# Latency Inflation with MPLS-based Traffic Engineering

Abhinav Pathak
Purdue University
pathaka@purdue.edu

Ming Zhang
Microsoft Research
mzh@microsoft.com

Y. Charlie Hu
Purdue University
ychu@purdue.edu

Ratul Mahajan
Microsoft Research
ratul@microsoft.com

Dave Maltz
Microsoft Research
dmaltz@microsoft.com

## ABSTRACT

While MPLS has been extensively deployed in recent years, little is known about its behavior in practice. We examine the performance of MPLS in Microsoft's online service network (MSN), a well-provisioned multi-continent production network connecting tens of data centers. Using detailed traces collected over a 2-month period, we find that many paths experience significantly inflated latencies. We correlate occurrences of latency inflation with routers, links, and DC-pairs. This analysis sheds light on the causes of latency inflation and suggests several avenues for alleviating the problem.

## Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Performance Attributes

## General Terms

Measurement, Performance

## Keywords

MPLS, LSP, Autobandwidth, Latency

## 1. INTRODUCTION

Traffic engineering (TE) is the process of deciding how traffic is routed through the service provider network. Its goal is to accommodate the given traffic matrix (from ingress to egress routers) while optimizing for performance objectives of low latency and loss rate. Effective TE mechanisms are key to efficiently using network resources and maintaining good performance for traffic.

The importance of TE has motivated the development of many schemes (*e.g.,* [5, 7]), but little is known today about the effectiveness or behavior of schemes that have been deployed in practice. Much of the prior work is based on various forms of simulations and emulations rather than based on real measurements taken from an operational network.

In this paper, we present a case study of the behavior of TE as deployed in a large network. This network (MSN) is the one that connects Microsoft's data centers to each other and to peering ISPs.
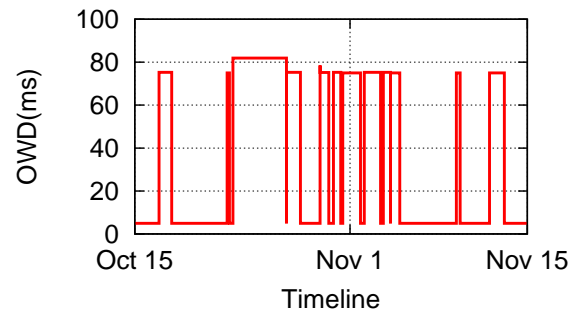
**Figure 1: Tunnel latency during one month period**

MSN uses MPLS-based TE, which is perhaps the most widely used TE mechanism and is supported by major router vendors such as Cisco and Juniper. Although we use MSN as a case study, we believe that our findings are general because the behaviors we uncover are tied to the MPLS-TE algorithms themselves rather than to MSN. The contribution of our work lies in uncovering and quantifying problems with MPLS-TE in a production network, as a first step towards improving the TE algorithms.

Our interest in studying the behavior of MPLS-TE was not purely academic but was motivated by anomalous behavior observed by the operators of Bing Search which uses the MSN network. During the period of our study, Bing Search experienced incidents of unexpectedly high latencies between two of its DCs from time to time. Because these two DCs are in the same city, Bing's operations expected the latency between them to be negligible. In fact, this assumption is also made by Bing's planning team when they chose to distribute Bing's backend services across the two DCs.

Figure 1 plots the latency of a tunnel (also known as LSP for label switched path) between the two DCs during a month period. We describe how MPLS-TE works in more detail later; but briefly, it works at the granularity of tunnels between ingress and egress routers. There can be multiple tunnels between a pair of routers. MPLS-TE uses a greedy algorithm that periodically finds the shortest path that can accommodate the tunnel's estimated traffic demand. The figure shows that the tunnel latency switched frequently between 5 and 75 ms. It stayed at 75 ms for almost half of the time, which adversely impacted Bing's backend services.

