

An Automated Traffic Engineering Algorithm for MPLS-Diffserv Domain

Junaid Ahmed Zubairi
Department of Mathematics and Computer Science
SUNY at Fredonia, Fredonia NY 14063, USA
junaid.zubairi@fredonia.edu

Keywords: Traffic Engineering, Diffserv, MPLS, QoS

Abstract

Traffic Engineering seeks to optimize the handling of the traffic in a network such that the capacity in various parts is utilized efficiently. IETF has developed new protocols and techniques to meet the emerging traffic requirements and to manage and control the network and its services. With MPLS, Diffserv, RSVP-TE/CR-LDP, SLA and COPS in place, manual provisioning of the network is impossible and it is expected that several algorithms will be developed to automate traffic engineering. In this paper, an efficient dynamic link-coloring algorithm is proposed to engineer QoS paths within a Diffserv aware MPLS domain. This algorithm applies a set of rules across the domain to allocate LSP's to traffic trunks based on the Diffserv classes of service and dynamic link metrics. Initial results of applying the algorithm to meet the demands of traffic sets consisting of several LSP requests with varying classes of service are presented.

1 INTRODUCTION

The transformation of Internet into a worldwide information network has caused the introduction of different Internet enabled applications. Some of the applications such as file-transfer and e-mail expect reliability in communication. In addition, an increasing number of applications demand timeliness in delivery of data. For example, the e-mail application can wait for a random amount of time for delivery of messages. However, a telemedicine application or a database update transaction must be finished within a bounded time period. The IP-based network makes its best effort to deliver the data in a reliable and timely way. However, if the data is delayed or discarded, the network cannot alleviate this problem and the upper layers have to take the corrective action.

In order to meet the service requirements of the modern applications on the Internet, new techniques and protocols are being developed. These protocols are expected to cater to the time-sensitive applications while keeping the network fair and efficient for all. IETF (Internet Engineering Task Force) has been working on developing new protocols and techniques for supporting time-sensitive services in the

Internet. Among the new protocols, RSVP (Resource Reservation Protocol) [1,2,3,4] provides quantitative guarantees to flows by reservations whereas Diffserv (Differentiated Services) [5,6,7,8] provides qualitative assurances by using appropriate behaviors for packets marked with Diffserv codepoints in the IP headers. MPLS (Multi Protocol Label Switching) has been developed to accelerate routing of traffic aggregates destined towards a common point by using LSP's so that the intermediate routers do not have to make routing decisions and they simply forward the traffic based on the interface and the value of the label.

One of the most important capabilities of MPLS is to map traffic on paths established with traffic engineering principles, resulting in load balancing and fault tolerance of the underlying network. Traffic Engineering deals with performance optimization and evaluation so that the overall performance for a network can be enhanced [9]. Networks running various modern protocols cannot be managed manually and automated traffic engineering algorithms are required [10]. In this paper, an algorithm TELIC (Traffic Engineering with Link Coloring) is presented. TELIC seeks to automate the traffic engineering process offering appropriate services for various classes of traffic. It determines LSP's for traffic trunks based on dynamic link coloring. Rest of the paper is organized in 4 sections. In the next section, traffic engineering principles are discussed. In section 3, MPLS and constrained routing is highlighted. TELIC is outlined in section 4 and section 5 gives the implementation details and initial results of TELIC with several traffic sets.

2 TRAFFIC ENGINEERING

Mapping traffic flows onto the physical topology to enhance overall network utilization and create a uniform distribution of traffic is referred to as traffic engineering [9,11,12,13]. Traffic engineering optimizes network efficiency through the control of the mapping and distribution of traffic over the network to assure satisfactory service delivery, maximize resource efficiency, and to avoid or relieve congestion on any single path. Other aspects of traffic engineering include methods that control a network's response to traffic demands and other stimuli, such as link or node failure [14].

