TEAS Working Group   A. Farrel, Ed.
Internet-Draft   Old Dog Consulting
Obsoletes: 3272 (if approved)   April 7, 2021
Intended status: Informational
Expires: October 9, 2021

Overview and Principles of Internet Traffic Engineering
draft-ietf-teas-rfc3272bis-11

Abstract

This document describes the principles of traffic engineering (TE) in the Internet. The document is intended to promote better understanding of the issues surrounding traffic engineering in IP networks and the networks that support IP networking, and to provide a common basis for the development of traffic engineering capabilities for the Internet. The principles, architectures, and methodologies for performance evaluation and performance optimization of operational networks are also discussed.

This work was first published as RFC 3272 in May 2002. This document obsoletes RFC 3272 by making a complete update to bring the text in line with best current practices for Internet traffic engineering and to include references to the latest relevant work in the IETF.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document describes the principles of Internet ~~traffic~~Traffic ~~engineering~~ Engineering (TE). The objective of the document is to articulate the general issues and principles for Internet traffic engineering, and where appropriate to provide recommendations, guidelines, and options for the development of online and offline Internet traffic engineering capabilities and support systems.

Commenté [BMT1]: TE can be enabled in closed networks, not only "Internet".
I see the note about single domain, but I wonder whether that text can be positioned righter after this one.

Commenté [BMT2]: Not introduced yet.

This document provides a terminology and taxonomy for describing and understanding common Internet traffic engineering concepts.

Even though Internet traffic engineering is most effective when applied end-to-end, the focus of this document is traffic engineering within a given domain (such as an autonomous system). However, because a preponderance of Internet traffic tends to originate in one autonomous system and terminate in another, this document also provides an overview of aspects pertaining to inter-domain traffic engineering.

Mis en forme : Surlignage

This work was first published as [RFC3272] in May 2002. This document obsoletes [RFC3272] by making a complete update to bring the text in line with best current practices for Internet traffic engineering and to include references to the latest relevant work in the IETF. It is worth noting around three fifths of the RFCs referenced in this document post-date the publication of RFC 3272. Appendix C provides a summary of changes between RFC 3272 and this document.

1.1. What is Internet Traffic Engineering?

One of the most significant functions performed ~~by~~ in the Internet is the forwarding/routing of traffic from ingress nodes to egress nodes. Therefore, one of the most distinctive functions performed by Internet traffic engineering is the control and optimization of the ~~Routing~~routing/forwarding functions, to steer traffic through the network.

Internet traffic engineering is defined as that aspect of Internet network engineering dealing with the issues of performance evaluation and performance optimization of operational IP networks. Traffic engineering encompasses the application of technology and scientific principles to the measurement, characterization, modeling, and control of Internet traffic [RFC2702], [AWD2].

It is the performance of the network as seen by end users of network services that is paramount. The characteristics visible to end users are the emergent properties of the network, which are the characteristics of the network when viewed as a whole. A central goal of the service provider, therefore, is to enhance the emergent properties of the network while taking economic considerations into account. This is accomplished by addressing traffic oriented performance requirements while utilizing network resources economically and reliably. Traffic oriented performance measures include delay, delay variation, packet loss, and throughput.

Internet traffic engineering responds to network events. Aspects of capacity management respond at intervals ranging from days to years. Routing control functions operate at intervals ranging from milliseconds to days. Packet level processing functions operate at very fine levels of temporal resolution, ranging from picoseconds to milliseconds while reacting to the real-time statistical behavior of traffic.

Thus, the optimization aspects of traffic engineering can be viewed from a control perspective, and can be both pro-active and reactive. In the pro-active case, the traffic engineering control system takes preventive action to protect against predicted unfavorable future network states, for example, by engineering backup paths. It may also take action that will lead to a more desirable future network state. In the reactive case, the control system responds to correct issues and adapt to network events, such as routing after failure.

Another important objective of Internet traffic engineering is to facilitate reliable network operations [RFC2702]. Reliable network operations can be facilitated by providing mechanisms that enhance network integrity and by embracing policies emphasizing network survivability. This reduces the vulnerability of services to outages arising from errors, faults, and failures occurring within the network infrastructure.

The optimization aspects of traffic engineering can be achieved through capacity management and traffic management. In this document, capacity management includes capacity planning, routing control, and resource management. Network resources of particular interest include link bandwidth, buffer space, and computational resources. In this document, traffic management includes:

1. nodal traffic control functions such as traffic conditioning, queue management, and scheduling

2. other functions that regulate traffic ~~flow~~ flows through the network or that arbitrate access to network resources between different packets or between different traffic streams.

One major challenge of Internet traffic engineering is the realization of automated control capabilities that adapt quickly and cost effectively to significant changes in network state, while still maintaining stability of the network. Performance evaluation can assess the effectiveness of traffic engineering methods, and the results of this evaluation can be used to identify existing problems, guide network re-optimization, and aid in the prediction of potential future problems. However, this process can also be time consuming and may not be suitable to act on short-lived changes in the network.

Performance evaluation can be achieved in many different ways. The most notable techniques include analytical methods, simulation, and empirical methods based on measurements.

Traffic engineering comes in two flavors: either a background process that constantly monitors traffic and optimizes the use of resources to improve performance; or a form of a pre-planned optimized traffic distribution that is considered optimal. In the later case, any deviation from the optimum distribution (e.g., caused by a fiber cut) is reverted upon repair without further optimization. However, this form of traffic engineering relies upon the notion that the planned state of the network is optimal. Hence, in such a mode there are two levels of traffic engineering: the TE-planning task to enable optimum traffic distribution, and the routing/forwarding task that ~~keeping~~ keeps traffic flows attached to the pre-planned distribution.

As a general rule, traffic engineering concepts and mechanisms must be sufficiently specific and well-defined to address known requirements, but simultaneously flexible and extensible to accommodate unforeseen future demands.

Commenté [BMT3]: This is mentioned in Section 6.1. A pointer to that section where this requirement is further elaborated, would be useful. Thanks.

1.2. Components of Traffic Engineering

As mentioned in Section 1.1, Internet traffic engineering provides performance optimization of ~~operational~~ IP networks while utilizing network resources economically and reliably. Such optimization is supported at the control/controller level and within the data/forwarding plane.

Mis en forme : Surlignage

The key elements required in any TE solution are as follows:

1. Policy

2. Path steering

3. Resource management

Some TE solutions rely on these elements to a lesser or greater extent. Debate remains about whether a solution can truly be called traffic engineering if it does not include all of these elements. For the sake of this document, we assert that all TE solutions must include some aspects of all of these elements. Other solutions can be classed as "partial TE" and also fall in scope of this document.

Policy allows for the selection of ~~next hops and~~ paths (including next-hops) based on information beyond basic reachability. Early definitions of routing policy, e.g., [RFC1102] and [RFC1104], discuss routing policy being applied to restrict access to network resources at an aggregate level. BGP is an example of a commonly used mechanism for applying such policies, see [RFC4271] and [RFC8955]. In the traffic engineering context, policy decisions are made within the control plane~~ or by controllers~~, and govern the selection of paths. Examples can be found in [RFC4655] and [RFC5394]. Standard TE solutions may cover the mechanisms to distribute and/or enforce polices, but specific policy definition is generally unspecified.

Path steering is the ability to forward packets using more information than just knowledge of the next hop. Examples of path steering include IPv4 source routes [RFC0791], RSVP-TE explicit routes [RFC3209], ~~and~~ Segment Routing [RFC8402], and SFC [RFC7665]. Path steering for TE can be supported via control plane protocols, by encoding in the data plane headers, or by a combination of the two. This includes when control is provided by a controller using a ~~southbound~~ network-facing ~~(i.e., controller to router)~~ control protocol.

Resource management provides ~~resource~~ resource-aware control and forwarding. Examples of resources are bandwidth, buffers, and queues, all of which can be managed to control loss and latency.

> Resource reservation is the control aspect of resource management. It provides for domain-wide consensus about which network resources are used by a particular flow. This determination may be made at a very course or very fine level. Note that this consensus exists at the network control or controller level, not within the data plane. It may be composed purely of accounting/bookkeeping, but it typically includes an ability to admit, reject, or reclassify a flow based on policy. Such accounting can be done based on any combination of a static understanding of resource requirements, and the use of dynamic mechanisms to collect requirements (e.g., via [RFC3209]) and resource availability (e.g., via [RFC4203]).

Resource allocation is the data plane aspect of resource management. It provides for the allocation of specific node and link resources to specific flows. Example resources include buffers, policing, and rate-shaping mechanisms that are typically supported via queuing. It also includes the matching of a flow (i.e., flow classification) to a particular set of allocated resources. The method of flow classification and granularity of resource management is technology specific. Examples include Diffserv with dropping and remarking [RFC4594], MPLS-TE [RFC3209], and GMPLS based label switched paths [RFC3945], as well as controller-based solutions [RFC8453]. This level of resource control, while optional, is important in networks that wish to support congestion management policies to control or regulate the offered traffic to deliver different levels of service and alleviate congestion problems, or those networks that wish to control latencies experienced by specific traffic flows.

Commenté [BMT4]: What about resource-based access control. Should it be considered as part a sub-component of this one?

1.3. Scope

The scope of this document is intra-domain traffic engineering. That is, traffic engineering within a given autonomous system in the Internet. This document discusses concepts pertaining to intra-domain traffic control, including such issues as routing control, micro and macro resource allocation, and the control coordination problems that arise consequently.

This document describes and characterizes techniques already in use or in advanced development for Internet traffic engineering. The way these techniques fit together is discussed and scenarios in which they are useful will be identified.

Although the emphasis in this document is on intra-domain traffic engineering, in Section 7, an overview of the high level considerations pertaining to inter-domain traffic engineering ~~will be~~are provided. Inter-domain Internet traffic engineering is crucial to the performance enhancement of the global Internet infrastructure.

Whenever possible, relevant requirements from existing IETF documents and other sources are incorporated by reference.

1.4. Terminology

This section provides terminology which is useful for Internet traffic engineering. The definitions presented apply to this document. These terms may have other meanings elsewhere.

Busy hour: A one hour period within a specified interval of time (typically 24 hours) in which the traffic load in a network or sub-network is greatest.

Congestion: A state of a network resource in which the traffic incident on the resource exceeds its output capacity over an interval of time.

Congestion avoidance: An approach to congestion management that attempts to obviate the occurrence of congestion.

Congestion control: An approach to congestion management that attempts to remedy congestion problems that have already occurred.

Constraint-based routing: A class of routing protocols that take specified traffic attributes, network constraints, and policy constraints into account when making routing decisions. Constraint-based routing is applicable to traffic aggregates as well as flows. It is a generalization of QoS-based routing.

Demand side congestion management: A congestion management scheme that addresses congestion problems by regulating or conditioning offered load.

Effective bandwidth: The minimum amount of bandwidth that can be assigned to a flow or traffic aggregate in order to deliver 'acceptable service quality' to the flow or traffic aggregate.

Hot-spot: A network element or subsystem which is in a state of congestion.

Inter-domain traffic: Traffic that originates in one Autonomous system and terminates in another.

Metric: A parameter defined in terms of standard units of measurement.

Measurement methodology: A repeatable measurement technique used to derive one or more metrics of interest.

Network survivability: The capability to provide a prescribed level of QoS for existing services after a given number of failures occur within the network.

Offline traffic engineering: A traffic engineering system that exists outside of the network.

Online traffic engineering: A traffic engineering system that exists within the network, typically implemented on or as adjuncts to operational network elements.

Performance measures: Metrics that provide quantitative or qualitative measures of the performance of systems or subsystems of interest.

Performance metric: A performance parameter defined in terms of standard units of measurement.

Provisioning: The process of assigning or configuring network resources to meet certain requests.

QoS routing: Class of routing systems that selects paths to be used by a flow based on the QoS requirements of the flow.

Service Level Agreement (SLA): A contract between a provider and a customer that guarantees specific levels of performance and reliability at a certain cost.

Service Level Objective (SLO): A key element of an SLA between a provider and a customer. SLOs are agreed upon as a means of measuring the performance of the Service Provider and are outlined as a way of avoiding disputes between the two parties based on misunderstanding.

Stability: An operational state in which a network does not oscillate in a disruptive manner from one mode to another mode.

Supply-side congestion management: A congestion management scheme that provisions additional network resources to address existing and/or anticipated congestion problems.

Traffic characteristic: A description of the temporal behavior or a description of the attributes of a given traffic flow or traffic aggregate.

Traffic engineering system: A collection of objects, mechanisms, and protocols that are used together to accomplish traffic engineering objectives.

Traffic flow: A stream of packets between two end-points that can be characterized in a certain way. A micro-flow has a more specific ~~definition~~ definition. A micro-flow is a stream of packets with the same source and destination addresses, source and destination ports, and protocol ID.

Commenté [BMT5]: I'm not sure there a need to define this new term. I would provide the 5-uple as an example to characterize a flow.

Traffic matrix: A representation of the traffic demand between a set of origin and destination abstract nodes. An abstract node can consist of one or more network elements.

Traffic monitoring: The process of observing traffic characteristics at a given point in a network and collecting the traffic information for analysis and further action.

Traffic trunk: An aggregation of traffic flows belonging to the same class which are forwarded through a common path. A traffic trunk may be characterized by an ingress and egress node, and a set of attributes which determine its behavioral characteristics and requirements from the network.

2. Background

The Internet must convey IP packets from ingress nodes to egress nodes efficiently, expeditiously, and economically. Furthermore, in a multiclass service environment (e.g., Diffserv capable networks - see Section 4.1.4), the resource sharing parameters of the network must be appropriately determined and configured according to prevailing policies and service models to resolve resource contention issues arising from mutual interference between packets traversing through the network. Thus, consideration must be given to resolving competition for network resources between traffic ~~streams~~ flows belonging to the same service class (intra-class contention resolution) and traffic streams belonging to different classes (inter-class contention resolution).

Mis en forme : Surlignage

Commenté [BMT6]: There are many factors (power consumption, for example) that are not taken into account in current forwarding. I'm not sure I would maintain this mention.

2.1. Context of Internet Traffic Engineering

The context of Internet traffic engineering includes:

1. A network domain context that defines the scope under consideration, and in particular the situations in which the traffic engineering problems occur. The network domain context includes network structure, network policies, network characteristics, network constraints, network quality attributes, and network optimization criteria.

2. A problem context defining the general and concrete issues that traffic engineering addresses. The problem context includes identification, abstraction of relevant features, representation, formulation, specification of the requirements on the solution space, and specification of the desirable features of acceptable solutions.

3. A solution context suggesting how to address the issues identified by the problem context. The solution context includes analysis, evaluation of alternatives, prescription, and resolution.

4. An implementation and operational context in which the solutions are instantiated. The implementation and operational context includes planning, organization, and execution.

The context of Internet traffic engineering and the different problem scenarios are discussed in the following subsections.

2.2. Network Domain Context

IP networks range in size from small clusters of routers situated within a given location, to thousands of interconnected routers, switches, and other components distributed all over the world.

At the most basic level of abstraction, an IP network can be represented as a distributed dynamic system consisting of:

o a set of interconnected resources which provide transport services for IP traffic subject to certain constraints

o a demand system representing the offered load to be transported through the network

o a response system consisting of network processes, protocols, and related mechanisms which facilitate the movement of traffic through the network (see also [AWD2]).

The network elements and resources may have specific characteristics restricting the manner in which the traffic demand is handled. Additionally, network resources may be equipped with traffic control mechanisms managing the way in which the demand is serviced. Traffic control mechanisms may be used to:

o control packet processing activities within a given resource

o arbitrate contention for access to the resource by different packets

o regulate traffic behavior through the resource.

A configuration management and provisioning system may allow the settings of the traffic control mechanisms to be manipulated by external or internal entities in order to exercise control over the way in which the network elements respond to internal and external stimuli.

The details of how the network carries packets are specified in the policies of the network administrators and are installed through network configuration management and policy based provisioning systems. Generally, the types of service provided by the network also depend upon the technology and characteristics of the network elements and protocols, the prevailing service and utility models, and the ability of the network administrators to translate policies into network configurations.

Internet networks have three significant characteristics:

- they provide real-time services

- they are mission critical

- their operating environments are very dynamic.

Commenté [BMT7]: I would keep the last bullet, which is BTW elaborated in the paragraph right after.

Not all networks provide the second bullet, for example.

The dynamic characteristics of IP and IP/MPLS networks can be attributed in part to fluctuations in demand, to the interaction between various network protocols and processes, to the rapid evolution of the infrastructure which demands the constant inclusion of new technologies and new network elements, and to transient and persistent faults which occur within the system.

Packets contend for the use of network resources as they are conveyed through the network. A network resource is considered to be congested if, for an interval of time, the arrival rate of packets exceed the output capacity of the resource. Congestion may result in some of the arriving packets being delayed or even dropped.

Congestion increases transit delay, delay variation, may lead to packet loss, and reduces the predictability of network services. Clearly, congestion is highly undesirable. Combating congestion at a reasonable cost is a major objective of Internet traffic engineering.

Efficient sharing of network resources by multiple traffic ~~streams~~ flows is a basic operational premise for the Internet. A fundamental challenge in network operation is to increase resource utilization while minimizing the possibility of congestion.

The Internet has to function in the presence of different classes of traffic with different service requirements. RFC 2475 provides an architecture for Differentiated Services (Diffserv) and makes this requirement clear [RFC2475]. The RFC allows packets to be grouped into behavior aggregates such that each aggregate has a common set of

Mis en forme : Surlignage

behavioral characteristics or a common set of delivery requirements. Delivery requirements of a specific set of packets may be specified explicitly or implicitly. Two of the most important traffic delivery requirements are: ~~capacity constraints and QoS constraints~~.

- Capacity constraints can be expressed statistically as peak rates, mean rates, burst sizes, or as some deterministic notion of effective bandwidth.

Mis en forme : Avec puces + Niveau : 1 + Alignement : 0,66 cm + Retrait : 1,3 cm

Commenté [BMT8]: In order to distinguish capacity from QoS metrics.

- QoS requirements can be expressed in terms of:

Mis en forme : Avec puces + Niveau : 1 + Alignement : 0,66 cm + Retrait : 1,3 cm

o integrity constraints such as packet loss

o temporal constraints such as timing restrictions for the delivery of each packet (delay) and timing restrictions for the delivery of consecutive packets belonging to the same traffic stream (delay variation).

2.3. Problem Context

There are several ~~large~~ problems associated with operating a network described in the previous section. This section analyzes the problem context in relation to traffic engineering. The identification, abstraction, representation, and measurement of network features relevant to traffic engineering are significant issues.

A particular challenge is to formulate the problems that traffic engineering attempts to solve. For example:

o how to identify the requirements on the solution space

o how to specify the desirable features of solutions

o how to actually solve the problems

o how to measure and characterize the effectiveness of solutions.

Another class of problems is how to measure and estimate relevant network state parameters. Effective traffic engineering relies on a good estimate of the offered traffic load as well as a view of the underlying topology and associated resource constraints. A network-wide view of the topology is also a must for offline planning.

Still another class of problem is how to characterize the state of the network and how to evaluate its performance. The performance evaluation problem is two-fold: one aspect relates to the evaluation of the system-level performance of the network; the other aspect relates to the evaluation of resource-level performance, which restricts attention to the performance analysis of individual network resources.

