Beyond MLU: An Application-Centric Comparison of Traffic Engineering Schemes

Abhigyan Sharma, Aditya Mishra, Vikas Kumar, Arun Venkataramani
Dept. of Computer Science, University of Massachusetts Amherst

Abstract—**In this work, we revisit the traffic engineering (TE) problem focusing on user-perceived application performance, an aspect that has largely been ignored in prior work. Using real traffic matrices and topologies from three ISPs, we conduct very large-scale experiments simulating ISP traffic as an aggregate of a large number of TCP flows. Our application-centric, empirical approach yields two rather unexpected findings. First, link utilization metrics, and MLU in particular, are poor predictors of application performance. Despite significant differences in MLU, all TE schemes and even a static shortest-path routing scheme achieve nearly identical application performance. Second, application adaptation in the form of location diversity, i.e., the ability to download content from multiple potential locations, significantly impacts TE. Even the ability to download from just 2–4 locations enables all TE schemes to achieve near-optimal capacity, and static routing to be within 30% of optimal. Our findings call into question the value of TE as practiced today, and compel us to significantly rethink the TE problem in the light of application adaptation.**

I. Introduction

Traditionally, the traffic engineering (TE) problem has been studied as an optimization problem that takes as input a traffic matrix (TM) and seeks to compute routes so as to minimize a network cost function. The cost function is intended to capture the severity of congestion hotpsots based on link utilization levels. For example, the most widely used cost function, MLU, is simply the utilization of the most utilized link in the network [10], [9], [7], [6]; others sum over all links a convex function of their utilization (so as to penalize highly utilized links more) [1], [2]. There are two implicit assumptions underlying this line of work. First, maintaining low link utilization improves user-perceived application performance under typical load conditions. Second, maintaining low link utilization increases the effective capacity of the network by enabling it to accommodate unexpected surges in the traffic demand.

Our work questions both of the above assumptions. The distinguishing aspect of our work is an application-centric approach to the traffic engineering problem: instead of posing TE as as optimization problem seeking to minimize link utilization, we focus on application performance metrics such as TCP throughput or delay for elastic traffic and quality-of-service metrics (e.g., MOS metric for VoIP quality [36]) for inelastic traffic. Accordingly, our evaluation methodology is empirical: instead of relying on mathematical simulations based on linear programming or heuristic techniques for NP-complete problems, our experiments carefully and at scale simulate end-to-end application behavior so as to compare TE schemes with respect to their impact on application performance.

Our application-centric and empirical approach reveals rather unexpected results. Our first finding is that metrics based on link utilization alone, and in particular MLU, are a poor proxy for application performance. For example, a TE scheme may incur twice the MLU of another TE scheme and yet achieve as good or better application performance. The key reason for this mismatch is that application performance is largely determined by end-to-end loss rate and delay, but link utilization does not capture them accurately. At typical Internet loads, and in fact until the utilization starts approaching the capacity, link loss rates remain negligibly small. This observation has also been confirmed by explicit measurements on Internet backbones [29], and is consistent with studies on ISP backbones showing that over 90% of all packet loss is caused by interdomain routing fluctuations as opposed to high utilization [50] and 90% of TCP flows experience no packet loss [47]. Furthermore, end-to-end Internet path delays are largely determined by propagation delays as opposed to queueing delays [47], [48].

As a result, we find that all state-of-the-art TE schemes achieve nearly identical application performance at typical Internet load levels. In fact, even static shortest-path routing with link weights inversely proportional to the capacity (InvCap) (i.e., no engineering at all) achieves the same application performance as optimal TE. On the other hand, TE schemes attempting to account for unexpected spikes in traffic (e.g., COPE [10]) hurt TCP throughput slightly (by up to 10%) despite achieving near-optimal MLU.

More surprisingly, we find similar conclusions characterizing the achieved capacity of different TE schemes. When we account for application adaptation to location diversity, i.e., the ability to download content from multiple potential locations, all TE schemes achieve near-optimal capacity. With location diversity, we find that the inverse of the MLU is no longer a meaningful metric of capacity. Instead, we formalize a new metric of the capacity achieved by a TE scheme called the *surge protection factor* (SPF) that captures the factor of increase in demand that can be sustained while accounting for location diversity. Although location diversity significantly increases the SPF of all TE schemes, it benefits sub-optimal TE schemes like OSPF weight-tuning [1] more, enabling them to catch up with optimal TE. Even the static routing scheme, InvCap, achieves an SPF at most 30% worse than optimal TE.

Section II discusses the effect of location diversity on TE problem. Section III presents our simulation setup. Section IV presents comparison of application performance for TE schemes and Section V compares them based on capacity taking location diversity into account. In Section VI we point out limitations of our study and discuss related work in Section VII before concluding (Section VIII).

II. Engineering traffic with location diversity

In this section, we introduce location diversity, explain how it changes the traffic engineering problem, and introduce a new metric to quantify the capacity achieved by traffic engineering schemes with location diversity.

A. Location diversity: Prevalence

Location diversity, or the ability to download content from multiple potential locations, is widespread in the Internet today.

Major commercial CDNs, e.g., Akamai [32], Level-3 [34], EdgeCast [33] etc., commonly replicate content at hundreds of locations and redirect users to the best server based on proximity, dynamic monitoring of server and network congestion [35]. Popular P2P applications such as BitTorrent [22], PPLive [49] download content simultaneously from many peers that keep changing based on a number of factors including network congestion. Other examples of location diversity include cloud computing infrastructure providers such as Google and Amazon with geographically distributed data centers; content hosting services such as Carpathia [40], Rapidshare [45] etc.; mirrored websites such as SourceForge, Debian, etc.

Although quantifying the extent of location diversity in today's Internet is difficult, back-of-the-envelope calculations based on existing measurement studies suggests that it is significant. CDNs alone are estimated to account for 10% of Internet traffic [46]. Major cloud computing and content hosting companies with location diversity contribute to a significant fraction of Internet traffic, e.g., Google (6%), Comcast (3%), RapidShare (5%) and Carpathia (0.5%), a trend that is projected to increase in the near future [13], [46]. The fraction of P2P traffic in Internet is estimated to be between18-60% by different measurement studies in the year 2008-09. Although the fraction of P2P traffic is decreasing relative to managed hosting sites, its overall volume is still believed to be on the rise [13], [46].

B. Location diversity: Impact on TE

Location diversity necessitates revisiting traffic engineering as it changes the assumptions underlying the traditional formulation of the problem, as described next.

1) Location diversity increases capacity: Location diversity can significantly increase the capacity of a network. For example, consider the three-node network in Figure 1. Suppose each link has 100 Mbps of capacity and each node seeks to download some content. Without location diversity, each node can download its content from exactly one location, say its counter-clockwise neighbor, i.e., 1 downloads from 2, 2 from 3, and 3 from 1. In this case, each node gets 150 Mbps of flow using both the direct and the 2-hop path to its source node. With location diversity. each node can download from both adjacent nodes. Now each node can receive a total of 200 Mbps. In this example, a diversity of two locations increases the capacity of the network by 200/150 = 1.33.

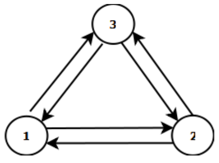


Fig. 1. Triangle network

2) Location diversity changes the TE problem: A key assumption underlying the traditional formulation of the TE problem is that the input traffic matrix is fixed, i.e., computing routes by itself does not change the traffic matrix (although it may change over time due to inherent variation in user demand). However, when applications can leverage location diversity, the traffic matrix itself depends upon the TE scheme, i.e., the very act of computing routes can change the matrix.