Initially, the mapping of traffic to links was a by-product of routing configuration and it followed the shortest path calculated by the Interior Gateway Protocol (IGP). Congestion was a result of lack of network resources due to overloading in routers and links and uneven distribution of traffic [12,15,16]. The problem of lack of network resources can be resolved by providing more resources. Uneven distribution of traffic is more complicated since it can be the product of the routing protocols such as OSPF and IS-IS, that select the shortest paths to forward packets. While using shortest path conserves network resources, it may cause some other problems, such as:

- ◆ When traffic from a source to a destination exceeds the capacity of the shortest path, the shortest path will become congested. However, a longer path between source and destination nodes will remain under-utilized.
- ◆ In case of multiple shortest paths from different sources converging on a preferential link, the total traffic from different sources may exceed the capacity of the chosen link [11, 15].

Traffic engineering for improving network utilization includes congestion avoidance, utilization improvement, fairness, reliability and QoS support [12,14,15,16]. TE (Traffic Engineering) also evaluates network performance and compliance to TE rules. Results from the evaluation can be used to improve the network topology and structure [9].

3 MPLS

MPLS (Multi-Protocol Label Switching) is based on identifying an end to end path before starting to transmit the data . It combines the L3 routing and L2 switching into "L2.5 forwarding" and provides a way to define connections in a connectionless network. MPLS works on the principle of providing a "virtual path" from ingress to egress router in an MPLS domain. The same idea has been used in the Internet to provide VPN tunnels across the public network and IPv6 tunnels across IPv4 networks. In tunneling, the packets originating at a router and destined for a specific node are labeled in such a way that the intermediate routers forward them towards a common destination through the same path. Thus, tunneling implicitly introduces the notion of a connection because all packets in the tunnel follow the same path and experience the same routing through the network. In MPLS, the connection between the source and destination router can be defined in a variety of ways. It can be defined very specifically, as in ER-LSP (Explicitly routed label switched path) or it can be loosely defined as any path between the source and destination. In case of ER-LSP, it may be called an LSP tunnel [17].

When a path creation request arrives, the MPLS performs constrained routing to find a suitable path. Constrained routing searches for a suitable path by applying the extended IGP parameters reported in link state advertisements to the overall tree as per the QoS and other policies. Link state advertisements may include reservable bandwidth at different priority levels; static link colors indicating capacities of links and TE-specific metrics. Path selection can follow a narrowing down of available choices by using Boolean operators. Once this path is selected, an LSP can be established by sending a setup message with CR-LDP (Constrained routing Label Distribution Protocol) or RSVP-TE (Traffic Engineering extensions to RSVP) that can pin down the path through the routers. Each LSP has a number of parameters that include its resilience, fault tolerance, preemptivity and associated QoS features.

The main advantage of MPLS besides its capability of providing L2 functionality within routers, is that traffic engineering is implemented using explicitly routed paths. The LSP's are created independently specifying different paths that are based on management-defined policies. Constrained routing avoids congestion and uneven network utilization by optimized arrangement of traffic flows through the network. Routes that are subject to constraints such as bandwidth, delay, jitters, and administrative policy are computed considering the dynamic traffic load conditions in addition to the common metrics [11, 12, 15, 16]. Thus a longer but lightly loaded path may be preferred over the heavily loaded shortest path. Given the QoS request of a flow or an aggregation of flows, QoS-routing can return the route that is most likely able to meet the QoS requirements.

4 TELIC: DYNAMIC LINK COLORING ALGORITHM FOR TRAFFIC ENGINEERED PATHS

In this section, TELIC (Traffic Engineering with Link Coloring) algorithm is introduced. As explained earlier, constrained routing works with metrics reported by extended IS-IS and OSPF. Metrics can be divided into following categories:

- (1) Additive such as cost or delay
- (2) Multiplicative such as reliability
- (3) Concave such as bandwidth

Computing optimal routes subject to two or more constraints is a NP-complete problem. Mostly, the algorithms work on "bandwidth available" and "hop count" for selecting a path between a source and destination. A constrained routing scheme can choose one of the followings as the route for a

destination (with a tradeoff between resource conservation and load balancing):

- (1) Shortest path, if multiple found select the widest one (the one with most available BW) (*shortest-widest*)
- (2) Widest path, if multiple found, select shortest one (*widest-shortest*)
- (3) The *shortest-distance* path. Here the bandwidth is replaced by its inverse value that makes it possible to express the total distance of a k-hop path p as the sum of this inverse value:

$$\text{dist}(p) = \sum_{i=1 \rightarrow k} (1/n_i)$$
 where n_i is the bandwidth of the link i and $i=1 \rightarrow k$

The shortest-distance approach favors shortest paths when network load is heavy and favors widest paths when network load is moderate. However, this scheme does not differentiate between various classes of traffic as its only measure of the cost is the available bandwidth.