In this document, we refer to the system-level characteristics of the network as the "macro-states" and the resource-level characteristics as the "micro-states." The system-level characteristics are also known as the emergent properties of the network. Correspondingly, we refer to the traffic engineering schemes dealing with network performance optimization at the systems level as "macro-TE" and the schemes that optimize at the individual resource level as "micro-TE." Under certain circumstances, the system-level performance can be derived from the resource-level performance using appropriate rules of composition, depending upon the particular performance measures of interest.

Another fundamental class of problem concerns how to effectively optimize network performance. Performance optimization may entail translating solutions for specific traffic engineering problems into network configurations. Optimization may also entail some degree of resource management control, routing control, and capacity augmentation.

2.3.1. Congestion and its Ramifications

Congestion is one of the most significant problems in an operational IP context. A network element is said to be congested if it experiences sustained overload over an interval of time. Congestion almost always results in degradation of service quality to end users. Congestion control schemes can include demand-side policies and supply-side policies. Demand-side policies may restrict access to congested resources or dynamically regulate the demand to alleviate the overload situation. Supply-side policies may expand or augment network capacity to better accommodate offered traffic. Supply-side policies may also re-allocate network resources by redistributing traffic over the infrastructure. Traffic redistribution and resource re-allocation serve to increase the 'effective capacity' of the network.

The emphasis of this document is primarily on congestion management schemes falling within the scope of the network, rather than on congestion management systems dependent upon sensitivity and adaptivity from end-systems. That is, the aspects that are considered in this document with respect to congestion management are those solutions that can be provided by control entities operating on the network and by the actions of network administrators and network operations systems.

2.4. Solution Context

The solution context for Internet traffic engineering involves analysis, evaluation of alternatives, and choice between alternative courses of action. Generally the solution context is based on making 'reasonable' inferences about the current or future state of the network, and making decisions that may involve a preference between alternative sets of action. More specifically, the solution context demands reasonable estimates of traffic workload, characterization of network state, derivation of solutions which may be implicitly or explicitly formulated, and possibly instantiating a set of control actions. Control actions may involve the manipulation of parameters associated with routing, control over tactical capacity acquisition, and control over the traffic management functions.

The following list of instruments may be applicable to the solution context of Internet traffic engineering~~.~~:

- A set of policies, objectives, and requirements (which may be context dependent) for network performance evaluation and performance optimization.

- A collection of online and possibly offline tools and mechanisms for measurement, characterization, modeling, and control traffic, and control over the placement and allocation of network resources, as well as control over the mapping or distribution of traffic onto the infrastructure.

- A set of constraints on the operating environment, the network protocols, and the traffic engineering system itself.

- A set of quantitative and qualitative techniques and methodologies for abstracting, formulating, and solving traffic engineering problems.

- A set of administrative control parameters which may be manipulated through a c~~C~~onfiguration ~~Management ~~management~~(CM)~~ system. ~~The CM ~~Such a system itself may include a configuration control subsystem, a configuration repository, a configuration accounting subsystem, and a configuration auditing subsystem.

- A set of guidelines for network performance evaluation, performance optimization, and performance improvement.

Determining traffic characteristics through measurement or estimation is very useful within the realm of the traffic engineering solution space. Traffic estimates can be derived from customer subscription information, traffic projections, traffic models, and from actual measurements. The measurements may be performed at different levels, e.g., at the traffic-aggregate level or at the flow level. Measurements at the flow level or on small traffic aggregates may be performed at edge nodes, when traffic enters and leaves the network. Measurements for large traffic-aggregates may be performed within the core of the network.

To conduct performance studies and to support planning of existing and future networks, a routing analysis may be performed to determine the paths the routing protocols will choose for various traffic demands, and to ascertain the utilization of network resources as traffic is routed through the network. Routing analysis captures the selection of paths through the network, the assignment of traffic across multiple feasible routes, and the multiplexing of IP traffic over traffic trunks (if such constructs exist) and over the underlying network infrastructure. A model of network topology is necessary to perform routing analysis. A network topology model may be extracted from:

- network architecture documents
- network designs
- information contained in router configuration files
- routing databases
- routing tables
- automated tools that discover and collate network topology information.

Topology information may also be derived from servers that monitor network state, and from servers that perform provisioning functions.

Routing in operational IP networks can be administratively controlled at various levels of abstraction including the manipulation of BGP attributes and interior gateway protocol (IGP) metrics. For path oriented technologies such as MPLS, routing can be further controlled by the manipulation of relevant traffic engineering parameters, resource parameters, and administrative policy constraints. Within the context of MPLS, the path of an explicitly routed label switched path (LSP) can be computed and established in various ways including:

- manually
- automatically, online using constraint-based routing processes implemented on label switching routers

- automatically, offline using constraint-based routing entities implemented on external traffic engineering support systems.

2.4.1. Combating the Congestion Problem

Minimizing congestion is a significant aspect of Internet traffic engineering. This subsection gives an overview of the general approaches that have been used or proposed to combat congestion.

Congestion management policies can be categorized based upon the following criteria (see [YARE95] for a more detailed taxonomy of congestion control schemes):

1. Congestion ~~Management~~ management based on ~~Response~~ response ~~Time~~ time ~~Scales~~scales:

    * Long (weeks to months): Expanding network capacity by adding new equipment, routers, and links takes time and is comparatively costly. Capacity planning needs to take this into consideration. Network capacity is expanded based on estimates or forecasts of future traffic development and traffic distribution. These upgrades are typically carried out over weeks or months, or maybe even years.

    * Medium (minutes to days): Several control policies fall within the medium timescale category. Examples include:

        a. Adjusting routing protocol parameters to route traffic away or towards certain segments of the network.

        b. Setting up or adjusting explicitly routed LSPs in MPLS networks to route traffic trunks away from possibly congested resources or toward possibly more favorable routes.

        c. Re-configuring the logical topology of the network to make it correlate more closely with the spatial traffic distribution using, for example, an underlying path-oriented technology such as MPLS LSPs or optical channel trails.

      Many of these adaptive schemes rely on measurement systems. A measurement system monitors changes in traffic distribution, traffic loads, and network resource utilization and then provides feedback to the online or offline traffic engineering mechanisms and tools so that they can trigger control actions within the network. The traffic engineering mechanisms and tools can be implemented in a distributed or centralized fashion. A centralized scheme may have global visibility into the network state and may produce more optimal solutions. However, centralized schemes are prone to single points of failure and may not scale as well as distributed schemes. Moreover, the information utilized by a centralized scheme may be stale and might not reflect the actual state of the network. It is not an objective of this document to make a recommendation between distributed and centralized schemes: that is a choice that network administrators must make based on their specific needs.

* Short (picoseconds to minutes): This category includes packet level processing functions and events that are recorded on the order of several round trip times. It also includes router mechanisms such as passive and active buffer management. All of these mechanisms are used to control congestion or signal congestion to end systems so that they can adaptively regulate the rate at which traffic is injected into the network. One of the most popular active queue management schemes, especially for TCP traffic, is Random Early Detection (RED) [FLJA93]. During congestion (but before the queue is filled), the RED scheme chooses arriving packets to "mark" according to a probabilistic algorithm which takes into account the average queue size. A router that does not utilize explicit congestion notification (ECN) [FLOY94] can simply drop marked packets to alleviate congestion and implicitly notify the receiver about the congestion. On the other hand, if the router supports ECN, it can set the ECN field in the packet header. Several variations of RED have been proposed to support different drop precedence levels in multi-class environments [RFC2597]. RED provides congestion avoidance which is not worse than traditional Tail-Drop (TD) queue management (drop arriving packets only when the queue is full). Importantly, RED reduces the possibility of global synchronization where retransmission burst become synchronized across the whole network, and improves fairness among different TCP sessions. However, RED by itself cannot prevent congestion and unfairness caused by sources unresponsive to RED, e.g., UDP traffic and some misbehaved greedy connections. Other schemes have been proposed to improve the performance and fairness in the presence of unresponsive traffic. Some of those schemes (such as Longest Queue Drop (LQD) and Dynamic Soft Partitioning with Random Drop (RND) [SLDC98]) were proposed as theoretical frameworks and are typically not available in existing commercial products.

2. ~~Congestion Management: ~~Reactive ~~Versus ~~vs. ~~Preventive ~~preventive ~~Schemes~~ congestion management schemes

- ~~Reactive: ~~Reactive (recovery) congestion management policies: react to existing congestion problems. All the policies described above for the long and medium time scales can be categorized as being reactive. They are based on monitoring and identifying congestion problems that exist in the network, and on the initiation of relevant actions to ease a situation.

- ~~Preventive: ~~Preventive (predictive/avoidance) policies: take proactive action to prevent congestion based on estimates and predictions (e.g., traffic matrix forecast) of future congestion problems. Some of the policies described for the long and medium time scales fall into this category. Preventive policies do not necessarily respond immediately to existing congestion problems. Instead, forecasts of traffic demand and workload distribution are considered, and action may be taken to prevent potential future congestion problems. The schemes described for the short time scale can also be used for congestion avoidance because dropping or marking packets before queues actually overflow would trigger corresponding TCP sources to slow down.

3. ~~Congestion Management: ~~Supply-~~Side ~~side ~~Versus ~~vs. ~~Demand~~demand-~~Side ~~side Congestion management s~~S~~chemes

- ~~Supply-side: ~~Supply-side congestion management policies: increase the effective capacity available to traffic in order to control or reduce congestion. This can be accomplished by increasing capacity or by balancing distribution of traffic over the network. Capacity planning aims to provide a physical topology and associated link bandwidths that match or exceed estimated traffic workload and traffic distribution subject to traffic forecasts and budgetary or other constraints. If the actual traffic distribution does not fit the topology derived from capacity panning, then the traffic can be mapped onto the topology by using routing control mechanisms, by applying path oriented technologies (e.g., MPLS LSPs and optical channel trails) to modify the logical topology, or by employing some other load redistribution mechanisms.

- ~~Demand-side: ~~Demand-side congestion management policies: control or regulate the offered traffic to alleviate congestion problems. For example, some of the short time scale mechanisms described earlier as well as policing and rate-shaping mechanisms attempt to regulate the offered load in various ways.

2.5. Implementation and Operational Context

The operational context of Internet traffic engineering is characterized by constant changes that occur at multiple levels of abstraction. The implementation context demands effective planning, organization, and execution. The planning aspects may involve determining prior sets of actions to achieve desired objectives. Organizing involves arranging and assigning responsibility to the various components of the traffic engineering system and coordinating the activities to accomplish the desired TE objectives. Execution involves measuring and applying corrective or perfective actions to attain and maintain desired TE goals.

3. Traffic Engineering Process Models

This section describes a generic process model that captures the high-level practical aspects of Internet traffic engineering in an operational context. The process model is described as a sequence of actions that must be carried out to optimize the performance of an operational network (see also [RFC2702], [AWD2]). This process model may be enacted explicitly or implicitly, by a software process or by a human.

The traffic engineering process model is iterative [AWD2]. The four phases of the process model described below are repeated as a continual sequence.

- Define the relevant control policies that govern the operation of the network.

- Acquire measurement data from the operational network.

- Analyze the network state and characterize the traffic workload. Proactive analysis identifies potential problems that could manifest in the future. Reactive analysis identifies existing problems and determines their causes.

- Optimize the performance of the network. This involves a decision process which selects and implements a set of actions from a set of alternatives given the results of the three previous steps. Optimization actions may include the use of techniques to control the offered traffic and to control the distribution of traffic across the network.

3.1. Components of the Traffic Engineering Process Model

The key components of the traffic engineering process model are as follows.

1. Measurement is crucial to the traffic engineering function. The operational state of a network can only be conclusively determined through measurement. Measurement is also critical to the optimization function because it provides feedback data which is used by traffic engineering control subsystems. This data is used to adaptively optimize network performance in response to events and stimuli originating within and outside the network. Measurement in support of the TE function can occur at different levels of abstraction. For example, measurement can be used to derive packet level characteristics, flow level characteristics, user or customer level characteristics, traffic aggregate characteristics, component level characteristics, and network wide characteristics.

2. Modeling, analysis, and simulation are important aspects of Internet traffic engineering. Modeling involves constructing an abstract or physical representation which depicts relevant traffic characteristics and network attributes. A network model is an abstract representation of the network which captures relevant network features, attributes, and characteristic. Network simulation tools are extremely useful for traffic engineering. Because of the complexity of realistic quantitative analysis of network behavior, certain aspects of network performance studies can only be conducted effectively using simulation.

3. Network performance optimization involves resolving network issues by transforming such issues into concepts that enable a solution, identification of a solution, and implementation of the solution. Network performance optimization can be corrective or perfective. In corrective optimization, the goal is to remedy a problem that has occurred or that is incipient. In perfective optimization, the goal is to improve network performance even when explicit problems do not exist and are not anticipated.

4. Review of TE Techniques

Commenté [BMT9]: It may be useful to indicate for each solution which/whether all TE key elements are provided (policy, path steering, resource management)

This section briefly reviews different traffic engineering approaches proposed and implemented in telecommunications and computer networks using IETF protocols and architectures. The discussion is not intended to be comprehensive. It is primarily intended to illuminate existing approaches to traffic engineering in the Internet. A historic overview of traffic engineering in telecommunications networks is provided in Appendix A, while Appendix B describes approaches in other standards bodies.

4.1. Overview of IETF Projects Related to Traffic Engineering

This subsection reviews a number of IETF activities pertinent to Internet traffic engineering.

4.1.1. Constraint-Based Routing

Constraint-based routing refers to a class of routing systems that compute routes through a network subject to the satisfaction of a set of constraints and requirements. In the most general case, constraint-based routing may also seek to optimize overall network performance while minimizing costs.

The constraints and requirements may be imposed by the network itself or by administrative policies. Constraints may include bandwidth, hop count, delay, and policy instruments such as resource class attributes. Constraints may also include domain specific attributes of certain network technologies and contexts which impose restrictions on the solution space of the routing function. Path oriented technologies such as MPLS have made constraint-based routing feasible and attractive in public IP networks.

The concept of constraint-based routing within the context of MPLS traffic engineering requirements in IP networks was first described in [RFC2702] and led to developments such as MPLS-TE [RFC3209] as described in Section 4.1.6.

Unlike QoS-based routing (for example, see [RFC2386] and [MA]) which generally addresses the issue of routing individual traffic flows to satisfy prescribed flow-based QoS requirements subject to network resource availability, constraint-based routing is applicable to traffic aggregates as well as flows and may be subject to a wide variety of constraints which may include policy restrictions.

4.1.1.1. IGP Flexible Algorithms (Flex-Algos)

The traditional approach to routing in an IGP network relies on the IGPs deriving "shortest paths" over the network based solely on the IGP metric assigned to the links. Such an approach is often limited: traffic may tend to converge toward the destination, possibly causing congestion; and it is not possible to steer traffic onto paths depending on the end-to-end qualities demanded by the applications.

To overcome this limitation, various sorts of traffic engineering have been widely deployed (as described in this document), where the

Commenté [BMT10]: Some of the listed mechanisms are TE ones, others can be used by TE mechanisms, while some of them will rely on TE.

I would restructure this section among these lines or similar ones. The logic for listing these techniques is not clear to me.

Commenté [BMT11]: A more recent example can be found here: draft-ietf-idr-performance-routing

TE component is responsible for computing the path based on ~~additionalcmetrics~~ additional metrics and/or constraints. Such paths (or tunnels) need to be installed in the routers' forwarding tables in addition to, or as a replacement for the original paths computed by IGPs. The main drawback of these TE approaches is the additional complexity of protocols and management, and the state that may need to be maintained within the network.

IGP flexible algorithms (flex-algos) [I-D.ietf-lsr-flex-algo] allow IGPs to construct constraint-based paths over the network by computing constraint- based next hops. The intent of flex-algos is to reduce TE complexity by letting an IGP perform some basic TE computation capabilities. Flex-algo includes a set of extensions to the IGPs that enable a router to send TLVs that:

o describe a set of constraints on the topology

o identify calculation-type

o describe a metric-type that is to be used to compute the best paths through the constrained topology.

A given combination of calculation-type, metric-type, and constraints is known as a "Flexible Algorithm Definition" (or FAD). A router that sends such a set of TLVs also assigns a specific identifier (the Flexible Algorithm) to the specified combination of calculation-type, metric-type, and constraints.

There are two use cases for flex-algo: in IP networks [I-D.ietf-lsr-ip-flexalgo] and in segment routing networks [I-D.ietf-lsr-flex-algo]. In the first case, flex-algo computes paths to an IPv4 or IPv6 address, in the second case, flex-algo computes paths to a prefix SID (see Section 4.1.16).

There are many use cases where flex-algo can bring big value, such as:

o Expansion of functionality of IP Performance metrics [RFC5664] when points of interest could instantiate specific constraint-based routing (flex-algo) based on the measurement results.

o Nested usage of flex-algo and TE extensions for IGP (see Section 4.1.11) when we can form 'underlay' by means of flex-algo and 'overlay' by TE.

o Flex-algo in SR-MPLS (Section 4.1.16) is a base use case when we can easily benefit from TE-like topology that will be built without external TE component on routers or PCE (see Section 4.1.13).

- Building of network slices [I-D.nsdt-teas-ietf-network-slice-definition] where particular IETF network slice SLO can be guaranteed by flex-algo.

4.1.2. Integrated Services

The IETF developed the Integrated Services (Intserv) model that requires resources, such as bandwidth and buffers, to be reserved a priori for a given traffic flow to ensure that the quality of service requested by the traffic flow is satisfied. The Integrated Services model includes additional components beyond those used in the best-effort model such as packet classifiers, packet schedulers, and admission control. A packet classifier is used to identify flows that are to receive a certain level of service. A packet scheduler handles the scheduling of service to different packet flows to ensure that QoS commitments are met. Admission control is used to determine whether a router has the necessary resources to accept a new flow.

The main issue with the Integrated Services model has been scalability [RFC2998], especially in large public IP networks which may potentially have millions of active micro-flows in transit concurrently.

A notable feature of the Integrated Services model is that it requires explicit signaling of QoS requirements from end systems to routers [RFC2753]. The Resource Reservation Protocol (RSVP) performs this signaling function and is a critical component of the Integrated Services model. RSVP is described in Section 4.1.3.

4.1.3. RSVP

RSVP is a soft state signaling protocol [RFC2205]. It supports receiver initiated establishment of resource reservations for both multicast and unicast flows. RSVP was originally developed as a signaling protocol within the Integrated Services framework (see Section 4.1.2) for applications to communicate QoS requirements to the network and for the network to reserve relevant resources to satisfy the QoS requirements [RFC2205].

In RSVP, the traffic sender or source node sends a PATH message to the traffic receiver with the same source and destination addresses as the traffic which the sender will generate. The PATH message contains: (1) a sender traffic specification describing the characteristics of the traffic, (2) a sender template specifying the format of the traffic, and (3) an optional advertisement specification which is used to support the concept of One Pass With Advertising (OPWA) [RFC2205]. Every intermediate router along the path forwards the PATH message to the next hop determined by the routing protocol. Upon receiving a PATH message, the receiver responds with a RESV message which includes a flow descriptor used to request resource reservations. The RESV message travels to the sender or source node in the opposite direction along the path that the PATH message traversed. Every intermediate router along the path can reject or accept the reservation request of the RESV message. If the request is rejected, the rejecting router will send an error message to the receiver and the signaling process will terminate. If the request is accepted, link bandwidth and buffer space are allocated for the flow and the related flow state information is installed in the router.

