MPLS & QoS in Virtual Environments

Amar Madhavan
axm4836@rit.edu

Follow this and additional works at: https://scholarworks.rit.edu/theses

Recommended Citation

This Thesis is brought to you for free and open access by RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.
Master of Science in

Applied Networking and Systems Administration

Thesis Approval Form

Student Name: Amar Madhavan

Thesis Title: MPLS & QoS in Virtual Environments

Thesis Area: ☒ Networking ☐ Systems Administration

☐ Security ☐ Other ______________

MS Thesis Committee

Name                  Signature                  Date

Ali Raza

Chair

Harry Manifavas

Committee Member

Charles Border

Committee Member
MPLS & QoS in Virtual Environments

by

Amar Madhavan

Committee Members:

Ali Raza

Harry Manifavas

Charles Border

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Networking & Systems Administration

B. Thomas Golisano College

Of

Computing and Information Sciences

Rochester Institute of Technology

RIT Dubai

April 5, 2017
Abstract

The rise of high performance computing has seen a shift of services from locally managed Data Centers, to centralized globally redundant Data Centers (Cloud Computing). The scale of operation and churn required for cloud computing has in turn led to the rise of faster and programmable network pathing, via SDN & NFV. Cloud compute resources are accessible to individual researchers, as well as larger organizations. Cloud computing relies heavily on virtualization and abstraction of resources. The interconnect between these resources is more complex than ever, due to the need to seamlessly move from virtual to physical to hybrid networks and resources. MPLS as a technology is robust and has been used as transport for decades with a good track record. QoS has been available within most protocols to ensure service levels are maintained. The integration of MPLS, QoS and virtual environments is a space of increasing interest. It would allow for the seamless movement of traffic from end to end without the need for specialized hardware or vendor lock-in.

In this thesis, the performance gains of IP/MPLS networks utilizing QoS on commercially available virtual environments has been investigated and studied. Latency was captured via round trip time metrics and tabulated for voice, video and data, with QoS and congestion as the primary differentiators. The study discusses the approach taken, the common thinking, and finally analyzes the results of a simulation, in order to show that MPLS & QoS benefits are viable in virtualized environments.
# Table of Contents

Thesis Approval Form ........................................................................................................... 1

MPLS & QoS in Virtual Environments .................................................................................. 2

Abstract ................................................................................................................................. 3

Table of Contents .................................................................................................................. 4

Introduction ............................................................................................................................. 8

Technology in the Industry ................................................................................................. 10

MPLS Transport ..................................................................................................................... 10

Service Levels & QoS ............................................................................................................. 12

Virtualization and NFV ........................................................................................................ 17

Simulation ............................................................................................................................... 19

Components ........................................................................................................................... 19

Virtual Machine Network Architecture ................................................................................ 22

Design and Methodology ..................................................................................................... 25

Scenario 1 – No Quality of Service; Low Congestion ......................................................... 33

Scenario 2 – No Quality of Service; High Congestion ......................................................... 35

Scenario 3 – Quality of Service; Low Congestion ............................................................... 37
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario 4 – Quality of Service; High Congestion</td>
<td>39</td>
</tr>
<tr>
<td>Checks</td>
<td>41</td>
</tr>
<tr>
<td>Statistical Data Capture</td>
<td>42</td>
</tr>
<tr>
<td>Roadblocks</td>
<td>44</td>
</tr>
<tr>
<td>Results &amp; Analysis</td>
<td>47</td>
</tr>
<tr>
<td>Scenario 1 – Voice</td>
<td>47</td>
</tr>
<tr>
<td>Scenario 1 – Video</td>
<td>48</td>
</tr>
<tr>
<td>Scenario 1 – Data</td>
<td>49</td>
</tr>
<tr>
<td>Scenario 1 – Observation</td>
<td>50</td>
</tr>
<tr>
<td>Scenario 2 – Voice</td>
<td>51</td>
</tr>
<tr>
<td>Scenario 2 - Video</td>
<td>52</td>
</tr>
<tr>
<td>Scenario 2 - Data</td>
<td>53</td>
</tr>
<tr>
<td>Scenario 2 – Observation</td>
<td>54</td>
</tr>
<tr>
<td>Scenario 3 – Voice</td>
<td>55</td>
</tr>
<tr>
<td>Scenario 3 – Video</td>
<td>56</td>
</tr>
<tr>
<td>Scenario 3 - Data</td>
<td>57</td>
</tr>
<tr>
<td>Scenario 3 – Observation</td>
<td>58</td>
</tr>
<tr>
<td>Scenario 4 – Voice</td>
<td>59</td>
</tr>
</tbody>
</table>

Amar Madhavan
Scenario 4 - Video ........................................................................................................60

Scenario 4 - Data .........................................................................................................61

Scenario 4 – Observation ..............................................................................................62

Comparative Analysis ..................................................................................................63

Conclusion ...................................................................................................................65

Future Work ..................................................................................................................66

Appendix A – Router Configuration ............................................................................68

CE-1 .............................................................................................................................68

PE-1 .............................................................................................................................74

P-1 ................................................................................................................................81

P-2 ................................................................................................................................85

P-3 ................................................................................................................................90

P-4 ................................................................................................................................94

P-5 ................................................................................................................................98

P-6 ...............................................................................................................................102

P-7 ................................................................................................................................106

P-8 ................................................................................................................................111

P-9 ................................................................................................................................115
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-10</td>
<td>119</td>
</tr>
<tr>
<td>P-11</td>
<td>124</td>
</tr>
<tr>
<td>PE-2</td>
<td>128</td>
</tr>
<tr>
<td>CE-2</td>
<td>134</td>
</tr>
<tr>
<td>References</td>
<td>137</td>
</tr>
</tbody>
</table>
Introduction

The last two decades has seen the internet become a major part of every person’s life. There have been many calls to have access to the internet specified as a fundamental right of every human being, and the United Nations has even passed a resolution safeguarding this right (United Nations, 2016). While this has meant greater connectivity and the shrinking of the world, it has also meant that the already congested pathways used by the internet has had to look for ways to manage this traffic and ensure availability of both the traversing infrastructure as well as the packets on the wire. A quick analogy is that of Dubai’s own Sheikh Zayed road, which no matter the upgrade, always finds itself congested with traffic during peak hours. While the government takes care of the roads, the internet ‘roads’ are taken care of by various Internet Service Providers (ISPs). ISPs are businesses and are unlike governments where breaking even on cost and infrastructure efficiency is secondary to the public interest. In an ISP, reuse of existing infrastructure with low touch high reward technologies are seen as the way forward. With this perspective in mind the addition of MPLS & QoS has allowed existing IP based devices to deliver more service with the same equipment.

The use of MPLS & QoS in a service provider network is not a new concept. What is newer is the usage of MPLS to connect two disparate locations together thereby allowing them to work as one for compute intensive resources. This newer concept is seen when datacenters at different locations need to have direct LAN-to-LAN connectivity with each other. The concepts of virtualized computing, and even NFV has only seen an increase in the concept of encapsulated traffic running over shared infrastructures, and even here MPLS has been seen as one of the options for bridging the gap. This means that QoS effects on MPLS should be
studied in more detail, as there is still much more that it can provide. Indeed, the aim of this thesis is to showcase this differentiation.
Technology in the Industry

**MPLS Transport**

With so many people online, and more coming online every day, the stress on Internet Service Providers (ISP) has continued to mount. This has driven investment in multipurpose edge and core technologies so as to allow ISPs to provide service to both individual consumers as well as organizations. In a whitepaper from Alcatel (Bocci, Watkinson, & Aissaoui, 2004), this approach is outlined and described in further detail.

IP as a protocol was limited to the edge of ISP networks, with ATM being favored for the core. Multiple advantages caused this adoption of ATM over IP, although it necessitated multiple architectures. MPLS was able to mitigate some of these flaws with the added advantage of collapsing the complexity associated with maintaining differing architectures. Fundamentally the issue with using IP throughout the network was the limitations inherent in dynamic routing protocols. Three of the points that were examined in additional detail are described in the next paragraphs.

“Routing functionality in IP is notoriously difficult to evolve because of the close coupling between the control and forwarding components” (Escobar, 2001). I agree with Escobar in this context and indeed this was one of the drawbacks that MPLS was designed to address. *Figure 1* shows the concept of separating these two functions.

![Figure 1 - Conceptual view of MPLS control and forwarding plane (Jabbarib & Daniel, 2002)](image-url)
Another problem with IP routing protocols like RIP, OSPF & ISIS was that they did not take into consideration capacity and traffic constraints while making decisions (Jabbarib & Daniel, 2002). Jabbarib & Daniel make a very salient point here and this too was addressed in MPLS utilizing additional constraints in the standard Djikstra (Shortest Path First) algorithm (Ziegelmann, 2010).

An additional overhead was the requirement for every single router to examine Layer 3 packet information, and then to perform a processor intensive search for a matching destination within its routing table. This process of looking for the appropriate forwarding next hop, was distilled into a combination of two functions – a first function that partitioned a group of packets (with the same final destination) into a single Forwarding Equivalence Class (FEC); and a second function that linked every FEC to a next hop (Rosen, Viswanathan, & Callon, 2001).

MPLS is a scalable protocol independent transport method, that is often referred to as a layer 2.5 protocol. It sits between the typical layer 2 (ex: Ethernet) and layer 3 protocols (ex: IP) and allows for a unified data transport service. Each MPLS packet is simplified in comparison to normal protocol packets. The defining feature on these packets is the existence of a 20-bit label value (Rosen, et al.). This label value is directly linked to an outgoing link on every router, and allows MPLS routers to simply read the label and send the packet out, without having to inspect further.

While MPLS succeeded as a converged transport technology decoupling Layer 2 and Layer 3, the rapid adoption of the internet also saw the rise of corporate consumers with their own extensive network domain, requiring transport via external carriers. Here ISPs had typically
positioned Leased Circuits. The high cost of these dedicated lines, did not match customer demands, and thus many ISPs started using their existing converged MPLS network to carry customer traffic to and from fixed ingress/egress points in their network. MPLS labels facilitated the separation of customer traffic at the forwarding plane, but the control plane was not so easily separated. Customer routing was merged with Service Provider routes which decreased scalability, and increased complexity in terms of deployment, operation and troubleshooting. Here Layer 3 VPNs provided a mechanism for the separation of the Service Provider’s routing from that of their customers (Rosen & Rekhter, 2006).

MPLS VPNs provide the ability to carry multiple customer's traffic over the same backbone, whilst providing isolation between each other, and the carriers native traffic. Additionally, the unique and possibly overlapping routing tables of different customers are maintained in isolated instances on the edge of the carrier's network. Through BGP these individual routing tables can communicate and exchange information, with other routing tables, as long as they belong to the same customer (Rosen & Rekhter, RFC 4364, 2006). This provides isolation of the data plane through MPLS, and isolation of the control plane through BGP.

**Service Levels & QoS**

An important feature of corporate services is the provision of an SLA or Service Level Agreement. These provide a guarantee of certain conditions from the Service Provider to the corporate customer, in exchange for which a premium charge is levied. This was much simpler to provide when corporates made use of dedicated connections, but not so when providing similar services on the collapsed IP-MPLS backhaul network. “The conventional approach that treats the link entities as monolithic is no longer applicable. There is a need to
further express link entities by path entities, where the traffic through each path may belong
to a specific traffic class.” (Lu, 2004)

Networks that do not have a provision for service guarantees are considered best effort
networks. Corporate applications require some form of a data delivery guarantee and it is up
to the service provider to ensure that these requirements are provided. Whilst the requirement
seems straightforward it is far from it, owing to the heterogeneous nature of today’s
networks, that involve a number of local area access architectures (wired and wireless) and an
equal number and variety of wide area architectures (Hunt, 2002)

QoS provides a method to mark and prioritize specific flows of traffic, so as to ensure that
they are always provided certain resources. In much the same way that traffic makes way for
an ambulance, high priority traffic will supersede lower priority traffic. This is done through
a variety of mechanisms, where packets are marked or tagged with a code that specifies their
importance. The earliest such mechanism for IP is IP precedence (DARPA).

When the architects of MPLS Layer 3 VPN technologies were designing the Layer 3
framework, the fact that quality of service was a key component of VPN services was made
note of. Towards this end, it was even suggested that the three (3) bits marked ‘experimental’
in the MPLS shim header be utilized (Rosen & Rekhter, 2006).

That MPLS packet processing is given considerable priority on an MPLS aware router is
something that Zhu demonstrates in his thesis. He has shown how generating a high number
of large packets, behavior that would normally overwhelm his testbed, was processed without
significant impact due to the built-in load balancing mechanism that are inherent to MPLS
enabled VPNs (Zhu, 2001). In figure 2 this behavior is depicted where at the 12:14 mark a
traffic generator starts to produce 200,000 packets per second of size 16,000, with no significant variation (Zhu, 2001)

A separate study conducted by Maini, utilizing Linux machines as MPLS aware routers, found that as the data to be transferred increases, the MPLS aware systems, performed better because of its ability to reserve bandwidth and therefore avoid dropping packets (Maini, 2002). Figure 3 & Figure 4 show the improvement that MPLS has over standalone IP.
The ability to provide higher bandwidth with delivery guarantees, and the quest for cheaper infrastructure saw the moving of telephony services to the network. Voice over IP (VoIP) is today an integral part of the network traffic mix and had a traffic profile very different from anything prior that flowed on the network. Typical network traffic has a high tolerance for latency and has a high ability to recover from packets lost on the network. VoIP has a traffic profile that is the exact opposite of this, and as opined by Tan, the idea behind CoS is to classify network traffic into multiple priority levels for traffic differentiation. Prioritized VoIP packets will experience a lower loss probability and an improvement in end-to-end delay when traversing the network (Tan, 2002).

These days, video is a major component of internet traffic, so much so, that on mobile networks it is 55% of the traffic and is projected to become as much as 75% by 2020 (Cisco, 2016). Video has the advantage of having a traffic profile similar to voice, and coming into the network mix after these traffic delay optimizations were worked out for voice. Bartlett & Wetzel mention that large multinational enterprises are picking MPLS for their networks. This is due to its ability to handle video and voice traffic along with mission critical applications, whilst maintaining application performance, as well as providing guarantees for

![Figure 4 - Transfer times as the size of the network increases (Maini, 2002)](image-url)
the voice and video. Additionally, the ability to outsource the management of these protocols (due to the control plane separation) provides further consideration (Bartlett & Wetzel, 2006).

Mobile traffic also benefits from MPLS with QoS as made evident by Palmieri when he mentions, “In our MPLS-powered mobile network backbone, both static and dynamically routed paths can be setup to bypass congested areas and take advantage of underutilized network resources without having to modify IGP link metrics” (Palmieri, 2005).

Carter, writing for BT’s technology journal talks about how MPLS-VPN technology has emerged as a success story in data networks (Carter, 2005). He also talks about network delay on data packets, and how these can be sub-divided into transport delay, and router queuing delay (Carter, 2005). QoS provides a solution in negating these delays to a large extent by virtue of CoS markings utilized between transport devices, and in device prioritization helping to mitigate queuing delay.

