CHAPTER 3

TRAFFIC ENGINEERING: AN APPLICATION-CENTRIC COMPARISON

Traditionally, traffic engineering (TE) 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 [117, 70, 122, 20]; others sum over all links a convex function of their utilization (so as to penalize highly utilized links more) [50, 49]. 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 problem: instead of posing TE as as optimization problem seeking to minimize link utilization, we focus on application performance metrics such as TCP throughput for elastic traffic and quality-of-service metrics (e.g., MOS score for VoIP quality [35]) 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 [25], 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 [64] and 90% of TCP flows experience no packet

loss [53]. Furthermore, end-to-end Internet path delays are known to be largely determined by propagation delays as opposed to queueing delays [53, 89].

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. Ironically, TE schemes that engineer for unexpected traffic spikes (e.g., COPE [117]) consistently hurt TCP throughput despite achieving near-optimal MLU.

More surprisingly, we find that application adaptation to location diversity, i.e., the ability to download content from multiple potential locations, blurs differences even in the achieved capacities of different TE schemes enabling all of them to be near-optimal. 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. TE schemes calculate routing based on a measured traffic matrix to achieve a desired network cost, but application adaptation to location diversity changes the traffic matrix itself in response to a change in routing resulting a different network cost than expected. As a result, the optimal TE scheme with perfect knowledge of traffic matrix and sub-optimal TE schemes like OSPF weight-tuning [50] end-up achieving the same SPF. Even the static routing scheme, InvCap, achieves an SPF at most 30% worse than optimal TE.

The rest of the chapter is as follows. Section 3.1 explains how location diversity changes the TE problem. Section 3.2 presents our simulation setup. Section 3.3 compares the application performance of TE schemes and Section 3.4 compares their achieved capacity under location diversity.

3.1 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.

3.1.1 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 [2], Level-3 [74], EdgeCast [30] etc., commonly replicate content at hundreds of locations and redirect users to the best server based on proximity or dynamic monitoring of server and network congestion [111]. Popular P2P applications such as BitTorrent [34], PPLive [79] download content simultaneously from many peers that are chosen 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 sites; content host-

ing services such as Carpathia [29], Rapidshare [17], 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 [103]. 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 [110, 103]. The fraction of P2P traffic in Internet was estimated to be between 18-60% by different measurement studies in 2009.

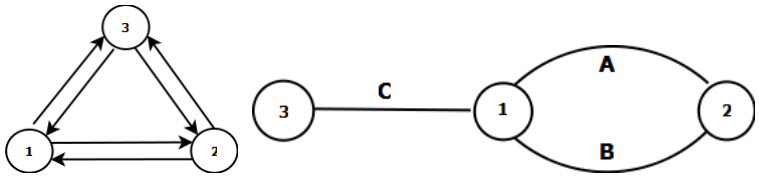
3.1.2 Location diversity: Impact on TE

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

Location diversity can significantly increase the capacity of a network. For example, consider the triangular network in Figure 3.1(a). 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.

3.1.2.1 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.



(a) Triangle network (b) Lasso network

Figure 3.1. Example networks

The three-node network in Figure 3.1(b) 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 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.

3.1.3 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 $4\times$ surge in the load represented by the matrix. Unfortunately, as the example in Figure 3.1(b) 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 3.1(b)). 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 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.

3.1.3.1 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.

3.2 Experimental setup

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

3.2.1 Simulating traffic matrices in ns-2

Figure 3.2.1 illustrates the experimental process. Each simulation has three inputs: (1) *ISP Topology* (2) a sequence of *File Arrivals* at each node based on the current *TM* (3) *Routing*, as computed using a *TE* scheme.