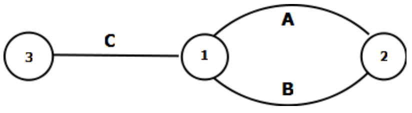


Fig. 2. Lasso network

The three-node network in Figure 2 exemplifies the above phenomenon. All links are assumed to have a capacity of 100 units and a constant delay. The top link A has a very small delay compared to the other two links that both have equal delay. Node 1 has 100 Mbps of demand that it can obtain from 2 as well as 3. In addition, there is 20 Mbps of demand at node 1 which it can obtain only from 2. We assume that that the aggregate demand at a node consists of a large number of user-initiated connections. When content can be downloaded from multiple locations, users initiate parallel TCP connections and the throughputs along paths in a parallel TCP connection are inversely proportional to the path delays. The TE scheme is assumed to be OSPF-based, i.e., shortest-path routing using configured link weights and traffic split equally among multiple paths with equal weights.

Suppose the weights of the links A and B are unequal and the link A has more weight. As a result, all of the traffic between 1 and 2 is routed using only link B. 1 splits its demand of 100 Mbps using parallel TCP equally between links B and C. Thus, the traffic on links A, B, and C is 0, 70, and 50 respectively. In the next step, seeking to balance load better for this resultant matrix, the TE scheme sets both the links A and B to the same weight (hoping to achieve link utilizations of 35, 35, and 50 respectively). Consider how parallel TCP connections respond to this change. Assuming each TCP connection between 1–2 is pinned to only one of the two paths—as is commonly done in practice to achieve equal-cost multi-path (ECMP) splitting—50 Mbps of demand at 1 gets routed using parallel TCP connections over the link A and link C, and an equal amount using parallel TCP connections along the link B and link C. In addition, the 20 Mbps of background traffic is split equally among link A and link B as per ECMP. Since link A has a much smaller delay than link C, the 50 Mbps of demand at 1 using parallel TCP along those two paths will flow entirely through link A. The remaining 50 Mbps using B and link C will get split equally across the two paths by parallel TCP. Thus, the traffic on the links A, B and C is 60, 35, and 25 respectively, which is different from what the TE scheme engineered for (namely, 35, 35, and 50). The resulting MLU of 0.6 is different compared to 0.5, the value that the TE scheme expected.

C. Location diversity: Quantifying capacity

How can we quantify the capacity achieved by a TE scheme in the presence of location diversity? In general, the capacity is a *region* that includes all of the traffic matrices that it can accommodate. However, quantifying the capacity of a TE scheme as a region may shed little light on its ability to tolerate typically encountered load spikes. Furthermore, it is cumbersome to compare TE schemes that achieve overlapping capacity regions. So, it is common to use a more concise metric such as the MLU to characterize the capacity with respect to a given traffic matrix. Intuitively, the inverse of the MLU serves as a metric of capacity, e.g., if a TE scheme achieves an MLU of 0.25 for a given matrix, then it can tolerate up to a 4× surge in the load represented by the matrix. Unfortunately, as the example in Figure 2 shows, MLU is not a meaningful metric of capacity when application adaptation to location diversity determines the traffic matrix.

With location diversity, the demand is best represented as a "content matrix" that specifies for each node and each content the traffic for that content at that node and the set of source locations from where that content can be downloaded (e.g., 100 Mbps at node 1 downloadable from 2 and 3, and 20 Mbps at node 1 downloadable from node 2, in Figure 2). The traffic matrix corresponding to this demand depends upon the underlying routes and application behavior (e.g., how parallel

TCP splits traffic across the download locations). Furthermore, scaling the demand by some factor does not simply scale the traffic matrix entries by the same factor. In general, it is difficult to predict how application behavior might change the traffic matrix for a projected surge in demand, as that change depends upon the underlying routes that in turn depend upon the original traffic matrix. Indeed, as the example shows, even if the demand is unchanged, the mere act of engineering routes can change the traffic matrix yielding a different MLU than expected.

1) An empirical capacity measure: We propose a new metric, *surge protection factor* (SPF), to quantify the capacity achieved by a TE scheme with respect to a traffic matrix. Let E denote a TE scheme, M the demand specified as a content matrix. When there is no location diversity, M can be easily transformed to a unique traffic matrix $T(M)$. Let MLU$(E, T(M))$ denote the MLU achieved by E given the traffic matrix $T(M)$. In this case, SPF(E, M) is simply the inverse of MLU$(E, T(M))$, i.e., the factor of increase in the demand that can be satisfied. However, in the case when there is location diversity, SPF(E, M) is an *empirical* measure of the satisfiable increase in demand computed as follows. Let kM denote the demand that scales each entry in M by a factor $k > 1$. Then, SPF(E, M) is defined as the largest k such that the routing computed by E (for the matrix $T(M)$) can satisfy the demand kM.

Determining if an engineering scheme can satisfy a projected demand is difficult as it requires us to accurately model application adaptation to location diversity, so SPF is useful mainly as an empirically measured capacity metric. To this end, we describe our experimental setup next.

III. Experimental setup

In this section, we describe our ns-2 simulation setup used to compare traffic engineering with respect to their impact on application performance. We chose ns-2 as the simulation platform as it is well-suited to simulate thousands of flows in an ISP network at the packet level while also incorporating transport and application behavior in a fine-grained manner.

A. Simulating traffic matrices in ns-2

We construct an ISP network topology from our dataset consisting of PoP-level ISP topology maps. PoPs are represented as nodes and links between nodes representing PoPs are the backbone links of the ISP. Each PoP node has a number of users connected to it via separate access links. Each PoP node in our topology also has 5 server nodes connected via very high capacity links which act as servers for file downloads by users. The number of user nodes in our simulation ranges from 300-6000 nodes and the capacity of backbone links varies from 50Mbps to 1Gbps.

We translate a traffic matrix to a sequence of file downloads as follows. Suppose the traffic matrix entry from A to B is 100 Mbps and the duration being simulated is 200 seconds. We generate a sequences of file downloads from A to B whose total size is 100Mbps × 200 seconds during the experiment interval. The routing used in the network is computed is computed using a TE scheme. For each file, we specify the path using the source routing option in ns-2. When an engineering scheme yields a routing topology with mutiple paths between two nodes, we assign files proportional to the traffic on each path. We are able to simulate matrices accurately using this method and the difference between empirical link utilization from ns-2 and the value obtained using a linear program based calculation is at most 0.1.

ISP networks have backbone links of few tens of Gbps. Simulating such a huge network even for 100 seconds would require sending data of the order of terabytes (or equivalently, a million 100KB files). Experimentally, we find that simulating at a tenth of this scale, i.e., 100 thousand files, is feasible given our computational and memory constraints. A typical scale in our simulation is 1/20, i.e., we simulate the backbone link with 1/20 the capacity and also divide the traffic between each source destination pair proportionally.

1) ISP topologies and traffic matrices: We used datasets from three ISPs for our experiments as described below: **Abilene:** Publicly available Abilene ISP data [16]. **Geant:** Publicly available Geant ISP data [17]. But, we used the un-anonymized version of topology obtained from the project personnel. **US-ISP:** Data for a Tier-1 US-ISP obtained from authors of [7]. TMs for all ISPs were logged in the period from 2004-2005. Figure 3 shows number of nodes, number of links and the interval at which TMs are logged for each ISP. The information of the number of nodes and links for US-ISP is proprietary.