In the proposed TELIC algorithm, an efficient approach is adopted for selecting paths based on their service classes in order to map the traffic flows onto an MPLS domain. TELIC processes a set of LSP requests received at the ingress of an MPLS-Diffserv domain. Various modules of TELIC are shown in Figure 2. Each LSP request specifies the amount of bandwidth requested and the class of service (EF,AF,DF). Based on the information in the request, the algorithm tries to locate the LSP that best meets the request using a subgraph of the domain. Once an LSP is determined, it is registered in the master LSP table in the ingress node and the status of the links on this LSP is updated. If an LSP cannot be established for an EF request, a lower priority DF LSP currently active is de-registered and the links associated to such an LSP are de-allocated in order to allow the EF request to be completed. The de-registered LSP may enter the request queue for an attempt to get network resources after the current EF request has been met.

When TELIC is implemented in a Diffserv-aware MPLS domain, it can achieve the following targets:

- ◆ Balance the load across the domain
- ◆ The congested or heavily utilized links are avoided, specially for QoS flows
- ◆ In Diffserv coded flows, the EF traffic is allocated LSP's that avoid links being used by other traffic thus minimizing the chances of EF behavior's failure.
- ◆ Installed paths can be relocated if the requirements change over time
- ◆ Every flow gets its fair share of the network resources. If the network is heavily overloaded, best effort traffic will pass through most congested links, EF traffic will

```

Find shortest path( )
BEGIN
Given G is a directed graph with nodes I
and E defined as source and destination

Find the shortest path from I to E in G
using Dijkstra's shortest path algorithm

If path found report success; else report
failure
END

Register LSP( )
BEGIN
Given selected path with a list of
member nodes and links
◆ Allocate requested LSP
◆ Update the link costs and re-assign
colors on this path
◆ Find the lowest level color in this
path (link colors are listed highest to
lowest in this order: silver, white,
green, yellow, red) and designate it
as the color of the new LSP
◆ Register this LSP in the master LSP
table in chronological order with all
links, nodes , service class and LSP
color
END

Check_and_remove( )
BEGIN
◆ Consult the master LSP table;
locate the oldest DF LSP
◆ Remove the DF LSP from LSP
table to LSP request queue
◆ Update all link costs and nodes
involved in this LSP and change
link colors if necessary
◆ If no DF LSP found, flag=false
else flag=true
END

```

Figure 1: Some Modules of TELIC

- ◆ pass through least congested links and AF traffic will pass through moderately congested links.
- ◆ In a heavily congested network, request for LSP establishment for an EF flow may be turned down instead of compromising on its quality because the

congested network cannot meet the EF service requirements.

Let us consider an MPLS domain to illustrate the application of TELIC. It is assumed that there is one ingress node of this domain. All the traffic that enters via this node flows towards a common egress node using multiple paths available inside the domain. Each path is composed of links between interior nodes that lead from the ingress to the egress node. Every link has a cost associated that takes into account the available bandwidth, delay and the reliability of the link. Based on the granularity of bandwidth requests, the available bandwidth can be expressed as a fraction of the total bandwidth.