One of the issues with the original RSVP specification was Scalability. This is because reservations were required for micro-flows, so that the amount of state maintained by network elements tends to increase linearly with the number of micro-flows. These issues are described in [RFC2961] which also modifies and extends RSVP to mitigate the scaling problems to make RSVP a versatile signaling protocol for the Internet. For example, RSVP has been extended to reserve resources for aggregation of flows, to set up MPLS explicit label switched paths (see Section 4.1.6), and to perform other signaling functions within the Internet. [RFC2961] also describes a mechanism to reduce the amount of Refresh messages required to maintain established RSVP sessions.

4.1.4. Differentiated Services

Commenté [BMT12]: This is more a QoS mechanism and not a TE as per the definition provided in previous sections, but I understand this section is about mechanisms that are useful for TE.

I was expecting to see RFC4124, rather than the base diffserv arch.

The goal of Differentiated Services (Diffserv) within the IETF was to devise scalable mechanisms for categorization of traffic into behavior aggregates, which ultimately allows each behavior aggregate to be treated differently, especially when there is a shortage of resources such as link bandwidth and buffer space [RFC2475]. One of the primary motivations for Diffserv was to devise alternative mechanisms for service differentiation in the Internet that mitigate the scalability issues encountered with the Intserv model.

Diffserv uses the Differentiated Services field in the IP header (the DS field) consisting of six bits in what was formerly known as the Type of Service (TOS) octet. The DS field is used to indicate the forwarding treatment that a packet should receive at a transit node [RFC2474]. Diffserv includes the concept of Per-Hop Behavior (PHB) groups. Using the PHBs, several classes of services can be defined using different classification, policing, shaping, and scheduling rules.

For an end-user of network services to utilize Differentiated Services provided by its Internet Service Provider (ISP), it may be necessary for the user to have an SLA with the ISP. An SLA may explicitly or implicitly specify a Traffic Conditioning Agreement (TCA) which defines classifier rules as well as metering, marking, discarding, and shaping rules.

Packets are classified, and possibly policed and shaped at the ingress to a Diffserv network. When a packet traverses the boundary between different Diffserv domains, the DS field of the packet may be re-marked according to existing agreements between the domains.

Differentiated Services allows only a finite number of service classes to be specified by the DS field. The main advantage of the Diffserv approach relative to the Intserv model is scalability. Resources are allocated on a per-class basis and the amount of state information is proportional to the number of classes rather than to the number of application flows.

The Diffserv model deals with traffic management issues on a per hop basis. The Diffserv control model consists of a collection of micro-TE control mechanisms. Other traffic engineering capabilities, such as capacity management (including routing control), are also required in order to deliver acceptable service quality in Diffserv networks. The concept of Per Domain Behaviors has been introduced to better capture the notion of Differentiated Services across a complete domain [RFC3086].

4.1.5. QUIC

Commenté [BMT13]: I'm not to understand the reasoning for listing QUIC not TCP or SCTP. QUIC does not support path steering or exchange of policies. QUIC support a mechanism for path migration, but does not support multi-path nor scheduling mechanisms among multiple paths.

There out there some transport mechanisms that may even be tagged as TE as they offer traffic steering, policy, and resource management. I'm referring to MPTCP (RFC 8684) or RFC8803.

More background on how transport protocols are used to provide traffic steering/etc., please refer to: https://tools.ietf.org/html/draft-bonaventure-quic-atsss-overview-00

QUIC [I-D.ietf-quic-transport] is a UDP-based multiplexed and secure transport protocol. QUIC provides applications with flow-controlled streams for structured communication, low-latency connection establishment, and network path migration.

QUIC is a connection-oriented protocol that creates a stateful interaction between a client and server. QUIC uses a handshake procedure that combines negotiation of cryptographic and transport parameters. This is a key differentiation from other transport protocols.

Endpoints communicate in QUIC by exchanging QUIC packets that use a customized framing for protection. Most QUIC packets contain frames, which carry control information and application data between endpoints. QUIC authenticates all packets and encrypts as much as is practical. QUIC packets are carried in UDP datagrams to better facilitate deployment within existing systems and networks.

Application protocols exchange information over a QUIC connection via streams, which are ordered sequences of bytes. Two types of stream can be created: bidirectional streams, which allow both endpoints to send data; and unidirectional streams, which allow a single endpoint to send data. A credit-based scheme is used to limit stream creation and to bound the amount of data that can be sent.

QUIC provides the necessary feedback to implement reliable delivery and congestion control to avoid network congestion.

4.1.6. Multiprotocol Label Switching (MPLS)

MPLS is an advanced forwarding scheme which also includes extensions to conventional IP control plane protocols. MPLS extends the Internet routing model and enhances packet forwarding and path control [RFC3031].

At the ingress to an MPLS domain, Label Switching Routers (LSRs) classify IP packets into Forwarding Equivalence Classes (FECs) based on a variety of factors, including, e.g., a combination of the information carried in the IP header of the packets and the local routing information maintained by the LSRs. An MPLS label stack entry is then prepended to each packet according to their forwarding equivalence classes. The MPLS label stack entry is 32 bits long and contains a 20-bit label field.

An LSR makes forwarding decisions by using the label prepended to packets as the index into a local next hop label forwarding entry (NHLFE). The packet is then processed as specified in the NHLFE. The incoming label may be replaced by an outgoing label (label swap), and the packet may be forwarded to the next LSR. Before a packet leaves an MPLS domain, its MPLS label may be removed (label pop). A Label Switched Path (LSP) is the path between an ingress LSRs and an egress LSRs through which a labeled packet traverses. The path of an explicit LSP is defined at the originating (ingress) node of the LSP. MPLS can use a signaling protocol such as RSVP or LDP to set up LSPs.

MPLS is a very powerful technology for Internet traffic engineering because it supports explicit LSPs which allow constraint-based routing to be implemented efficiently in IP networks [AWD2]. The requirements for traffic engineering over MPLS are described in [RFC2702]. Extensions to RSVP to support instantiation of explicit LSP are discussed in [RFC3209].

4.1.7. Generalized MPLS (GMPLS)

GMPLS extends MPLS control protocols to encompass time-division (e.g., Synchronous Optical Network / Synchronous Digital Hierarchy (SONET/SDH), Plesiochronous Digital Hierarchy (PDH), Optical Transport Network (OTN)), wavelength (lambdas), and spatial switching (e.g., incoming port or fiber to outgoing port or fiber) as well as continuing to support packet switching. GMPLS provides a common set of control protocols for all of these layers (including some technology-specific extensions) each of which has a diverse data or forwarding plane. GMPLS covers both the signaling and the routing part of that control plane and is based on the Traffic Engineering extensions to MPLS (see Section 4.1.6).

In GMPLS, the original MPLS architecture is extended to include LSRs whose forwarding planes rely on circuit switching, and therefore cannot forward data based on the information carried in either packet or cell headers. Specifically, such LSRs include devices where the switching is based on time slots, wavelengths, or physical ports. These additions impact basic LSP properties: how labels are requested and communicated, the unidirectional nature of MPLS LSPs, how errors are propagated, and information provided for synchronizing the ingress and egress LSRs.

4.1.8. IP Performance Metrics

The IETF IP Performance Metrics (IPPM) working group has developed a set of standard metrics that can be used to monitor the quality, performance, and reliability of Internet services. These metrics can be applied by network operators, end-users, and independent testing groups to provide users and service providers with a common understanding of the performance and reliability of the Internet component 'clouds' they use/provide [RFC2330]. The criteria for performance metrics developed by the IPPM working group are described in [RFC2330]. Examples of performance metrics include one-way packet loss [RFC7680], one-way delay [RFC7679], and connectivity measures between two nodes [RFC2678]. Other metrics include second-order measures of packet loss and delay.

Some of the performance metrics specified by the IPPM working group are useful for specifying SLAs. SLAs are sets of service level objectives negotiated between users and service providers, wherein each objective is a combination of one or more performance metrics, possibly subject to certain constraints.

4.1.9. Flow Measurement

The IETF Real Time Flow Measurement (RTFM) working group produced an architecture that defines a method to specify traffic flows as well as a number of components for flow measurement (meters, meter readers, manager) [RFC2722]. A flow measurement system enables network traffic flows to be measured and analyzed at the flow level for a variety of purposes. As noted in RFC 2722, a flow measurement system can be very useful in the following contexts:

o understanding the behavior of existing networks

o planning for network development and expansion

o quantification of network performance

o verifying the quality of network service

o attribution of network usage to users.

A flow measurement system consists of meters, meter readers, and managers. A meter observes packets passing through a measurement point, classifies them into groups, accumulates usage data (such as the number of packets and bytes for each group), and stores the usage data in a flow table. A group may represent any collection of user applications, hosts, networks, etc. A meter reader gathers usage data from various meters so it can be made available for analysis. A manager is responsible for configuring and controlling meters and meter readers. The instructions received by a meter from a manager include flow specifications, meter control parameters, and sampling techniques. The instructions received by a meter reader from a manager include the address of the meter whose date is to be collected, the frequency of data collection, and the types of flows to be collected.

4.1.10. Endpoint Congestion Management

[RFC3124] provides a set of congestion control mechanisms for the use of transport protocols. It is also allows the development of mechanisms for unifying congestion control across a subset of an endpoint's active unicast connections (called a congestion group). A congestion manager continuously monitors the state of the path for each congestion group under its control. The manager uses that information to instruct a scheduler on how to partition bandwidth among the connections of that congestion group.

4.1.11. TE Extensions to the IGPs

[RFC5305] describes the extensions to the Intermediate System to Intermediate System (IS-IS) protocol to support TE, similarly [RFC3630] specifies TE extensions for OSPFv2 ([RFC5329] has the same description for OSPFv3).

The idea of redistribution TE extensions such as link type and ID, local and remote IP addresses, TE metric, maximum bandwidth, maximum reservable bandwidth and unreserved bandwidth, admin group in IGP is a common for both IS-IS and OSPF.

The difference is in the details of their transmission: IS-IS uses the Extended IS Reachability TLV (type 22) and Sub-TLVs for those TE parameters, OSPFv2 uses Opaque LSA [RFC5250] type 10 (OSPFv3 uses Intra-Area-TE-LSA) with two top-level TLV (Router Address and Link) also with Sub-TLVs for that purpose.

IS-IS also uses the Extended IP Reachability TLV (type 135, which have the new 32 bit metric) and the TE Router ID TLV (type 134). Those Sub-TLV details are described in [RFC8570] for IS-IS and in [RFC7471] for OSPFv2 ([RFC5329] for OSPFv3).

4.1.12. Link-State BGP

In a number of environments, a component external to a network is called upon to perform computations based on the network topology and current state of the connections within the network, including traffic engineering information. This is information typically distributed by IGP routing protocols within the network (see Section 4.1.11.

The Border Gateway Protocol (BGP) ~~Section 7~~ is one of the essential routing protocols that glue the Internet together. BGP Link State (BGP-LS) [RFC7752] is a mechanism by which link-state and traffic engineering information can be collected from networks and shared with external components using the BGP routing protocol. The mechanism is applicable to physical and virtual IGP links, and is subject to policy control.

Information collected by BGP-LS can be used to construct the Traffic Engineering Database (TED, see Section 4.1.20) for use by the Path Computation Element (PCE, see Section 4.1.13), or may be used by Application-Layer Traffic Optimization (ALTO) servers (see Section 4.1.15).

4.1.13. Path Computation Element

Constraint-based path computation is a fundamental building block for traffic engineering in MPLS and GMPLS networks. Path computation in large, multi-domain networks is complex and may require special computational components and cooperation between the elements in different domains. The Path Computation Element (PCE) [RFC4655] is an entity (component, application, or network node) that is capable of computing a network path or route based on a network graph and applying computational constraints.

Thus, a PCE can provide a central component in a traffic engineering system operating on the Traffic Engineering Database (TED, see Section 4.1.20) with delegated responsibility for determining paths in MPLS, GMPLS, or Segment Routing networks. The PCE uses the Path Computation Element Communication Protocol (PCEP) [RFC5440] to communicate with Path Computation Clients (PCCs), such as MPLS LSRs, to answer their requests for computed paths or to instruct them to initiate new paths [RFC8281] and maintain state about paths already installed in the network [RFC8231].

PCEs form key components of a number of traffic engineering systems. More information about the applicability of PCE can be found in [RFC8051], while [RFC6805] describes the application of PCE to determining paths across multiple domains. PCE also has potential use in Abstraction and Control of TE Networks (ACTN) (see Section 4.1.17), Centralized Network Control [RFC8283], and Software Defined Networking (SDN) (see Section 5.3.2).

4.1.14. Multi-Layer Traffic Engineering

Commenté [BMT14]: Should this be listed separately as many of other sections are not TE mechanisms.

Networks are often arranged as layers. A layer relationship may represent the interaction between technologies (for example, an IP network operated over an optical network), or the relationship between different network operators (for example, a customer network operated over a service provider's network). Note that a multi-layer network does not imply the use of multiple technologies, although some form of encapsulation is often applied.

Multi-layer traffic engineering presents a number of challenges associated with scalability and confidentiality. These issues are addressed in [RFC7926] which discusses the sharing of information between domains through policy filters, aggregation, abstraction, and virtualization. That document also discusses how existing protocols can support this scenario with special reference to BGP-LS (see Section 4.1.12).

PCE (see Section 4.1.13) is also a useful tool for multi-layer networks as described in [RFC6805] and [RFC8685]. Signaling techniques for multi-layer traffic engineering are described in [RFC6107].

See also Appendix A.3.1 for a discussion of how the overlay model has been important in the development of traffic engineering.

4.1.15. Application-Layer Traffic Optimization

This document describes various TE mechanisms available in the network. However, distributed applications in general and, in particular, bandwidth-greedy P2P applications that are used, for example, for file sharing, cannot directly use those techniques. As per [RFC5693], applications could greatly improve traffic distribution and quality by cooperating with external services that are aware of the network topology. Addressing the Application-Layer Traffic Optimization (ALTO) problem means, on the one hand, deploying an ALTO service to provide applications with information regarding the underlying network (e.g., basic network location structure and preferences of network paths) and, on the other hand, enhancing applications in order to use such information to perform better-than-random selection of the endpoints with which they establish connections.

The basic function of ALTO is based on abstract maps of a network. These maps provide a simplified view, yet enough information about a network for applications to effectively utilize them. Additional services are built on top of the maps. [RFC7285] describes a protocol implementing the ALTO services as an information-publishing interface that allows a network to publish its network information such as network locations, costs between them at configurable granularities, and end-host properties to network applications. The information published by the ALTO Protocol should benefit both the network and the applications. The ALTO Protocol uses a REST-ful design and encodes its requests and responses using JSON [RFC8259] with a modular design by dividing ALTO information publication into multiple ALTO services (e.g., the Map service, the Map-Filtering Service, the Endpoint Property Service, and the Endpoint Cost Service).

[RFC8189] defines a new service that allows an ALTO Client to retrieve several cost metrics in a single request for an ALTO filtered cost map and endpoint cost map. [RFC8896] extends the ALTO cost information service so that applications decide not only 'where' to connect, but also 'when'. This is useful for applications that need to perform bulk data transfer and would like to schedule these transfers during an off-peak hour, for example.

[I-D.ietf-alto-performance-metrics] introducing network performance metrics, including network delay, jitter, packet loss rate, hop count, and bandwidth. The ALTO server may derive and aggregate such performance metrics from BGP-LS (see Section 4.1.12) or IGP-TE (see Section 4.1.11), or management tools, and then expose the information to allow applications to determine 'where' to connect based on network performance criteria. ALTO WG is evaluating the use of network TE properties while making application decisions for new use-cases such as Edge computing and Datacenter interconnect.

4.1.16. Segment Routing with MPLS Encapsulation (SR-MPLS)

Segment Routing (SR) [RFC8402] leverages the source routing and tunneling paradigms. The path a packet takes is defined at the ingress and the packet is tunneled to the egress. A node steers a packet through a controlled set of instructions, called segments, by prepending the packet with an SR header: a label stack in MPLS case.

A segment can represent any instruction, topological or service-based, thanks to the MPLS architecture [RFC3031]. Labels can be looked up in a global context (platform wide) as well as in some other context (see "context labels" in Section 3 of [RFC5331]).

4.1.16.1. Base Segment Routing Identifier Types

Commenté [BMT15]: I'm afraid this going into more details that are not required for this document. I would avoid having subsections for one specific mechanism.

Segments are identified by Segment Identifiers (SIDs). There are four types of SID that are relevant for traffic engineering.

Prefix SID: Uses the SR Global Block (SRGB), must be unique within the routing domain SRGB, and is advertised by an IGP. The Prefix-SID can be configured as an absolute value or an index.

Node SID: A Prefix SID with the 'N' (node) bit set. It is associated with a host prefix (/32 or /128) that identifies the node. More than 1 Node SID can be configured per node.

Adjacency SID: Locally significant by default, an Adjacency SID can be made globally significant through use of the 'L' flag. It identifies a unidirectional adjacency. In most implementations Adjacency SIDs are automatically allocated for each adjacency. They are always encoded as an absolute (not indexed) value.

Binding SID: A Binding SID has two purposes:

1. Mapping Server in ISIS

Commenté [BMT16]: ??

   The SID/Label Binding TLV is used to advertise the mappings of prefixes to SIDs/Labels. This functionality is called the Segment Routing Mapping Server (SRMS). The behavior of the SRMS is defined in [RFC8661]

2. Cross-connect (label to FEC mapping)

This is fundamental for multi-domain/multi-layer operation. The Binding SID identifies a new path available at the anchor point. It is always local to the originator, must not be present at the top of the stack, and must be looked up in the context of the Node SID. It could be provisioned through Network Configuration Protocol (NETCONF) [RFC6241] or the RESTCONF Protocol [RFC8040]~~Netconf/Restconf~~, PCEP, BGP, or the CLI.

4.1.16.2. Segment Routing Policy

SR Policy [I-D.ietf-spring-segment-routing-policy] is an evolution of Segment Routing to enhance the TE capabilities. It is a framework that enables instantiation of an ordered list of segments on a node for implementing a source routing policy with a specific intent for traffic steering from that node.

An SR Policy is identified through the tuple <headend, color, endpoint>. The headend is the IP address of the node where the policy is instantiated. The endpoint is the IP address of the destination of the policy. The color is an index that associates the SR Policy with an intent (e.g., low-latency).

The headend node is notified of SR Policies and associated SR paths via configuration or by a extensions to protocols such as PCEP [RFC8664] or BGP [I-D.ietf-idr-segment-routing-te-policy]. Each SR path consists of a Segment-List (an SR source-routed path), and the headend uses the endpoint and color parameters to classify packets to match the SR policy and so determine along which path to forward them. If an SR Policy is associated with a set of SR paths, each is associated with a weight for weighted load balancing. Furthermore, multiple SR Policies may be associated with a set of SR paths to allow multiple traffic flows to be placed on the same paths.