ISP	Nodes	Links	Duration of each TM
Abilene	12	30	5min
Geant	22	68	15min
US-ISP	-	-	1hr

Fig. 3. ISP Data

2) Simulation parameters: **Scale:** We experiment with Abilene, Geant and US-ISP datasets at scales 1/10, 1/20 and 1/100 respectively. These are the largest scales we can experiment with for each network given our computational constraints.
Duration: The simulation duration for each experiment is 300 seconds. We did some experiments with longer durations of 500 seconds, but it did not qualitatively affect our findings. We note that the duration here refers to the real time being simulated in ns-2, not the system time required to run the simulation.
Bandwidth of users: We use the bandwidth distribution of Internet users from the "State of the Internet Report" [20] released by Akamai, one of the largest commercial content distribution networks in operation today. In Figure 4, we tabulate this data for US and Europe users.

BW (Mbps)	US users %	Europe users %
0.25	4.9	1.5
2.0	38.1	26.2
5.0	32.4	57.8
10.0	20.0	14.5
20.0	4.6	-

Fig. 4. Bandwidth Distribution

File sizes: We simulate three file sizes of 100KB, 1MB and 10MB respectively contributing to fraction of 8%, 3% and 89% of the total traffic. These values are the fraction of traffic due to small files (<200KB), medium size files (200KB to 2MB) and large files (>2MB) in the Internet. We obtained these numbers by collating data from multiple sources [13] [15] [12] [14].
Propagation Delay: We calculated the propagation delay of backbone links from geographic distances between nodes for Geant and US-ISP. For Abilene, we measured the propagation delay of backbone links using traceroute and ping between PlanetLab [24] nodes in cities where the PoPs are located.

3) Computational resources: We used a shared cluster of 60 machines. Each machine has a 8-Core Intel Xeon processor and 16GB of memory. Each ns-2 simulation consists of up to 500s of simulated time and 10K to 200K file downloads, which results in a memory footprint of up to 10GB and takes between 1 to 48 hours to complete.

B. *Traffic engineering schemes*

We select a subset of TE schemes reflecting a variety of proposed approaches in the literature.

Optimal, the minimum MLU TE scheme for a TM. We consider it as being representative of online TE schemes.

InvCap, a simple routing scheme which does not “engineer” traffic, rather simply uses shortest-path routing using the inverse of the link capacity as the link weight. InvCap is a common default routing protocol supported by popular commercial router vendors [18].

OptWt, a shortest path routing algorithm which computes link weights using a heuristic to optimize a cost function [2]. We use its implementation in the Totem Toolbox [17]. Typically, ISPs recompute routing a few times a day based on a set of measured TMs, so we simulate OptWt by computing a new routing every 3 hours based on the average of matrices in the past 3 hours.

MPLS, a TE scheme that minimizes the MLU in an offline manner. Similar to OptWt, MPLS recomputes a new routing once every 3 hours based on average of TMs in past 3 hours.

COPE, a TE scheme that seeks to minimize the MLU while limiting the worst-case link utilization caused by unpredictable spikes in the traffic matrix. We use the same parameters settings and compute routing once a day based on previous days TMs as in [10]. We use the authors’ code for experiments.

IV. Application Performance

In this section, we present a comparative analysis of the impact of different TE schemes on end-to-end application performance. A summary of our findings is as follows. First, all TE schemes including InvCap show nearly identical application performance for TCP and UDP traffic. Second, different TE schemes do achieve different MLUs as expected, suggesting that MLU is a poor predictor of application performance. Third, COPE consistently performs slightly worse than all other schemes in TCP throughput, suggesting that accounting for unpredictable variations in traffic hurts the common case application performance.

A. *TCP performance*

We simulated TMs from 2 days of data for each ISP. For each day, we simulated 50 matrices measured at 5-minute intervals for Abilene, 25 matrices measured at 15-minute intervals for Geant, and 24 matrices measured hourly for US-ISP. We present results for the second day. The metric of application performance is the *download rate* of files using TCP, where the file arrival workload is generated using the traffic matrices as described in Section III-A.

Figure 5 shows the mean download rate of files, where the average is across all files across all of the simulated matrices for each TE scheme. The results show that all schemes achieve nearly same mean download rates with the exception of COPE that is consistently worse by up to 10%. Furthermore, as expected, Optimal (the leftmost bar in each group) is not always the best as minimizing MLU is not the same as optimizing TCP performance. Figure 6 shows the corresponding CDFs for the mean download rates shown in Figure 5. The CDFs show that the near-identical TCP performance achieved by all TE schemes is not an artifact of presenting a specific statistic such as the mean, but is reflected by the entire distribution. All distributions show a stepwise increase which suggests that access links are a bottleneck for a significant fraction of file transfers.

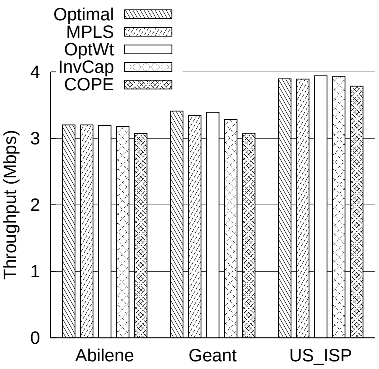


Fig. 5. Mean download rates

1) MLU vs. TCP performance: To further investigate the results in Figure 5 and Figure 6, we analyzed the empirically observed MLU for all TE schemes in the experiments. Figure 7 plots the MLUs for all matrices considered. For US-ISP the MLU data is proprietary hence we plot the ratio of MLU with respect to Optimal. As expected, different TE schemes do show a substantial difference in MLU, e.g., the MLU for InvCap and OptWt is up to twice the MLU of Optimal in some cases. These results suggest that MLU is a poor predictor of download rate performance: schemes with near-identical TCP throughput have very different MLUs, and COPE despite achieving near-optimal MLU consistently shows sub-optimal TCP throughput.

The main reason why MLU does not affect download rate is because queuing delay and loss rates are negligible until link utilization reaches a threshold. In our experiment, link utilization below 0.7 causes near negligible loss rates and queuing delays. Since the MLUs on most of the traffic matrices are below this value, loss rates on backbone links minimally impact the throughput of file downloads. These observations are consistent with a recent study on Level-3 network [29] showing that loss rates on backbone links are zero even at 95% link utilization. This threshold is expected to be higher for actual backbone traffic as our experiments are at scale 1/10 or smaller. At larger scales, there would be more concurrent flows resulting in less bursty traffic and lower loss rates.

The second reason why MLU minimally impacts the aggregate download rate as well as the distribution is because it is largely determined by the traffic of only one link. Even under high MLU, the rest of the network may not be congested. File download rates are affected only for flows on this link, which may a small fraction of total traffic.

2) The price of predictability: Why is COPE’s performance consistently (even if only slightly) worse than the other schemes? To investigate this, we analyzed the propagation delays of routes computed by COPE. Given uniformly low loss rates and queueing delays, propagation delays primarily determine TCP performance.

Figure 8 shows the path delay averaged across all files and across all matrices for the different TE schemes. COPE has a significantly higher delay compared to all other schemes. We attribute this phenomenon to COPE’s optimization approach, which engineers for unpredictable spikes in traffic demands. Specifically, COPE attempts to bound the worst-case MLU for any traffic matrix similar to oblivious routing like schemes [6]. COPE intentionally routes some traffic along longer paths so as to leave room for occasional spikes in the traffic along shorter paths. While this approach makes COPE robust with respect to MLU to rare spikes in traffic, it comes at the cost of hurting common case application performance. Although we have not experimented with other oblivious routing schemes, these results suggest that any oblivious routing scheme that attempts to optimize MLU, e.g., [6] is likely to incur a similar penalty in application performance in the common case.

