Delay Tolerant Bulk Data Transfers on the Internet

Nikolaos Laoutaris
Telefonica Research
nikos@tid.es

Georgios Smaragdakis *
Deutsche Telekom Labs/TU Berlin
georgios.smaragdakis@telekom.de

Pablo Rodriguez
Telefonica Research
pablorr@tid.es

Ravi Sundaram †
Northeastern University
koods@ccs.neu.edu

ABSTRACT

Many emerging scientific and industrial applications require transferring multiple Tbytes of data on a daily basis. Examples include pushing scientific data from particle accelerators/colliders to laboratories around the world, synchronizing data-centers across continents, and replicating collections of high definition videos from events taking place at different time-zones. A key property of all above applications is their ability to tolerate delivery delays ranging from a few hours to a few days. Such *Delay Tolerant Bulk* (DTB) data are currently being serviced mostly by the postal system using hard drives and DVDs, or by expensive dedicated networks. In this work we propose transmitting such data through commercial ISPs by taking advantage of already-paid-for off-peak bandwidth resulting from diurnal traffic patterns and percentile pricing. We show that between sender-receiver pairs with small time-zone difference, simple source scheduling policies are able to take advantage of most of the existing off-peak capacity. When the time-zone difference increases, taking advantage of the full capacity requires performing store-and-forward through intermediate storage nodes. We present an extensive evaluation of the two options based on traffic data from 200+ links of a large transit provider with PoPs at three continents. Our results indicate that there exists huge potential for performing multi Tbyte transfers on a daily basis at little or no additional cost.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Store and Forward Networks; C.2.2 [**Network Protocols**]: Routing Protocols.

General Terms

Economics, Measurement, Performance.

Keywords

Delay Tolerant Networks, 95-percentile Pricing, Content Distribution.

* Work conducted while the author was an intern at Telefonica Research.
† Work conducted while the author was a research visitor at Telefonica Research.

1. INTRODUCTION

Several important scientific and industrial applications require exchanging *Delay Tolerant Bulk* (DTB) data. For instance, CERN's Large Hadron Collider (LHC) is producing daily 27 Tbytes of particle collision data that need to be pushed to storage and processing centers in Europe, Asia, and North America. Google and other operators of large data-centers hosting cloud computing applications need to replicate and synchronize raw and processed data across different facilities. Rich media need to be transfered across time-zones as in the recent Beijing Olympic games in which large video collections needed to be replicated at US VoD servers before morning time. All the above mentioned data have delay tolerances that range from several hours (Olympic games) to a few days (LHC), *i.e.*, they are several orders of magnitude greater than the time scales of Internet traffic engineering and congestion avoidance. Depending on the application, DTB data are currently being serviced by either expensive dedicated networks like the LHC Computing Grid, or by the postal system using hard drives and DVDs.

ISPs and DTB traffic: In this work we examine the potential of sending DTB traffic over commercial ISPs that carry mostly residential and corporate TCP traffic that is not tolerant to long delays [2]. To handle the hard QoS requirements of interactive traffic, ISPs have been dimensioning their networks based on peak load. This is reflected in the *95-percentile pricing* scheme [9] used by transit ISPs to charge their customers according to (almost) peak demand. Therefore, access ISPs pay according to the few hours of peak load of their typical *diurnal variation pattern* [12, 11], in which the load peaks sometime between the afternoon and midnight, then falls sharply, and starts increasing again in the next day. Diurnal patterns combined with 95-percentile pricing leave large amounts of off-peak transmission capacity that can be used at no additional transit cost.

Our contribution: We propose using this already paid for off-peak capacity to perform global DTB transfers. Due to their inherent elasticity to delay, DTB transmissions can be shifted to off-peak hours when interactive traffic is low and thus (1) avoid increasing the transit costs paid at charged links under 95-percentile pricing, and (2) avoid negative impacts on the QoS of interactive traffic. Unlike previous network-layer proposals like the Scavenger service of Qbone [14], we follow an application-layer approach based on the regulation of transmission rates of DTB sources and assisting intermediate storage nodes (see the related work section for more on the Scavenger service).

We first consider End-to-End (E2E) transfers in which DTB data flow from a sender directly to a receiver over a connection-oriented session optimized for long lived bulk transfers (we assume that performance issues of TCP have been resolved using efficient implementations or multiple parallel connections [16]). An *E2E-CBR* policy of almost constant rate B/T can deliver volume B within deadline T. In the case of LHC data this would require a stream of at least 2.5 Gbps (27 Tbytes per day). Assuming that the transfer has to reoccur every day, E2E-CBR would push up the 95-percentiles of the sending and receiving access ISPs by exactly B/T=2.5 Gbps costing them anything between \$75K and \$225K in additional monthly transit costs (\$30K-90K per Gbps according to recent prices). In other words, since E2E-CBR is bound to increase the charged volume by exactly its mean rate, it provides no advantage compared to buying dedicated lines of the exact same rate.

A more prudent E2E approach is to perform scheduling at the sender and avoid, or slow down, transmissions during peak hours. An *E2E-Sched* policy can thus take advantage of "load valleys" during the off-peak hours of the sender and transmit DTB traffic without impacting the charged volume of the sending access ISP. The problem with this policy is that due to time-zone or traffic profile (residential/corporate) differences, oftentimes the off-peak hours of the sending ISP do not coincide in time with the off-peak hours of the receiving ISP. When such *non-coinciding valleys* occur, end-to-end transfers are unable to fully utilize the free capacity of both ends.

A natural approach for solving this problem is to perform Store-and-Forward (SnF) using an assisting storage node inside the transit ISP. Having transit storage allows an *SnF* transfer policy to buffer DTB traffic inside the network, and allow it to ride on top of multiple load valleys one by one, even if they do not coincide in time.

Summary of results: Our main contribution goes towards the improvement of our understanding of the performance comparison between E2E-Sched and SnF. Let $F(\mathcal{P})$ denote the maximum volume of DTB data that can be delivered for free by policy $\mathcal{P}$ between nodes v and u within a time allowance of T. Then if an application has to send a volume of B, our strategy would be as follows.

- if $B < F$(E2E-Sched), then E2E-Sched can send them for free. In that case there is no need to deploy storage inside the network.
- if F(E2E-Sched) $< B < F$(SnF) and the gap is wide enough, SnF can utilize relatively cheap network storage to send the data at zero transit cost.
- if $B > F$(SnF), SnF can utilize network storage to send the data at the smallest possible transit cost.