By applying Diffserv (Differential Services) in such a fashion, the benefits that can be realized as the requiring of less bandwidth per link to achieve the same SLA as a non-Diffserv case and in more aggregate traffic being supported for the same provisioned bandwidth as the non-Diffserv case (Evans & Filsfils, 2007).

![Figure 5 - Diffserv Bandwidth gains (Evans & Filsfils, 2007)](image)
In figure 5 the most significant relative benefits in terms of bandwidth savings from deploying Diffserv are achieved when the proportion of Class 1 load (the class with the tightest SLA commitment and therefore the highest over-provisioning factor) is low relative to the Class 2 load, and when the ratio of the over-provisioning factor for Class 1 is high relative to the over-provisioning factor for Class 2 (i.e. OP$_1$/OP$_2$). Conversely, if all traffic is of the tightest SLA class, intuitively there is no benefit in Diffserv (Evans & Filsfils, 2007).

**Virtualization and NFV**

Virtualization is seeing a lot of adoption these days, with many organizations using it to fuel consolidation and reduce capital expenditure on the latest and greatest IT machines. NFV is the utilization of high volume performance servers to consolidate and perform different network functions, that typically require different hardware platforms running different software and service stacks, supported by different vendors (ETSI).

A distributed approach to NFV involves a setup where routing functionality is deployed within specific areas (a particular vlan interconnecting multiple datacenters located in multiple geographies) and no external routing is required. This results in shorter delays for packet transfers and improved quality of service (Kubayashi, 2016). A centralized approach sees all traffic from multiple areas being directed to a central area where routing is performed (Kubayashi, 2016). This approach has been seen before with OSPF and ISIS routing protocols making use of backbone areas. The primary advantage of NFV, is the programmability of the solution and the ability for Software Defined Networking (SDN) to enable greater scale and faster reactivity through the automation of network path deployment.

The potential benefits from software/hardware independence and automated elasticity for service providers are multiple. Dynamic shaping and tuning of application traffic, through on
demand deployment of network functionality at confluences, helping to limit the complexity of transport network, and reduce the demand on operations staff. Capacity management and provisioning efforts will be reduced as there will be fewer capacity expansion activities needed (Hernandez-Valencia, Izzo, & Polonsky, 2015). The less manual intervention is required, the more stable the system will perform.

Most NFV resource allocation methods have made guarantees for bandwidth between Virtual Network Functions or VNFs (Bari, Chowdhury, Ahmed, & Butaba, 2015), compute time (Meghraghdam, Keller, & Karl, 2014), memory allocation on physical nodes (Mijumbi, et al., 2015) and end to end latency (Luizelli, Bays, Buriol, Barcellos, & Gaspary, 2015). While benchmarks for resource monitoring, run-time resource evaluation and QoS compliant orchestration are proposed and being researched, some parameters like jitter have yet to be considered (Herrera & Botero, 2016).
Simulation

Components

In any network prior to the deployment of a solution, it is necessary that the technology is demonstrated and confirmed to be viable. These tests are called ‘Proof of Concepts’ and can vary from a simple simulation in demo apparatus, to an extensive 3-month cycle of embedded testing in a siloed portion of the customer production network. No single technology exists that can integrate and play well with all known technology in the world, and it falls to the professionals in the industry to get the technologies working with each other.

Test lab environments are finite, and the ability to build on the fly on general purpose compute systems, is something that IT professionals greatly appreciate. Towards this end, utilizing virtualization software and building out complex technologies in a controlled environment, at a fraction of the cost is a lesson that is quickly taken to heart, and has been made use of here, to engineer a testbed that can be used to quantify the hypothesis that is being made in this thesis.

Multiple tools were selected for each phase of this simulation for the purposes of design, simulation, data collection and data analysis. These tools were selected based on their availability and usability.

VMware Workstation, is a type-2 hypervisor, built by VMWare Inc a division of Dell Technologies, that was utilized as the underlying system on which the simulation was built. VMware Workstation runs on top of an existing installation of 64 bit Windows, and thereby provides a self-contained environment that is portable to alternate hardware, as long as the new hardware also runs Windows as it’s OS. VMware Workstation provides a virtual
network editor, that allows the creation of multiple individual virtual network links, named ‘VMnet’. Each VMnet is isolated from each other and behaves as a single layer 2 broadcast domain.

This functionality was a major factor in the utilization of this software, as the testbed required an extensive set of dedicated switch links. All traffic that was generated remained isolated to the virtual network and never exit the type-2 hypervisor.

Upon this environment, Juniper Network vSRX virtual machines were deployed as Customer Edge (CE), Provider Edge (PE) and Provider (P) routers. vSRX are primarily meant as virtual security devices, but also have a pure router (packet based) mode that has rich support for ISIS, MPLS, MPLS VPNs, and Class-of-Service (Juniper Networks). These deployed virtual
machines function as the basic building blocks for the network topology on which the tests are run.

The vSRX machines make use of JunOS, which is a network operating system built and maintained by Juniper Networks (Juniper Networks). It is FreeBSD based (iXsystems), and the same codebase is used across their routing, switching and security portfolio. The advantage of having the same codebase, is that the configurations can be quickly migrated across devices with minimal modification. Indeed, in most cases the only change that would be required, is the renumbering of physical interfaces to match the new hardware. JunOS also supports the partitioning of a single device into separate routing spaces, with independent decision making for each separate routing instance.
Virtual Machine Network Architecture

Figure 7 shows the settings that were used for each VMnet that was created. VMware Workstation limits the total number of VMnets that can be created to twenty (20). For the simulation, a total of nineteen (19) VMnets were utilized. As each VMnet works as a single layer 2 broadcast domain they are akin to a switch. Any interface on a VM that is connected to a different VMnet is essentially connected to a different switch, and hence are isolated from each other. As these VMnets are only needed by the host VMs themselves, the option for “Host-only” is selected. The “Connect a host virtual adapter to this network” option is not selected, as this option links the base machine to the VMs, which was unnecessary in this
case. As the VMs that were connected all had static IPs assigned, there was no need to select the option “Use local DHCP service to distribute IP address to VMs”. The subnet IP can be configured here as well, but is mostly of use if utilizing VMWare DHCP server. This can be likened to a switch connected to a DHCP server that is assigning IPs from a certain pool (ex: 172.16.0.0/12), with an administrator overriding this assignment by manually assigning static IPs to the machines connected to this switch.

Figure 8 shows the mapping between the virtual NICs that provides Ethernet interface capability to each VM. There are a total of twenty-eight (28) point-to-point links required for the logical topology (Figure 10). As point-to-point links are not directly available, each VMnet is assigned only two interfaces, one from each device, and is used in place of a point-to-point link. Due to the limitation of a maximum of 20 VMnets being supported in VMWare Workstation, there are not enough VMnets for the point-to-point links. As the simulation traffic primarily flows over sixteen (16) point-to-point links (Figure 11), the remaining 12 links can be considered to not need as high a level of isolation. These twelve (12) links were
distributed into 3 groups of 4 links each, and one VMnet was assigned to each group. This mapping and grouping of interfaces is listed in figure 9.

While the interfaces grouped on the same VMnet are losing layer 2 isolation, they are still isolated from each other at layer 3, due to the unique 2-host (subnet mask 0xFFFFFFFFFC) networks that are assigned to each interface, and shown in figure 9.

<table>
<thead>
<tr>
<th>VMNet</th>
<th>Link Description</th>
<th>IP Subnet/Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CE1 ge-0/0/1 &lt;-&gt; PE1 ge-0/0/1</td>
<td>20.20.20.0/30</td>
</tr>
<tr>
<td>2</td>
<td>CE1 ge-0/0/3 &lt;-&gt; Win7 Video/Data Nic #1</td>
<td>10.10.20.0/30</td>
</tr>
<tr>
<td>3</td>
<td>CE1 ge-0/0/2 &lt;-&gt; Win7 Voice Nic #2</td>
<td>10.10.10.0/30</td>
</tr>
<tr>
<td>4</td>
<td>PE1 ge-0/0/2 &lt;-&gt; P1 ge-0/0/1</td>
<td>10.10.10.0/30</td>
</tr>
<tr>
<td>5</td>
<td>PE1 ge-0/0/4 &lt;-&gt; P2 ge-0/0/1</td>
<td>10.10.4/30</td>
</tr>
<tr>
<td>6</td>
<td>P1 ge-0/0/3 &lt;-&gt; P3 ge-0/0/1</td>
<td>10.10.12/30</td>
</tr>
<tr>
<td>7</td>
<td>P2 ge-0/0/4 &lt;-&gt; P7 ge-0/0/1</td>
<td>10.10.32/30</td>
</tr>
<tr>
<td>8</td>
<td>P3 ge-0/0/4 &lt;-&gt; P6 ge-0/0/1</td>
<td>10.10.16/30</td>
</tr>
<tr>
<td>9</td>
<td>P5 ge-0/0/1 &lt;-&gt; P6 ge-0/0/3</td>
<td>10.10.48/30</td>
</tr>
<tr>
<td>10</td>
<td>P5 ge-0/0/3 &lt;-&gt; P8 ge-0/0/2</td>
<td>10.10.52/30</td>
</tr>
<tr>
<td>11</td>
<td>P7 ge-0/0/5 &lt;-&gt; PE2 ge-0/0/4</td>
<td>10.10.88/30</td>
</tr>
<tr>
<td>12</td>
<td>P8 ge-0/0/5 &lt;-&gt; P9 ge-0/0/1</td>
<td>10.10.60/30</td>
</tr>
<tr>
<td>13</td>
<td>P9 ge-0/0/2 &lt;-&gt; P10 ge-0/0/2</td>
<td>10.10.76/30</td>
</tr>
<tr>
<td>14</td>
<td>P10 ge-0/0/3 &lt;-&gt; PE2 ge-0/0/3</td>
<td>10.10.80/30</td>
</tr>
<tr>
<td>15</td>
<td>PE2 ge-0/0/1 &lt;-&gt; CE2 ge-0/0/1</td>
<td>20.20.20.4/30</td>
</tr>
<tr>
<td>16</td>
<td>CE2 ge-0/0/4 &lt;-&gt; Linux Listener</td>
<td>10.10.30.0/30</td>
</tr>
<tr>
<td>17</td>
<td>PE1 ge-0/0/3 &lt;-&gt; P3 ge-0/0/2</td>
<td>10.10.20.30</td>
</tr>
<tr>
<td>17</td>
<td>P1 ge-0/0/2 &lt;-&gt; P2 ge-0/0/2</td>
<td>10.10.8/30</td>
</tr>
<tr>
<td>17</td>
<td>P2 ge-0/0/3 &lt;-&gt; P3 ge-0/0/3</td>
<td>10.10.24/30</td>
</tr>
<tr>
<td>17</td>
<td>P3 ge-0/0/5 &lt;-&gt; P4 ge-0/0/1</td>
<td>10.10.28/30</td>
</tr>
<tr>
<td>18</td>
<td>P4 ge-0/0/2 &lt;-&gt; P7 ge-0/0/2</td>
<td>10.10.40/30</td>
</tr>
<tr>
<td>18</td>
<td>P5 ge-0/0/2 &lt;-&gt; P7 ge-0/0/3</td>
<td>10.10.44/30</td>
</tr>
<tr>
<td>18</td>
<td>P6 ge-0/0/2 &lt;-&gt; P8 ge-0/0/1</td>
<td>10.10.36/30</td>
</tr>
<tr>
<td>18</td>
<td>P9 ge-0/0/3 &lt;-&gt; PE2 ge-0/0/2</td>
<td>10.10.72/30</td>
</tr>
<tr>
<td>19</td>
<td>P7 ge-0/0/4 &lt;-&gt; P11 ge-0/0/1</td>
<td>10.10.84/30</td>
</tr>
<tr>
<td>19</td>
<td>P8 ge-0/0/3 &lt;-&gt; P11 ge-0/0/2</td>
<td>10.10.56/30</td>
</tr>
<tr>
<td>19</td>
<td>P8 ge-0/0/4 &lt;-&gt; P10 ge-0/0/1</td>
<td>10.10.64/30</td>
</tr>
<tr>
<td>19</td>
<td>P10 ge-0/0/4 &lt;-&gt; P11 ge-0/0/3</td>
<td>10.10.68/30</td>
</tr>
</tbody>
</table>

*Figure 9 - Map of VMnet to interfaces to network*
**Design and Methodology**

The IGP selected for operation in the service provider core was IS-IS. IS-IS was originally a protocol developed by Digital Equipment Corporation as part of DECnet and was standardized as ISO 10589. It was designed for usage on ISO’s Connectionless-mode Network Service or CLNS networks (ISO/IEC). IS-IS initially did not support the carrying of IP information, but was later extended to allow this ability (Digital Equipment Corporation). The updates in IS-IS allowed for it to be agnostic to the information it was carrying, and thus IS-IS is also able to carry IPv6 information. In contrast OSPF, the dynamic routing protocol native to IPv4, needed to be redesigned and rewritten as OSPFv3 to enable IPv6 functionality. IS-IS’s inherent extendibility also allows it to carry MPLS traffic engineering information internally, while OSPF needed to be extended to enable the same functionality using opaque LSAs (Katz, Kompella, & Yeung). With multiple revisions and extensions, to both protocols over the years, both protocols have the same features and functions available to them. Industry best practices are based on tried and tested principles where IS-IS is considered the choice protocol for large flat topologies (Bhattia).

Windows 7, a consumer OS from Microsoft (Microsoft News Center), was utilized as the traffic generating client in the topology. An instance of Debian 8.5, an open source Unix like operating system (Debian Documentation Team), was utilized for receiving the generated traffic. Wireshark, an open source packet analyzer, was used for packet capture and dissection. VLC, an open source video player, was used to generate a stream of video traffic. Filezilla (Filezilla Project, n.d.), an open source ftp client, was used to generate ftp-data traffic.
Figure 10 depicts the logical topology that was built for this simulation. vSRX virtual machines are being used to simulate the routers in this network. Every link is mapped to a unique VMnet, and simulated a broadcast point-to-point gigabit link. The interfaces within the routers are all manually set to fast-ethernet (100 mb) speeds. All addressing was performed using IPv4.

The dynamic routing protocol IS-IS was configured in the service provider core with a single level 2 area. All internal service provider interfaces were configured to allow MPLS. All communication that is native to the service provider utilized IP packets, while all traffic that is being carried across on behalf of the customer was tunneled across the network utilizing MPLS packets. This MPLS tunnel is more formally known as an MPLS Label Switched Path (LSP). The MPLS tunneling of traffic external to the service provider ensured that minimal changes were made to the packet structure as received from the customer, and provided isolation between service provider IP packets, and customer IP packets.
Two MPLS LSP paths were defined as shown in figure 11. The primary path is the path that was used to send any incoming packet from Site 1 to Site 2. The routers that are part of the primary path are, in order from ingress to egress, PE-1, P2, P7, PE-2. Another advantage of utilizing MPLS LSPs, was that once the original customer packet is inserted into the LSP at the head end of the path, all subsequent routers in the provider network did not need to read IP information, and simply had to forward the packet to the next device in the path. In case of link or node failure on the primary path, a secondary path was also provisioned to prevent a loss of service for the customer. The routers that are part of the secondary path are, in order from ingress to egress, PE1, P1, P3, P6, P5, P8, P9, P10, PE2. This path is a much longer path as backup links are generally allocated as a matter of availability, rather than efficiency.