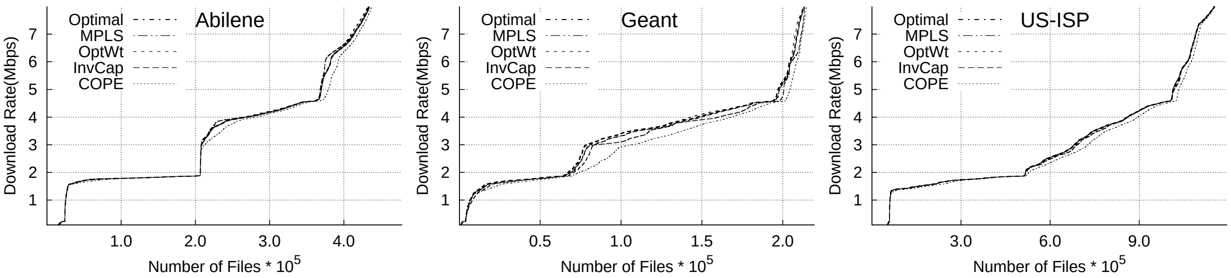


Fig. 6. Download rate CDFs for all TE schemes are near identical except COPE which has slightly lower performance

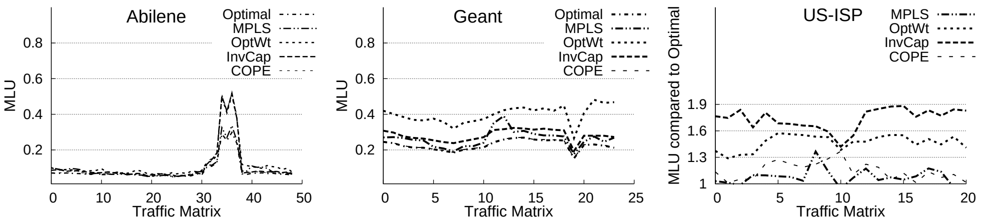


Fig. 7. TE schemes differ as much as 2× in MLU

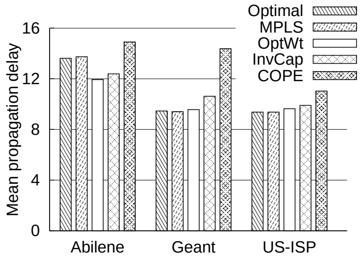


Fig. 8. COPE has the highest propagation delay among TE schemes

B. UDP performance

1) Measuring UDP performance: We derive delay and loss rate for UDP traffic using the average loss rate and queuing delay on backbone links measured during TCP experiments. This assumption is true in practice since TCP dominates Internet traffic by more than 90% [47]. For each path we compute its average loss rate by combining the average loss rates of links in the path; we compute average delay by summing the propagation and queuing delay of links in the path.

We compare performance of VoIP traffic (which uses UDP) using Mean Opinion Score (MOS). MOS score is VoIP call quality metric for which a score of above 4 is considered good and below 3 is considered bad. For all pairs of nodes in a topology, we calculate MOS score using the formula in [36] which calculates MOS score based on loss rate and delay values.

2) Results: The range of values of MOS scores for all TE schemes for Abilene is (4.07,4.14), for Geant is (3.75,4.14) and for US-ISP is (4.11,4.14). The greater difference for Geant is because one of the nodes in its topology is connected using two links which have 155Mbps capacity each, while all other links have more than 1Gbps capacity. At scale 1/20, 155Mbps link is simulated using a 7Mbps link which has significantly greater burstiness of traffic and greater loss rate and queuing delay. Excluding these links, the range of values of MOS scores is (4.07,4.14).

All TE schemes have qualitatively same performance for VoIP traffic. All MOS scores are above 4.0 and even the variation is less than 0.1. These results are not surprising since loss rates and queuing delay have near negligible value for all links in the network. There is some difference in propagation delay for TE schemes but the MOS score is not very sensitive to a difference of a few milliseconds of delay.

V. Capacity and Location Diversity

The results in the previous section show that different TE schemes yield nearly identical application performance at traffic demands encountered today. In this section, we compare TE schemes with respect to their potential capacity, i.e., the ability to accommodate surges in the traffic demand. In contrast to most prior work, our capacity analysis incorporates the ability of applications to leverage location diversity, i.e., the ability to download content from multiple locations. Our main findings are that (1) location diversity can significantly increase (by up to 2×) the capacity achieved by all engineering schemes; (2) even a modest amount of location diversity (e.g., the ability to download content from two locations) enables all engineering schemes to achieve near-Optimal capacity; (3) with location diversity even simple routing scheme of InvCap has at most 30% less capacity compared to Optimal.

A. Empirically measuring capacity

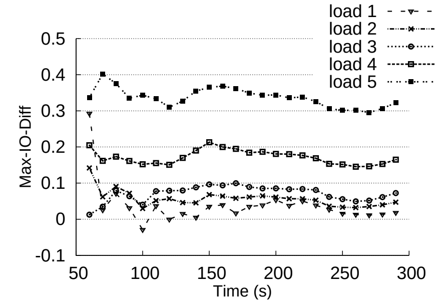


Fig. 9. Profile of Max-IO-Diff at increasing loads for a Geant TM

Our metric of capacity is the SPF, i.e., the maximum surge in demand that can be satisfied, formally defined in Section II-C1. Analytically determining whether an engineering scheme can satisfy a projected demand is difficult as it requires us to accurately model application adaptation to location diversity, so the SPF must be determined empirically. In our experiments, we use a metric called *maximum input output difference* (or Max-IO-Diff) to determine whether a given demand can be satisfied. For each node, the *input* is the total traffic (bits/sec) requested by that node, while the *output* is the total traffic received by that node. Max-IO-Diff is defined as the maximum across all nodes of the relative difference between the input and output, i.e., *(input - output)/input*. If Max-IO-Diff is measured to be less than 0.1, then the demand is considered as satisfiable. We allow for a small difference in order to account for measurement error as well as to account for bursts in demand over the measurement duration.

Max-IO-Diff helps clearly distinguish workloads that can be satisfied. For example, in Figure 9, we show a Max-IO-Diff profile for five ns-2 simulations at surge factors of 1, 2, 3, 4 and 5 for a Geant TM with InvCap routing. The Y-axis in the graph is the Max-IO-Diff measured at intervals of 10 seconds during the simulation. The X-axis shows the progress of simulation. We ignore the first 50 seconds of simulation since input significantly exceeds output at the start of simulation. We observe that beyond the initial period of fluctuation, Max-IO-Diff is relatively stable and below 0.1 for surge factors 1–3 that can be satisfied, but significantly higher for surge factors 4 and 5 that can not be satisfied.

B. Simulating location diversity

In our experiments, we enable location diversity by downloading in parallel from multiple locations. Each download initiates parallel TCP connections to k locations hosting the requested content, where k is a location diversity parameter. A download is considered as completed when the total number of bytes download across all k connections equals the size of the requested content.

As application adaptation to location diversity can change the traffic matrix, our experiments are conducted in a two-step manner as follows. We start with a set of periodically logged traffic matrices from real ISPs in our dataset. Let $M_1, M_2, \cdots$ denote a sequence of such matrices and E a TE scheme. As in the previous section, we translate each matrix M_i to a file request arrival process. However, instead of downloading each file from just one location, we add $k - 1$ randomly chosen source locations from which the file is downloaded in parallel. The routes from the k source locations to the sink are computed by applying E to M_i. As a result of parallel downloads, each matrix M_i will have changed, say, to a new matrix N_i.