An SR Binding SID (BSID) are also be associated with each candidate path associated with an SR Policy, or with the SR Policy itself. The headend node installs a BSID-keyed entry in the forwarding plane and assigns it the action of steering packets that match the entry to the selected path of the SR Policy. This steering can be done in various ways:

o SID Steering: Incoming packets have an active SID matching a local BSID at the headend.

- Per-destination Steering: Incoming packets match a BGP/Service route which indicates an SR Policy.

- Per-flow Steering: Incoming packets match a forwarding array (for example, the classic 5-tuple) which indicates an SR Policies.

- Policy-based Steering: Incoming packets match a routing policy which directs them to an SR Policy.

4.1.17. Network Virtualization and Abstraction

One of the main drivers for Software Defined Networking (SDN) [RFC7149] is a decoupling of the network control plane from the data plane. This separation has been achieved for TE networks with the development of MPLS/GMPLS (see Section 4.1.6 and Section 4.1.7) and the Path Computation Element (PCE) (Section 4.1.13). One of the advantages of SDN is its logically centralized control regime that allows a global view of the underlying networks. Centralized control in SDN helps improve network resource utilization compared with distributed network control.

Abstraction and Control of TE Networks (ACTN) [RFC8453] defines a hierarchical SDN architecture which describes the functional entities and methods for the coordination of resources across multiple domains, to provide end-to-end traffic engineered services. ACTN facilitates end-to-end connections and provides them to the user. ACTN is focused on:

- Abstraction of the underlying network resources and how they are provided to higher-layer applications and customers.

- Virtualization of underlying resources for use by the customer, application, or service. The creation of a virtualized environment allows operators to view and control multi-domain networks as a single virtualized network.

- Presentation to customers of networks as a virtual network via open and programmable interfaces.

The ACTN managed infrastructure is built from traffic engineered network resources, which may include statistical packet bandwidth, physical forwarding plane sources (such as wavelengths and time slots), forwarding and cross-connect capabilities. The type of network virtualization seen in ACTN allows customers and applications (tenants) to utilize and independently control allocated virtual network resources as if resources as if they were physically their own resource. The ACTN network is "sliced", with tenants being given a different partial and abstracted topology view of the physical underlying network.

4.1.18. Network Slicing

Commenté [BMT17]: Slicing will rely upon TE techniques. Not sure why this one is listed.

An IETF Network Slice is a logical network topology connecting a number of endpoints using a set of shared or dedicated network resources [I-D.nsdt-teas-ietf-network-slice-definition]. The resources are used to satisfy specific Service Level Objectives (SLOs) specified by the consumer.

IETF Network Slices are created and managed within the scope of one or more network technologies (e.g., IP, MPLS, optical). They are intended to enable a diverse set of applications that have different requirements to coexist on the same network infrastructure. IETF Network Slices are defined such that they are independent of the underlying infrastructure connectivity and technologies used. This is to allow an IETF Network Slice consumer to describe their network connectivity and relevant objectives in a common format, independent of the underlying technologies used.

An IETF Network Slice is a well-defined composite of a set of endpoints, the connectivity requirements between subsets of these endpoints, and associated service requirements. The service requirements are expressed in terms of quantifiable characteristics or service level objectives (SLOs). SLOs along with terms Service Level Indicator (SLI) and Service Level Agreement (SLA) are used to define the performance of a service at different levels [I-D.nsdt-teas-ietf-network-slice-definition].

~~The concept of an IETF network slice is consistent with an enhanced VPN (VPN+) [I-D.ietf-teas-enhanced-vpn]. That is, from a consumer's perspective it looks like a VPN connectivity matrix with additional information about the level of service required between endpoints, while from an operator's perspective it looks like a set of routing or tunneling instructions with the network resource reservations necessary to provide the required service levels as specified by the SLOs.~~

IETF network slices are not, of themselves, TE constructs. However, a network operator that offers IETF network slices is likely to use many TE tools in order to manage their network and provide the services.

Commenté [BMT18]: Great.

Other services may also rely on TE but are not listed in the document. I would personally remove it.

4.1.19. Deterministic Networking

Deterministic Networking (DetNet) [RFC8655] is an architecture for applications with critical timing and reliability requirements. The layered architecture particularly focuses on developing DetNet service capabilities in the data plane [RFC8938]. The DetNet service sub-layer provides a set of Packet Replication, Elimination, and Ordering Functions (PREOF) functions to provide end-to-end service assurance. The DetNet forwarding sub-layer provides corresponding forwarding assurance (low packet loss, bounded latency, and in-order delivery) functions using resource allocations and explicit route mechanisms.

The separation into two sub-layers allows a greater flexibility to adapt Detnet capability over a number of TE data plane mechanisms such as IP, MPLS, and Segment Routing. More importantly it interconnects IEEE 802.1 Time Sensitive Networking (TSN) [I-D.ietf-detnet-ip-over-tsn] deployed in Industry Control and Automation Systems (ICAS).

DetNet can be seen as a specialized branch of TE, since it sets up explicit optimized paths with allocation of resources as requested. A DetNet application can express its QoS attributes or traffic behavior using any combination of DetNet functions described in sub-layers. They are then distributed and provisioned using well-established control and provisioning mechanisms adopted for traffic-engineering.

In DetNet, a considerable state information is required to maintain per flow queuing disciplines and resource reservation for a large number of individual flows. This can be quite challenging for network operations during network events such as faults, change in traffic volume or re-provisioning. Therefore, DetNet recommends support for aggregated flows, however, it still requires large amount of control signaling to establish and maintain DetNet flows.

4.1.20. Network TE State Definition and Presentation

The network states that are relevant to the traffic engineering need to be stored in the system and presented to the user. The Traffic Engineering Database (TED) is a collection of all TE information about all TE nodes and TE links in the network, which is an essential component of a TE system, such as MPLS-TE [RFC2702] and GMPLS [RFC3945]. In order to formally define the data in the TED and to present the data to the user with high usability, the data modeling language YANG [RFC7950] can be used as described in [RFC8795].

4.1.21. System Management and Control Interfaces

The traffic engineering control system needs to have a management interface that is human-friendly and a control interfaces that is programmable for automation. The Network Configuration Protocol (NETCONF) [RFC6241] or the RESTCONF Protocol [RFC8040] provide programmable interfaces that are also human-friendly. These protocols use XML or JSON encoded messages. When message compactness or protocol bandwidth consumption needs to be optimized for the control interface, other protocols, such as Group Communication for the Constrained Application Protocol (CoAP) [RFC7390] or gRPC, are available, especially when the protocol messages are encoded in a binary format. Along with any of these protocols, the data modeling language YANG [RFC7950] can be used to formally and precisely define the interface data.

The Path Computation Element Communication Protocol (PCEP) [RFC5440] is another protocol that has evolved to be an option for the TE system control interface. The messages of PCEP are TLV-based, not defined by a data modeling language such as YANG.

4.2. Content Distribution

The Internet is dominated by client-server interactions, principally Web traffic although in the future, more sophisticated media servers may become dominant. The location and performance of major information servers has a significant impact on the traffic patterns within the Internet as well as on the perception of service quality by end users.

A number of dynamic load balancing techniques have been devised to improve the performance of replicated information servers. These techniques can cause spatial traffic characteristics to become more dynamic in the Internet because information servers can be dynamically picked based upon the location of the clients, the location of the servers, the relative utilization of the servers, the relative performance of different networks, and the relative performance of different parts of a network. This process of assignment of distributed servers to clients is called traffic directing. It is an application layer function.

Traffic directing schemes that allocate servers in multiple geographically dispersed locations to clients may require empirical network performance statistics to make more effective decisions. In the future, network measurement systems may need to provide this type of information.

When congestion exists in the network, traffic directing and traffic engineering systems should act in a coordinated manner. This topic is for further study.

The issues related to location and replication of information servers, particularly web servers, are important for Internet traffic engineering because these servers contribute a substantial proportion of Internet traffic.

5. Taxonomy of Traffic Engineering Systems

Commenté [BMT19]: I would put this one before the previous section

This section presents a short taxonomy of traffic engineering systems constructed based on traffic engineering styles and views as listed below and described in greater detail in the following subsections of this document.

- Time-dependent versus State-dependent versus Event-dependent
- Offline versus Online
- Centralized versus Distributed
- Local versus Global Information
- Prescriptive versus Descriptive
- Open Loop versus Closed Loop
- Tactical versus Strategic

5.1. Time-Dependent Versus State-Dependent Versus Event-Dependent

Traffic engineering methodologies can be classified as time-dependent, state-dependent, or event-dependent. All TE schemes are considered to be dynamic in this document. Static TE implies that no traffic engineering methodology or algorithm is being applied - it is a feature of network planning, but lacks the reactive and flexible nature of traffic engineering.

In time-dependent TE, historical information based on periodic variations in traffic (such as time of day) is used to pre-program routing and other TE control mechanisms. Additionally, customer subscription or traffic projection may be used. Pre-programmed routing plans typically change on a relatively long time scale (e.g., daily). Time-dependent algorithms do not attempt to adapt to short-term variations in traffic or changing network conditions. An example of a time-dependent algorithm is a global centralized optimizer where the input to the system is a traffic matrix and multi-class QoS requirements as described [MR99]. Another example of such a methodology is the application of data mining to Internet traffic [AJ19] which enables the use of various machine learning algorithms to identify patterns within historically collected datasets about Internet traffic, and to extract information in order to guide decision-making, and to improve efficiency and productivity of operational processes.

State-dependent TE adapts the routing plans based on the current state of the network which provides additional information on variations in actual traffic (i.e., perturbations from regular variations) that could not be predicted using historical information. Constraint-based routing is an example of state-dependent TE operating in a relatively long time scale. An example operating in a relatively short timescale is a load-balancing algorithm described in [MATE]. The state of the network can be based on parameters flooded by the routers. Another approach is for a particular router performing adaptive TE to send probe packets along a path to gather the state of that path. [RFC6374] defines protocol extensions to collect performance measurements from MPLS networks. Another approach is for a management system to gather the relevant information directly from network elements using telemetry data collection "publication/subscription" techniques [RFC7923]. Timely gathering and distribution of state information is critical for adaptive TE. While time-dependent algorithms are suitable for predictable traffic variations, state-dependent algorithms may be applied to increase network efficiency and resilience to adapt to the prevailing network state.

Event-dependent TE methods can also be used for TE path selection. Event-dependent TE methods are distinct from time-dependent and state-dependent TE methods in the manner in which paths are selected. These algorithms are adaptive and distributed in nature and typically use learning models to find good paths for TE in a network. While state-dependent TE models typically use available-link-bandwidth (ALB) flooding for TE path selection, event-dependent TE methods do not require ALB flooding. Rather, event-dependent TE methods typically search out capacity by learning models, as in the success-to-the-top (STT) method. ALB flooding can be resource intensive, since it requires link bandwidth to carry LSAs, processor capacity to process LSAs, and the overhead can limit area/Autonomous System (AS) size. Modeling results suggest that event-dependent TE methods could lead to a reduction in ALB flooding overhead without loss of network throughput performance [I-D.ietf-tewg-qos-routing].

5.2. Offline Versus Online

Traffic engineering requires the computation of routing plans. The computation may be performed offline or online. The computation can be done offline for scenarios where routing plans need not be executed in real-time. For example, routing plans computed from forecast information may be computed offline. Typically, offline computation is also used to perform extensive searches on multi-dimensional solution spaces.

Online computation is required when the routing plans must adapt to changing network conditions as in state-dependent algorithms. Unlike offline computation (which can be computationally demanding), online computation is geared toward relative simple and fast calculations to select routes, fine-tune the allocations of resources, and perform load balancing.

5.3. Centralized Versus Distributed

Under centralized control there is a central authority which determines routing plans and perhaps other TE control parameters on behalf of each router. The central authority periodically collects network-state information from all routers, and sends routing information to the routers. The update cycle for information exchange in both directions is a critical parameter directly impacting the performance of the network being controlled. Centralized control may need high processing power and high bandwidth control channels.

Distributed control determines route selection by each router autonomously based on the router's view of the state of the network. The network state information may be obtained by the router using a probing method or distributed by other routers on a periodic basis using link state advertisements. Network state information may also be disseminated under exception conditions. Examples of protocol extensions used to advertise network link state information are defined in [RFC5305], [RFC6119], [RFC7471], [RFC8570], and [RFC8571]. See also Section 4.1.11.

5.3.1. Hybrid Systems

In practice, most TE systems will be a hybrid of central and distributed control. For example, a popular MPLS approach to TE is to use a central controller based on an active, stateful PCE, but to use routing and signaling protocols to make local decisions at routers within the network. Local decisions may be able to respond more quickly to network events, but may result in conflicts with decisions made by other routers.

Network operations for TE systems may also use a hybrid of offline and online computation. TE paths may be precomputed based on stable-state network information and planned traffic demands, but may then be modified in the active network depending on variations in network state and traffic load. Furthermore, responses to network events may be precomputed offline to allow rapid reactions without further computation, or may be derived online depending on the nature of the events.

Lastly, note that a fully functional TE system is likely to use all aspects of time-dependent, state-dependent, and event-dependent methodologies as described in Section 5.1.

5.3.2. Considerations for Software Defined Networking

As discussed in Section 4.1.17, one of the main drivers for SDN is a decoupling of the network control plane from the data plane [RFC7149]. However, SDN may also combine centralized control of resources, and facilitate application-to-network interaction via an application programming interface (API) such as [RFC8040]. Combining these features provides a flexible network architecture that can adapt to network requirements of a variety of higher-layer applications, a concept often referred to as the "programmable network" [RFC7426].

The centralized control aspect of SDN helps improve global network resource utilization compared with distributed network control, where local policy may often override global optimization goals. In an SDN environment, the data plane forwards traffic to its desired destination. However, before traffic reaches the data plane, the logically centralized SDN control plane often determines the end-to-end path the application traffic will take in the network. Therefore, the SDN control plane needs to be aware of the underlying network topology, capabilities and current node and link resource state.

Using a PCE-based SDN control framework [RFC7491], the available network topology may be discovered by running a passive instance of OSPF or IS-IS, or via BGP-LS [RFC7752], to generate a TED (see Section 4.1.20). The PCE is used to compute a path (see Section 4.1.13) based on the TED and available bandwidth, and further path optimization may be based on requested objective functions [RFC5541]. When a suitable path has been computed the programming of the explicit network path may be performed using either end-to-end signaling protocol [RFC3209] or per-hop with each node being directly programmed [RFC8283] by the SDN controller.

By utilizing a centralized approach to network control, additional network benefits are also available, including Global Concurrent Optimization (GCO) [RFC5557]. A GCO path computation request will simultaneously use the network topology and set of new end-to-end path requests, along with their respective constraints, for optimal placement in the network. Correspondingly, a GCO-based computation may be applied to recompute existing network paths to groom traffic and to mitigate congestion.

5.4. Local Versus Global

Traffic engineering algorithms may require local and global network-state information.

Local information is the state of a portion of the domain. Examples include the bandwidth and packet loss rate of a particular path, or the state and capabilities of a network link. Local state information may be sufficient for certain instances of distributed control TE.

Global information is the state of the entire TE domain. Examples include a global traffic matrix, and loading information on each link throughout the domain of interest. Global state information is typically required with centralized control. Distributed TE systems may also need global information in some cases.

5.5. Prescriptive Versus Descriptive

TE systems may also be classified as prescriptive or descriptive.

Prescriptive traffic engineering evaluates alternatives and recommends a course of action. Prescriptive traffic engineering can be further categorized as either corrective or perfective. Corrective TE prescribes a course of action to address an existing or predicted anomaly. Perfective TE prescribes a course of action to evolve and improve network performance even when no anomalies are evident.

Descriptive traffic engineering, on the other hand, characterizes the state of the network and assesses the impact of various policies without recommending any particular course of action.

5.5.1. Intent-Based Networking

Commenté [BMT20]: I would remove this section as this is just one way to express a service request. Furthermore, an intent may be less expressive in terms of constraints and guidelines.

Intent is defined in [I-D.irtf-nmrg-ibn-concepts-definitions] as a set of operational goals (that a network should meet) and outcomes (that a network is supposed to deliver), defined in a declarative manner without specifying how to achieve or implement them. This definition is based on [RFC7575] where, in the context of Autonomic Networks, it is described as "an abstract, high-level policy used to operate a network."

Thus, intent-based management or Intent-Based Networking (IBN) is the concept of operating a network based on the concept of intent.

Intent-Based Networking aims to produce networks that are simpler to manage and operate, requiring only minimal intervention. Networks have no way of automatically knowing operational goals nor which instances of networking services to support, thus the operator's intent needs to be communicated to the network.

More specifically, intent is a declaration of operational goals that a network should meet and outcomes that the network is supposed to deliver, without specifying how to achieve them. Those goals and outcomes are defined in a purely declarative way: they specify what to accomplish, not how to achieve it. Intent applies two concepts:

o It provides data abstraction: users and operators do not need to be concerned with low-level device configuration.

o It provides functional abstraction: users and operators do not need to be concerned with how to achieve a given intent. What is specified is the desired outcome which is converted by the management system into the actions that will achieve the outcome.

Intent-Based Networking is applicable to traffic engineering because many of the high-level objectives may be expressed as "intent." For example, load balancing, delivery of services, and robustness against failures. The intent is converted by the management system into traffic engineering actions within the network.

5.6. Open-Loop Versus Closed-Loop

Open-loop traffic engineering control is where control action does not use feedback information from the current network state. The control action may use its own local information for accounting purposes, however.

Closed-loop traffic engineering control is where control action utilizes feedback information from the network state. The feedback information may be in the form of historical information or current measurement.

5.7. Tactical versus Strategic

Tactical traffic engineering aims to address specific performance problems (such as hot-spots) that occur in the network from a tactical perspective, without consideration of overall strategic imperatives. Without proper planning and insights, tactical TE tends to be ad hoc in nature.

Strategic traffic engineering approaches the TE problem from a more organized and systematic perspective, taking into consideration the immediate and longer term consequences of specific policies and actions.

6. Recommendations for Internet Traffic Engineering

This section describes high-level recommendations for traffic engineering in the Internet in general terms.

The recommendations describe the capabilities needed to solve a traffic engineering problem or to achieve a traffic engineering objective. Broadly speaking, these recommendations can be categorized as either functional or non-functional recommendations.

- Functional recommendations describe the functions that a traffic engineering system should perform. These functions are needed to realize traffic engineering objectives by addressing traffic engineering problems.

- Non-functional recommendations relate to the quality attributes or state characteristics of a traffic engineering system. These recommendations may contain conflicting assertions and may sometimes be difficult to quantify precisely.

6.1. Generic Non-functional Recommendations

The generic non-functional recommendations for Internet traffic engineering are listed in the paragraphs that follow. In a given context, some of these recommendations may be critical while others may be optional. Therefore, prioritization may be required during the development phase of a traffic engineering system to tailor it to a specific operational context.

Usability: Usability is a human aspect of traffic engineering systems. It refers to the ease with which a traffic engineering system can be deployed and operated. In general, it is desirable to have a TE system that can be readily deployed in an existing network. It is also desirable to have a TE system that is easy to operate and maintain.