Our systematic evaluation using data from a 2-month period in 2010 reveals that it is not only these two DCs that are impacted. 22% of the DC pairs experience significant latency spikes. 20% of the tunnels exhibit more than 20 ms of latency spikes. Over 5% of the tunnels experience high latency inflation for a cumulative duration of over 10 days in the 2 months.
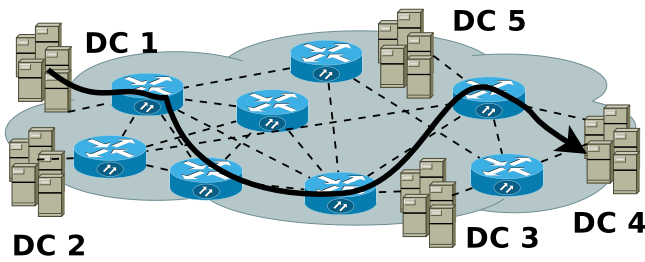
**Figure 2: OSP network topology**

To gain insight into the causes for latency inflation, we correlate such occurrences with specific links, routers, and DC pairs. Our analysis shows that 80% of latency inflation occur due to changes in tunnel paths concentrated on 9% of the links, 30% of the routers, and 3% of the active DC-pairs. This confirms traffic load changes exceeding the capacity of a small set of links along the shortest paths of tunnels as the primary culprit. MSN operators have since added capacity along these paths to alleviate the problem.

But to understand the effectiveness of MPLS at using available resources, we compare the latency with MPLS-TE to that with an optimal strategy based on linear programming. We find that the weighted and the $99^{th}$ percentile byte latency under MPLS-TE are 10%-22% and 35%-40% higher than that under optimal routing strategy, respectively, suggesting there exists room for improvement under MPLS-TE.

We identify several problems caused by sub-optimal setting of MPLS parameters but leave as future work automatic parameter setting and *on-the-fly* LSP split as two methods to fix the latency inflation problem.

## 2. BACKGROUND

A service provider network is composed of multiple points-of-presence (PoPs). Our work is in the context of an online service provider (OSP), where these PoPs serve as data centers (DCs) for hosting services as well as peering with neighboring transit ISPs (Internet service provider). The distinction between an OSP and ISP is not important for our work, though we note that the nature of traffic may be different in these two kinds of networks and there is a higher premium placed on latency reduction in OSP networks.

Figure 2 illustrates the topology of a large OSP network. It comprises multiple DCs at different geographical locations to serve users around the world. To save inter-DC bandwidth cost, these DCs are often interconnected with dedicated or leased links, forming a mesh-like topology.

### 2.1 MPLS-TE Basics

A growing number of OSPs and ISPs have adopted MPLS networks which offer more TE [2] flexibility than the traditional IGPs such as OSPF and IS-IS. The former allows traffic to be arbitrarily distributed and routed between a source and a destination while the latter only allows traffic to be evenly distributed and routed on the shortest paths. Such restriction may cause IGP TE to be far from optimal under certain circumstances.

**LSP: Label Switched Path.** An LSP is an one-way tunnel in MPLS network over which data packets are routed. Packets are forwarded using MPLS labels instead of IP addresses inside LSP tunnels. The labels are inserted into packets according to local policy at ingress routers, which are later stripped by egress routers. Unlike in IGP routing, an LSP tunnel does not have to follow the shortest path from an ingress to egress. In an OSP network, each DC pair is provided with multiple LSPs in either direction to leverage path diversity in the underlying physical network. Traffic between the same DC pair can be split among different LSPs either equally or unequally.

An LSP has several attributes such as the current path, allocated bandwidth, priority, *etc.*. There are two types of LSP: static and dynamic. The former is allocated a static bandwidth and path at setup stage which remains the same thereafter. The latter continually monitors the traffic rate flowing through the tunnel and adapts its allocated bandwidth accordingly. It may also switch path when there are changes in its own allocated bandwidth or the available bandwidth in the network.

### 2.2 MPLS-TE algorithms

An LSP path is either configured manually or computed using Constrained Shortest Path First Algorithm (CSPF). After a path is selected, the LSP reserves the required bandwidth at the outgoing interface of each router along the path. Each router outgoing interface maintains a counter for its current reservable bandwidth. The reservable bandwidth information along with network topology (also called Traffic Engineering Database (TED)) is periodically flooded throughout the network.

**Priority and preemption.** Each LSP is configured with two priority values: *setup priority* and *hold priority*. Setup priority determines whether a new LSP can be established by preempting an existing LSP. Hold priority determines to what extent an existing LSP can keep its reservation. A new LSP with high setup priority can preempt an existing LSP with low hold priority if: (a) there is insufficient reservable bandwidth in the network; and (b) the new LSP cannot be setup unless the existing LSP is torn down.

**CSPF.** CSPF sorts LSPs based on their priority and uses a greedy algorithm to select the shortest path for each LSP. Starting with the highest priority LSP, it prunes the TED to remove links that do not have sufficient reservable bandwidth or do not satisfy a preconfigured access control policy. It then assigns the shortest path in the pruned TED (using tie-breaking if multiple) to the LSP and updates the reservable bandwidth on the relevant links. This process continues until no LSP is left.

