Network Working Group
Request for Comments: 2702
Category: Informational

D. Awduche
J. Malcolm
J. Agogbua
M. O'Dell
J. McManus
UUNET (MCI Worldcom)

September 1999

Requirements for Traffic Engineering Over MPLS

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Abstract

This document presents a set of requirements for Traffic Engineering over Multiprotocol Label Switching (MPLS). It identifies the functional capabilities required to implement policies that facilitate efficient and reliable network operations in an MPLS domain. These capabilities can be used to optimize the utilization of network resources and to enhance traffic oriented performance characteristics.

Table of Contents

1.0 Introduction

Multiprotocol Label Switching (MPLS) [1,2] integrates a label swapping framework with network layer routing. The basic idea involves assigning short fixed length labels to packets at the ingress to an MPLS cloud (based on the concept of forwarding equivalence classes [1,2]). Throughout the interior of the MPLS domain, the labels attached to packets are used to make forwarding decisions (usually without recourse to the original packet headers).

A set of powerful constructs to address many critical issues in the emerging differentiated services Internet can be devised from this relatively simple paradigm. One of the most significant initial applications of MPLS will be in Traffic Engineering. The importance of this application is already well-recognized (see [1,2,3]).

This manuscript is exclusively focused on the Traffic Engineering applications of MPLS. Specifically, the goal of this document is to highlight the issues and requirements for Traffic Engineering in a large Internet backbone. The expectation is that the MPLS specifications, or implementations derived therefrom, will address

the realization of these objectives. A description of the basic capabilities and functionality required of an MPLS implementation to accommodate the requirements is also presented.

It should be noted that even though the focus is on Internet backbones, the capabilities described in this document are equally applicable to Traffic Engineering in enterprise networks. In general, the capabilities can be applied to any label switched network under a single technical administration in which at least two paths exist between two nodes.

Some recent manuscripts have focused on the considerations pertaining to Traffic Engineering and Traffic management under MPLS, most notably the works of Li and Rekhter [3], and others. In [3], an architecture is proposed which employs MPLS and RSVP to provide scalable differentiated services and Traffic Engineering in the Internet. The present manuscript complements the aforementioned and similar efforts. It reflects the authors' operational experience in managing a large Internet backbone.

1.1 Terminology

The reader is assumed to be familiar with the MPLS terminology as defined in [1].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [11].

1.2 Document Organization

The remainder of this document is organized as follows: Section 2 discusses the basic functions of Traffic Engineering in the Internet. Section 3, provides an overview of the traffic Engineering potentials of MPLS. Sections 1 to 3 are essentially background material. Section 4 presents an overview of the fundamental requirements for Traffic Engineering over MPLS. Section 5 describes the desirable attributes and characteristics of traffic trunks which are pertinent to Traffic Engineering. Section 6 presents a set of attributes which can be associated with resources to constrain the routability of traffic trunks and LSPs through them. Section 7 advocates the introduction of a "constraint-based routing" framework in MPLS domains. Finally, Section 8 contains concluding remarks.

2.0 Traffic Engineering

This section describes the basic functions of Traffic Engineering in an Autonomous System in the contemporary Internet. The limitations of current IGPs with respect to traffic and resource control are highlighted. This section serves as motivation for the requirements on MPLS.

Traffic Engineering (TE) is concerned with performance optimization of operational networks. In general, it encompasses the application of technology and scientific principles to the measurement, modeling, characterization, and control of Internet traffic, and the application of such knowledge and techniques to achieve specific performance objectives. The aspects of Traffic Engineering that are of interest concerning MPLS are measurement and control.

A major goal of Internet Traffic Engineering is to facilitate efficient and reliable network operations while simultaneously optimizing network resource utilization and traffic performance. Traffic Engineering has become an indispensable function in many large Autonomous Systems because of the high cost of network assets and the commercial and competitive nature of the Internet. These factors emphasize the need for maximal operational efficiency.

2.1 Traffic Engineering Performance Objectives

The key performance objectives associated with traffic engineering can be classified as being either:

1. traffic oriented or

2. resource oriented.

Traffic oriented performance objectives include the aspects that enhance the QoS of traffic streams. In a single class, best effort Internet service model, the key traffic oriented performance objectives include: minimization of packet loss, minimization of delay, maximization of throughput, and enforcement of service level agreements. Under a single class best effort Internet service model, minimization of packet loss is one of the most important traffic oriented performance objectives. Statistically bounded traffic oriented performance objectives (such as peak to peak packet delay variation, loss ratio, and maximum packet transfer delay) might become useful in the forthcoming differentiated services Internet.

Resource oriented performance objectives include the aspects pertaining to the optimization of resource utilization. Efficient management of network resources is the vehicle for the attainment of

resource oriented performance objectives. In particular, it is generally desirable to ensure that subsets of network resources do not become over utilized and congested while other subsets along alternate feasible paths remain underutilized. Bandwidth is a crucial resource in contemporary networks. Therefore, a central function of Traffic Engineering is to efficiently manage bandwidth resources.

Minimizing congestion is a primary traffic and resource oriented performance objective. The interest here is on congestion problems that are prolonged rather than on transient congestion resulting from instantaneous bursts. Congestion typically manifests under two scenarios:

1. When network resources are insufficient or inadequate to accommodate offered load.

2. When traffic streams are inefficiently mapped onto available resources; causing subsets of network resources to become over-utilized while others remain underutilized.

