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.
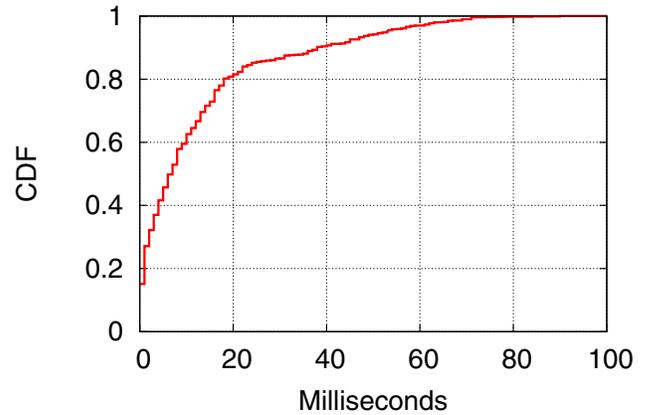

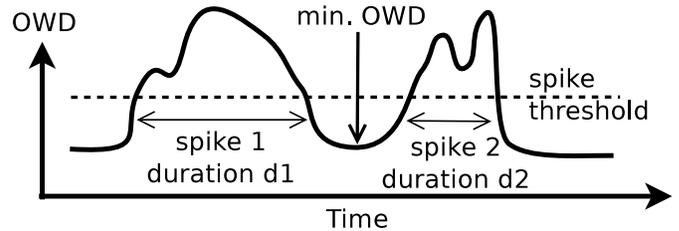**Figure 3: LSP latency difference (max-min) CDF**

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
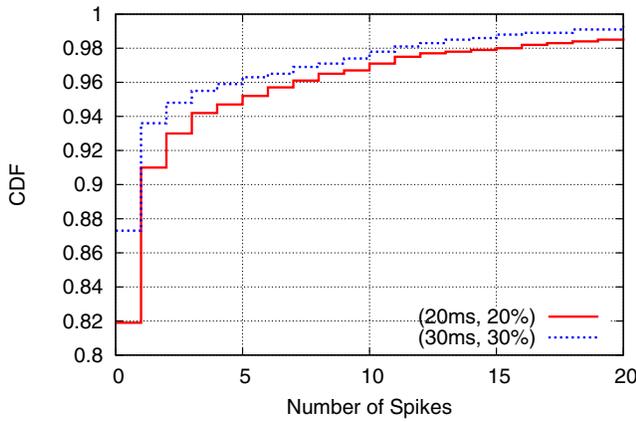
Figure 5: CDF of number of spikes observed by LSPs for different values of spikes
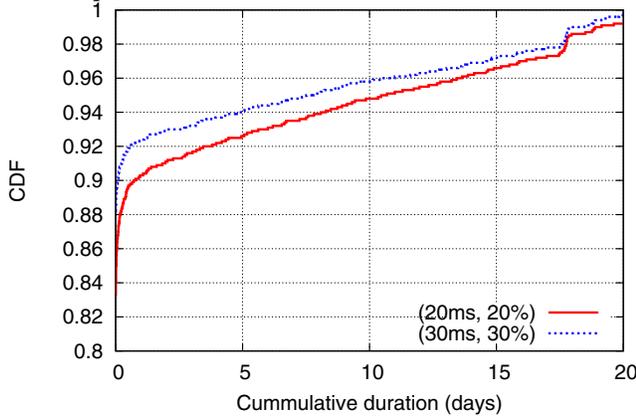

Figure 6: CDF of cumulative duration of spikes observed by LSPs for different values of spikes

