

## **Quality of Service using Traffic Engineering over MPLS: An Analysis**

Status of this Memo

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as ``work in progress.''

To learn the current status of any Internet-Draft, please check the ``1id-abstracts.txt' listing contained in the Internet- Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

### **NOTE**

This document is not to be taken as a finished product. Some of the sections are rough and are included in order to obtain comments from the community that will benefit future iterations of this document. This is simply a step in the ongoing conversation about this document. Finally, all the authors of this draft do not necessarily agree with and/or advocate all the mechanisms outlined in this document.

### **1. Abstract**

We compare the service received by TCP and UDP flows when they share either a link or an MPLS traffic trunk. Since traffic trunks allow non shortest path links also to be used, the total network throughput goes up with proper traffic engineering. If UDP and TCP flows are mixed in a trunk, TCP flows receive reduced service as the UDP flows increase their rates. Also, we found that in order to benefit from traffic engineering, MPLS trunks should be implemented end-to-end (first router to last router). If some part of the network is MPLS-unaware, the benefits are reduced or eliminated.

## **2. Introduction**

This draft presents an analysis of performance of TCP and UDP flows using MPLS traffic trunks. According to Li and Rekhter [1] and also in MPLS traffic engineering requirements documents [2], a traffic trunk is an aggregation of traffic flows of the same class which are placed inside a Label Switched Path (LSP). Traffic trunks are routable objects like virtual circuits in ATM and Frame Relay networks. These trunks can be established either statically or dynamically between any two nodes in an MPLS domain.

A trunk can carry any aggregate of micro-flows, where each micro-flow consists of packets belonging to a single TCP or UDP flow. In general, trunks are expected to carry several such micro-flows of different transport types. However, as shown in this analysis, mixing different transport types can cause performance problems such as starvation and unfairness for certain traffic.

In this draft, we analyze the performance impact of mixing TCP and UDP traffic. The two transports have very different congestion response. TCP has congestion control and so it reduces its traffic in response to packet loss. UDP has no congestion control and does not respond to losses. While it is possible to design UDP applications that are sensitive to congestion losses, very few such applications exist.

## **3. Network Topology:**

We used network simulator version 2 (ns-2) [3] for our analysis with new modules for label switching and trunk queueing.

In the simulations, the network topology shown in Figure 1 was used. It consists of six routers and six end-systems. There are 3 flows. Source S1 sends UDP traffic to destination D1. Sources S2 and S3 send TCP traffic to destination D2 and D3, respectively. The UDP source sends traffic at a given rate. The TCP sources are "infinite ftp" sources and send packets whenever its congestion window allows.

The two TCP flows are different only in their packet sizes. The first TCP flow (TCP1) between S2 and D2 uses an MSS of 512 bytes. The second flow (TCP2) between S3 and D3 uses an MSS of 1024 bytes. Thus, the two flows are almost identical except for the packet sizes.

There exist two parallel paths between routers R2 and R5, one of a high bandwidth (45 Mbps) and another of comparatively low bandwidth (15 Mbps).



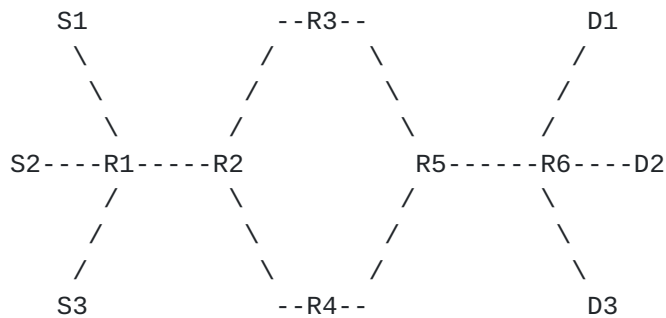


Figure 1: Network Topology

#### 4. Simulation Results

We analyzed four different scenarios with different intermixing of TCP and UDP flows. These scenarios and the analysis results are as follows.

##### Case 1: No Trunks

The first case we analyzed is current IP with best effort shortest path routing without any trunks. All data in this case follows the high-speed path R2-R3-R5 and the alternative path R2-R4-R5 is unused. We vary the UDP rate and measure the throughput of the three flows. The results are shown in Table 1.

Table 1: Results using normal IP routing

UDP Source rate (Mbps)	Throughput (Mbps)
	UDP      TCP1      TCP2
16.8	15.90      2.66      18.56
18.7	17.80      2.76      16.52
21.0	19.86      2.15      16.43
24.0	22.63      1.99      14.59
28.0	26.19      1.35      12.48

Notice that as the UDP flow increases its rate, the throughputs of both TCP flows decrease. That is, the congestion-sensitive traffic suffers at the hands of congestion-insensitive traffic.

Notice also that the two TCP flows that are different only in their packet sizes get very different throughput. The flow with smaller packet size gets very small throughput. This is a known behaviour of TCP



congestion mechanism [3].