Initially, link colors are assigned based on the costs computed for all links in the domain using the cost function. During the normal operation, whenever an update is received, the available bandwidth is used to update the link colors. The values of the cost metrics are as follows with the reservable bandwidth range for each color indicated in parenthesis:

- Silver (100-SL): 1-1.7
- White (SL-10): 1.7-10
- Green (100-GL): 2-2.8
- Yellow (GL-YL): 2.8-5
- Red (YL-10): 5-20

The reservable bandwidth limits are set empirically to SL (Silver Limit), GL (Green Limit) and YL (Yellow Limit). Initially all links will be either carrying silver or green color. However, links of all colors may be present in the domain after the network has been in operation for some time. Traffic trunks are defined as aggregates of traffic that need to get the same treatment in this domain. Given a set of traffic trunks and a set of links, the problem of identifying TE-compliant LSP's reduces to splitting the domain into a number of subgraphs and identifying the shortest path in the appropriate subgraph. In case the egress is not reachable in a particular subgraph, this subgraph is merged with a higher cost subgraph. For example, if no path is found from the ingress to egress node using silver subgraph, the white subgraph is merged into it and path search begins again. In case of EF trunks, we can report failure if no path is found even on merging green subgraph. The search failure can also be viewed on the scale of a node in the domain. As no outgoing link of desired color is found, the outgoing links of other colors are selected in descending order of priority. Once an LSP is established, the lowest color link in this LSP is reported back to the originator node and recorded in the LSP table in the ingress router.

Figure 2 shows an MPLS domain having single source-destination pair ISP topology [18] with 9 intermediate routers, two edge routers and 16 links that can be used between the ingress and the egress node. Initially, two paths are marked silver for QoS sensitive traffic (S and S') and the remaining links are all marked green.

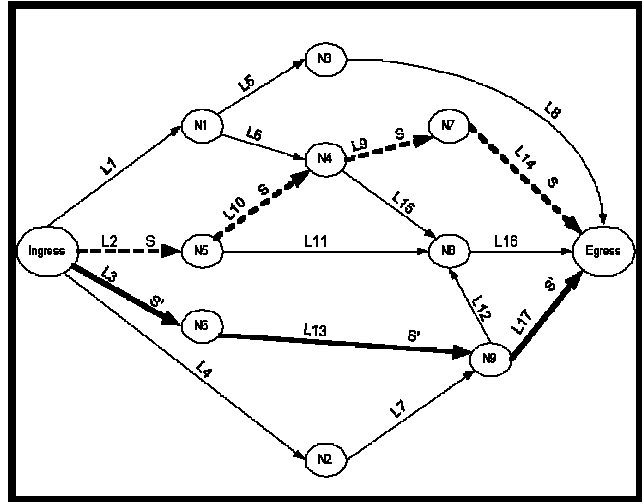


Figure 2: An MPLS Domain Showing Various Links and Nodes

If a path is to be established from the ingress node to the egress node through a given domain or a flow is to be aggregated into an already established LSP, its requirements are provided to TELIC. For link selection, TELIC follows some built-in rules to provision separate paths to EF coded traffic and other traffic, thus minimizing the mixing of traffic that could jeopardize EF performance.

As an LSP is installed, the color of the links is changed to prepare the domain for new path computations. Following rules are enforced:

- (1) As the path requests are received, each path request is taken in a queue. QoS path requests may be handled first if network operator maintains a CBQ approach.
- (2) A non-QoS flow should not be assigned silver links unless there are no other alternatives
- (3) In case a non-QoS flow is assigned silver links and a QoS flow arrives, the non-QoS flow is to be re-routed and the QoS flow is admitted on silver links
- (4) The color of the whole path for an allocated flow is checked and updated as the paths are installed and terminated. The rule for coloring a path is to use the lowest color. For example, if there is at least one red color link found, the flow state table will be marked as red.

- (5) When a path is terminated, the link states within the domain are upgraded. TE rules are activated to make sure that all QoS flows are moved to silver links and green links

5 IMPLEMENTATION AND RESULTS

TELIC is implemented in C++. Traffic request set is entered in a file and the file name is specified as a command line argument. The program allows the user to display the links and their associated costs after the allocation. The Master LSP table and remaining bandwidth on all links can also be printed.