Figure 7: Byte latency bar chart for LSP and LP

Figure 8: 99th 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
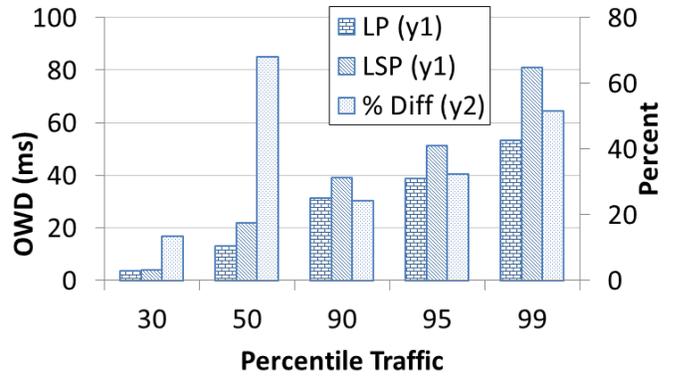
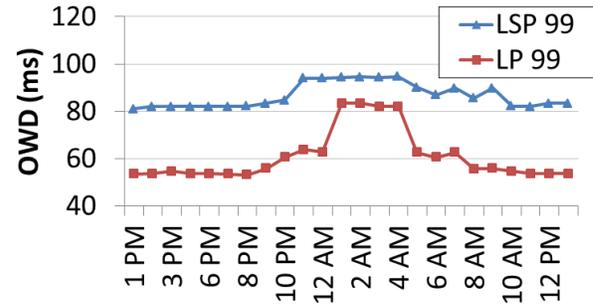impact on latency or reduce latency. For an LLPC, we attribute it to the old path rather than the new one because it is triggered by insufficient bandwidth on the former.

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.
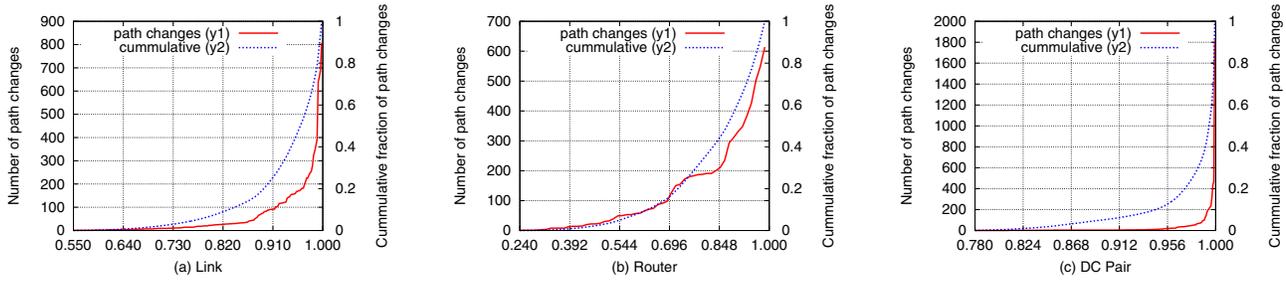
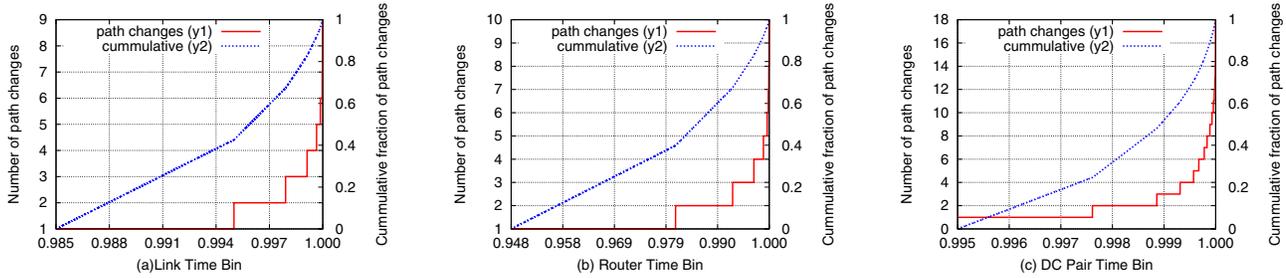**Figure 9: Histogram and cumulative path change per link, router and dcpair**

