The Pennsylvania State University
The Graduate School

A TRAFFIC ENGINEERING ATTRIBUTE FOR BGP

A Thesis in
Computer Science and Engineering
by
Todd Arnold

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2008

The thesis of Todd Arnold was reviewed and approved* by the following:

George Kesidis
Professor of Computer Science and Engineering
Professor of Electrical Engineering
Thesis Advisor

Guohong Cao
Associate Professor of Computer Science and Engineering

Raj Acharya
Professor of Computer Science and Engineering
Head of the Department of Computer Science and Engineering

*Signatures are on file in the Graduate School.

Abstract

In the Internet, "traffic engineering" seeks to satisfy a packet stream's, or flow's, quality of service requirements while simultaneously leveraging existing resources to distribute a network's traffic load across all available paths, in order to provide congestion control. Traditional interior gateway protocols were expanded to distribute detailed interface attributes which are used by a constrained shortest path first algorithm to calculate a path which will best satisfy a flow's quality requirements. Multi-protocol label switching and reservation protocols are used to allocate resources along the path and to ensure the flow will traverse the calculated path.

Traffic engineering is also performed between domains using the interdomain routing protocol, BGP. However, BGP's ability to balance network requirements across multiple links is severely limited. BGP's limited abilities only provide the means for designating the desired ingress point for a destination network. To rectify interdomain traffic engineering shortfalls, path reservations for MPLS using RSVP-TE was extended to cross domain boundaries. Despite the ability to reserve resources across domain boundaries, a method for distributing information to calculate which path will best satisfy a flow's QoS requirements across domain boundaries does not exist.

We present a BGP attribute for distributing path quality information across domain boundaries. We explore a method for calculating the values to advertise while preserving the requirement to abstract a domain's internal policies. We also investigate the impact on routing table memory requirements for distributing this information by using simulations based on real-world topology and prefix information.

Table of Contents

List of Figures

List of Tables

List of Symbols

X	Constant for OSPF metric calculation, p. 3
$Bandwidth$	The maximum bandwidth capacity of an interface, p. 3
N	The number of routers in a network, p. 9
V	The number of vertices in a graph, p. 16
$NULL$	Invalid or non-existant assignment, p. 16
E	The number of edges in a graph, p. 16
∞	Infinity, p. 16
v	The vertice selected as the root of a shortest path tree, p. 16
v_n	The destination vertice of the shortest edge in Dijkstra's Algorithm, p. 19
N	The number of Autonomous Systems in the simulation, p. 62
ρ	The number of prefixes each Autonomous System advertises during the simulation, p. 62
r	The average number of neighbors per router, p. 62
α,β,δ	Proportionality constants, p. 62

Acknowledgments

I would like to thank Dr. Kesidis and Dr. Cao for their time, patience, and mentorship. I would also like to thank Glenn Carl for his advice and assistance.

Dedication

To Jamie, Alley, and Isabelle.

Chapter 1

Introduction

1.1 Overview

The Internet has evolved from a small research oriented resource to a global communication infrastructure encompassing the realms of research, entertainment, business, banking, news, and communication. During the evolution from a small research network to global resource, the network transfored from a best effort service to become a heavily relied upon resource which is required to be stable and robust. The architecture and supporting transport protocols have also drastically changed due to new requirements placed on the network. The migration of real-time services, such as Voice over Internet Protocol (VoIP), Video Teleconferencing (VTC), and high definition interactive gaming onto the Internet has driven an increasing demand for better than best effort transport. Quality of service (QoS) requirements and the need for efficient use of available resources has driven the development of traffic engineering and its supporting protocols.

Traffic engineering seeks to provide the desired QoS for traffic as it traverses the Internet, while also efficiently leveraging existing resources to provide congestion control. Within an autonomous system (AS), the capability exists to distribute the status of network resource availability and to use the information for providing QoS [1]. Protocols for establishing *tunnels*, a virtual connection between two network devices created by encapsulating one protocol within another [2], across domain boundaries to satisfy QoS also exists [3]. However, a method for distributing the information required to accurately calculate supportable interdomain *paths*, the

sequence of routers or domains a packet will traverse, does not. The challenge in distributing this information is the amount of information required as well as concerns about the scalability of possible solutions in terms of memory consumption and update messages generated.

We explore a method to provide accurate path attributes across domain boundaries in a scalable manner while respecting the privacy required by most domains. Our proposal will allow an AS to advertise the quality of a path in a manner consistent with current protocol advertisements without placing an undue burden on the AS border routers (ASBR).

This chapter is organized as follows. We first provide an overview of the Internet architecture and its challenges. In Section 1.3 we discuss the motivation behind this work. Sections 1.4, 1.5, and 1.6 discuss the advantages and disadvantages of connection-oriented vs. connectionless forwarding, and the benefits of hybrid models, respectively. Section 1.7 provides a description of QoS and the historical proposals to support it. Section 1.8 describes traffic engineering, its goals, and support. Finally, Section 1.9 provides an overview for the rest of the document.

1.2 Internet Architecture

Thousands of interconnected domains comprise the architecture of the Internet. It is maintained by Internet Service Providers (ISPs) and carriers, each controlling their portion of the network inside an Autonomous System (AS) or *domain*. An AS is defined as "a set of routers under a single technical administration, using an interior gateway protocol (IGP) and common metrics to determine how to route packets within the AS, and using an inter-AS routing protocol to determine how to route packets to other ASes"[4].

When a packet enters an AS, it is forwarded based on the IGP's selected best path for the destination network. An IGP's best path is determined in terms of a *metric*, which is a numerical value assigned to each individual connection between two neighboring nodes, termed a *link*. A metric is typically based on some measurable attribute of a link. One example of a measurable attribute is *bandwidth*, which is the amount of data a node can transmit on a link within a given time period and is usually calculated in bits per second (bps). IGPs may

choose to use different metrics; the Open Shortest Path First (OSPF) protocol sometimes uses additive costs based on a constant times the inverse maximum bandwidth capacity of a link ($\frac{X}{Bandwidth}$) [5]. The default Cisco IOS value for X is 10^8 [6]. Another common metric is *hop count*, which is the number of nodes a packet would have to traverse to reach a destination. Despite the strengths of the different IGPs, simple best path algorithms based on a single metric can cause *congestion*, which we define as a link carrying such a high percentage of its maximum bandwidth so as to increase packet delay. Since routing decisions are based on metrics and are topology driven, best paths algorithms tend to converge onto a few links within a network; convergence happens despite the existence of viable alternate paths within the network. The convergence of traffic onto a small number of paths within the network can cause congestion [7].

When calculating the best path, the degree of individual link and queue congestion is typically not taken into account; neither is bursty traffic, reliability, or resource availability, among other factors. *Load balancing*, having one node distribute network traffic over more than one path, can ease congestion on individual links. However, load balancing typically requires each alternate path be equivalent in cost to the original best path, which is not always feasible or practical.

Confounding the issue of congestion, the introduction of real-time services and its exponential growth in usage has increased the demand for minimal delay data transfers. For example, VoIP can only tolerate a maximum one way delay of 200 milliseconds (ms) before quality degrades [8]. This migration phenomenon, known as Internet Protocol (IP) Convergence [9], leads to subscribers demanding additional bandwidth to receive these services in real-time, maximizing the utilization of links on the Internet.

End user systems are not much help in relieving network congestion. The Transmission Control Protocol (TCP) provides congestion control through its reaction to lost packets by reducing the transmission rate [10]. However, TCP immediately begins increasing the amount of traffic it believes the network can sustain until the connection once again suffers lost packets. TCP initially places a large amount of traffic on the network without knowledge of sustainability, degrading the performance of networks without the ability to handle bursty traffic [11]. The alternative to the reliable TCP is the User Datagram Protocol (UDP), which does

not provide delivery guarantees, flow control, or congestion detection [12], and is used by a large segment of real-time services, such as VoIP and VTC. Networks cannot rely on end user systems to fully cooperate with congestion control nor can congestion control by end users be considered a practical or complete solution.

In order to provide reliable service to customers, ISPs may be forced to develop alternative end-to-end congestion relief methods. The most common solution to relieve network congestion is deploying additional bandwidth to maintain low link utilization. This is not necessarily a cost-effective solution. To maintain reliability and robustness, an ISP must maintain more than one physical path between locations within their network. Purchasing additional bandwidth along one path requires upgrading multiple paths to ensure backups are viable. Despite the additional bandwidth, all packets are still funneled onto a common path and are placed together into a common queue within each router. This effect can cause variable delay times, *jitter*, and unexpected packet loss within a network.

A QoS framework involves a set of service requirements to be met by the network for a sequence of data packets exchanged between two devices on the network, also known as a *flow* [13]. Common QoS parameters define thresholds for delay, jitter, maximum and minimum throughput, or packet loss. The QoS method for ensuring required thresholds are achieved is based on *priority queuing*. Priority queuing services award preferential treatment to certain types of packets, allowing those packets quicker access to the physical medium. The priority queues are partitioned based on the different service requirements for each traffic type. For example, real-time voice and multimedia services across the Internet are not very tolerant of delay, jitter, and packet loss, which are common in a best effort network. As a result, they would be afforded a priority status over delay tolerant packets in a proper QoS policy. On the other hand, TCP based communications, such as large file transfers, are tolerant of some packet loss and will scale back their bandwidth utilization based on availability across the network, so they are afforded a lower priority status. However, QoS alone is not capable of solving all congestion problems; saturated links or full queues are not avoided, which can lead to packet loss.

Another framework, termed traffic engineering (TE), attempts to utilize the available bandwidth of redundant network paths, thus distributing the traffic load

more evenly. By distributing the load evenly, more bandwidth is available to subscribers, points of congestion can be avoided, and QoS requirements can be better fulfilled. TE is similar to load balancing, but unlike load balancing, TE may distribute the load across multiple paths of non equal cost [7]. Common TE techniques for current protocols consists of attribute manipulation, such as in the Border Gateway Protocol (BGP) [14] or OSPF [15]. However, attribute manipulation does not provide enough granularity to allow a robust and dynamic TE solution, so IGPs were enhanced for TE purposes [7]. Presently, BGP has yet to be enhanced to more fully support TE despite proposed modifications [16].

1.3 Motivation

Data networks have become a critical medium for constant and instant communication around the globe. Networks now carry everything from entertainment news to battlefield information for the military and require improved reliability and performance. Despite the increased demands on the Internet backbone, a solution for the efficient use of existing resources does not exist. Deploying additional bandwidth to solve congestion problems is not always a feasible or cost effective solution. Traffic engineering provides the means to leverage existing resources for congestion relief and can provide additional bandwidth to customers. However, current TE methods for BGP provide little control for ASes to manipulate ingress or egress traffic, and is performed on a trial and error basis [14]. A significant amount of progress was recently made for intradomain TE solutions, but an interdomain strategy is lacking.

BGP only advertises network layer reachability information (NLRI) in the form of a network prefix and AS path vector with no consideration for current network conditions. Simple reachability and a deterministic ranking of reachability information based on the number of AS path hops is grossly inadequate for designing robust and survivable network solutions. By expanding BGP to include TE information, the current network architecture can become more robust and reliable, as well as better able to utilize existing resources.

1.4 Connection-oriented Forwarding

Connection-oriented forwarding protocols require a session and path be established between the source and destination nodes prior to the start of communication. Once the end to end path is established, all communication between the two nodes traverses the path [8]. Connection-oriented forwarding also requires all data to arrive at the destination, in order, as sent by the source.

There are two methods for establishing the required end-to-end circuits: statically configured paths and dynamically established paths. Statically configuring a path establishes a permanent connection between the two nodes which is always in place, regardless of use. This type of path referred to as a Permanent Virtual Circuit (PVC). A dynamic connection establishes a path prior to the start of communications and tears down the path when the conversation is concluded. This type of circuit is referred to as a Switched Virtual Circuit (SVC). In order to establish an SVC, a method for signaling must be in place between the source and destination to inform each transit node of session establishment and subsequent tear down. Each participating node in the network is assigned a unique address to facilitate contact from other nodes. When one node needs to contact another, an open connection signal is transmitted to the destination, the path parameters are negotiated, and the session is established.

A common example of an SVC based network is the Plain Old Telephone Services (POTS, aka PSTN). In a POTS system, all devices, telephones, have a dedicated connection into the network. The dedicated lines are multiplexed once inside the network to allow for multiple connections to traverse the same physical path. To place a telephone call, the user lifts the source receiver which signals the network of an incoming connection request. The dialed digits inform the network of the destination device, allowing it to determines the best available path for the connection. Once the network has established a path, the destination device is informed of the incoming connection causing the phone to ring. The destination responds by lifting its receiver, establishing a dedicated connection which is maintained until the connection is broken by one of the two devices. In reality, the dedicated connection is logical over multiple physical connections [8].

Another fundamental concept of connection-oriented forwarding is the idea of

packet switching. The idea behind packet switching is to take a long message, segment the message into several smaller pieces, encapsulate each piece with routing and protocol information, and send the segments to the destination where they are reassembled [8]. Asynchronous Transfer Mode (ATM) arose from the requirement for fast multiplexing and SVC establishment as a packet switching protocol. Each SVC may involve multiple ATM switches, and each ATM switch may be a member of multiple SVCs. To recall from which neighbor an SVC's *cells*, fixed-length ATM packets, were forwarded and the identity of the appropriate next hop, every ATM switch maintains a *forwarding table*. The forwarding table contains a 1:1 mapping of the incoming interface to the outbound interface. However, not all traffic entering a switch on one interface belongs to the same SVC. As a result, ATM uses labels to designate to which SVC a cell belongs. When a cell enters a switch, the switch looks in its forwarding table for an entry with the corresponding interface and label. The entry instructs the switch which new label to use; the cell is then placed in the queue for the specified outbound interface with the new label attached. If the switch is the last along the path, the label is removed and the payload is forwarded to the final destination. Since forwarding tables are a 1:1 mapping, the time required to find an entry and determine the outbound interface is very low, allowing for a minimal processing delay within each switch the cell must traverse.

ATM inherently provides a means for reliably multiplexing several different types of services together into a single connection, as well as providing a method for prioritizing cells and bandwidth allocation. The inherent abilities make it possible to provide QoS guarantees within an ATM network.

1.5 Connectionless Forwarding

Connectionless forwarding, also known as best effort forwarding, does not rely on pre-established circuits. Rather than establishing a dedicated path prior to the commencement of communications, each forwarding node in the network examines the destination address field in each packet header, determining to which neighbor the node should forward the packet [8]. This process eliminates the overhead for path establishment and tear-down of connection-oriented models. However,

it still requires a method for calculating how to reach the appropriate next hop. To address this problem, Ethernet and IP routing algorithms were developed to address best effort forwarding on different network scales.

Ethernet originated as a local area network (LAN) standard. As the simpler ALOHA protocol, it was designed for radio communications between islands in Hawaii. Ethernet is a shared broadcast medium for all devices, so multiple devices transmitting simultaneously will cause a *collision*, resulting in both message being garbled and neither succeeding. Prior to transmission, a device must listen to the network and determine whether the channel is available; during transmission the node must continue to listen for a collision and stop if one is detected. Each device on the network is identified by a unique 48 bit identifier, known as a Media Access Control (MAC) address. Connecting multiple Ethernet networks requires a *bridge*, which is a device that maintains a MAC address forwarding table. The forwarding table allows the bridge to determine to which port a packet should be forwarded based on the destination MAC address in the Ethernet header. However, having every device connected to a huge Ethernet network would be very inefficient and would require every bridge to maintain a table listing every unique destination as in connection-oriented forwarding tables.

MAC addresses are unique to the hardware interface of a node, so they are unable to be organized in a discernible fashion. IP provides a means to identify each interface, as well as a method to organize networks to locate destinations in structured manner, and performs this task without the overhead of establishing a circuit or maintaining a huge forwarding table. Each device in an IP network is assigned a 32 bit identifier, known as an *IP address*. IP addresses on a common network are grouped into a *subnet* based on their *prefix*, or number of most significant bits they have in common [8]. The IP hierarchy makes each address spatially significant which is much more scalable than the unorganized MAC address structure.