We model the traffic trunks as requests for setting up LSP's through the given MPLS domain. Currently we use a static traffic set and assume an 80-20 split in the type of traffic, with 20 percent of the requested bandwidth being in the EF class and 80 percent of the requested bandwidth being in the AF and DF classes. For the later part, the ratio of AF-DF split is varied in order to account for all possible variations in the request types. The bandwidth allocation trend is plotted in Figure 3,4 and 5 for EF, AF and DF classes of requests. It is seen that the EF class of traffic has 100 percent success rate and all its bandwidth requests are met. Since the EF class is the premier class of traffic, this trend is highly desirable.

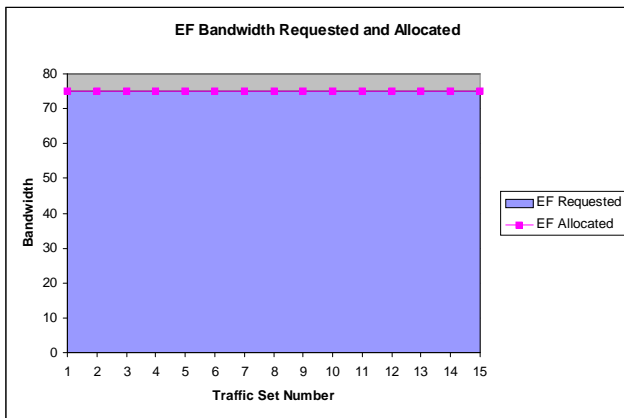


Figure 3: EF Class BW Request and Allocation Trend with TELIC

For the AF class, the bandwidth requested in slowly increased from the first set to the last set. It is obvious that the algorithm works in a very predictable way by meeting the total BW request of AF class until a threshold point. From this point onwards, the AF class allocation stays fixed and the excess requests are denied. For the DF class, the graph has two sections. First TELIC maintains a constant gap between request and allocation. The second part is when the allocated bandwidth reduces to zero while DF demand is still present. The best effort traffic

is not allocated its full requested bandwidth at any time in this network due to the presence of some bottleneck links. The gap between request and allocation is noticeable for DF traffic because of the specific topology of the network. The network, as shown in Figure 2, has two bottleneck links. The first bottleneck link is between node 8 and egress node. This link completes the path to egress for several paths, resulting in an overloaded condition. The second bottleneck link is from ingress to node 1. Both of these links are converted to red quickly, resulting in denial of bandwidth to pending requests. Thus, monitoring the performance of the network against various requests exposes the topology problems. Correcting these problems can lead to an improvement in the performance.

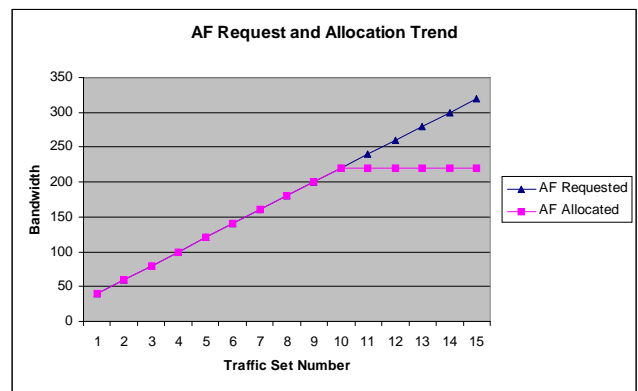


Figure 4: AF Class BW Request and Allocation Trend with TELIC

6 CONCLUSION AND FUTURE WORK

An automated traffic engineering algorithm has been proposed. The results of applying this algorithm on the ingress node of an MPLS-Diffserv domain are presented. The algorithm has been implemented in C++ and tested with various traffic sets with varying mix of traffic classes. Future work includes modeling the request arrival and LSP holding time with different probability distributions and introducing fault tolerance and rerouting mechanisms. Also TELIC's performance will be evaluated specifically against the least distance constrained routing. Enhancements proposed for this algorithm include fault tolerance and prevention of excessive delays in best effort traffic.

ACKNOWLEDGMENTS

This work has been supported by a grant through the scholarly incentive award program of SUNY at Fredonia. Jason Beuckman, a senior student in computer science, has implemented TELIC in C++.