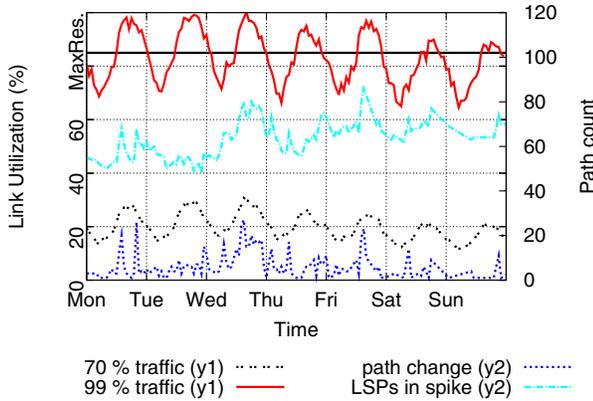**Figure 10: Histogram and cumulative path change per time bin per link, router and dcpair**

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

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,

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

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: RSVPâĂŤE: 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`.

Summary Review Documentation for

"Latency Inflation with MPLS-based Traffic Engineering"

Authors: A. Pathak, M. Zhang, Y. Hu, R. Mahajan, D. Maltz

Reviewer #1

Strengths: MPLS traffic engineering mechanisms are likely in wide use. Understanding how some of these mechanisms work in practice, and some of their pitfalls is likely to be of interest and value to the measurement community. Measurements are from a well-known online service provider (MSN).

Weaknesses: A key implication of the work is that MPLS TE deployments that use autobandwidth also experience the kinds of problems identified in the paper, though it is not clear that that is the case.

Comments to Authors: MPLS is used widely in the Internet these days, and it's nice to see a empirical study of the effectiveness (problems!) of some of traffic engineering mechanisms. Since many ISPs don't reveal much of anything about their MPLS networks, this study is likely to be useful and to spur additional work in this area. I appreciate the fact that MSN was willing to air a bit of dirty laundry, because some of the inter-DC latency numbers are pretty ugly (i.e., Fig 1).

There's an apparent mismatch in the caption of Figure 3 and the text describing it. In the caption, it refers to spike magnitudes (and spikes haven't yet been defined). In the text, it refers to the CDF of the latency difference for all LSPs. Which is it? It seemed to me that the description of latency spikes might be best described first, before any of the results in that section.

In the all-or-nothing discussion in \S 5, you might mention that some ISPs are already doing some manual splitting of LSPs. See, e.g.,http://www.nanog.org/meetings/nanog49/presentations/Sunday/mpls-nanog49.pdf

There are a number writing and presentation issues that should be addressed. The writing and presentation were, for me, the very weakest aspects of the paper. While these are all basically "minor" issues, addressing them would help greatly in making the paper more readable and useful. There are precious few references to related work. In particular, there are *zero* citations of any IETF RFCs that specify the behavior of MPLS, RSVP-TE, or any of the other protocols or mechanisms that you describe. RFC 3209 should at least be cited, and it should be mentioned that the autobandwidth algorithms are (as far as I understand) vendor-specific (i.e., not standardized), though Cisco and Juniper implement them similarly. Also, ref [6] should really be a reference to RFC 4379 rather than a ref to a proprietary Cisco document: MPLS Ping is a standard mechanism. It might also be helpful to have references to both Cisco and Juniper (perhaps others) documents that specify how to configure autobandwidth parameters.

There are a number of instances of the phrase "LSP Path", which is of course redundant. There are a number of cases in which it is stated that "the LSP reserves the required bandwidth" (\S 2.2), or an "[LSP] continually monitors the traffic rate" (\S 2.1), or some other, similar statement that assigns action to the LSP. An LSP is

simply the label switched path. It isn't a router or any other "active" agent. In a number of places, you could simply say that the label edge router, or LSR, or simply router, performs a specific action. In Section 5, LSPs are referred to as "fickle" (twice), which also is a bit of anthropomorphising.

Section 5 is really a general discussion section rather than specifically a discussion on autobandwidth parameters (that is, you discuss issues beyond those parameters). Might it be better to simply name this section "Discussion"?