Evidently, the above guidelines depend on *"how wide the performance gap between E2E-Sched and SnF is?"*. To answer this question we quantify the comparison between the two policies by driving them with real background traffic from 200+ links of a large transit provider with PoP in three continents. The results indicate that:

- Over ISP links of 10-40 Gbps both policies can transfer in more than half of sender-receiver pairs anything between 10 and 30 Tbytes of DTB traffic on a daily basis, at no additional transit cost.
- The ratio between F(SnF)/F(E2E-Sched) stays close to 1 for time-zone differences $<$ 5 hours and then increases quickly to values above 2. For pair on opposite sides of the world, the ratio peaks to about 3.
- The above ratio also depend on the amounts of free capacity at the two endpoints. If only one is the bottleneck, then time-zone difference does not have a significant impact. SnF's gains peak for network of similar capacity at distant time-zones.

We add to our evaluation bandwidth prices and look at the cost of sending volumes that exceed the free capacity.

- For 50% of the pairs in TR, E2E-Sched has to pay in transit cost at least \$5K to match the volume that SnF sends at zero transit cost.
- We show that although for individual 27 Tbyte daily transfers a courier service is cheaper than SnF, things get reversed when having to service a continuous flow of data that repeats every day. In this case SnF amortizes the increased charged volume throughout a month, and thus achieves a lower daily transfer cost.

The remainder of the article is structured as follows. In Sect. 2 we present background information. In Sect. 3 we detail E2E-Sched and SnF. Sect. 4 goes to quantifying the volume of DTB traffic that can be sent for free by the two policies during one day. In Sect. 5 we show how to modify the policies to allow them to meet delivery deadlines at minimum transit cost. Is Sect. 6 we compare SnF against a courier service. In Sect. 7 we discuss potential reactions on the part of ISPs. In Sect. 8 we present related work and conclude in Sect. 9.

2. BACKGROUND

2.1 Network model

Consider a sender of DTB traffic v connected to an access ISP, ISP(v), and a receiver u connected to an access ISP, ISP(u). The two access ISPs communicate through a common transit provider TR who offers them transit service (Fig. 1). The charged links ISP(v) $\leftrightarrow$TR and ISP(u) $\leftrightarrow$TR are subjected to 95-percentile pricing as defined below.

Definition 1. *(95-percentile pricing) Let x denote a time series containing 5-minute transfer volumes between a customer and a transit provider in the duration of a* charging period, *typically a month. The customer pays the transit provider an amount given by a* charging function $c(\cdot)$ *that takes as input the* charged volume $q(x)$ *defined to be the 95-percentile value of x.*

To avoid unnecessary complications we will assume that each direction is charged independently. Therefore, a DTB flow from v to u may increase the percentile of the uplink ISP$(v)$$\rightarrow$TR and/or the downlink TR$\rightarrow$ISP(u), and thus create additional transit costs for the two access ISPs. For each charged link l we know its capacity C_l and its background load x_l generated by other clients of the access ISP. We assume that there aren't any bottlenecks inside TR. We also assume that there exists a *transit storage node* w inside TR that may be used for performing store-and-forward of DTB traffic. Since TR is bottleneck free, we don't need to consider placement issues of w, or multiple transit storage nodes.

We focus on the above network model because it is commonly encountered in practice and also because we have exact data to evaluate it, including traffic volumes, routing rules, and bandwidth prices. It is straightforward to generalize our methods to more than 2 charged links, but this is not very common in practice and we don't have data to evaluate it. Also, we could examine multi-homed ISPs, but this doesn't add much flexibility because the correlation between load and time-of-day would also exist in this case.

2.2 Mixing DTB & background traffic

Next we look at the problem of computing $F(C, x, t_0, T)$, the volume of DTB traffic that can be pushed through a single charged link of capacity C carrying background volume x in the interval $[t_0, t_0 + T)$ without increasing its charged volume $q(x)$. That much DTB traffic can be sent for free, *i.e.*, at no additional transit cost.

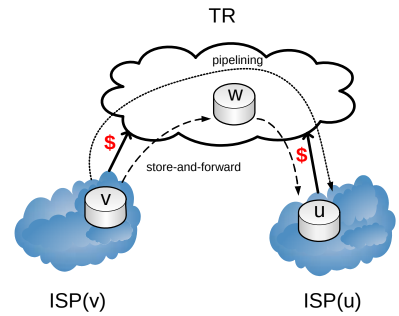


Figure 1: Sender v at ISP(v) and receiver u at ISP(u). 95-percentile pricing on links ISP(v) $\leftrightarrow$TR and ISP(u) $\leftrightarrow$TR. A transit storage node w inside the bottleneck-free transit provider TR to be used for store-and-forward of DTB flows from v to u.

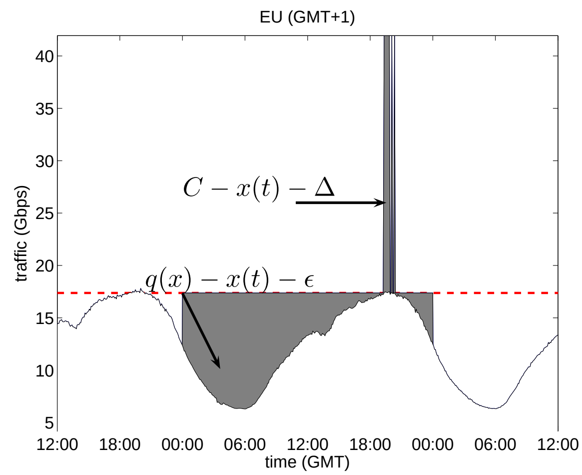


Figure 2: Water-filling on 40 Gbps link. For illustration purposes ϵ, Δ are set to 0.

In Fig. 2 we plot using a solid line the uplink traffic of one real access ISP during all 288 5-minute intervals of a day. We also mark with a dashed horizontal line the corresponding charged volume $q(x)$ based on the background traffic x of the entire month. The gray shaded area on the same picture indicates the extra amount of DTB traffic that can be mixed with the background by a simple *"water-filling"* strategy that does not increase the charged volume $q(x)$. The water-filing strategy is quite simple. At slot t, $t_0 \leq t \leq t_0 + 287$, it sends $f(C, x, t)$ additional DTB data, where:

$$f(C, x, t) = \begin{cases} q(x) - x(t) - \epsilon, & \text{if} \quad x(t) < q(x) \\ C - x(t) - \Delta, & \text{o.w.} \end{cases} \tag{1}$$

Thus overall

$$F(C, x, t_0, T) = \sum_{t=t_0}^{t_0+T-1} f(C, x, t) \tag{2}$$

All these additional transmissions come for free because the 95-percentile of the background plus the DTB traffic is again $q(x)$. Some observations to be made here: The ϵ in the first case is a "guard" margin for avoiding exceeding $q(x)$ due to estimation errors. In the second case, the injected traffic can be made Δ less than $C - x(t)$ to protect the QoS of the background traffic. We could permit up to little above 50% utilization during such bursts to be aligned with the common ISP practice of upgrading links when their average peak-hour utilization approaches 50%. This is done to preserve user QoS and allow the links to operate as backup in case of failure of other links. In practice, we don't use at all this extra burst capacity in our experiments, and stick to using only free capacity during times that the background is below the charged volume.