Automation: Whenever feasible, a TE system should automate as many TE functions as possible to minimize the amount of human effort needed to analyze and control operational networks. Automation is particularly important in large-scale public networks because of the high cost of the human aspects of network operations and the high risk of network problems caused by human errors. Automation may entail the incorporation of automatic feedback and intelligence into some components of the TE system.

Scalability: Public networks continue to grow rapidly with respect to network size and traffic volume. Therefore, to remain applicable as the network evolves, a TE system should be scalable. In particular, a TE system should remain functional as the network expands with regard to the number of routers and links, and with respect to the traffic volume. A TE system should have a scalable architecture, should not adversely impair other functions and processes in a network element, and should not consume too many network resources when collecting and distributing state information, or when exerting control.

Stability: Stability is a very important consideration in TE systems that respond to changes in the state of the network. State-dependent TE methodologies typically include a trade-off between responsiveness and stability. It is strongly recommended that when a trade-off between responsiveness and stability is needed, it should be made in favor of stability (especially in public IP backbone networks).

Flexibility: A TE system should allow for changes in optimization policy. In particular, a TE system should provide sufficient configuration options so that a network administrator can tailor the system to a particular environment. It may also be desirable to have both online and offline TE subsystems which can be independently enabled and disabled. TE systems that are used in multi-class networks should also have options to support class based performance evaluation and optimization.

Visibility: Mechanisms should exist as part of the TE system to collect statistics from the network and to analyze these statistics to determine how well the network is functioning. Derived statistics such as traffic matrices, link utilization, latency, packet loss, and other performance measures of interest which are determined from network measurements can be used as indicators of prevailing network conditions. The capabilities of the various components of the routing system are other examples of status information which should be observable.

Simplicity: A TE system should be as simple as possible and easy to use (i.e., have clean, convenient, and intuitive user interfaces). Simplicity in user interface does not necessarily imply that the TE system will use naive algorithms. When complex algorithms and internal structures are used, the user interface should hide such complexities from the network administrator as much as possible.

Commenté [BMT21]: This is redundant with the usability requirement.

Interoperability: Whenever feasible, TE systems and their components should be developed with open standards-based interfaces to allow interoperation with other systems and components.

Security: Security is a critical consideration in TE systems. Such systems typically exert control over functional aspects of the network to achieve the desired performance objectives. Therefore, adequate measures must be taken to safeguard the integrity of the TE system. Adequate measures must also be taken to protect the network from vulnerabilities that originate from security breaches and other impairments within the TE system.

The remaining subsections of this section focus on some of the high-level functional recommendations for traffic engineering.

6.2. Routing Recommendations

Routing control is a significant aspect of Internet traffic engineering. Routing impacts many of the key performance measures associated with networks, such as throughput, delay, and utilization. Generally, it is very difficult to provide good service quality in a wide area network without effective routing control. A desirable TE routing system is one that takes traffic characteristics and network constraints into account during route selection while maintaining stability.

Shortest path first (SPF) IGPs are based on shortest path algorithms and have limited control capabilities for TE [RFC2702], [AWD2]. These limitations include:

1. Pure SPF protocols do not take network constraints and traffic characteristics into account during route selection. For example, IGPs always select the shortest paths based on link metrics assigned by administrators) so load sharing cannot be performed across paths of different costs. Using shortest paths to forward traffic may cause the following problems:

   * If traffic from a source to a destination exceeds the capacity of a link along the shortest path, the link (and hence the shortest path) becomes congested while a longer path between these two nodes may be under-utilized

- The shortest paths from different sources can overlap at some links. If the total traffic from the sources exceeds the capacity of any of these links, congestion will occur.

- Problems can also occur because traffic demand changes over time, but network topology and routing configuration cannot be changed as rapidly. This causes the network topology and routing configuration to become sub-optimal over time, which may result in persistent congestion problems.

2. The Equal-Cost Multi-Path (ECMP) capability of SPF IGPs supports sharing of traffic among equal cost paths between two nodes. However, ECMP attempts to divide the traffic as equally as possible among the equal cost shortest paths. Generally, ECMP does not support configurable load sharing ratios among equal cost paths. The result is that one of the paths may carry significantly more traffic than other paths because it may also carry traffic from other sources. This situation can result in congestion along the path that carries more traffic. Weighted ECMP (WECMP) (see, for example, [I-D.ietf-bess-evpn-unequal-lb]) provides some mitigation.

3. Modifying IGP metrics to control traffic routing tends to have network-wide effects. Consequently, undesirable and unanticipated traffic shifts can be triggered as a result. Work described in Section 8 may be capable of better control [FT00], [FT01].

Because of these limitations, new capabilities are needed to enhance the routing function in IP networks. Some of these capabilities are summarized below.

- Constraint-based routing computes routes to fulfill requirements subject to constraints. This can be useful in public IP backbones with complex topologies. Constraints may include bandwidth, hop count, delay, and administrative policy instruments such as resource class attributes [RFC2702], [RFC2386]. This makes it possible to select routes that satisfy a given set of requirements. Routes computed by constraint-based routing are not necessarily the shortest paths. Constraint-based routing works best with path-oriented technologies that support explicit routing, such as MPLS.

  Constraint-based routing can also be used as a way to distribute traffic onto the infrastructure, including for best effort traffic. For example, congestion problems caused by uneven traffic distribution may be avoided or reduced by knowing the reservable bandwidth attributes of the network links and by specifying the bandwidth requirements for path selection.

- A number of enhancements to the link state IGPs are needed to allow them to distribute additional state information required for constraint-based routing. The extensions to OSPF are described in [RFC3630], and to IS-IS in [RFC5305]. Some of the additional topology state information includes link attributes such as reservable bandwidth and link resource class attribute (an administratively specified property of the link). The resource class attribute concept is defined in [RFC2702]. The additional topology state information is carried in new TLVs and sub-TLVs in IS-IS, or in the Opaque LSA in OSPF [RFC5305], [RFC3630].

  An enhanced link-state IGP may flood information more frequently than a normal IGP. This is because even without changes in topology, changes in reservable bandwidth or link affinity can trigger the enhanced IGP to initiate flooding. A trade-off between the timeliness of the information flooded and the flooding frequency is typically implemented using a threshold based on the percentage change of the advertised resources to avoid excessive consumption of link bandwidth and computational resources, and to avoid instability in the TED.

- In a TE system, it is also desirable for the routing subsystem to make the load splitting ratio among multiple paths (with equal cost or different cost) configurable. This capability gives network administrators more flexibility in the control of traffic distribution across the network. It can be very useful for avoiding/relieving congestion in certain situations. Examples can be found in [XIAO] and [I-D.ietf-bess-evpn-unequal-lb].

- The routing system should also have the capability to control the routes of subsets of traffic without affecting the routes of other traffic if sufficient resources exist for this purpose. This capability allows a more refined control over the distribution of traffic across the network. For example, the ability to move traffic away from its original path to another path (without affecting other traffic paths) allows the traffic to be moved from resource-poor network segments to resource-rich segments. Path oriented technologies such as MPLS-TE inherently support this capability as discussed in [AWD2].

- Additionally, the routing subsystem should be able to select different paths for different classes of traffic (or for different traffic behavior aggregates) if the network supports multiple classes of service (different behavior aggregates).

6.3. Traffic Mapping Recommendations

Commenté [BMT22]: Absent resource-based access control, a TE system may have a suboptimal behavior.

Traffic mapping is the assignment of traffic workload onto (pre-Established) paths to meet certain requirements. Thus, while constraint-based routing deals with path selection, traffic mapping deals with the assignment of traffic to established paths which may have been generated by constraint-based routing or by some other means. Traffic mapping can be performed by time-dependent or state-dependent mechanisms, as described in Section 5.1.

An important aspect of the traffic mapping function is the ability to establish multiple paths between an originating node and a destination node, and the capability to distribute the traffic between the two nodes across the paths according to some policies. A pre-condition for this scheme is the existence of flexible mechanisms to partition traffic and then assign the traffic partitions onto the parallel paths as noted in [RFC2702]. When traffic is assigned to multiple parallel paths, it is recommended that special care should be taken to ensure proper ordering of packets belonging to the same application (or micro-flow) at the destination node of the parallel paths.

Mechanisms that perform the traffic mapping functions should aim to map the traffic onto the network infrastructure to minimize congestion. If the total traffic load cannot be accommodated, or if the routing and mapping functions cannot react fast enough to changing traffic conditions, then a traffic mapping system may use short time scale congestion control mechanisms (such as queue management, scheduling, etc.) to mitigate congestion. Thus, mechanisms that perform the traffic mapping functions complement existing congestion control mechanisms. In an operational network, traffic should be mapped onto the infrastructure such that intra-class and inter-class resource contention are minimized (see Section 2).

When traffic mapping techniques that depend on dynamic state feedback (e.g., MATE [MATE] and such like) are used, special care must be taken to guarantee network stability.

6.4. Measurement Recommendations

The importance of measurement in traffic engineering has been discussed throughout this document. A TE system should include mechanisms to measure and collect statistics from the network to support the TE function. Additional capabilities may be needed to help in the analysis of the statistics. The actions of these mechanisms should not adversely affect the accuracy and integrity of the statistics collected. The mechanisms for statistical data acquisition should also be able to scale as the network evolves.

Traffic statistics may be classified according to long-term or short-term timescales. Long-term traffic statistics are very useful for traffic engineering. Long-term traffic statistics may periodicity record network workload (such as hourly, daily, and weekly variations in traffic profiles) as well as traffic trends. Aspects of the traffic statistics may also describe class of service characteristics for a network supporting multiple classes of service. Analysis of the long-term traffic statistics may yield other information such as busy hour characteristics, traffic growth patterns, persistent congestion problems, hot-spot, and imbalances in link utilization caused by routing anomalies.

A mechanism for constructing traffic matrices for both long-term and short-term traffic statistics should be in place. In multi-service IP networks, the traffic matrices may be constructed for different service classes. Each element of a traffic matrix represents a statistic about the traffic flow between a pair of abstract nodes. An abstract node may represent a router, a collection of routers, or a site in a VPN.

Traffic statistics should provide reasonable and reliable indicators of the current state of the network on the short-term scale. Some short term traffic statistics may reflect link utilization and link congestion status. Examples of congestion indicators include excessive packet delay, packet loss, and high resource utilization. Examples of mechanisms for distributing this kind of information include SNMP, probing tools, FTP, IGP link state advertisements, and ~~Netconf~~NETCONF/~~Restconf~~RESTCONF, etc.

6.5. Network Survivability

Network survivability refers to the capability of a network to maintain service continuity in the presence of faults. This can be accomplished by promptly recovering from network impairments and maintaining the required QoS for existing services after recovery. Survivability is an issue of great concern within the Internet community due to the demand to carry mission critical traffic, real-time traffic, and other high priority traffic over the Internet. Survivability can be addressed at the device level by developing network elements that are more reliable; and at the network level by incorporating redundancy into the architecture, design, and operation of networks. It is recommended that a philosophy of robustness and survivability should be adopted in the architecture, design, and operation of traffic engineering that control IP networks (especially public IP networks). Because different contexts may demand different levels of survivability, the mechanisms developed to support network survivability should be flexible so that they can be tailored to different needs. A number of tools and techniques have been developed to enable network survivability including MPLS Fast Reroute [RFC4090], RSVP-TE Extensions in Support of End-to-End GMPLS Recovery [RFC4872], and GMPLS Segment Recovery [RFC4873].

The impact of service outages varies significantly for different service classes depending on the duration of the outage which can vary from milliseconds (with minor service impact) to seconds (with possible call drops for IP telephony and session time-outs for connection oriented transactions) to minutes and hours (with potentially considerable social and business impact). Different duration outages have different impacts depending on the throughput of the traffic flows that are interrupted.

Failure protection and restoration capabilities are available in multiple layers as network technologies have continued to evolve. Optical networks are capable of providing dynamic ring and mesh restoration functionality at the wavelength level. At the SONET/SDH layer survivability capability is provided with Automatic Protection Switching (APS) as well as self-healing ring and mesh architectures. Similar functionality is provided by layer 2 technologies such as Ethernet.

Rerouting is used at the IP layer to restore service following link and node outages. Rerouting at the IP layer occurs after a period of routing convergence which may require seconds to minutes to complete. Path-oriented technologies such ~~a~~ as MPLS ([RFC3469]) can be used to enhance the survivability of IP networks in a potentially cost effective manner.

An important of multi-layer survivability is that technologies at different layers may provide protection and restoration capabilities at different granularities in terms of time scales and at different bandwidth granularity (from packet-level to wavelength level). Protection and restoration capabilities can also be sensitive to different service classes and different network utility models. Coordinating different protection and restoration capabilities across multiple layers in a cohesive manner to ensure network survivability is maintained at reasonable cost is a challenging task. Protection and restoration coordination across layers may not always be feasible, because networks at different layers may belong to different administrative domains.

The following paragraphs present some of the general recommendations for protection and restoration coordination.

- Protection and restoration capabilities from different layers should be coordinated to provide network survivability in a flexible and cost effective manner. Avoiding duplication of functions in different layers is one way to achieve the coordination. Escalation of alarms and other fault indicators from lower to higher layers may also be performed in a coordinated manner. The order of timing of restoration triggers from different layers is another way to coordinate multi-layer protection/restoration.

- Network capacity reserved in one layer to provide protection and restoration is not available to carry traffic in a higher layer: it is not visible as spare capacity in the higher layer. Placing protection/restoration functions in many layers may increase redundancy and robustness, but it can result in significant inefficiencies in network resource utilization. Careful planning is needed to balance the trade-off between the desire for survivablity and the optimal use of resources.

- It is generally desirable to have protection and restoration schemes that are intrinsically bandwidth efficient.

- Failure notifications throughout the network should be timely and reliable if they are to be acted on as triggers for effective protection and restoration actions.

- Alarms and other fault monitoring and reporting capabilities should be provided at the right network layers so that the protection and restoration actions can be taken in those layers.

6.5.1. Survivability in MPLS Based Networks

Because MPLS is path-oriented, it has the potential to provide faster and more predictable protection and restoration capabilities than conventional hop by hop routed IP systems. Protection types for MPLS networks can be divided into four categories.

- Link Protection: The objective of link protection is to protect an LSP from the failure of a given link. Under link protection, a protection or backup LSP (the secondary LSP) follows a path that is disjoint from the path of the working or operational LSP (the primary LSP) at the particular link where link protection is required. When the protected link fails, traffic on the working LSP is switched to the protection LSP at the head-end of the failed link. As a local repair method, link protection can be fast. This form of protection may be most appropriate in situations where some network elements along a given path are known to be less reliable than others.

- Node Protection: The objective of node protection is to protect an LSP from the failure of a given node. Under node protection, the secondary LSP follows a path that is disjoint from the path of the primary LSP at the particular node where node protection is required. The secondary LSP is also disjoint from the primary LSP at all links attached to the node to be protected. When the protected node fails, traffic on the working LSP is switched over to the protection LSP at the upstream LSR directly connected to the failed node. Node protection covers a slightly larger part of the network compared to link protection, but is otherwise fundamentally the same.

- Path Protection: The goal of LSP path protection (or end-to-end protection) is to protect an LSP from any failure along its routed path. Under path protection, the path of the protection LSP is completely disjoint from the path of the working LSP. The advantage of path protection is that the backup LSP protects the working LSP from all possible link and node failures along the path, except for failures of ingress or egress LSR. Additionally, path protection may be more efficient in terms of resource usage than link or node protection applied at every jop along the path. However, path protection may be slower than link and node protection because the fault notifications have to be propagated further.

- Segment Protection: An MPLS domain may be partitioned into multiple subdomains (protection domains). Path protection is applied to the path of each LSP as it crosses the domain from its ingress to the domain to where it egresses the domain. In cases where an LSP traverses multiple protection domains, a protection mechanism within a domain only needs to protect the segment of the LSP that lies within the domain. Segment protection will generally be faster than end-to-end path protection because recovery generally occurs closer to the fault and the notification doesn't have to propagate as far.

See [RFC3469] and [RFC6372] for a more comprehensive discussion of MPLS based recovery.

6.5.2. Protection Options

Another issue to consider is the concept of protection options. We use notation such as "m:n protection", where m is the number of protection LSPs used to protect n working LSPs. In all cases except 1+1 protection, the resources associated with the protection LSPs can be used to carry preemptable best-effort traffic when the working LSP is functioning correctly.

- 1:1 protection: One working LSP is protected/restored by one protection LSP.

- 1:n protection: One protection LSP is used to protect/restore n working LSPs. Only one failed LSP can be restored at any time.

- n:1 protection: One working LSP is protected/restored by n protection LSPs, possibly with load splitting across the protection LSPs. This may be especially useful when it is not feasible to find one path for the backup that can satisfy the bandwidth requirement of the primary LSP.

- 1+1 protection: Traffic is sent concurrently on both the working LSP and a protection LSP. The egress LSR selects one of the two LSPs based on local policy (usually based on traffic integrity). When a fault disrupts the traffic on one LSP, the egress switches to receive traffic from the other LSP. This approach is expensive in how it consumes network but recovers from failures most rapidly.

6.6. Traffic Engineering in Diffserv Environments

Increasing requirements to support multiple classes of traffic in the Internet, such as best effort and mission critical data, calls for IP networks to differentiate traffic according to some criteria and to give preferential treatment to certain types of traffic. Large numbers of flows can be aggregated into a few behavior aggregates based on some criteria based on common performance requirements in terms of packet loss ratio, delay, and jitter, or in terms of common fields within the IP packet headers.

Differentiated Services (Diffserv) [RFC2475] can be used to ensure that SLAs defined to differentiate between traffic flows are met. Classes of service (CoS) can be supported in a Diffserv environment by concatenating per-hop behaviors (PHBs) along the routing path. A PHB is the forwarding behavior that a packet receives at a Diffserv-compliant node, and it can be configured at each router. PHBs are delivered using buffer management and packet scheduling mechanisms and require that the ingress nodes use traffic classification, marking, policing, and shaping.

Traffic engineering can ~~compliment~~complement Diffserv to improve utilization of network resources. Traffic engineering can be operated on an aggregated basis across all service classes [RFC3270], or on a per service class basis. The former is used to provide better distribution of the traffic load over the network resources (see [RFC3270] for detailed mechanisms to support aggregate traffic engineering). The latter case is discussed below since it is specific to the Diffserv environment, with so called Diffserv-aware traffic engineering [RFC4124].

For some Diffserv networks, it may be desirable to control the performance of some service classes by enforcing relationships between the traffic workload contributed by each service class and the amount of network resources allocated or provisioned for that service class. Such relationships between demand and resource allocation can be enforced using a combination of, for example:

o TE mechanisms on a per service class basis that enforce the relationship between the amount of traffic contributed by a given service class and the resources allocated to that class.

o Mechanisms that dynamically adjust the resources allocated to a given service class to relate to the amount of traffic contributed by that service class.