In the second step, we recompute routes by applying E periodically to the appropriately time-averaged matrix. For example, if E is Optimal, we recompute routes periodically (once every 5, 15, or 60 minutes depending upon our ISP dataset) based on N_i at that time instant. For OptWt, we recompute routes once every 3 hours using a matrix that is the average of the matrices in the past 3 hours, and so on.

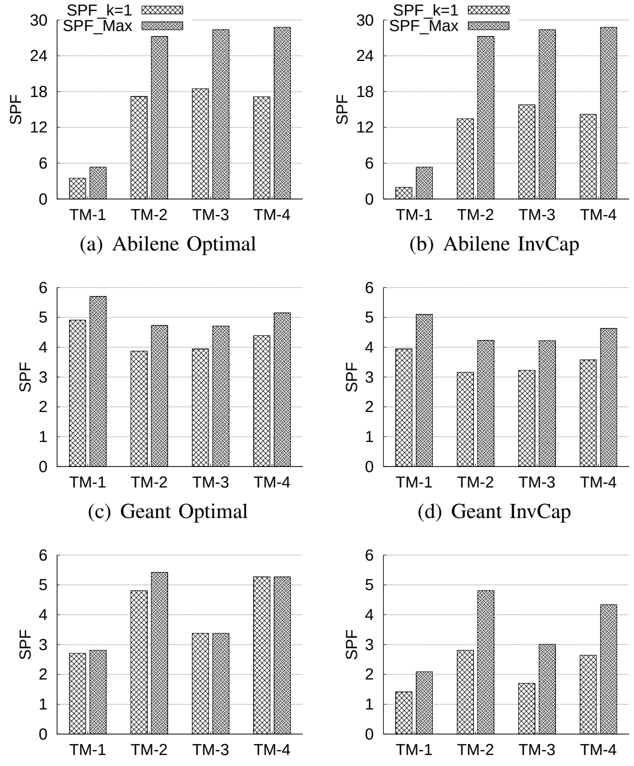
C. Experimental procedure

The experiments to determine SPF involve a computationally intensive search across many different surge factors for each matrix. Furthermore, at high surge factors, the number of ns-2 data structures required to simulate ongoing parallel TCP connections becomes prohibitively high. So for computational tractability, we selected 4 matrices each from one day of data of each ISP. The matrices were selected randomly, one from each 6-hour duration during the day. For each matrix and each engineering scheme, we conduct an experiment at each value of the surge factor starting from 1 in increments of 0.25 until the capacity point is reached, i.e., the Max-IO-Diff value exceeds 0.1. Each experiment is run until the Max-IO-Diff value stabilizes or 300 seconds, whichever is greater.

D. Capacity increase with location diversity

We first present our results for the maximum capacity increase as calculated by the linear program in technical report [30], and then as measured experimentally.

1) Theoretical capacity increase with location diversity: In Figure 10, we present the results for the maximum SPF using Optimal and InvCap routing for the selected TMs. For each TM, the left bar shows the SPF without location diversity ($k = 1$) and the right bar shows the SPF when a node can obtain content from all nodes in the network except itself ($k = n - 1$, $n =$ number of nodes). Note that the later case is the maximum SPF which can be achieved with location diversity.



(a) Abilene Optimal (b) Abilene InvCap

(c) Geant Optimal (d) Geant InvCap

(e) US-ISP Optimal (f) US-ISP InvCap

Fig. 10. Maximum SPF for Optimal and InvCap routing calculated using linear program.

The capacity increase in the network for Optimal is 1.6× for Abilene, 1.2× for Geant and less than 1.1× for US-ISP topology. The capacity increase for InvCap is approx 2.1× for Abilene, 1.3× for Geant topology and 1.5× for US-ISP topology. Clearly location diversity increases the capacity both for Optimal and InvCap but the increase depends on the ISP topology and traffic matrix. InvCap has less capacity than Optimal even with location diversity but the relative difference between the two has diminished. This result shows that location diversity reduces the difference in capacity among TE schemes.

While $(n-1)$ locations can certainly give the maximum SPF, we find only 2-4 locations can give the maximum SPF for most matrices both for InvCap and Optimal). The reason is that the immediate links to some nodes often become the bottleneck first. As each node in these ISP topologies typically has on average only a few incoming links (3-4 links), even a small number of additional locations suffice to saturate all incoming links for the most loaded node in the network.

2) Capacity increase for location diversity using ns-2 simulations: The capacity increase calculated above using a linear program may not be achievable in practice. There are two reasons for this. First, users in a network may not split their flows optimally among multiple locations as in the linear program solution. Second, the routes computed using any realistic TE scheme would in general be different from the optimal routing computed above. Nevertheless, we find that location diversity increases the capacity significantly for all TE schemes.