The first type of congestion problem can be addressed by either: (i) expansion of capacity, or (ii) application of classical congestion control techniques, or (iii) both. Classical congestion control techniques attempt to regulate the demand so that the traffic fits onto available resources. Classical techniques for congestion control include: rate limiting, window flow control, router queue management, schedule-based control, and others; (see [8] and the references therein).

The second type of congestion problems, namely those resulting from inefficient resource allocation, can usually be addressed through Traffic Engineering.

In general, congestion resulting from inefficient resource allocation can be reduced by adopting load balancing policies. The objective of such strategies is to minimize maximum congestion or alternatively to minimize maximum resource utilization, through efficient resource allocation. When congestion is minimized through efficient resource allocation, packet loss decreases, transit delay decreases, and aggregate throughput increases. Thereby, the perception of network service quality experienced by end users becomes significantly enhanced.

Clearly, load balancing is an important network performance optimization policy. Nevertheless, the capabilities provided for Traffic Engineering should be flexible enough so that network administrators can implement other policies which take into account the prevailing cost structure and the utility or revenue model.

2.2 Traffic and Resource Control

Performance optimization of operational networks is fundamentally a control problem. In the traffic engineering process model, the Traffic Engineer, or a suitable automaton, acts as the controller in an adaptive feedback control system. This system includes a set of interconnected network elements, a network performance monitoring system, and a set of network configuration management tools. The Traffic Engineer formulates a control policy, observes the state of the network through the monitoring system, characterizes the traffic, and applies control actions to drive the network to a desired state, in accordance with the control policy. This can be accomplished reactively by taking action in response to the current state of the network, or pro-actively by using forecasting techniques to anticipate future trends and applying action to obviate the predicted undesirable future states.

Ideally, control actions should involve:

1. Modification of traffic management parameters,

2. Modification of parameters associated with routing, and

3. Modification of attributes and constraints associated with resources.

The level of manual intervention involved in the traffic engineering process should be minimized whenever possible. This can be accomplished by automating aspects of the control actions described above, in a distributed and scalable fashion.

2.3 Limitations of Current IGP Control Mechanisms

This subsection reviews some of the well known limitations of current IGPs with regard to Traffic Engineering.

The control capabilities offered by existing Internet interior gateway protocols are not adequate for Traffic Engineering. This makes it difficult to actualize effective policies to address network performance problems. Indeed, IGPs based on shortest path algorithms contribute significantly to congestion problems in Autonomous Systems within the Internet. SPF algorithms generally optimize based on a simple additive metric. These protocols are topology driven, so bandwidth availability and traffic characteristics are not factors considered in routing decisions. Consequently, congestion frequently occurs when:

1. the shortest paths of multiple traffic streams converge on specific links or router interfaces, or

2. a given traffic stream is routed through a link or router interface which does not have enough bandwidth to accommodate it.

These scenarios manifest even when feasible alternate paths with excess capacity exist. It is this aspect of congestion problems (-- a symptom of suboptimal resource allocation) that Traffic Engineering aims to vigorously obviate. Equal cost path load sharing can be used to address the second cause for congestion listed above with some degree of success, however it is generally not helpful in alleviating congestion due to the first cause listed above and particularly not in large networks with dense topology.

A popular approach to circumvent the inadequacies of current IGPs is through the use of an overlay model, such as IP over ATM or IP over frame relay. The overlay model extends the design space by enabling arbitrary virtual topologies to be provisioned atop the network's physical topology. The virtual topology is constructed from virtual circuits which appear as physical links to the IGP routing protocols. The overlay model provides additional important services to support traffic and resource control, including: (1) constraint-based routing at the VC level, (2) support for administratively configurable explicit VC paths, (3) path compression, (4) call admission control functions, (5) traffic shaping and traffic policing functions, and (6) survivability of VCs. These capabilities enable the actualization of a variety of Traffic Engineering policies. For example, virtual circuits can easily be rerouted to move traffic from over-utilized resources onto relatively underutilized ones.

For Traffic Engineering in large dense networks, it is desirable to equip MPLS with a level of functionality at least commensurate with current overlay models. Fortunately, this can be done in a fairly straight forward manner.

3.0 MPLS and Traffic Engineering

This section provides an overview of the applicability of MPLS to Traffic Engineering. Subsequent sections discuss the set of capabilities required to meet the Traffic Engineering requirements.

MPLS is strategically significant for Traffic Engineering because it can potentially provide most of the functionality available from the overlay model, in an integrated manner, and at a lower cost than the currently competing alternatives. Equally importantly, MPLS offers

the possibility to automate aspects of the Traffic Engineering function. This last consideration requires further investigation and is beyond the scope of this manuscript.

A note on terminology: The concept of MPLS traffic trunks is used extensively in the remainder of this document. According to Li and Rekhter [3], a traffic trunk is an aggregation of traffic flows of the same class which are placed inside a Label Switched Path. Essentially, a traffic trunk is an abstract representation of traffic to which specific characteristics can be associated. It is useful to view traffic trunks as objects that can be routed; that is, the path through which a traffic trunk traverses can be changed. In this respect, traffic trunks are similar to virtual circuits in ATM and Frame Relay networks. It is important, however, to emphasize that there is a fundamental distinction between a traffic trunk and the path, and indeed the LSP, through which it traverses. An LSP is a specification of the label switched path through which the traffic traverses. In practice, the terms LSP and traffic trunk are often used synonymously. Additional characteristics of traffic trunks as used in this manuscript are summarized in section 5.0.