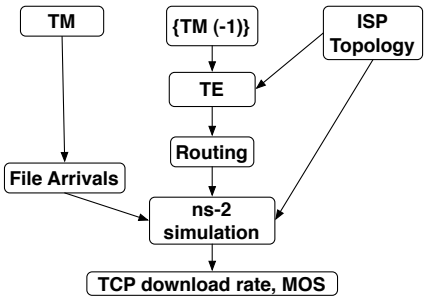


Figure 3.2. Block diagram of experiment process

We construct an ISP network topology from our dataset consisting of PoP-level ISP topology maps. PoPs are represented as nodes and links between these nodes 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 also has five server nodes connected to it via high capacity links that serve files to 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 *TM* to a sequence of *File Arrivals* as follows. Suppose the traffic matrix entry from A to B is 100 Mbps and the duration being simulated is 200 seconds. During the experiment interval, we generate a sequences of file arrivals from A to B whose total size is 100Mbps $\times$ 200 seconds and the sizes are chosen from a realistic distribution.

A traffic engineering scheme *TE* calculates routing for *TM* based on a set of matrices *TM(-1)* which consists of either the current traffic matrix (for Optimal) or a set of matrices from the previous traffic engineering *epoch* (for other TE schemes). The length of the epoch depends on *TE*, e.g., the epoch length for OptWt is 3 hours and for COPE is 1 day. When *TE* yields a routing that splits flows across multiple paths between two nodes, the number of files assigned to each path is proportional to the flow along that path. We use the source routing option in ns-2 to pin a file to a path. We note that the link utilization values obtained using this ns-2 methodology are consistent with those obtained using a simple linear program with the difference being at most 0.1.

In order to make the simulation complexity tractable, we scale down the topology and matrices. ISP backbone link capacities run into tens of Gbps. Simulating such a network at scale even for 100 seconds would require sending data on the order of terabytes (or equivalently, a million 100KB files). Experimentally, we find that simulating at a tenth of this scale, i.e., 100K files, is feasible given the computational and memory constraints of our machines. A typical scale in our simulation is 1/20, i.e., we simulate the backbone link with 1/20 the capacity and also scale down the traffic between each source-destination pair accordingly.

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

Figure 3.3. ISP Data

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	-

Figure 3.4. Bandwidth Distribution

3.2.1.1 ISP topologies and traffic matrices

We use datasets from the following three ISPs for our experiments:

(1) **Abilene**, from the publicly available Abilene ISP data [112]. (2) **Geant**, the un-anonymized version of the Geant topology obtained from the TotemData [94] project personnel. (3) **US-ISP**, a large Tier-1 ISP topology obtained from authors of [122]. TMs for all ISPs were logged in the period from 2004-2005. Figure 3.3 shows number of nodes, number of links, and the interval at which TMs are logged for each ISP. The number of nodes and links for US-ISP is proprietary information.

3.2.1.2 Simulation parameters

Unless otherwise stated, we choose the following parameters for all of our simulations. Our goal is to choose parameters that are close to realistic values for ISPs.
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 most experiments is 300 seconds. We verified that running the simulations for longer durations did not qualitatively affect our results. 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" [87] released by Akamai, one of the largest commercial content distribution networks in operation today. Figure 3.4 tabulates this data for US and Europe.
File sizes: We simulate three file sizes of 100KB, 1MB and 10MB respectively contributing to 8%, 3% and 89% of the total traffic respectively. These values are the fractions 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 [110, 61, 59, 120].
Link delay: We calculate the propagation delay of backbone links from geographic distances between nodes for Geant and US-ISP. For Abilene, we measure the propagation delay of backbone links using *traceroute* and *ping* between PlanetLab [93] nodes in cities where the PoPs are located. All links use drop-tail queuing.

File inter-arrival time: We assume an exponential distribution of file inter-arrival times.

3.2.1.3 Computational resources

We use 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 300–500s 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.

3.2.2 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 that does not "engineer" traffic, but instead simply relies on 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 [88].

OptWt, a shortest-path routing algorithm using link weights computed using a heuristic algorithm to optimize a cost function [50]. We use its implementation in the Totem Toolbox [94]. Typically, ISPs recompute routes 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 minimizes the common-case MLU while limiting the worst-case MLU caused by unpredictable spikes in the traffic matrix. We use the authors' implementation and parameters settings, and recompute routes once a day based on the previous day's TMs as in [117].