Rather than maintaining a forwarding table for every unique destination, IP based network nodes, *routers*, maintain a list of subnets known as a *routing table*. Populated by the routing protocols discussed in Chapter 2.2, the routing table contains the destination network and the next hop for forwarding the packet. Routing tables do not require as much overhead as forwarding tables, which must list every

unique destination. Each packet must be inspected to determine the destination IP address, then the routing table is referenced to determine the most specific subnet match for the destination. Since a packet inbound on an interface could be from any network, the destination address must be compared to the entire routing table.

Neither Ethernet nor IP provide a method for ensuring in order delivery or retransmission of packets by networking devices. To provide reliable services and to dynamically determine how much bandwidth the network can handle, TCP is typically implemented in conjunction with the other two protocols. By combining TCP, IP, and Ethernet, a hop by hop network with reliability at the end-devices has become the de facto network standard.

1.6 Overlay Models and Hybrid Systems

Less than 20 years ago, router ports were expensive financially and in terms of processing delay. Switching ports were cheaper and faster, so a method for deploying IP over ATM was developed. An *overlay network* is superimposing one type of network onto a different type. For example, a single router port could be connected to a robust ATM network, allowing a router to be virtually connected to any other router also connected to the ATM network. Rather than purchasing a new router port when a new or redundant connection was required, only a configuration change adding the new ATM virtual circuit (VC) was needed [8].

Transporting IP over ATM allowed IP to leverage several benefits of ATM networks, such as limiting bandwidth and prioritizing traffic. Another major benefit of the overlay network was by using multiple ATM paths or multiple VCs, the underlying ATM network could distribute the load across multiple paths, better leveraging existing infrastructure. However, as router port cost decreased and processing power increased, the benefits of overlay methods no longer existed and the underlying ATM networks became unmanageable as the size of networks increased [8]. If a network of N routers required a full mesh, that would require $\frac{N(N-1)}{2}$ VCs to configure [8] .

A method to incorporate the beneficial properties of overlay networks without incurring the overhead cost and with current hardware was required. Multi-

protocol label switching (MPLS) was developed to assimilate the benefits of ATM into Ethernet and IP based networks. Rather than manipulating a hidden physical topology, MPLS works by creating tunnels through the existing topology which can bypass the best path selected by an IGP. The first MPLS capable router in the network will inspect an IP packet and decide which tunnel the packet should traverse. Based on the inspection, a label is attached in between the Ethernet and IP headers. Each node along the tunnel's path will use its MPLS forwarding table to inspect the label and forward the packet until the end of the tunnel is reached.

1.7 Quality of Service

QoS is the level of performance a packet or flow receives while traversing a network. The definition of performance varies according to the type of traffic; some common QoS requirements are minimizing end-to-end delay, packet loss, or jitter, or maximizing bandwidth. QoS as a requirement arose from certain flows suffering poor performance on best effort networks, where all packets are treated identically. QoS separates traffic into multiple *classes*, with each class treated differently by the network based on the network's *QoS policy*. A QoS policy defines resources allocated to each class, the type of queuing to be used, and which types of flows belong in each class.

Actually treating packets differently according to their class requires the ability to identify, separate, and provide preference for certain traffic. This capability is known as *traffic control* [17]. Traffic control is implemented by three distinct components: the packet scheduler, the classifier, and admission control. The packet scheduler is in charge of forwarding packets; placing each flow into the appropriate queue and allocating the appropriate amount of time and bandwidth to each queue according to its priority and the queuing algorithm(s) being used. The classifier may add a flow number, or tag, to each packet indicating how the packet should be handled by the scheduler. The classifier makes its decision based on the QoS policy which defines each class according to combinations of source and destination ports and IPs. Admission control is the decision making process which determines whether or not the acceptance of a new flow will adversely impact previously accepted flows [17].

To address IP's lack of QoS support, the Internet Engineering Task Force's (IETF) initial specification was the Integrated Services (IntServ) model. IntServ allows every user service requiring QoS guarantees to reserve resources along the best effort path, providing the flow with priority over other traffic equivalent to its class. Reserving resources along a path requires a signaling protocol, provided by the Resource ReSerVation Protocol (RSVP) which is explored in Chapter 2.4.1.2. For IntServ to be effective, all routers along a path must support RSVP, track reservation states, and support the IntServ model [17].

Unfortunately, the overhead requirements of having to track per flow reservations requires too much state tracking and is unscalable. As a result, the Differentiated Services (DiffServ) model was introduced as a less resource intensive alternative by eliminating the state tracking requirements. Rather than reserving resources, the edge routers mark the type of service (ToS) bits in the IP header to indicate the per hop behavior (PHB) of the packet. The new structure reduced the core router requirements by placing the burden of identifying a packet's *class of service* (CoS) and controlling resources via *policing* (strict enforcement) or *shaping* (loose enforcement) on the edge routers according to the domain's QoS policy [18]. However, DiffServ provides no explicit QoS guarantees for any particular CoS; it only provides a discernible difference in performance for the different classes. To provide the differences, DiffServ must restrict the performance of all classes, which can affect performance once a class approaches its restricted limits. For example, packets in the express forwarding class can have the same performance as best effort if the express forwarding queue is congested. Despite its shortcomings, DiffServ is the model upon which most current QoS implementations are based.

1.8 Traffic Engineering

TE is the aspect of network engineering that addresses the issues of both traffic flow performance evaluation and the optimization of IP networks [19]. The goal of traffic engineering is to "facilitate efficient and reliable network operations while simultaneously optimizing network resource utilization and traffic performance" [7]. Traffic performance optimization focuses on enhancing the QoS of traffic streams, which for a single stream may include the minimization of packet loss, minimiza-

tion of delay, maximization of throughput, minimizing jitter, or a combination of factors. Optimizing resource management attempts to efficiently manage network resources, such as bandwidth, in an attempt to minimize congestion and over-utilization [7].

TE is concerned with QoS, but it attempts to optimize resource management which may allow QoS goals to be better achieved. By leveraging network feedback and the status of available resources, TE can map a flow onto a path that satisfies the flow's QoS requirements rather than simply following the best path as selected by an IGP such as OSPF or IS-IS. However, traditional protocols do not provide such an intricate level of detail, so IGPs were modified to distribute a more detailed network status for use by TE algorithms. The TE algorithms select a path based on the new information and attempt to balance the requirement for optimizing resources as well as achieving the QoS requirements of a flow. Achieving these goals inherently requires the ability to use more than one path within the network. Once again, traditional IGPs are not capable of fully supporting this requirement as they select only a single best path. MPLS is used in conjunction with TE as the transportation model for creating connection-oriented paths based on the best traffic engineered path.

1.9 Thesis Overview

Within an AS, optimizing the flow of traffic is possible by an IGP with TE properties and MPLS. However, TE at the Internet level is extremely complex. Traditional interdomain routing decisions are either locally optimal or based on the domain's policy, which may be far from optimal. There is a requirement for sharing TE information across domain boundaries, but a scalable method for sharing TE information has not been proposed. The proposal must also satisfy ISPs requirements for internal topology secrecy, which inhibits optimal path selection.

This thesis explores a method to share TE attributes across domain boundaries by extending the proposed internal BGP advertisements proposed in [16]. By extending this proposition to all BGP advertisements, both iBGP and external BGP (eBGP), up to date TE information depicting the quality of a path across the Internet can be advertised in a scalable manner. Combining TE information with

inter-domain MPLS path reservations, as proposed by [3, 20, 21, 22], intelligent inter-domain TE is attainable.

Although this modification is not globally optimal, it is locally optimal with respect to each AS the path traverses. In order to actually achieve global optimality, the full sharing of information would be required, which is not currently possible or likely to happen in the foreseeable future. The information would also have to be verifiable; solutions for security are available to verify AS paths, prefix ownership, etc., and could be extended to include traffic engineering information.

In Chapter 2, we explore traditional intra-domain routing and explore IGPs. Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS) are explored in detail, as well as their extensions for sharing TE information within a domain. To conclude our exploration of intra-domain routing and traffic engineering, we examine MPLS, how it works, and how it is utilized for TE purposes.

Chapter 3 examines traditional BGP, the de facto inter-domain routing protocol, which provides reachability information for the backbone of the Internet. Methods to leverage BGP for traffic engineering are also explored, along with its performance and limitations. MPLS extensions to establish inter-domain labeled switched paths (LSPs) are also examined.

Chapter 4 explains our proposal for the advertising requirements of an inter-domain BGP-TE attribute. We also propose a step by step process for determining the advertised TE values and propose a path selection process from the source networks perspective. We also survey the current state of the art for proposed interdomain TE advertisements.

Chapter 5 tests the scalability of the BGP-TE attribute against traditional BGP advertisements. We test the different protocol modifications based on current Internet level topology and prefix information obtained from RouteViews [23].

Chapter 2

Background on Intradomain Path Selection

2.1 Overview

The Internet consists of over 26,000 autonomous systems (ASes) of varying sizes and topologies. Internally, each AS manages its portion of the network according to its own policies, priorities, and objectives. The set of factors contributing to the management requirements may vary from domain to domain, so to satisfy the diverse requirements for managing an AS, multiple interior gateway protocols (IGPs) are available.

To efficiently route traffic, every router within a domain must have an accurate picture of the network; the purpose of an IGP is to construct the accurate picture and to select the best path based on the advertised picture. Periodically, each router will send updates to ensure all routers in the network are up to date and able to accurately determine the best path to reach each destination. To calculate the best path, each IGP must have a method of ranking links or determining the cost of traversing a link.

IGPs assign a numerical value, known as a *metric*, to each link. A metric consists of one or more measurable network attributes, which may be combined in some manner if more than one is used. Also, metrics may be either statically assigned or may be reassigned if they are dependent upon network status. Once

a metric is calculated for each link, the routers within the domain exchange information according to their IGP. The IGP determines the best path for reaching any destination from every router's point of view. IGPs may consider an interface's maximum bandwidth capacity, interface utilization (current queue utilization or utilized bandwidth), end to end or propagation delay, available bandwidth capacity, reliability, or hop count as possible metrics.

The simple best path algorithms and metrics satisfy the network requirements for end to end best effort services. However, as Quality of Service (QoS) and traffic engineering (TE) requirements surfaced, the traditional IGPs were extended to include additional link attributes to enable traffic engineering. Extending IGPs to advertise additional interface attributes provides the potential to use constraint-based routing algorithms for creating traffic engineered label switched paths (LSPs) for multi-protocol label switching (MPLS), bypassing the shortfalls of IGPs. MPLS provides the means to forward traffic based on priority, leveraging multiple network paths for congestion control and avoidance, redundancy, and fast failover recovery.

2.2 Traditional Intra-Domain Routing

The most basic method to establish communication between two routers is to statically configure each to route packets. Manually configuring entries into the routing table associates destinations with the appropriate interface to use for outbound traffic. Static routing is simple and easy to configure; the ability to configure static routes with weights allows for redundant and hierarchical static route structure to be configured in case of failure.

Despite its simplicity, static routing has some serious drawbacks. Even if multiple static routes are configured, should a scenario occur that a router is not preconfigured to handle, the router has no method for recovering. Static routing becomes complicated to configure and maintain the larger a network grows. Also, in the case of a topology change, the entire network may need to be reconfigured. In order to decrease configuration complexity, a dynamic method for route selection was developed. A dynamic and reactive protocol requires a significantly more complex and intelligent protocol; the ability to discover, select, and maintain network paths are requirements. The methods for performing these three tasks differ

amongst the various IGPs, and they are broken into two categories: distance-vector (DV) and link-state (LS).

Although they have several differences at the core of how they operate, all dynamic routing protocols have several aspects in common. All dynamic protocols require a method to discover neighbors, which is typically performed in a per-interface manner. By configuring which subnets are eligible to transmit or receive protocol specific packets, the routing protocol is informed of which interfaces may be used to establish an *adjacency* since each interface has a specific Internet Protocol (IP) address [24]. An adjacency is a stateful connection to a directly connected neighbor for exchanging protocol specific information. Once all information is exchanged, the protocol considers the routers to be adjacent.

Sending updates only at regular intervals does not inform the network of changes in a timely manner. If a change occurs in the network topology, a router needs to inform all appropriate parties in a timely manner. For this purpose, routing protocols also rely on *events* to initiate *triggered updates*. Triggered updates occur in response to an event which other routers need to be informed of immediately. Events such as link or neighbor status change must be propagated to ensure all nodes in the network have accurate information.

The final commonality amongst IGPs is the idea of *convergence*. Convergence is the state achieved when all nodes in a network possess a non-fluctuating view of the network. The amount of time required for convergence varies greatly depending on the type of IGP. We now examine the differences between the IGPs with respect to their path selection processes, strengths, weaknesses, and convergence times.

2.2.1 Distance-vector Protocols

Distance-vector routing is a router by router decision making process and is ordinarily based on the Bellman-Ford algorithm. The Bellman-Ford algorithm inputs a graph with V vertices (nodes), each with a predecessor initially set to $NULL$, and E edges (links) which each have an associated distance. Initially, the distance to every node is set to infinity, ∞, except for the distance to the source or root node, v, which is set to 0. The algorithm then examines every edge in the network to determine whether or not the distance to the source of the edge is less than

infinity. During the first iteration, only the distance to v is less than infinity, so every edge attached to v will qualify to be examined. More generally, the destination vertices of qualifying edges are assigned a new distance, which is equal to the distance associated with the edge connecting them to the source vertice of the edge. The destination vertice's predecessor is then set to the source of the edge.

This process is repeated $V-1$ times; every pass, if an edge connects two vertices whose distance is already less than ∞, the old distance and the new distance are compared; the lower distance is assigned and the predecessor is set to whichever source is associated with the edge.

DV routing algorithms use a distributed form of the Bellman-Ford algorithm. In a network setting, nodes do not have knowledge of the entire graph, or *network topology*. The only knowledge each router in a DV environment possesses is the distance to reach each of its neighbors [2]. Each node must maintain a list of destinations, or routing table, containing the destination network, next-hop to reach the destination (equivalent to the predecessor in the true Bellman-Ford algorithm), and the cost to reach the destination. To determine the cost to reach any destination, a router needs to know the destination exists; since the topology is not known, new routes are learned in an incremental fashion.

When a router in a DV network begins the routing process, it assigns a cost of 0 to all locally attached networks. After the router has found at least one neighbor, it advertises its entire routing table to each of its neighbors. When a router receives an advertisement, it needs to determine whether or not the paths contained in the advertisement are better than those currently residing in its routing table. To decide which path is best, the router adds the cost contained in the advertisement to the cost of the link connecting it to the neighbor that sent the advertisement. If an entry for the destination already exists in the routing table, the two costs are compared, the path with the lowest cost is selected, and the next-hop is updated as necessary.

A source for disagreement amongst the different routing protocols is what to choose as the metric. Within an AS, the primary concerns when traversing a network may vary. Typical metrics of concern are number of hops, end to end delay, and bandwidth capacity. Originally, when the bandwidth capacity of links in a network were low, minimizing the number of hops was equivalent to minimizing

end to end delay and limiting the amount of network resources consumed. One of the first routing protocols, Routing Information Protocol (RIP), uses hop count as its metric. When a router originates a network advertisement, it sets the cost to 0; each subsequent router who receives the advertisement increments the cost and sends the advertisement to its neighbors. RIP sets a cap on the number of hops a path could contain at 15 [2].

One benefit of DV protocols is every router in a network may not need to be informed of a topology change. However, DV protocols are a hop by hop implementation, full convergence of a network may be time consuming and require a significant amount of message passing to complete. Due to varying timers and each node knowing a different portion of the network topology, any two neighbors may require several routing table exchanges in order to achieve convergence. If there is a large amount of fluctuation in the network, such as a link constantly changing state which is known as *route-flapping*, all routers in the network may need to exchange routing tables multiple times.

