. This also prevents unwanted bursts or starvation of background traffic after a long period of low or high foreground activity.

We use the recent history of network traffic when setting the aggregate bandwidth limit. We showed that past demand is a reasonable predictor of current demand in Section 2. Since the best we can hope is to bring traffic peaks down to the average, we set the daily bandwidth limit to the average bandwidth observed in the previous day. However, we increase this limit by 5% to account for occasional load swings and increasing traffic demands over time.

3.3.1 Evaluation: The good, the bad, and the ugly

We implemented our traffic shaper in the ns-2 network simulator, and evaluated it using our traces. We present results obtained using a week-long trace of outgoing traffic at Ohio State University. We also evaluated incoming and outgoing traffic at this and other universities and observed similar results.

We set the threshold between foreground and background flows at 10 MB. Since routers have no a priori knowledge of flow sizes, flows are moved from the foreground queue to the background queues after they have transmitted their first 10MB. Note that this does not necessarily require keeping per flow state [26].

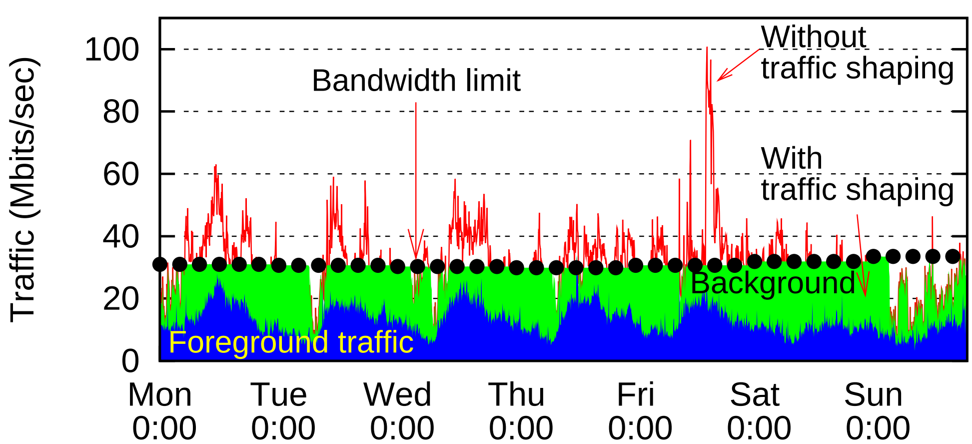


Figure 3.4: Impact of traffic shaping: While the foreground traffic is not affected, the total traffic is kept below the enforced bandwidth limit.

The good: Near-optimal reduction in peak bandwidth consumption. Figure 3.4 plots the aggregate bandwidth of flows during the week before and after using our traffic shaper. At no time does the aggregate traffic noticeably exceed the bandwidth limit, confirming that the techniques we used to select the aggregate bandwidth limit and threshold size for background flows work well. Our traffic shaping reduced the peak load by 63%, which is very close to the optimal 65% reduction one could achieve by capping the bandwidth at average load.

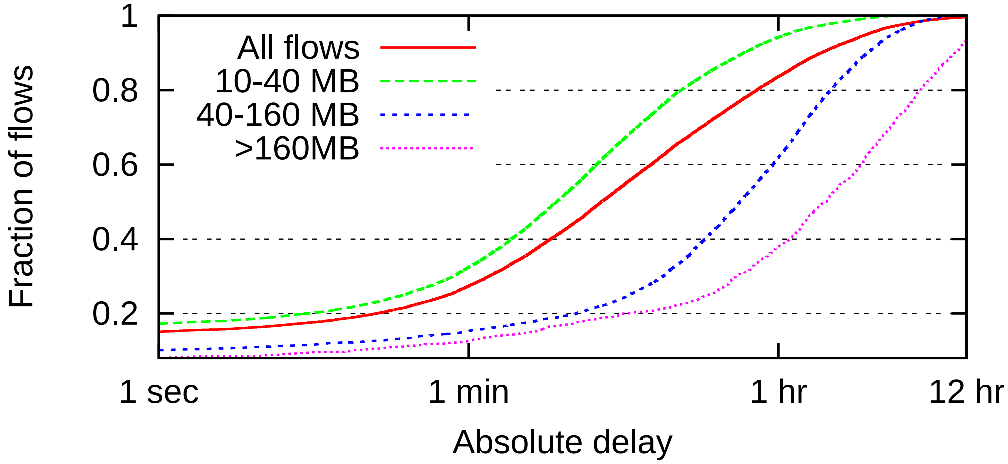
The good: Interactive foreground flows remain unaffected. One of the goals of our traffic shaper is to preserve the performance of foreground flows. We verified this by comparing the completion times of individual foreground flows with and without the traffic shaper. Virtually no flows show an increase in completion time. On the contrary, a small fraction of flows performed better when the traffic shaper is active because of reduced competition from background traffic. Note that foreground flows account for over 99.5% of all flows, and hence our traffic shaper leaves a vast majority of network flows unaffected.

The bad: Bulk background flows suffer noticeable delays in completion times. The traffic shaper gives lower priority to packets from background flows. Background packets are also occasionally blocked by the token bucket algorithm during the peak hours. Figures 3.5 (a) and (b) show the absolute and relative delays in the completion times of background flows due to traffic shaping. The delays are quite noticeable; 50% of background flows suffered a delay of 6 minutes or more. In terms of relative delay, the completion times of 50% of flows increased by a factor of 2 or more.