The primary and secondary paths allowed for the recovery and forwarding of any incoming packets at the head end router of PE-1. Any packets that are already transiting through the primary LSP would be lost with only this level of redundancy. To mitigate this further loss, the additional functionality of fast-reroute or facility backup, was also configured.
PE-1 and PE-2 were both configured as BGP neighbors exchanging IPv4 and VPN-IPv4 NLRIs. PE-1 and PE-2, was each configured with a Virtual Routing and Forwarding (VRF) instance, that is used to communicate with the customer router using IPv4. While customer data packets traversed the service provider network via MPLS, customer routes traversed the service provider network via BGP.

The MPLS path on the service provider was also rate-limited to accepting only 10 Mb of traffic, and any traffic above this was immediately dropped.

The customer CE-1 and CE-2 routers were configured to communicate with PE-1 and PE-2 respectively. This communication was through BGP, and entailed the exchange of single IPv4 NLRIs from each site. CE-1 was also tasked with the additional duty of shaping traffic from the traffic generators prior to reaching the PE-1 interface.

To simulate a mix of traffic and collect data about their behavior, 6 types of traffic were generated. The 6 types that were generated are

1. Data
2. Data probes
3. Video
4. Video probes
5. Voice
6. Voice probes

As I was interested in the behavior of the packets over the service provider network, rather than the application itself, it was more important to have packets that had matching protocol headers, packet sizes and payload characteristics. All test data was being collected in the
single direction of flow, from Site 1 to Site 2. With these factors in mind, rather than running
live applications, sample packet captures were made and then replayed over the network.

Data packets were generated by making a standard ftp connection between the Windows
sender on Site 1, and the Debian receiver at Site 2. Seven 1000mb noise filled files were
generated utilizing the following command.

```
dd if=/dev/urandom of=test.file bs=1M count=1000
```

The files were placed on the Windows sender, and then uploaded via ftp to the Debian
listener. Wireshark was used to capture this data stream from the Windows sender, and the
packet capture was saved for reuse. The data stream generated in this way had an average
packet size of 930 bytes.

Voice packets were challenging to setup due to the lack of a working VoIP infrastructure.
Initially I had planned to get permission to tap voice traffic from a working organizations
network, but this was a nonstarter due to the security and privacy challenges involved. After
unsuccessfully trying to build a functional VoIP infrastructure using Asterisk, I decided to
make use of sample pcap files from Wireshark’s public repository for replaying voice
through the test environment. The average packet size of the voice stream was 200 bytes.

*Figure 12* shows the pcap file.
Video packets were generated by using VLC on the Windows sender to stream video to the Debian listener. This traffic was sent through the environment for 7 minutes. These packets were captured utilizing Wireshark and saved for subsequent use. The average packet size for the video stream was 741 bytes.

While the Data, Voice and Video packets that were captured provided a mix of traffic to send through the environment, they did not provide a means to calculate behavior over the network. To facilitate this, I made use of JunOS’s Realtime Performance Monitoring (RPM) functionality. RPM allows you to specify a server, and behaves as a client sending probes through the connected infrastructure. It additionally provides the means to mark the probe traffic with specific QoS markings, which the downstream Provider Edge would use to place into the matching service class. The markings used were...
- Voice probes: ef or 101110 (Decimal: 46)
- Video probes: af11 or 001010 (Decimal: 10)
- Data probes: be or 000000 (Decimal: 0)

Once the traffic was captured, on the Windows sender these captured pcap files were played in burst mode. Burst mode was necessary as the packet captures by themselves did not meet the thresholds of 8 Mbps and 12 Mbps that were required for the uncongested and congested scenarios, explained later. A free packet player application called Colasoft Packet Player (Colasoft, n.d.) was used to replay the packets, as it had the ability to burst the packets. In burst mode, the packet player application, ignores the timestamps within the pcap file, and attempts to push all captured packets onto the wire as soon as possible. The player was also set to loop the traffic, thereby generating a continuous burst of traffic. Multiple instances of the Colasoft packet player were run simultaneously for each type of traffic (3 instances replaying Voice, 3 instances replaying Video, 3 instances replaying Data) to generate an appropriate amount of traffic. This process was run simultaneously for the three classes of traffic – Voice, Video & Data, and now the network had a burst of continuous mixed traffic.

As the service provider LSP was limited to 10 Mbps, it was necessary to find a way to shape the traffic that was being sent from Site 1. This was done by configuring two traffic shapers on the CE-1 router. One setting was to rate limit traffic to 8 Mbps, and the second setting was to rate limit traffic to 12 Mbps. The first setting, being less than the LSP limit, was used to simulate uncongested traffic flow. The second setting, being more than the LSP limit, was used to simulate congestion on the LSP.

The RPM probes were generated from CE-1 to CE-2 and hence were mixed into the traffic prior to hitting the LSP ingress, so as to get appropriate end to end statistics.
To ensure that the entire testbed system was performing as expected, all routers were configured with the settings for the most complex of the required testing scenarios. This is the scenario where the network is congested and makes use of Class-of-Service for traffic optimization. I have subsequently referred to these test conditions as “Scenario 4”.
Scenario 1 – No Quality of Service; Low Congestion

There are two major factors to this scenario. Firstly, no Quality of Service configuration is enabled in the testbed. Secondly, the traffic being transferred between customer sites does not exceed the limits of the provisioned system, thereby having very low traffic congestion.

The following commands are run on the respective routers to facilitate this environment

CE-1

- set interface ge-0/0/1.0 family inet filter input normal-voice
- set interface ge-0/0/3.0 family inet filter input normal-davi
- deactivate class-of-service interfaces ge-* unit * rewrite-rules inet-precedence traffic_rewrite_rule

The 1\textsuperscript{st} and 2\textsuperscript{nd} command is used to enable a traffic shaper that will only send 8 Mb of traffic, which is less than the 10 Mb limit that the service provider has provisioned. The 3\textsuperscript{rd}
command ensures that traffic coming from the traffic generator is not marked with any QoS attributes before entering the SP PE-1 device.

**PE-1**

```
deactivate interface ge-0/0/1.0 family inet filter
deactivate class-of-service
```

The 1st command is used to disable the QoS classification packet filter on PE-1’s customer facing interface. The 2nd command is used to disable all QoS related functionality on the device.

**P1 through P11 and PE-2**

```
deactivate class-of-service
```

This command is used to disable all QoS related functionality on the device.

Once the routers are configured with the necessary settings, traffic is generated on the Windows sender device, utilizing the methodology described earlier.
In the 2\textsuperscript{nd} scenario, QoS is still not enabled, but the traffic being transferred between customer sites now exceeds the limits of the provisioned system and therefore we will be in a congested state.

The following commands are run on the respective routers to facilitate this environment

**CE-1**

- `set interface ge-0/0/1.0 family inet filter input congested-voice`
- `set interface ge-0/0/3.0 family inet filter input congested-dav`
- `deactivate class-of-service interfaces ge-* unit * rewrite-rules inet-precedence traffic_rewrite_rule`

The 1\textsuperscript{st} and 2\textsuperscript{nd} command is used to change the traffic shaper so as to send 12mb of traffic, which is greater than the 10mb limit that the service provider has provisioned. The 3\textsuperscript{rd}
command ensures that traffic coming from the traffic generator is not marked with any QoS attributes before entering the SP PE-1 device.

**PE-1**

| deactivate interface ge-0/0/1.0 family inet filter  
| deactivate class-of-service |

The 1\textsuperscript{st} command is used to disable the QoS classification packet filter on PE-1’s customer facing interface. The 2\textsuperscript{nd} command is used to disable all QoS related functionality on the device.

**P1 through P11 and PE-2**

| deactivate class-of-service |

This command is used to disable all QoS related functionality on the device.

Once the routers are configured with the necessary settings, traffic is generated on the Windows sender device.
**Scenario 3 – Quality of Service; Low Congestion**

In the 3rd scenario, QoS is now enabled, and traffic has been reduced to levels under the limit of the system. This leads to a situation similar to the 1st scenario, but with the addition of QoS.

The following commands are run on the respective routers to facilitate this environment

**CE-1**

```plaintext
set interface ge-0/0/1.0 family inet filter input normal-voice
set interface ge-0/0/3.0 family inet filter input normal-davi
activate class-of-service interfaces ge-* unit * rewrite-rules inet-precedence traffic_rewrite_rule
```

The 1st and 2nd command is used to set a traffic shaper that will only send 8 Mb of traffic, which is less than the 10mb limit that the service provider has provisioned. The 3rd command
ensures that traffic coming from the traffic generator is marked with QoS values before entering the SP PE-1 device.

**PE-1**

```plaintext
activate interface ge-0/0/1.0 family inet filter
activate class-of-service
```

The 1\textsuperscript{st} command is used to enable the QoS classification packet filter on PE-1’s customer facing interface. The 2\textsuperscript{nd} command is used to enable all QoS related functionality on the device.

**P1 through P11 and PE-2**

```plaintext
activate class-of-service
```

This command is used to enable all QoS related functionality on the device.

Once the routers are configured with the necessary settings, traffic is generated on the Windows sender device.
The 4th and final test is performed with QoS enabled and congestion in the system. This can be contrasted with the 2nd scenario where we have congestion, but no QoS.

The following commands are run on the respective routers to facilitate this environment

**CE-1**

```plaintext
set interface ge-0/0/1.0 family inet filter input congested-voice
set interface ge-0/0/3.0 family inet filter input congested-davi
activate class-of-service interfaces ge-* unit * rewrite-rules inet-precedence traffic_rewrite_rule
```

The 1st and 2nd command is used to set a traffic shaper that will send 12 Mb of traffic, which is greater than the 10 Mb limit that the service provider has provisioned. The 3rd command ensures that traffic coming from the traffic generator is marked with configured QoS values before entering the SP PE-1 device.
PE-1

activate interface ge-0/0/1.0 family inet filter
activate class-of-service

The 1st command is used to enable the QoS classification packet filter on PE-1’s customer facing interface. The 2nd command is used to disable all QoS related functionality on the device.

P1 through P11 and PE-2

activate class-of-service

This command is used to enable all QoS related functionality on the device.

Once the routers are configured with the necessary settings, traffic is generated on the Windows sender device.
**Checks**

As each scenario required a change in configuration on individual routers in the network, and as this was a manual process, a paper checklist (*Figure 14*) was created for each scenario. This served as a checklist to the changes and worked as an audit list to compare the configuration between test scenarios.

*Figure 17 - Manual Config Change Checklist*
**Statistical Data Capture**

For each scenario, once the traffic was flowing, the RPM probes were initialized as well, and the system was allowed to run traffic for approximately 300 seconds. *Figure 15* shows operational output from the PE-1 device confirming this.

```
amar@PE-1# run show mpls lsp extensive | no-more
Ingress LSP: 1 sessions

10.10.255.200
  **ActivePath: Best_Path (primary)**
  LSPtype: Static Configured
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4

  *Primary*    **Best_Path**     State: Up
  Priorities: 7 0
  Bandwidth: 10Mbps
  SmartOptimizeTimer: 180
  Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
    10.10.10.6 10.10.10.34 10.10.10.90
```

*Figure 18 – LSP on primary path*

Halfway through the test, a link failure was introduced on P7, causing the primary path to fail and the secondary path to get selected, as shown in *figure 16*.

```
amar@PE-1# run show mpls lsp extensive | no-more
Ingress LSP: 1 sessions

10.10.255.200
  **ActivePath: Failover_Path (secondary)**
  LSPtype: Static Configured
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
  Primary    **Best_Path**     State: Dn
  Priorities: 7 0
  Bandwidth: 10Mbps
  SmartOptimizeTimer: 180
```

*Figure 19 - LSP on secondary path*
Traffic was automatically rerouted over the backup path at this point. Throughout the test, RPM results were logged on CE-1. At the end of the test, this information was saved to a file and extracted from the system. The command used to get and save the results is given below:

```
show services rpm history-results detail | no-more | save results.txt
```

Once the raw output files are extracted, the relevant RTT & jitter values need to be extracted from the file. I wrote some python code to extract the interesting information and place it in a csv file. This file was then imported into MS Excel and used for further analysis.
Roadblocks

With any complex setup, issues were expected, and this testbed was no exception. There were multiple challenges, and while some of the challenges were straightforward in fixing, some required extensive research, and multiple rebuilds of the entire environment.

The first challenge was with regards to the generation of voice traffic. While my attempts at building a VoIP environment were unsuccessful, I eventually settled on using sample packet captures available on Wireshark’s public repository. Unfortunately, these packets did not work in the testbed when played, and while the packets did leave the Windows sender, they never arrived on CE-1. Close inspection of the packet capture, showed that the problem was due to the original Layer 2 Ethernet Media Access Control (MAC) address on the sample. As the sample was captured from an external environment, the destination MAC on the packets did not match that of the incoming interface on CE-1. Also, the sample packet captures had multiple packets, with multiple destination MAC addresses, so a simple solution of changing the MAC address of the incoming CE interface was not optimal. The best solution would be to rewrite the destination MAC address. This was done by using an open source utility by the name of Tcpreplay. Tcpreplay is an open source Linux utility that can be used for packet replay and packet modification (AppNeta). Once the utility is installed, the following command was used to modify the packet capture.

```
tcpreplay -i=SIP_CALL_RTP_G711.pcap -o=voice.pcap --enet-dmac=00:0c:29:f4:f5:f6
```

A second challenge, was in classifying voice traffic into its own queue. In the majority of cases, voice traffic is usually confined to its own VLAN, and any QoS capable device can simply classify all traffic in the ‘Voice’ VLAN as part of the voice queue. VLANs were not provisioned for the testbed, and would have required significant changes in the environment.
To work around this issue, a new VMnet link was provisioned between the Windows Sender and the CE-1 device. Colasoft Packet Player also helped with this, as it had the ability to select the interface on the system from where traffic needed to be sent, and so all voice traffic was sent through the newly provisioned NIC within the Windows sender (connected to the new link), while data and video remained together on the existing NIC (existing link). All traffic on this new interface was allocated to the voice queue.

The third challenge was one that took a lot of time, research and testing to overcome. The problem stemmed from the link failure that was being simulated halfway into the test. At this point, traffic was supposed to see minimal packet loss, and simply switch from the primary LSP to the secondary LSP. Unfortunately, in all my testing there were 27 packets being dropped no matter what. The congestion and QoS changes from the different scenarios did not change this behavior. Multiple options from researching on this behavior were tried, and no information was forthcoming utilizing Google and its vast index of information. I eventually moved the entire topology to a dedicated physical server that I was able to get some time on. The problem persisted even in this setup, but did help to rule out that the problem was related to the hardware that I had built my testbed on.