Reviewer #2

Strengths: Latency between data centers is a critically important topic as is MPLS TE. The results are not surprising but are critically important for operators.

Weaknesses: The experimental methodology uses geographic information to approximate latency instead of active probing to infer the real latency.

Comments to Authors: Good short paper. This paper confirms what many including myself have expected that MPLS-TE is not optimal, quantifies the added delay, and attributes the root cause of delay to the hard-coded MPLS autobandwidth parameters. I believe this is an important result (read: we now have data that there really is a problem and it's degree) and I encourage the authors to follow through on their future plans to come up with a good solution.

My one big quibble with the paper was the experimental methodology. If I understand S3 correctly (it was a bit terse for a measurement paper), you did not do any active probing but rather summed the geographic distances between points in the MPLS tunnel -- is that correct? This methodology fails to account for delay inherent to store and forward devices or to the fact that fibers layouts have addition constraints (mountains, roads, private property, etc.) and are not always able to take the shortest path between two points. None of this contradicts your results, but in your followup work, I would recommend that you present some active probing data to better validate your claim that this is a reasonable approximation of latency (I found the one sentence disclaimer "We verified with a few LSPs that..." a bit of a red flag). Ok, so Microsoft doesn't use LP ping... just use standard ping from inside the data centers - it shouldn't be too hard to setup a vantage point in each data center.

Fig 9c: it seems like a very small number of DC-pairs are causing a lot of the flapping -- did you talk to the operators about this? If yes, what was the response? Is this simply that these nodes are more heavily loaded, under-provisioned, or because they are sub-optimally configured? Adding feedback from the operators could improve this paper.

Another point to look (likely in follow up work) at is the 5 minute timer that autobandwidth uses to recalculate routes: if that were decreased and the algorithms made more responsive, is there any additional benefit?

Reviewer #3

Strengths: The first look into MPLS-TE in practice. Nice dataset. Paper shows some problems with current TE implementations that should lead to further research.

Weaknesses: The measurements of latency are not direct and some of the analysis could be a little deeper.

Comments to Authors: This paper presents an interesting study of MPLS-TE. To the best of my knowledge this is the first study of a large-scale production network running MPLS-TE. I feel that I've learned more about MPLS-TE and the issues that can arise in practice. I just have a few comments.

From your introduction, it is clear that OSP operations have strict requirements for latency between data centers, could you give some numbers of the ranges of acceptable latencies? It would help people who want to find solutions to the problems you are pointing out.

When you present the network in Sec. 3, it would be nice to give an idea of the geographical spread. You only say that it is intercontinental in the conclusion.

At the last paragraph of Sec. 3, it would be nice to add a small explanation of how you verify that your latency estimation actually matches latency. It is a little disappointing that you can't get direct measurements of latency (why not just do pings from the data centers?), at least elaborating on the validation would help.

Why do label Figures 4, 7, and 8 with OWD, if earlier you've defined latency and that is what you use in the text?

People usually use the term stretch to refer to latency inflation.

Sec. 4.2 talks about correlation, but it uses no metric of statistical correlation. One can kind of see the correlation in the example week in Fig. 11, but the numbers would be more general.

Reviewer #4

Strengths: This is the first paper I know that demonstrates the impact of MPLS TE on latency across a network. The authors show a good understanding of how MPLS TE works and discuss possible reasons behind such a phenomenon.

Weaknesses: Unfortunately, the paper does not really prove that the reason behind the latency increase is the way autobandwidth works. The correlation with utilization is weak at best and there is no significant analysis performed to convince that this is the case - for that reason my assessment would be between a 2 and a 3 with a technical correctness at similar levels.

Comments to Authors: I find that this paper makes an attempt to look into a very interesting problem - that of the performance of traffic engineering mechanisms like MPLS TE. Providers have deployed such a solution in the hope that they will be able to better handle performance in their network, but as the authors mention very little is known as to how successful the employed

mechanisms actually are. Moreover, as the authors demonstrate these mechanisms are typically accompanied by a number of thresholds that network operators rarely change - and reasonably so, since nobody could really estimate the impact of those changes.