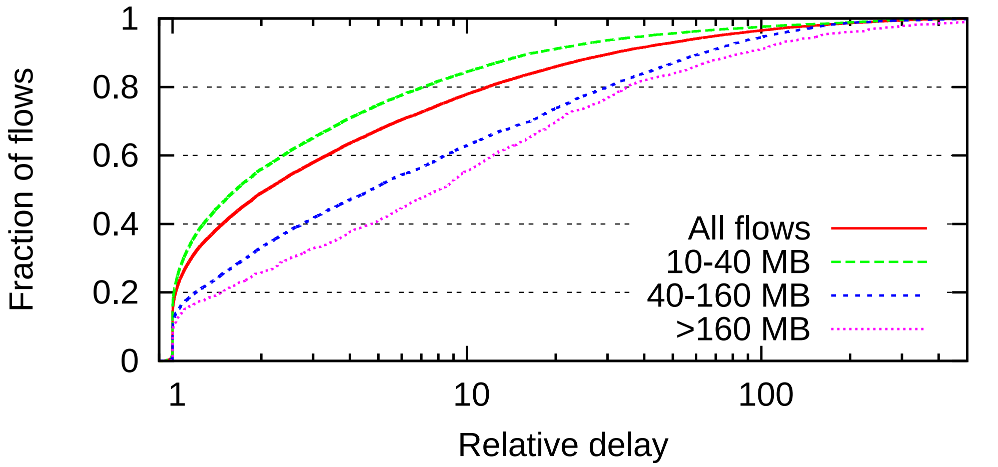
The performance loss suffered by flows varies based on their size. Figures 3.5 (a) and (b) also show the absolute and relative delays of background flows with different sizes. Larger flows incur longer absolute delays in their completion times, but they incur similar relative delays as smaller flows. This can be explained by the fact that all background flows compete equally for available background bandwidth independent of their size.

However, the number of flows with a given size decreases exponentially with flow size (see Figure 3.6). This suggests that scheduling policies like shortest-job-first might be effective at reducing the aggregate completion times of flows. By prioritizing small background flows over large flows, such schemes could significantly reduce the completion times of a lot of small flows at the expense of adding a modest delay to a few very large flows. We explore this idea in more detail later in Section 3.5.

The ugly: A non-negligible fraction of bulk background flows terminate before completing their transfers. A more important concern than performance loss of flows is TCP connection termination. This can



(a)



(b)

Figure 3.5: Background flows experience noticeable delays: 50% of all background flows get delayed by 6 minutes or more. In terms of relative delay, for 50% of bulk flows their completion time at least doubles.

occur when foreground flows reach close to or exceed the bandwidth limit of the traffic shaper, forcing background TCP flows to be excessively throttled. Packets can be stuck in the background queue for a long duration or be dropped when the queue is full. TCP would time-out waiting for the packets and retransmit them [2]. After a certain number of failed retransmissions, TCP breaks the connection concluding that the other end node has departed. Our experiments with TCP implementations in Windows and Linux showed that in practice TCP terminates flows when it fails to retransmit a packet between 5 to 15 times over a time period ranging from 60 seconds to 30 minutes. The default ns-2 TCP implementation does not demonstrate this behavior, but we configured our simulations to terminate TCP connections if they fail to retransmit a packet at least 5 times over a period of 60 seconds or more.

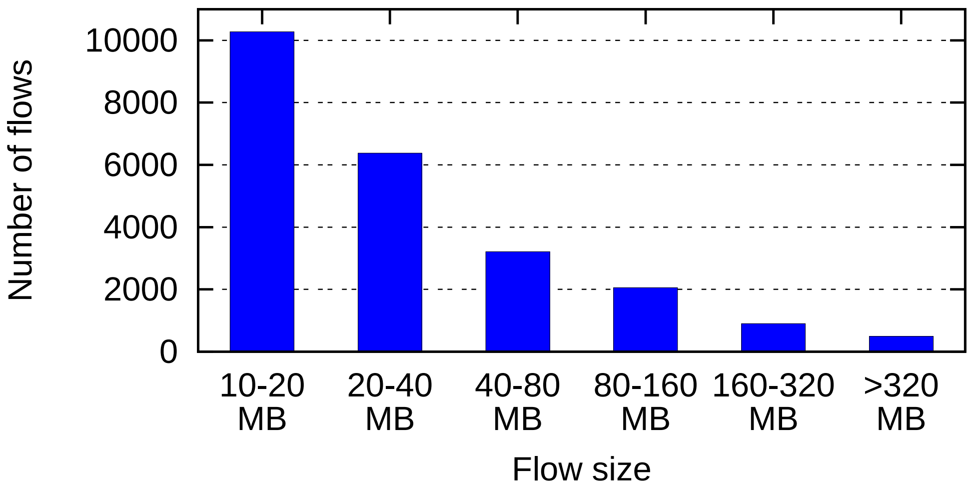


Figure 3.6: Distribution of background flow sizes: The number of flows with a given size decreases exponentially with the flow size.