Having worked for a company that had an extensive lab environment of multiple Juniper devices, I was able to get some time on their physical Juniper routers. The MX80 routers that I had used are 80G rated Service Provider routers. By migrating the configuration to these devices, I hoped to find that the issue was related to buggy virtualization of JunOS and the vSRX. Unfortunately, the same issue persisted, and once again I saw the loss of 27 packets. Considering that these were service provider routers, I knew that any setup with a 27-packet loss would be unacceptable and tried to research some more on this issue.
At this time, since I still had access to physical routers, I decided to rebuild the entire topology with new addressing on all links. As this was meant to be a troubleshooting build, I did not plan to configure the topology with the extensive options that were used in my normal testbed. I decided to skip ISO addressing, and make use of OSPF as the IGP. On generation of traffic, I once again had packet drops occurring when the intermediate link on the primary path was forced to fail. This time, the number of packets that were lost was 40. It was at this point that a connection was made between IS-IS and OSPF. The dead timers for ISIS and OSPF are 27 seconds and 40 seconds respectively. This number directly corresponded with the number of packets that were lost, and it immediately struck me that the dead timers on the IGP could be the reason for traffic not switching to the secondary LSP. In order to force faster IGP convergence, I enabled bidirectional forwarding detection (BFD) on the OSPF protocol. This time there was almost no packet loss. On rerunning the tests, it became clear that either there was no packet loss, or a single packet was lost with BFD enabled. I immediately switched to using IS-IS and confirmed that the same behavior was occurring with IS-IS. On switching to the completely virtual primary testbed, and enabling BFD, this same behavior was confirmed.

On further investigation, I understood that the issue stemmed from the fact that when the primary path link failed, the secondary path was immediately activated thanks to the RSVP RESV Tear message. While this changed the MPLS path, the primary routing table on PE-1 still waited on the IGP to converge before updating its routing table with the new next hop via the secondary LSP.
Results & Analysis

The simulation results were tabulated via excel and a comparison was generated between the various traffic types, under the different conditions. These results were used to generate graphs to provide for a visual means of identifying the characteristics of traffic flow, where QoS or Congestion conditions are modified.

**Scenario 1 – Voice**

The graph shows the round-trip time (RTT) per packet, and the average RTT per packet for Voice traffic, without the benefit of QoS in an uncongested infrastructure. The rolling average of RTT, has been calculated separately for the primary LSP and for the secondary LSP. The graph clearly depicts, that post the link failure, and subsequent to the rerouting of traffic onto the secondary LSP, the average RTT has increased by a significant amount. This can be attributed to the longer path that the secondary LSP has been established over.

![Figure 20- Scenario 1 Voice](image-url)
Scenario 1 – Video

The graph shows the round-trip time (RTT) per packet, and the average RTT per packet for Video traffic, without the benefit of QoS in an uncongested infrastructure. The rolling average of RTT, has been calculated separately for the primary LSP and for the secondary LSP. The graph once again shows, that subsequent to the link failure and rerouting of traffic onto the secondary LSP, the average RTT has increased. As the underlying infrastructure remains the same, this too can be attributed to the longer path that the secondary LSP must traverse.
Scenario 1 - Data

The graph shows the round-trip time (RTT) per packet, and the average RTT per packet for Data traffic, without the benefit of QoS in an uncongested infrastructure. The rolling average of RTT, has been calculated separately for the primary LSP and for the secondary LSP. Here again we see the same behavior as with Voice & Video traffic. A higher average RTT due to the longer path for the secondary LSP.
Scenario 1 – Observation

The network is showing a higher latency, post the failure of the primary path, which can be explained by the length of the secondary path. The secondary path traverses over 8 links as compared to the primary which traverses 3, leading to a backup path that is approximately 3 times longer.

The link failure is being discovered and then rerouted without any loss of traffic, via the configured secondary paths, and facility backup bypass lslps. All three types of traffic are showing similar average RTT curves. This is to be expected as all traffic is being treated the same by the routers. These results match those of Maini, and show how traffic transfer time increases with the size of the traversed network (Maini, 2002).
Scenario 2 – Voice

The graph shows the round-trip time (RTT) per packet, and the average RTT per packet for Voice traffic, without the benefit of QoS in a congested infrastructure. The rolling average of RTT, has been calculated separately for the primary LSP and for the secondary LSP. The graph clearly depicts, that post the link failure, and subsequent to the rerouting of traffic onto the secondary LSP, the average RTT has increased by a very significant amount. This can be attributed to the longer path that the secondary LSP has been established over. Packet drops are observable throughout the timeline, with a significant loss at the point of failure. The significant loss shows a subsequent drop in the average RTT at the point of failure.
**Scenario 2 - Video**

The graph shows the round-trip time (RTT) per packet, and the average RTT per packet for Video traffic, without the benefit of QoS in a congested infrastructure. The rolling average of RTT, has been calculated separately for the primary LSP and for the secondary LSP. The graph clearly depicts, that post the link failure, and subsequent to the rerouting of traffic onto the secondary LSP, the average RTT has increased by a very significant amount. This can be attributed to the longer path that the secondary LSP has been established over. Packet drops are observable throughout the timeline. The drops are not significant enough to affect the average RTT at the point of failure, although a cluster of packet losses are still observable at this point.

*Figure 25 – Scenario 2 Video*
Scenario 2 - Data

The graph shows the round-trip time (RTT) per packet, and the average RTT per packet for Data traffic, without the benefit of QoS in a congested infrastructure. The rolling average of RTT, has been calculated separately for the primary LSP and for the secondary LSP. The graph clearly depicts, that post the link failure, and subsequent to the rerouting of traffic onto the secondary LSP, the average RTT has increased by a very significant amount. This can be attributed to the longer path that the secondary LSP has been established over. Packet drops are observable throughout the timeline, with a serious loss of traffic at the failure point. This drop of traffic is further reflected by the flat average RTT until the subsequent recovery of the data traffic.
Scenario 2 – Observation

The network is showing a higher latency, post the failure of the primary path, which can be explained by the length of the secondary path. All three types of traffic show similar average RTT curves 30 seconds after the failure. The higher penalty on RTT being observed on Voice traffic, in comparison to the other traffic types, is worthy of note.

The link failure is being discovered and then rerouted but loss of traffic is observable. Voice and Video are both seen to recover very quickly, but Data is lost for a longer period.
Scenario 3 – Voice

The graph shows the round-trip time (RTT) per packet, and the average RTT per packet for Voice traffic, with the benefit of QoS in an uncongested infrastructure. The rolling average of RTT, has been calculated separately for the primary LSP and for the secondary LSP. The graph clearly depicts, that post the link failure, and subsequent to the rerouting of traffic onto the secondary LSP, the average RTT has increased by a significant amount. This can be attributed to the longer path that the secondary LSP has been established over. There is a single packet lost at the time of failure, and another one within the next 45 seconds after which traffic is seen as stable.
Scenario 3 – Video

The graph shows the round-trip time (RTT) per packet, and the average RTT per packet for Video traffic, with the benefit of QoS in an uncongested infrastructure. The rolling average of RTT, has been calculated separately for the primary LSP and for the secondary LSP. The graph shows, that post the link failure and rerouting of traffic onto the secondary LSP, the average RTT has increased but not by a significant amount. The slight increase can be attributed to the longer path that the secondary LSP has been established over. The fact that average RTT is stable can be attributed to the QoS queue that was assigned to Video, and the ability of QoS to ensure that this particular queue is serviced at a higher priority than before. A few packet drops are observable through the timeline of the secondary path, but no serious loss of traffic anywhere, including the failure point.
**Scenario 3 - Data**

The graph shows the round-trip time (RTT) per packet, and the average RTT per packet for **Data** traffic, *with* the benefit of **QoS** in an *uncongested* infrastructure. The rolling average of RTT, has been calculated separately for the primary LSP and for the secondary LSP. The graph shows, that after the link failure point, the average RTT has increased but not by a significant amount. The slight increase can be attributed to the longer path of the secondary LSP. Packet drops are observable through the timeline of the secondary path, with a serious loss of traffic at the failure point. This drop of traffic is reflected in the flat average RTT until the recovery of the data traffic. This could be explained by the classification of data as the least priority best-effort queue, and thus the routers are dropping the least important traffic to ensure prioritization of the other traffic classes. The lack of congestion in this scenario helps the data traffic to recover but only after the servicing of the other queues.
Scenario 3 – Observation

The network is showing a higher latency, post the failure of the primary path, which can be explained by the length of the secondary path. This latency is most significant in the case of Voice traffic, and least significant in the case of Data. All three types of traffic show similar average RTT curves 60 seconds after the failure. Again, the penalty on Voice is the highest, for utilizing the backup path.

The link failure is being discovered and then rerouted but loss of traffic is observable. Voice and Video are both seen to recover very quickly, but Data is lost for a longer period.
Scenarios 4 - Voice

The graph shows the round-trip time (RTT) per packet, and the average RTT per packet for Voice traffic, with the benefit of QoS in a congested infrastructure. The rolling average of RTT, has been calculated separately for the primary LSP and for the secondary LSP. The graph clearly depicts, that post the link failure, and subsequent to the rerouting of traffic onto the secondary LSP, the average RTT has increased by a significant amount. This can be attributed to the longer path that the secondary LSP has been established over. Packet drops are observable through the timeline, with a higher number of drops while on the secondary path. The voice queue having the highest priority allows it to maintain and recover from the failure much better than the other types of traffic classes.
**Scenario 4 - Video**

The graph shows the round-trip time (RTT) per packet, and the average RTT per packet for Video traffic, with the benefit of QoS in a congested infrastructure. The rolling average of RTT, has been calculated separately for the primary LSP and for the secondary LSP. After the failure point, the average RTT can be seen to have increased by a significant amount. The increase can be attributed to the longer path of the secondary LSP and the lower QoS prioritization of Video. There is a serious loss of traffic at the failure point, also reflected by the flat average RTT. This could be explained by the classification of video as less important than the voice queue, and thus the routers are dropping video traffic to ensure the servicing of the voice queue. This is especially observed in the seconds after link failure, and can be explained by the requirement for existing traffic, on the broken link, and new traffic, direct from the CE, both having to contend for resources at the same time. The congestion in this scenario keeps all queues full, and thus exacerbates the loss of packets, across traffic classes.
Scenario 4 - Data

The graph shows the round-trip time (RTT) per packet, and the average RTT per packet for Data traffic, with the benefit of QoS in a congested infrastructure. The rolling average of RTT, has been calculated separately for the primary LSP and for the secondary LSP. The graph shows, that after the link failure point, the average RTT has increased by a significant amount. The increase can be attributed to the longer path of the secondary LSP and the low QoS prioritization of Data. Packet drops are observable through the timeline, with a greater number observed on the secondary path. The high number of drops could be explained by the classification of data as the least important traffic, and thus the routers are dropping data traffic to ensure the servicing of the voice and video queues. The congestion in this scenario keeps all queues full, and this results in significantly higher loss of packets, across traffic classes.
Scenario 4 – Observation

The network is showing a higher latency, post the failure of the primary path, which can be explained by the length of the secondary path. Voice traffic shows an initial spike in RTT that comes down, and is subsequently the lowest of all the traffic types on the backup path. All three types of traffic show similar average RTT curves 45 seconds after the failure.

The link failure is being discovered and then rerouted but loss of traffic is observable. Video traffic can be seen to drop for 30 seconds after the failure of the primary LSP.
Figure 36 shows the average latency of traffic across all the 4 scenarios. In the first and third scenario, the application of QoS under uncongested conditions, shows a small improvement in overall latency.

The second and fourth scenario, which are under congested conditions, shows an improvement of ten percent in average RTT when QoS profiles are used.
Figure 37 shows the effect of QoS and congestion on RTT and packet loss. The second scenario has the highest amount of packet drops, with no mechanism to select one traffic type over the other for drop. The 4th scenario is showing the least RTT, with a high number of packet drops, but this would be as per the service priorities that were defined. The graph shows how QoS ensures service levels by dropping less important traffic. The amount of traffic dropped increases as congestion increases, and classification of traffic helps to reduce both packet drops, and average RTT, as shown by the second and fourth pairings in the graphs.
Conclusion

Research into the current environment and technology that is being used, shows that MPLS and QoS aware networks are utilized throughout the industry, to enable oversubscription of services. Business efficiency requires that organizations do more with less. By prioritizing the relevant traffic, and dropping the lower priority traffic, it is possible to offer better service for specific classes, on a single infrastructure.

The simulation constructed using widely available software tools, consisted of 4 scenarios. The 4 scenarios explored the permutations of QoS and congestion being enabled or disabled. The simulation results show that under conditions of congestion, QoS has a significant impact on improving the performance of traffic, based on the metrics of round trip time. It also showed that in an uncongested environment, QoS did provide some benefit but lead to more lost packets. More varied traffic types, heavier traffic flows, and modified QoS profiles could all be tested in a like manner so as to further quantify the results of this simulation.
Future Work

One of the points that I would like to investigate in the future, is the effect of modifying the QoS profiles under congestion conditions. In my current research, all scenarios made use of the same QoS settings for the queue buffer sizes, and the queue transmission rates. By performing tests under various queue settings, it should be possible to further optimize the effects of QoS.

Live voice traffic was also not available in this testbed, and it would be very useful to validate the test results with actual phone calls, and to measure whether the improvement in transfer of the voice queue, translates into a practical gain for any VoIP users.

The current environment was built on VMWare’s commercial product. I would like to investigate this same environment built on Linux KVM. I would also like to automate the test and data collection process. This would require a non-trivial effort in building and coding an appropriate environment, and would be a challenge for those unfamiliar with Unix systems and coding.

As the testing revolves around network performance, I would also like to make use of Single Root Input Output Virtualization (SR-IOV), and study the ability of having hardware performance assistance.