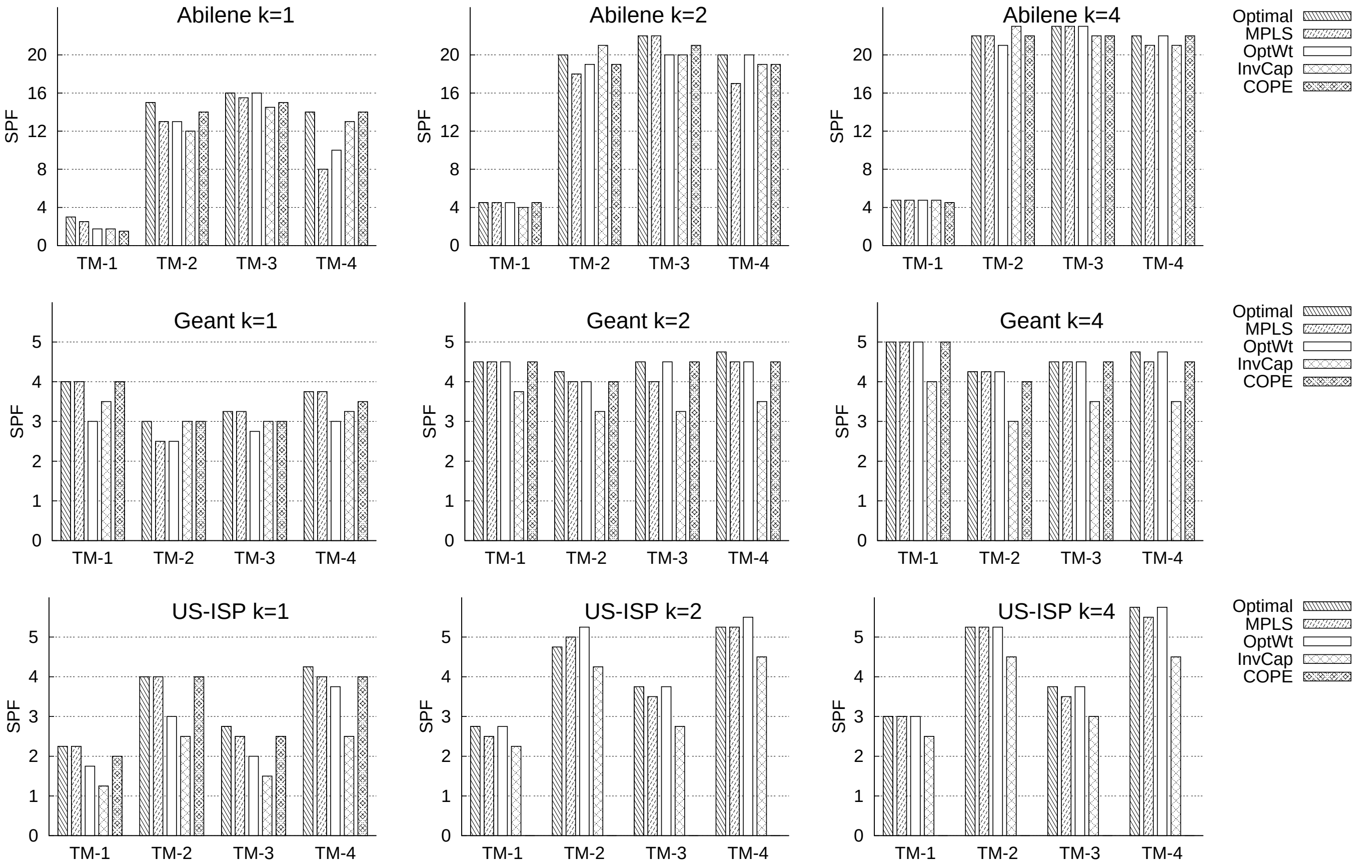


Fig. 11. Comparison of SPF among TE schemes for different levels of location diversity; SPF values are obtained using ns-2 simulations

In Figure 11, we present the SPF values obtained using ns-2 simulations for the selected TMs. We compared all TE schemes for three levels of location diversity $k = 1, 2$ and 4.

Note that we do not present the results for COPE for US-ISP (k =2 and k = 4), since the implementation of COPE's algorithm failed to compute a feasible set of routes even after 12 hours of simulation time (1 million iterations) after which we aborted the simulation. We have used authors' implementation of the algorithm and communication with them confirmed that indeed in some cases COPE's implementation can take a long time to terminate. This happens in cases where barrier-crossover method to solve linear program fails and COPE instead uses simplex method which is much slower.

The average capacity increase for Optimal from $k = 1$ to $k = 4$ is 1.41× and from $k = 1$ to $k = 2$ is 1.31×. Optimal is the maximum SPF for a network with no location diversity ($k = 1$) . This shows that a network with location diversity of $k = 4$ has 40% greater capacity than a network with no location diversity. Even location diversity of $k = 2$ gives 75% of capacity increase obtained from location diversity of $k = 4$.

Location diversity enables all TE schemes to achieve near-optimal capacity. In Figure 12 we compare the SPF of Optimal to that of other TE schemes. The statistic presented is ratio of SPF of TE scheme to SPF of Optimal for the same level of location diversity averaged over all TMs. Except InvCap, all TE schemes have SPF within 5% of Optimal for location diversity of $k = 4$ as well as $k = 2$. Figure 11 shows that with location diversity any TE scheme has at most 10% capacity difference compared to Optimal .

The above result calls into question the usefulness of online TE schemes. In today's Internet, offline TE schemes such as OptWt or MPLS are commonly used. It is believed that these schemes are suboptimal and online TE schemes (e.g., TeXCP, MATE etc.) can achieve near-optimal capacity. However, our results suggest that application adaptation to location diversity results in near-optimal SPF for all TE schemes. Even the shortest-path routing scheme, OptWt, achieves the same SPF as TE schemes employing MPLS for flow splitting.

With location diversity, InvCap achieves a capacity that is at most 30% worse than Optimal (Figure 11). On average InvCap has 15% less capacity compared to Optimal for a location diversity of $k = 4$.

E. Partial location diversity

Next, we consider a scenario where only a fraction of traffic can leverage location diversity, as is the case in today's Internet. In particular, we examine the impact of traffic that can leverage location diversity on the traffic that cannot.

We experiment with Geant matrices used in previous experiments and measure the SPF values when 0, 25, 50, 75 and 100 percent of traffic respectively have a location diversity of $k = 4$. In Figure 13, we plot the ratio of the SPF of each TE scheme to the SPF of Optimal without diversity ($k = 1$). The plotted value is the average over 4 matrices.

The curve of capacity vs fraction of traffic having location diversity shows a concave behavior. All TE schemes get more than 90% of capacity increase even with 50% traffic having location diversity. Moreover, the difference in capacity among for TE schemes remains less than 5% even when only half of the traffic can leverage location diversity. Thus, our main conclusion, i.e., even a modest amount of location diversity

TE/Optimal	k=1	k=2	k=4
Optimal/Optimal	1	1	1
MPLS/ Optimal	0.89	98	0.99
OptWt/Optimal	0.73	0.99	0.99
InvCap/Optimal	0.91	0.86	0.85
COPE/Optimal	0.91	0.99	0.98

Fig. 12. Comparison of SPF values

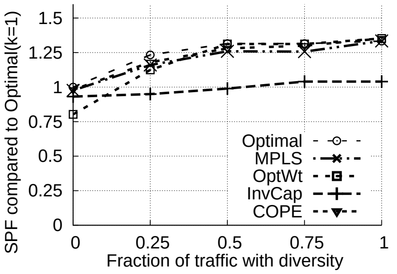


Fig. 13. Effect of partial location diversity for Geant TMs.

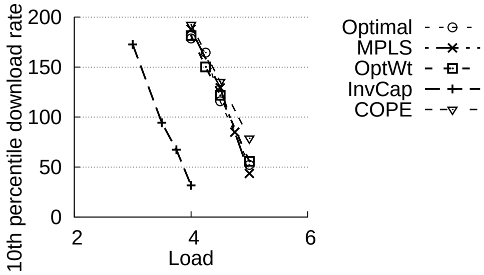


Fig. 14. TCP performance at increasing loads for Geant TM

enables all TE schemes to achieve near-optimal capacity, remains unchanged even when only half the traffic has location diversity.

F. Application performance at high loads

The near-optimality of the capacity of all TE schemes is reflected in application performance metrics as well. Due to sub-optimal capacity of InvCap, it has a sub-optimal performance at loads near capacity. We show the result for one TM from Geant with location diversity of k = 4 (Geant TM-3 in Figure 11). Other TMs have similar graphs which we omit due to lack of space. We obtain this data using output from experiments in preceding section. As we performed experiments only at loads near the SPF, we show only these points.

Figure 14 shows the 10th percentile of the download throughput distribution at different loads. All TE schemes have near identical values except InvCap. The horizontal distance between the curve of InvCap and that of other TE schemes curves is approximately equal to the difference in SPF values between them which is indicative of capacity difference between them. Other statistics, e.g. mean, median show similar trends but the horizontal distance between the curve for InvCap and other TE schemes reduces at higher percentile values.

We compute MOS scores for all source-destination pairs as in Section IV-B. We find the distribution of MOS scores by selecting node pairs randomly but giving more weight to pairs with more traffic between them. This distribution shows similar trends as the distribution of download throughputs. We do not present the graph due to lack of space. All TE schemes have no significant difference in performance for VoIP traffic based on MOS scores, except InvCap which has lower performance at loads near SPF. Note that a VoIP call is a single UDP connection between source and destination nodes without any location diversity, but it still benefits from the rest of the traffic being able to leverage location diversity.