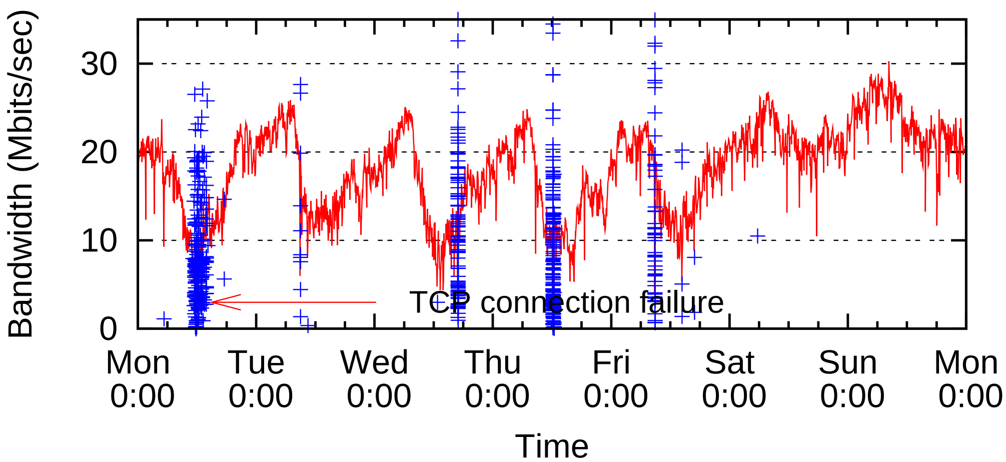


Figure 3.7: Traffic shaping and TCP: TCP connections fail when there is very little background bandwidth.

We found that 570 background flows terminated early due to broken TCP connections over the course of a week. Figure 3.7 plots the times when the flows terminated along with the aggregate bandwidth available to all background flows at the time. Not surprisingly, it shows that flows break down when background flows have very little or zero available bandwidth. Broken TCP connections have to be restarted by end users and run counter to our goal of transparent traffic shaping. We investigate mechanisms to keep TCP connections alive in Section 3.4

3.3.2 Summary

In summary, our analysis of a simple single-background-queue traffic shaper shows that it effectively lowers peak link utilization. On the positive side, it leaves the vast majority of network flows unaffected. On the negative side, background flows suffer noticeable delays in their completion times, and more worryingly, a non-negligible fraction of the flows are starved to early termination. We investigate mechanisms to address these two problems in the rest of this section.

3.4 Keeping background TCP flows alive

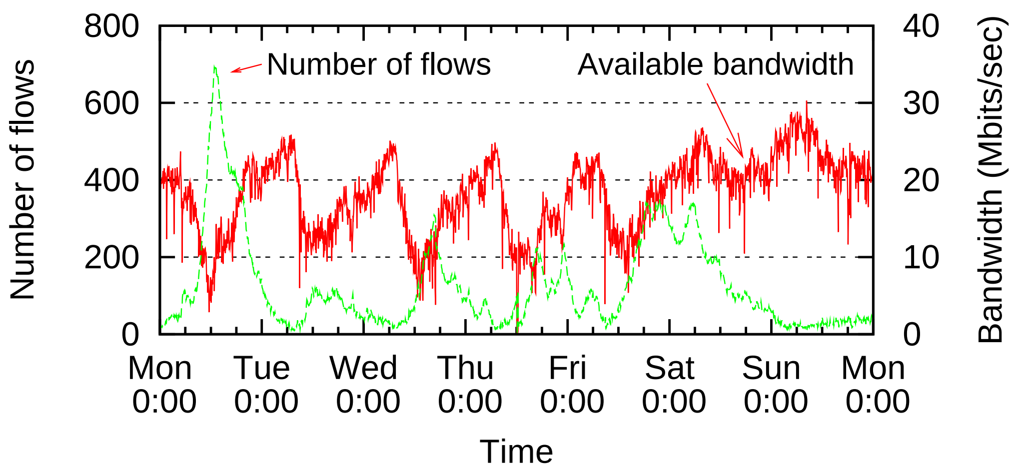


Figure 3.8: Limiting bandwidth for background flows: When the overall available bandwidth for background flows decreases, the number of active background flows rises sharply.

When the foreground traffic reaches or exceeds the bandwidth limit, our initial traffic shaper allocates zero bandwidth to background flows causing

TCP connections to time-out and expire. To avoid such failures, each background flow needs to transmit packets at some minimal rate. The key question is how to determine the minimal aggregate bandwidth for background flows required to keep the flows alive.

Our analysis shows that allocating a constant aggregate bandwidth for all background flows, as ISPs typically do today [6] with their traffic shapers [31], will not work. Figure 3.8 shows shows that the number of active background flows rises sharply and unpredictably when there is a trough in bandwidth allocation for bulk flows. This is expected because when foreground traffic peaks, the traffic shaper rate limits bulk flows causing them to complete at a much slower rate than they arrive. With any constant bandwidth allocation to bulk flows, each flow receives decreasing bandwidth as the number of background flows grows, and eventually, this will cause TCP flows to terminate.

We choose to allocate bandwidth proportional to the number of background flows.[1] For each additional flow, we allocate an additional 10Kbps of bandwidth to the background queue. 10Kbps is sufficient to allow a TCP flow to transmit 1 packet every $1 - 2$ seconds, a minimal rate sufficient to prevent connection termination.

We repeated the experiment with our bandwidth allocation and it resulted in zero connection breakups, down from 570 failures without this technique. The peak packet loss rate dropped from 47% to 5%, while the additional allotted bandwidth caused peak utilization to increase by a negligible 9%.

Our simulations of traffic shaping using other Abilene traces (not shown here) revealed a few scenarios where our bandwidth allocation still resulted in termination for some connections. A closer analysis revealed the cause to be the delay suffered by packets waiting in the background queue rather than excessive packet loss. In these cases, packets were stuck in the background queue for over 10 minutes. To bound the queuing delay, we impose a lower bound on the bandwidth available to background flows.[2] We repeated the simulations setting the bandwidth lower bound such that a full background queue can be drained in 10 seconds and observed near-zero connection failures in all our simulations. This increases the overall peak bandwidth consumption by the lower bound needed to drain the queue.

[1] There are well known techniques to estimate the number of flows in a router queue without maintaining per-flow state [26].

[2] An alternative would have been to dynamically change or shorten the queue sizes. We explored this option and found it runs into problems studied in detail in [27].