MinDelay scheme minimizes average latency for all traffic, unlike the schemes above which optimize link utilization based metrics. Specifically, the objective function MinDelay optimizes is $\sum_{e \in E} d_e T_e$, where E is the set of all links, d_e is the propagation delay of link $e \in E$, and T_e is the total traffic (in bits/sec) for $e \in E$. We evaluate MinDelay to answer whether end-to-end application performance improves if we optimize latency instead of MLU. We constrain the linear program so that MLU does not exceed 0.6, thereby ensuring that high link utilization does not hurt application performance. Like Optimal, MinDelay has perfect knowledge of traffic matrix.

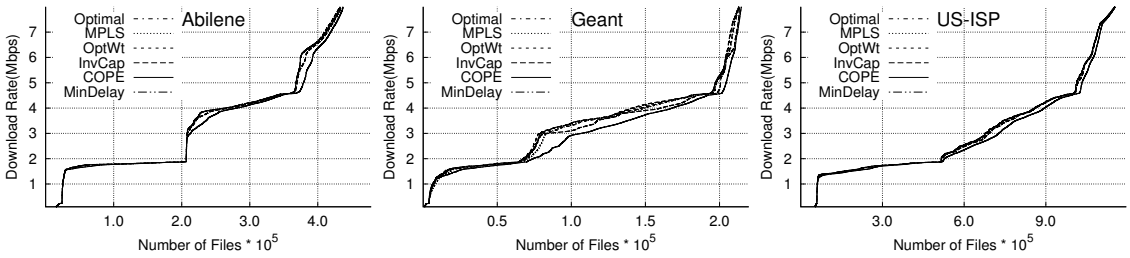


Figure 3.5. Download rate CDFs for all TE schemes are near identical except COPE which has slightly lower performance

3.3 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.

3.3.1 TCP performance

We simulate 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 3.2.1.

Figure 3.6 shows the mean download rate of files, where the average is across all files across all of the simulated matrices for each TE scheme. We make three observations from this graph. First, all schemes achieve nearly same mean download rates with the exception of COPE that is consistently worse by up to 10%. Next, Optimal (the leftmost bar in each group) is not always the best as minimizing MLU is not the same as optimizing TCP performance. Finally, MinDelay (the leftmost bar in each group) that optimizes latency performs the same as other TE schemes that optimize link utilization based metrics.

Figure 3.5 shows the corresponding CDFs for the mean download rates in Figure 3.6. 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

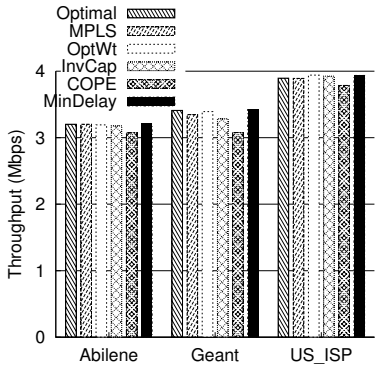


Figure 3.6. Mean download rates

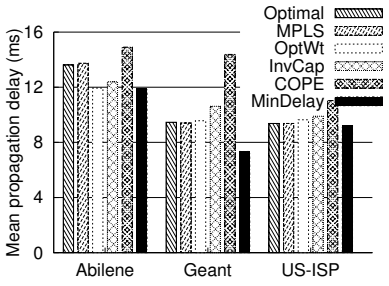


Figure 3.7. COPE has the highest propagation delay among TE schemes

transfers. This observation partly explains why MinDelay fails to improve TCP throughput over other schemes: TCP throughput cannot be improved for flows bottlenecked at access link even if MinDelay scheme reduced the RTT for these flows.

3.3.1.1 MLU vs. TCP performance

To further investigate the results in Figure 3.6 and Figure 3.5, we analyze the empirically observed MLU for all TE schemes in the experiments. Figure 3.8 plots the MLUs for all matrices considered. For US-ISP the MLU data is proprietary, so we present the ratio of MLU with respect to Optimal. As expected, different TE schemes do show substantially different MLUs. For example, the MLU for InvCap and OptWt is up to twice the MLU of Optimal in some cases; MinDelay has a MLU of 0.6 on Geant, which is more than twice of Optimal's MLU. 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 experiments, 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 ISP network [25] 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 hardly impacts the average 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.