Interestingly, this paper shows that employing MPLS-TE can lead to significant increases in latency as compared to shortest paths. A large number of paths inside Microsoft's Online Service Network are showing spikes in latency, thus affecting the performance of the provided services.

The authors have studied the magnitude of latency increases, the number of data centers that they affect, as well as routers. They have further attempted to correlate it with link utilization with the goal to demonstrate that such latency increases may be due to the reaction of autobandwidth mechanism in high load conditions. Unfortunately, that point is never really proven. I would really have liked to see the authors formulating a hypothesis and testing it according to their data. Showing the result on Figure 11 is by no means a proof of that point. For each time bin they have identified the 70% and 99% percentile load across the network and then counted the number of path changes. They claim that the correlation is high but the presented figure is far from conclusive. I would like to have seen something more precise, where the authors track utilization across the network, the LSP messages that will be triggered for sure if autobandwidth kicks in, and then the precise correlation of that LSP change with the utilization increase in one of the links of the LSP. Otherwise, you have not really shown what you claim.

For that reason, i would say that this paper should only be accepted conditionally and published only if the authors manage to properly demonstrate such a correlation of remove that claim from the paper. You need to define the several causes that could lead to a path change (such as failure, etc) and then clearly test the hypothesis of link utilization being the culprit. Given the novelty of the data set and the rest of the analysis I would not say that this paper is a straight reject.

Finally, the authors compare MPLS TE latency with that obtained through an optimization problem that is never precisely defined. I think it is worth spending some space trying to make the paper self-consistent. What is byte weighted latency.

Reviewer #5

Strengths: This is the very first paper that I have read on the measured performance of MPLS networks. These results would be interesting to network operators.

Weaknesses: The paper simply reported the results in a straightforward way. Rather than just scratching the surface of the problem (hey look the paths are not optimal), I wish the paper had gone deeper to analyze the causes of why these sub-optimal paths.

Comments to Authors: This is an interesting paper, perhaps the first one, to report MPLS path quality in a real, large operational network. The specific measurement results would be interesting to people in operations and perhaps also in research (to understand why sub-optimal paths occurring from time to time).

However as someone who knows a bit about MPLS, the paper did not seem that exciting. I think a main reason is perhaps the result is a bit shallow. MPLS is simply a means to balance traffic flows across network, this measurement study shows that the decision

process (on the path selection and dynamic adjustment of MPLS circuits) does not pick the best decision all the time -- this is expected (I actually thought the performance was not bad:). To me the more interesting bit is not how bad the delay can be inflated, but to explain why MPLS picked sub-optimal paths from time to time.

Response from the Authors

We mention the two important points raised by the reviewers and how we addressed them in our paper.

1. The paper does not prove directly that autobandwidth is the root cause of latency spikes

MSN comprises tens of DCs interconnected with a dedicated network. All the traffic in MSN is carried over LSPs which are managed by autobandwidth algorithm.

- Most inter-dc links have three 9's availability (figure 9(c) in sigcomm'11 paper (Understanding Network Failures in Data Centers: Measurement, Analysis, and Implications)).

- Three 9's for 50 days is 1.2 hours. However, about 82.8% of LSPs which show severe spikes have cumulative spike durations more than 1.2 hrs. Also, about 98.9% of global cumulative LSP spike duration (sum of all severe LSP spikes across all LSPs) is originated from LSP spikes where individual spikes are more than 1.2 hrs.

This suggests that LSPs spikes which are severe are caused by autobw instead of failures. We confirmed similar LSP spikes in MPLS simulations.

2. Use of geographical distances to estimate latency instead of directly measuring it.

We currently cannot obtain the data for direct LSP latency measurement. The MSN operators do not provide LSP ping. It is also difficult to associate end-to-end ping data with an individual LSP due to load-balancing across multiple LSPs between the same DC pair. The latency estimate based on great-circle distance and speed-of-light is the best approximate we can get.

We have included the above details in the paper.