While building, and coding an automated environment is what SDN orchestration is about, the scalability is also something to be investigated. On a small scale of automation, the use of Puppet or Chef could be used to build the setup locally. The environment could also be built on the cloud, possibly on Amazon Web Services (AWS), thereby allowing the entire test bed
to be abstracted onto generic hardware, allowing for a greater number of devices, and a shorter build time. This would help confirm the advantages of a working SDN deployment.
Appendix A – Router Configuration

**CE-1**

groups {
    eth-speed;
}
system {
    host-name CE-1;
    root-authentication {
        encrypted-password "$1$7f6kjzRZ$yaz5mU2vTuAeguzvVHR80"; ## SECRET-DATA
    }
services {
        ftp;
        ssh;
        telnet;
        web-management {
            http {
                interface ge-0/0/0.0;
            }
        }
    }
    syslog {
        user * {
            any emergency;
        }
        file messages {
            any any;
            authorization info;
        }
        file interactive-commands {
            interactive-commands any;
        }
    }
    license {
        autoupdate {
            url https://ae1.juniper.net/junos/key_retrieval;
        }
    }
    interfaces {
        ge-0/0/0 {
            unit 0 {
                family inet {
                    address 10.210.30.7/24;
                }
            }
        }
        ge-0/0/1 {
            unit 0 {
                family inet {
                    address 20.20.20.1/30;
                }
            }
        }
    }
}
ge-0/0/2 {
    mac 00:0c:29:f4:f5:f6;
    unit 0 {
        family inet {
            filter {
                input normal-voice;
            }
            inactive: policer {
                input normal-voice;
            }
            address 10.10.10.1/30;
        }
    }
}
ge-0/0/3 {
    enable;
    unit 0 {
        family inet {
            filter {
                input normal-davi;
            }
            address 10.10.20.1/30;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            filter {
                inactive: input abc;
            }
            address 20.20.255.1/32;
        }
    }
}
forwarding-options {
    packet-capture {
        file filename abc world-readable;
    }
}
routing-options {
    static {
        route 50.50.50.50/32 receive;
        route 0.0.0.0/0 next-hop 40.40.40.2;
    }
    router-id 20.20.255.1;
    autonomous-system 50;
}
protocols {
    bgp {
        group external {
            type external;
            export test-route;
            peer-as 100;
            neighbor 20.20.20.2;
        }
    }
}
policy-options {
    policy-statement test-route {
        term 1 {
            from protocol [ static direct ];
            then accept;
        }
    }
}

class-of-service {
    classifiers {
        inet-precedence traffic_classifier {
            forwarding-class DATA_BE {
                loss-priority low code-points 000;
            }
            forwarding-class VIDEO_AF {
                loss-priority low code-points 010;
            }
            forwarding-class VOICE_EF {
                loss-priority low code-points 101;
            }
            forwarding-class NETWORK_CONTROL {
                loss-priority low code-points 110;
            }
        }
    }
    forwarding-classes {
        queue 0 DATA_BE priority low;
        queue 1 VIDEO_AF priority low;
        queue 2 VOICE_EF priority high;
        queue 3 NETWORK_CONTROL priority high;
    }
}

interfaces {
    ge-* {
        scheduler-map traffic-scheduler-map;
        unit * {
            classifiers {
                inet-precedence traffic_classifier;
            }
            rewrite-rules {
                inet-precedence traffic_rewrite_rule;
            }
        }
    }
}

rewrite-rules {
    inet-precedence traffic_rewrite_rule {
        forwarding-class DATA_BE {
            loss-priority low code-point 000;
        }
        forwarding-class VIDEO_AF {
            loss-priority low code-point 010;
        }
        forwarding-class VOICE_EF {
            loss-priority low code-point 101;
        }
        forwarding-class NETWORK_CONTROL {
            loss-priority low code-point 110;
        }
    }
}
scheduler-maps {
    traffic-scheduler-map {
        forwarding-class DATA_BE scheduler data;
        forwarding-class VOICE_EF scheduler voice;
        forwarding-class VIDEO_AF scheduler video;
        forwarding-class NETWORK_CONTROL scheduler network;
    }
}
schedulers {
    voice {
        transmit-rate percent 15;
        buffer-size percent 0;
        priority high;
    }
    video {
        transmit-rate percent 60;
        buffer-size percent 60;
        priority low;
    }
    network {
        transmit-rate percent 10;
        buffer-size percent 10;
        priority high;
    }
    data {
        transmit-rate {
            remainder;
        }
        buffer-size {
            remainder;
        }
        priority low;
    }
}
security {
    forwarding-options {
        family {
            mpls {
                mode packet-based;
            }
        }
    }
}
firewall {
    family inet {
        filter normal-davi {
            term 1 {
                from {
                    port 5004;
                }
                then {
                    policer normal-video;
                    count video-in-profile;
                    forwarding-class VIDEO_AF;
                    accept;
                }
            }
        }
    }
}
term 2 {
    then {
        policer normal-data;
        count data-in-profile;
        forwarding-class DATA_BE;
        accept;
    }
}
filter normal-voice {
    term 1 {
        then {
            policer normal-voice;
            count voice-in-profile;
            forwarding-class VOICE_EF;
            accept;
        }
    }
}
filter congested-davi {
    term 1 {
        from {
            port 5004;
        }
        then {
            policer congested-video;
            count video-in-profile;
            forwarding-class VIDEO_AF;
            accept;
        }
    }
    term 2 {
        then {
            policer congested-data;
            count data-in-profile;
            forwarding-class DATA_BE;
            accept;
        }
    }
}
filter congested-voice {
    term 1 {
        then {
            policer congested-voice;
            count voice-in-profile;
            forwarding-class VOICE_EF;
            accept;
        }
    }
}
policer normal-video {
    if-exceeding {
        bandwidth-limit 4m;
        burst-size-limit 15k;
    }
    then discard;
}
policer congested-video {
if-exceeding {
    bandwidth-limit 6m;
    burst-size-limit 15k;
} then discard;
}
policer normal-voice {
    if-exceeding {
        bandwidth-limit 2m;
        burst-size-limit 15k;
    } then discard;
}
policer congested-voice {
    if-exceeding {
        bandwidth-limit 3m;
        burst-size-limit 15k;
    } then discard;
}
policer normal-data {
    if-exceeding {
        bandwidth-limit 2m;
        burst-size-limit 15k;
    } then discard;
}
policer congested-data {
    if-exceeding {
        bandwidth-limit 3m;
        burst-size-limit 15k;
    } then discard;
}
filter abc {
    term 1 {
        then {
            sample;
            accept;
        }
    }
}
services {
    rpm {
        probe Amar {
            inactive: test video_test {
                probe-type icmp-ping-timestamp;
                target address 20.20.20.6;
                probe-count 15;
                probe-interval 1;
                test-interval 1;
                source-address 20.20.20.1;
                source-address 20.20.20.1;
                history-size 600;
                dscp-code-points af11;
                data-size 1370;
            }
            test voice_test {
                probe-type icmp-ping-timestamp;
            }
        }
    }
}
target address 20.20.20.6;
probe-count 15;
probe-interval 1;
test-interval 1;
source-address 20.20.20.1;
history-size 600;
dscp-code-points ef;
data-size 214;
}
inactive: test data_test {
    probe-type icmp-ping-timestamp;
target address 20.20.20.6;
probe-count 15;
probe-interval 1;
test-interval 1;
source-address 20.20.20.1;
history-size 600;
data-size 1436;
hardware-timestamp;
}
}
}

PE-1

system {
    host-name PE-1;
    root-authentication {
        encrypted-password "$1$HbwHNy.o$so8YQRduNh3HniuhX01WDo."; ## SECRET- DATA
    }
    services {
        ssh;
        web-management {
            http {
                interface ge-0/0/0.0;
            }
        }
    }
    syslog {
        user * {
            any emergency;
        }
        file messages {
            any any;
            authorization info;
        }
        file interactive-commands {
            interactive-commands any;
        }
    }
    license {
        autoupdate {
            url https://ae1.juniper.net/junos/key_retrieval;
        }
    }
}

interfaces {
MPLS & QoS in Virtual Environments

ge-0/0/0 {
    unit 0 {
        family inet {
            address 10.210.30.2/24;
        }
    }
}
ge-0/0/1 {
    speed 100m;
    link-mode full-duplex;
    gigether-options {
        no-auto-negotiation;
    }
    unit 0 {
        family inet {
            filter {
                input rpm-classifier;
            }
            address 20.20.20.30/24;
        }
    }
}
ge-0/0/2 {
    speed 100m;
    link-mode full-duplex;
    gigether-options {
        no-auto-negotiation;
    }
    unit 0 {
        family inet {
            address 10.10.1/30;
            family iso;
            family mpls;
        }
    }
}
ge-0/0/3 {
    speed 100m;
    link-mode full-duplex;
    gigether-options {
        no-auto-negotiation;
    }
    unit 0 {
        family inet {
            address 10.10.21/30;
            family iso;
            family mpls;
        }
    }
}
ge-0/0/4 {
    speed 100m;
    link-mode full-duplex;
    gigether-options {
        no-auto-negotiation;
    }
    unit 0 {
        family inet {
            address 10.10.5/30;
        }
    }
}
family iso;
family mpls;
}
lo0 {
  unit 0 {
    family inet {
      address 10.10.255.100/32;
    }
    family iso {
      address 49.2222.0100.1025.5100.00;
    }
    family mpls;
  }
}
}

routing-options {
  router-id 10.210.255.100;
  autonomous-system 100;
  forwarding-table {
    export load-bal;
  }
}

protocols {
  rsvp {
    interface all;
  }
  mpls {
    auto-policing {
      class all drop;
    }
    icmp-tunneling;
    no-cspf;
    oam {
      bfd-liveness-detection {
        minimum-interval 300;
        multiplier 2;
        failure-action teardown;
      }
    }
    label-switched-path PE-1-to-PE-2 {
      to 10.10.255.200;
      bandwidth 10m;
      fast-reroute {
        hop-limit 15;
      }
      primary Best_Path;
      secondary Failover_Path;
    }
    path Best_Path {
      10.10.10.6 strict;
      10.10.10.34 strict;
      10.10.10.90 strict;
    }
    path Failover_Path {
      10.10.10.2 strict;
      10.10.10.14 strict;
      10.10.10.18 strict;
    }
  }
}
10.10.10.50 strict;
10.10.10.54 strict;
10.10.10.62 strict;
10.10.10.78 strict;
10.10.10.82 strict;
}
} interface all;
}
} bgp {
    group internal {
        type internal;
        local-address 10.10.255.100;
        family inet {
            unicast;
        }
        family inet-vpn {
            unicast;
        }
        export nhs;
        neighbor 10.10.255.200;
    }
}
} isis {
    interface ge-0/0/2.0 {
        point-to-point;
        level 1 disable;
    }
    interface ge-0/0/3.0 {
        point-to-point;
        level 1 disable;
    }
    interface ge-0/0/4.0 {
        point-to-point;
        level 1 disable;
    }
    interface lo0.0 {
        level 1 disable;
    }
}
} policy-options {
    policy-statement load-bal {
        then {
            load-balance per-packet;
        }
    }
    policy-statement nhs {
        then {
            next-hop self;
        }
    }
    policy-statement vrf-export-policy {
        term 1 {
            from protocol [ bgp direct ];
            then {
                community add customer-a;
                accept;
            }
        }
    }
}
term 2 {
    then reject;
}

policy-statement vrf-import-policy {
    term 1 {
        from {
            protocol bgp;
            community customer-a;
        }
        then accept;
    }
    term 2 {
        then reject;
    }
    community customer-a members target:100:200;
}

class-of-service {
    classifiers {
        exp traffic_classifier_p_devices {
            forwarding-class DATA_BE {
                loss-priority low code-points 000;
            }
            forwarding-class VIDEO_AF {
                loss-priority low code-points 010;
            }
            forwarding-class NETWORK_CONTROL {
                loss-priority low code-points 110;
            }
            forwarding-class VOICE_EF {
                loss-priority low code-points 101;
            }
        }
        inet-precedence traffic_classifier_pe_devices {
            forwarding-class DATA_BE {
                loss-priority low code-points 000;
            }
            forwarding-class VIDEO_AF {
                loss-priority low code-points 010;
            }
            forwarding-class VOICE_EF {
                loss-priority low code-points 101;
            }
            forwarding-class NETWORK_CONTROL {
                loss-priority low code-points 110;
            }
        }
    }
    forwarding-classes {
        queue 0 DATA_BE priority low;
        queue 1 VIDEO_AF priority low;
        queue 2 VOICE_EF priority high;
        queue 3 NETWORK_CONTROL priority high;
        queue 4 test priority high;
    }
    interfaces {
        ge-* {
            scheduler-map traffic-scheduler-map;
        }
    }
}
unit * {
    classifiers {
        exp traffic_classifier_p_devices;
    }
    rewrite-rules {
        exp traffic_rewrite_rule;
    }
}
}
ge-0/0/1 {
    scheduler-map traffic-scheduler-map;
    unit 0 {
        classifiers {
            inet-precedence traffic_classifier_pe_devices;
        }
        rewrite-rules {
            exp traffic_rewrite_rule;
        }
    }
}
rewrite-rules {
    exp traffic_rewrite_rule {
        forwarding-class DATA_BE {
            loss-priority low code-point 000;
        }
        forwarding-class VIDEO_AF {
            loss-priority low code-point 010;
        }
        forwarding-class NETWORK_CONTROL {
            loss-priority low code-point 110;
        }
        forwarding-class VOICE_EF {
            loss-priority low code-point 101;
        }
    }
}
scheduler-maps {
    traffic-scheduler-map {
        forwarding-class DATA_BE scheduler data;
        forwarding-class VOICE_EF scheduler voice;
        forwarding-class VIDEO_AF scheduler video;
        forwarding-class NETWORK_CONTROL scheduler network;
    }
}
schedulers {
    voice {
        transmit-rate percent 15;
        buffer-size percent 0;
        priority high;
    }
    video {
        transmit-rate percent 40;
        buffer-size percent 40;
        priority low;
    }
    network {
        transmit-rate percent 10;
        buffer-size percent 10;
    }
priority high;
}
data {
    transmit-rate {
        remainder;
    }
    buffer-size {
        remainder;
    }
    priority low;
}
}
}
security {
    forwarding-options {
        family {
            mpls {
                mode packet-based;
            }
        }
    }
}
}
firewall {
    family inet {
        filter rpm-classifier {
            term rpm-video-classifier {
                from {
                    dscp af11;
                }
                then {
                    count rpm-video-classifier-counter;
                    forwarding-class VIDEO_AF;
                    accept;
                }
            }
            term video-traffic-classifier {
                from {
                    destination-port 5004;
                }
                then {
                    count video-traffic-classifier-counter;
                    forwarding-class VIDEO_AF;
                    accept;
                }
            }
            term rpm-voice-classifier {
                from {
                    dscp ef;
                }
                then {
                    count rpm-voice-classifier-counter;
                    forwarding-class VOICE_EF;
                    accept;
                }
            }
            term data-rpm-and-traffic-classifier {
                then {
                    count data-rpm-and-traffic-counter;
                    forwarding-class DATA_BE;
routing-instances {
  CE-A {
    instance-type vrf;
    interface ge-0/0/1.0;
    route-distinguisher 10.10.255.100:1;
    vrf-import vrf-import-policy;
    vrf-export vrf-export-policy;
    vrf-table-label;
    protocols {
      bgp {
        group external {
          type external;
          peer-as 50;
          as-overide;
          neighbor 20.20.20.1;
        }
      }
    }
  }
}