2.3 Traffic data from a large transit provider

We obtained detailed traffic load information for all the PoPs of a very large transit provider (TR) that is the sole transit provider for a large number of access ISPs, collectively holding more than 12 millions ADSL users, spread mostly in Europe, North, and South America. TR has peering agreements with 140 other large networks from all continents, as well as with most large creators (YouTube), distributors (Akamai, Limelight), and indexers (Google) of content. More specifically, our dataset includes information for 448 links with 140 ISPs. Out of those links we kept only 280 that have nominal capacities exceeding 1 Gbps and can thus support Tbyte-sized DTB transfers (several of them are 10 and 40 Gbps links). For each one of these links between TR and another peer or client network, we obtained several weeks worth of uplink and downlink load data aggregated over 5-minute intervals.[1] We geo-located all the links based on PoP and interface names. Our measurements reflect real traffic loads in TR during the first quarter of 2008. To make our study richer and cover more geographic areas we assumed that all links were paid links, although several are unpaid peerings.

3. BULK TRANSFER POLICIES

For a given transfer policy $\mathcal{P}$ we are interested to know $F(\mathcal{P})$, the volume of DTB traffic that can be pushed from v to u in the interval $[t_0, t_0 + T)$ without increasing the percentile of $x_{\text{ISP}(v)\to\text{TR}}$ and $x_{\text{TR}\to\text{ISP}(u)}$ (for simplicity x_v and x_u hereafter). We show how to compute this in the next two subsections. At the end we discuss some implementation issues.

3.1 End-to-End with source scheduling

Let's start by considering a transfer policy employing source scheduling at the sender to regulate the amount of DTB traffic that is sent to the received at each 5-minute slot over an end-to-end connection. We will refer to this policy as *E2E-Sched*. In Sect. 2.2 we saw how to compute $F(C, x, t_0, T)$ for a single charged link. We can apply a similar water-filling strategy with the only exception that we have to make sure that 5-minute DTB transmissions respect both $q(x_v)$ and $q(x_u)$. This is necessary because in the current case the end-to-end DTB flow "pipeliness" through both charged links (Fig. 1). The free capacity achieved by E2E-Sched is thus:

$$F(\text{E2E-Sched}) = \sum_{t=t_0}^{t_0+T-1} \min\Big(f(C_v, x_v, t), f(C_u, x_u, t)\Big) \tag{3}$$

If the volume of data to be transmit by E2E-Sched is $B < F(\text{E2E-Sched})$ then we can drive it using artificially smaller charged volumes $q_v < q(x_v)$ and $q_u < q(x_u)$ so as to force it to follow a smoother schedule than the one that achieves the maximum free capacity.

3.2 Store-and-Forward

Next we consider a store-and-forward policy that first uploads data from the sender v to the transit storage node w within TR, and then pushes them from w towards the final receiver u. We call this policy *SnF*. The transit storage node permits SnF to perform independent water-fillings in the two charged links $\text{ISP}(v) \to \text{TR}$ and $\text{TR} \to \text{ISP}(u)$, minding to respect in each case only the local charged volume. As a result, SnF has much more freedom than E2E-Sched that is constrained by the min operator of Eq. (3) that applies to the charged volumes and background traffic of both links. Consequently, SnF can be uploading from v to w faster than what w can be pushing to u. The difference between the two rates accumulates at w and starts draining once more free capacity becomes available on the charged downlink $\text{TR} \to \text{ISP}(u)$. We can compute $F(\text{SnF})$ with a simple iteration staring with $F(\text{SnF}, t_0) = 0$.

$$F(\text{SnF}, t) = F(\text{SnF}, t-1) + f(t), \quad t_0 \le t < T \tag{4}$$

$$f(t) = \begin{cases} f(C_u, x_u, t), \text{ if } & f(C_u, x_u, t) < f(C_v, x_v, t) \\ f(C_v, x_v, t) + \min(f(C_u, x_u, t) - f(C_v, x_v, t), b_w(t-1)), & \text{o.w.} \end{cases}$$

[1] We do not currently have clearance to release the above dataset, but we are considering possible ways to allow reproducing some of our results without compromising privacy. Until we have a solution to this difficult problem, an interested reader can easily perform a sanity check by creating a synthetic diurnal pattern (*e.g.*, using a trigonometric function) and generating sender-receiver pairs using the original form and a shifted version of itself.

where $b_w(t)$ denotes the buffer occupancy at the storage node w at time t. Again this can be computed iteratively starting with $b_w(t_0) = 0$.

$$b_w(t) = \begin{cases} b_w(t-1) + f(C_v, x_v, t) - f(C_u, x_u, t), \\ \qquad\qquad \text{if} \quad f(C_v, x_v, t) > f(C_u, x_u, t) \\ \\ b_w(t-1) - \min(f(C_u, x_u, t) - f(C_v, x_v, t), b_w(t-1)), \\ \qquad\qquad \text{o.w.} \end{cases}$$

All that Eq. (4) is saying is that the amount of data delivered to the receiver u increases at time t by the free capacity on the downlink, if the downlink is the bottleneck during t, or by the free capacity on the uplink, augmented with an additional amount drained from the buffer of the storage node w.

3.3 Implementing E2E-Sched and SnF

Implementing the two policies requires knowing the basic information for performing water-filling on links, *i.e.*, the load in the next 5-minute interval, and the charged volume of the month. The first one is easy to predict since the load of successive slots is highly correlated. The second one is also possible to predict by accumulating data from the current charging period, or using last month's if still at the beginning. These two methods were tested and shown to be quite accurate in [19]. We will use them during our evaluations.

4. HOW MUCH CAN WE SEND FOR FREE?

Next we compare the volumes of data that can be delivered for free by the two policies.

4.1 Methodology

We consider the standard network model of Sect. 2.1 involving two charged links. For all our experiments we set the deadline T to 1 day. We do this because due to the 24 hour period of diurnal patterns, all our performance results also become periodic and we take advantage of this to avoid having the starting time t_0 affecting the comparison between pairs that have different time-zone differences. We obtain load time series x_v, x_u and capacities C_v, C_u for 280 links with capacity higher than 1 Gbps from our dataset introduced in Sect. 2.3. For each sender-receiver pair we use the results of Sect. 3 to compute $F(\mathcal{P})$ for a day. We repeat the experiment for all the working days of a week and report median values.