The simplicity of DV protocols can also lead to a potential problem known as *count to infinity*. Every DV router generates a view of the network based on the knowledge it possesses at any given moment in time, which it then sends to all of its neighbors. However, if a router advertises a route back to the neighbor from which it received the route, a routing loop is encountered. The routers would advertise the path back and forth until the metric reaches the maximum allowable, which is known as count to infinity. There are three methods to combat this problem. The first is called *split horizon*. Split horizon prohibits a router from advertising a path to a neighbor for which the neighbor is the next hop, thus preventing a router from propagating bad information or from initiating routing loops. Another prevention method is split horizon with poison reverse. In this version of split horizon, routers learned from a neighbor are sent back to the neighbor, but the cost of the path is set to ∞ (15 in the case of RIP). Poison reverse ensures the loop cannot be created by having a router explicitly informing the router through which it learned a route it has no other way to reach the specific destination. The final method for loop prevention has already been mentioned, but triggered updates are an important method to preventing such routing loops. A router must determine what the failure means to its routing table; for instance, if a link fails, then all paths learned via

that link are no longer accessible, so all neighbors should be informed immediately. Triggered updates combined with one of the two split horizon variants are a very effective method for reducing the the risk of routing loops in a DV protocol.

2.2.2 Link-state Protocols

LS protocols draw their name from the the requirement for every router in the network to know the status of every link. While DV protocols are based on Bellman-Ford, LS protocols are based on Dijkstra's algorithm for computing paths. Dijkstra's algorithm also uses a connected graph with V vertices (nodes) and E edges (links), and uses this information to construct a tree of shortest paths between the source node, v, and all other nodes in the graph. The edges and vertices are divided into three and two sets respectively:

I	Edges assigned as a branch in the tree
II	Edges eligible to be selected for placement in set I
III	All other edges
A	The vertices connected by Edges in set I
B	All other vertices

Initially sets I, II, and A are empty. The node to act as v is arbitrarily selected and the process begins by placing all edges for which v is a vertice into set II. The iterative process begins by selecting the shortest link in set II and placing it into set I, which causes a vertice, v_n, to be moved from B to A. The edges connected to v_n need to be moved into set II, but must be compared with current members of set II. If two edges share a vertice which is not in set A, the edge that results in the shortest distance to reach the unknown vertice is selected and placed in set II; the longer edge will be placed in set III. The iterative process continues until sets II and B are exhausted [25]. A typical cost for traversing a link is a constant times the inverse bandwidth capacity of the link, $\frac{X}{Bandwidth}$.

There are two primary LS protocols: Open Shortest Path First (OSPF) and Intermediate-System to Intermediate-System (IS-IS). The two protocols are similar except for the hierarchical fashion in which they organize routers and different sections of the network. Because they are so similar, we will focus on OSPF and

LSA Type	Description
1	Router-LSA. Describes what other routers a node is connected to by describing all the links the node is connected to. There are four types of Links: Point to Point (includes neighbor router ID), Link to transit network (includes IP of DR), link to stub network (IP Prefix), and Virtual Link (neighbor router ID)
2	Network-LSA. The DR on a network list which routers are attached to a network segment. It must have at least two routers attached.
3	Summary-LSA. An ABR sumarizes the networks inside its area for distribution into other areas.
4	Summary-LSA. An ABR provides information detailing which router created a Type-5 LSA.
5	External-LSA. Any information imported into OSPF by an external protocol or AS.
6	Multicast Group Membership-LSA. Used by Multicast OSPF (MOSPF)
7	NSSA-External-LSA. Generated by an ASBR within an NSSA. Converted to a Type 5 at the ABR before being flooded into another area.
8	Link-LSA. LSA local only to the attached link. Used by IPv6 to inform other routers on a link of a router's IP address, the list of prefixes associated with the link, and collects options bits.
9--11	Opaque-LSA. Each has a different flooding scope. Used by optional OSPF extensions, such as OSPF-TE, that all routers must forward even if they do not understand the information contained within the LSA.

Figure 2.1. The 11 OSPF LSA types

will describe typical LS behavior in OSPF terms. A link state advertisement (LSA) is a packet of data describing a router, link, or network, and is used to inform other LS routers of the device's current state [24]. OSPF defines 11 types of LSAs, listed and defined in Fig. 2.1 [24, 26, 27, 28, 29].

An *area* is a grouping of networks whose topology is hidden from the rest of the AS. Areas are used to partition the network into manageable sections and to restrict the distribution scope of LSAs. Other areas in the network must be aware of the networks within a given area but do not need to know the internal topology of every area, so each area contains its own shortest path tree. To distribute network information, a router bordering more than one area, called an Area Border Routers (ABR), will send the appropriate update messages to the rest of the network. The types of LSAs an ABR is allowed to advertise depends on the types of areas for

Area Type	Description and Restrictions
Backbone	Defined as "area 0." Every area in the AS needs to have a direct connection to Area 0. It is used to relay information between areas.
Stub	An area with no connections to another AS. It uses an area default route to send traffic outside the area. No Type 5 LSAs are allowed to be advertised.
Not So Stubby Area (NSSA)	Similar to stub areas, but it may have connections to external Ases. External type 7 LSAs may be generated inside the area.

Figure 2.2. OSPF areas and the restrictions on LSA advertisements into or out of the area type.

which it is acting as an ABR. OSPF defines three area types designated by the types of LSAs they are allowed to advertise. See Fig. 2.2.

Since every router is aware of each link's status, every node maintains an identical database listing each node, network, and interconnect. This database is used generate the directional graph of shortest paths. To populate the LS database, OSPF routers establish adjacencies with their neighbors to exchange database information. Adjacency establishment is a multistep process, commencing with the exchange of *hello packets*. After initiating an adjacency with hello packets, the next step is to exchange LS database descriptions. The database descriptions inform neighbors of what a router knows, the age of the information, as well as a list of neighbors with whom the router is fully adjacent. Once the database exchange process is complete, each router compares the received database description to its own database; if it does not have the information or if the neighbor has newer data, the router will request an update containing the full details of the missing/old data. After all requests are successfully filled, the databases are in sync, the routers are considered fully adjacent, and can reflect this in their individual database.

It is imperative that every node in the network maintain an identical picture of the network. However, OSPF relaxes this requirement by allowing a router to remain in sync with its neighbors. Although synchronization is a neighbor to neighbor process, the process for updating routers of state changes and selecting the best paths through the network are drastically different than DV. LS routers do not rely directly on their neighbors for best path selection. Rather, they rely on them to receive updates about the current state of the network. This dependence

leads to one of the most important LS aspects, which is *LSA flooding*. Flooding quickly updates every node in the network of a status change to facilitate a timely recalculation of the best path algorithm by synchronizing neighbor databases. An LSA update message is used to start the flooding process on each router. Only 10 incidents can trigger the creation of an LSA update, including an interface state change or neighbor adjacency change to or from fully adjacent. To ensure database synchronization, the network must ensure that every router receives every LSA update.

The generation of an LSA update requires the originating router to decide which of its neighbors are eligible to receive the update. The restrictions on eligibility mainly refer to area borders since only certain LSA updates can cross area boundaries. For conciseness, the primary restriction we are concerned with is whether or not the LSA was received on a particular interface and will focus only on the LSAs within a single area. The originating router transmits the LSA update out all eligible interfaces and tracks whether or not each neighbor acknowledges receipt of the LSA.

When an LSA update is received, a router stores which interface received the update and immediately checks whether the update LSA is newer than the version stored in its database. If the LSA is newer, or it does not have a copy of the LSA in its database, it immediately broadcasts the LSA update on all its eligible interfaces. The router then removes the old LSA and installs the LSA contained in the update into its database. Acknowledging the LSA can either be done immediately or after a certain delay. The purpose for the delay is to possibly combine LSA acknowledgments or to send the LSA acknowledgment to multiple neighbors at once. An immediate acknowledgment is sent if the LSA update is a duplicate; otherwise, a delayed acknowledgment is sent [24].

LS protocols converge quickly, especially if there are few routers in an area. However, any topology change in the network requires every router in to be informed which can incur significant overhead. Limiting how often a router can generate LSAs helps to limit the number of updates.

2.3 TE and QoS Extensions

IGPs are not omniscient; shortest path protocols are typically based on an additive metric, obtained from an administratively configured value. Based on the best path selected by an IGP, traffic converges onto a few primary links or routers, whether or not there is sufficient bandwidth to support additional traffic [7]. Routing based on QoS requirements, or *constraint-based routing*, seeks to find a path between the source and destination that will satisfy a flow's requirements [30]. Routing decisions are based on the attributes of an incoming flow, the current network resource status, and other topology information [7]. IGP metrics are not sophisticated enough for making constraint-based decisions. However, IGPs are deployed on every router in the network and are in a perfect position to record and distribute network status. The distributed information can be used for some type of reservation method to ensure QoS for a flow.

2.3.1 OSPF QoS Extensions

OSPF's QoS extension (OSPF-QoS) addresses the concerns of QoS based routing by distributing the information necessary to determine whether or not a path will satisfy a flow's QoS requirements. The metrics distributed by OSPF-QoS are the link's available bandwidth, path hop-count, and link propagation delay. All flows are concerned with being allocated enough bandwidth to support their requirements, so in order for a path to be eligible it must have at least the required amount of bandwidth available. Propagation delay is used to prune high latency links, such as satellite based communications, from consideration for flows with low tolerance for delay. Hop-count is included as traversing fewer nodes is preferable. The delay and available bandwidth are encoded both encoded as a 16 bit floating point value, which requires an extension to the Type 1 LSA [30].

Two methods for selecting paths are proposed in [30], the first of which is based on the Bellman-Ford algorithm. Hop-count is not distributed because it can be inferred from the number of iterations required to reach a destination in the Bellman-Ford algorithm based on the full topology information. The Bellman-Ford version is used in the situation where the QoS best paths need to be calculated by the algorithm prior to a request's arrival. The algorithm performs the calculations

by finding the path with the highest amount of available bandwidth. At each iteration of the algorithm, the maximum bandwidth to all available paths is stored.

The Dijkstra's algorithm version computes QoS paths on demand. The first step is to prune all edges that have less than the requested bandwidth. At that point, the minimum hop-count is used as the metric for calculating the best path [30].

2.3.2 TE Extensions

The goal of OSPF-QoS was to distribute information relevant to making QoS based routing decisions. However, the protocol assumed that the Resource ReSerVation Protocol (RSVP) was implemented as the underlying reservation protocol and did not make any assumptions about how to actually maintain the reserved path. The development of MPLS traffic engineering (MPLS-TE) required the distribution of different link attributes.

Both OSPF and IS-IS were enhanced for describing and distributing topology information within an area for making MPLS-TE path decisions. By providing a status of the current reservation and utilization state of point-to-point links, the extensions provide the ability to monitor extended link attributes, provide accurate information for constraint-based routing, and for intra-AS level traffic engineering. OSPF and IS-IS extensions contain the same attributes, albeit in a slightly different order [1, 31], so we focus on the OSPF version only. We refer to the extension as OSPF-TE for the rest of the document.

The OSPF-TE LSA contains nine attributes for describing the TE attributes of a link. The first attribute is the one-byte link-type attribute, which can either be point-to-point or multi-access. The link-ID is a four-byte value that specifies the router ID of the remote router for a point-to-point LSA. The next two attributes are the local router and remote router's IP address. A TE metric is also included, which is a four byte value assigned by the network administrator.

The following set of attributes are all bandwidth measurements and are four byte IEEE floating point values, representing bytes per second. The maximum bandwidth indicates the maximum bandwidth capacity of the link. The maximum reservable bandwidth specifies the maximum total bandwidth that may be

reserved for this link. The unreserved bandwidth attribute indicates the amount of bandwidth still available at each priority level, and each of the eight values must be less than or equal to the maximum reservable bandwidth. The final value is the administrative group which is assigned by the network administrator, and is used to cluster resources together into classes [1].

In OSPF, LSAs are typically distributed by the designated router (DR) of a link. However, the available bandwidth on a link may differ dependent upon the direction of travel across the link. To overcome this issue, each OSPF-TE LSA is also directional. Every router must describe the status of the link from its point of view in the network [1]. That is to say that each router will advertise an OSPF-TE LSA describing the unreserved bandwidth the router is able to transmit on a link.

These attributes are distributed via traditional OSPF methods, but their use is not defined. A constraint-based routing algorithms can leverage the information to select network paths and to reserve resources via MPLS.

2.3.3 Performance

The advantages of using a traditional routing protocol are the stability, dependability, low overhead, and compatibility they offer. It is even possible to perform traffic engineering with traditional routing protocols. By managing the weights associated with each link, IGPs are able to distribute traffic more evenly across the network, rather than funneling traffic across low cost links. Achieving the TE goals of the network with traditional routing protocols requires a three step process. Measuring the traffic demands placed on the network provides the basis for modeling IGP weight changes and the resulting path selection and traffic patterns. Link weights are calculated by an off-line process that analyzes traffic over a given period of time to determine what the optimal link weights are, given the traffic patterns. The final step is to modify the link weights of the network [15].

Even though traditional IGPs are able to perform TE functions, optimal link settings are unable to react dynamically to changes in topology or shifts in traffic patterns. To shift traffic within a traditional IGP network, the traditional protocols still require modifying the link weights either by manually changing them or with a centralized monitoring system that can accurately monitor traffic patterns or

efficiently react to loads within the network. However, modifying link weights frequently can cause instability within the routing protocol.

Algorithms using the information provided by OSPF-TE are able to react dynamically to changes in network status, including topology changes or failures. The TE extensions are also able to eliminate the burden of changing LS weights by distributing information for use by a constraint-based routing protocol. However, the increased view of the network comes with an increase in overhead. The trade-off is a balancing act between accuracy and overhead. One example is how up to date a TE routing table must be in order to accurately select the best path for a new flow. If the routers were updated with every change in reservation status, a significant number of LSAs would be generated, creating a huge increase in protocol overhead. However, relatively good performance can still be achieved by having a less up to date view of the current network status. There are methods for reducing the overhead incurred as a result of the trade-off problem. Periodic updates, only generating LSAs when the reservation status exceeds a threshold, and if a reservation causes a link to go beyond a certain maximum bandwidth threshold can all assist in reducing the number of LSA updates to an acceptable level [32].

2.4 MPLS

Asynchronous Transfer Mode (ATM) provides a robust, fast switching network that inherently supports a variety of services. The multiplexing capabilities of ATM allows a single desktop network connection to support simultaneous voice, video, and data transfers. ATM also supports QoS; its fixed packet length, combined with packet prioritization, allows the simultaneous transfer of multiple traffic types without causing degradation to time sensitive data, such as voice traffic.

Another advantage of ATM is the use of a forwarding table; packets experience less processing delay at each network node resulting in reduced latency. Forwarding table use also allows for multiple paths within a network. The 1:1 mapping of interface and label pairs for ingress and egress traffic means traffic traveling towards the same destination, even from a single source, can have multiple labels, resulting in multiple paths to a destination.

Despite ATM's inherent support for a large set of capabilities, including QoS, virtual networking, and bandwidth guarantees, ATM failed to achieve market dominance. The combination of lower cost and the simple plug and play nature of Ethernet based networks allowed for it to become the more widely deployed Local Area Network (LAN) standard. Ethernet has also extended to the Wide Area Network (WAN) as a transport mechanism with speeds approaching 10 gigabits per second (Gbps). The TCP/IP/Ethernet stack of protocols prevalent in most networks does not provide the same array of services which are native to ATM. The lack of capabilities was not an issue when Ethernet based networks carried only data traffic, but with voice traffic migrating onto the network the lack became evident [8].

Multi-protocol Label Switching (MPLS) was the response for providing the services of ATM on, but not limited to, IP based networks. In traditional IP forwarding, each router must inspect the header of every IP packet in order to determine the appropriate next hop. Depending on a networks QoS policy, the router may need to inspect not only the layer 3 header, but the layer 4 header as well to determine the type of traffic. This inspection process incurs processing delay and must be performed for every single packet. The labeling process of MPLS virtually eliminates the header inspection requirement. The first router along an LSP, the label edge router (LER), inspects the headers in the typical manner. However, the LER inserts an MPLS header matching the appropriate next hop according the forwarding equivalence class (FEC) a packet matches. At each subsequent hop, the label switching router (LSR) inspects the label, looks up the new label and outbound interface according to the entry in its forwarding table, and forwards the packet along the LSP. [33].

An MPLS label is a fixed length, 4 byte, locally significant identifier of a FEC which contains a 20-bit label value, a 3-bit QoS priority level, and an 8-bit time to live value. The label is known as a *shim* header due to its placement. The shim is placed between the traditional layer 2 and layer 3 headers, and a packet can have multiple shims allowing for MPLS based tunnels [33].