**Re-optimization.** CSPF is run periodically based on a configurable timer to reassign each LSP a better path if possible.

**Autobandwidth.** MPLS does not have a bandwidth policing mechanism — an LSP may carry any traffic demand irrespective of its reserved bandwidth. Instead, router vendors (Cisco, Juniper) supporting MPLS, provides *autobandwidth* which permits an LSP to adjust its reserved bandwidth according to current traffic demand. To use autobandwidth, an LSP needs several additional parameters (Table 1), including minimum/maximum bandwidth, adjustment threshold, adjustment interval and sampling interval. Once every sampling interval (*e.g.,* 5 minutes), an LSP measures the average traffic demand flowing through it. Once every adjustment interval (*e.g.,* 15 minutes), it computes the maximum of the average traffic demand measured in each sampling interval. If the maximum traffic demand differs from the current reserved bandwidth by more than the adjust threshold and is within the minimum and maximum bandwidth, the LSP will invoke CSPF with the maximum traffic demand as the new reserved bandwidth.

## 3. EXPERIMENTAL METHODOLOGY

To study how MPLS-TE algorithms affect inter-DC traffic, we collected various types of data from October 15 to December 5 2010 from a portion of Microsoft's intercontinental production network (MSN), one of the largest OSP networks today. This portion of MSN comprises of several tens of DCs interconnected with

**Table 1: Autobandwidth parameters**

| Parameter | Description |
|-----------|-------------|
| Subscription factor | % of interface bandwidth that can be reserved |
| Adjust interval | Time interval to trigger autobandwidth |
| Adjust threshold | % of change in reserved bandwidth to trigger autobandwidth |
| Min/max bw | Bandwidth limits of an LSP |
| Setup/hold priority | Priorities for determining LSP preemption |

high-speed dedicated links with the core of network in US. All the inter-DC traffic is carried over 5K LSPs, each using autobandwidth, with 1-32 LSPs between each pair of DC. The data contains network topology and router and LSP configurations. For each LSP, it also contains each path change event and traffic volume in each 5-minute sampling interval.

Measuring LSP latency is a challenging task for two reasons (a) LSPs are unidirectional; as a result a simple ping would return One-Way Delay (OWD) latencies of two LSPs (the forward LSP and the reverse direction LSP). Separating out the two latencies would require strict time synchronization between the probers across DCs (b) Traffic between a DC pair is load balanced using hashing algorithms on all the LSPs (between 1 to 30) between the DCs. Hash functions are based on IP/TCP or even application level headers. As a result, to probe all the LSPs between a DC pair using simple ping, we must have one prober covering all the possible IP ranges allocated as well as a applications running in DCs.

Another way to measure LSP latency is to use LSP ping [6, 9]. However, because LSP ping is disabled in MSN , we choose to estimate LSP latency based on the geographical locations of the routers along an LSP path. Given an LSP, we calculate the great-circle distance between each pair of intermediate routers and sum it up to obtain the total geographical distance of the LSP. We then dividing the total distance by the speed of light in fiber to obtain the LSP latency. We verified for a few LSPs that the conversion indeed estimates correct delay with minimal error. Note that LSP is unidirectional, as a result, the latency measured in this mechanism is One-Way-Delay (OWD) estimation.

# 4. LSP LATENCY INFLATION

In this section we first describe the severity of the latency problem in an MPLS based network and then correlate latency inducing LSP path changes with dc pairs, routers and links in the network.

## 4.1 How badly is latency inflated?

**Prevalence of latency inflation** To quantify how widespread latency inflation is, we compute the difference between the minimum and maximum latency for each LSP during the 50-day period. Figure 3 plots the CDF of latency difference of all LSPs. We observe that a substantial number of LSPs encounter severe latency inflation. 20% (over 1K) of the LSPs experience latency inflation of over 20 ms. Moreover, the latency of 10% (over 500) of the LSPs is inflated by more than 40 ms! Because a single user request may trigger many round trips of inter-DC communication, such latency inflation could noticeably impair user-perceived performance.