It may also be desirable to limit the performance impact of high priority traffic on relatively low priority traffic. This can be achieved, for example, by controlling the percentage of high priority traffic that is routed through a given link. Another way to accomplish this is to increase link capacities appropriately so that lower priority traffic can still enjoy adequate service quality. When the ratio of traffic workload contributed by different service classes varies significantly from router to router, it may not be enough to rely on conventional IGP routing protocols or on TE mechanisms that are not sensitive to different service classes. Instead, it may be desirable to perform traffic engineering, especially routing control and mapping functions, on a per service class basis. One way to accomplish this in a domain that supports both MPLS and Diffserv is to define class specific LSPs and to map traffic from each class onto one or more LSPs that correspond to that service class. An LSP corresponding to a given service class can then be routed and protected/restored in a class dependent manner, according to specific policies.

Performing traffic engineering on a per class basis may require per-class parameters to be distributed. It is common to have some classes share some aggregate constraints (e.g., maximum bandwidth requirement) without enforcing the constraint on each individual class. These classes can be grouped into class-types, and per-class-type parameters can be distributed to improve scalability. This also allows better bandwidth sharing between classes in the same class-type. A class-type is a set of classes that satisfy the following two conditions:

- o Classes in the same class-type have common aggregate requirements to satisfy required performance levels.

- o There is no requirement to be enforced at the level of an individual class in the class-type. Note that it is, nevertheless, still possible to implement some priority policies for classes in the same class-type to permit preferential access to the class-type bandwidth through the use of preemption priorities.

See [RFC4124] for detailed requirements on Diffserv-aware traffic engineering.

6.7. Network Controllability

Offline and online (see Section 5.2) TE considerations are of limited utility if the network cannot be controlled effectively to implement the results of TE decisions and to achieve the desired network performance objectives.

Capacity augmentation is a coarse-grained solution to TE issues. However, it is simple and may be advantageous if bandwidth is abundant and cheap. However, bandwidth is not always abundant and cheap, and additional capacity might not always be the best solution. Adjustments of administrative weights and other parameters associated with routing protocols provide finer-grained control, but this approach is difficult to use and imprecise because of the ~~the~~ way the routing protocols interact occur across the network.

Control mechanisms can be manual (e.g., static configuration), partially-automated (e.g., scripts), or fully-automated (e.g., policy based management systems). Automated mechanisms are particularly useful in large scale networks. Multi-vendor interoperability can be facilitated by standardized management systems (e.g., YANG models) to support the control functions required to address TE objectives.

Network control functions should be secure, reliable, and stable as these are often needed to operate correctly in times of network impairments (e.g., during network congestion or security attacks).

7. Inter-Domain Considerations

Inter-domain TE is concerned with performance optimization for traffic that originates in one administrative domain and terminates in a different one.

BGP [RFC4271] is the standard exterior gateway protocol used to exchange routing information between autonomous systems (ASes) in the

Internet. BGP includes a sequential decision process that calculates the preference for routes to a given destination network. There are two fundamental aspects to inter-domain TE using BGP:

- Route Redistribution: Controlling the import and export of routes between ASes, and controlling the redistribution of routes between BGP and other protocols within an AS.

- Best path selection: Selecting the best path when there are multiple candidate paths to a given destination network. This is performed by the BGP decision process, selecting preferred exit points out of an AS towards specific destination networks taking a number of different considerations into account. The BGP path selection process can be influenced by manipulating the attributes associated with the process, including NEXT-HOP, WEIGHT, LOCAL-PREFERENCE, AS-PATH, ROUTE-ORIGIN, MULTI-EXIT-DESCRIMINATOR (MED), IGP METRIC, etc.

Route-maps provide the flexibility to implement complex BGP policies based on pre-configured logical conditions. They can be used to control import and export policies for incoming and outgoing routes, control the redistribution of routes between BGP and other protocols, and influence the selection of best paths by manipulating the attributes associated with the BGP decision process. Very complex logical expressions that implement various types of policies can be implemented using a combination of Route-maps, BGP-attributes, Access-lists, and Community attributes.

When considering inter-domain TE with BGP, note that the outbound traffic exit point is controllable, whereas the interconnection point where inbound traffic is received typically is not. Therefore, it is up to each individual network to implement TE strategies that deal with the efficient delivery of outbound traffic from its customers to its peering points. The vast majority of TE policy is based on a "closest exit" strategy, which offloads inter-domain traffic at the nearest outbound peering point towards the destination AS. Most methods of manipulating the point at which inbound traffic enters a are either ineffective, or not accepted in the peering community.

Inter-domain TE with BGP is generally effective, but it is usually applied in a trial-and-error fashion because a TE system usually only has a view of the available network resources within one domain (an AS in this case). A systematic approach for inter-domain TE requires cooperation between the domains. Further, what may be considered a good solution in one domain may not necessarily be a good solution in another. Moreover, it is generally considered inadvisable for one domain to permit a control process from another domain to influence the routing and management of traffic in its network.

MPLS TE-tunnels (LSPs) can add a degree of flexibility in the selection of exit points for inter-domain routing by applying rhe concept of relative and absolute metrics. If BGP attributes are defined such that the BGP decision process depends on IGP metrics to select exit points for inter-domain traffic, then some inter-domain traffic destined to a given peer network can be made to prefer a specific exit point by establishing a TE-tunnel between the router making the selection and the peering point via a TE-tunnel and assigning the TE-tunnel a metric which is smaller than the IGP cost to all other peering points.

Similarly to intra-domain TE, inter-domain TE is best accomplished when a traffic matrix can be derived to depict the volume of traffic from one AS to another.

8. Overview of Contemporary TE Practices in Operational IP Networks

This section provides an overview of some traffic engineering practices in IP networks. The focus is on aspects of control of the routing function in operational contexts. The intent here is to provide an overview of the commonly used practices: the discussion is not intended to be exhaustive.

Service providers apply many of the traffic engineering mechanisms described in this document to optimize the performance of their IP networks. These techniques include capacity planning for long timescales; routing control using IGP metrics and MPLS, as well as path planning and path control using MPLS and Segment Routing for medium timescales; and traffic management mechanisms for short timescale.

Capacity planning is an important component of how a service provider plans an effective IP network. These plans may take the following aspects into account: location of and new links or nodes, existing and predicted traffic patterns, costs, link capacity, topology, routing design, and survivability.

Performance optimization of operational networks is usually an ongoing process in which traffic statistics, performance parameters, and fault indicators are continually collected from the network. This empirical data is analyzed and used to trigger TE mechanisms. Tools that perform what-if analysis can also be used to assist the TE process by reviewing scenarios before a new set of configurations are implemented in the operational network.

Real-time intra-domain TE using the IGP is done by increasing the OSPF or IS-IS metric of a congested link until enough traffic has been diverted away from that link. This approach has some limitations as discussed in Section 6.2. Intra-domain TE approaches ([RR94] [FT00] [FT01] [WANG]) take traffic matrix, network topology, and network performance objectives as input, and produce link metrics and load-sharing ratios. These processes open the possibility for intra-domain TE with IGP to be done in a more systematic way.

Administrators of MPLS-TE networks specify and configure link attributes and resource constraints such as maximum reservable bandwidth and resource class attributes for the links in the domain. A link state IGP that supports TE extensions (IS-IS-TE or OSPF-TE) is used to propagate information about network topology and link attributes to all routers in the domain. Network administrators specify the LSPs that are to originate at each router. For each LSP, the network administrator specifies the destination node and the attributes of the LSP which indicate the requirements that are to be satisfied during the path selection process. The attributes may include and explicit path for the LSP to follow, or originating router uses a local constraint-based routing process to compute the path of the LSP. RSVP-TE is used as a signaling protocol to instantiate the LSPs. By assigning proper bandwidth values to links and LSPs, congestion caused by uneven traffic distribution can be avoided or mitigated.

The bandwidth attributes of an LSP relates to the bandwidth requirements of traffic that flows through the LSP. The traffic attribute of an LSP can be modified to accommodate persistent shifts in demand (traffic growth or reduction). If network congestion occurs due to some unexpected events, existing LSPs can be rerouted to alleviate the situation or network administrator can configure new LSPs to divert some traffic to alternative paths. The reservable bandwidth of the congested links can also be reduced to force some LSPs to be rerouted to other paths. A traffic matrix in an MPLS domain can also be estimated by monitoring the traffic on LSPs. Such traffic statistics can be used for a variety of purposes including network planning and network optimization.

Network management and planning systems have evolved and taken over a lot of the responsibility for determining traffic paths in TE networks. This allows a network-wide view of resources, and facilitates coordination of the use of resources for all traffic flows in the network. Initial solutions using a PCE to perform path computation on behalf of network routers have given way to an approach that follows the SDN architecture. A stateful PCE is able to track all of the LSPs in the network and can redistribute them to make better use of the available resources. Such a PCE can forms part of a network orchestrator that uses PCEP or some other southbound interface to instruct the signaling protocol or directly program the routers.

Segment routing leverages a centralized TE controller and either an MPLS or IPv6 forwarding plane, but does not need to use a signaling protocol or management plane protocol to reserve resources in the routers. All resource reservation is logical within the controller, and not distributed to the routers. Packets are steered through the network using segment routing.

As mentioned in Section 7, there is usually no direct control over the distribution of inbound traffic to a domain. Therefore, the main goal of inter-domain TE is to optimize the distribution of outbound traffic between multiple inter-domain links. When operating a global network, maintaining the ability to operate the network in a regional fashion where desired, while continuing to take advantage of the benefits of a global network, also becomes an important objective.

Inter-domain TE with BGP begins with the placement of multiple peering interconnection points that are in close proximity to traffic sources/destination, and offer lowest cost paths across the network between the peering points and ~~and~~ the sources/destinations. Some location-decision problems that arise in association with inter-domain routing are discussed in [AWD5].

Once the locations of the peering interconnects have been determined and implemented, the network operator decides how best to handle the routes advertised by the peer, as well as how to propagate the peer's routes within their network. One way to engineer outbound traffic flows in a network with many peering interconnects is to create a hierarchy of peers. Generally, the shortest AS paths will be chosen to forward traffic but BGP metrics can be used to prefer some peers and so favor particular paths. Preferred peers are those peers attached through peering interconnects with the most available capacity. Changes may be needed, for example, to deal with a "problem peer" who is difficult to work with on upgrades or is charging high prices for connectivity to their network. In that case, the peer may be given a reduced preference. This type of change can affect a large amount of traffic, and is only used after other methods have failed to provide the desired results.

When there are multiple exit points toward a given peer, and only one of them is congested, it is not necessary to shift traffic away from the peer entirely, but only from the one congested connections. This can be achieved by using passive IGP-metrics, AS-path filtering, or prefix filtering.

9. Security Considerations

This document does not introduce new security issues.

Network security is, of course, an important issue. In general, TE mechanisms are security neutral: they may use tunnels which can slightly help protect traffic from inspection and which, in some cases, can be secured using encryption; they put traffic onto predictable paths within the network that may make it easier to find and attack; they increase the complexity or operation and management of the network; and they enable traffic to be steered onto more secure links or to more secure parts of the network.

The consequences of attacks on the control and management protocols used to operate TE networks can be significant: traffic can be hijacked to pass through specific nodes that perform inspection, or even to be delivered to the wrong place; traffic can be steered onto paths that deliver quality that is below the desired quality; and, networks can be congested or have resources on key links consumed. Thus, it is important to use adequate protection mechanisms on all protocols used to deliver TE.

Certain aspects of a network may be deduced from the details of the TE paths that are used. For example, the link connectivity of the network, and the quality and load on individual links may be assumed from knowing the paths of traffic and the requirements they place on the network (for example, by seeing the control messages or through path- trace techniques). Such knowledge can be used to launch targeted attacks (for example, taking down critical links) or can reveal commercially sensitive information (for example, whether a network is close to capacity). Network operators may, therefore, choose techniques that mask or hide information from within the network.

10. IANA Considerations

This draft makes no requests for IANA action.

11. Acknowledgments

Much of the text in this document is derived from RFC 3272. The authors of this document would like to express their gratitude to all involved in that work. Although the source text has been edited in the production of this document, the original authors should be considered as Contributors to this work. They were:

Daniel O. Awduche
Movaz Networks

Angela Chiu
Celion Networks

Anwar Elwalid
Lucent Technologies

Indra Widjaja
Bell Labs, Lucent Technologies

XiPeng Xiao
Redback Networks

The acknowledgements in RFC3272 were as below. All people who helped in the production of that document also need to be thanked for the carry-over into this new document.

> The authors would like to thank Jim Boyle for inputs on the recommendations section, Francois Le Faucheur for inputs on Diffserv aspects, Blaine Christian for inputs on measurement, Gerald Ash for inputs on routing in telephone networks and for text on event-dependent TE methods, Steven Wright for inputs on network controllability, and Jonathan Aufderheide for inputs on inter-domain TE with BGP. Special thanks to Randy Bush for proposing the TE taxonomy based on "tactical versus strategic" methods. The subsection describing an "Overview of ITU Activities Related to Traffic Engineering" was adapted from a contribution by Waisum Lai. Useful feedback and pointers to relevant materials were provided by J. Noel Chiappa. Additional comments were provided by Glenn Grotefeld during the working last call process. Finally, the authors would like to thank Ed Kern, the TEWG co-chair, for his comments and support.

The early versions of this document were produced by the TEAS Working Group's RFC3272bis Design Team. The full list of members of this team is:

Acee Lindem
Adrian Farrel
Aijun Wang
Daniele Ceccarelli
Dieter Beller
Jeff Tantsura
Julien Meuric
Liu Hua
Loa Andersson
Luis Miguel Contreras
Martin Horneffer
Tarek Saad
Xufeng Liu

The production of this document includes a fix to the original text resulting from an Errata Report by Jean-Michel Grimaldi.

The author of this document would also like to thank Dhurv Dhody for review comments.

12. Contributors

The following people contributed substantive text to this document:

Gert Grammel
EMail: ggrammel@juniper.net

Loa Andersson
EMail: loa@pi.nu

Xufeng Liu
EMail: xufeng.liu.ietf@gmail.com

Lou Berger
EMail: lberger@labn.net

Jeff Tantsura
EMail: jefftant.ietf@gmail.com

Daniel King
EMail: daniel@olddog.co.uk

Boris Hassanov
EMail: bhassanov@yandex-team.ru

Kiran Makhijani
Email: kiranm@futurewei.com

Dhruv Dhody
Email: dhruv.ietf@gmail.com

13. Informative References

[AJ19] Adekitan, A., Abolade, J., and O. Shobayo, "Data mining approach for predicting the daily Internet data traffic of a smart university", Article Journal of Big Data, 2019, Volume 6, Number 1, Page 1, 1998.

[ASH2] Ash, J., "Dynamic Routing in Telecommunications Networks", Book McGraw Hill, 1998.

[AWD2] Awduche, D., "MPLS and Traffic Engineering in IP Networks", Article IEEE Communications Magazine, December 1999.

[AWD5] Awduche, D., "An Approach to Optimal Peering Between Autonomous Systems in the Internet", Paper International Conference on Computer Communications and Networks (ICCCN'98), October 1998.

[FLJA93] Floyd, S. and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", Article IEEE/ACM Transactions on Networking, Vol. 1, p. 387-413, November 1993.

[FLOY94] Floyd, S., "TCP and Explicit Congestion Notification", Article ACM Computer Communication Review, V. 24, No. 5, p. 10-23, October 1994.

[FT00] Fortz, B. and M. Thorup, "Internet Traffic Engineering by Optimizing OSPF Weights", Article IEEE INFOCOM 2000, March 2000.

[FT01] Fortz, B. and M. Thorup, "Optimizing OSPF/IS-IS Weights in a Changing World", n.d., <http://www.research.att.com/~mthorup/PAPERS/papers.html>.

[HUSS87] Hurley, B., Seidl, C., and W. Sewel, "A Survey of Dynamic Routing Methods for Circuit-Switched Traffic", Article IEEE Communication Magazine, September 1987.

[I-D.ietf-alto-performance-metrics]
WU, Q., Yang, Y., Lee, Y., Dhody, D., Randriamasy, S., and L. Contreras, "ALTO Performance Cost Metrics", draft-ietf-alto-performance-metrics-14 (work in progress), January 2021.

[I-D.ietf-bess-evpn-unequal-lb]
Malhotra, N., Sajassi, A., Rabadan, J., Drake, J., Lingala, A., and S. Thoria, "Weighted Multi-Path Procedures for EVPN All-Active Multi-Homing", draft-ietf-bess-evpn-unequal-lb-07 (work in progress), October 2020.

[I-D.ietf-detnet-ip-over-tsn]
Varga, B., Farkas, J., Malis, A., and S. Bryant, "DetNet Data Plane: IP over IEEE 802.1 Time Sensitive Networking (TSN)", draft-ietf-detnet-ip-over-tsn-05 (work in progress), December 2020.

[I-D.ietf-idr-segment-routing-te-policy]
Previdi, S., Filsfils, C., Talaulikar, K., Mattes, P., Rosen, E., Jain, D., and S. Lin, "Advertising Segment Routing Policies in BGP", draft-ietf-idr-segment-routing-te-policy-11 (work in progress), November 2020.

[I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-13 (work in progress), October 2020.

[I-D.ietf-lsr-ip-flexalgo]
Britto, W., Hegde, S., Kaneriya, P., Shetty, R., Bonica, R., and P. Psenak, "IGP Flexible Algorithms (Flex-Algorithm) In IP Networks", draft-ietf-lsr-ip-flexalgo-00 (work in progress), December 2020.

[I-D.ietf-quic-transport]
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", draft-ietf-quic-transport-34 (work in progress), January 2021.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-09 (work in progress), November 2020.

[I-D.ietf-teas-enhanced-vpn]
Dong, J., Bryant, S., Li, Z., Miyasaka, T., and Y. Lee, "A Framework for Enhanced Virtual Private Networks (VPN+) Service", draft-ietf-teas-enhanced-vpn-06 (work in progress), July 2020.

[I-D.ietf-tewg-qos-routing]
Ash, G., "Traffic Engineering & QoS Methods for IP-, ATM-, & Based Multiservice Networks", draft-ietf-tewg-qos-routing-04 (work in progress), October 2001.

[I-D.irtf-nmrg-ibn-concepts-definitions]
Clemm, A., Ciavaglia, L., Granville, L., and J. Tantsura, "Intent-Based Networking - Concepts and Definitions", draft-irtf-nmrg-ibn-concepts-definitions-02 (work in progress), September 2020.

[I-D.nsdt-teas-ietf-network-slice-definition]
Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "Definition of IETF Network Slices", draft-nsdt-teas-ietf-network-slice-definition-02 (work in progress), December 2020.

[ITU-E600]
"Terms and Definitions of Traffic Engineering", Recommendation ITU-T Recommendation E.600, March 1993.

[ITU-E701]
"Reference Connections for Traffic Engineering", Recommendation ITU-T Recommendation E.701, October 1993.

[ITU-E801]
"Framework for Service Quality Agreement", Recommendation ITU-T Recommendation E.801, October 1996.

[MA] Ma, Q., "Quality of Service Routing in Integrated Services Networks", Ph.D. PhD Dissertation, CMU-CS-98-138, CMU, 1998.

[MATE] Elwalid, A., Jin, C., Low, S., and I. Widjaja, "MATE - MPLS Adaptive Traffic Engineering", Proceedings INFOCOM'01, April 2001.