The attractiveness of MPLS for Traffic Engineering can be attributed to the following factors: (1) explicit label switched paths which are not constrained by the destination based forwarding paradigm can be easily created through manual administrative action or through automated action by the underlying protocols, (2) LSPs can potentially be efficiently maintained, (3) traffic trunks can be instantiated and mapped onto LSPs, (4) a set of attributes can be associated with traffic trunks which modulate their behavioral characteristics, (5) a set of attributes can be associated with resources which constrain the placement of LSPs and traffic trunks across them, (6) MPLS allows for both traffic aggregation and disaggregation whereas classical destination only based IP forwarding permits only aggregation, (7) it is relatively easy to integrate a "constraint-based routing" framework with MPLS, (8) a good implementation of MPLS can offer significantly lower overhead than competing alternatives for Traffic Engineering.

Additionally, through explicit label switched paths, MPLS permits a quasi circuit switching capability to be superimposed on the current Internet routing model. Many of the existing proposals for Traffic Engineering over MPLS focus only on the potential to create explicit LSPs. Although this capability is fundamental for Traffic Engineering, it is not really sufficient. Additional augmentations are required to foster the actualization of policies leading to performance optimization of large operational networks. Some of the necessary augmentations are described in this manuscript.

3.1 Induced MPLS Graph

This subsection introduces the concept of an "induced MPLS graph" which is central to Traffic Engineering in MPLS domains. An induced MPLS graph is analogous to a virtual topology in an overlay model. It is logically mapped onto the physical network through the selection of LSPs for traffic trunks.

An induced MPLS graph consists of a set of LSRs which comprise the nodes of the graph and a set of LSPs which provide logical point to point connectivity between the LSRs, and hence serve as the links of the induced graph. it may be possible to construct hierarchical induced MPLS graphs based on the concept of label stacks (see [1]).

Induced MPLS graphs are important because the basic problem of bandwidth management in an MPLS domain is the issue of how to efficiently map an induced MPLS graph onto the physical network topology. The induced MPLS graph abstraction is formalized below.

Let G = (V, E, c) be a capacitated graph depicting the physical topology of the network. Here, V is the set of nodes in the network and E is the set of links; that is, for v and w in V, the object (v,w) is in E if v and w are directly connected under G. The parameter "c" is a set of capacity and other constraints associated with E and V. We will refer to G as the "base" network topology.

Let H = (U, F, d) be the induced MPLS graph, where U is a subset of V representing the set of LSRs in the network, or more precisely the set of LSRs that are the endpoints of at least one LSP. Here, F is the set of LSPs, so that for x and y in U, the object (x, y) is in F if there is an LSP with x and y as endpoints. The parameter "d" is the set of demands and restrictions associated with F. Evidently, H is a directed graph. It can be seen that H depends on the transitivity characteristics of G.

3.2 The Fundamental Problem of Traffic Engineering Over MPLS

There are basically three fundamental problems that relate to Traffic Engineering over MPLS.

- The first problem concerns how to map packets onto forwarding equivalence classes.
- The second problem concerns how to map forwarding equivalence classes onto traffic trunks.
- The third problem concerns how to map traffic trunks onto the physical network topology through label switched paths.

This document is not focusing on the first two problems listed. (even-though they are quite important). Instead, the remainder of this manuscript will focus on the capabilities that permit the third mapping function to be performed in a manner resulting in efficient and reliable network operations. This is really the problem of mapping an induced MPLS graph (H) onto the "base" network topology (G).

4.0 Augmented Capabilities for Traffic Engineering Over MPLS

The previous sections reviewed the basic functions of Traffic Engineering in the contemporary Internet. The applicability of MPLS to that activity was also discussed. The remaining sections of this manuscript describe the functional capabilities required to fully support Traffic Engineering over MPLS in large networks.

The proposed capabilities consist of:

1. A set of attributes associated with traffic trunks which collectively specify their behavioral characteristics.
2. A set of attributes associated with resources which constrain the placement of traffic trunks through them. These can also be viewed as topology attribute constraints.
3. A "constraint-based routing" framework which is used to select paths for traffic trunks subject to constraints imposed by items 1) and 2) above. The constraint-based routing framework does not have to be part of MPLS. However, the two need to be tightly integrated together.

The attributes associated with traffic trunks and resources, as well as parameters associated with routing, collectively represent the control variables which can be modified either through administrative action or through automated agents to drive the network to a desired state.

In an operational network, it is highly desirable that these attributes can be dynamically modified online by an operator without adversely disrupting network operations.

5.0 Traffic Trunk Attributes and Characteristics

This section describes the desirable attributes which can be associated with traffic trunks to influence their behavioral characteristics.

First, the basic properties of traffic trunks (as used in this manuscript) are summarized below:

- A traffic trunk is an *aggregate* of traffic flows belonging to the same class. In some contexts, it may be desirable to relax this definition and allow traffic trunks to include multi-class traffic aggregates.

- In a single class service model, such as the current Internet, a traffic trunk could encapsulate all of the traffic between an ingress LSR and an egress LSR, or subsets thereof.

- Traffic trunks are routable objects (similar to ATM VCs).

- A traffic trunk is distinct from the LSP through which it traverses. In operational contexts, a traffic trunk can be moved from one path onto another.

- A traffic trunk is unidirectional.