Reserving network resources and distributing labels is required in order for MPLS to properly provide the QoS and multipath functions of ATM. There are two protocols, Resource ReSerVation Protocol RSVP and Label Distribution Protocol

(LDP), which perform these functions for MPLS, albeit in two distinctly different manners.

2.4.1 Label Distribution

Each LSR needs to exchange label information with its neighbors in order to create the 1:1 entries in its forwarding table. The neighboring LSRs need to agree upon the meaning of each label in order to forward traffic, which is the function of a label distribution protocol [33].

2.4.1.1 LDP

LDP employs a pro-active approach to label distribution. Labels may be requested during LSP establishment, but LDP provides a method to bind labels to an interface between LDP peers prior to a requirement for the path.

Each LDP enabled LSR on a network must go through a discovery process, during which it establishes a *hello adjacency* and a *LDP session* with discovered peers. An LSR will transmit a *LDP hello message* on each enabled interface containing the LDP identifier for the label space it prefers to use. Once each peer has received a hello message, the hello adjacency is established and triggers the establishment of an LDP session. The newly adjacent neighbors open a TCP connection and negotiate the label distribution method, timer values, and link layer specific information if necessary. Each connection must be maintained with periodic message exchanges [34].

After the session establishment, label advertisement is performed. There are two methods for label advertisement, which primarily differ according to which LSR is responsible for initiating the mapping requests and advertisements. As such, it is useful to describe LSR decisions in terms of *upstream* and *downstream* LSRs. An LSR closer to the source along an LSP for a given FEC is referred to as an upstream LSR. LSRs closer to the destination along the LSP are referred to as downstream LSRs [33]. In *Downstream on Demand Label Advertisement*, the upstream LSR is responsible for requesting label mappings from the downstream LSR. *Downstream Unsolicited Label Advertisement* makes the downstream LSR responsible for advertising label mappings. The downstream LSR will select a la-

bel it wishes an upstream router to use for a given destination, whether or not the upstream LSR requested a label to the destination [34]. The mapping advertisements and requests are communicated via *label mapping messages* and *label request messages*.

Along with labeling the LSP, the type of traffic allowed to traverse the LSP must be defined. To inform neighbor LSRs of eligible traffic, the LDP mapping and request messages include the LSP's FEC. Traffic is defined based on FEC elements which are defined by LDP as either a host address or an address prefix. It is possible for an LSR to have multiple FECs that match an incoming packet, but how to separate the traffic onto those LSPs is beyond the scope of LDP [34].

Despite being quite thorough, LDP has several shortfalls. The first is its inability to define FECs in a more fine-grained manner; the use of host address allows for splitting communication between two destinations onto multiple paths, it but does not allow for separating traffic types onto different LSPs. Another issue is that every LSR in the network must know the complete FEC for every LSP to which it belongs, creating a large amount of overhead during label distribution. The final issue is LDP only provides a hop by hop distribution protocol; there is no procedure for specifying or requesting an end-to-end LSP through the network.

2.4.1.2 RSVP

RSVP was originally developed for use with the Integrated Services based approach. The original design of RSVP was for a host to request specific QoS guarantees from the network for a data stream and act as a control protocol prior to the initiation of traffic transmission from a source to a destination [8].

RSVP operates by communicating with the QoS admission control component, briefly mentioned in Chapter 1.7, as well as an optional QoS component: the policy control module. The admission control module determines whether or not there are sufficient resources available on the node; the policy control module determines whether the requester has sufficient privileges to make the request [35]. Both decision modules must approve a request, otherwise it is rejected.

RSVP requests are unidirectional and receiver-oriented; the receiver of a data flow initiates and maintains the flow's reservation [35]. However, since paths between any given source and destination may be asymmetrical, a series of message

exchanges must take place in order for the sender to reserve resources along the path to the receiver. The sender and receiver communicate via an out-of-band method to agree a path is required and the value of its attributes. The sender initiates the message exchange by sending a *path message* to the receiver, containing the traffic requirements and the *destination descriptor*. The descriptor consists of the destination IP, protocol ID, and an optional destination port. Along the path traveled by the path message, each node will record next-hop information, which is the node from whom it received the path message. The next-hop record provides the reverse path for the receiver to send its *resv message*, which requests the reservation of resources. As the resv message travels the reverse path, each node performs the admission and policy control checks, reserves the resources if approved, and forwards the resv message to the next-hop.

2.4.2 TE Extensions

Although not widely used in its original format, RSVP possessed multiple characteristics to perform label distribution for constraint-based routing and traffic engineering. LDP also displayed several shortcomings, as mentioned earlier, but was an actual label distribution method. Both were extended to support traffic engineering in the form of CR-LDP and RSVP-TE. Since CR-LDP and RSVP-TE provide identical functionality [8], this section will focus on RSVP-TE and its operations.

RSVP-TE specifies that an LSP is defined by the label applied at the first LSR along the path, the LER, which allows LSPs to be treated as tunnels below the normal IP routing and filtering mechanisms [36]. Based on this definition, the tunnels can have multiple policies applied to them as related to network optimization. Separating traffic, creating multiple paths, avoiding congestion and network failure are all achievable. To provide this capability, the path and resv messages were modified to include label requests and labels, respectively.

Another important function in RSVP-TE is the ability of the LER to request explicitly routed LSPs by including an explicit route object (ERO) in the path message. Based on network information in an LER's possession, it may calculate a path through the network which optimizes network resources or satisfies certain

QoS requirements for the incoming flow. The ERO specifies the LSP as a sequence of abstract nodes, which allows a node to be either an IP address or an AS. There are two types of EROs: strict and loose. A strict request specifies every transit node for the LSP while a loose specification only identifies nodes that must be included in the path. Using explicit paths and RSVP-TE allows LERs to perform TE based on the current network status and the QoS requirements of incoming flows. The ability to separate traffic based on both IP and ports allows for a finer granularity to separate traffic and optimize paths [36].

RSVP-TE can also exclude specific abstract nodes from consideration along a path via an exclude route object (XRO). This is useful when a router is performing multipath calculations and wants to maintain disjoint paths [37].

One advantage of RSVP-TE over LDP is FECs are only required at the LERs, reducing overhead for transit LSRs as well as for the network. The burden of mapping a flow onto an LSP is placed primarily on the LER and transit LSRs simply function as forwarding devices.

2.4.3 MPLS-TE Path Selection

The TE goals of congestion avoidance, improved resources utilization, and providing QoS guarantees can be achieved by mapping traffic flows onto a physical topology via MPLS-TE with resource reservations, multiple priority levels, and explicit path requests [38]. A critical aspect of ensuring MPLS-TE satisfies the goals of traffic engineering is path selection.

To satisfy QoS requirements, an MPLS path selection algorithm must consider the flow's QoS requirements, not the simple fixed value metric used for traditional IGPs. The TE extensions to IGPs, discussed in Chapter 2.4.2, allows each router to build a current picture of network resources from a TE perspective and to track the dynamic nature of these values. However, leveraging these new values for some type of best path selection algorithm presents a challenging set of problems.

2.4.3.1 Single Path Selection

Selecting a path that is not necessarily the shortest path introduces additional complexity into route calculation algorithms. The increased complexity is based

on the number of metrics used to select the path, so multiple strategies to reduce the complexity of the route selection process were proposed. The path computation and complexity reduction techniques typically fall into three categories. The first is to use multiple metrics, such as bandwidth and an additional metric, where paths are optimized based on the additional metric and bandwidth is used to prune unusable links. Another approach combines multiple metrics to produce one overall metric. The final method is to use heuristic models, which typically use a centralized and all knowing node to calculate optimal network paths based on previously recorded traffic [39].

Each method has its own set of complexities and difficulties. The multiple metric method can typically only use two metrics, otherwise the complexity of the solution becomes too great. Combining metrics into one value causes the algorithm to lose site of actual attribute values and can allow the selection of a less that adequate path, even though multiple metrics may be used with an acceptable level of complexity. Heuristic methods tend to make assumptions about the network topology and traffic loads which may not always be accurate and they create central points of failure for a network. Another point against the offline method is a single request may cause all LSPs to be recalculated, shifting all traffic on the entire network [39].

Hybrid methods can produce the benefits of mixed metrics and the multiple metric method. A hybrid method can effectively use three metrics; one metric can be used to trim routes and the other two can be combined in a mixed metric fashion, allowing the use of multiple metrics with reduced complexity. However, current constraint-based algorithms incorporate available bandwidth from the TE advertisements as their primary metric when calculating a shortest path.

2.4.3.2 Multipath Selection

Unlike traditional IGPs, MPLS provides the capability for data packets to traverse separate and unequal paths. A multipath solution may be required by the network if the the request cannot be filled by a single reservation due to network saturation, the size of the request is simply too large, or to split a flow that is causing congestion. In addition to the complexity of single path selection based on multiple metrics, calculating multipath selection has the additional burden of determining

the number of paths required. Another consideration is selecting disjoint paths; the potential failure of a single LSR or link would cause the failure of multiple LSPs which is unacceptable if avoidable. The solution to each concern depends on which multipath approach is implemented.

One multipath approach aims to prevent network saturation and congestion by distributing flows evenly around the network onto eligible links in an effort to keep overall link utilization low. Although this approach will result in a more even distribution of traffic, the non optimal paths may in fact consume a higher percentage of the overall available network bandwidth due to their longer path length [40]. Other approaches attempt to minimize the impact on the network by requesting the fewest possible LSPs to satisfy the request. Yet another is to split traffic in an attempt to minimize the maximum link utilization within the network [41]. By minimizing some aspect of the network, a multipath algorithm can ensure the overall health of the network and resource preservation will be maximized.

Chapter 3

Interdomain Path Selection

3.1 Background

Intradomain traffic engineering (TE) focuses on optimizing traffic flow inside a given domain with respect to network resources. Its solutions are based on the assumption all routers within the domain openly share as much information as possible without degrading network performance. How much information and how to leverage the information were primary concerns for IGPs and constrained intradomain routing. Interdomain path selection also seeks to provide efficient routing of packets and resources management, but it must perform its functions in a completely different manner. A router can belong to only one autonomous system (AS), so any node directly connected to a router in a different AS is known as an AS border router (ASBR). Due to the sheer size of the Internet, over 26,000 ASes and 250,000 individual routes [42], ASBRs cannot know the Internet's full topology. The Border Gateway Protocol (BGP) was developed to overcome the scalability issues of sharing information on such a large scale while maintaining efficient routing of traffic. Rather than exchanging the full details of the network topology like an Interior Gateway Protocol (IGP), BGP focuses on exchanging network layer reachability information (NLRI) and the number of AS hops required to reach the destination network.

BGP is able to perform TE based on the manipulation of its metrics and by enforcing a domain's advertisement policy. To facilitate traffic engineering at the Internet level, label switched path (LSP) reservation for multi-protocol label

switching (MPLS) was extended to create interdomain LSPs.

3.2 BGP

BGP is a distance vector protocol designed for scalability at the Internet level. Scalability is achieved by only distributing NLRI for a subnet and the list of transit ASes for reaching the advertised destination network [4]. An ASBR exchanges NLRI advertisements with its peers, which come in two flavors: internal BGP (iBGP) and external BGP (eBGP). For a router to be considered an iBGP peer, it must be within the same AS as the source router; eBGP peers reside in different domains. Another difference between iBGP and eBGP peers is how a router handles the advertised metrics received from the peer. For example, *local preference* (local_pref) is a domain specific degree of preference for a prefix. Although a router's local policy information base (PIB) may change the value, typically a router will maintain the local_pref value received from an iBGP peer. A route received from an eBGP peer will have its local_pref set to the default value unless otherwise specified in the PIB [4].

When configuring iBGP, iBGP peers must maintain a full iBGP mesh, meaning each BGP speaker maintains a peer relationship with every BGP speaker within the AS. Routers must be statically configured with the address for each peer, so configuring a full mesh in a large AS can cause significant administration overhead. To simplify configurations, a special type of iBGP peer called a route reflector can be used. Route reflectors allow a BGP speaker to establish a single peer connection and still receive updates from any other router who peers with the route reflector [43].

To store the advertisements, each BGP router maintains a BGP specific routing table known as the local routing information base (Loc-RIB). The Loc-RIB contains the best accepted routes as determined by BGP. For each peer, a router maintains two additional tables of routes associated with the peer: Adj-RIB-In and Adj-RIB-Out. The Adj-RIB-In table contains the full list of NLRI advertisements received from a neighbor, while the Adj-RIB-Out contains the routes which are eligible for advertisement to the specific peer [4].

BGP populates the Loc-RIB table via a three step process, which considers

the router's PIB, all routes contained in Adj-RIB-In tables, and locally generated routes. The first phase is triggered whenever a router receives an *update message* from a peer, which is used for adding, updating, or withdrawing routes. The phase calculates the degree of preference for all received routes prior to placement in the respective Adj-RIB-In table by setting their local_pref metric. A higher local_pref indicates a higher degree of preference for the route. The local preference for a route is set to the default value when a route is received from an eBGP peer, and is unchanged when received from an iBGP peer. However, this rule may be overridden by the local router's PIB.

The second phase is triggered by the completion of phase 1 and is responsible for calculating the best path to every known destination based on the information available in all Adj-RIB-In tables. The phase goes through the multi-step selection process, eliminating routes until only one remains as the best path. The path selection follows the checklist sequentially:

1. Highest local_pref is selected.
2. Select the path with the least number of ASes to traverse.
3. Select the lowest origin. IGP $<$ eBGP $<$ Incomplete.
4. Highest Multi-exit Discriminator (MED). Only MEDs from the same AS can be compared and the lower MED is preferred.
5. Prefer eBGP routes over iBGP.
6. Prefer routes with a lower IGP cost to reach the next-hop address.
7. Prefer routes with the highest BGP ID.
8. Prefer route from peer with lowest peer address.

The final phase is the the route dissemination process. This phase places routes from the Loc-RIB table into each Adj-RIB-Out table. The local PIB affects which routes are placed in the Adj-RIB-Out tables; routes can be excluded based on the PIB and some metrics, such as MED and AS Path, may be manipulated prior to placement into an Adj-RIB-Out table [4].

3.3 Interdomain Traffic Engineering

As BGP is the standard for interdomain routing, interdomain traffic engineering (TE) must be performed within the confines of BGP. At its most basic, the BGP path selection process is equivalent to minimizing the number of AS path hops required to reach a destination. However, the full selection process provides methods to override AS path count, as well as multiple tie breaking procedures in case multiple routes contain identical length AS paths.

A router's local policy has a significant amount of influence over which routes are selected and advertised, which makes predicting how the network will react to changes difficult [44]. Each AS on the Internet is its own entity, with most run by different and competing companies which have their own policies and goals. For TE, there are essentially two types of ASes: stub and transit. Stub ASes have one or more connections and only provide connectivity services to customers. Transit ASes may provide services to locally connected entities, but they provide other ASes a path to connect with the rest of the Internet. Transit ASes and stubs servicing content providers are typically concerned with traffic exiting their AS, while stubs acting as access providers are concerned with incoming traffic flow [45]. Despite the different TE requirements, the common denominator is both need to balance ingress and egress traffic to external ASes and to reduce the cost of passing traffic on those connections.

3.3.1 Outbound

BGP allows for the direct manipulation of internally preferred paths, so two methods for outbound TE exist. The first is the manipulation of the local_pref attribute, allowing the network administrator to rank order the AS's egress point(s). The second method for outbound TE with BGP is to rely on the IGP to influence how an egress point is selected. If multiple routes have the same AS path length, one of the trailing steps for selecting the best path is the route with the lowest IGP cost, so manipulating the IGP metrics can influence the AS egress point [45].

3.3.2 Inbound

Inbound traffic manipulation with BGP is much more complex because it involves influencing the decision making process of external ASes. Despite having multiple metrics to manipulate, a router's control over how a peer manipulates metrics and selects a best path is limited. Despite the limitation, there are five methods for an AS to influence external ingress points [45].