P-1

system {
  host-name P-1;
  root-authentication {
    encrypted-password "$1$HbwHNy.o$o8YQRduNh3HniuhX01WDo."; ## SECRET-DATA
  }
  services {
    ssh;
    web-management {
      http {
        interface ge-0/0/0.0;
      }
    }
  }
  syslog {
    user * {
      any emergency;
    }
    file messages {
      any any;
      authorization info;
    }
    file interactive-commands {
      interactive-commands any;
    }
  }
  license {
    autoupdate {
      url https://ae1.juniper.net/junos/key_retrieval;
    }
  }
}
interfaces {
  ge-0/0/0 {
    unit 0 {
      family inet {
        address 10.210.30.10/24;
      }
    }
  }
  ge-0/0/1 {
    speed 100m;
    link-mode full-duplex;
    gigether-options {
      no-auto-negotiation;
    }
    unit 0 {
      family inet {
        address 10.10.10.2/30;
      }
      family iso;
      family mpls;
    }
  }
  ge-0/0/2 {
    speed 100m;
    link-mode full-duplex;
    gigether-options {
      no-auto-negotiation;
    }
    unit 0 {
      family inet {
        address 10.10.10.9/30;
      }
      family iso;
      family mpls;
    }
  }
  ge-0/0/3 {
    speed 100m;
    link-mode full-duplex;
    gigether-options {
      no-auto-negotiation;
    }
    unit 0 {
      family inet {
        address 10.10.10.13/30;
      }
      family iso;
      family mpls;
    }
  }
  ge-0/0/4 {
    speed 100m;
    link-mode full-duplex;
    gigether-options {
      no-auto-negotiation;
    }
  }
}
lo0 {
    unit 0 {
        family inet {
            address 10.10.255.1/32;
        }
        family iso {
            address 49.2222.0100.0125.5001.00;
        }
        family mpls;
    }
}
}

routing-options {
    router-id 10.210.255.1;
    autonomous-system 100;
}

protocols {
    rsvp {
        interface all;
    }
    mpls {
        icmp-tunneling;
        interface all;
    }
    isis {
        interface ge-0/0/1.0 {
            point-to-point;
            level 1 disable;
        }
        interface ge-0/0/2.0 {
            point-to-point;
            level 1 disable;
        }
        interface ge-0/0/3.0 {
            point-to-point;
            level 1 disable;
        }
        interface lo0.0 {
            level 1 disable;
        }
    }
}

class-of-service {
    classifiers {
        exp traffic classifier_p_devices {
            forwarding-class DATA_BE {
                loss-priority low code-points 000;
            }
            forwarding-class VIDEO_AF {
                loss-priority low code-points 010;
            }
            forwarding-class NETWORK_CONTROL {
                loss-priority low code-points 110;
            }
            forwarding-class VOICE_EF {
                loss-priority low code-points 101;
            }
        }
        inet-precedence traffic classifier_pe_devices {
forwarding-class DATA_BE {
    loss-priority low code-points 000;
}
forwarding-class VIDEO_AF {
    loss-priority low code-points 010;
}
forwarding-class VOICE_EF {
    loss-priority low code-points 101;
}
forwarding-class NETWORK_CONTROL {
    loss-priority low code-points 110;
}
}

forwarding-classes {
    queue 0 DATA_BE priority low;
    queue 1 VIDEO_AF priority low;
    queue 2 VOICE_EF priority high;
    queue 3 NETWORK_CONTROL priority high;
    queue 4 test priority high;
}

interfaces {
    ge-* {
        scheduler-map traffic-scheduler-map;
        unit * {
            classifiers {
                exp traffic_classifier_p_devices;
            }
            rewrite-rules {
                exp traffic_rewrite_rule;
            }
        }
    }
}

rewrite-rules {
    exp traffic_rewrite_rule {
        forwarding-class DATA_BE {
            loss-priority low code-point 000;
        }
        forwarding-class VIDEO_AF {
            loss-priority low code-point 010;
        }
        forwarding-class NETWORK_CONTROL {
            loss-priority low code-point 110;
        }
        forwarding-class VOICE_EF {
            loss-priority low code-point 101;
        }
    }
}

scheduler-maps {
    traffic-scheduler-map {
        forwarding-class DATA_BE scheduler data;
        forwarding-class VOICE_EF scheduler voice;
        forwarding-class VIDEO_AF scheduler video;
        forwarding-class NETWORK_CONTROL scheduler network;
    }
}

schedulers {
voice {
transmit-rate percent 15;
buffer-size percent 0;
priority high;
}
video {
transmit-rate percent 40;
buffer-size percent 40;
priority low;
}
network {
transmit-rate percent 10;
buffer-size percent 10;
priority high;
}
data {
transmit-rate {
remainder;
}
buffer-size {
remainder;
}
priority low;
}
}
security {
forwarding-options {
family {
mpls {
mode packet-based;
}
}
}
}
system {
host-name P-2;
root-authentication {
encrypted-password "$1$HbwHny.oSo8YQRduNh3HniuhX01WDo."; ## SECRET-DATA
}
services {
ssh;
web-management {
http {
interface ge-0/0/0.0;
}
}
}
syslog {
user * {
any emergency;
}
file messages {
any any;
authorization info;
}
file interactive-commands {
  interactive-commands any;
}

license {
  autoupdate {
    url https://ae1.juniper.net/junos/key_retrieval;
  }
}

interfaces {
  ge-0/0/0 {
    unit 0 {
      family inet {
        address 10.210.30.11/24;
      }
    }
  }
  ge-0/0/1 {
    speed 100m;
    link-mode full-duplex;
    gigether-options {
      no-auto-negotiation;
    }
    unit 0 {
      family inet {
        address 10.10.10.6/30;
      }
      family iso;
      family mpls;
    }
  }
  ge-0/0/2 {
    speed 100m;
    link-mode full-duplex;
    gigether-options {
      no-auto-negotiation;
    }
    unit 0 {
      family inet {
        address 10.10.10.10/30;
      }
      family iso;
      family mpls;
    }
  }
  ge-0/0/3 {
    speed 100m;
    link-mode full-duplex;
    gigether-options {
      no-auto-negotiation;
    }
    unit 0 {
      family inet {
        address 10.10.10.25/30;
      }
      family iso;
      family mpls;
    }
  }
}
MPLS & QoS in Virtual Environments

ge-0/0/4 {
    speed 100m;
    link-mode full-duplex;
    gigether-options {
        no-auto-negotiation;
    }
    unit 0 {
        family inet {
            address 10.10.10.33/30;
        }
        family iso;
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.10.255.2/32;
        }
        family iso {
            address 49.2222.0100.1025.5002.00;
        }
        family mpls;
    }
}
routing-options {
    router-id 10.210.255.2;
    autonomous-system 100;
}
protocols {
    rsvp {
        interface all;
    }
    mpls {
        icmp-tunneling;
        interface all;
    }
    isis {
        interface ge-0/0/1.0 {
            point-to-point;
            level 1 disable;
        }
        interface ge-0/0/2.0 {
            point-to-point;
            level 1 disable;
        }
        interface ge-0/0/3.0 {
            point-to-point;
            level 1 disable;
        }
        interface ge-0/0/4.0 {
            point-to-point;
            level 1 disable;
        }
        interface lo0.0 {
            level 1 disable;
        }
    }
}
class-of-service {
    classifiers {
        exp traffic_classifier_p_devices {
            forwarding-class DATA_BE {
                loss-priority low code-points 000;
            }
            forwarding-class VIDEO_AF {
                loss-priority low code-points 010;
            }
            forwarding-class NETWORK_CONTROL {
                loss-priority low code-points 110;
            }
            forwarding-class VOICE_EF {
                loss-priority low code-points 101;
            }
        }
        inet-precedence traffic_classifier_pe_devices {
            forwarding-class DATA_BE {
                loss-priority low code-points 000;
            }
            forwarding-class VIDEO_AF {
                loss-priority low code-points 010;
            }
            forwarding-class VOICE_EF {
                loss-priority low code-points 101;
            }
            forwarding-class NETWORK_CONTROL {
                loss-priority low code-points 110;
            }
        }
    }
    forwarding-classes {
        queue 0 DATA_BE priority low;
        queue 1 VIDEO_AF priority low;
        queue 2 VOICE_EF priority high;
        queue 3 NETWORK_CONTROL priority high;
        queue 4 test priority high;
    }
    interfaces {
        ge-* {
            scheduler-map traffic-scheduler-map;
            unit * {
                classifiers {
                    exp traffic_classifier_p_devices;
                }
                rewrite-rules {
                    exp traffic_rewrite_rule;
                }
            }
        }
    }
    rewrite-rules {
        exp traffic_rewrite_rule {
            forwarding-class DATA_BE {
                loss-priority low code-point 000;
            }
        }
    }
}
forwarding-class VIDEO_AF {
    loss-priority low code-point 010;
}
forwarding-class NETWORK_CONTROL {
    loss-priority low code-point 110;
}
forwarding-class VOICE_EF {
    loss-priority low code-point 101;
}
}

scheduler-maps {
    traffic-scheduler-map {
        forwarding-class DATA_BE scheduler data;
        forwarding-class VOICE_EF scheduler voice;
        forwarding-class VIDEO_AF scheduler video;
        forwarding-class NETWORK_CONTROL scheduler network;
    }
}
schedulers {
    voice {
        transmit-rate percent 15;
        buffer-size percent 0;
        priority high;
    }
    video {
        transmit-rate percent 40;
        buffer-size percent 40;
        priority low;
    }
    network {
        transmit-rate percent 10;
        buffer-size percent 10;
        priority high;
    }
    data {
        transmit-rate {
            remainder;
        }
        buffer-size {
            remainder;
        }
        priority low;
    }
}

security {
    forwarding-options {
        family {
            mpls {
                mode packet-based;
            }
        }
    }
}
system {
    host-name P-3;
    root-authentication {
        encrypted-password "$1$HbwHNY.o$0YQRduNh3HniuhX01WDo."; ## SECRET-DATA
    }
    services {
        ssh;
        web-management {
            http {
                interface ge-0/0/0.0;
            }
        }
    }
    syslog {
        user * {
            any emergency;
        }
        file messages {
            any any;
            authorization info;
        }
        file interactive-commands {
            interactive-commands any;
        }
    }
    license {
        autoupdate {
            url https://ae1.juniper.net/junos/key_retrieval;
        }
    }
    interfaces {
        ge-0/0/0 {
            unit 0 {
                family inet {
                    address 10.210.30.12/24;
                }
            }
        }
        ge-0/0/1 {
            speed 100m;
            link-mode full-duplex;
            gigether-options {
                no-auto-negotiation;
            }
            unit 0 {
                family inet {
                    address 10.10.10.14/30;
                }
                family iso;
                family mpls;
            }
        }
        ge-0/0/2 {
            speed 100m;
            link-mode full-duplex;
            gigether-options {
                no-auto-negotiation;
            }
            unit 0 {
                family inet {
                    address 10.210.30.13/24;
                }
                family iso;
                family mpls;
            }
        }
    }
}
no-auto-negotiation;
}
unit 0 {
  family inet {
    address 10.10.10.22/30;
  }
  family iso;
  family mpls;
}
}
ge-0/0/3 {
  speed 100m;
  link-mode full-duplex;
  gigether-options {
    no-auto-negotiation;
  }
  unit 0 {
    family inet {
      address 10.10.10.26/30;
    }
    family iso;
    family mpls;
  }
}
ge-0/0/4 {
  speed 100m;
  link-mode full-duplex;
  gigether-options {
    no-auto-negotiation;
  }
  unit 0 {
    family inet {
      address 10.10.10.17/30;
    }
    family iso;
    family mpls;
  }
}
ge-0/0/5 {
  speed 100m;
  link-mode full-duplex;
  gigether-options {
    no-auto-negotiation;
  }
  unit 0 {
    family inet {
      address 10.10.10.29/30;
    }
    family iso;
    family mpls;
  }
}
lo0 {
  unit 0 {
    family inet {
      address 10.10.255.3/32;
    }
    family iso {
      address 49.2222.0100.1025.5003.00;
family mpls;
}
}

routing-options {
    router-id 10.210.255.3;
    autonomous-system 100;
}

protocols {
    rsvp {
        interface all;
    }
    mpls {
        icmp-tunneling;
        interface all;
    }
    isis {
        interface ge-0/0/1.0 {
            point-to-point;
            level 1 disable;
        }
        interface ge-0/0/2.0 {
            point-to-point;
            level 1 disable;
        }
        interface ge-0/0/3.0 {
            point-to-point;
            level 1 disable;
        }
        interface ge-0/0/4.0 {
            point-to-point;
            level 1 disable;
        }
        interface ge-0/0/5.0 {
            point-to-point;
            level 1 disable;
        }
        interface lo0.0 {
            level 1 disable;
        }
    }
}

class-of-service {
    classifiers {
        exp traffic_classifier_p_devices {
            forwarding-class DATA_BE {
                loss-priority low code-points 000;
            }
            forwarding-class VIDEO_AF {
                loss-priority low code-points 010;
            }
            forwarding-class NETWORK_CONTROL {
                loss-priority low code-points 110;
            }
            forwarding-class VOICE_EF {
                loss-priority low code-points 101;
            }
        }
    }
}
inet-precedence traffic_classifier_pe_devices {
  forwarding-class DATA_BE {
    loss-priority low code-points 000;
  }
  forwarding-class VIDEO_AF {
    loss-priority low code-points 010;
  }
  forwarding-class VOICE_EF {
    loss-priority low code-points 101;
  }
  forwarding-class NETWORK_CONTROL {
    loss-priority low code-points 110;
  }
}

forwarding-classes {
  queue 0 DATA_BE priority low;
  queue 1 VIDEO_AF priority low;
  queue 2 VOICE_EF priority high;
  queue 3 NETWORK_CONTROL priority high;
  queue 4 test priority high;
}

interfaces {
  ge-* {
    scheduler-map traffic-scheduler-map;
    unit * {
      classifiers {
        exp traffic_classifier_p_devices;
      }
      rewrite-rules {
        exp traffic_rewrite_rule;
      }
    }
  }
}

rewrite-rules {
  exp traffic_rewrite_rule {
    forwarding-class DATA_BE {
      loss-priority low code-point 000;
    }
    forwarding-class VIDEO_AF {
      loss-priority low code-point 010;
    }
    forwarding-class NETWORK_CONTROL {
      loss-priority low code-point 110;
    }
    forwarding-class VOICE_EF {
      loss-priority low code-point 101;
    }
  }
}

scheduler-maps {
  traffic-scheduler-map {
    forwarding-class DATA_BE scheduler data;
    forwarding-class VOICE_EF scheduler voice;
    forwarding-class VIDEO_AF scheduler video;
    forwarding-class NETWORK_CONTROL scheduler network;
  }
}
schedulers {
    voice {
        transmit-rate percent 15;
        buffer-size percent 0;
        priority high;
    }
    video {
        transmit-rate percent 40;
        buffer-size percent 40;
        priority low;
    }
    network {
        transmit-rate percent 10;
        buffer-size percent 10;
        priority high;
    }
    data {
        transmit-rate {
            remainder;
        }
        buffer-size {
            remainder;
        }
        priority low;
    }
}
security {
    forwarding-options {
        family {
            mpls {
                mode packet-based;
            }
        }
    }
}