In practice, a traffic trunk can be characterized by its ingress and egress LSRs, the forwarding equivalence class which is mapped onto it, and a set of attributes which determine its behavioral characteristics.

Two basic issues are of particular significance: (1) parameterization of traffic trunks and (2) path placement and maintenance rules for traffic trunks.

5.1 Bidirectional Traffic Trunks

Although traffic trunks are conceptually unidirectional, in many practical contexts, it is useful to simultaneously instantiate two traffic trunks with the same endpoints, but which carry packets in opposite directions. The two traffic trunks are logically coupled together. One trunk, called the forward trunk, carries traffic from an originating node to a destination node. The other trunk, called the backward trunk, carries traffic from the destination node to the originating node. We refer to the amalgamation of two such traffic trunks as one bidirectional traffic trunk (BTT) if the following two conditions hold:

- Both traffic trunks are instantiated through an atomic action at one LSR, called the originator node, or through an atomic action at a network management station.

- Neither of the composite traffic trunks can exist without the other. That is, both are instantiated and destroyed together.

The topological properties of BTTs should also be considered. A BTT can be topologically symmetric or topologically asymmetric. A BTT is said to be "topologically symmetric" if its constituent traffic trunks are routed through the same physical path, even though they operate in opposite directions. If, however, the component traffic trunks are routed through different physical paths, then the BTT is said to be "topologically asymmetric."

It should be noted that bidirectional traffic trunks are merely an administrative convenience. In practice, most traffic engineering functions can be implemented using only unidirectional traffic trunks.

5.2 Basic Operations on Traffic Trunks

The basic operations on traffic trunks significant to Traffic Engineering purposes are summarized below.

- Establish: To create an instance of a traffic trunk.

- Activate: To cause a traffic trunk to start passing traffic. The establishment and activation of a traffic trunk are logically separate events. They may, however, be implemented or invoked as one atomic action.

- Deactivate: To cause a traffic trunk to stop passing traffic.

- Modify Attributes: To cause the attributes of a traffic trunk to be modified.

- Reroute: To cause a traffic trunk to change its route. This can be done through administrative action or automatically by the underlying protocols.

- Destroy: To remove an instance of a traffic trunk from the network and reclaim all resources allocated to it. Such resources include label space and possibly available bandwidth.

The above are considered the basic operations on traffic trunks. Additional operations are also possible such as policing and traffic shaping.

5.3 Accounting and Performance Monitoring

Accounting and performance monitoring capabilities are very important to the billing and traffic characterization functions. Performance statistics obtained from accounting and performance monitoring systems can be used for traffic characterization, performance optimization, and capacity planning within the Traffic Engineering realm..

The capability to obtain statistics at the traffic trunk level is so important that it should be considered an essential requirement for Traffic Engineering over MPLS.

5.4 Basic Traffic Engineering Attributes of Traffic Trunks

An attribute of a traffic trunk is a parameter assigned to it which influences its behavioral characteristics.

Attributes can be explicitly assigned to traffic trunks through administration action or they can be implicitly assigned by the underlying protocols when packets are classified and mapped into equivalence classes at the ingress to an MPLS domain. Regardless of how the attributes were originally assigned, for Traffic Engineering purposes, it should be possible to administratively modify such attributes.

The basic attributes of traffic trunks particularly significant for Traffic Engineering are itemized below.

- Traffic parameter attributes

- Generic Path selection and maintenance attributes

- Priority attribute

- Preemption attribute

- Resilience attribute

- Policing attribute

The combination of traffic parameters and policing attributes is analogous to usage parameter control in ATM networks. Most of the attributes listed above have analogs in well established technologies. Consequently, it should be relatively straight forward to map the traffic trunk attributes onto many existing switching and routing architectures.

Priority and preemption can be regarded as relational attributes because they express certain binary relations between traffic trunks. Conceptually, these binary relations determine the manner in which traffic trunks interact with each other as they compete for network resources during path establishment and path maintenance.

5.5 Traffic parameter attributes

Traffic parameters can be used to capture the characteristics of the traffic streams (or more precisely the forwarding equivalence class) to be transported through the traffic trunk. Such characteristics may include peak rates, average rates, permissible burst size, etc. From a traffic engineering perspective, the traffic parameters are significant because they indicate the resource requirements of the traffic trunk. This is useful for resource allocation and congestion avoidance through anticipatory policies.

For the purpose of bandwidth allocation, a single canonical value of bandwidth requirements can be computed from a traffic trunk's traffic parameters. Techniques for performing these computations are well known. One example of this is the theory of effective bandwidth.

5.6 Generic Path Selection and Management Attributes

Generic path selection and management attributes define the rules for selecting the route taken by a traffic trunk as well as the rules for maintenance of paths that are already established.

Paths can be computed automatically by the underlying routing protocols or they can be defined administratively by a network operator. If there are no resource requirements or restrictions associated with a traffic trunk, then a topology driven protocol can be used to select its path. However, if resource requirements or policy restrictions exist, then a constraint-based routing scheme should be used for path selection.

In Section 7, a constraint-based routing framework which can automatically compute paths subject to a set of constraints is described. Issues pertaining to explicit paths instantiated through administrative action are discussed in Section 5.6.1 below.

Path management concerns all aspects pertaining to the maintenance of paths traversed by traffic trunks. In some operational contexts, it is desirable that an MPLS implementation can dynamically reconfigure itself, to adapt to some notion of change in "system state." Adaptivity and resilience are aspects of dynamic path management.