The first two methods are closely related. *Selective advertisement* is used to announce certain routes to neighbor ASes while advertising only a portion or an entirely different set of routes to other neighbors. Selective advertisement guarantees external entities will have the ability to reach certain networks only through the AS's chosen ingress point. Advertising *specific prefixes* is a complementary method to selective advertisement; routers will always forward traffic towards the most specific route known for a given destination. So, by advertising different length prefixes to different neighbors, an AS can influence ingress traffic for certain destinations.

Another popular technique is *AS path prepending*. When a route is advertised from AS to AS, each AS manipulates the AS path for the route by adding its own AS number (ASN). Prepending is when a BGP router attaches multiple copies of its ASN prior to placing the route in a peer's Adj-RIB-Out table. This artificially extends the length of the AS path, potentially breaking ties based on AS path which is the second criteria for path selection.

The MED attribute provides an AS with multiple connections to a neighbor AS influence over which of the connections is used as the ingress point for a destination. Since the two ASes are neighbors, the first three selection criteria will be identical so influencing a routes MED will modify the neighbor ASes selection process.

Finally, the use of *communities* allows an AS to attach optional attributes to a route when advertising it to a neighbor. The community attributes can define how to handle the route when advertising it to upstream ASes. For example, it is possible to attach a community that informs the neighbor it should not advertise the route to any of its peers or to prepend the route when advertising it to certain peers. Communities provide an ability to influence how a neighbor AS treats and advertises routes [45].

3.3.3 Shortfalls

Despite the ability of BGP to manipulate ingress and egress points, it does not provide enough support for fine tuning interdomain TE. Prepending provides a method for manipulating the primary path selection criteria used by other domains. However, prepending does not provide the granularity required to balance traffic between two ingress points. At best, it provides a method to inform downstream ASes which path to avoid [14].

Selective advertisement may not guarantee redundancy for networks if their chosen ingress point suffers a failure. Advertising specific prefixes allows for redundancy in case of failure and easily influences the ingress point of a network. However, if an upstream AS is performing aggregation, then the effort may be nullified. It also does not provide the ability to balance traffic, only to select a single entry point and increases the size of routing tables by advertising multiple copies of a route.

MED manipulation requires multiple connections between neighbor ASes and is only a single hop attribute. That means the neighbor will not necessarily pass along the preference to neighbor ASes. Community values are useful and provide some control over upstream advertisements, but due to the limited knowledge pertaining to an AS's policy, it is difficult to predict the impact of any given community value. Communities can also be complex to configure due to the amount of communities that may be required to have the desired effect [14].

3.4 Interdomain LSPs

MPLS-TE provides the ability to explicitly route packets within a network and is often deployed by network administrators for TE purposes within ISP networks [46]. There is now significant research and standardization efforts towards forming interdomain LSPs. Intradomain TE was able to support flow QoS guarantees, perform network resource optimization, and to provide fast recovery for LSPs [47]. Interdomain MPLS-TE must achieve these goals as well, but without full knowledge of the topology and with the additional requirement of protecting each ISP's internal path information [46].

3.4.1 LSP Establishment

Intradomain paths were relatively simple to calculate and to establish; all routers within the domain openly share information to enable each other to determine the best path for a packet to travel. Every router ran the same IGP as well as the same label distribution protocol. Once a reservation request crosses domain boundaries, the assumptions about uniformity are no longer valid. Another limitation is the lack of visibility for TE properties across domain boundaries.

Three general types of interdomain LSPs were proposed to allow for LSP establishment across domain boundaries. LSP nesting builds on the foundations of allowing packets to carry multiple MPLS headers, creating embedded LSP tunnels. The nesting of a tunnel allows the two end devices to exchange label information, but at each domain, the end to end tunnel is carried across the domain inside another LSP. A contiguous LSP is a tunnel established from ingress to egress via a single signaling session, meaning the initial RSVP-TE path message traverses the entire path of the LSP. LSP stitching is a string of multiple LSPs, each constructed within a domain, and used consecutively to form an end to end path in support of a flow [48]. During the establishment of an end to end LSP, any combination of the three methods may be used. Stitching is very similar to nesting in that each LSP within a domain has its own LSP ID, but in nesting the end to end LSP has its own LSP ID as well [21].

To accommodate the establishment of the different LSP types, RSVP-TE was once again enhanced to support the general LSP types as well as to allow for maintaining AS privacy during establishment and troubleshooting. Which type of LSP a request is able to establish is dependent upon the signaling methods supported by the domain border routers along the end to end path. The burden of processing, enforcement, and establishing the LSP primarily fall on the border routers. When an ASBR receives an RSVP-TE request, it must follow four rules [21]:

1. Apply domain policies regarding interdomain LSP establishment.

2. Determine the signaling method to use.

3. Follow exclusive route object (ERO) procedures.

4. Perform any path computation procedures required to select the path to cross the domain and to potentially select the domain egress point.

If an ERO is included in the path message, as specified in rule 3 above, the LSR must:

1. Ensure any policies relating to the ERO are followed.
2. Determine whether to use an existing LSP or to signal a new one if either nesting or stitching is used.
3. Check to ensure that a contiguous ERO is not attempting to use an advertised TE path.
4. Expand a loose hop to a more specific path, potentially using additional loose hops.
5. If the ERO does not contain hops beyond the local domain for a non-local destination, the destination is considered as a loose hop and rule 4 is applied.
6. Generate an error message in case of failure.

A record route object (RRO) is an optional attribute for recording the path an LSP travels and may be included in the resv message. However, recording the full RRO of a path may reveal too much information about the internal workings of a domain. Therefore, the domain's egress LSR is allowed to manipulate the RRO and may remove some or all information pertaining to the hops traversed within the domain. However, the domain border router must leave its own information inside the RRO [21].

3.4.2 Failure Recovery

Network stability may be an issue within a domain, and is an even bigger concern when crossing multiple domains. A failure within any of the traversed domains may impact either the establishment of an LSP or the performance of an already established LSP. To ensure LSP performance, domains must have a method for dealing with each scenario.

Dissemination of information is not instantaneous, and is non-existent for interdomain TE, so an LSR may have out of date information when calculating the path for an LSP. Multiple reasons exist for a reservation request to be rejected, including: policy failure, path computation error, explicit route rejected, and signaling type not supported. At any time, if a node along the path cannot support a reservation with the requested resources, a path error message (patherr) must be reported back to the *head end* LSR, i.e., the LSR originating the request.

The head end LSR may want to allow LSRs along the path to calculate detour routes, a process known as *crankback*. An LSR closer to the problem likely has a more accurate status pertaining to its portion of the network, so it is more capable of making an accurate path decision. The head end LSR may, therefore, indicate its desire for which LSRs along the path may perform alternate path calculations on its behalf in the case of a setup failure. In the path message, the head end LSR may allow either any LSR or only border routers (AS or area) to attempt crankback [22].

When allowed, an LSR needs to know detailed information pertaining to the problem that caused the generation of a patherr. Crankback requires the interface and node address, at a minimum, to calculate alternate paths within a domain. Even with this information, the LSR calculating the alternate path may not have enough information to generate an ERO to route around the problem. In that case, the LSR may use the exclude route object (XRO) to indicate avoidance of the problem link [22]. If an LSR is unable to determine an alternate path, it must forward the patherr message to the head end LSR [22]. As the patherr message travels towards the head end LSR, each domain's LER may remove or modify domain specific information contained in the patherr message. Changing specific errors and node information to a more general error and indicating the domain, rather than a specific interface, as the failure point is acceptable [3].

In case of a link or node failure after LSP establishment, existing methods such as MPLS-TE fast reroute, which specifies different methods of creating backup LSPs, can serve as protection. As a result, if a failure within a domain occurs, recovery is the domains responsibility. MPLS-TE fast reroute techniques can still be applied to failures of a link between two domains or for the failure of a border LSR [3]. Crankback can also be used to recover an LSP on demand. Both the

upstream and downstream LSRs of the failure may issue a *pathtear* message to release the resources occupied by the failed LSP. For the upstream case, it may be the nearest domain border router that issues the pathtear message. Dependent upon which LSRs are able to perform crankback, each LSR may attempt to determine an alternate path according to the aforementioned rules. If no path is available, then the error message is passed upstream, with each eligible router free to attempt crankback.

3.4.3 Reacting to Network Congestion

The capability of MPLS to re-optimize an established LSP based on network situations is a fundamental requirement of the protocol. MPLS takes a *make before break* mentality towards re-routing LSPs, allowing an LSR along a path to trigger re-routing based on some detection or calculation method [49, 50, 51, 52]. However, interdomain LSPs have special considerations depending on which type of LSP is established.

If a continuous LSP is established, the head end LSR must be the one to indicate a reroute is required. Based on poor QoS, the head end LSR may signal all routers included in the original ERO, if used, to determine whether any alternate paths exist. Any LSR may also signal the head end LSR of a potential alternative path, which can then trigger the re-optimization procedure [3].

Nested or stitched LSPs are multiple individual LSPs and can be re-optimized without sending notification to the head end LSR. If an LSR determines it must re-optimize a path, it can do so using the make before break principle within its domain and the overall LSP is not affected. [3].

3.4.4 Path Computation Element

RSVP-TE allows for calculating loose hops in an ERO, allowing the LSR to pass the responsibility of calculating a full path to a downstream LSR. However, due to limited visibility and loose hops in the ERO, there is no guarantee a path exists or the next hop is able to calculate the next section of the path. Unfortunately, the only method for determining whether a path exists may be through multiple attempts. Each attempt consumes resources at the head end LSR, as well as at

each LSR along the path which must process the requests and allocate resources which will be released a short time later. The path computation element (PCE) architecture provides the structure for one device to request a path from another device without requiring the overhead in each transit device.

A PCE is any device capable of calculating a constraint-based path. The architecture provides the head end LSR, or any LSR along the path, with the ability to contact another device and request the contacted device calculate a route for an LSP [53]. The architecture employs a client (PCC) server (PCE) model; the capability of a device to serve as a PCE and its scope for calculating paths can be distributed via OSPF [54] or BGP [55]. When a PCC sends a requests to a PCE, the PCC must indicate all requirements for the path which may include the source and destination of the path, QoS parameters of the path, possible links or nodes to use or avoid, whether disjoint paths are required and how many, required reliability, and any policy related information. The requests may cross domain boundaries, but ISP confidentiality must be protected. Therefore, when a PCC requests a path computation from a PCE, the result may be in the form of a loose ERO, most likely specifying ASBRs to use in the RSVP-TE path message. While servicing a request, a PCE may also request assistance from another PCE in an effort to ensure the next section of the path can support the request [53].

The PCE structure is extremely useful for calculating paths between areas or ASes, especially when the domain contains more than one egress node. If the head end LSR initiates the LSP setup after its own route calculation, the egress point it chooses may reject the request. The node will have no choice but to recalculate the path and direct another request towards a different egress node. If the node were to request an egress LSR to perform the calculation, it could be provided with a path specifying which egress node could better support the requested path. The same could apply to ASBR selection, however, the information required for an ASBR to make an informed decision on which egress point best supports the request does not exist.

Chapter 4

Interdomain Traffic Engineering Advertisements

4.1 Background

Traffic engineering (TE) focuses on optimizing traffic flows to satisfy quality of service (QoS) requirements, avoid congestion, and to better utilize available network resources within a given domain. Whether within an area or an autonomous system (AS), internal gateway protocols (IGPs), TE extensions, and multi-protocol label switching (MPLS) are able to achieve all of the stated goals. Interdomain TE seeks to achieve the same goals, but the current interdomain routing protocol, Border Gateway Protocol (BGP), provides at best a method for designating a domain's ingress and egress points, which is not true traffic engineering. Extending intradomain principles to interdomain solutions has proven challenging, due to the unique nature of interdomain connectivity.

Unlike intradomain TE, interdomain TE carries the additional burden of having to navigate the business agreements negotiated between heterogeneous organizations. Internet service providers (ISPs) may barely cooperate with one another and are in direct competition for the same customer base. However, they still enter into contracts with each other and form network interconnections based on Service Level Agreements (SLAs). SLAs and the prices negotiated pertaining to how each AS will handle the other's traffic greatly influences how packets travel

across the Internet. Based on an SLA, select ISP interconnects are not advertised to certain ASes, while other interconnects are preferred regardless of the impact on traffic. These policies complicate attempts to perform TE as well as potentially using MPLS for TE across the Internet.

ISPs also guard the internal workings of their networks as proprietary information. An AS's IGP, number of nodes, topology, MPLS capabilities, label distribution, and label switched path (LSP) establishment protocols are all subject to the proprietary veil of secrecy. Another major issue with advertising TE information is scalability; BGP routing tables are already very large and policies complex. Adding additional details could overwhelm the existing infrastructure, making any solution unscalable at the Internet level.

To provide a palatable solution for ISPs, the released information must be abstract, so as to prevent revealing internal details. However, a label switched router (LSR) must have enough details about each alternate route to make an informed decision when requesting an interdomain LSP. We propose to extend BGP in a way similar to OSPF-TE, making BGP a transport mechanism for interdomain TE information. By including TE attributes in BGP network layer reachability information (NLRI) advertisements, each transit AS will maintain control of advertisements to each of its neighbors, but the neighbors will have an insight into the path's resources. The BGP-TE attribute is an optional, transitive attribute that provides an end to end view of a path's resources. Each AS along the path is responsible for calculating an overall view of the path based on their internal network status and information provided by the previous domains. The amount of information advertised by OSPF-TE may be too specific and contains IGP specific data, so the BGP-TE attribute is scaled down to primarily include only the bandwidth available to potential LSPs when traversing the domain towards a destination.

4.2 Related Work

The shortfalls of BGP for traffic engineering purposes are well documented [14, 44, 45]. MPLS-TE and RSVP-TE create interdomain LSPs and provide the ability to perform traffic engineering on a per domain basis, as discussed in Chapter 2.4.

However, each domain is essentially guessing whether or not a neighboring domain can support establishing an LSP when a request is sent across domain boundaries. Even with PCE communication, a request across AS boundaries has no information on which to base its decision and multiple requests may be required to find a suitable path. The exchange of information across domain boundaries to assist in the creation of interdomain LSPs is lacking. Proposed solutions are to advertise either QoS or TE information via BGP, or to allow multi-path advertisements.

By adding optional QoS information to BGP advertisements, domains could calculate constrained interdomain paths. One such proposal suggests adding optional QoS information to BGP advertisements. The advertisements are varying lengths and may include multiple attributes: per hop behavior identification, maximum bandwidth, available bandwidth, maximum and minimum transit delay, and required signaling [56]. The amount of information advertised may be too detailed for ISPs to accept, and the non standardization of which attributes to advertise could provide either too much or too little information for making a decision. Cristallo and Jacquenet propose a method for attaching a single QoS metric, which could be one of several different metrics, to a prefix advertisement. The attribute is optional and transitive which makes it incrementally deployable [57, 58]. However, it requires an advertisement for each QoS parameter it wishes to advertise which limits the scalability and usefulness of the information.

Abarbanel and Venkatachalam proposed an extension to BGP that includes several TE attributes in advertisements such as maximum bandwidth available, maximum number of IGP hops, maximum transit delay, color, etc [59]. In their approach, the authors propose to embed the information from every router that is advertising the information along the path. The information could be used to build a verifiable and usable database of information for any router to examine all paths and associated TE weights for any given prefix or router. This approach is very comprehensive, but it reveals an enormous amount if internal information that most ASes consider private. In addition, all of the additional information would make the Adj-RIB-In table larger than most routers could handle.

Leveraging the PCE architecture to gather information for distribution via BGP is proposed in [60]. Each PCE capable of calculating interdomain LSPs will gather information relating to LSP tunnels within the domain connecting ASBRs,

since they are capable of transporting interdomain traffic as hierarchical or stitched LSPs. The PCE would distribute this information to all of its eBGP peers, informing them of the full range of interdomain LSPs available within the domain. Each domain will in turn advertise the information to their neighbors. Divulging the full range of potential interdomain capable LSPs may reveal internal information relating to ISP agreements or reveal connections which a domain prefers to conceal from neighbors.

Brahim et. al. proposed a TE attribute for BGP that is a subset of the attributes standardized in [1, 31]. The maximum bandwidth at each priority, minimum LSP bandwidth, and the interface MTU are to be an optional non-transitive attribute [16]. The attribute standardizes the TE information used in IGPs and BGP which simplifies sharing information between the two.