VI. Discussion

Limitations of our study We now point out the limitations of our work considering both the experiment design as well as evaluation objectives. The TE problem for ISPs has other objectives as well, e.g., reducing interdomain traffic costs, computing backup routes for link failures, ensuring QoS for different classes of traffic [31] which we do not compare in our current study. We plan to include them in future work. We perform experiments based on dataset from three ISPs and we experiment with a small set of TMs given our resources. Our simulation setup approximates or ignores some real world network topology entities such as multi-Gigabit backbone links, the router software/hardware at backbone and access links, TCP/IP software used at users and servers, to name a few. We also do not model all aspects of Internet traffic demand pattern such as interdomain traffic, different variety of application layer protocols (HTTP, FTP, ICMP etc.), and the pattern of usage of these protocols. We show results using the adaptation scheme of parallel downloads of each file. Accurately quantifying the capacity for TE schemes for Internet traffic would involve modeling the different adaptation schemes in Internet, which we consider a complex problem.

VII. Related Work

Traffic engineering methods Over the past decade considerable work has been done in the area of traffic engineering with the objective of optimizing link utilization based metrics using OSPF [1] or MPLS [4]. Offline TE is done using measured TMs by ISPs today [3]. Different formulations of offline TE problem optimize by making a few changes in link weights [2]; optimize over multiple TMs [7]; optimize for unpredictable traffic demands [10]. In contrast to offline TE, online TE computes routing using online measurements of network conditions [5], [10]. Since online TE reacts in short time scales, these schemes claim they can achieve close to Optimal TE. The class of oblivious routing algorithms compute routing which performs well across all traffic matrices and hence aims to obviate the need for TE [6]. Our work sheds new light on this area and shows that link utilization based metrics are poor predictors of application performance, and oblivious routing like schemes increase delay and hurt TCP throughput. Further, the problem of TE does not take into account the location diversity in Internet which, as we have shown, nearly vanishes capacity difference between Optimal and other TE schemes.

Interaction of location diversity and TE Recent work has explored the joint optimization of TE and content distribution (choosing the best location/s to download content). The solution proposed for example by P4P [23] improves application performance for P2P traffic and also reduces cost for ISP by cutting interdomain traffic and maximum link utilization. In [25] and [26], the authors study the interaction using a game theoretic perspective and show that without joint optimization, the equilibrium of this interaction may not be socially optimal solution. The three node network in Section II illustrates this point. In [25], it is shown that a joint optimization can achieve benefits of up to 20 % for ISP and up to 30 % for content distribution as compared to the case when there is no cooperation between them. While these proposals may be adopted in future, TE and content distribution are done by separate entities today. Our work studies this interaction in current setting and empirically shows that even without any joint optimization, there is a significant capacity increase of upto 1.4× for TE using a simple adaptation scheme of parallel downloads. We also demonstrate an important consequence of location diversity that all TE schemes have near-optimal capacity with location diversity.

Performance evaluation A wide variety of techniques have been developed for performance evaluation of large scale networks: Theoretical models such as fixed point model [8], fluid model [42] and hybrid model [43]; network simulators such as ns-2 [19] and GTNetS [52]; emulation environments such as Emulab [28]; virtualized infrastructures such as VINI and OpenFlow [27], [44]. While a wide variety of performance evaluation techniques are available, to our knowledge our work is the first which has utilized this infrastructure to compare application performance metrics of TE schemes based on large scale simulation of traffic matrices on ISP topologies.

Application adaptation We utilize a simple adaptation technique of parallel TCP downloads. But, adaptation techniques using both *path diversity* and *location diversity* are widely in use in Internet as well as extensively explored in research. Research in a wide variety of topics such as detour routing [37], DHTs [41], CDNs, P2P networks falls under the purview of application adaptation. In the Internet, huge infrastructures such as CDNs [32], content hosting services [40], cloud computing platforms[21], mirrored websites [39] as well as end-user applications such BitTorrent [22], eMule [51], PPLive [49] and Skype [38] use adaptation techniques.

VIII. Conclusion

We revisited the traffic engineering problem focusing on user-perceived application performance metrics. Our application-centric goals and empirical evaluation methodology reveal unexpected results that challenge conventional wisdom in the area. We find that link utilization is a poor predictor of application performance. Under typical Internet load conditions, all TE schemes and even static routing achieve nearly identical application performance despite achieving vastly different MLUs. In fact, engineering for unexpected utilization spikes to optimize MLU can actually hurt common-case application performance. More intriguingly, we find that application adaptation to location diversity, or the ability to download content from multiple locations, eliminates differences in the achieved capacity of all TE schemes including optimal TE. In this case, even static routing achieves a capacity that is at most 30% (and typically significantly less) worse than the optimal TE scheme. A provocative interpretation of our findings is that which TE scheme is used or whether TE is employed at all matters little for application performance under typical load conditions today, and matters little for application performance under reasonable projections of increased traffic demand in the future.

References

[1] B Fortz, M Thorup. Internet traffic engineering by optimizing OSPF weights. In *IEEE INFOCOM*, 2000

[2] B Fortz, M Thorup. Optimizing OSPF/IS-IS weights in a changing world. In *IEEE Journal on Selected Areas in Communications*, May 2002

[3] J Rexford. Route optimization in IP networks. Chapter in *Handbook of Optimization in Telecommunications, Springer Science + Business Media*, February 2006.

[4] X Xiao, A Hannan, B Bailey, LM Ni. Traffic Engineering with MPLS in the Internet. In *IEEE Network Magazine,* Mar. 2000

[5] A Elwalid, C Jin, S Low, I Widjaja. MATE: MPLS adaptive traffic engineering. In*IEEE INFOCOM*, 2001

[6] D Applegate,E Cohen. Making intra-domain routing robust to changing and uncertain traffic demands: understanding fundamental tradeoffs. In *SIGCOMM*, 2003.

[7] C Zhang, Z Ge, J Kurose, Y Liu, D Towsley. Optimal routing with multiple traffic matrices tradeoff between average and worst case performance. In *ICNP*, 2005

[8] T Bu, D Towsley. Fixed point approximations for TCP behavior in an AQM network. In *SIGMETRICS*, 2001

[9] S Kandula,D Katabi, B Davie,A Charny. Walking the tightrope: responsive yet stable traffic engineering, In *SIGCOMM*, 2006.

[10] H Wang, H Xie, L Qiu, Y Richard Yang, Y Zhang, A Greenberg. COPE: Traffic Engineering in Dynamic Networks. In *SIGCOMM*, 2006.

[11] R. Bush and D. Meyer. RFC 3439: Some internet architectural guidelines and philosophy, December 2003.

[12] P Gill, M Arlitt, Z Li, A Mahanti. Youtube traffic characterization: a view from the edge. In *IMC*, 2007

[13] Ipoque Internet Study http://www.ipoque.com/resources/internet-studies/

[14] A Williams, M Arlitt, C Williamson, K Barker. Web Workload Characterization: Ten Years Later. DOI 10.1007/0-387-27727-7

[15] KP Gummadi , RJ. Dunn , S Saroiu , SD Gribble , HM Levy , J Zahorjan. Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload. In SOSP, 2003

[16] Abilene dataset. http://www.cs.utexas.edu/ yzhang/research/AbileneTM

[17] TOTEM Project http://totem.info.ucl.ac.be/

[18] Cisco. Configuring OSPF.
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/.

[19] The Network Simulator. http://www.isi.edu/nsnam/ns/.

[20] Akamai State of the Internet Reports http://www.akamai.com/stateoftheinternet/

[21] Map of all Google data center locations http://royal.pingdom.com/2008/04/11/map-of-all-google-data-center-locations/

[22] BitTorrent http://www.bittorrent.com/

[23] H Xie, Y Richard Yang, A Krishnamurthy, Y Liu, A Silberschatz. P4P: Provider Portal for Applications. In *SIGCOMM*, 2008.