[MCQ80] McQuillan, J., Richer, I., and E. Rosen, "The New Routing Algorithm for the ARPANET", Transaction IEEE Transactions on Communications, vol. 28, no. 5, p. 711-719, May 1980.

[MR99] Mitra, D. and K. Ramakrishnan, "A Case Study of Multiservice, Multipriority Traffic Engineering Design for Data Networks", Proceedings Globecom'99, December 1999.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <https://www.rfc-editor.org/info/rfc791>.

[RFC1102] Clark, D., "Policy routing in Internet protocols", RFC 1102, DOI 10.17487/RFC1102, May 1989, <https://www.rfc-editor.org/info/rfc1102>.

[RFC1104] Braun, H., "Models of policy based routing", RFC 1104, DOI 10.17487/RFC1104, June 1989, <https://www.rfc-editor.org/info/rfc1104>.

[RFC1992] Castineyra, I., Chiappa, N., and M. Steenstrup, "The Nimrod Routing Architecture", RFC 1992, DOI 10.17487/RFC1992, August 1996, <https://www.rfc-editor.org/info/rfc1992>.

[RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <https://www.rfc-editor.org/info/rfc2205>.

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <https://www.rfc-editor.org/info/rfc2328>.

[RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, DOI 10.17487/RFC2330, May 1998, <https://www.rfc-editor.org/info/rfc2330>.

[RFC2386] Crawley, E., Nair, R., Rajagopalan, B., and H. Sandick, "A Framework for QoS-based Routing in the Internet", RFC 2386, DOI 10.17487/RFC2386, August 1998, <https://www.rfc-editor.org/info/rfc2386>.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <https://www.rfc-editor.org/info/rfc2474>.

[RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, DOI 10.17487/RFC2475, December 1998, <https://www.rfc-editor.org/info/rfc2475>.

[RFC2597] Heinanen, J., Baker, F., Weiss, W., and J. Wroclawski, "Assured Forwarding PHB Group", RFC 2597, DOI 10.17487/RFC2597, June 1999, <https://www.rfc-editor.org/info/rfc2597>.

[RFC2678] Mahdavi, J. and V. Paxson, "IPPM Metrics for Measuring Connectivity", RFC 2678, DOI 10.17487/RFC2678, September 1999, <https://www.rfc-editor.org/info/rfc2678>.

[RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, DOI 10.17487/RFC2702, September 1999, <https://www.rfc-editor.org/info/rfc2702>.

[RFC2722] Brownlee, N., Mills, C., and G. Ruth, "Traffic Flow Measurement: Architecture", RFC 2722, DOI 10.17487/RFC2722, October 1999, <https://www.rfc-editor.org/info/rfc2722>.

[RFC2753] Yavatkar, R., Pendarakis, D., and R. Guerin, "A Framework for Policy-based Admission Control", RFC 2753, DOI 10.17487/RFC2753, January 2000, <https://www.rfc-editor.org/info/rfc2753>.

[RFC2961] Berger, L., Gan, D., Swallow, G., Pan, P., Tommasi, F., and S. Molendini, "RSVP Refresh Overhead Reduction Extensions", RFC 2961, DOI 10.17487/RFC2961, April 2001, <https://www.rfc-editor.org/info/rfc2961>.

[RFC2998] Bernet, Y., Ford, P., Yavatkar, R., Baker, F., Zhang, L., Speer, M., Braden, R., Davie, B., Wroclawski, J., and E. Felstaine, "A Framework for Integrated Services Operation over Diffserv Networks", RFC 2998, DOI 10.17487/RFC2998, November 2000, <https://www.rfc-editor.org/info/rfc2998>.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <https://www.rfc-editor.org/info/rfc3031>.

[RFC3086] Nichols, K. and B. Carpenter, "Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification", RFC 3086, DOI 10.17487/RFC3086, April 2001, <https://www.rfc-editor.org/info/rfc3086>.

[RFC3124] Balakrishnan, H. and S. Seshan, "The Congestion Manager", RFC 3124, DOI 10.17487/RFC3124, June 2001, <https://www.rfc-editor.org/info/rfc3124>.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <https://www.rfc-editor.org/info/rfc3209>.

[RFC3270] Le Faucheur, F., Wu, L., Davie, B., Davari, S., Vaananen, P., Krishnan, R., Cheval, P., and J. Heinanen, "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services", RFC 3270, DOI 10.17487/RFC3270, May 2002, <https://www.rfc-editor.org/info/rfc3270>.

[RFC3272] Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., and X. Xiao, "Overview and Principles of Internet Traffic Engineering", RFC 3272, DOI 10.17487/RFC3272, May 2002, <https://www.rfc-editor.org/info/rfc3272>.

[RFC3469] Sharma, V., Ed. and F. Hellstrand, Ed., "Framework for Multi-Protocol Label Switching (MPLS)-based Recovery", RFC 3469, DOI 10.17487/RFC3469, February 2003, <https://www.rfc-editor.org/info/rfc3469>.

[RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <https://www.rfc-editor.org/info/rfc3630>.

[RFC3945] Mannie, E., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", RFC 3945, DOI 10.17487/RFC3945, October 2004, <https://www.rfc-editor.org/info/rfc3945>.

[RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, DOI 10.17487/RFC4090, May 2005, <https://www.rfc-editor.org/info/rfc4090>.

[RFC4124] Le Faucheur, F., Ed., "Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering", RFC 4124, DOI 10.17487/RFC4124, June 2005, <https://www.rfc-editor.org/info/rfc4124>.

[RFC4203] Kompella, K., Ed. and Y. Rekhter, Ed., "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4203, DOI 10.17487/RFC4203, October 2005, <https://www.rfc-editor.org/info/rfc4203>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[RFC4594] Babiarz, J., Chan, K., and F. Baker, "Configuration Guidelines for DiffServ Service Classes", RFC 4594, DOI 10.17487/RFC4594, August 2006, <https://www.rfc-editor.org/info/rfc4594>.

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <https://www.rfc-editor.org/info/rfc4655>.

[RFC4872] Lang, J., Ed., Rekhter, Y., Ed., and D. Papadimitriou, Ed., "RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery", RFC 4872, DOI 10.17487/RFC4872, May 2007, <https://www.rfc-editor.org/info/rfc4872>.

[RFC4873] Berger, L., Bryskin, I., Papadimitriou, D., and A. Farrel, "GMPLS Segment Recovery", RFC 4873, DOI 10.17487/RFC4873, May 2007, <https://www.rfc-editor.org/info/rfc4873>.

[RFC5250] Berger, L., Bryskin, I., Zinin, A., and R. Coltun, "The OSPF Opaque LSA Option", RFC 5250, DOI 10.17487/RFC5250, July 2008, <https://www.rfc-editor.org/info/rfc5250>.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <https://www.rfc-editor.org/info/rfc5305>.

[RFC5329] Ishiguro, K., Manral, V., Davey, A., and A. Lindem, Ed., "Traffic Engineering Extensions to OSPF Version 3", RFC 5329, DOI 10.17487/RFC5329, September 2008, <https://www.rfc-editor.org/info/rfc5329>.

[RFC5331] Aggarwal, R., Rekhter, Y., and E. Rosen, "MPLS Upstream Label Assignment and Context-Specific Label Space", RFC 5331, DOI 10.17487/RFC5331, August 2008, <https://www.rfc-editor.org/info/rfc5331>.

[RFC5394] Bryskin, I., Papadimitriou, D., Berger, L., and J. Ash, "Policy-Enabled Path Computation Framework", RFC 5394, DOI 10.17487/RFC5394, December 2008, <https://www.rfc-editor.org/info/rfc5394>.

[RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <https://www.rfc-editor.org/info/rfc5440>.

[RFC5541] Le Roux, JL., Vasseur, JP., and Y. Lee, "Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)", RFC 5541, DOI 10.17487/RFC5541, June 2009, <https://www.rfc-editor.org/info/rfc5541>.

[RFC5557] Lee, Y., Le Roux, JL., King, D., and E. Oki, "Path Computation Element Communication Protocol (PCEP) Requirements and Protocol Extensions in Support of Global Concurrent Optimization", RFC 5557, DOI 10.17487/RFC5557, July 2009, <https://www.rfc-editor.org/info/rfc5557>.

[RFC5664] Halevy, B., Welch, B., and J. Zelenka, "Object-Based Parallel NFS (pNFS) Operations", RFC 5664, DOI 10.17487/RFC5664, January 2010, <https://www.rfc-editor.org/info/rfc5664>.

[RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, DOI 10.17487/RFC5693, October 2009, <https://www.rfc-editor.org/info/rfc5693>.

[RFC6107] Shiomoto, K., Ed. and A. Farrel, Ed., "Procedures for Dynamically Signaled Hierarchical Label Switched Paths", RFC 6107, DOI 10.17487/RFC6107, February 2011, <https://www.rfc-editor.org/info/rfc6107>.

[RFC6119] Harrison, J., Berger, J., and M. Bartlett, "IPv6 Traffic Engineering in IS-IS", RFC 6119, DOI 10.17487/RFC6119, February 2011, <https://www.rfc-editor.org/info/rfc6119>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6372] Sprecher, N., Ed. and A. Farrel, Ed., "MPLS Transport Profile (MPLS-TP) Survivability Framework", RFC 6372, DOI 10.17487/RFC6372, September 2011, <https://www.rfc-editor.org/info/rfc6372>.

[RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, DOI 10.17487/RFC6374, September 2011, <https://www.rfc-editor.org/info/rfc6374>.

[RFC6805] King, D., Ed. and A. Farrel, Ed., "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, DOI 10.17487/RFC6805, November 2012, <https://www.rfc-editor.org/info/rfc6805>.

[RFC7149] Boucadair, M. and C. Jacquenet, "Software-Defined Networking: A Perspective from within a Service Provider Environment", RFC 7149, DOI 10.17487/RFC7149, March 2014, <https://www.rfc-editor.org/info/rfc7149>.

[RFC7285] Alimi, R., Ed., Penno, R., Ed., Yang, Y., Ed., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", RFC 7285, DOI 10.17487/RFC7285, September 2014, <https://www.rfc-editor.org/info/rfc7285>.

[RFC7390] Rahman, A., Ed. and E. Dijk, Ed., "Group Communication for the Constrained Application Protocol (CoAP)", RFC 7390, DOI 10.17487/RFC7390, October 2014, <https://www.rfc-editor.org/info/rfc7390>.

[RFC7426] Haleplidis, E., Ed., Pentikousis, K., Ed., Denazis, S., Hadi Salim, J., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", RFC 7426, DOI 10.17487/RFC7426, January 2015, <https://www.rfc-editor.org/info/rfc7426>.

[RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015, <https://www.rfc-editor.org/info/rfc7471>.

[RFC7491] King, D. and A. Farrel, "A PCE-Based Architecture for Application-Based Network Operations", RFC 7491, DOI 10.17487/RFC7491, March 2015, <https://www.rfc-editor.org/info/rfc7491>.

[RFC7575] Behringer, M., Pritikin, M., Bjarnason, S., Clemm, A., Carpenter, B., Jiang, S., and L. Ciavaglia, "Autonomic Networking: Definitions and Design Goals", RFC 7575, DOI 10.17487/RFC7575, June 2015, <https://www.rfc-editor.org/info/rfc7575>.

[RFC7679] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <https://www.rfc-editor.org/info/rfc7679>.

[RFC7680] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Loss Metric for IP Performance Metrics (IPPM)", STD 82, RFC 7680, DOI 10.17487/RFC7680, January 2016, <https://www.rfc-editor.org/info/rfc7680>.

[RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <https://www.rfc-editor.org/info/rfc7752>.

[RFC7923] Voit, E., Clemm, A., and A. Gonzalez Prieto, "Requirements for Subscription to YANG Datastores", RFC 7923, DOI 10.17487/RFC7923, June 2016, <https://www.rfc-editor.org/info/rfc7923>.

[RFC7926] Farrel, A., Ed., Drake, J., Bitar, N., Swallow, G., Ceccarelli, D., and X. Zhang, "Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks", BCP 206, RFC 7926, DOI 10.17487/RFC7926, July 2016, <https://www.rfc-editor.org/info/rfc7926>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8051] Zhang, X., Ed. and I. Minei, Ed., "Applicability of a Stateful Path Computation Element (PCE)", RFC 8051, DOI 10.17487/RFC8051, January 2017, <https://www.rfc-editor.org/info/rfc8051>.

[RFC8189] Randriamasy, S., Roome, W., and N. Schwan, "Multi-Cost Application-Layer Traffic Optimization (ALTO)", RFC 8189, DOI 10.17487/RFC8189, October 2017, <https://www.rfc-editor.org/info/rfc8189>.

[RFC8231] Crabbe, E., Minei, I., Medved, J., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE", RFC 8231, DOI 10.17487/RFC8231, September 2017, <https://www.rfc-editor.org/info/rfc8231>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/info/rfc8259>.

[RFC8281] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model", RFC 8281, DOI 10.17487/RFC8281, December 2017, <https://www.rfc-editor.org/info/rfc8281>.

[RFC8283] Farrel, A., Ed., Zhao, Q., Ed., Li, Z., and C. Zhou, "An Architecture for Use of PCE and the PCE Communication Protocol (PCEP) in a Network with Central Control", RFC 8283, DOI 10.17487/RFC8283, December 2017, <https://www.rfc-editor.org/info/rfc8283>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC 8453, DOI 10.17487/RFC8453, August 2018, <https://www.rfc-editor.org/info/rfc8453>.

[RFC8570] Ginsberg, L., Ed., Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 8570, DOI 10.17487/RFC8570, March 2019, <https://www.rfc-editor.org/info/rfc8570>.

[RFC8571] Ginsberg, L., Ed., Previdi, S., Wu, Q., Tantsura, J., and C. Filsfils, "BGP - Link State (BGP-LS) Advertisement of IGP Traffic Engineering Performance Metric Extensions", RFC 8571, DOI 10.17487/RFC8571, March 2019, <https://www.rfc-editor.org/info/rfc8571>.

[RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <https://www.rfc-editor.org/info/rfc8655>.

[RFC8661] Bashandy, A., Ed., Filsfils, C., Ed., Previdi, S., Decraene, B., and S. Litkowski, "Segment Routing MPLS Interworking with LDP", RFC 8661, DOI 10.17487/RFC8661, December 2019, <https://www.rfc-editor.org/info/rfc8661>.

[RFC8664] Sivabalan, S., Filsfils, C., Tantsura, J., Henderickx, W., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing", RFC 8664, DOI 10.17487/RFC8664, December 2019, <https://www.rfc-editor.org/info/rfc8664>.

[RFC8685] Zhang, F., Zhao, Q., Gonzalez de Dios, O., Casellas, R., and D. King, "Path Computation Element Communication Protocol (PCEP) Extensions for the Hierarchical Path Computation Element (H-PCE) Architecture", RFC 8685, DOI 10.17487/RFC8685, December 2019, <https://www.rfc-editor.org/info/rfc8685>.

[RFC8795] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Gonzalez de Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", RFC 8795, DOI 10.17487/RFC8795, August 2020, <https://www.rfc-editor.org/info/rfc8795>.

[RFC8896] Randriamasy, S., Yang, R., Wu, Q., Deng, L., and N. Schwan, "Application-Layer Traffic Optimization (ALTO) Cost Calendar", RFC 8896, DOI 10.17487/RFC8896, November 2020, <https://www.rfc-editor.org/info/rfc8896>.

[RFC8938] Varga, B., Ed., Farkas, J., Berger, L., Malis, A., and S. Bryant, "Deterministic Networking (DetNet) Data Plane Framework", RFC 8938, DOI 10.17487/RFC8938, November 2020, <https://www.rfc-editor.org/info/rfc8938>.

[RFC8955] Loibl, C., Hares, S., Raszuk, R., McPherson, D., and M. Bacher, "Dissemination of Flow Specification Rules", RFC 8955, DOI 10.17487/RFC8955, December 2020, <https://www.rfc-editor.org/info/rfc8955>.

[RR94] Rodrigues, M. and K. Ramakrishnan, "Optimal Routing in Shortest Path Networks", Proceedings ITS'94, Rio de Janeiro, Brazil, 1994.

[SLDC98] Suter, B., Lakshman, T., Stiliadis, D., and A. Choudhury, "Design Considerations for Supporting TCP with Per-flow Queueing", Proceedings INFOCOM'98, p. 299-306, 1998.

[WANG] Wang, Y., Wang, Z., and L. Zhang, "Internet traffic engineering without full mesh overlaying", Proceedings INFOCOM'2001, April 2001.

[XIAO] Xiao, X., Hannan, A., Bailey, B., and L. Ni, "Traffic Engineering with MPLS in the Internet", Article IEEE Network Magazine, March 2000.

[YARE95] Yang, C. and A. Reddy, "A Taxonomy for Congestion Control Algorithms in Packet Switching Networks", Article IEEE Network Magazine, p. 34-45, 1995.

Appendix A. Historic Overview

A.1. Traffic Engineering in Classical Telephone Networks

This subsection presents a brief overview of traffic engineering in telephone networks which often relates to the way user traffic is steered from an originating node to the terminating node. This subsection presents a brief overview of this topic. A detailed description of the various routing strategies applied in telephone networks is included in the book by G. Ash [ASH2].

The early telephone network relied on static hierarchical routing, whereby routing patterns remained fixed independent of the state of the network or time of day. The hierarchy was intended to accommodate overflow traffic, improve network reliability via alternate routes, and prevent call looping by employing strict hierarchical rules. The network was typically over-provisioned since a given fixed route had to be dimensioned so that it could carry user traffic during a busy hour of any busy day. Hierarchical routing in the telephony network was found to be too rigid upon the advent of digital switches and stored program control which were able to manage more complicated traffic engineering rules.

Dynamic routing was introduced to alleviate the routing inflexibility in the static hierarchical routing so that the network would operate more efficiently. This resulted in significant economic gains [HUSS87]. Dynamic routing typically reduces the overall loss probability by 10 to 20 percent (compared to static hierarchical routing). Dynamic routing can also improve network resilience by recalculating routes on a per-call basis and periodically updating routes.

There are three main types of dynamic routing in the telephone network. They are time-dependent routing, state-dependent routing (SDR), and event dependent routing (EDR).

In time-dependent routing, regular variations in traffic loads (such as time of day or day of week) are exploited in pre-planned routing tables. In state-dependent routing, routing tables are updated online according to the current state of the network (e.g., traffic demand, utilization, etc.). In event dependent routing, routing changes are triggers by events (such as call setups encountering congested or blocked links) whereupon new paths are searched out using learning models. EDR methods are real-time adaptive, but they do not require global state information as does SDR. Examples of EDR schemes include the dynamic alternate routing (DAR) from BT, the state-and-time dependent routing (STR) from NTT, and the success-to-the-top (STT) routing from AT&T.