To have a more accurate picture of the current network status, especially with QoS or TE information, multiple routes would provide an additional level of detail. The size of the Internet and routing table precludes the potential advertisement of multiple routes. [61] suggests a method for including QoS information as well as for advertising multiple paths, but using a path reduction algorithm to reduce the number of additional routes advertised. Walton et. al. proposed allowing BGP neighbors to inform each other whether or not they are capable of receiving and processing multiple paths to the same destination prefix. The additional paths can be advertised to BGP peers without any of the new advertisements implicitly replacing the old path by specifying a new path identifier [62]. Bhatia and Halpern expands the idea to specify the number of next-hops and to advertise multiple next-hops via BGP, but only as a non-transitive attribute. Each NLRI has multiple next hops associated with it so that paths can be identified by their NLRI and next hop, rather than just NLRI [63]. Multi-path Interdomain Routing (MIRO) proposes a pull method for multi-path BGP. The MIRO method allows a source AS to request alternate paths for reaching a destination from one or multiple neighbors. With this new alternate path information, a tunnel is created which allows the desired traffic to take the alternate route. With over 60% of ASes being multi-homed and several thousand advertising more specific prefixes to affect path selection [64], allowing the ability to request additional paths could reduce routing table sizes.

4.3 Protocol Goals and Assumptions

The purpose of our proposal is to ensure traditional BGP policies can remain in place, but additional information is distributed. With the additional detail, an AS can more knowledgeably create interdomain LSPs based on the provided network status snapshot. We provide the ability to weight or override the selection a constraint-based routing decision with policy, similar to current BGP methods; best effort traffic can be routed by traditional means.

In proposing the BGP-TE attribute, as well as the methods for calculating and distributing the information, we assume each domain is able to establish interdomain LSPs via RSVP-TE and its associated extensions. Another assumption is each domain is capable of distributing TE information internally via their IGP (OSPF-TE or IS-IS-TE). Based on the ability to distribute TE information, we also assume each AS border router (ASBR) is able to leverage the information to determine a best path towards an iBGP peer based on some internal constrained shortest path first algorithm (CSPF). We also assume the neighboring ASes will negotiate the amount of bandwidth, both the maximum total and at each priority level, they will allow the other to request for traversing their network via an out of band method.

4.4 Path Selection and Advertisement

In order for the BGP-TE attributes to provide relevant information, we recommend a procedure to determine the values for advertising BGP-TE attributes to peers. There are several factors to consider when determining what value to advertise to a peer; the calculation of a best path or the available bandwidth at each priority level is dependent upon which neighbor sent the advertisement and to which peer the router is sending an advertisement. We will discuss each of the combinations as well as how to handle route reflectors (RR) and neighbors who are not capable of establishing LSPs.

4.4.1 Advertisement Requirements

As we mentioned, each domain will maintain control over what it advertises, identical to the current format of BGP; the BGP path selection process remains unchanged for best effort traffic. A domain can specify in each router's policy information base (PIB) a level of preference for individual paths, and as a result advertise paths according to the networks administrative policy. However, once a BGP router selects the best path for a given destination, it must determine the internal TE attributes to reach the selected egress point. To calculate the attribute values, each router approaches the problem from the perspective of being the domain's ingress and receiving an RSVP-TE request from an eBGP peer to establish an LSP towards the destination. The router's goal then becomes selecting the best path to reach the egress router based on the domain's CSPF algorithm.

4.4.1.1 BGP-TE Attribute Format

Since ISPs are secretive regarding the details of internal paths, the amount of information provided by the OSPF-TE extension may be too revealing for an abstract advertisement, and some of the details are not useful for interdomain purposes. Therefore, the only information retained from the OSPF-TE attributes is the maximum available bandwidth at each priority level. To satisfy the requirements for Generalized MPLS (GMPLS), an MPLS control plane extension, the BGP-TE attribute also includes the potential LSPs switching capability, bandwidth encoding, and a variable length switching capability information block which depends on the value of the switching capability [16]. The resulting attribute format is seen in Figure 4.1.

We will now discuss the process for calculating the maximum LSP bandwidth at each priority level, according to each LSR that handles the advertisement beginning with the originating AS.

4.4.1.2 Originating AS

At the originating AS, the destination network is imported into eBGP either by redistribution from an IGP or by originating the network on a BGP speaking router. The BGP NLRI origin attribute describes whether the route was originated

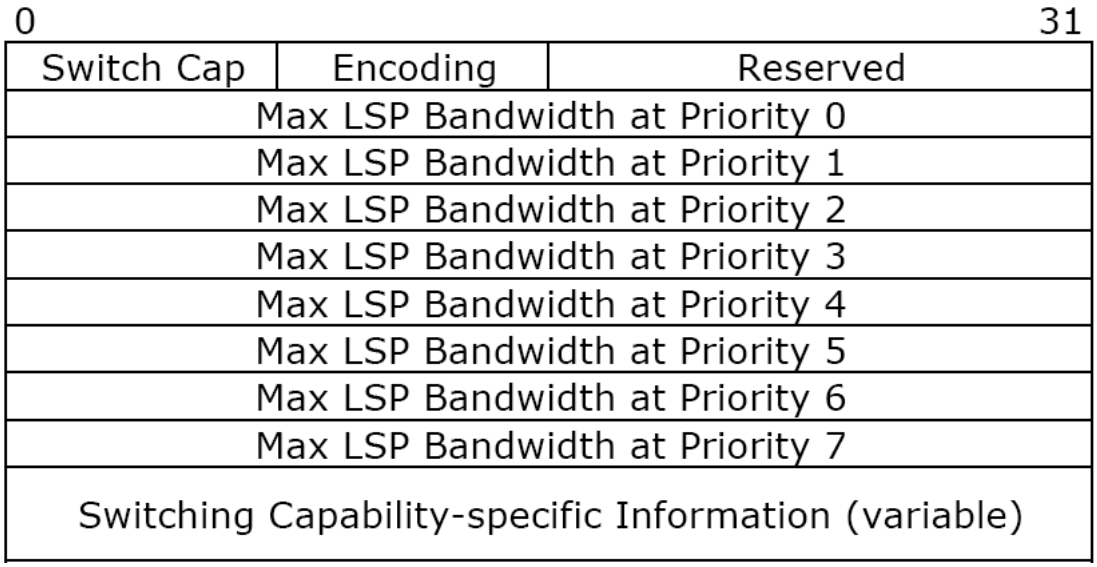

Figure 4.1. The BGP-TE attributes

by an AS internal node, by an external AS, or if the information is incomplete because it originates in an IGP. Based on the origin information and AS path hops, a BGP speaking router is able to determine how it should handle an internally originated route.

If a BGP router locally originates a network, it must include the outbound interface's maximum bandwidth as the maximum available bandwidth at each priority level for the given network when advertising the network to iBGP peers. The reason for potentially limiting the maximum reservable bandwidth to the destination is to prevent saturation of a local connection or interface. The router must inform peers how much traffic the outbound interface can handle, even though the local interface may not support LSP establishment which is the case if the NLRI is for a directly connected local area network (LAN).

Reception of an internally generated route from an iBGP peer requires the router to determine a best path to reach the next hop indicated in the advertisement, prior to placing the route in the Adj-RIB-In table for the peer. The router will use the best path according to the IGP's distributed TE attributes to calculate the path to the destination. When the receiver of the internally generated BGP route is preparing to advertise the route to another internal peer, it must leave the next hop attribute unchanged, unless specifically configured to indicate itself as the next hop [4]. The new receiving router follows the same guidelines.

If a router is performing redistribution, it must determine the path towards the destination using the IGP values and CSPF prior to placing the route in its Loc-RIB. The origin is marked as incomplete for a redistributed route, so any iBGP peer receiving the route will know it must not look at the next hop value. Rather, the router will use its IGP and CSPF to determine the values to assign the route as the originating router did.

If a router is also an ASBR, it must indicate the maximum amount of reservable bandwidth available to reach the destination prefix when advertising the route to eBGP peers. To accomplish this, each egress ASBR determines the availability of incoming reservation bandwidth at each priority level according to current reservations and the maximum allowable amount. The available unreserved and maximum allowable bandwidth are compared to the best path towards the prefix calculated by the internal CSPF algorithm. The minimum of the three values is stored with the advertisement in the appropriate Adj-RIB-Out table for advertisement.

4.4.1.3 Transit AS eBGP Neighbor

When a transit AS receives an NLRI advertisement with TE information from an eBGP neighbor, it must compare policy information with the current status of reservations between itself and the eBGP neighbor from whom it received the advertisement. The router examines its locally configured policy for the maximum reservable bandwidth destined for the eBGP neighbor and lowers it by the amount of currently reserved bandwidth at each priority level. The routers local status is compared to the amount received from its peer, and the minimum of the comparison is stored in the Adj-RIB-In table for the respective eBGP neighbor.

If the route is selected as the best path according to BGP, the router must calculate path's status from its perspective prior to propagating the advertisement to eBGP neighbors. The router must examine its locally configured maximums and the current inbound status to determine the amount of bandwidth available and compare it to the previously calculated values. The minimum of the comparison is stored as the current status of the path within each eBGP neighbor's Adj-RIB-Out table.

When advertising to an iBGP peer, no additional calculations or comparisons are required. These are done by the domain's CSPF algorithm and are performed

by the iBGP neighbor receiving the advertisement.

4.4.1.4 Transit AS iBGP Neighbor

When an router receives an external BGP route containing TE information from a peer, it follows a process very similar to that performed by the origin domain's iBGP neighbors. The router calculates the best path towards the next hop indicated in the advertisement. The values determined by the CSPF are compared to those in the advertisement, and the minimum values are placed in the iBGP neighbor's Adj-RIB-In table.

If the neighbor has any eBGP peers, it must follow the guidelines stated above for advertising to an eBGP peer.

4.4.1.5 Route Reflectors

A route reflector will perform the typical iBGP neighbor TE calculations prior to placing a received advertisement, either from a client or iBGP peer, into the appropriate Adj-RIB-In table. The route reflector will then select its best path normally and reflect routes accordingly. However, the receiving router must know the original BGP-TE attribute values for its comparisons and advertisements, so a route reflector must retain the BGP-TE attributes it receives from iBGP peers and clients. When a route is reflected, the original BGP-TE attributes are advertised. Since route reflectors are recommended to not modify the next hop of a reflected route [43], any client or peer of a route reflector will perform the calculations as if they had received the original router.

A route reflector will act as defined above when exchanging routes with eBGP peers; the route reflector will appropriately modify the TE information of routes it receives from iBGP neighbors prior to advertising to an eBGP neighbor and it will modify eBGP routes prior to advertising to iBGP neighbors.

4.4.2 LSP Selection Process

Rather than simply follow the best path as selected by traditional BGP means, which would be the same as following best effort traffic for a given destination, we propose a BGP-TE path selection criteria when choosing the egress for an LSP. Our

version is a modified and shortened version of the traditional BGP path selection process, which will query the Adj-RIB-In tables to examine the TE attributes of all eligible routes. We propose to allow a domain the ability to override the potential path with the highest amount of available resources by specifying a local preference for traffic engineering (te_local_pref), identical to traditional BGP. The te_local_pref attribute allows a domain to rank order external connections, so as long as the highest preferred path has the resources to support the request, it will be chosen. The second, and final, selection criteria is which path is best able to support the path as determined by a constraint-based algorithm. During each of these steps, however, the router's local PIB may eliminate a path from contention based on the agreements between the ASes about which types of traffic may traverse a link.

4.4.3 Backwards Compatibility

The proposed TE attributes are an optional, transitive attribute. Since the attribute is transitive, even domains which are unable to utilize the information contained in the attribute are required to forward it, unchanged. However, advertising the attribute unchanged may provide inaccurate information regarding a path's quality. The BGP-TE attribute and a domain's local policy allow for dealing with domains that either do not support the BGP-TE attribute or do not support interdomain LSP establishment. If a domain does not support interdomain LSP establishment, then its incoming advertisements should be set to the minimum, which is 0.

If a domain is unable to support the attribute but does support interdomain LSPs, the process for calculating the quality of the path should still not be carried out normally. A single domain's inability to calculate the attribute cannot be overcome. Each ASBR examines the inbound status from a peer prior to advertising the TE attributes to that peer. The peer in turn inspects the outbound status. The interface status *should* be identical on both sides, so the amount advertised by the non-compliant domain is equivalent to the path according to the previous domain. However, if a domain does not include its internal status, the actual quality of the entire path is inaccurate, even though the receiving ASBR has information

pertaining to allowable resources across the non-compliant domain. The amount to set for this may be dictated by policy, but should be set to 0 as well.

4.5 Summary

In this chapter we proposed the inclusion of a minimal amount of TE information required to facilitate the calculation of constrained paths. The attributes are not overly revealing, yet they provide sufficient information to allow neighboring ASes to view a path's quality towards any destination. The maximum reservable bandwidth at each priority level corresponds to those of IGP TE modifications and will be easily adaptable and importable to a domain's IGP. We also propose methods for calculating the values to advertise, which can differ according to how the route was received and who the route will be advertised to for every router. Our modified path selection process for BGP-TE provides a simple yet robust enough implementation to allow for domain policy preferences when selecting one path over another and still satisfying the LSP QoS requirements.

Chapter 5

Evaluation

In Chapter 4, we discussed two major obstacles related to implementing an interdomain traffic engineering (TE) solution. The first is the requirement to maintain a high level of privacy by not revealing any specific information regarding an ISP's internal operations. Our proposal for calculating a path's quality and providing the priority specific bandwidth information reveals nothing that would not otherwise be revealed during the course of establishing an actual label switched path (LSP) using RSVP-TE or the path computation element (PCE) architecture. The proposal simply provides this information to neighboring domains prior to the initiation of a request, thereby reducing failed LSP attempts and allowing a domain to better manage available interdomain resources.

The second obstacle facing any potential interdomain TE solution is scalability. BGP routing table sizes are already extremely large, with large transit ASes having to maintain over 250,000 routes [23]. To be a viable solution, any TE proposal must not cause a significant increase to average routing table sizes. To demonstrate the scalability of our proposal, we simulate a realistic Internet level topology using existing BGP simulators, and perform routing table size comparison and analyses.

5.1 Experiment Setup

5.1.1 Tools

To test the memory requirements of our proposal, we needed a method to simulate a very large scale BGP network. The BGP solver C-BGP provides the ability to simulate the propagation of BGP advertisements between autonomous systems (ASes), resulting in a realistic simulation of routing table content and size. C-BGP takes into account a routers local policy information base (PIB), supports the full BGP decision making process, and, most importantly, allows for adding experimental BGP attributes [65]. C-BGP is designed to fully simulate an ISP's entire network, including the IGP. However, we want to simulate the effect our attribute will have on routing table size at the Internet level, so we use only one router per AS. This allowed us to simulate a large number of ASes while testing the impact of the TE attribute on routing table memory requirements.

In order to determine how much memory our simulations consumed, we had to monitor how much RAM the simulation required. Unfortunately, C-BGP does not provide the ability to monitor actual memory consumption on the underlying machine. To perform this task, we used the memtime utility [66]. Memtime uses sampling, taken ever 100 ms, and keeps the maximum amount of memory from the samples [5].

To generate the network topology for testing the scalability of our proposal, we used routing table information provided by Routeviews. Routeviews provide Internet routing table snapshots gleaned from several dozen peering points across the Internet. We used a snapshot taken on September 25, 2007 as the basis for our network topology, which included 26,539 ASes and approximately 239,000 prefixes [23].

5.1.2 Topology Generation

Simulating our proposal in a realistic environment will generate realistic routing table requirements. To achieve a realistic environment, we created topologies of various sizes, each generated from our Routeviews snapshot. In generating our different topologies, we selected a group of autonomous systems to act as our

foundation. The Internet is generally considered a flat network of interconnected ASes, with the interconnects based on peering agreements. However, due to the peering agreements, a hierarchical classification system was developed that separates ISPs into three tiers. A tier 1 ISP has access to the global Internet but does not purchase transit capabilities from any other ISP [67], thereby establishing a *peering relationship*. We selected nine tier 1 ISPs to act as the basis for our topologies.