4.2 Overview of E2E-Sched vs. SnF

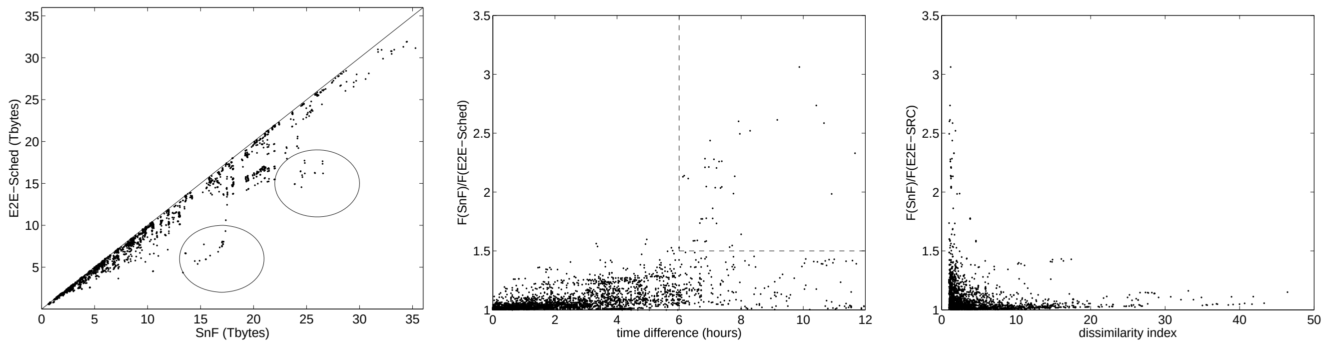
For each possible sender-receiver pair we compute F(E2E-Sched) and F(SnF) as explained in Sect. 4.1. In Fig. 3(a) we plot the volume of DTB data that can be delivered for free by E2E-Sched versus the corresponding volume by SnF for all sender-receiver pairs. We observe that daily free capacities in the range of 10-25 Tbytes are quite frequent, verifying our intuition that there exists huge potential for pushing DTB traffic during off-peak hours. Most of the pairs are closely below the 100%-diagonal, indicating that E2E-Sched matches the performance of SnF in these cases, but there are also several cases in which the performance of the two policies diverges substantially. We annotate the figure to point to some of these cases.

4.3 Looking deeper

We now want to understand in which cases the two policies perform the same and in which they diverge. This is an important question to answer for deciding whether it is worth providing a given pair v, u with transit storage or not. In Fig. 3(b) we plot the ratio F(SnF)/F(E2E-Sched) against the time-zone difference between the sender and the receiver for each one of our pairs. There is a clear correlation between the ratio and the time-zone difference. Also, we see a sudden increase of the ratio after 5-6 hours of time-zone difference. This is due to the fact that TR connects several ISPs on the two sides of the Atlantic. Pairs with even higher ratios (above 2) have one end-point in Asia and the second in Europe or America. Overall, although the density of points across the x-axis depends on the particular properties of TR, like its geographic and ethnic coverage (which may differ vastly across providers), the values on the y-axis verify our basic intuition that *the performance gain of store-and-forward increases with the appearance of non-coinciding off-peak hours, which in turn correlates with large time-zone difference.*

A large time zone difference, however, is not the only pre-requisite for a high F(SnF)/F(E2E-Sched) ratio. Observe for example that in the ranges of 6-7 and 11-12 hours where high ratios appear, there still exist pairs with rather low ratios. To understand this, we need to notice that *non-coinciding off-peak hours bias the ratio only when the individual off-peak capacities of the two end-points are comparable.* By off-peak capacity we mean the volume F from Eq. (2) that local water-filling can get through the link in one day, without caring about the other end-point. When one end-point has a much lower off-peak capacity, it means that either its link speed is much lower (*e.g.*, one is a 10 Gbps link and the other is 40 Gbps) or its utilization is much higher (*e.g.*, one having peak hour utilization 50% and the other less than 15%). In such cases, the link with the smaller off-peak capacity is always the end-to-end bottleneck independently of time-zone difference. Transit storage becomes useful when the bottleneck shifts from one end to the other.

To verify the above point, we define the *dissimilarity index* to be the ratio between the larger and the smaller off-peak capacity of a pair. Smaller values of the index indicate links with comparable off-peak capacities. In



(a) Daily free volume, E2E-Sched, SnF. (b) Free volume ratio vs time-zone diff. (c) Free volume ratio vs dissimilarity

Figure 3: Results on $F(\mathbf{SnF})$, $F(\mathbf{E2E\text{-}Sched})$, and their ratio for links with capacity >1 Gbps.

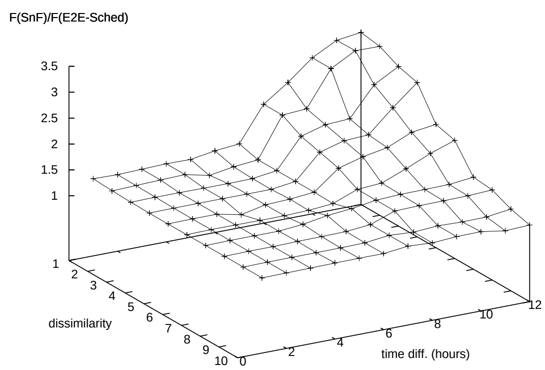


Figure 4: Summary of the effects of time-zone difference and dissimilarity index on the ratio $F(\mathbf{SnF})/F(\mathbf{E2E\text{-}Sched})$.

Fig. 3(c) we plot again the $F(\text{SnF})/F(\text{E2E-Sched})$ ratio but this time against the dissimilarity of a pair. The figure shows that high ratios occur with dissimilarity close to 1. In Fig. 4 we plot in 3-D the ratio against the dissimilarity index and the time-zone difference of pairs, summarizing our observations that *in the case of TR store-and forward becomes worthwhile in pairs of similar capacity and utilization that have at least 5 hours of time-zone difference.*

5. THE COST OF DEADLINES

What happens if an application requires transmitting within a deadline of length T a volume B that is greater than what E2E-Sched, or even SnF can deliver for free? Then either policy will have to transmit with rates that will increase the charged volume of the uplink, downlink, or both. In this section we first show how to modify the two policies to allow them meeting specific deadlines with minimum additional transit cost. Then we return to our traffic data from TR and compute actual transit costs for Tbyte-sized DTB transfers under current bandwidth prices.

5.1 Meeting deadlines with E2E-Sched and SnF

Suppose that in a charged link with capacity C and background traffic x we perform standard water-filling (Eq. (1)) but with a higher charged volume $q > q(x)$ given by the extra transit cost $c(q) - c(q(x))$ we are willing to pay for transmitting our DTB data on top of the existing background. Then as in Eq. (2), we can compute $B(q, C, x, t_0, T)$, the maximum volume we can ship under the new (non-zero) transit cost.