To systematically measure the frequency and duration of latency inflation, we define a *latency spike* as the contiguous period of time during which the latency of an LSP is at least $x$ ms *and* $y$% more than the minimum latency observed for the LSP. These two conditions capture the significance of latency inflation in both absolute and relative terms. As shown in Figure 4, a spike starts when both conditions are met and ends when either condition becomes false.
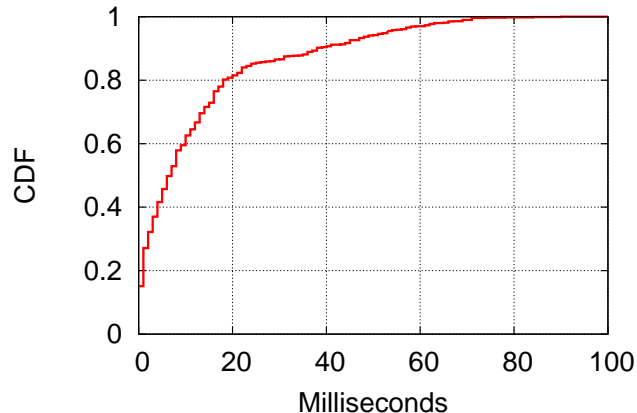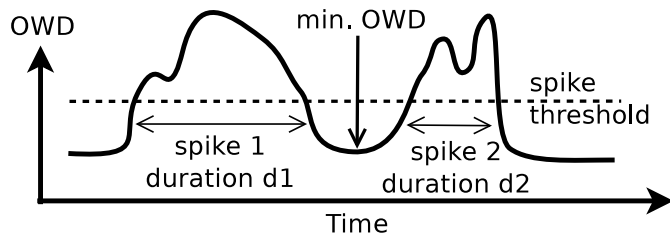


**Figure 3: LSP latency difference (max-min) CDF**



**Figure 4: Definition of** $spike(x, y)$

Figure 5 and 6 plot the CDF of total number and cumulative duration of latency spikes for each LSP. We observe that latency inflation is quite common. Under the (20ms, 20%) spike threshold, roughly 18% (over 900) of the LSPs experience at least one latency spike during the 50-day period. For 10% (over 500) of the LSPs, the cumulative duration of latency spikes is over 1 day. This problem becomes even more severe for the top 5% (250) of the LSPs whose cumulative spike duration is more than 10 days! This indicates persistent latency problem for the inter-DC traffic carried by those LSPs. Figure 5 and 6 also show similar curves under a more aggressive spike threshold of (30ms, 30%), where the total number and duration of latency spikes are only slightly smaller. This suggests the latency inflation experienced by many of the LSPs is indeed quite significant.

The traffic in the core of MSN network consists of only the traffic generated by inter-DC communications. All the 5K LSPs in the network use autobandwidth algorithms to manage paths and reserve bandwidth. Inter-DC links in the core of network exhibit over 99.9% of availability [4]. However, most LSPs exhibits long cumulative durations in spikes in order of days(figure 6). This suggests that severe LSPs spikes are caused by autobandwidth instead of failures.

**Comparison with optimal TE strategy** Although we have shown many LSPs encounter latency spikes frequently, so far it is unclear if those spikes are caused by insufficient network capacity or by inefficiency of MPLS-TE algorithms. To answer this question, we compute the *optimal* TE strategy that minimizes the weighted byte latency($\sum_{\forall LSP} lat. * bw / \sum_{\forall LSP} bw$) for all inter-DC traffic. Given the network topology and traffic matrix, this can be formulated as a multi-commodity flow problem and solved using linear programming (LP) [8, 3, 1]. Note that although it is relatively easy to find the optimal TE strategy offline, the problem is much harder
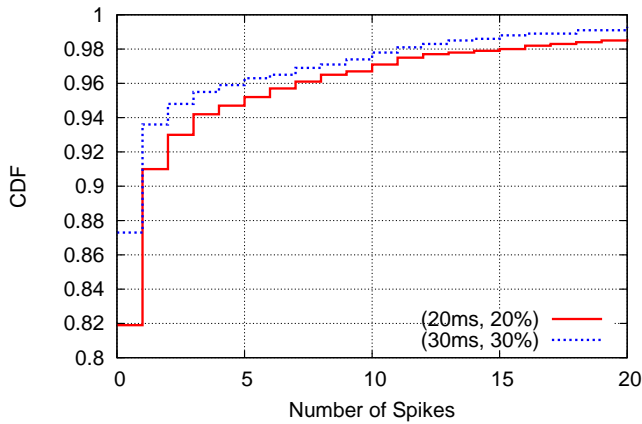
Figure 5: CDF of number of spikes observed by LSPs for different values of spikes
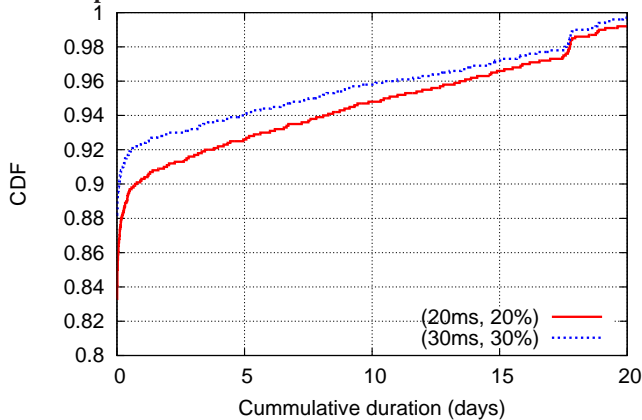