Queuing Policy	Average Delay
Single queue	39.5 min
Optimal shortest flow first	4.4 min
Offline, 6 queues	8.5 min
Online, 6 queues	13 min

Table 3.1: Absolute delays with ideal scheduling policies: Giving strict priority to shortest flows considerably reduces the average delay.

3.5 Improving average completion times

Having addressed the problems with liveness of flows, we turn our attention to reducing the completion times of flows. When traffic shaping with only one background queue, background flows get lower priority than foreground traffic, but no priority is enforced among the background flows. On the other hand, we know that giving priority to the shortest background flow would minimize the mean completion time. Shortest-job-first scheduling is especially attractive for workloads such as ours where the number of jobs (flows) decreases exponentially with the size of the job (See Figure 3.6).

Ideally, shortest-flow-first queuing could be achieved by using a different priority queue for each background flow. When transmitting, packets from the queue belonging to the smallest flow are given the highest priority. In practice, such shortest-flow-first scheduling faces two problems: first, routers may not have resources to implement per-flow queuing. Second, routers do not have a priori information about the flow sizes. In the rest of this section, we first quantify the performance of optimal shortest-flow-first queuing assuming a priori knowledge of flow size, and then analyze the relative performance of a practical implementation.

3.5.1 Optimal shortest-flow-first queuing

We simulated optimal shortest-flow-first queueing by implementing one priority queue per flow in our ns-2 simulator. Figure 3.9 compares the distributions of absolute delays in completion times suffered by flows using a single and optimal shortest-flow-first queueing. It shows that shortest-first-queueing improves the median delay in completion time by a factor of 11. Table 3.1 shows the improvement in the average delay of completion times. Optimal shortest-flow-first queueing improves the average delay by a factor of 8, demonstrating the huge potential of shortest-flow-first queueing.

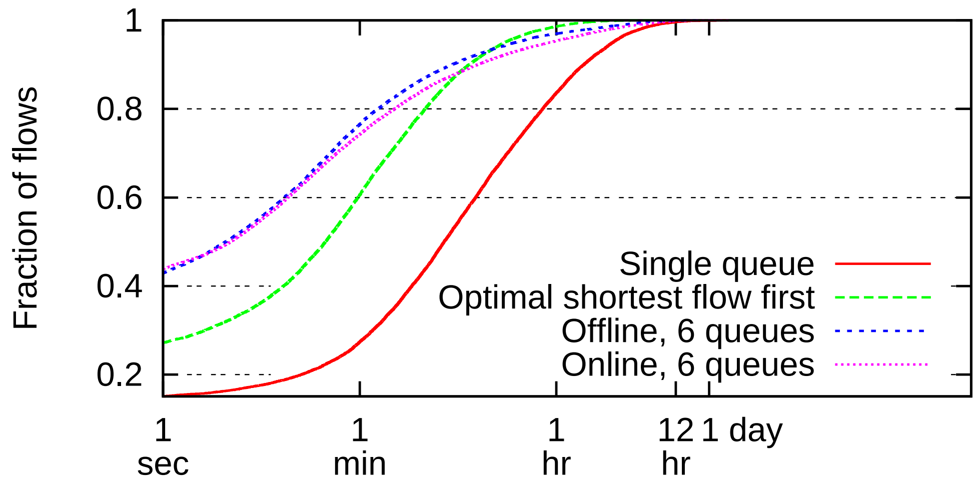


Figure 3.9: Absolute delays in flows completion times when using different queuing policies: Both offline and online scheduling with multiple queues improve the completion times significantly.

Queuing	Median	90th perc.	Max.
Single queue	5.8 min	1.8 hrs	36 hrs
Online, 6 queues	4.1 sec	11.7 min	37 hrs

Table 3.2: Absolute delays when multiple queues are used: Traffic shaping using a low number of background queues to prioritize short flows results in a dramatic reduction of the relative delays.

3.5.2 Shortest-flow-first queueing with bounded number of queues

In practice, it is not possible to allocate a separate router queue for each flow. So we need to map groups of flows with similar sizes to each queue. We simulated a router with 6 background priority queues. Each queue receives packets from flows whose size falls in a given range. Since the number of flows decreases rapidly with the size, we choose to exponentially increase the range of flow sizes allocated to queues. Thus, our first queue handles $10 - 20$MB flows, the second queue handles $20 - 40$MB flows and so on. The last queue handles all flows larger than 320MB.

Figure 3.9 shows the delays in flow completion times using 6 background priority queues. Performance closely matches the ideal traffic shaper that uses one queue per flow. Table 3.1 shows that while the average delay increases by a factor of 2 compared to the ideal traffic shaper, it is still a factor of 5 better than the performance of a single background queue. Thus, shortest-flow-first queueing is very effective at reducing completion times even when limited to a small number of queues.

3.5.3 Online shortest-flow-first queueing with bounded number of queues

Thus far, we assumed offline knowledge of flow size. In practice, traffic shapers do not have this information, and they have to infer the flow size online. One simple way is to assign a flow to a priority queue based on the amount of transmitted thus far. For example, a flow would be assigned to a queue meant for $10 - 20$MB flows after it sends 10MB of data. This flow would be moved to a lower priority queue meant for $20 - 40$MB flows after it transfers 20MB of data. We simulated online shortest-flow-first queueing with 6 background priority queues.

Figure 3.9 shows the delays in flow completion times using online shortest-flow-first queueing with 6 background priority queues. It shows that the online traffic shaper performs nearly as well as the traffic shaper with identical number of queues but with a priori knowledge of flow sizes. Table 3.1 shows that, compared to a single background queue, our online traffic shaper improves the average delays by a factor of 3.