[24] http://www.planet-lab.org

[25] W Jiang, R Zhang-Shen, J Rexford, M Chiang. Cooperative Content Distribution and Traffic Engineering in an ISP Network. In *SIGMETRICS*, 2009.

[26] D DiPalantino, R Johari. Traffic Engineering vs Content Distribution: A Game Theoretic Perspective. In *INFOCOM*, 2009

[27] VINI http://www.vini-veritas.net/

[28] Emulab http://www.emulab.net

[29] N Beheshti, Y Ganjali, M Ghobadi, N McKeown, G Salmon. Experimental study of router buffer sizing. In *IMC*, 2007

[30] Abhigyan, Aditya Mishra, Vikas Kumar, Arun Venkataramani Beyond MLU : An Application Centric Comparison of Traffic Engineering Schemes *UMASS Computer Science Technical Report*, UM-CS-2010-012

[31] D Awduche, A Chiu, A Elwalid, I Widjaja, X Xiao. Overview and Principles of Internet Traffic Engineering. *RFC 3272*

[32] Akamai http://www.akamai.com/

[33] EdgeCast http://www.edgecast.com/

[34] Level-3 http://www.level3.com/

[35] AJ Su, DR Choffnes, A Kuzmanovic, F Bustamante. Drafting behind Akamai. In *SIGCOMM*, 2006

[36] RG Cole, JH Rosenbluth. Voice over IP performance monitoring. In *SIGCOMM CCR*, 2001

[37] S Savage, T Anderson, A Aggarwal, D Becker, N Cardwell, A Collins, E Hoffman, J Snell, A Vahdat, G Voelker, J Zahorjan. Detour: Informed Internet routing and transport, In *IEEE MICRO* 1999

[38] Skype http://www.skype.com/intl/en/support/user-guides/p2pexplained/

[39] Mirroring http://en.wikipedia.org/wiki/Mirror_(computing)

[40] Carpathia http://www.carpathia.com/assets/files/carpathialoadbalancesheet.pdf

[41] I Stoica, R Morris, D Karger, MF Kaashoek, H Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications, In *SIGCOMM*, 2001

[42] V Misra, WB Gong, D Towsley. Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. In *SIGCOMM CCR*, Oct 2000

[43] S Bohacek, JP H, J Lee, K Obraczka. A hybrid systems modeling framework for fast and accurate simulation of data communication networks. In *SIGMETRICS*, 2003

[44] N McKeown, T Anderson, H Balakrishnan, G Parulkar, L Peterson, J Rexford, S Shenker, J Turner. OpenFlow: enabling innovation in campus networks. In *SIGCOMM CCR*, Apr 2008

[45] D Antoniades, EP Markatos, C Dovrolis. One-Click Hosting Services: A File-Sharing Hideout. In *IMC*, 2009

[46] ATLAS Internet Observatory 2009 Annual Report
http://www.nanog.org/meetings/nanog47/presentations/-Monday/Labovitz_ObserveReport_N47_Mon.pdf

[47] C Fraleigh, S Moon, B Lyles, C Cotton, M Khan, D Moll, R Rockell, T Seely, C Diot. Packet-Level Traffic Measurements from the Sprint IP Backbone. In *IEEE Network*, 2003

[48] K Papagiannaki, S Moon, C Fraleigh, P Thiran, C Diot. Measurement and Analysis of Single-Hop Delay on an IP Backbone Network. In *IEEE JSAC*, Aug. 2006

[49] PPLive http://www.pplive.com/en/index.html

[50] Hengartner, S Moon, R Mortier, C Diot. Detection and analysis of routing loops in packet traces. In *IMW*, 2002.

[51] eMule http://www.emule-project.net

[52] GTNetS http://www.ece.gatech.edu/research/labs/MANIACS/GTNetS/

[53] J He, M Bresler, M Chiang, J Rexford. Towards Robust Multi-layer Traffic Engineering: Optimization of Congestion Control and Routing. In *INFOCOM*, 2006

IX. Appendix

A. Linear program to calculate optimal routing for flows with multiple sources

Network $G = (V, E)$.

Nodes $V = \{n_1, n_2, ...n_a\}$

Links $E = \{e_{ij}\}$ for link between node i and j and link capacity = $C = \{c_{ij}\}$.

Location diversity parameter k. This is the number of sources for each flow.

Flows $F = \{f_1, f_2, ...f_b\}$

Sources for flow f_i, $S_i = \{s_{i1}, s_{i2}, ...s_{ik}\}$. Destination for flow f_i = d_i.

For the above network and traffic demands, we have to solve the following linear program.

Minimize: α, the maximum link utilization.

Subject to:

1) Each flow demand must be routed. For each flow f_i,

$$f_i = \sum_{i=1tok} f_{is_{ij}}, 0 \leq f_{is_{ij}} \leq f_i$$

where each $f_{is_{ij}}$ amount of flow f_i routed from source node s_{ij} to destination node t_i.

2) Treat each flow as an independent flow which can be routed independently. For flow $f_{is_{ij}}$, . The constraints for routing this flow are :
For each node q,

$$\sum_{r \in outgoing(q)} h_{qr}^{is_{ij}} - \sum_{p \in incoming(q)} h_{pq}^{is_{ij}} = d$$

$d = -f_{is_{ij}}$ if $q = s_{ij}$, $d = f_{is_{ij}}$ if $q = t_i$, otherwise $d = 0$. In addition $0 \leq h_{ij}^{st} \leq f_{is_{ij}}$ as mentioned above.

3) (Total flow on each link) $\leq \alpha$ (capacity of link). For each link e_{pq},

$$\sum h_{pq}^{is_{ij}} \leq \alpha \times c_{pq}$$

for each link.

The above formulation can provide us the global optimal solution for flows having multiple sources but a single destination.

B. Linear program to calculate optimal MLU for a given routing

The variables defined above are reused for this program. We define a new set of variables for the routing to be configured.

$$R = \{r_{ij}^{st}\}, 0 \leq r_{ij}^{st} \leq 1$$

r_{ij}^{st} is the fraction of flow from source node s to destination node t which passes on the link between node i and j. The routing defined must should obey following constraints. For each node j,

$$\sum_{k \in outgoing(j)} r_{jk}^{st} - \sum_{i \in incoming(j)} r_{ij}^{st} = d$$

$d = -1$ if $j = s$, $d = 1$ if $j = t$, otherwise $d = 0$. In addition $0 \leq r_{ij}^{st} \leq 1$ as mentioned above. The linear program is as follows:

Minimize: α, the maximum link utilization. Subject to:

1)

2) Each flow must be routed: For each flow f_i,

$$f_i = \sum_{j=1 \text{ to } k} f_{is_{ij}}, 0 \leq f_{is_{ij}} \leq f_i$$

where each $f_{is_{ij}}$ amount of flow f_i routed from source node s_{ij} to destination node t_i.

3) Total flow between a source and destination is defined as g_{st}

$$g_{st} = \Sigma f_{is_{ij}}, \forall i, j : s_{ij} = s, t_i = t$$

4) (Total flow on each link) $\leq \alpha$ (capacity of link). For each link e_{ij},

$$\sum g_s t \times r_{ij}^{st} \leq \alpha \times c_{ij}$$

for each link.

If the routing in the network is defined, i.e. r_{ij}^{st} are constants, then above defition is a linear program. This formulation can be used to compute the capacity for InvCap or any other known routing scheme. When r_{ij}^{st} are variables, the above program is not a linear program as defined above.