#### Case 2: Two Trunks with TCP and UDP mixed on one Trunk

In this case we have configured two trunks over the network in the following way. The first trunk carries UDP and TCP1 and follows the label switched path (LSP) R1-R2-R4-R5-R6 (high bandwidth path). The second trunk carries TCP2 and follows the path R1-R2-R3-R5-R6 (low bandwidth path). We vary the UDP source rate and observe the throughput for the various flows. The result are shown in Table 2.

Table 2: Results using two trunks, with TCP and UDP mixed on one Trunk

UDP Source rate (Mbps)	Throughput (Mbps)
	UDP TCP1 TCP2
16.8	16.49 18.91 12.97
18.7	18.32 17.46 12.98
21.0	20.61 16.17 12.94
24.0	23.52 15.00 12.33
28.0	27.45 4.54 12.97

Notice that there is a significant improvement in the throughput of the flows. This is because the low bandwidth link which was not being utilised till now is also being used. Hence, TCP2 which is directed on a separate trunk and follows the low bandwidth link is unaffected by the increase in the UDP source rate and hence shows an almost constant throughput. But, we also observe that as the UDP source rate increases, the throughput of the TCP1 flow mixed with it on the same trunk suffers. This is because the TCP flow cuts down its rate in response to network congestion. Thus, we can say that use of separate trunks definitely improves the network performance but, is not sufficient to provide the quality of service which might be desirable for some applications. Thus, in order to guarantee the quality of service to a given flow, we need to isolate the various flows in different trunks so that even if a given flow increases its source rate, other flows are unaffected by this.

#### Case 3: Three Trunks with isolated TCP and UDP flows

In this case, we use three isolated trunks, one each for the UDP and TCP flows, respectively. This is done using separate queues for all the different trunks at the routers R1 and R2. By doing this, we essentially separate the UDP flow from the TCP flows and thus get the results shown



in Table 3.

Table 3: Results using a separate trunk for each flow

UDP Source rate (Mbps)	Throughput (Mbps)
	UDP TCP1 TCP2
16.8	15.64 21.27 12.10
18.7	16.33 21.21 12.09
21.0	17.34 21.08 12.08
24.0	18.13 21.02 12.22
28.0	19.06 21.23 12.06

Here we see that the increase in the UDP source rate does not affect either of the TCP sources and both are able to get a fairly constant throughput. Thus, by isolating the various flows, we can guarantee a given quality of service to sources which are responsive to congestion control also. Although this is achieved at the overhead of maintaining separate queues for each of the trunks at each of the routers.

#### Case 4: Non-end-to-end Trunks

In this case we analyse a scenario where the trunks are not end to end. Specifically, the trunks are being initialized onlt at R2. In this case, the flows interfere with each other for part of the path since R1 does not distinguish between the various flows. The results are shown in Table 4.

Table 4: Results using Non-end-to-end trunks

UDP Source rate (Mbps)	Throughput (Mbps)
	UDP TCP1 TCP2
16.8	16.33 18.28 11.88
18.7	18.27 17.41 13.03
21.0	20.60 4.87 13.09
24.0	18.27 4.86 13.12
28.0	26.89 10.90 11.11





The above results show that the network really cannot guarantee anything in such a case. This happens because at R1 the flows are treated identically and during periods of congestion the TCP sources reduce their source rates and this leads to very poor throughput for the TCP sources even though they are treated distinctly at R2. Thus, in order that this kind of a scheme be successful, the flows should be separated as soon as they share a common link in the network.

## **5. Summary**

The above simulations clearly indicate the effectiveness of traffic trunks in providing different qualities of service and the related issues. We can infer that the use of traffic trunks would be effective only if they are end-to-end and should be able to isolate congestion insensitive traffic from congestion sensitive traffic.

## **6. References**

- [1] Li, T., Rekhter, Y. "A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)". [RFC 2430](#). October 1998.
- [2] Awduche, D.O., Malcolm, J., O'Dell, M., McManus, J. "Requirements for Traffic Engineering Over MPLS". [draft-ietf-mpls-traffic-eng-00.txt](#).
- [3] UCB, LBNL, VINT Network Simulator - ns  
<http://www-mash.cs.berkeley.edu/ns/ns.html>
- [4] Rosen, E.C., Vishwanathan, A., Callon, R. "Multiprotocol Label Switching Architecture". [draft-ietf-mpls-arch-02.txt](#).

## **7. Contact Addresses**

Praveen Bhaniramka, Wei Sun and Raj Jain  
Department of Computer and Information Science  
The Ohio State University  
**2015 Neil Ave, DL395**  
Columbus, OH 43210  
Phone: +1 614-292-3989  
Email: [bhaniram@cis.ohio-state.edu](mailto:bhaniram@cis.ohio-state.edu)  
Email: [wsun@cis.ohio-state.edu](mailto:wsun@cis.ohio-state.edu)  
Email: [jain@cis.ohio-state.edu](mailto:jain@cis.ohio-state.edu)