system {
    host-name P-4;
    root-authentication {
        encrypted-password "$1$HbwHNy.oSo8YQRduNh3HnihuX01WDo.";  ## SECRET-DATA
    }
    services {
        ssh;
        web-management {
            http {
                interface ge-0/0/0.0;
            }
        }
    }
    syslog {
        user * {
            any emergency;
        }
        file messages {
            any any;
        }
    }
}
authorization info;
}
file interactive-commands {
    interactive-commands any;
}
}
license {
    autoupdate {
        url https://ae1.juniper.net/junos/key_retrieval;
    }
}
}
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 10.210.30.13/24;
            }
        }
    }
    ge-0/0/1 {
        speed 100m;
        link-mode full-duplex;
        gigether-options {
            no-auto-negotiation;
        }
        unit 0 {
            family inet {
                address 10.10.10.30/30;
            }
            family iso;
            family mpls;
        }
    }
    ge-0/0/2 {
        speed 100m;
        link-mode full-duplex;
        gigether-options {
            no-auto-negotiation;
        }
        unit 0 {
            family inet {
                address 10.10.10.41/30;
            }
            family iso;
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.10.255.4/32;
            }
            family iso {
                address 49.2222.0100.1025.5004.00;
            }
            family mpls;
        }
    }
MPLS & QoS in Virtual Environments

```plaintext
routing-options {
    router-id 10.210.255.4;
    autonomous-system 100;
}
protocols {
    rsvp {
        interface all;
    }
    mpls {
        icmp-tunneling;
        interface all;
    }
    isis {
        interface ge-0/0/1.0 {
            point-to-point;
            level 1 disable;
        }
        interface ge-0/0/2.0 {
            point-to-point;
            level 1 disable;
        }
        interface lo0.0 {
            level 1 disable;
        }
    }
}
class-of-service {
    classifiers {
        exp traffic_classifier_p_devices {
            forwarding-class DATA_BE {
                loss-priority low code-points 000;
            }
            forwarding-class VIDEO.AF {
                loss-priority low code-points 010;
            }
            forwarding-class NETWORK_CONTROL {
                loss-priority low code-points 110;
            }
            forwarding-class VOICE_EF {
                loss-priority low code-points 101;
            }
        }
        inet-precedence traffic_classifier_pe_devices {
            forwarding-class DATA_BE {
                loss-priority low code-points 000;
            }
            forwarding-class VIDEO_AF {
                loss-priority low code-points 010;
            }
            forwarding-class VOICE_EF {
                loss-priority low code-points 101;
            }
            forwarding-class NETWORK_CONTROL {
                loss-priority low code-points 110;
            }
        }
    }
}
```
queue 0 DATA_BE priority low;
queue 1 VIDEO_AF priority low;
queue 2 VOICE_EF priority high;
queue 3 NETWORK_CONTROL priority high;
queue 4 test priority high;
}

interfaces {
  ge-* {
    scheduler-map traffic-scheduler-map;
    unit * {
      classifiers {
        exp traffic_classifier_p_devices;
      }
      rewrite-rules {
        exp traffic_rewrite_rule;
      }
    }
  }
}

rewrite-rules {
  exp traffic_rewrite_rule {
    forwarding-class DATA_BE {
      loss-priority low code-point 000;
    }
    forwarding-class VIDEO_AF {
      loss-priority low code-point 010;
    }
    forwarding-class NETWORK_CONTROL {
      loss-priority low code-point 110;
    }
    forwarding-class VOICE_EF {
      loss-priority low code-point 101;
    }
  }
}

scheduler-maps {
  traffic-scheduler-map {
    forwarding-class DATA_BE scheduler data;
    forwarding-class VOICE_EF scheduler voice;
    forwarding-class VIDEO_AF scheduler video;
    forwarding-class NETWORK_CONTROL scheduler network;
  }
}

schedulers {
  voice {
    transmit-rate percent 15;
    buffer-size percent 0;
    priority high;
  }
  video {
    transmit-rate percent 40;
    buffer-size percent 40;
    priority low;
  }
  network {
    transmit-rate percent 10;
    buffer-size percent 10;
    priority high;
  }
}
data {
    transmit-rate {
        remainder;
    }
    buffer-size {
        remainder;
    }
    priority low;
}
}
security {
    forwarding-options {
        family {
            mpls {
                mode packet-based;
            }
        }
    }
}
}
P-5

system {
    host-name P-5;
    root-authentication {
        encrypted-password "$1$HbwHNY.o$08YQRduNh3HniuhX01WDo."; ## SECRET-DATA
    }
    services {
        ssh;
        web-management {
            http {
                interface ge-0/0/0.0;
            }
        }
    }
    syslog {
        user * {
            any emergency;
        }
        file messages {
            any any;
            authorization info;
        }
        file interactive-commands {
            interactive-commands any;
        }
    }
    license {
        autoupdate {
            url https://ae1.juniper.net/junos/key_retrieval;
        }
    }
    interfaces {
        ge-0/0/0 {
            unit 0 {
                family inet {
                }
            }
        }
    }
}
address 10.210.30.14/24;
}
}
ge-0/0/1 {
speed 100m;
link-mode full-duplex;
gigether-options {
    no-auto-negotiation;
}
unit 0 {
    family inet {
        address 10.10.10.50/30;
    }
    family iso;
    family mpls;
}
}
ge-0/0/2 {
speed 100m;
link-mode full-duplex;
gigether-options {
    no-auto-negotiation;
}
unit 0 {
    family inet {
        address 10.10.10.45/30;
    }
    family iso;
    family mpls;
}
}
ge-0/0/3 {
speed 100m;
link-mode full-duplex;
gigether-options {
    no-auto-negotiation;
}
unit 0 {
    family inet {
        address 10.10.10.53/30;
    }
    family iso;
    family mpls;
}
}
lo0 {
    unit 0 {
        family inet {
            address 10.10.255.5/32;
        }
        family iso {
            address 49.2222.0100.1025.5005.00;
        }
        family mpls;
    }
}
}
routing-options {

router-id 10.210.255.5;
autonomous-system 100;
}
protocols {
  rsvp {
    interface all;
  }
  mpls {
    icmp-tunneling;
    interface all;
  }
  isis {
    interface ge-0/0/1.0 {
      point-to-point;
      level 1 disable;
    }
    interface ge-0/0/2.0 {
      point-to-point;
      level 1 disable;
    }
    interface ge-0/0/3.0 {
      point-to-point;
      level 1 disable;
    }
    interface lo0.0 {
      level 1 disable;
    }
  }
}
class-of-service {
  classifiers {
    exp traffic_classifier_p_devices {
      forwarding-class DATA_BE {
        loss-priority low code-points 000;
      }
      forwarding-class VIDEO_AF {
        loss-priority low code-points 010;
      }
      forwarding-class NETWORK_CONTROL {
        loss-priority low code-points 110;
      }
      forwarding-class VOICE_EF {
        loss-priority low code-points 101;
      }
    }
    inet-precedence traffic_classifier_pe_devices {
      forwarding-class DATA_BE {
        loss-priority low code-points 000;
      }
      forwarding-class VIDEO_AF {
        loss-priority low code-points 010;
      }
      forwarding-class VOICE_EF {
        loss-priority low code-points 101;
      }
      forwarding-class NETWORK_CONTROL {
        loss-priority low code-points 110;
      }
    }
  }
}
forwarding-classes {
    queue 0 DATA_BE priority low;
    queue 1 VIDEO_AF priority low;
    queue 2 VOICE_EF priority high;
    queue 3 NETWORK_CONTROL priority high;
    queue 4 test priority high;
}

interfaces {
    ge-* {
        scheduler-map traffic-scheduler-map;
        unit * {
            classifiers {
                exp traffic_classifier_p_devices;
            }
            rewrite-rules {
                exp traffic_rewrite_rule;
            }
        }
    }
}

rewrite-rules {
    exp traffic_rewrite_rule {
        forwarding-class DATA_BE {
            loss-priority low code-point 000;
        }
        forwarding-class VIDEO_AF {
            loss-priority low code-point 010;
        }
        forwarding-class NETWORK_CONTROL {
            loss-priority low code-point 110;
        }
        forwarding-class VOICE_EF {
            loss-priority low code-point 101;
        }
    }
}

scheduler-maps {
    traffic-scheduler-map {
        forwarding-class DATA_BE scheduler data;
        forwarding-class VOICE_EF scheduler voice;
        forwarding-class VIDEO_AF scheduler video;
        forwarding-class NETWORK_CONTROL scheduler network;
    }
}

schedulers {
    voice {
        transmit-rate percent 15;
        buffer-size percent 0;
        priority high;
    }
    video {
        transmit-rate percent 40;
        buffer-size percent 40;
        priority low;
    }
    network {
        transmit-rate percent 10;
        buffer-size percent 10;
    }
priority high;
}
data {
transmit-rate {
    remainder;
}
buffer-size {
    remainder;
}
priority low;
}
}
}
security {
    forwarding-options {
        family {
            mpls {
                mode packet-based;
            }
        }
    }
}
}

P-6

system {
    host-name P-6;
    root-authentication {
        encrypted-password "$1$HbwHNg.0S08YQRduNh3HniuhX01WDo."; # SECRET-DATA
    }
    services {
        ssh;
        web-management {
            http {
                interface ge-0/0/0.0;
            }
        }
    }
    syslog {
        user * {
            any emergency;
        }
        file messages {
            any any;
            authorization info;
        }
        file interactive-commands {
            interactive-commands any;
        }
    }
    license {
        autoupdate {
            url https://ae1.juniper.net/junos/key_retrieval;
        }
    }
}
interfaces {
    ge-0/0/0 {
unit 0 {
    family inet {
        address 10.210.30.15/24;
    }
}

ge-0/0/1 {
    speed 100m;
    link-mode full-duplex;
    gigether-options {
        no-auto-negotiation;
    }
    unit 0 {
        family inet {
            address 10.10.10.18/30;
        }
        family iso;
        family mpls;
    }
}

ge-0/0/2 {
    speed 100m;
    link-mode full-duplex;
    gigether-options {
        no-auto-negotiation;
    }
    unit 0 {
        family inet {
            address 10.10.10.37/30;
        }
        family iso;
        family mpls;
    }
}

ge-0/0/3 {
    speed 100m;
    link-mode full-duplex;
    gigether-options {
        no-auto-negotiation;
    }
    unit 0 {
        family inet {
            address 10.10.10.49/30;
        }
        family iso;
        family mpls;
    }
}

ge-0/0/4 {
    speed 100m;
}

lo0 {
    unit 0 {
        family inet {
            address 10.10.255.6/32;
        }
        family iso {
            address 49.2222.0100.1025.5006.00;
        }
    }
}
family mpls;
}
}
}

routing-options {
    router-id 10.210.255.6;
    autonomous-system 100;
}

protocols {
    rsvp {
        interface all;
    }
    mpls {
        icmp-tunneling;
        interface all;
    }
    isis {
        interface ge-0/0/1.0 {
            point-to-point;
            level 1 disable;
        }
        interface ge-0/0/2.0 {
            point-to-point;
            level 1 disable;
        }
        interface ge-0/0/3.0 {
            point-to-point;
            level 1 disable;
        }
        interface lo0.0 {
            level 1 disable;
        }
    }
}

class-of-service {
    classifiers {
        exp traffic_classifier_p_devices {
            forwarding-class DATA_BE {
                loss-priority low code-points 000;
            }
            forwarding-class VIDEO_AF {
                loss-priority low code-points 010;
            }
            forwarding-class NETWORK_CONTROL {
                loss-priority low code-points 110;
            }
            forwarding-class VOICE_EF {
                loss-priority low code-points 101;
            }
        }
        inet-precedence traffic_classifier_pe_devices {
            forwarding-class DATA_BE {
                loss-priority low code-points 000;
            }
            forwarding-class VIDEO_AF {
                loss-priority low code-points 010;
            }
            forwarding-class VOICE_EF {
                loss-priority low code-points 101;
            }
        }
    }
}
forwarding-class NETWORK_CONTROL {
    loss-priority low code-points 110;
}

forwarding-classes {
    queue 0 DATA_BE priority low;
    queue 1 VIDEO_AF priority low;
    queue 2 VOICE_EF priority high;
    queue 3 NETWORK_CONTROL priority high;
    queue 4 test priority high;
}

interfaces {
    ge-* {
        scheduler-map traffic-scheduler-map;
        unit * {
            classifiers {
                exp traffic_classifier_p_devices;
            }
            rewrite-rules {
                exp traffic_rewrite_rule;
            }
        }
    }
}

rewrite-rules {
    exp traffic_rewrite_rule {
        forwarding-class DATA_BE {
            loss-priority low code-point 000;
        }
        forwarding-class VIDEO_AF {
            loss-priority low code-point 010;
        }
        forwarding-class NETWORK_CONTROL {
            loss-priority low code-point 110;
        }
        forwarding-class VOICE_EF {
            loss-priority low code-point 101;
        }
    }
}

scheduler-maps {
    traffic-scheduler-map {
        forwarding-class DATA_BE scheduler data;
        forwarding-class VOICE_EF scheduler voice;
        forwarding-class VIDEO_AF scheduler video;
        forwarding-class NETWORK_CONTROL scheduler network;
    }
}

schedulers {
    voice {
        transmit-rate percent 15;
        buffer-size percent 0;
        priority high;
    }
    video {
        transmit-rate percent 40;
        buffer-size percent 40;
    }
}
priority low;
}

network {
  transmit-rate percent 10;
  buffer-size percent 10;
  priority high;
}
data {
  transmit-rate {
    remainder;
  }
  buffer-size {
    remainder;
  }
  priority low;
}
}

security {
  forwarding-options {
    family {
      mpls {
        mode packet-based;
      }
    }
  }
}