Figure 6: CDF of cumulative duration of spikes observed by LSPs for different values of spikes



Figure 7: Byte latency bar chart for LSP and LP



Figure 8: $99^{th}$ percentile byte latency under MPLS (LSP) and Optimal Routing (LP)

to tackle online due to the of size of the topology (resulting in million variable LP) and volatility traffic demand.

We divide the time into 5-minute intervals and compute the optimal TE strategy for each interval using the method stated above. Compared to optimal routing, we found that MPLS based routing incurs an overall 10% to 22% increase in weighted byte latency over different snapshots spanning over a one day period. Figure 7 compares the latency at different traffic volume percentile under the optimal and MPLS-TE in a typical interval. There is substantial latency gap between the two TE strategies — the relative latency difference stays above 30% (y2 axis) at 50th, 90th, 95th, and 99th-percentile of traffic volume. Figure 8 plots the latency at the 99th-percentile of traffic volume under both TE strategies during an entire day. Except between midnight and early morning hour, the latency under MPLS-TE is consistently 20 ms (35%-40%) larger than that under the optimal TE, leaving enough space for improvement.

## 4.2 Is there a pattern in latency inflation?

LSP latency inflation is triggered by an LSP switching from a short to a long path. We now study the patterns of LSP path changes to see if they cluster at certain links, routers, DC pairs or time periods. Although there are many LSP path changes, we consider only those that cause a latency spike (*e.g.,*, latency jumps by more than 20 ms and 20%) and call them LLPC's (large latency path changes). We ignore the remaining path changes since they either have little
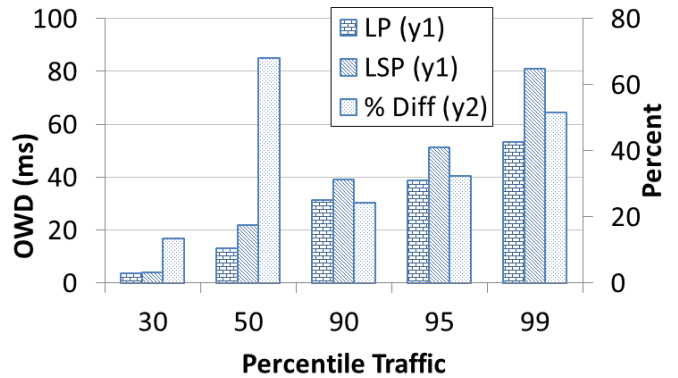
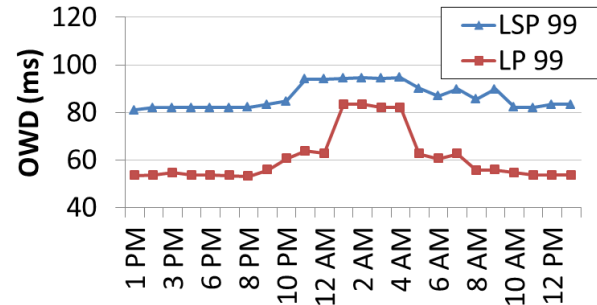impact on latency or reduce latency. For an LLPC, we attribute it to the old path rather than the new one because it is triggered by insufficient bandwidth on the former.

**Correlation with links, routers and DC pairs.** We first correlate each LLPC with the links, routers, and DC pair the corresponding LSP traverses. In Figure 9(a), the y-axis on the left shows the number of LLPC's per link sorted in an increasing order and the y-axis on the right shows the cumulative fraction of LLPC's observed by the links. The x-axis is normalized to anonymize the total number of links in MSN . Figure 9(b) and 9(c) plots similar curves for routers and DC pairs respectively. From these figures, we find that the LLPC's occur mostly at a small fraction of links and DC pairs — the top 10% of links and DC pairs account for 80% and 95% of the LLPC's respectively. This pattern is true for routers as well, although less pronounced. Our analysis suggests that the latency inflation problem could be significantly alleviated by adding capacity to a small subset of links.