Table 3.2 compares the performance of our single background queue traffic shaper with our online, shortest-flow-first traffic shaper with 6 background queues. When we use the online traffic shaper, the median delay in completion times decreases by a factor of 80 from 5.8 minutes to 4.1 seconds. This improvement primarily affects short background flows (i.e., those between 10 and 20MB) and in practice, this could prove crucial for increasingly popular soft real time TCP flows like YouTube clips [40]. Similarly, the 90^{th} percentile delay decreases from 1.8 hours to 11.7 minutes benefiting flows smaller than 100MB. In practice, this could greatly benefit software downloads such as OS updates and games. These short flows benefit at the expense of extremely large flows ($>$ 100 MB). Interestingly, the table shows that the delay penalty suffered by the largest flow in our trace increased from 36 to 37 hours, negligible at this scale. This strongly suggests that ISPs should prioritize short background flows at the expense of large background flows using multiple priority queues.

4 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 drive ISPs to deploy traffic shaping at their inter-AS links, long flows may be subject to traffic shaping at multiple inter-AS links (see Figure 4.1)

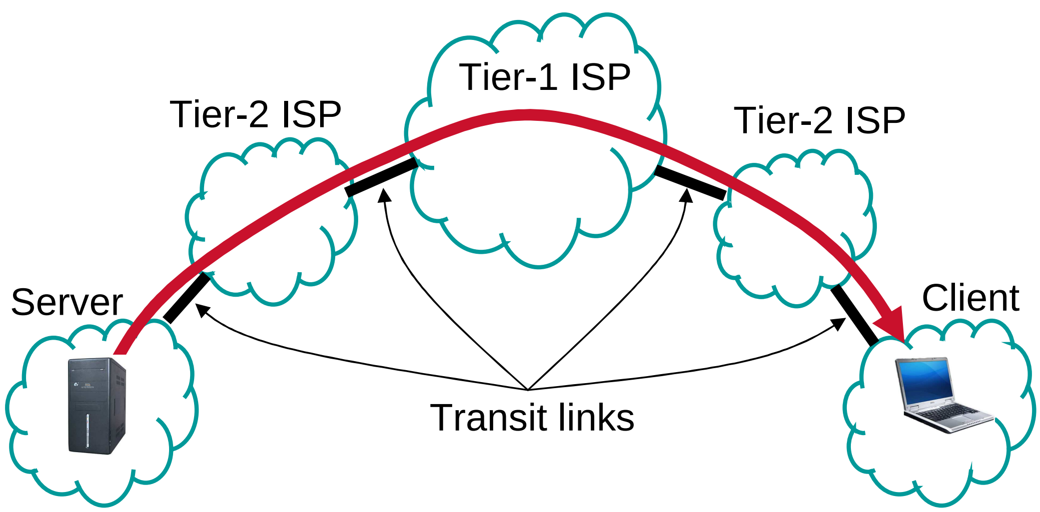


Figure 4.1: Flow traversing multiple ISPs: There is an incentive for each ISP along the path to perform traffic shaping at its transit links.

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.

Figure 4.2 shows the ns-2 topology we used for our analysis. We simulated a long-running TCP flow over a multi-hop Internet path, and we used our university traces to simulate inter-AS traffic shaping on individual hops along the Internet path. The details of traffic shaping are described in Section 3.1.

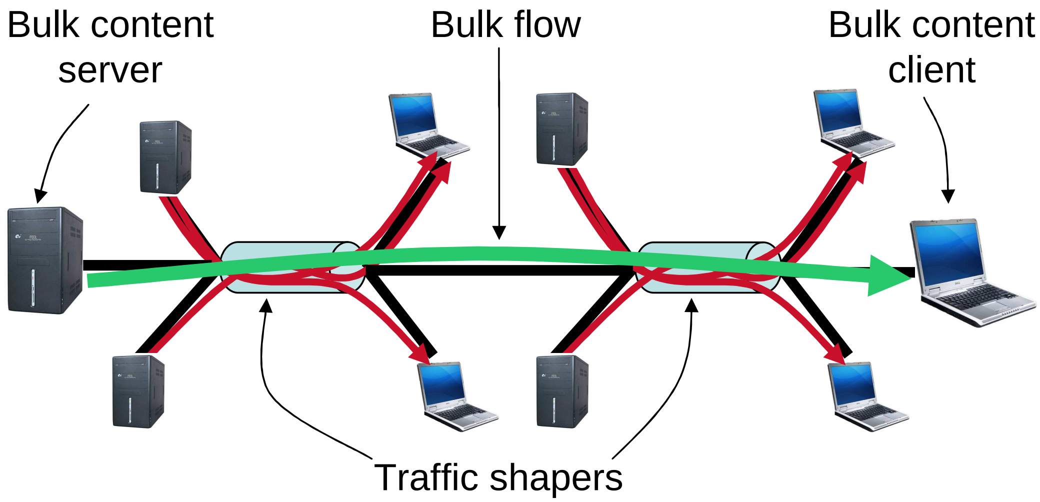


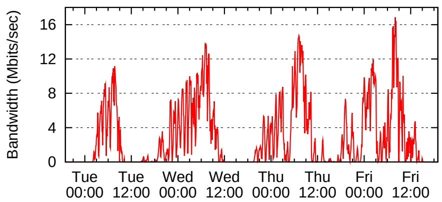
Figure 4.2: Simulation of a long network path: We simulated a long-distance network path by connecting 2 traffic shapers. Each traffic shaper acts independently. A bulk transfer traverses all traffic shapers.