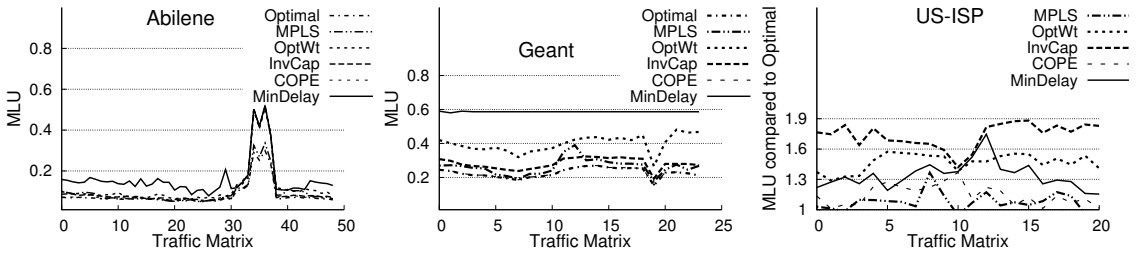


Figure 3.8. TE schemes differ as much as 2× in MLU

File download rates are affected only for flows on this link, which may be a tiny fraction of the total traffic.

3.3.1.2 The price of predictability

Why is COPE's performance consistently 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 3.7 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 [20]. COPE intentionally routes some traffic along longer paths so as to leave room for occasional traffic spikes along shorter paths. While this approach makes COPE robust with respect to MLU under 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., [20], is likely to incur a similar penalty in application performance in the common case.

3.3.2 UDP performance

3.3.2.1 Measuring UDP performance

We assume that the loss rate and the queuing delay on each link for UDP traffic is the same as that measured during experiments with TCP traffic. This assumption is reasonable as TCP accounts for over 90% of Internet traffic [53]. We calculate the loss rate and delay for a path by combining the loss rates of links along the path; we compute the delay by summing the propagation and queuing delay of links along the path.

We compare performance of VoIP traffic (which uses UDP) using Mean Opinion Score (MOS). MOS is an industry standard VoIP call quality metric for which a

score of above 4 is considered good and below 3 is considered bad. We calculate MOS using the formula in [35] which calculates MOS given the loss rate and delay for a path.

We calculate MOS for VoIP calls between all pairs of source and destination PoP nodes in an ISP. First, we measure loss rates and queuing delay on backbone links for each 10-second interval. For each interval, we calculate the MOS for a path based on its end-to-end loss rate and delay. The mean MOS for a path is the average value of MOS over all intervals. For TE schemes that split traffic across multiple paths between a source-destination pair, the mean MOS for a source-destination node pair is calculated as the weighted average of mean MOS weighted by the fraction of the traffic split along each path between the node pair. We similarly calculate the 5th percentile MOS for a source-destination pair by taking the weighted average of 5th percentile MOS values for all its paths.

3.3.2.2 Results

We obtain a distribution of mean MOS values for a TE scheme by combining mean MOS values for all pairs of source and destination nodes for all traffic matrices. We find that the minimum and the maximum values of mean MOS for all TE schemes are in the range (4.08, 4.14) for Abilene, (4.07, 4.14) for Geant and (4.08, 4.14) for US-ISP. The range of values for 5th percentile MOS are (4.07, 4.13) for Abilene, (4.08, 4.14) for Geant and (4.05, 4.14) for US-ISP. MOS scores for all schemes are always above 4.0 and the differences between different TE schemes is at most 0.1. These results are not surprising since loss rates and queuing delay are near-negligible for most links in the network. Furthermore, MOS is not very sensitive to few milliseconds difference in propagation delay among TE schemes.

3.4 Capacity and location diversity

The results in the previous section may seem unsurprising—different TE schemes yield nearly identical application performance simply because today's low traffic demand levels obviate the need to engineer traffic. However, in this section, we show that similar conclusions hold when we compare TE schemes with respect to their potential capacity, i.e., their ability to accommodate surges in traffic demand in the future.

The key factor that explains our unexpected findings is location diversity, i.e., the ability to download content from multiple locations. Our main findings are that (1) location diversity can significantly increase the capacity (by up to 2×) 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.