To guide the path selection and management process, a set of attributes are required. The basic attributes and behavioral characteristics associated with traffic trunk path selection and management are described in the remainder of this sub-section.

5.6.1 Administratively Specified Explicit Paths

An administratively specified explicit path for a traffic trunk is one which is configured through operator action. An administratively specified path can be completely specified or partially specified. A path is completely specified if all of the required hops between the endpoints are indicated. A path is partially specified if only a subset of intermediate hops are indicated. In this case, the underlying protocols are required to complete the path. Due to operator errors, an administratively specified path can be inconsistent or illogical. The underlying protocols should be able to detect such inconsistencies and provide appropriate feedback.

A "path preference rule" attribute should be associated with administratively specified explicit paths. A path preference rule attribute is a binary variable which indicates whether the administratively configured explicit path is "mandatory" or "non-mandatory."

If an administratively specified explicit path is selected with a "mandatory attribute, then that path (and only that path) must be used. If a mandatory path is topological infeasible (e.g. the two endpoints are topologically partitioned), or if the path cannot be instantiated because the available resources are inadequate, then the path setup process fails. In other words, if a path is specified as mandatory, then an alternate path cannot be used regardless of prevailing circumstance. A mandatory path which is successfully instantiated is also implicitly pinned. Once the path is instantiated it cannot be changed except through deletion and instantiation of a new path.

However, if an administratively specified explicit path is selected with a "non-mandatory" preference rule attribute value, then the path should be used if feasible. Otherwise, an alternate path can be chosen instead by the underlying protocols.

5.6.2 Hierarchy of Preference Rules For Multi-Paths

In some practical contexts, it can be useful to have the ability to administratively specify a set of candidate explicit paths for a given traffic trunk and define a hierarchy of preference relations on the paths. During path establishment, the preference rules are applied to select a suitable path from the candidate list. Also, under failure scenarios the preference rules are applied to select an alternate path from the candidate list.

5.6.3 Resource Class Affinity Attributes

Resource class affinity attributes associated with a traffic trunk can be used to specify the class of resources (see Section 6) which are to be explicitly included or excluded from the path of the traffic trunk. These are policy attributes which can be used to impose additional constraints on the path traversed by a given traffic trunk. Resource class affinity attributes for a traffic can be specified as a sequence of tuples:

<resource-class, affinity>; <resource-class, affinity>; ..

The resource-class parameter identifies a resource class for which an affinity relationship is defined with respect to the traffic trunk. The affinity parameter indicates the affinity relationship; that is, whether members of the resource class are to be included or excluded from the path of the traffic trunk. Specifically, the affinity parameter may be a binary variable which takes one of the following values: (1) explicit inclusion, and (2) explicit exclusion.

If the affinity attribute is a binary variable, it may be possible to use Boolean expressions to specify the resource class affinities associated with a given traffic trunk.

If no resource class affinity attributes are specified, then a "don't care" affinity relationship is assumed to hold between the traffic trunk and all resources. That is, there is no requirement to explicitly include or exclude any resources from the traffic trunk's path. This should be the default in practice.

Resource class affinity attributes are very useful and powerful constructs because they can be used to implement a variety of policies. For example, they can be used to contain certain traffic trunks within specific topological regions of the network.

A "constraint-based routing" framework (see section 7.0) can be used to compute an explicit path for a traffic trunk subject to resource class affinity constraints in the following manner:

1. For explicit inclusion, prune all resources not belonging to the specified classes prior to performing path computation.

2. For explicit exclusion, prune all resources belonging to the specified classes before performing path placement computations.

5.6.4 Adaptivity Attribute

Network characteristics and state change over time. For example, new resources become available, failed resources become reactivated, and allocated resources become deallocated. In general, sometimes more efficient paths become available. Therefore, from a Traffic Engineering perspective, it is necessary to have administrative control parameters that can be used to specify how traffic trunks respond to this dynamism. In some scenarios, it might be desirable to dynamically change the paths of certain traffic trunks in response to changes in network state. This process is called re-optimization. In other scenarios, re-optimization might be very undesirable.

An Adaptivity attribute is a part of the path maintenance parameters associated with traffic trunks. The adaptivity attribute associated with a traffic trunk indicates whether the trunk is subject to re-optimization. That is, an adaptivity attribute is a binary variable which takes one of the following values: (1) permit re-optimization and (2) disable re-optimization.

If re-optimization is enabled, then a traffic trunk can be rerouted through different paths by the underlying protocols in response to changes in network state (primarily changes in resource availability). Conversely, if re-optimization is disabled, then the traffic trunk is "pinned" to its established path and cannot be rerouted in response to changes in network state.

Stability is a major concern when re-optimization is permitted. To promote stability, an MPLS implementation should not be too reactive to the evolutionary dynamics of the network. At the same time, it must adapt fast enough so that optimal use can be made of network assets. This implies that the frequency of re-optimization should be administratively configurable to allow for tuning.

It is to be noted that re-optimization is distinct from resilience. A different attribute is used to specify the resilience characteristics of a traffic trunk (see section 5.9). In practice, it would seem reasonable to expect traffic trunks subject to re-optimization to be implicitly resilient to failures along their paths. However, a traffic trunk which is not subject to re-optimization and whose path is not administratively specified with a "mandatory" attribute can also be required to be resilient to link and node failures along its established path

Formally, it can be stated that adaptivity to state evolution through re-optimization implies resilience to failures, whereas resilience to failures does not imply general adaptivity through re-optimization to state changes.