Dynamic non-hierarchical routing (DNHR) is an example of dynamic routing that was introduced in the AT&T toll network in the 1980's to respond to time-dependent information such as regular load variations as a function of time. Time-dependent information in terms of load may be divided into three timescales: hourly, weekly, and yearly. Correspondingly, three algorithms are defined to pre-plan the routing tables. The network design algorithm operates over a year-long interval while the demand servicing algorithm operates on a weekly basis to fine tune link sizes and routing tables to correct forecast errors on the yearly basis. At the smallest timescale, the routing algorithm is used to make limited adjustments based on daily traffic variations. Network design and demand servicing are computed using offline calculations. Typically, the calculations require extensive searches on possible routes. On the other hand, routing may need online calculations to handle crankback. DNHR adopts a "two-link" approach whereby a path can consist of two links at most. The routing algorithm presents an ordered list of route choices between an originating switch and a terminating switch. If a call overflows, a via switch (a tandem exchange between the originating switch and the terminating switch) would send a crankback signal to the originating switch. This switch would then select the next route, and so on, until there are no alternative routes available in which the call is blocked.

A.2. Evolution of Traffic Engineering in Packet Networks

This subsection reviews related prior work that was intended to improve the performance of data networks. Indeed, optimization of the performance of data networks started in the early days of the ARPANET. Other early commercial networks such as SNA also recognized the importance of performance optimization and service differentiation.

In terms of traffic management, the Internet has been a best effort service environment until recently. In particular, very limited traffic management capabilities existed in IP networks to provide differentiated queue management and scheduling services to packets belonging to different classes.

In terms of routing control, the Internet has employed distributed protocols for intra-domain routing. These protocols are highly scalable and resilient. However, they are based on simple algorithms for path selection which have very limited functionality to allow flexible control of the path selection process.

In the following subsections, the evolution of practical traffic engineering mechanisms in IP networks and its predecessors are reviewed.

A.2.1. Adaptive Routing in the ARPANET

The early ARPANET recognized the importance of adaptive routing where routing decisions were based on the current state of the network [MCQ80]. Early minimum delay routing approaches forwarded each packet to its destination along a path for which the total estimated transit time was the smallest. Each node maintained a table of network delays, representing the estimated delay that a packet would experience along a given path toward its destination. The minimum delay table was periodically transmitted by a node to its neighbors. The shortest path, in terms of hop count, was also propagated to give the connectivity information.

One drawback to this approach is that dynamic link metrics tend to create "traffic magnets" causing congestion to be shifted from one location of a network to another location, resulting in oscillation and network instability.

A.2.2. Dynamic Routing in the Internet

The Internet evolved from the ARPANET and adopted dynamic routing algorithms with distributed control to determine the paths that packets should take en-route to their destinations. The routing algorithms are adaptations of shortest path algorithms where costs are based on link metrics. The link metric can be based on static or dynamic quantities. The link metric based on static quantities may be assigned administratively according to local criteria. The link metric based on dynamic quantities may be a function of a network congestion measure such as delay or packet loss.

It was apparent early that static link metric assignment was inadequate because it can easily lead to unfavorable scenarios in which some links become congested while others remain lightly loaded. One of the many reasons for the inadequacy of static link metrics is that link metric assignment was often done without considering the traffic matrix in the network. Also, the routing protocols did not take traffic attributes and capacity constraints into account when making routing decisions. This results in traffic concentration being localized in subsets of the network infrastructure and potentially causing congestion. Even if link metrics are assigned in accordance with the traffic matrix, unbalanced loads in the network can still occur due to a number factors including:

- Resources may not be deployed in the most optimal locations from a routing perspective.

- Forecasting errors in traffic volume and/or traffic distribution.

- Dynamics in traffic matrix due to the temporal nature of traffic patterns, BGP policy change from peers, etc.

The inadequacy of the legacy Internet interior gateway routing system is one of the factors motivating the interest in path oriented technology with explicit routing and constraint-based routing capability such as MPLS.

A.2.3. ToS Routing

Type-of-Service (ToS) routing involves different routes going to the same destination with selection dependent upon the ToS field of an IP packet [RFC2474]. The ToS classes may be classified as low delay and high throughput. Each link is associated with multiple link costs and each link cost is used to compute routes for a particular ToS. A separate shortest path tree is computed for each ToS. The shortest path algorithm must be run for each ToS resulting in very expensive computation. Classical ToS-based routing is now outdated as the IP header field has been replaced by a Diffserv field. Effective traffic engineering is difficult to perform in classical ToS-based routing because each class still relies exclusively on shortest path routing which results in localization of traffic concentration within the network.

A.2.4. Equal Cost Multi-Path

Equal Cost Multi-Path (ECMP) is another technique that attempts to address the deficiency in the Shortest Path First (SPF) interior gateway routing systems [RFC2328]. In the classical SPF algorithm, if two or more shortest paths exist to a given destination, the algorithm will choose one of them. The algorithm is modified slightly in ECMP so that if two or more equal cost shortest paths exist between two nodes, the traffic between the nodes is distributed among the multiple equal-cost paths. Traffic distribution across the equal-cost paths is usually performed in one of two ways: (1) packet-based in a round-robin fashion, or (2) flow-based using hashing on source and destination IP addresses and possibly other fields of the IP header. The first approach can easily cause out- of-order packets while the second approach is dependent upon the number and distribution of flows. Flow-based load sharing may be unpredictable in an enterprise network where the number of flows is relatively small and less heterogeneous (for example, hashing may not be uniform), but it is generally effective in core public networks where the number of flows is large and heterogeneous.

In ECMP, link costs are static and bandwidth constraints are not considered, so ECMP attempts to distribute the traffic as equally as possible among the equal-cost paths independent of the congestion status of each path. As a result, given two equal-cost paths, it is possible that one of the paths will be more congested than the other. Another drawback of ECMP is that load sharing cannot be achieved on multiple paths which have non-identical costs.

A.2.5. Nimrod

Nimrod was a routing system developed to provide heterogeneous service specific routing in the Internet, while taking multiple constraints into account [RFC1992]. Essentially, Nimrod was a link state routing protocol to support path oriented packet forwarding. It used the concept of maps to represent network connectivity and services at multiple levels of abstraction. Mechanisms allowed restriction of the distribution of routing information.

Even though Nimrod did not enjoy deployment in the public Internet, a number of key concepts incorporated into the Nimrod architecture, such as explicit routing which allows selection of paths at originating nodes, are beginning to find applications in some recent constraint-based routing initiatives.

A.3. Development of Internet Traffic Engineering

A.3.1. Overlay Model

In the overlay model, a virtual-circuit network, such as Synchronous Optical Network / Synchronous Digital Hierarchy (SONET/SDH), Optical Transport Network (OTN), or Wavelength Division Multiplexing (WDM), provides virtual-circuit connectivity between routers that are located at the edges of a virtual-circuit cloud. In this mode, two routers that are connected through a virtual circuit see a direct adjacency between themselves independent of the physical route taken by the virtual circuit through the ATM, frame relay, or WDM network. Thus, the overlay model essentially decouples the logical topology that routers see from the physical topology that the ATM, frame relay, or WDM network manages. The overlay model based on ATM or frame relay enables a network administrator or an automaton to employ traffic engineering concepts to perform path optimization by re-configuring or rearranging the virtual circuits so that a virtual circuit on a congested or sub-optimal physical link can be re-routed to a less congested or more optimal one. In the overlay model, traffic engineering is also employed to establish relationships between the traffic management parameters (e.g., Peak Cell Rate, Sustained Cell Rate, and Maximum Burst Size for ATM) of the virtual-circuit technology and the actual traffic that traverses each circuit. These relationships can be established based upon known or projected traffic profiles, and some other factors.

Appendix B. Overview of Traffic Engineering Related Work in Other SDOs

B.1. Overview of ITU Activities Related to Traffic Engineering

This section provides an overview of prior work within the ITU-T pertaining to traffic engineering in traditional telecommunications networks.

ITU-T Recommendations E.600 [ITU-E600], E.701 [ITU-E701], and E.801 [ITU-E801] address traffic engineering issues in traditional telecommunications networks. Recommendation E.600 provides a vocabulary for describing traffic engineering concepts, while E.701 defines reference connections, Grade of Service (GoS), and traffic parameters for ISDN. Recommendation E.701 uses the concept of a reference connection to identify representative cases of different types of connections without describing the specifics of their actual realizations by different physical means. As defined in Recommendation E.600, "a connection is an association of resources providing means for communication between two or more devices in, or attached to, a telecommunication network." Also, E.600 defines "a resource as any set of physically or conceptually identifiable entities within a telecommunication network, the use of which can be unambiguously determined" [ITU-E600]. There can be different types of connections as the number and types of resources in a connection may vary.

Typically, different network segments are involved in the path of a connection. For example, a connection may be local, national, or international. The purposes of reference connections are to clarify and specify traffic performance issues at various interfaces between different network domains. Each domain may consist of one or more service provider networks.

Reference connections provide a basis to define grade of service (GoS) parameters related to traffic engineering within the ITU-T framework. As defined in E.600, "GoS refers to a number of traffic engineering variables which are used to provide a measure of the adequacy of a group of resources under specified conditions." These GoS variables may be probability of loss, dial tone, delay, etc. They are essential for network internal design and operation as well as for component performance specification.

GoS is different from quality of service (QoS) in the ITU framework. QoS is the performance perceivable by a telecommunication service user and expresses the user's degree of satisfaction of the service. QoS parameters focus on performance aspects observable at the service access points and network interfaces, rather than their causes within the network. GoS, on the other hand, is a set of network oriented measures which characterize the adequacy of a group of resources under specified conditions. For a network to be effective in serving its users, the values of both GoS and QoS parameters must be related, with GoS parameters typically making a major contribution to the QoS.

Recommendation E.600 stipulates that a set of GoS parameters must be selected and defined on an end-to-end basis for each major service category provided by a network to assist the network provider with improving efficiency and effectiveness of the network. Based on a selected set of reference connections, suitable target values are assigned to the selected GoS parameters under normal and high load conditions. These end-to-end GoS target values are then apportioned to individual resource components of the reference connections for dimensioning purposes.

Appendix C. Summary of Changes Since RFC 3272

The changes to this document since RFC 3272 are substantial and not easily summarized as section-by-section changes. The material in the document has been moved around considerably, some of it removed, and new text added.

The approach taken here is to list the table of content of both the previous RFC and this document saying, respectively, where the text has been place and where the text came from.

C.1. RFC 3272

1.0 Introduction: Edited in place in Section 1.

1.1 What is Internet Traffic Engineering?: Edited in place in Section 1.1.

1.2 Scope: Moved to Section 1.3.

1.3 Terminology: Moved to Section 1.4 with some obsolete terms removed and a little editing.

2.0 Background: Retained as Section 2 with some text removed.

2.1 Context of Internet Traffic Engineering: Retained as Section 2.1.

2.2 Network Context: Rewritten as Section 2.2.

2.3 Problem Context: Rewritten as Section 2.3.

2.3.1 Congestion and its Ramifications: Retained as Section 2.3.1.

2.4 Solution Context: Edited as Section 2.4.

2.4.1 Combating the Congestion Problem: Reformatted as Section 2.4.1.

2.5 Implementation and Operational Context: Retained as Section 2.5.

3.0 Traffic Engineering Process Model: Retained as Section 3.

3.1 Components of the Traffic Engineering Process Model: Retained as Section 3.1.

3.2 Measurement: Merged into Section 3.1.

3.3 Modeling, Analysis, and Simulation: Merged into Section 3.1.

3.4 Optimization: Merged into Section 3.1.

4.0 Historical Review and Recent Developments: Retained as Section 4, but the very historic aspects moved to Appendix A.

4.1 Traffic Engineering in Classical Telephone Networks: Moved to Appendix A.1.

4.2 Evolution of Traffic Engineering in the Internet: Moved to Appendix A.2.

4.2.1 Adaptive Routing in ARPANET: Moved to Appendix A.2.1.

4.2.2 Dynamic Routing in the Internet: Moved to Appendix A.2.2.

4.2.3 ToS Routing: Moved to Appendix A.2.3.

4.2.4 Equal Cost Multi-Path: Moved to Appendix A.2.4.

4.2.5 Nimrod: Moved to Appendix A.2.5.

4.3 Overlay Model: Moved to Appendix A.3.1.

4.4 Constraint-Based Routing: Retained as Section 4.1.1, but moved into Section 4.1.

4.5 Overview of Other IETF Projects Related to Traffic Engineering:
Retained as Section 4.1 with many new subsections.

4.5.1 Integrated Services: Retained as Section 4.1.2.

4.5.2 RSVP: Retained as Section 4.1.3 with some edits.

4.5.3 Differentiated Services: Retained as Section 4.1.4.

4.5.4 MPLS: Retained as Section 4.1.6.

4.5.5 IP Performance Metrics: Retained as Section 4.1.8.

4.5.6 Flow Measurement: Retained as Section 4.1.9 with some reformatting.

4.5.7 Endpoint Congestion Management: Retained as Section 4.1.10.

4.6 Overview of ITU Activities Related to Traffic Engineering: Moved to Appendix B.1.

4.7 Content Distribution: Retained as Section 4.2.

5.0 Taxonomy of Traffic Engineering Systems: Retained as Section 5.

5.1 Time-Dependent Versus State-Dependent: Retained as Section 5.1.

5.2 Offline Versus Online: Retained as Section 5.2.

5.3 Centralized Versus Distributed: Retained as Section 5.3 with additions.

5.4 Local Versus Global: Retained as Section 5.4.

5.5 Prescriptive Versus Descriptive: Retained as Section 5.5 with additions.

5.6 Open-Loop Versus Closed-Loop: Retained as Section 5.6.

5.7 Tactical vs Strategic: Retained as Section 5.7.

6.0 Recommendations for Internet Traffic Engineering: Retained as Section 6.

6.1 Generic Non-functional Recommendations: Retained as Section 6.1.

6.2 Routing Recommendations: Retained as Section 6.2 with edits.

6.3 Traffic Mapping Recommendations: Retained as Section 6.3.

6.4 Measurement Recommendations: Retained as Section 6.4.

6.5 Network Survivability: Retained as Section 6.5.

6.5.1 Survivability in MPLS Based Networks: Retained as Section 6.5.1.

6.5.2 Protection Option: Retained as Section 6.5.2.

6.6 Traffic Engineering in Diffserv Environments: Retained as Section 6.6 with edits.

6.7 Network Controllability: Retained as Section 6.7.

7.0 Inter-Domain Considerations: Retained as Section 7.

8.0 Overview of Contemporary TE Practices in Operational IP Networks:

Retained as Section 8.

9.0 Conclusion: Removed.

10.0 Security Considerations: Retained as Section 9 with considerable new text.

C.2. This Document

- o Section 1: Based on Section 1 of RFC 3272.
  - * Section 1.1: Based on Section 1.1 of RFC 3272.
  - * Section 1.2: New for this document.
  - * Section 1.3: Based on Section 1.2 of RFC 3272.
  - * Section 1.4: Based on Section 1.3 of RFC 3272.
- o Section 2: Based on Section 2. of RFC 3272.
  - * Section 2.1: Based on Section 2.1 of RFC 3272.
  - * Section 2.2: Based on Section 2.2 of RFC 3272.
  - * Section 2.3: Based on Section 2.3 of RFC 3272.
    - + Section 2.3.1: Based on Section 2.3.1 of RFC 3272.
  - * Section 2.4: Based on Section 2.4 of RFC 3272.

        + Section 2.4.1: Based on Section 2.4.1 of RFC 327

    - Section 2.5: Based on Section 2.5 of RFC 3272.

- Section 3: Based on Section 3 of RFC 3272.

    - Section 3.1: Based on Sections 3.1, 3.2, 3.3, and 3.4 of RFC 3272.

- Section 4: Based on Section 4 of RFC 3272.

    - Section 4.1: Based on Section 4.5 of RFC 3272.

        - Section 4.1.1: Based on Section 4.4 of RFC 3272.

            - Section 4.1.1.1: New for this document.

        - Section 4.1.2: Based on Section 4.5.1 of RFC 3272.

        - Section 4.1.3: Based on Section 4.5.2 of RFC 3272.

        - Section 4.1.4: Based on Section 4.5.3 of RFC 3272.

        - Section 4.1.5: New for this document.

        - Section 4.1.6: Based on Section 4.5.4 of RFC 3272.

        - Section 4.1.7: New for this document.

        - Section 4.1.8: Based on Section 4.5.5 of RFC 3272.

        - Section 4.1.9: Based on Section 4.5.6 of RFC 3272.

        - Section 4.1.10: Based on Section 4.5.7 of RFC 3272.

        - Section 4.1.11: New for this document.

        - Section 4.1.12: New for this document.

        - Section 4.1.13: New for this document.

        - Section 4.1.14: New for this document.

        - Section 4.1.15: New for this document.

        - Section 4.1.16: New for this document.

            - Section 4.1.16.1: New for this document.

      - Section 4.1.16.2: New for this document.

    + Section 4.1.17: New for this document.

    + Section 4.1.18: New for this document.

    + Section 4.1.19: New for this document.

    + Section 4.1.20: New for this document.

    + Section 4.1.21: New for this document.

  * Section 4.2: Based on Section 4.7 of RFC 3272.

- Section 5: Based on Section 5 of RFC 3272.

  * Section 5.1: Based on Section 5.1 of RFC 3272.

  * Section 5.2: Based on Section 5.2 of RFC 3272.

  * Section 5.3: Based on Section 5.3 of RFC 3272.

    + Section 5.3.1: New for this document.

    + Section 5.3.2: New for this document.

  * Section 5.4: Based on Section 5.4 of RFC 3272.

  * Section 5.5: Based on Section 5.5 of RFC 3272.

    + Section 5.5.1: New for this document.

  * Section 5.6: Based on Section 5.6 of RFC 3272.

  * Section 5.7: Based on Section 5.7 of RFC 3272.

- Section 6: Based on Section 6 of RFC 3272.

  * Section 6.1: Based on Section 6.1 of RFC 3272.

  * Section 6.2: Based on Section 6.2 of RFC 3272.

  * Section 6.3: Based on Section 6.3 of RFC 3272.

  * Section 6.4: Based on Section 6.4 of RFC 3272.

  * Section 6.5: Based on Section 6.5 of RFC 3272.

    + Section 6.5.1: Based on Section 6.5.1 of RFC 3272.

        + Section 6.5.2: Based on Section 6.5.2 of RFC 3272.

    * Section 6.6: Based on Section 6.6. of RFC 3272.

    * Section 6.7: Based on Section 6.7 of RFC 3272.

- o Section 7: Based on Section 7 of RFC 3272.

- o Section 8: Based on Section 8 of RFC 3272.

- o Section 9: Based on Section 10 of RFC 3272.

- o Appendix A: New for this document.

    * Appendix A.1: Based on Section 4.1 of RFC 3272.

    * Appendix A.2: Based on Section 4.2 of RFC 3272.

        + Appendix A.2.1: Based on Section 4.2.1 of RFC 3272.

        + Appendix A.2.2: Based on Section 4.2.2 of RFC 3272.

        + Appendix A.2.3: Based on Section 4.2.3 of RFC 3272.

        + Appendix A.2.4: Based on Section 4.2.4 of RFC 3272.

        + Appendix A.2.5: Based on Section 4.2.5 of RFC 3272.

    * Appendix A.3: New for this document.

        + Appendix A.3.1: Based on Section 4.3 of RFC 3272.

- o Appendix B: New for this document.

    * Appendix B.1: Based on Section 4.7 of RFC 3272.

Author's Address

Adrian Farrel (editor)
Old Dog Consulting

Email: adrian@olddog.co.uk