**Correlation with time.** Next we correlate LLPC's with time. We divide the time into 1-hour bins and compute the number of LLPC's observed per (link, time-bin) pair. Figure 10(a) plots the number of LLPC's of each (link, time-bin) pair in an increasing order and the cumulative fraction of LLPC's of all the pairs. It shows the LLPC's are highly concentrated both at certain links and in certain time. 1% of the (link, time-bin) pairs witness 80% of the LLPC's. This distribution is even more skewed than that in Figure 9(a). Since bandwidth change is the primary cause of LSP path changes (§2.2), this is likely due to dramatic traffic surge in those (link, time-bin) pairs. We observe similar patterns for (router, time-bin) and (DC-pair, time-bin) pairs which are illustrated in Figure 10(b,c) respectively.
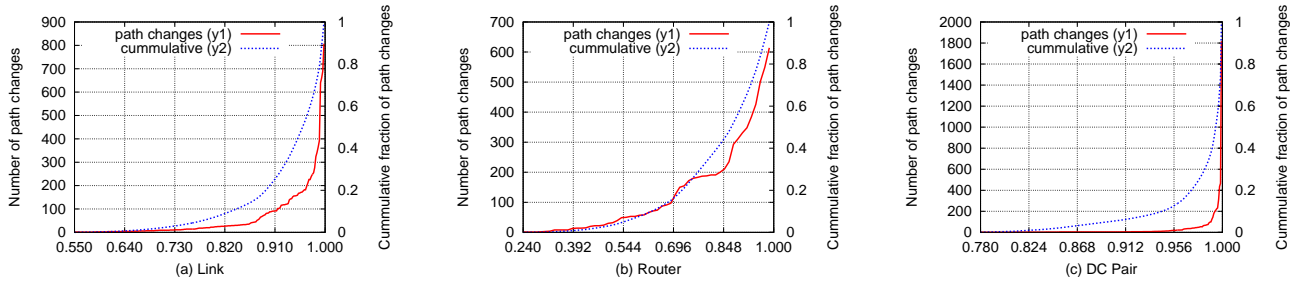
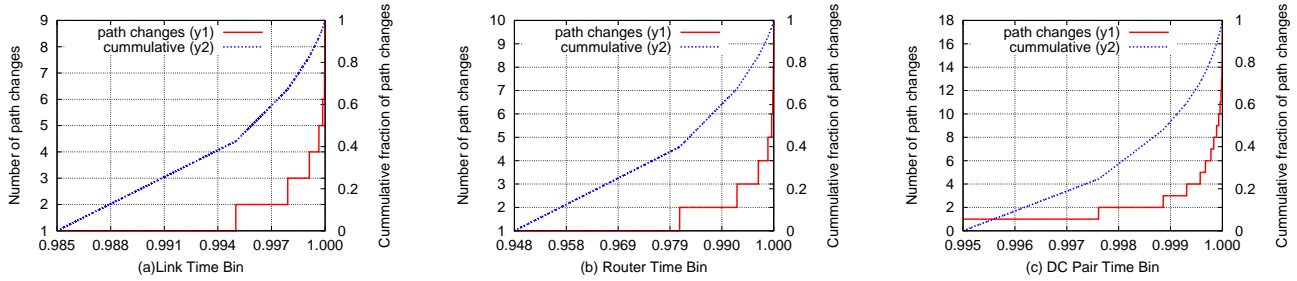**Figure 9: Histogram and cumulative path change per link, router and dcpair**



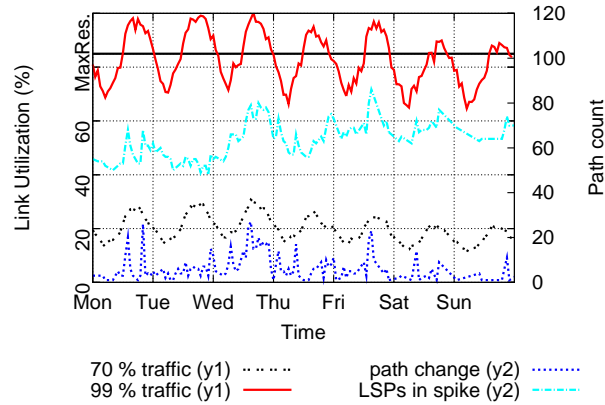**Figure 10: Histogram and cumulative path change per time bin per link, router and dcpair**