5.6.5 Load Distribution Across Parallel Traffic Trunks

Load distribution across multiple parallel traffic trunks between two nodes is an important consideration. In many practical contexts, the aggregate traffic between two nodes may be such that no single link (hence no single path) can carry the load. However, the aggregate flow might be less than the maximum permissible flow across a "min-cut" that partitions the two nodes. In this case, the only feasible solution is to appropriately divide the aggregate traffic into sub-streams and route the sub-streams through multiple paths between the two nodes.

In an MPLS domain, this problem can be addressed by instantiating multiple traffic trunks between the two nodes, such that each traffic trunk carries a proportion of the aggregate traffic. Therefore, a flexible means of load assignment to multiple parallel traffic trunks carrying traffic between a pair of nodes is required.

Specifically, from an operational perspective, in situations where parallel traffic trunks are warranted, it would be useful to have some attribute that can be used to indicate the relative proportion of traffic to be carried by each traffic trunk. The underlying protocols will then map the load onto the traffic trunks according to the specified proportions. It is also, generally desirable to maintain packet ordering between packets belong to the same micro-flow (same source address, destination address, and port number).

5.7 Priority attribute

The priority attribute defines the relative importance of traffic trunks. If a constraint-based routing framework is used with MPLS, then priorities become very important because they can be used to determine the order in which path selection is done for traffic trunks at connection establishment and under fault scenarios.

Priorities are also important in implementations permitting preemption because they can be used to impose a partial order on the set of traffic trunks according to which preemptive policies can be actualized.

5.8 Preemption attribute

The preemption attribute determines whether a traffic trunk can preempt another traffic trunk from a given path, and whether another traffic trunk can preempt a specific traffic trunk. Preemption is useful for both traffic oriented and resource oriented performance

objectives. Preemption can used to assure that high priority traffic trunks can always be routed through relatively favorable paths within a differentiated services environment.

Preemption can also be used to implement various prioritized restoration policies following fault events.

The preemption attribute can be used to specify four preempt modes for a traffic trunk: (1) preemptor enabled, (2) non-preemptor, (3) preemptable, and (4) non-preemptable. A preemptor enabled traffic trunk can preempt lower priority traffic trunks designated as preemptable. A traffic specified as non-preemptable cannot be preempted by any other trunks, regardless of relative priorities. A traffic trunk designated as preemptable can be preempted by higher priority trunks which are preemptor enabled.

It is trivial to see that some of the preempt modes are mutually exclusive. Using the numbering scheme depicted above, the feasible preempt mode combinations for a given traffic trunk are as follows: (1, 3), (1, 4), (2, 3), and (2, 4). The (2, 4) combination should be the default.

A traffic trunk, say "A", can preempt another traffic trunk, say "B", only if *all* of the following five conditions hold: (i) "A" has a relatively higher priority than "B", (ii) "A" contends for a resource utilized by "B", (iii) the resource cannot concurrently accommodate "A" and "B" based on certain decision criteria, (iv) "A" is preemptor enabled, and (v) "B" is preemptable.

Preemption is not considered a mandatory attribute under the current best effort Internet service model although it is useful. However, in a differentiated services scenario, the need for preemption becomes more compelling. Moreover, in the emerging optical internetworking architectures, where some protection and restoration functions may be migrated from the optical layer to data network elements (such as gigabit and terabit label switching routers) to reduce costs, preemptive strategies can be used to reduce the restoration time for high priority traffic trunks under fault conditions.

5.9 Resilience Attribute

The resilience attribute determines the behavior of a traffic trunk under fault conditions. That is, when a fault occurs along the path through which the traffic trunk traverses. The following basic problems need to be addressed under such circumstances: (1) fault detection, (2) failure notification, (3) recovery and service restoration. Obviously, an MPLS implementation will have to incorporate mechanisms to address these issues.

Many recovery policies can be specified for traffic trunks whose established paths are impacted by faults. The following are examples of feasible schemes:

1. Do not reroute the traffic trunk. For example, a survivability scheme may already be in place, provisioned through an alternate mechanism, which guarantees service continuity under failure scenarios without the need to reroute traffic trunks. An example of such an alternate scheme (certainly many others exist), is a situation whereby multiple parallel label switched paths are provisioned between two nodes, and function in a manner such that failure of one LSP causes the traffic trunk placed on it to be mapped onto the remaining LSPs according to some well defined policy.

2. Reroute through a feasible path with enough resources. If none exists, then do not reroute.

3. Reroute through any available path regardless of resource constraints.

4. Many other schemes are possible including some which might be combinations of the above.

A "basic" resilience attribute indicates the recovery procedure to be applied to traffic trunks whose paths are impacted by faults. Specifically, a "basic" resilience attribute is a binary variable which determines whether the target traffic trunk is to be rerouted when segments of its path fail. "Extended" resilience attributes can be used to specify detailed actions to be taken under fault scenarios. For example, an extended resilience attribute might specify a set of alternate paths to use under fault conditions, as well as the rules that govern the relative preference of each specified path.

Resilience attributes mandate close interaction between MPLS and routing.

5.10 Policing attribute

The policing attribute determines the actions that should be taken by the underlying protocols when a traffic trunk becomes non-compliant. That is, when a traffic trunk exceeds its contract as specified in the traffic parameters. Generally, policing attributes can indicate whether a non-conformant traffic trunk is to be rate limited, tagged, or simply forwarded without any policing action. If policing is used, then adaptations of established algorithms such as the ATM Forum's GCRA [11] can be used to perform this function.