Using the snapshot from Routeviews, we were able to generate a connected graph of autonomous systems which represents a partial topology of the Internet. From the graph and our selected starting ASes, we began to select subsets of the graph to form multiple topologies for use in our simulations. The initial graph consisted of only our selected tier 1 ISPs, which were fully interconnected. From the initial network, we added the collective set of neighbors for one or more of the tier 1 ISPs to increase the size of our network, expanding the network until all one hop neighbors were included. After including all one hop neighbors, we generated a graph including every AS. The result was graphs of 9, 100, 990, 7752, and 26,539 ASes.

5.1.3 Assumptions

BGP is highly influenced by a router's policy information base (PIB). The router's PIB may alter metrics or communities, aggregate networks, or restrict which routes are advertised to, or accepted from, peers. Each router's PIB is based on the domain's policies, which reflects inter-ISP agreements, the domain's attempts at managing ingress and egress traffic, its own goals, and network conditions. As mentioned earlier, ISPs generally do not reveal their internal policies and peering arrangements. A lot of work has been performed in order to infer policy characteristics based on available information such as Routeviews. To create a realistic scenario for how each AS would act, we would need to use the inferred policies in our simulations. However, policies tend to restrict routes from entering Adj-RIB-Out tables, which results in advertising fewer routes to a peer [5]. Rather than implement realistic policy in our experiment, we chose to implement a uniform and open policy across all domains, allowing every AS to advertise all prefixes to every

neighbor. This decision allows for every AS to have a path to every destination.

Populating the routing tables by originating each prefix in its real-world AS is not necessary. Since we are not implementing policy variations or tracking AS paths, just routing table growth, allowing each domain to generate an even number of prefixes would suffice.

Even with the optimizations performed by C-BGP, it is not possible to fully simulate a network of 25,000+ ASes and 239,000+ prefixes due to memory constraints. To simulate the results, we ran combinations of varying numbers of ASes with an increasing number of prefixes, beginning with a single prefix, until our simulation server was unable to complete the scenario. This provided enough results to calculate the amount of memory required for larger simulations.

5.1.4 C-BGP Modifications

We implement the BGP-TE attribute in two separate scenarios. The first is an iBGP implementation, which reflects BGP becoming a transport for TE information similar to that of OSPF-TE. In this scenario, BGP-TE will only relay interdomain TE attributes to iBGP peers. The second implementation is a full implementation, allowing advertisement of BGP-TE attributes to cross AS boundaries, conveying end to end path quality to neighbors. Implementing these versions of BGP-TE required altering C-BGP to create, modify, and destroy the additional attributes. In order to modify C-BGP to handle the new attribute for the various implementations, we had to understand how it and traditional BGP handles message passing and populates the Loc-RIB, Adj-RIB-In, and Adj-RIB-Out tables.

C-BGP simulates a fully functional BGP network including full message passing and decision making phases, as described in Chapter 3.2. To more easily depict where our modifications occur, Figure 5.1 represents the full process a router must undergo whenever it locally generates a route or receives an update from a peer. The following sections will discuss specific modifications required to implement our models.

To test our implementation, we were required to not only make modifications to the processes, but also to content included in BGP advertisements and routing table entries. C-BGP implements a BGP route in two structures: SRoute and

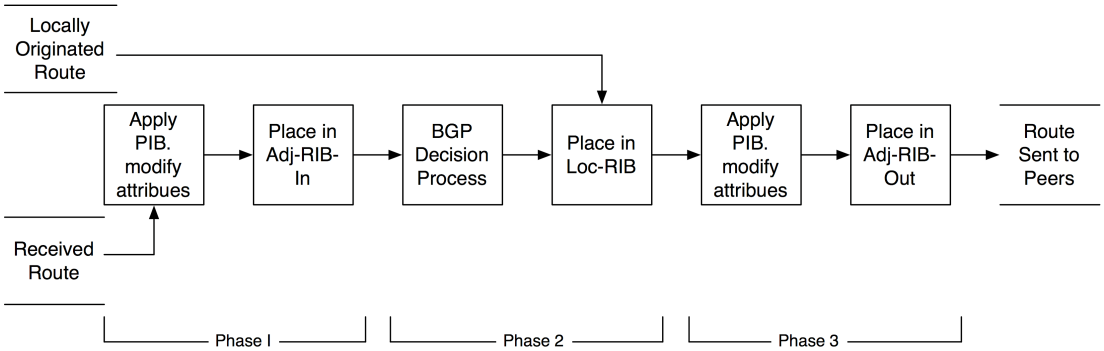

Figure 5.1. A simplified version of the traditional BGP process a router executes when receiving an update from a peer or locally generates a route. The phases refer to the in-depth process described in Chapter 3.2.

SBGPAttr. All data structures are included in Appendix A. The SRoute structure contains the prefix information, peer address, and a pointer to the route attributes, which are contained in SBGPAttr. SBGPAttr contains origin, AS path, next hop, local_pref, and is also where we included a pointer to our attribute. The use of a pointer rather than simply adding our attribute to the list of BGP attributes for implementing our scenarios is explained in the next section.

5.1.4.1 iBGP Modifications

An iBGP implementation of the BGP-TE attributes allows for a router to advertise the properties of its external connections via BGP, similar to methods currently available with OSPF-TE. Even if advertisements are not allowed across domain boundaries, a domain may wish to implement BGP-TE internally to allow for the rank ordering of egress points or for distributing TE information via BGP.

Compared to traditional iBGP implementations, iBGP-TE needs to add the BGP-TE attributes to all inbound advertisements and remove the attributes before placement into an eBGP peer's Adj-RIB-Out table. To ensure this is performed correctly in C-BGP, we added a check of the neighbor's AS prior to placement of the route into an Adj-RIB table. For routes received from eBGP peers or generated locally, the attribute must be created and added to the route prior to placement in the Adj-RIB-In or Loc-RIB table. In contrast, routes destined for eBGP peers must have the attributes removed prior to placement into the Adj-RIB-Out table. Advertisements to iBGP peers should not require the attributes to be added or

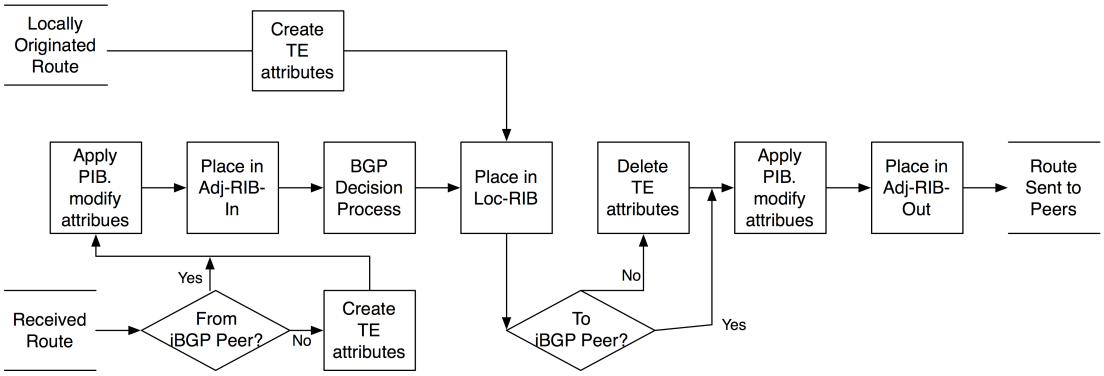

Figure 5.2. The BGP process, modified for including the iBGP-TE attribute

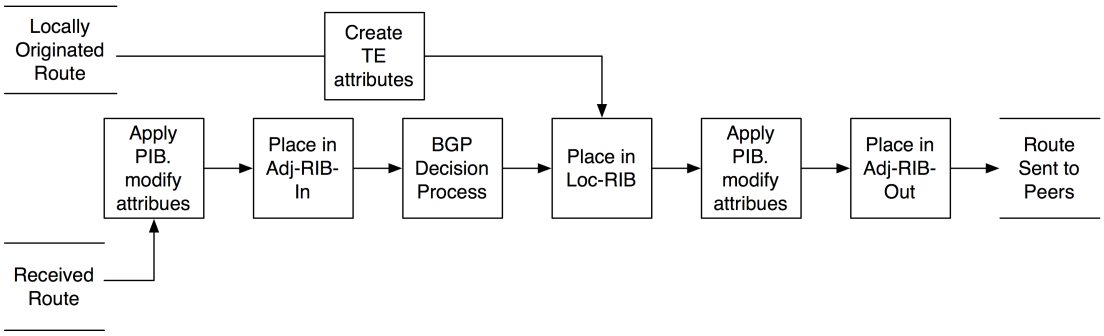

Figure 5.3. The BGP process, modified for including the BGP-TE attribute for inter-domain advertisements

deleted, as long as the attribute is attached prior to placement into the Loc-RIB. Figure 5.2 depicts the changes and decisions required for implementing our iBGP modification in C-BGP.

5.1.4.2 eBGP Modifications

The full implementation of BGP-TE eliminates the need for AS number checks. When a router locally generates a route, it will add the attribute prior to placement of the route in its LOC-RIB table. The attributes are propagated to all peers and included in each Adj-RIB-Out or Adj-RIB-In table, regardless of the peer relationship. Figure 5.3 shows the process changes required in comparison with the traditional BGP implementation.

5.2 Results and Analyses

We ran our scenarios, different combinations of the topologies and number of generated prefixes, to measure the resulting increase in routing table memory consumption. To ensure the accuracy of the simulation results, we ran each of the smaller simulations 10 times for statistical confidence. The larger simulations required such a large amount of time and memory we were only able to simulate each scenario 1-2 times.

To graph our results, we used the following equation to track memory demand:

$$Memory \approx \alpha N + \beta\rho N^2 + \gamma\rho N^2 r + \delta \tag{5.1}$$

Where N is the number of ASes, ρ is the number of prefixes each router originates, r is the average number of neighbors per router, and α, β, and δ are proportionality constants [68]. Each term of the equation accounts for a different aspect of each router in the simulation; the term αN accounts of the initialization overhead incurred for each router. The $\beta\rho N^2$ term accounts for each router's Loc-RIB table, and $\gamma\rho N^2 r$ accounts for the Adj-RIB-In and Adj-RIB-Out tables [68]. Unfortunately, the memtime utility only provides us with the ability to calculate how much memory the overall simulation consumed. For the amount of memory required per AS, we were able to determine the overall memory consumed for the simulation and average the memory consumed over each router involved. From the average memory consumed per AS for each simulation, we used Equation 5.1 to generate the results for the simulations beyond our computing capabilities.

Figure 5.4 shows the full results of our experiments. The increase in the number of ASes causes the size requirements to increase more rapidly, due to the increased overhead the simulation must maintain per router. Also, additional routers create additional RIB tables according to their total number of neighbors, as well as adding an Adj-RIB-In and Adj-RIB-Out table in each neighbor [5], signified by the $\beta\rho N^2$ portion of the equation. If we look at a cross section of the graph while keeping the number of ASes constant, as seen in Figure 5.5 with nine ASes, we can see the number of prefixes increases the size of the routing table linearly. This is not unexpected; if the number of ASes is held constant, each additional prefix will be added to a fixed number of RIB tables [5].

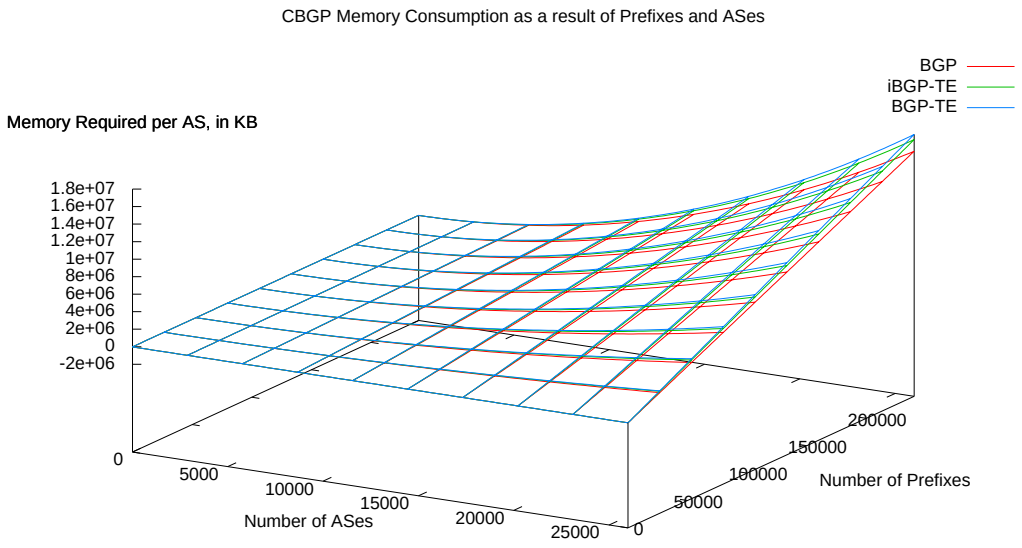

Figure 5.4. The memory requirements for each router when simulating a full Internet scale topology using traditional BGP, iBGP-TE, and BGP-TE.

However, with the cross section we can also see the BGP-TE attribute requires approximately a 50% increase in simulated routing table memory consumption. The increase can be seen by examining the traditional attributes associated with a BGP NLRI advertisement, which require approximately 40 bytes to advertise, as seen in Appendix Tables A.1 and A.2. The attributes included in the BGP-TE attribute, see Table A.3, require 40 bytes to advertise as well, which leads to doubling the size of each NLRI's attributes and accounts for the increase in memory consumption. The cost as a percentage of the overall simulation decreases with the increased number of ASes since the simulation must maintain a larger amount of overhead for the increased number of routers and longer AS paths, but the attribute still requires an increase in routing table size to implement.

While a 50% increase in attribute overhead is significant, it does represent a worst case scenario. Optimizations could reduce the penalty for implementing the BGP-TE attribute. One such optimization would be to consolidate TE attributes into a single instance allowing a router to associate a single BGP-TE attribute with multiple routes for a neighbor.

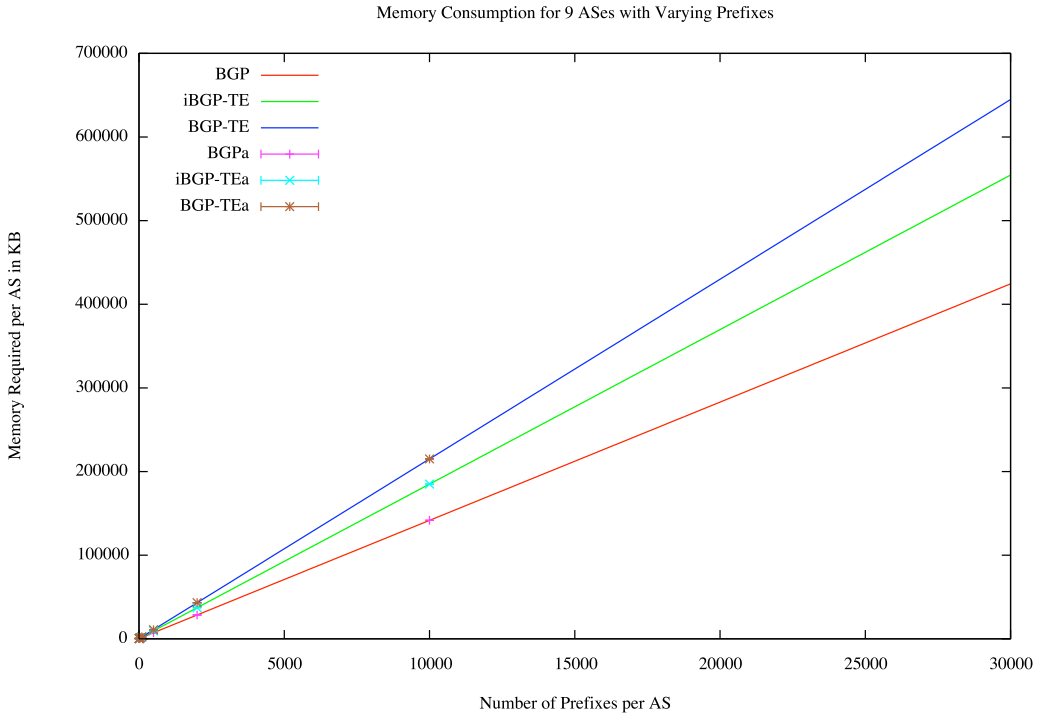

Figure 5.5. A cross section of the full memory requirements graph holding the number of ASes fixed at nine

5.3 Conclusions

The advertisement of TE attributes across domain boundaries can assist in the establishment of interdomain LSPs by better informing a router's decision process of a path's quality and reducing failed LSP requests, similar to what TE attribute advertisements perform for IGPs. However, BGP was designed to convey the minimal amount of information related to potential paths towards a destination. BGP's design makes it difficult to advertise additional information without incurring a fairly large percentage increase in overhead. However, a known tradeoff exists between having a more accurate network picture and reducing a protocol's operating overhead. To convey a more accurate picture of the network, whether it be by advertising multiple alternate paths or by including TE attributes with BGP advertisements, any additional information will increase the size of routing tables.