**Figure 11: Link utilization and path change count as a function of time of day for one week period**

**Table 2: Impact of MPLS autobandwidth parameters**

| Parameter | Low | High |
|---|---|---|
| min bw | Large # of path changes | reserved bw wastage |
| max bw | No guarantee of traffic delivery if high requirement | harder to find new path (same as static LSP) |
| setup priority | Harder to find better path | Easy to find better path |
| hold priority | Easily give up current path | Stick to current path |
| adjust thres. | High # of path changes | LSP growth is harder |
| subscription | Less headroom for LSPs | Resource wastage |
| # of LSPs | High traffic per LSP | High LSP overhead |

**Correlation with link utilization.** Finally we study the impact of link utilization on LLPC's and latency inflation. Again we divide the time into 1-hour bins. In Figure 11, the left y-axis shows the 70th and 99th-percentile link utilization in each time bin during one week (the MaxRes line shows the maximum allowed reservation in links: configured at 85%). The right y-axis shows the number of LLPC's and the number of LSPs in latency spike in each time bin during the same period. We observe that although MSN is generally over-provisioned (70th-percentile link utilization is only around 20%), there are always certain links which are fully saturated (99th-percentile link utilization is around 100%). This is somewhat surprising because we would expect the traffic load is more evenly distributed during off-peak hours when there is abundant spare capacity. However, remember that in autobandwidth each LSP greedily looks for the shortest path, therefore it is possible that many LSPs are competing for just a few "critical" links even though other links are left idling.

Figure 11 also illustrates the number of LLPC's and number of

LSPs in latency spike are strongly correlated with link utilization while all of them exhibit a clear time-of-day pattern. This conforms to our expectation since more LSPs will be forced to switch to longer path when the overall network utilization becomes higher.

## 5. EFFECT OF AUTOBW PARAMETERS

LSP path changes and corresponding latency spikes are the direct consequences of both the autobandwidth algorithm and individual LSP's configuration parameters. Given the large number of LSPs in an OSP network, it is a common practice for operators to manually configure the LSP parameters with some static values which rarely change thereafter. In many cases, operators simply use the default values set by router vendors, *e.g.,* Cisco and Juniper. We study the impact of different LSP parameters on LSP latency spikes and summarize our findings in Table 2.

**LSP priority** We studied a few large latency spikes in LSPs traversing nearby DCs (as in figure 1). In this set of latency spikes, the LSPs, instead of traversing the direct shortest path between the DCs, traverse a considerably longer path (across the US). This was because the reserved bandwidth of the direct shortest path between the DCs was exhausted by another sets of LSPs with (mostly) equal or higher priority. Some of these LSPs traversed long distance and had several different path options available. Their decision to choose the particular link resulted in saturation of its reserved bandwidth. Later due to a bandwidth increase of a nearby DC LSP,

autobandwidth moved the LSP to a much longer path. The situation gets complicated when several such improper path selections form a chain of dependencies.

The spike in cases like these could have been mitigated by increasing the priority of the LSP between the nearby DCs. But setting priority across thousands of LSPs to globally minimize latency in the network is a hard problem. Further, it is also unclear whether a static set of priorities would be sufficient to reduce the problem. As a future work, we plan to investigate how to automatically adjust priorities for the LSPs in latency spike in an online manner, to force them to switch to a shorter path, while ensuring minimal impact on other LSPs.

**"All or nothing" autobandwidth policy** The second cause for latency spikes in LSPs stems from the 'all-or-nothing' policy of autobandwidth algorithm. This severely impacts high-volume LSPs. A bandwidth increase of an LSP, running on a short path, where at least one of the links is close to its reservation limit, forces the entire LSP to another, long latency path. This results in the entire LSP traffic to traverse the long path even though the short path is capable of carrying most of the traffic.

As a part of future work, we plan to devise algorithms to split LSPs in such cases (currently done manually in some networks [10]). When an entire LSP will be forced to switch to a long path due to its traffic demand increase, we could subdivide the LSP (*on-the-fly* LSP split up) into two smaller ones so that only the increased traffic traverses longer path.