Policing is necessary in many operational scenarios, but is quite undesirable in some others. In general, it is usually desirable to police at the ingress to a network (to enforce compliance with service level agreements) and to minimize policing within the core, except when capacity constraints dictate otherwise.

Therefore, from a Traffic Engineering perspective, it is necessary to be able to administratively enable or disable traffic policing for each traffic trunk.

6.0 Resource Attributes

Resource attributes are part of the topology state parameters, which are used to constrain the routing of traffic trunks through specific resources.

6.1 Maximum Allocation Multiplier

The maximum allocation multiplier (MAM) of a resource is an administratively configurable attribute which determines the proportion of the resource that is available for allocation to traffic trunks. This attribute is mostly applicable to link bandwidth. However, it can also be applied to buffer resources on LSRs. The concept of MAM is analogous to the concepts of subscription and booking factors in frame relay and ATM networks.

The values of the MAM can be chosen so that a resource can be under-allocated or over-allocated. A resource is said to be under-allocated if the aggregate demands of all traffic trunks (as expressed in the trunk traffic parameters) that can be allocated to it are always less than the capacity of the resource. A resource is said to be over-allocated if the aggregate demands of all traffic trunks allocated to it can exceed the capacity of the resource.

Under-allocation can be used to bound the utilization of resources. However,the situation under MPLS is more complex than in circuit switched schemes because under MPLS, some flows can be routed via conventional hop by hop protocols (also via explicit paths) without consideration for resource constraints.

Over-allocation can be used to take advantage of the statistical characteristics of traffic in order to implement more efficient resource allocation policies. In particular, over-allocation can be used in situations where the peak demands of traffic trunks do not coincide in time.

6.2 Resource Class Attribute

Resource class attributes are administratively assigned parameters which express some notion of "class" for resources. Resource class attributes can be viewed as "colors" assigned to resources such that the set of resources with the same "color" conceptually belong to the same class. Resource class attributes can be used to implement a variety of policies. The key resources of interest here are links. When applied to links, the resource class attribute effectively becomes an aspect of the "link state" parameters.

The concept of resource class attributes is a powerful abstraction. From a Traffic Engineering perspective, it can be used to implement many policies with regard to both traffic and resource oriented performance optimization. Specifically, resource class attributes can be used to:

1. Apply uniform policies to a set of resources that do not need to be in the same topological region.

2. Specify the relative preference of sets of resources for path placement of traffic trunks.

3. Explicitly restrict the placement of traffic trunks to specific subsets of resources.

4. Implement generalized inclusion / exclusion policies.

5. Enforce traffic locality containment policies. That is, policies that seek to contain local traffic within specific topological regions of the network.

Additionally, resource class attributes can be used for identification purposes.

In general, a resource can be assigned more than one resource class attribute. For example, all of the OC-48 links in a given network may be assigned a distinguished resource class attribute. The subsets of OC-48 links which exist with a given abstraction domain of the network may be assigned additional resource class attributes in order to implement specific containment policies, or to architect the network in a certain manner.

7.0 Constraint-Based Routing

This section discusses the issues pertaining to constraint-based routing in MPLS domains. In contemporary terminology, constraint-based routing is often referred to as "QoS Routing" see [5,6,7,10].

This document uses the term "constraint-based routing" however, because it better captures the functionality envisioned, which generally encompasses QoS routing as a subset.

constraint-based routing enables a demand driven, resource reservation aware, routing paradigm to co-exist with current topology driven hop by hop Internet interior gateway protocols.

A constraint-based routing framework uses the following as input:

- The attributes associated with traffic trunks.

- The attributes associated with resources.

- Other topology state information.

Based on this information, a constraint-based routing process on each node automatically computes explicit routes for each traffic trunk originating from the node. In this case, an explicit route for each traffic trunk is a specification of a label switched path that satisfies the demand requirements expressed in the trunk's attributes, subject to constraints imposed by resource availability, administrative policy, and other topology state information.

A constraint-based routing framework can greatly reduce the level of manual configuration and intervention required to actualize Traffic Engineering policies.

In practice, the Traffic Engineer, an operator, or even an automaton will specify the endpoints of a traffic trunk and assign a set of attributes to the trunk which encapsulate the performance expectations and behavioral characteristics of the trunk. The constraint-based routing framework is then expected to find a feasible path to satisfy the expectations. If necessary, the Traffic Engineer or a traffic engineering support system can then use administratively configured explicit routes to perform fine grained optimization.

7.1 Basic Features of Constraint-Based Routing

A constraint-based routing framework should at least have the capability to automatically obtain a basic feasible solution to the traffic trunk path placement problem.

In general, the constraint-based routing problem is known to be intractable for most realistic constraints. However, in practice, a very simple well known heuristic (see e.g. [9]) can be used to find a feasible path if one exists:

- First prune resources that do not satisfy the requirements of the traffic trunk attributes.

- Next, run a shortest path algorithm on the residual graph.

Clearly, if a feasible path exists for a single traffic trunk, then the above simple procedure will find it. Additional rules can be specified to break ties and perform further optimizations. In general, ties should be broken so that congestion is minimized. When multiple traffic trunks are to be routed, however, it can be shown that the above algorithm may not always find a mapping, even when a feasible mapping exists.