Using the above simple idea we can modify a transfer policy $\mathcal{P} \in \{\text{E2E-Sched,SnF}\}$ to allow it to deliver within deadline T volumes $B \geq F(\mathcal{P})$ at minimum extra transit cost $\mathcal{C}(\mathcal{P}, B)$. The approach is similar for both policies. It comes down to solving the following optimization problem.

Definition 2. *(min-cost transfer) Find charged volumes $q_v \geq q(x_v)$ and $q_u \geq q(x_u)$ to minimize the extra transit cost $\mathcal{C}(\mathcal{P}, B) = c_v(q_v) - c_v(q(x_v)) + c_u(q_u) - c_u(q(x_u))$, subject to constraint $B(\mathcal{P}, q_v, q_u) = B$.*

$B(\mathcal{P}, q_v, q_u)$ denotes the maximum volume of DTB data delivered by $\mathcal{P}$ to the receiver u by $t_0 + T$ without exceeding charged volumes q_v, q_u. It can be computed as follows: For E2E-Sched all we need to do is repeat the computation of $F(\text{E2E-Sched})$ from Sect. 3.1 substituting $q(x_v)$ and $q(x_u)$ with q_v and q_u, respectively. Performing the same substitution we can repeat the computation of $F(SnF)$ from Sect. 3.2 and obtain $B(\text{SnF}, q_v, q_u)$. It's easy to see that we can solve the min-cost problem in polynomial time even with an exhaustive search that will examine the cost of all the combinations of q_v, q_u, within some basic search quantum δq starting from the minimum values $q(x_v)$ and $q(x_u)$ and going up to the maximum charged volumes allowed by the capacities C_v, C_u of the two links. In practice we use a faster greedy search that assigns δq to the link that returns the biggest increase to $B(\mathcal{P}, q_v, q_u)$

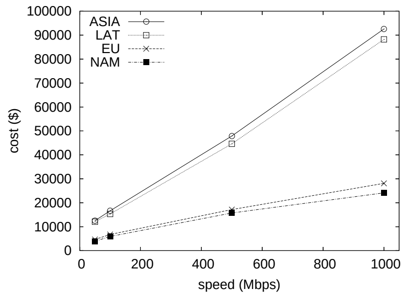


Figure 5: Bandwidth prices for different access speeds at the PoP's of TR as of 2008.

per dollar paid. It is easy to see that for $\delta q \to 0$, the above greedy converges to an optimal solution.

In terms of implementation there exists one significant difference with the corresponding versions of Sect. 3 that required predicting only the next 5-minute volume and the monthly charged volume. In the current version we need to estimate before initiating the transmission all $x(t)$ for $t \in [t_0, t_0 + T)$. This is necessary[2] for solving the min-cost transfer problem of Definitition 2 and getting q_v and q_u based on which the water-filling is performed. The approach we follow for this is very simple. We use as prediction of future $x(t)$'s the corresponding values from the same day of the previous week. It is well known that at multi-gigabit speeds the aggregated volumes are fairly stable across successive days and weeks [12, 11], something that applies also to our own traffic data. In all our experiments, optimizing based on such past data produced transmission schedules with charged volumes that were at most 1-2% off from the charged volumes we would get had we known the precise future traffic in $t \in [t_0, t_0 + T)$. Granted that charging functions are linear or concave-like, this does not affect the transit cost by more than 1-2%.

5.2 Wholesale monthly bandwidth prices

To be able to perform cost comparisons we surveyed the price of wholesale bandwidth at the geographic areas of PoPs that appear in our traffic dataset using multiple publicly available resources like the NANOG mailing list.[3] In Fig. 5 we give an overview for different areas. A high level summary is that transit bandwidth has similar price in Europe and North America , where it is almost 4 times cheaper than in Latin America, and certain high demand areas in Asia. We will use these values later as charging functions that take as parameter the 95-percentile of the combined background and DTB traffic of a link.

[2] More precisely, these $x(t)$'s are needed for being able to check the constraint $B(\mathcal{P}, q_v, q_u) = B$ while searching for the optimal charged volumes.

[3] http://merit.edu/mail.archives/nanog/2004-08/msg00269.html.

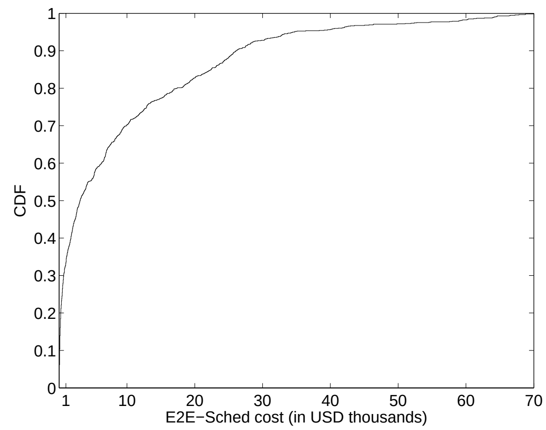


Figure 6: Transit cost paid by E2E-Sched to match the volume that SnF delivers for free.

5.3 The price of $F(\text{SnF}) - F(\text{E2E-Sched})$

Since SnF can use the storage node to push more data for free than E2E-Sched in the same duration T, an interesting question is, *"How much does it cost to send with E2E-Sched the same volume of data that SnF can send at zero transit cost?"*. We computed this cost for all the pairs of our dataset from TR that have at least 20% peak-hour utilization. We did this because our dataset includes some backup links that are empty. These links have no diurnal pattern (and thus free off-peak hours) and thus any traffic added to them increases immediately the cost just like buying a dedicated link. We plot the resulting cdf in Fig. 6. From this we can see that *for 50 percent of the pairs in TR, E2E-Sched has to pay a transit cost of at least $5K to match the volume that SnF sends at zero transit cost.* Of course, SnF, needs to use the transit node w and that introduces some additional costs that we discuss next.