4.1 Multiple traffic shapers and end-to-end performance

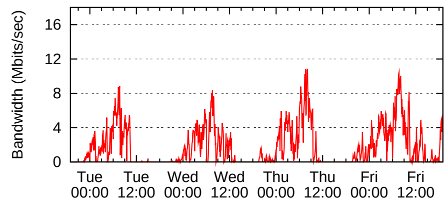
To understand the impact of multiple traffic shapers, we compared the performance of a flow that traverses two inter-AS links with the performance of flows that traverse each of the two inter-AS links separately. We present the results from our analysis using the outgoing link from Ohio State University and the incoming link at the University of Wisconsin.

Our simulations ran over a period of 4 days from Tuesday through Friday. We focus on the results from the weekdays as the traffic shaper is largely inactive on the weekends when there is plenty of available bandwidth.

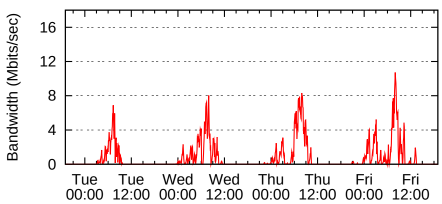
Figures 4.3 (a) and (b) show the bandwidths a long flow achieves when traversing only Ohio's access link or only Wisconsin's access links. In both cases, the flow gets most of its bandwidth between midnight and early morning 6 AM when link utilization is at its minimum. Figure 4.3 (c) shows the bandwidth when the flow passes through both traffic shapers. In this case, the flow receives considerably less bandwidth compared to the cases with only one active traffic shaper.



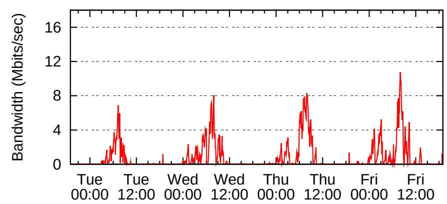
(a) Traffic shaping at Ohio only



(b) Traffic shaping at Wisconsin only



(c) Traffic shaping at Ohio and Wisconsin



(d) Available end-to-end bandwidth with both sites traffic shaping

Figure 4.3: Bandwidth used by a long flow between Wisconsin and Ohio over time: The available bandwidth is significant lower when both sites traffic shape compared to traffic shaping at only one site.

When both traffic shapers are active, the flow's instantaneous throughput is bounded by the minimum bandwidth available at either of the two traffic-shaped links. This could explain the decrease in bandwidth when using multiple traffic shapers. Figure 4.3 (d) confirms this hypothesis. At any given time, it shows the minimum bandwidth available at either of the two traffic shapers. This plot matches the bandwidth received by the TCP flow traversing both traffic shapers (shown in Figure 4.3 (c)) very well, showing that this decrease in performance is fundamental to traffic shaping and not due to some TCP inefficiency.

To quantify the loss in the end-to-end performance of the flow, we show in Table 4.1 the time required to complete data transfers of varying size when traffic shapers at Ohio and Wisconsin are operating in isolation and when both are active simultaneously. For example, transferring 15GB of data (the size of a high-definition DVD) takes at most 1.2 days when only one traffic shaper is active, but twice as long when both traffic shapers are operating.

4.2 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

Size	Time Required		
	Ohio	**Wisc.**	**Both**
4GB	9.8 hrs	9.6 hrs	13 hrs
10GB	12.1 hrs	12.7 hrs	1.5 days
15GB	13.4 hrs	1.2 days	2.4 days
30GB	1.3 days	1.7 days	3.5 days

Table 4.1: Completion time for different transfer sizes: Transfers take much longer to complete with two traffic shapers along the path than in the cases with just one traffic shaper.

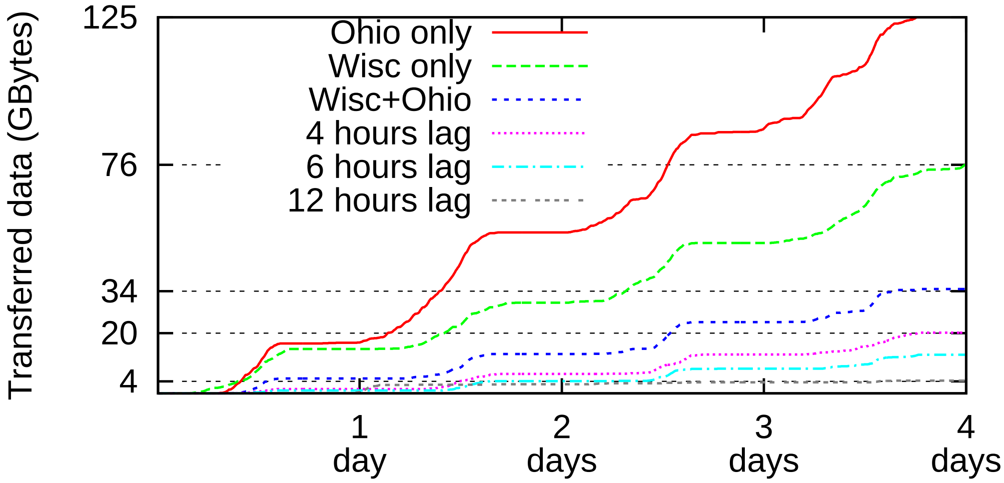


Figure 4.4: Data transferred by a long-running bulk flow: When traffic shapers are located in distant time zones, the performance of bulk data transfers can diminish by a factor of 20.

each other, their night and day time cycles may get out of phase. This can cause a severe decrease in the end-to-end performance of passing bulk flows. For example, previously we observed that each traffic shaper unchokes background flows between 0 and 6 AM local time (Figure 4.3 (a) and (b)) as this is the time with the lowest traffic load for both sites. But if the two traffic shapers are located in distant time zones the end-to-end path could potentially be subject to throttling for most of the day with the period of high load at one site coinciding with the period of low load at the other site.