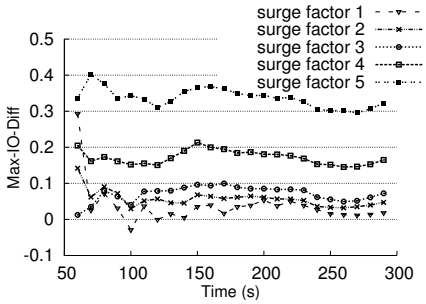


Figure 3.9. Profile of Max-IO-Diff at increasing surge factors for a Geant TM

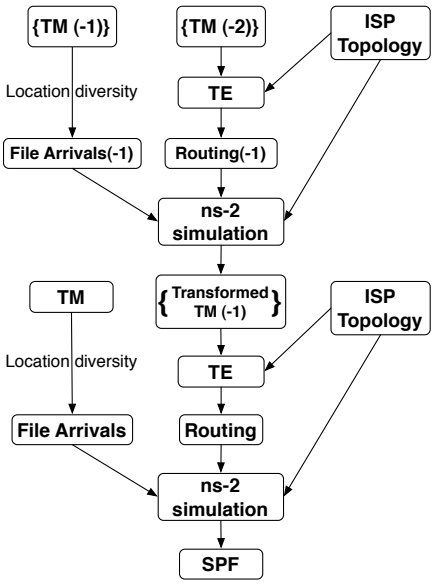


Figure 3.10. Block diagram of experiment process with location diversity

3.4.1 Empirically measuring capacity

Our metric of capacity is the SPF, i.e., the maximum surge in demand that can be satisfied (as defined formally in Section 3.1.3.1). 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 3.9, we show a Max-IO-Diff profile for five experiments at surge factors of 1, 2, 3, 4 and 5 for a Geant TM with InvCap routing. The graph shows the Max-IO-Diff measured at intervals of 10 seconds throughout the simulation. We ignore the first 50 seconds of simulation as the 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.

3.4.2 Simulating location diversity

Figure 3.10 illustrates the experimental process with location diversity. The lower half is similar to Figure 3.2.1 with two differences. First, to incorporate location diversity, we modify the procedure to transform *TM* to *File Arrivals* as follows. As in Section 3.2, we first transform PoP-to-PoP entries in *TM* to a sequence of file download requests. However, instead of downloading each file from just that one location, k-1 additional randomly chosen source locations are introduced so as to emulate a location diversity of k. The file is downloaded in parallel from all k locations using parallel TCPs. The download is considered complete when the total bytes downloaded across all k locations equals the size of the file.

Second, application adaptation to location diversity changes the input to *TE* as indicated by the block *Transformed TM(-1)* that is obtained as follows. Let *TM(-1)* and *TM(-2)* respectively denote the (set of) matrix(ces) in the last and last-to-last epochs. Recall that *TE* determines the length of the epoch (0 for Optimal, 3 hours for OptWt and MPLS and a day for COPE). *Transformed TM(-1)* is generated by the top simulation that takes as input the file arrivals obtained from *TM(-1)* and *Routing (-1)*. The latter is obtained by applying *TE* to *TM(-2)* in the previous epoch. This two-step simulation is intended to approximate the interaction of TE and application adaptation to location diversity that changes the TM.

3.4.3 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.