**Minimum/maximum bandwidth** Minimum and maximum bandwidth specify the bounds of the LSP bandwidth. A low minimum bandwidth value, renders an LSP fickle, triggering large number of path changes (minimal latency difference) since a small increase in bandwidth (a few 100KBs is sufficient) is now sufficient to trigger the bandwidth threshold. LSPs in this case change their path even though current path has sufficient available bandwidth because the autobandwidth tie-breaking algorithms (random, least filled, most filled) forces them to migrate to another equal cost path. A high value of minimum bandwidth wastes reservable bandwidth in the network.

A lower maximum bandwidth bounds the LSP bandwidth reservation forcing the fate of additional traffic on LSP to be uncertain and requires larger number of LSPs to be setup between the DC pair to accommodate the entire traffic. Each LSP incurs additional overhead in terms of computation and storage on ingress router and on the network. A large value for maximum bandwidth parameter makes LSP immobile, since during load, it gets harder for LSP to find path with free large reservable bandwidth. The 'all-or-nothing' policy causes further complications

**Bandwidth threshold and subscription factor** Bandwidth threshold dictates when should the autobandwidth algorithm be triggered. A small value renders and LSP to be fickle while a large value makes the LSP less responsive and requires larger headroom on link to absorb additional bandwidth. Similarly subscription factor which determines what fraction of link capacity must be reserved, play an important role. A small value wastes network capacity, while a large value diminishes the headroom for LSPs to grow so that autobandwidth gets triggered.

As a part of future work we will study how to adjust the autobandwidth parameters automatically. Also, so far, we have assumed all the LSPs and traffic are equally important. However, in an OSP network, different application traffic usually have distinct latency requirement. For instance, the traffic of most web applications is latency-sensitive, but the traffic of backup and replicate applications is not. A "smart" TE strategy would optimize latency only for the former while provisioning sufficient bandwidth for the latter. Such strategy could be implemented by classifying application traffic into delay-sensitive and delay-tolerant LSPs and assigning higher priority to the former type of LSPs. We will study effectiveness of such classification as our future work.

# 6. CONCLUSION

In this paper, we presented the first study of the effectiveness of MPLS-TE in a multi-continent production network connecting tens of data centers. Using detailed LSP traces collected over a 2-month period, we showed that a substantial number of LSPs encounter severe latency inflation. We further showed that 80% of latency inflation occur due to LSP path changes concentrated on 9% of the links, 30% of the routers, and 3% of the active DC-pairs. Our analysis confirms traffic load changes exceeding the capacity of a subset of links along the shortest paths of LSPs as the primary root cause of latency inflation but also uncovers poor configuration of MPLS-TE's autobandwidth algorithms in the studied network as a source of inefficacy. As future work, we are developing guidelines and automatic schemes to adjust autobandwidth configurations to changing traffic loads.

# 7. REFERENCES

[1] D. Applegate and E. Cohen. Making intra-domain routing robust to changing and uncertain traffic demands: Understanding fundamental tradeoffs. In *Proc. of SIGCOMM*. ACM, 2003.

[2] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow. RFC 3209: RSVPâĂŤTE: exâĂŤtensions to RSVP for LSP tunnels. 2001.

[3] D. Bertsekas and R. Gallager. *Data networks*. Prentice-hall New York, 1992.

[4] P. Gill, N. Jain, and N. Nagappan. Understanding network failures in data centers: measurement, analysis, and implications. In *Proc. of SIGCOMM*. ACM, 2011.

[5] S. Kandula, D. Katabi, B. Davie, and A. Charny. Walking the tightrope: Responsive yet stable traffic engineering. In *Proc. of SIGCOMM*. ACM, 2005.

[6] K. Kompella and G. Swallow. RFC 4379: Detecting multi-protocol label switched (MPLS) data plane failures. 2006.

[7] H. Wang, H. Xie, L. Qiu, Y. Yang, Y. Zhang, and A. Greenberg. COPE: traffic engineering in dynamic networks. In *Proc. of SIGCOMM*. ACM, 2006.

[8] Y. Wang, H. Wang, A. Mahimkar, R. Alimi, Y. Zhang, L. Qiu, and Y. Yang. R3: Resilient routing reconfiguration. In *Proc. of SIGCOMM*. ACM, 2010.

[9] LSP Ping: MPLS LSP ping/traceroute for LDP/TE, and LSP ping for VCCV. http://www.cisco.com/en/US/ docs/ios/12_4t/12_4t11/ht_lspng.html.

[10] MPLS for dummies, 2010. http://www.nanog.org/ meetings/nanog49/presentations/Sunday/ mpls-nanog49.pdf.