In our previous simulation, the traffic shapers in Ohio and Wisconsin are located in proximate time zones. As a result, both traffic shapers operate in close synchrony, allowing bulk flows to achieve their maximum throughput between midnight and early morning. We investigated the loss in performance as they are separated by an increasing number of time zones. To account for the time lag between the two traffic shapers, we replayed the trace at the distant location with the appropriate time offset.

We set up a long-running flow that transfers data over the duration of 4 days. Figure 4.4 plots the amount of data we were able to transfer for different time zone distances of the two traffic shapers. As a point of comparison we also plot the amount of data transferred with only one traffic shaper being active.

The plot shows that the data transferred decreases sharply as the time lag increases. When the traffic shapers are 12 hours apart, the diurnal patterns are completely out of phase resulting in minimal performance; less than 5GB were transferred during the 4 days of the simulation. In contrast, 34GB were transferred when the two traffic shapers are proximate to each other. Note, that taken individually, each traffic shaper can transmit at least 76 gigabytes of data. This suggests that the end-to-end performance of bulk flows can slow down by a factor of 20 when the path goes over two traffic shapers in distant time zones.

Finally, all of our experiments consider the simultaneous negative impact of only two out-of-phase traffic shapers. Real flows are likely to traverse even more (distant) traffic shapers as the typical Internet path length spans $4-5$ inter-AS links. Thus, the results in this section are likely a conservative estimate of the global impact of widespread traffic shaping.

5 Related work

There are a large number of studies on the composition of traffic in academic [32], residential [12], and backbone networks [5,24]. They all find that bulk data transfers account for the majority of the bytes transmitted over the Internet today. Applications accounting for these bulk transfers include peer-to-peer (P2P) systems like BitTorrent [7], remote backup services [3], online music and movie stores [4], and transfers of large scientific repositories [36]. The soaring popularity of video streaming [37, 40] and the recent move of movie rental companies like Netflix [28] to ship movies over the Internet means even more bandwidth will be consumed by bulk flows moving forward.

This deluge of bytes is not without consequences for bandwidth costs. Lower-tier ISPs are already starting to peer with each other [29] to reduce transit costs and content providers are deploying P2P techniques to distribute their bandwidth costs [23, 37]. These observations support the economic considerations motivating our work.

Traffic shaping to save bandwidth cost is already happening. Companies like Packeteer [31] offer middleboxes to identity and throttle bandwidth-hungry applications. Many access networks openly admit to throttling network traffic [6, 15]. Existing traffic shapers typically do not account for the diurnal patterns of traffic and simply continuously limit applications. Our traffic shaper on the other hand exploits diurnal variations to make intelligent use of available resources. The techniques we use to schedule bulk flows are similar to the well-known multilevel feedback queue scheduling algorithms used in OS schedulers [35].

There is a large body of work on network architectures that support a differentiated treatment of traffic classes [8, 10, 33]. Their focus is either on giving traffic certain high quality service guarantees, or to implement a lower service class to make use of spare capacity. While well intended, none of these architectures is widely deployed. The focus of our traffic shaper takes inspiration from these efforts, offering a lower service class for bulk data traffic. This service class minimally impacts bulk transfer performance while reducing peak bandwidth consumption.

6 Discussion and Conclusions

This paper explores the implications of the confluence of a number of recent trends. First, ISPs charge their customers based on their peak levels of utilization because they must internally provision their network for these peaks. Second, there is significant diurnal variation in bandwidth demand within individual ASs, a factor of 2 or more in traces we considered. Third, the vast majority of bytes are consumed by large flows. For instance, we found that 70% of bytes were consumed by flows larger than 10MB.

These trends taken together have led many ASs to rate limit subsets of their traffic to reduce their peak levels of utilization, and to correspondingly reduce their bandwidth costs. Existing techniques act as blunt instruments, arbitrarily restricting or shutting down entire application classes without considering the reduction to peak bandwidth consumption or the effect on applications. For example, rate limiting peer to peer traffic in the middle of the night is unlikely to reduce the 95th percentile in bandwidth consumption.

In this paper, we show how ISPs can take advantage of the wide variation between peak and average case utilization to effectively smooth bandwidth consumption over the day. Our techniques do not starve any flows and only moderately delay the completion time of bulk flows. Taken together, our proposed traffic shaping techniques hold the promise of significantly reducing peak bandwidth utilization, e.g., by a factor of two or more, with no impact on interactive traffic and only minimal slowdown for non-delay sensitive bulk transfers.

These benefits suggest that an increasing number of ASs will employ these techniques to reduce their bandwidth costs. Perhaps unexpectedly, we find that what appears to be near-optimal local traffic shaping policies may lead to global ruin for bulk flows. As more ASs perform traffic shaping, the probability continually rises that some AS between a source and destination is rate limiting bulk flows at a particular point in time. Interestingly, the bandwidth available to a bulk transfer decreases with the physical distance that the flow travels for two reasons. First, longer physical paths typically imply that more ASs will be responsible for data transport. Second, variation in time zones means that peak times in different ASs will be out of phase, making it more likely that at least one AS is currently throttling traffic.

If the trends predicted in this paper hold, many bulk transfers will essentially receive no bandwidth. This limitation would come at a time when the demand for bulk transfers is exploding, consider high-definition video downloads or large scientific data sets. In this context, we will require alternative bulk-transfer architectures that at least consider the economic incentives that led to the traffic shaping in the first place.