3.4.4 Capacity increase with location diversity

In Figure 3.11, 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

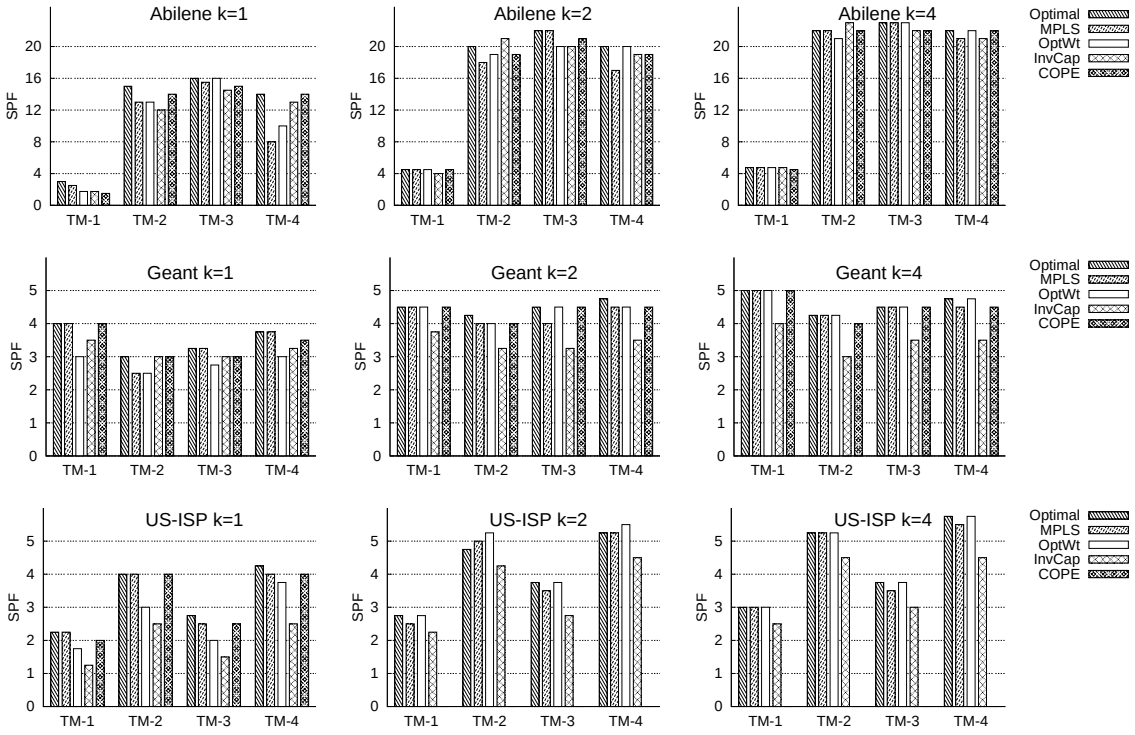


Figure 3.11. Comparison of SPF among TE schemes for different levels of location diversity; SPF values are obtained using ns-2 simulations

TE/Optimal	k=1	k=2	k=4
Optimal/Optimal	1	1	1
MPLS/ Optimal	0.89	0.98	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

Figure 3.12. Comparison of SPF values

where barrier-crossover method to solve a 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 3.12 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 2% of Optimal for $k = 4$ as well as $k = 2$. Figure 3.11 shows that with location diversity any TE scheme has at most 10% capacity difference compared to Optimal. On average InvCap has 15% less capacity compared to Optimal for a location diversity of $k = 4$. In the worst case InvCap achieves a capacity that is 30% less than Optimal (Figure 3.11, Geant k = 2).

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 sub-optimal 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.

3.4.4.1 Other results

We briefly summarize other experimental results deferred to a technical report [11]. First, SPF increases in a concave manner with the fraction of traffic that can leverage location diversity. Even if only half of the traffic has location diversity, it suffices to capture over 90% of the potential increase in SPF for each TE scheme, and the SPFs achieved by different TE schemes continues to be less that 5%. Second, the "near-optimality" of capacity achieved by all TE schemes is reflected not only in their SPFs but in application performance metrics as well, i.e., TCP download rates and MOS scores (in the mean as well as across various percentiles) degrade similarly for all TE schemes as the demand approaches the SPF capacity point. As expected, application performance starts to dip earlier under InvCap as its SPF is somewhat lower than TE schemes. Thus, these results also suggest that, unlike link utilization metrics, SPF is a sound empirical metric to measure how effectively a TE scheme can accommodate load surges under location diversity.

3.5 Conclusion

Our application-centric focus and empirical evaluation methodology reveal unexpected results that challenge conventional wisdom in traffic engineering. We find that link utilization, the most widely used metric to evaluate TE, 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 link utilization in order to accommodate unexpected traffic spikes 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. With location diversity, even static routing achieves a capacity that is at most 30% (and typically significantly less) worse than the optimal. Taken together,

our findings suggest that it matters little which TE scheme is used (or whether TE is used at all) at today's traffic levels as well as under reasonable projections of increased demand. A provocative message to network operators is to stop engineering traffic and let end-users do the work for them.