5.4 The cost of the storage node

From the results appearing in Fig. 6 we selected a pair with the sender in Europe (EU) and the receiver in Latin America (LAT). The 5 hours of time-zone difference create in this case a substantial misalignment between the off-peak hours of the sender and the receiver. This reflects on the performance comparison between SnF and E2E-Sched. For this pair, $F(\text{SnF})$=24 Tbytes and $F(\text{E2E-Sched})$=15 Tbytes and thus E2E-Sched has to pay a substantial additional transit cost ($60K) if it is to match the capacity of SnF (notice that bandwidth prices at LAT are 3-4 times higher than at EU, Fig. 5). This makes the particular pair candidate for deploying an SnF solution. Our objective is to estimate the amount of storage that SnF requires for achieving the superior capacity, and then do a back of the envelope calculation of the cost of deploying that much storage

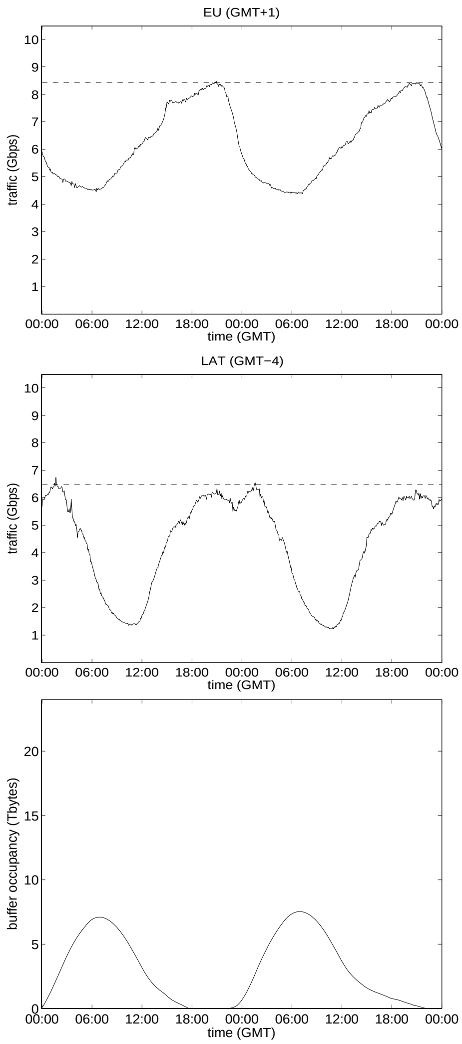


Figure 7: A DTB transfer from EU to LAT. Top, x_v (C_v=40 Gbps). Middle, x_u (C_u=10 Gbps). Bottom, buffer occupancy at the transit storage node w. Charged volumes indicated by horizontal lines.

and see whether it is justified given the transit cost paid by E2E-Sched.

To follow the example closely we plot on the top row of Fig. 7 $x_v(t)$, in the middle row $x_u(t)$, and on the bottom one $b_w(t)$, the buffer occupancy at the storage node w. Notice now that although F(SnF) is 24 Tbytes, the maximum buffer capacity required at w to bypass the non-coinciding off-peak hours between x_v and x_u is only 5 Tbytes (*i.e.*, around 20% of F(SnF)). This happens because w is used for absorbing rate differences between the two charged links, and thus in most cases it doesn't have to store the entire transferred volume at the same time.

With retail storage costing no more than $300 per Tbyte and adding the cost of the server, the capital cost of w cannot exceed $10K. Assuming conservatively that the server's lifetime is 2 years, the amortization cost comes to around $400 per month. Doubling this to amount for maintenance brings the final cost of SnF to less than $1K which is still way smaller than the $60K of E2E-Sched. Remember that from Fig. 6 we know that E2E-Sched is paying a median of $5K for the same volume that SnF delivers for free. Combining this with the results of Sect. 4 indicates that *if the amount of data to be pushed is less than what E2E-Sched delivers for free then E2E-Sched is the obvious choice as it doesn't need the transit storage node. Otherwise, the amortized cost of the storage node is quickly masked by bandwidth transit costs of E2E-Sched and thus SnF becomes a favorable option.*

6. SNF VS. A COURIER SERVICE

In this section we attempt to make a rough comparison between the transit cost of sending DTB data using SnF and the shipment cost of sending them in physical form using courier services. We will omit capital costs that we believe to be secondary, *e.g.*, the cost of purchasing storage nodes, or the cost of purchasing hard disks to ship with the courier service. We will also omit operating costs that may be more substantial, but we cannot evaluate easily, *e.g.*, the cost in personnel for maintaining a supply chain of disks (filling them with data, mounting/unmounting, passing them to the courier company, *etc.*).

6.1 Overview

Our high-level comparison is summarized in Fig. 8. To begin with, there exist sender-receiver pairs (v, u) that usual courier systems cannot service within deadline T. For example, destinations in different continents and deadlines smaller than one day. SnF has a clear win here since as shown earlier it can transfer huge amounts of data within a day. Now if the courier system can meet the deadline, then if one lets the DTB volume B grow too much and, *e.g.*, exceed the maximum transmission capacity of the network during T, then obviously SnF cannot do much whereas the courier can in principle fill a plane or a ship with hard disks and send them over. Returning to the more realistic and interesting case of jobs that both SnF and the courier system can service, we notice that, as shown before, there are many cases in which SnF can send the data at zero transit cost. Again, this is a clear win for SnF since the courier system will have to charge a non-zero shipment cost. Finally, it can be that SnF also charges some non-zero cost. For this case, we show a detailed example to support that, contrary to conventional wisdom, the courier is not cheaper if the flow of data is continuous.

6.2 Sending 27 Tbytes from EU to LAT

We return to the example of Fig. 7 in which SnF

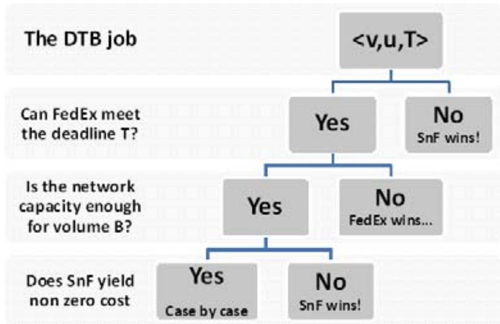


Figure 8: SnF vs. FedEx.

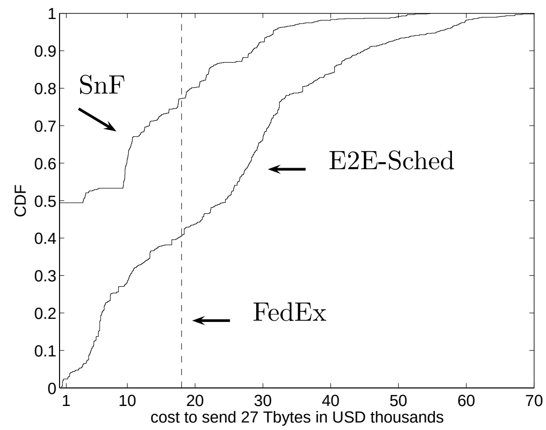


Figure 9: The cost of sending 27 Tbytes.

was able to push daily just over 24 Tbytes from EU to LAT for free. If we demand from SnF to carry an additional 3 Tbytes daily to, *e.g.*, match the 27 Tbytes of daily production from LHC, then using the methods of Sect. 5 we get that SnF will increase the monthly transit cost by less than \$10K. Notice that this example is a particularly bad one for SnF since bandwidth prices in LAT are quite high as shown in Fig. 5. In summary, by paying less than \$10K per month, SnF can be sending 27 Tbytes every day from EU to LAT.