Our proposed BGP-TE attribute is able to convey information related to a

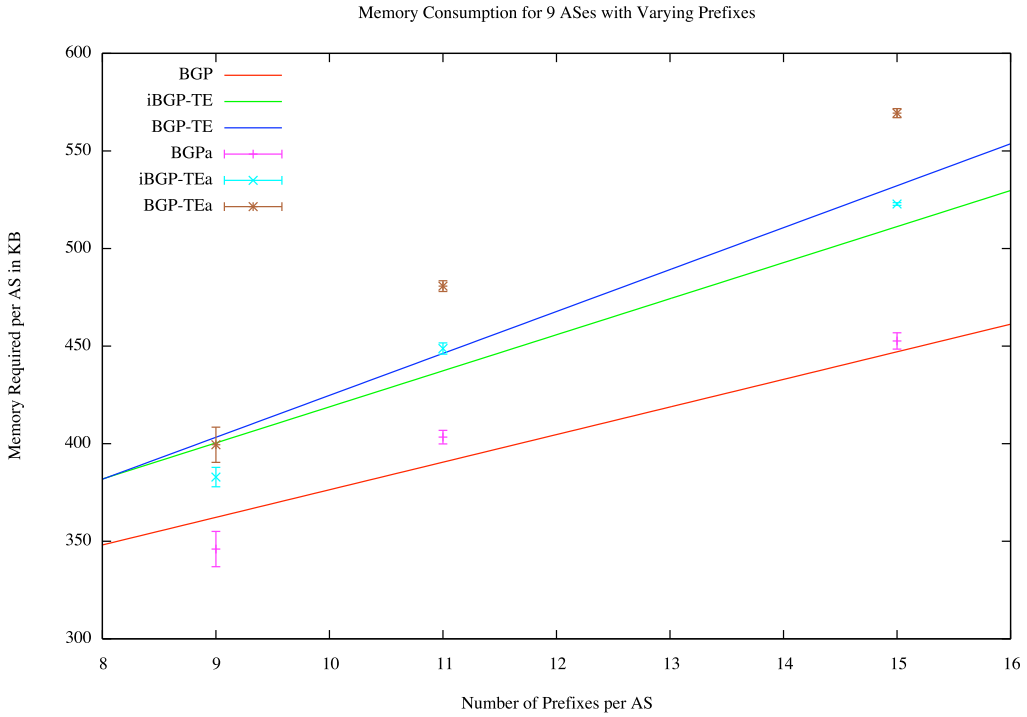

Figure 5.6. A closer view of the cross section graph, depicting the statistical confidence of the simulation results. The simulation results are designated with a trailing "a".

path's quality which is compatible with the attributes used by IGPs. Simultaneously, our proposal maintains each AS's requirement for privacy and help to reduce request rejections by providing TE information prior to a request being issued. The advertisement of TE attributes does not come without a cost; a small, but non negligible, increase in routing table memory consumption is required.

Appendix A

Data Structures

The data structures required in C-BGP for advertising a route. The size represents the minimum number of bytes required for each attribute. Pointers are designated with an asterisk (*) and will incur additional memory requirements when populated.

C-BGP Route Structure	Size in bytes
Prefix	5
Peer*	4
Flags (best, feasible, eligible, etc.)	2
Attributes*	4

Table A.1. Structure of a C-BGP route

C-BGP Route Attribute	Size in bytes
NextHop	4
Origin	1
AS Path*	4
Communities*	4
LocalPref	4
MED	4
Extended Communities*	4
TE Attributes*	4
/* Route-Reflection attributes */	—
Originator*	4
ClusterList*	4

Table A.2. Route attributes and their sizes

BGP-TE Attributes	Size in bytes
Overhead	4
Bandwidth[8]	32
Switching Capacity*	4

Table A.3. The BGP-TE attribute's components and their sizes

Bibliography

[1] D. KATZ, D. Y., K KOMPELLA, "Traffic Engineering (TE) Extensions to OSPF Version 2. RFC 3630," September 2003.

[2] PETERSON, L. L. and B. S. DAVIE *Computer Networks: A Systems Approach, 3rd Edition*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[3] VASSEUR, J.-P., A. AYYANGAR, and R. ZHANG, "A Per-domain path computation method for establishing Inter-domain Traffic Engineering (TE) Label Switched Paths (LSPs)," Internet draft, draft-ietf-ccamp-inter-domain-pd-path-comp-06, work in progress, November 2007.

[4] Y. REKHTER, S. H., T. LI, "A Border Gateway Protocol 4 (BGP-4). RFC 4271," January 2006.

[5] QUOITIN, B. *BGP-based Interdomain Traffic Engineering*, PhD dissertation, Université catholique de Louvain, August 2006.

[6] *OSPF Design Guide*, *Tech. rep.*, Cisco, available at http://www.cisco.com/warp/public/104/2.html. February 2004.

[7] AWDUCHE, D., J. MALCOLM, J. AGOGBUA, M. O'DELL, and J. MCMANUS, "Requirements for traffic engineering over MPLS. RFC 2702," September 1999.

[8] MCDYSAN, D. and D. PAW *ATM and MPLS Theory and Application: Foundations of Multi-Service Networking*, McGraw-Hill, Inc., New York, NY, USA, 2002.

[9] CROWCROFT, J. "Net Neutrality: The Technical Side of the Debate: a White Paper," *Computer Communication Review*, **37**, pp. 49–55, 2007.

[10] M. ALLMAN, W. S., V. PAXSON, "TCP Congestion Control. RFC 2581," August 1999.

[11] M. Allman, C. P., S. Floyd, "Increasing TCP's Initial Window. RFC 3390," October 2002.

[12] Postel, J., "User Datagram Protocol. RFC 768," August 1980.

[13] Crawley, E., R. Nair, B. Rajagopalan, and H. Sandick, "A Framework for QoS-based Routing in the Internet. RFC 2386," August 1998.

[14] Quoitin, B., C. Pelsser, O. Bonaventure, and S. Uhlig "A performance evaluation of BGP-based traffic engineering," *Int. J. Netw. Manag.*, **15**(3), pp. 177–191, 2005.

[15] Fortz, B., J. Rexford, and M. Thorup "Traffic Engineering with Traditional IP Routing Protocols," *IEEE Communications Magazine*, **40**(10), pp. 118–124, October 2002.

[16] Ould-Brahim, H., D. Fedyk, and Y. Rekhter, "Traffic Engineering Attribute," Technical report, draft-fedyk-bgp-te-attribute-03, June 2007.

[17] Braden, R., D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview. RFC 1633," June 1994.

[18] Nichols, K., S. Blake, F. Baker, and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. RFC 2474," December 1998.

[19] Awduche, D., A. Chiu, A. elwalid, I. Widjaja, and X. Xiao, "Overview and Principles of Internet Traffic Engineering. RFC 3272," May 2002.

[20] Ayyangar, A., K. Kompella, J.-P. Vasseur, and A. Farrel, "Label Switched Path Stitching with Generalized Multiprotocol Label Switching Traffic Engineering (GMPLS TE)," Technical report, draft-ietf-ccamp-lsp-stitching-06, April 2007.

[21] Farrel, A., A. Ayyangar, and J.-P. Vasseur, "Inter domain Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) Traffic Engineering - RSVP-TE extensions," Internet draft, draft-ietf-ccamp-inter-domain-rsvp-te-07, work in progress, September 2007.

[22] Farrel, A., A. Satyanarayana, A. Iwata, N. Fujita, and G. Ash, "Crankback Signaling Extensions for MPLS and GMPLS RSVP-TE. RFC 4920," July 2007.

[23] Meyer, D., "RouteViews Project," Available at http://www.routeviews.org. January 2005.

[24] Moy, J., "OSPF Version 2. RFC 2328," April 1998.

[25] Dijkstra, E. W. "A Note on Two Problems in Connexion With Graphs," *Numerische Mathematik*, **1**, pp. 269 – 271, 1959.

[26] Moy, J., "Multicast Extensions to OSPF. RFC 1584," March 1994.

[27] Coltun, R., D. Ferguson, and J. Moy, "OSPF for IPv6. RFC 2740," December 1999.

[28] Murphy, P., "The OSPF Not-So-Stubby Area (NSSA) Option. RFC 3101," January 2003.

[29] Coltun, R., "The OSPF Opaque LSA Option. RFC 2370," July 1998.

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

[31] H. Smit, T. L., "Intermediate System to Intermediate System (IS-IS) Extension fo rTraffic Engineering (TE). RFC 3784," June 2004.

[32] Alnuweiri, H. M., L.-Y. K. Wong, and T. Al-Khasib "Performance of new link state advertisement mechanisms in routing protocols with traffic engineering extensions," *IEEE Communications Magazine, May 2004*, **42**(5), pp. 151–162, May 2004.

[33] Rosen, E., A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture. RFC 3031," January 2001.

[34] Anderson, L., P. Doolan, N. Feldman, A. Fredette, and B. Thomas, "LDP Specification. RFC 3036," January 2001.

[35] Braden, R., L. Zhang, s. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. RFC 2205," September 1997.

[36] Awduche, D., L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels. RFC 3209," December 2001.

[37] Lee, C., A. Farrel, and S. D. Cnodder, "ReserVation Protocol-Traffic Engineering (RSVP-TE). RFC 4874," April 2007.

[38] Banerjee, G. and D. Sidhu "Comparative analysis of path computation techniques for MPLS traffic engineering," *Comput. Networks*, **40**(1), pp. 149–165, 2002.

[39] Costa, L. H. M. K., S. Fdida, and O. C. M. B. Duarte "Developing scalable protocols for three-metric QoS routing," *Comput. Networks*, **39**(6), pp. 713–727, 2002.

[40] Pompili, D., C. Scoglio, and V. C. Gungor "VFMAs, Virtual-flow Multipath Algorithms for MPLS," *Communications, 2006. ICC'06. IEEE International Conference on*, **2**, pp. 652–657, 2006.

[41] Lee, K., A. Toguyeni, and A. Rahmani "Hybrid Multipath Routing Algorithm for Load Balancing in MPLS Based IP Network," in *AINA '06: Proceedings of the 20th International Conference on Advanced Information Networking and Applications - Volume 1 (AINA'06)*, IEEE Computer Society, Washington, DC, USA, pp. 165–172, 2006.

[42] Bates, T., "The CIDR Report," Available at http://www.cidr-report.org. January 2008.

[43] Bates, T., E. Chen, and R. Chandra, "BGP Route Reflection - An Alternative to Full Mesh Internal BGP (IBGP). RFC 4456," April 2006.

[44] Winick, J., S. Jamin, and J. Rexford *Traffic engineering between neighboring domains.*, *Tech. rep.*, AT&T Labs-Research, http://www.research.att.com/~jrex/papers/interAS.pdf, 2002.

[45] Quoitin, B., S. Uhlig, C. Pelsser, L. Swinnen, and O. Bonaventure "Interdomain traffic engineering with BGP," *IEEE Communications Magazine*, **41**(5), pp. 122–128, May 2003.

[46] Pelsser, C. *Interdomain Traffic Engineering with MPLS*, PhD dissertation, Computer Sicence and Engineering Department, Universite catholique le Louvain, Belgium, October 2006.

[47] Zhang, R. and J. Vasseur, "MPLS Inter-Autonomous System (AS) Traffic Engineering (TE) Requirements. RFC 4216," November 2005.

[48] Farrel, A., J. Vasseur, and A. Ayyangar, "A Framework for Inter-Domain Multiprotocol Label Switching Traffic Engineering. RFC 4726," November 2006.

[49] Holness, F. M. *Congestion Control Mechanisms within MPLS Networks*, PhD dissertation, Queen Mary and Westfield College, University of London, Department of Electronic Engineering, 2000.

[50] Lien, P. H. and V. Q. Son "Congestion control using Fast Acting Traffic Engineering Mechanism in Strict and Loose MPLS Networks," Proceedings of the International Symposium on Electrical & Engineering (ISEE), 2005.

[51] Salvadori, E. and R. Battiti "A Load Balancing Scheme for Congestion Control in MPLS Networks," *iscc*, **0**, pp. 951–956, 2003.

[52] Salvadori, E., R. Battiti, and F. Ardito, "Lazy Rerouting for MPLS Traffic Engineering," Technical report, Universita di Trento, Dipartimento di Informatica e Telecomunicazioni, March 2003.

[53] Farrel, A., J.-P. Vasseur, and J. Ash, "A Path Computation Element (PCE)-Based Architecture. RFC 4655," August 2006.

[54] Roux, J. L. L., Jean-Phillipe, Y. Ikejiri, and R. Zhang, "OSPF Protocol Extensions for Path Computation Element (PCE) Discovery. RFC 5088," January 2008.

[55] Kumaki, K. and T. Murai, "BGP protocol extensions using attribute for Path Computation Element (PCE) Discovery," Internet draft, draft-kumaki-pce-bgp-disco-attribute-00, work in progress, November 2007.

[56] Bonaventure, O., "Using BGP to distribute flexible QoS information," Techincal Report, draft-bonaventure-bgp-qos-00, February 2000.

[57] Cristallo, G. and C. Jacquenet, "Providing Quality of Service Indication by the BGP-4 Protocol: the QoS_NLRI attribute," Technical Report, draft-jacquenet-qos-nlri-05, June 2003.

[58] ———, "The BGP QOS_NLRI Attribute," Technical Report, draft-jacquenet-bgp-qos-00, February 2004.

[59] Abarbanel, B. and S. Venkatachalam, "BGP-4 support for Traffic Engineering," Technical Report, draft-abarbanel-idr-bgp4-te-01, September 2000.

[60] Vijayanand, C. and S. Bhattacharya, "A BGP Based Method to Compute Inter-AS Traffic Engineering Label Switched Paths with PCE," Internet draft, draft-vijay-somen-pce-bgp-interas-te-path-01, July 2007.

[61] Zhang, T., Y. Cui, Y. Zhao, L. Fu, and T. Korkmaz "Scalable BGP QoS Extension with Multiple Metrics," *Networking and Services, 2006. ICNS '06. International conference on*, p. 80, 2006.

[62] Walton, D., A. Retana, and E. Chen, "Advertisement of Multiple Paths in BGP," Technical Report, draft-walton-bgp-add-paths-05, August 2006.

[63] Bhatia, M. and J. M. Halpern, "Advertising Multiple NextHop Routes in BGP," Technical report, draft-bhatia-bgp-multiple-next-hops-01, work in progress, August 2006.

[64] Xu, W. and J. Rexford "MIRO: multi-path interdomain routing," *SIGCOMM Comput. Commun. Rev.*, **36**(4), pp. 171–182, 2006.

[65] Quoitin, B. and S. Uhlig "Modeling the Routing of an Autonomous System with C-BGP," *Network, IEEE*, **19**(6), pp. 12–19, Nov.-Dec 2005.

[66] Bengtsson, J., "Memtime," Available at http://www.update.uu.se/ ˜johanb/memtime, 2002.

[67] Norton, W. B. "Internet Service Providers and Peering," in *Proceedings of NANOG 19*, Albuquerque, New Mexico, June 2000.

[68] Dimitropoulos, X. A. and G. F. Riley "Large-Scale Simulation Models of BGP," in *MASCOTS '04: Proceedings of the The IEEE Computer Society's 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS'04)*, IEEE Computer Society, Washington, DC, USA, pp. 287–294, 2004.