7.2 Implementation Considerations

Many commercial implementations of frame relay and ATM switches already support some notion of constraint-based routing. For such devices or for the novel MPLS centric contraptions devised therefrom, it should be relatively easy to extend the current constraint-based routing implementations to accommodate the peculiar requirements of MPLS.

For routers that use topology driven hop by hop IGPs, constraint-based routing can be incorporated in at least one of two ways:

1. By extending the current IGP protocols such as OSPF and IS-IS to support constraint-based routing. Effort is already underway to provide such extensions to OSPF (see [5,7]).

2. By adding a constraint-based routing process to each router which can co-exist with current IGPs. This scenario is depicted in Figure 1.

```
         ------------------------------------------
        |          Management Interface            |
         ------------------------------------------
             |                 |                 |
     ------------      ------------------     --------------
    |    MPLS    |<->| Constraint-Based |   | Conventional |
    |            |   | Routing Process  |   | IGP Process  |
     ------------      ------------------     --------------
                              |                   |
             -----------------------       --------------
            | Resource  Attribute   |     | Link State   |
            | Availability Database |     | Database     |
             -----------------------       --------------
```

Figure 1. Constraint-Based Routing Process on Layer 3 LSR

There are many important details associated with implementing constraint-based routing on Layer 3 devices which we do not discuss here. These include the following:

- Mechanisms for exchange of topology state information (resource availability information, link state information, resource attribute information) between constraint-based routing processes.

- Mechanisms for maintenance of topology state information.

- Interaction between constraint-based routing processes and conventional IGP processes.

- Mechanisms to accommodate the adaptivity requirements of traffic trunks.

- Mechanisms to accommodate the resilience and survivability requirements of traffic trunks.

In summary, constraint-based routing assists in performance optimization of operational networks by automatically finding feasible paths that satisfy a set of constraints for traffic trunks. It can drastically reduce the amount of administrative explicit path configuration and manual intervention required to achieve Traffic Engineering objectives.

8.0 Conclusion

This manuscript presented a set of requirements for Traffic Engineering over MPLS. Many capabilities were described aimed at enhancing the applicability of MPLS to Traffic Engineering in the Internet.

It should be noted that some of the issues described here can be addressed by incorporating a minimal set of building blocks into MPLS, and then using a network management superstructure to extend the functionality in order to realize the requirements. Also, the constraint-based routing framework does not have to be part of the core MPLS specifications. However, MPLS does require some interaction with a constraint-based routing framework in order to meet the requirements.

9.0 Security Considerations

This document does not introduce new security issues beyond those inherent in MPLS and may use the same mechanisms proposed for this technology. It is, however, specifically important that manipulation of administratively configurable parameters be executed in a secure manner by authorized entities.

10.0 References

[1] Rosen, E., Viswanathan, A. and R. Callon, "A Proposed Architecture for MPLS", Work in Progress.

[2] Callon, R., Doolan, P., Feldman, N., Fredette, A., Swallow, G. and A. Viswanathan, "A Framework for Multiprotocol Label Switching", Work in Progress.

[3] Li, T. and Y. Rekhter, "Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)", RFC 2430, October 1998.

[4] Rekhter, Y., Davie, B., Katz, D., Rosen, E. and G. Swallow, "Cisco Systems' Tag Switching Architecture - Overview", RFC 2105, February 1997.

[5] Zhang, Z., Sanchez, C., Salkewicz, B. and E. Crawley "Quality of Service Extensions to OSPF", Work in Progress.

[6] Crawley, E., Nair, F., Rajagopalan, B. and H. Sandick, "A Framework for QoS Based Routing in the Internet", RFC 2386, August 1998.

[7] Guerin, R., Kamat, S., Orda, A., Przygienda, T. and D. Williams, "QoS Routing Mechanisms and OSPF Extensions", RFC 2676, August 1999.

[8] C. Yang and A. Reddy, "A Taxonomy for Congestion Control Algorithms in Packet Switching Networks," IEEE Network Magazine, Volume 9, Number 5, July/August 1995.

[9] W. Lee, M. Hluchyi, and P. Humblet, "Routing Subject to Quality of Service Constraints in Integrated Communication Networks," IEEE Network, July 1995, pp 46-55.

[10] ATM Forum, "Traffic Management Specification: Version 4.0" April 1996.

11.0 Acknowledgments

The authors would like to thank Yakov Rekhter for his review of an earlier draft of this document. The authors would also like to thank Louis Mamakos and Bill Barns for their helpful suggestions, and Curtis Villamizar for providing some useful feedback.

12.0 Authors' Addresses

Daniel O. Awduche
UUNET (MCI Worldcom)
3060 Williams Drive
Fairfax, VA 22031

Phone: +1 703-208-5277
EMail: awduche@uu.net

Joe Malcolm
UUNET (MCI Worldcom)
3060 Williams Drive

Fairfax, VA 22031

Phone: +1 703-206-5895
EMail: jmalcolm@uu.net

Johnson Agogbua
UUNET (MCI Worldcom)
3060 Williams Drive
Fairfax, VA 22031

Phone: +1 703-206-5794
EMail: ja@uu.net

Mike O'Dell
UUNET (MCI Worldcom)
3060 Williams Drive
Fairfax, VA 22031

Phone: +1 703-206-5890
EMail: mo@uu.net

Jim McManus
UUNET (MCI Worldcom)
3060 Williams Drive
Fairfax, VA 22031

Phone: +1 703-206-5607
EMail: jmcmanus@uu.net

13.0 Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.