One scenario is for ISPs to stop charging for peak levels of utilization but to instead adopt a different pricing model, e.g., per byte accounting. Unfortunately, such charging is likely to result in additional imbalances because it does not recognize that "all bytes are not created equal". Not encouraging data to be sent during times of otherwise slack usage means that network resources that must still be provisioned for peak demand sit idle. More importantly, per-byte charging would introduce even larger incentives for ASs to more aggressively traffic shape bulk traffic.

Another approach would be to develop an alternative, incentive-compatible protocol for bulk transfers. While such a protocol is beyond the scope of this paper, we outline some high-level possibilities. First, we observe that bulk transfers may still perform well as long as they are subject to traffic shaping by only a single AS. Next, bulk transfers do not require much of the semantics of TCP, e.g., in order delivery or synchronous end-to-end data acknowledgment. Finally, we take inspiration from delay-tolerant networks [18] and postal networks that stage the delivery of transfers from point to point in the network. For instance, postal networks often take advantage of capacity as it becomes available to move data across the network. Similarly, delay tolerant networks leverage in-network storage to buffer data until connectivity becomes available. One could imagine analogs where data is buffered in network until traffic throttling abates.

Overall, we find interesting tradeoffs and opportunities from the ever-increasing demands placed by bulk transfers on the Internet infrastructure when viewed in light of current pricing incentives. These flows display unique characteristics, certainly relative to the use scenarios originally targeted by TCP and IP. While alternatives to TCP have not seen widespread deployment when promising somewhat improved performance, we feel that novel protocols supporting bulk transfers are much more likely to be adopted when they promise significantly reduced cost.

Bibliography

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

[2] M. Allman, V. Paxson, and W. Stevens. TCP Congestion Control. RFC 2581 (Proposed Standard), Apr. 1999. Updated by RFC 3390.

[3] Amazon Simple Storage Service. `http://www.amazon.com`.

[4] Apple iTunes Store. `http://www.apple.com/itunes/store/`.

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

[6] Bandwidth restrictions save almost $1 million, Oct 2002. `http://thedaily.washington.edu/2002/10/22/bandwidth-restrictions-save-almost-1-million/`.

[7] BitTorrent homepage. `www.bittorrent.org`.

[8] 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.

[9] Blizzard Entertainment. `http://www.blizzard.com/`.

[10] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. RFC 1633 (Informational), June 1994.

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

[12] 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.

[13] K. Claffy, H.-W. Braun, and G. Polyzos. A Parametrizable Methodology for Internet Traffic Flow Profiling. *IEEE JSAC*, 1995.

[14] B. Claise. Cisco Systems NetFlow Services Export Version 9. RFC 3954 (Informational), Oct. 2004.

[15] Packet forgery by isps: A report on the comcast affair. `http://www.eff.org/wp/packet-forgery-isps-report-comcast-affair`.

[16] Comcast PowerBoost press release, Jun 2006. `http://www.cmcsk.com/phoenix.zhtml?c=147565&p=irol-newsArticle&ID=890297`.

[17] Y. Elovici, Y. Ben-Shimol, and A. Shabtai. Per-packet pricing scheme for IP Networks. In *Proc. of 10th ICT*, Feb 2003.

[18] K. Fall. A Delay Tolerant Networking Architecture for Challenged Internets. In *Proc. of SIGCOMM*, 2003.

[19] J. Feigenbaum, C. Papadimitriou, R. Sami, and S. Shenker. A bgp-based mechanism for lowest-cost routing. *Distrib. Comput.*, 18(1):61–72, 2005.

[20] Google helps terabyte data swaps, Mar 2007. `http://news.bbc.co.uk/2/hi/technology/6425975.stm`.

[21] J. Gray, W. Chong, T. Barclay, A. Szalay, and J. vandenBerg. TeraScale SneakerNet: Using Inexpensive Disks for Backup, Archiving, and Data Exchange. Technical Report MSR-TR-2002-54, Microsoft Research, May 2002.

[22] 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.

[23] C. Huang, J. Li, and K. W. Ross. Can Internet Video-on-Demand be Profitable? In *Proc. of SIGCOMM'07*, 2007.

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

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

[26] M. S. Kodialam, T. V. Lakshman, and S. Mohanty. Runs based traffic estimator (rate): A simple, memory efficient scheme for per-flow rate estimation. In *Proc. of INFOCOM*, 2004.

[27] R. Morris. Tcp behavior with many flows. In *Proc. of ICNP '97*. IEEE Computer Society, 1997.

[28] Netflix Online Movie Rental. `http://www.netflix.com`.

[29] W. B. Norton. The Evolution of the U.S. Peering Ecosystem, 2003. white paper.

[30] A. M. Odlyzko. Internet Traffic Growth: Sources and Implications. In *Proc. Optical Transmission Systems and Equipment for WDM Networking II*, 2003.

[31] Packeteer. `http://www.packeteer.com`.

[32] S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy. An Analysis of Internet Content Delivery Systems. In *Proc. of OSDI*, 2002.

[33] QBone Scavenger Service. `http://qbone.internet2.edu/qbss/`.

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

[35] G. Silberschatz, Galvin. *Operating System Concepts, 7th Edition.*

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

[37] Vudu Box. `http://www.vudu.com`.

[38] Windows Marketplace. `http://www.windowsmarketplace.com/`.

[39] X. Yang. Nira: a new internet routing architecture. In *Proc. of FDNA '03*, Karlsruhe, Germany, 2003.

[40] YouTube. `http://www.youtube.com/`.