Let's see now how much it costs to perform the same using a courier service. We learned from the web site of FedEx that from EU to LAT deliveries take 2-4 days. We will assume that they can be completed in 2 days. Then a courier would have to deliver every 2 days a shipment carrying 27·2=54 Tbytes. Assuming that hard drives are used for carrying the data, it would require 54 1TByte disks. Assuming that each disk weights around 1.25Kgr (Hitachi H31000U), the resulting shipment would weight at least 68 Kgr (excluding packaging). We checked the cost that FedEx quotes on its web-site for the exact two cities in our experiment and for this weight and it turned out to be around \$1200. Multiplying this by 15 to cover a month, the final price is \$18K, *i.e.*, much higher than the monthly transit cost of SnF which was less than \$10K. The above is of course just a back of the envelope calculation, but it is useful in demonstrating that the common perception that physical delivery is always cheaper ceases to apply when taking advantage of free off-peak network capacity.

6.3 LHC data among other pairs of TR

Let's now see how much it costs to send 27 Tbytes in other pairs in TR. We kept links with capacity $>$ 10 Gbps and peak hour utilization $>$ 20%. In smaller links either the data didn't fit, or cost too much because there were no load valleys to take advantage of. In Fig. 9 we plot the cdf of the transit cost of delivering 27 Tbytes in 1 day with E2E-Sched and SnF for all the aforementioned pairs. For reference we also draw a horizontal line at \$18K to point to the previously computed indicative cost of FedEx (we verified that this cost did not vary much among different pairs). One can see that *38% of pairs achieve lower cost than FedEx using E2E-Sched, whereas the corresponding percentage using SnF is 70%!*

In conclusion, whereas for a single shipment the courier service is indeed cheaper, it stops being cheaper when considering a continuous flow of data. The courier also suffers from "packetization delays", *e.g.*, it takes 2+2 days from the creation of a bit up to its delivery, whereas SnF sends most of its bits instantly (with some minor delaying of some bits on the storage node). Also, in the case of multicast delivery to n receivers, SnF halves its cost as it pays the uplink cost only once for all n receivers. The courier being a "point-to-point" service cannot save in this case. Lastly, it should be pointed that up to now we have been conservative and used work-day background traffic which is rather high compared to weekend traffic. If transfers can wait until weekends, SnF can gain even more by exploiting the low weekend traffic.

7. DISCUSSION

In view of the large potential for low cost DTB transfers demonstrated in the previous sections, an important question is *"Whether transit ISPs will maintain 95-percentile pricing in view of DTB transfers?"*. This is a complicated question to answer. Next we make some inital observations.

(1) The potential of SnF would disappear if transit ISPs switched to pricing based on the total aggregate volume of a month. This, however, does not seem very likely to happen as it goes against basic network economics dictating that the cost of building and maintaining a network is given by the peak traffic that it has to support [5]. For example, such a switch would allow clients to transmit at high peak rates and still pay small amounts, as long as they keep their aggregate monthly volumes small. This is problematic as it

requires dimensioning the network for high peak rates, without the the necessary revenues to support the investment.

(2) Changes in pricing usually have to be justified on the basis of some additional cost that a new application is putting on the ISP. Most of the efficiency of SnF comes from using underutilized ISP bandwidth during off-peak hours. Putting this bandwidth to work does not increase the operational cost of a transit ISP.[4] When using bandwidth above the percentile, SnF is no different than any other paying client. Therefore a deviation from 95-percentile *only for DTB transfers* would constitute a kind of price customization that is difficult to justify based on added cost.

(3) Changing the percentile *for all traffic*, upwards *e.g.*, making it 99-percentile, would actually help SnF, because it would increase the volume that can be water-filled. Lowering it, *e.g.*, making it 50-percentile, would decrease the volume that can be water-filled by SnF, but would fail to punish traffic spikes from non-DTB clients and, therefore, would suffer from the shortcoming mentioned in (1).

(4) Transit ISPs could abandon percentile pricing altogether and adopt a more complicated rule *for all traffic* that would extract more revenue from DTB traffic without letting spikes get away for free. This would allow transit ISPs to claim part of the profit that a DTB transfer service around SnF can make. This is a possibility that we cannot preclude but it requires a thorough economic's analysis that goes beyond the scope of the current work.

8. RELATED WORK

There have been several proposals for bulk transfers at different layers of the protocol stack. The Scavenger service of Qbone [14] tags delay tolerant traffic so that routers can service it with lower priority. Its limitation is that it protects the QoS of interactive traffic, but cannot protect against high transit costs or meet specific deadlines. Also, due to TCP congestion avoidance, it allows a single congested link to block the opportunity to exploit cheap bandwidth at other links of a path.

At the application layer, P2P systems like Slurpie [15] have been developed for bulk transfers between flat-rate priced residential broadband customers. Such P2P approaches are more appropriate for one-to-many distribution services that benefit from the large numbers of receivers who, in view of flat-rate pricing, incur no additional monetary cost if they relay received data to other peers. Additionally, existing P2P systems attempt to reduce transit costs through spatial methods, *e.g.*, using locality-biased overlays that avoid peers from remote ASes [1, 19]. Our approach is temporal because the constraints we are facing are correlated with local times at the two extremes of flows.

Percentile charging schemes have been studied in close connection to multihoming and smart routing [17, 6]. Our paper is related to [9] which proposes offline and online smart routing techniques to minimize costs under percentile-based charging. All of the above proposals care only about the local percentile of a sender or receiver but not for both. Also, they do not permit time-shifting since they target interactive traffic. Time-based shifting has been used in the past, *e.g.*, for the smoothing of VBR video sources [13]. Our work operates at much larger time scales that make time-of-day effects, and their impact on ISP pricing, relevant. Also, because we care to smooth the aggregate traffic from the background and our source, E2E-Sched works on the opposite direction of smoothing (which is what E2E-CBR does).

Delay tolerant communications have received a lot of attention recently in the context of wireless intermittently connected networks of mobile devices that come into contact in public spaces [10, 4, 8, 7]. Upon contact, devices forward and store messages with the aim of eventually locating the intended final recipients whose locations are unknown and changing. Such applications utilize store-and-forward to solve the problem of unavailable end-to-end paths in wireless ad hoc networks. In our work, end-to-end paths exist at all times, but have time-varying costs, therefore, the scheduling problems arising in our case differ substantially from the ones in the wireless domain. At the far extreme of delay tolerance, there have been interesting proposals for hybrid combinations of the Internet and the postal system for delivering bulk data in hard disks in areas that lack broadband access [18]. These resemble the courier services discussed earlier.