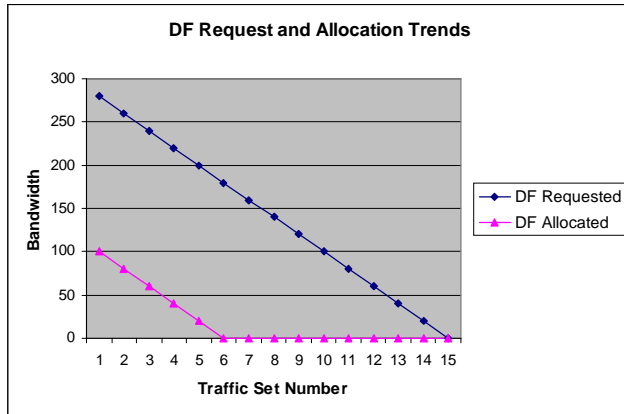


Figure 5: DF Class BW Request and Allocation Trend with TELIC

REFERENCES

- [1] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: a new resource ReSerVation protocol," *IEEE Network*, vol. 7, pp. 8-18, Sept. 1993.
- [2] B. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation protocol (RSVP) - version 1 functional specification" *RFC 2205*, IETF, Oct. 1997.
- [3] J. Wroclawski, "The use of RSVP with IETF integrated service" *RFC 2210*, IETF, Oct. 1997.
- [4] P. Pan and H. Schulzrinne, "Staged refresh timers for RSVP" in *Proceedings of Global Internet*, (Phoenix, Arizona), Nov. 1997. also IBM Re-search Technical Report TC20966.
- [5] K. Nichols et. al. "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers" *RFC 2474*, IETF, Dec. 1998,
- [6] S. Blake et. al. "An Architecture for Differentiated Services" *RFC 2475*, IETF, Dec. 1998,
- [7] V. Jacobson et. al. "An Expedited Forwarding PHB" *RFC 2598*, IETF, June 1999,
- [8] J. Heinanen et. al. "Assured Forwarding PHB Group" *RFC 2597*, IETF, June 1999
- [9] D. Awduche et. al. "A Framework for Internet Traffic Engineering", *Internet Draft IETF*, , work in progress, Jul 2000 < draft-ietf-tewg-framework-02>
- [10] R. Callon, "Predictions for the Core of the Network", *IEEE Internet Computing* volume 4, number 1, Jan/Feb 2000
- [11] C. Semeria "Traffic Engineering for the New Public Network" , *White Paper*, Juniper Network, 2000
- [12] X. Xiao and L. Ni "Traffic Engineering with MPLS in the Internet", *Michigan State University*, 2000
- [13] D. Awduche et. al. "Requirements for Traffic Engineering Over MPLS", *RFC 2702*, IETF, Sep 1999
- [14] G. Ash "Traffic Engineering & QoS Methods for IP, ATM, & TDM-Based Multi-service Networks", , *Internet draft IETF*, work in progress, < draft-ash-te-qos-routing-01>, Jul 2000
- [15] X. Xiao and L. Ni "Internet QoS: A Big Picture", *IEEE Network* March/April 1999, pp8-18
- [16] X. Xiao "Providing Quality Of Service In The Internet", *Ph. D. thesis*, Michigan State University, 2000
- [17] R. Callon et. al "A Framework for Multiprotocol Label Switching", *Internet Draft IETF*, work in progress, , <draft-ietf-mpls-framework-05>, Sept. 1999
- [18] Q. Ma and P. Steenkiste, "On Path Selection for Traffic with Bandwidth Guarantees", in *Proc. IEEE Intl' Conf. On Network Protocols*, Oct 1997

Biography

Junaid Ahmed Zubairi received his BE (Electrical Engineering) from NED University of Engineering, Pakistan and MS and Ph.D. (Computer Engineering) from Syracuse University, USA. He worked in Sir Syed University Pakistan and Intl' Islamic University Malaysia before joining SUNY at Fredonia where currently he is a faculty member in the Department of Mathematics and Computer Science. His research interests include traffic engineering, performance evaluation of networks and fault analysis of VLSI circuits. He can be reached at junaid.zubairi@fredonia.edu.