9. CONCLUSIONS

In this paper we have looked at the possibility of using already-paid-for bandwidth resulting from the combination of diurnal load fluctuation with 95-percentile pricing, for transferring Tbyte-sized Delay Tolerant Bulk (DTB) data. Our main objective was to compare a simple source scheduling policy (E2E-Sched) with a Store-and-Forward policy (SnF) utilizing storage inside transit ISPs. Based on extensive performance evaluation driven by real network traffic, routing, and bandwidth prices, we conclude on the following:

- If E2E-Sched can send the DTB data for free then it is an obvious solution since it doesn't require

[4] The ISP incurs some added energy cost due to increased utilization during off-peak hours. However, the utilization dependent consumption of networking equipment is known to be small compared to the base energy for keeping the equipment running [3]. Also, the impact to the environment is negligible compared to carrying disks in airplanes.

transit storage. For sender-receiver pairs with up to 5 hours of time zone difference, E2E-Sched is not much worse than SnF (only 20-30%) so if SnF can ship some data for free, it is highly probable that E2E-Sched can also ship them for free.

- As the time-zone difference increases, and granted that the two end-points have comparable free capacity, thus allowing the time-zone difference to impact the end-to-end performance, SnF starts having a much higher advantage. It can double the amount of free capacity for pairs with 6 hours difference and triple it at 12 hours. In that case it can easily be that a DTB job is transferred for free by SnF but incurs transit costs under E2E-Sched. Due to the large gap between the price of transit bandwidth and storage, SnF become much more economical in this case.

- Comparing the cost of SnF to the cost of shipping data in hard disks using a courier service, our high-level evaluation indicates that courier services are cheaper for individual shipments that occur infrequently, but when there is a constant flow of data to be transferred, then in many cases they are more expensive than SnF.

The above results establish that there exists significant potential for using commercial ISPs to perform low cost DTB transfers. Our evaluation of E2E-Sched and SnF against real data is a starting point but there's definitely much more to be done in this area. Several important implementation and architectural issues need to be studied and addressed. For example issues relating to data encoding, error recovery, optimization of transport (TCP timing issues, number of parallel TCP connections for a given job, *etc.*), and of course multiplexing of multiple concurrent DTB jobs.

At a higher level, there exist several business models for realizing the benefits of DTB transfers. It could be that an independent Content Distribution Network (CDN) installs and operates storage nodes, receiving money from DTB sources like CERN, and paying for incurred transit costs. Another option is to have a confederation of access ISPs operating their local access storage nodes and sharing the cost of transit storage nodes inside the transit provider. A third approach would have the transit provider installing and operating storage nodes and leasing them to access ISPs having DTB data in the same way that it leases its bandwidth to access ISPs having interactive data. Combining the above business models with different pricing schemes (discussed in Sect. 7) creates a wealth of interesting possibilities to be considered by future work.

10. REFERENCES

[1] Vinay Aggarwal, Anja Feldmann, and Christian Scheideler. Can ISPs and P2P users cooperate for improved performance? *ACM SIGCOMM Computer Communication Review*, 37(3):29–40, 2007.

[2] Eli Brosh, Salman Abdul Baset, Dan Rubenstein, and Henning Schulzrinne. The delay-friendliness of TCP. In *Proceedings of ACM SIGMETRICS '08*.

[3] Joe Chabarek, Joel Sommers, Paul Barford, Cristian Estan, David Tsiang, and Steve Wright. Power Awareness in Network Design and Routing. In *Proceedings of IEEE INFOCOM '08*.

[4] Augustin Chaintreau, Abderrahmen Mtibaa, Laurent Massoulie, and Christophe Diot. The Diameter of Opportunistic Mobile Networks. In *Proceedings of ACM CoNEXT '07*.

[5] Costas Courcoubetis and Richard Weber. *Pricing Communication Networks: Economics, Technology and Modelling*. Wiley, 2003.

[6] Amogh Dhamdhere and Constantinos Dovrolis. ISP and Egress Path Selection for Multihomed Networks. In *Proceedings of IEEE INFOCOM '06*.

[7] Vijay Erramilli, Augustin Chaintreau, Mark Crovella, and Christophe Diot. Delegation Forwarding. In *Proceedings of ACM MobiHoc '08*.

[8] Vijay Erramilli, Augustin Chaintreau, Mark Crovella, and Christophe Diot. Diversity of Forwarding Paths in Pocket Switched Networks. In *Proceedings of ACM IMC '07*.

[9] David K. Goldenberg, Lili Qiuy, Haiyong Xie, Yang Richard Yang, and Yin Zhang. Optimizing Cost and Performance for Multihoming. In *Proceedings of ACM SIGCOMM '04*.

[10] Sushant Jain, Kevin Fall, and Rabin Patra. Routing in a Delay Tolerant Network. In *Proceedings of ACM SIGCOMM '04*.

[11] Anukool Lakhina, Konstantina Papagiannaki, Mark Crovella, Christophe Diot, Eric D. Kolaczyk, and Nina Taft. Structural Analysis of Network Traffic Flows. In *Proceedings of ACM SIGMETRICS / Performance 2004*.

[12] Matthew Roughan, A. Greenberg, C. R. Kalmanek, Michael Peter Rumsewicz, J. Yates, and Y. Zhang. Experience in Measuring Internet Backbone Traffic Variability: Models Metrics Measurements and Meaning. In *Proc. of ITC-18*, 2003.

[13] James D. Salehi, Zhi-Li Zhang, Jim Kurose, and Don Towsley. Supporting Stored Video: Reducing Rate Variability and End-to-End Resource Requirements Through Optimal Smoothing. *IEEE/ACM Transactions on Networking*, 6(4):397–410, 1998.

[14] S. Shalunov and B. Teitelbaum. Qbone Scavenger Service (QBSS) Definition. Internet2 technical report, March 2001.
[15] R. Sherwood, R. Braud, and B. Bhattacharjee. Slurpie: A Cooperative Bulk Data Transfer Protocol. In *Proceedings of IEEE INFOCOM '04*.
[16] Matti Siekkinen, Guillaume Urvoy-Keller, Ernst W. Biersack, and Denis Collange. A Root Cause Analysis Toolkit for TCP. *Computer Networks*, 52(9):1846–1858, 2008.
[17] Hao Wang, Haiyong Xie, Lili Qiu, Abraham Silberschatz, and Yang Richard Yang. Optimal ISP Subscription for Internet Multihoming: Algorithm Design and Implication Analysis. In *Proceedings of IEEE INFOCOM '05*.
[18] Randolph Y. Wang, Sumeet Sobti, Nitin Garg, Elisha Ziskind, Junwen Lai, and Arvind Krishnamurthy. Turning the Postal System into a Generic Digital Communication Mechanism. In *Proceedings of ACM SIGCOMM '04*.
[19] Haiyong Xie, Yang Richard Yang, Arvind Krishnamurthy, Yanbin Liu, and Avi Silberschatz. P4P: Provider Portal for Applications. In *Proceedings of ACM SIGCOMM '08*.