P-7

system {
  host-name P-7;
  root-authentication {
    encrypted-password "$1$HbwHNY.o$08YQRduNh3HnihuX01WDo."; ## SECRET-DATA
  }
  services {
    ssh;
    web-management {
      http {
        interface ge-0/0/0.0;
      }
    }
  }
  syslog {
    user * {
      any emergency;
    }
    file messages {
      any any;
      authorization info;
    }
    file interactive-commands {
      interactive-commands any;
    }
  }
  license {
    autoupdate {
      url https://ae1.juniper.net/junos/key_retrieval;
interfaces {
  ge-0/0/0 {
    unit 0 {
      family inet {
        address 10.210.30.16/24;
      }
    }
  }
  ge-0/0/1 {
    speed 100m;
    link-mode full-duplex;
    gigether-options {
      no-auto-negotiation;
    }
    unit 0 {
      family inet {
        address 10.10.10.34/30;
      }
      family iso;
      family mpls;
    }
  }
  ge-0/0/2 {
    speed 100m;
    link-mode full-duplex;
    gigether-options {
      no-auto-negotiation;
    }
    unit 0 {
      family inet {
        address 10.10.10.42/30;
      }
      family iso;
      family mpls;
    }
  }
  ge-0/0/3 {
    speed 100m;
    link-mode full-duplex;
    gigether-options {
      no-auto-negotiation;
    }
    unit 0 {
      family inet {
        address 10.10.10.46/30;
      }
      family iso;
      family mpls;
    }
  }
  ge-0/0/4 {
    speed 100m;
    link-mode full-duplex;
    gigether-options {
      no-auto-negotiation;
    }
    unit 0 {
      family inet {
        address 10.10.10.50/30;
      }
      family iso;
      family mpls;
    }
  }
}
unit 0 {
  family inet {
    address 10.10.10.85/30;
  }
  family iso;
  family mpls;
}
}
ge-0/0/5 {
  speed 100m;
  link-mode full-duplex;
  gigether-options {
    no-auto-negotiation;
  }
  unit 0 {
    family inet {
      address 10.10.10.89/30;
    }
    family iso;
    family mpls;
  }
}
lo0 {
  unit 0 {
    family inet {
      address 10.10.255.7/32;
    }
    family iso {
      address 49.2222.0100.1025.5007.00;
    }
    family mpls;
  }
}
}
routing-options {
  router-id 10.210.255.7;
  autonomous-system 100;
}
protocols {
  rsvp {
    interface all;
  }
  mpls {
    icmp-tunneling;
    interface all;
  }
  isis {
    interface ge-0/0/1.0 {
      point-to-point;
      level 1 disable;
    }
    interface ge-0/0/2.0 {
      point-to-point;
      level 1 disable;
    }
    interface ge-0/0/3.0 {
      point-to-point;
      level 1 disable;
    }
  }
interface ge-0/0/4.0 {
    point-to-point;
    level 1 disable;
}
interface ge-0/0/5.0 {
    point-to-point;
    level 1 disable;
}
interface lo0.0 {
    level 1 disable;
}
}
}
class-of-service {
    classifiers {
        exp traffic_classifier_p_devices {
            forwarding-class DATA_BE {
                loss-priority low code-points 000;
            }
            forwarding-class VIDEO_AF {
                loss-priority low code-points 010;
            }
            forwarding-class NETWORK_CONTROL {
                loss-priority low code-points 110;
            }
            forwarding-class VOICE_EF {
                loss-priority low code-points 101;
            }
        }
        inet-precedence traffic_classifier_pe_devices {
            forwarding-class DATA_BE {
                loss-priority low code-points 000;
            }
            forwarding-class VIDEO_AF {
                loss-priority low code-points 010;
            }
            forwarding-class VOICE_EF {
                loss-priority low code-points 101;
            }
            forwarding-class NETWORK_CONTROL {
                loss-priority low code-points 110;
            }
        }
    }
    forwarding-classes {
        queue 0 DATA_BE priority low;
        queue 1 VIDEO_AF priority low;
        queue 2 VOICE_EF priority high;
        queue 3 NETWORK_CONTROL priority high;
        queue 4 test priority high;
    }
    interfaces {
        ge-* {
            scheduler-map traffic-scheduler-map;
            unit * {
                classifiers {
                    exp traffic_classifier_p_devices;
                }
                rewrite-rules {
            }
exp traffic_rewrite_rule;
}
}
}
rewrite-rules {
exp traffic_rewrite_rule {
  forwarding-class DATA_BE {
    loss-priority low code-point 000;
  }
  forwarding-class VIDEO_AF {
    loss-priority low code-point 010;
  }
  forwarding-class NETWORK_CONTROL {
    loss-priority low code-point 110;
  }
  forwarding-class VOICE_EF {
    loss-priority low code-point 101;
  }
}
}
scheduler-maps {
  traffic-scheduler-map {
    forwarding-class DATA_BE scheduler data;
    forwarding-class VOICE_EF scheduler voice;
    forwarding-class VIDEO_AF scheduler video;
    forwarding-class NETWORK_CONTROL scheduler network;
  }
}
schedulers {
  voice {
    transmit-rate percent 15;
    buffer-size percent 0;
    priority high;
  }
  video {
    transmit-rate percent 40;
    buffer-size percent 40;
    priority low;
  }
  network {
    transmit-rate percent 10;
    buffer-size percent 10;
    priority high;
  }
  data {
    transmit-rate {
      remainder;
    }
    buffer-size {
      remainder;
    }
    priority low;
  }
}
security {
  forwarding-options {
    family {

mpls {
    mode packet-based;
}
}
}

P-8

system {
    host-name P-8;
    root-authentication {
        encrypted-password "$1$HbwHNy.o$08YQRduNh3HniuhX01WDo."; ## SECRET-DATA
    }
    services {
        ssh;
        web-management {
            http {
                interface ge-0/0/0.0;
            }
        }
    }
    syslog {
        user * {
            any emergency;
        }
        file messages {
            any any;
            authorization info;
        }
        file interactive-commands {
            interactive-commands any;
        }
    }
    license {
        autoupdate {
            url https://ae1.juniper.net/junos/key_retrieval;
        }
    }
}

interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 10.210.30.17/24;
            }
        }
    }
    ge-0/0/1 {
        speed 100m;
        link-mode full-duplex;
        gigether-options {
            no-auto-negotiation;
        }
        unit 0 {
            family inet {
                address 10.10.10.38/30;
            }
        }
    }
}
family iso;
family mpls;
}

ge-0/0/2 {
speed 100m;
link-mode full-duplex;
gigether-options {
    no-auto-negotiation;
}
unit 0 {
    family inet {
        address 10.10.10.54/30;
    }
    family iso;
    family mpls;
}
}
ge-0/0/3 {
speed 100m;
link-mode full-duplex;
gigether-options {
    no-auto-negotiation;
}
unit 0 {
    family inet {
        address 10.10.10.57/30;
    }
    family iso;
    family mpls;
}
}
ge-0/0/4 {
speed 100m;
link-mode full-duplex;
gigether-options {
    no-auto-negotiation;
}
unit 0 {
    family inet {
        address 10.10.10.65/30;
    }
    family iso;
    family mpls;
}
}
ge-0/0/5 {
speed 100m;
link-mode full-duplex;
gigether-options {
    no-auto-negotiation;
}
unit 0 {
    family inet {
        address 10.10.10.61/30;
    }
    family iso;
    family mpls;
}
lo0 {
  unit 0 {
    family inet {
      address 10.10.255.8/32;
    }
    family iso {
      address 49.2222.0100.1025.5008.00;
    }
    family mpls;
  }
}

routin
forwarding-class NETWORK_CONTROL {
    loss-priority low code-points 110;
}
forwarding-class VOICE_EF {
    loss-priority low code-points 101;
}

inet-precedence traffic_classifier_pe_devices {
    forwarding-class DATA_BE {
        loss-priority low code-points 000;
    }
    forwarding-class VIDEO_AF {
        loss-priority low code-points 010;
    }
    forwarding-class VOICE_EF {
        loss-priority low code-points 101;
    }
    forwarding-class NETWORK_CONTROL {
        loss-priority low code-points 110;
    }
}

forwarding-classes {
    queue 0 DATA_BE priority low;
    queue 1 VIDEO_AF priority low;
    queue 2 VOICE_EF priority high;
    queue 3 NETWORK_CONTROL priority high;
    queue 4 test priority high;
}

interfaces {
    ge-* {
        scheduler-map traffic-scheduler-map;
        unit * {
            classifiers {
                exp traffic_classifier_p_devices;
            }
            rewrite-rules {
                exp traffic_rewrite_rule;
            }
        }
    }
}

rewrite-rules {
    exp traffic rewrite_rule {
        forwarding-class DATA_BE {
            loss-priority low code-point 000;
        }
        forwarding-class VIDEO_AF {
            loss-priority low code-point 010;
        }
        forwarding-class NETWORK_CONTROL {
            loss-priority low code-point 110;
        }
        forwarding-class VOICE_EF {
            loss-priority low code-point 101;
        }
    }
}
scheduler-maps {
  traffic-scheduler-map {
    forwarding-class DATA_BE scheduler data;
    forwarding-class VOICE_EF scheduler voice;
    forwarding-class VIDEO_AF scheduler video;
    forwarding-class NETWORK_CONTROL scheduler network;
  }
}
schedulers {
  voice {
    transmit-rate percent 15;
    buffer-size percent 0;
    priority high;
  }
  video {
    transmit-rate percent 40;
    buffer-size percent 40;
    priority low;
  }
  network {
    transmit-rate percent 10;
    buffer-size percent 10;
    priority high;
  }
  data {
    transmit-rate {
      remainder;
    }
    buffer-size {
      remainder;
    }
    priority low;
  }
}
security {
  forwarding-options {
    family {
      mpls {
        mode packet-based;
      }
    }
  }
}

P-9

system {
  host-name P-9;
  root-authentication {
    encrypted-password "$1$HbwHNY.o$O8YQRdUnh3HniuX01WDo.";## SECRET-­DATA
  }
  services {
    ssh;
    web-management {
      http {
        interface ge-0/0/0.0;
      }
    }
  }
}
syslog {
    user * {
        any emergency;
    }
    file messages {
        any any;
        authorization info;
    }
    file interactive-commands {
        interactive-commands any;
    }
}
license {
    autoupdate {
        url https://ae1.juniper.net/junos/key_retrieval;
    }
}
interfaces {
    ge-0/0/0 {
        unit 0 {
            family inet {
                address 10.210.30.18/24;
            }
        }
    }
    ge-0/0/1 {
        speed 100m;
        link-mode full-duplex;
        gigether-options {
            no-auto-negotiation;
        }
        unit 0 {
            family inet {
                address 10.10.10.62/30;
            }
            family iso;
            family mpls;
        }
    }
    ge-0/0/2 {
        speed 100m;
        link-mode full-duplex;
        gigether-options {
            no-auto-negotiation;
        }
        unit 0 {
            family inet {
                address 10.10.10.77/30;
            }
            family iso;
            family mpls;
        }
    }
    ge-0/0/3 {
        speed 100m;
        link-mode full-duplex;
gigether-options {
    no-auto-negotiation;
}
unit 0 {
    family inet {
        address 10.10.10.73/30;
    }
    family iso;
    family mpls;
}
}
lo0 {
    unit 0 {
        family inet {
            address 10.10.255.9/32;
        }
        family iso {
            address 49.2222.0100.1025.5009.00;
        }
        family mpls;
    }
}
}
routing-options {
    router-id 10.210.255.9;
    autonomous-system 100;
}
protocols {
    rsvp {
        interface all;
    }
    mpls {
        icmp-tunneling;
        interface all;
    }
    isis {
        interface ge-0/0/1.0 {
            point-to-point;
            level 1 disable;
        }
        interface ge-0/0/2.0 {
            point-to-point;
            level 1 disable;
        }
        interface ge-0/0/3.0 {
            point-to-point;
            level 1 disable;
        }
        interface lo0.0 {
            level 1 disable;
        }
    }
}
class-of-service {
    classifiers {
        exp traffic_classifier_p_devices {
            forwarding-class DATA_BE {
                loss-priority low code-points 000;
            }
        }
    }
}
forwarding-class VIDEO_AF {
    loss-priority low code-points 010;
}
forwarding-class NETWORK_CONTROL {
    loss-priority low code-points 110;
}
forwarding-class VOICE_EF {
    loss-priority low code-points 101;
}
}
inet-precedence traffic_classifier_pe_devices {
    forwarding-class DATA_BE {
        loss-priority low code-points 000;
    }
    forwarding-class VIDEO_AF {
        loss-priority low code-points 010;
    }
    forwarding-class VOICE_EF {
        loss-priority low code-points 101;
    }
    forwarding-class NETWORK_CONTROL {
        loss-priority low code-points 110;
    }
}
forwarding-classes {
    queue 0 DATA_BE priority low;
    queue 1 VIDEO_AF priority low;
    queue 2 VOICE_EF priority high;
    queue 3 NETWORK_CONTROL priority high;
    queue 4 test priority high;
}
interfaces {
    ge-* {
        scheduler-map traffic-scheduler-map;
        unit * {
            classifiers {
                exp traffic_classifier_p_devices;
            }
            rewrite-rules {
                exp traffic_rewrite_rule;
            }
        }
    }
    rewrite-rules {
        exp traffic_rewrite_rule {
            forwarding-class DATA_BE {
                loss-priority low code-point 000;
            }
            forwarding-class VIDEO_AF {
                loss-priority low code-point 010;
            }
            forwarding-class NETWORK_CONTROL {
                loss-priority low code-point 110;
            }
            forwarding-class VOICE_EF {
                loss-priority low code-point 101;
            }
        }
    }
MPLS & QoS in Virtual Environments

```c
scheduler-maps {
    traffic-scheduler-map {
        forwarding-class DATA_BE scheduler data;
        forwarding-class VOICE_EF scheduler voice;
        forwarding-class VIDEO_AF scheduler video;
        forwarding-class NETWORK_CONTROL scheduler network;
    }
}
schedulers {
    voice {
        transmit-rate percent 15;
        buffer-size percent 0;
        priority high;
    }
    video {
        transmit-rate percent 40;
        buffer-size percent 40;
        priority low;
    }
    network {
        transmit-rate percent 10;
        buffer-size percent 10;
        priority high;
    }
    data {
        transmit-rate {
            remainder;
        }
        buffer-size {
            remainder;
        }
        priority low;
    }
}
security {
    forwarding-options {
        family {
            mpls {
                mode packet-based;
            }
        }
    }
}
}

P-10

system {
    host-name P-10;
    root-authentication {
        encrypted-password "$1$HbwHNg.oSo8YQRduNh3HniuhX01WDo.\n"; ## SECRET-DATA
    }
    services {
        ssh;
        web-management {
            http {
```
interface ge-0/0/0;  }
}
syslog {
  user * {
    any emergency;
  }
  file messages {
    any any;
    authorization info;
  }
  file interactive-commands {
    interactive-commands any;
  }
}
license {
  autoupdate {
    url https://ae1.juniper.net/junos/key_retrieval;
  }
}
interfaces {
  ge-0/0/0 {
    unit 0 {
      family inet {
        address 10.210.30.19/24;
      }
    }
  }
  ge-0/0/1 {
    speed 100m;
    link-mode full-duplex;
    gigether-options {
      no-auto-negotiation;
    }
    unit 0 {
      family inet {
        address 10.10.10.66/30;
      }
      family iso;
      family mpls;
    }
  }
  ge-0/0/2 {
    speed 100m;
    link-mode full-duplex;
    gigether-options {
      no-auto-negotiation;
    }
    unit 0 {
      family inet {
        address 10.10.78/30;
      }
      family iso;
      family mpls;
    }
  }
  ge-0/0/3 {

speed 100m;
link-mode full-duplex;
gigether-options {
  no-auto-negotiation;
}
unit 0 {
  family inet {
    address 10.10.10.81/30;
  }
  family iso;
  family mpls;
}

gi-0/0/4 {
  speed 100m;
  link-mode full-duplex;
  gigether-options {
    no-auto-negotiation;
  }
  unit 0 {
    family inet {
      address 10.10.10.70/30;
    }
    family iso;
    family mpls;
  }
}
lo0 {
  unit 0 {
    family inet {
      address 10.10.255.10/32;
    }
    family iso {
      address 49.222.0100.1025.5010.00;
    }
    family mpls;
  }
}

routing-options {
  router-id 10.210.255.10;
  autonomous-system 100;
}

protocols {
  rsvp {
    interface all;
  }
  mpls {
    icmp-tunneling;
    interface all;
  }
  isis {
    interface ge-0/0/1.0 {
      point-to-point;
      level 1 disable;
    }
    interface ge-0/0/2.0 {
      point-to-point;
      level 1 disable;
    }
  }
}
interface ge-0/0/3.0 {
    point-to-point;
    level 1 disable;
}
interface ge-0/0/4.0 {
    point-to-point;
    level 1 disable;
}
interface lo0.0 {
    level 1 disable;
}

class-of-service {
    classifiers {
        exp traffic_classifier_p_devices {
            forwarding-class DATA_BE {
                loss-priority low code-points 000;
            }
            forwarding-class VIDEO_AF {
                loss-priority low code-points 010;
            }
            forwarding-class NETWORK_CONTROL {
                loss-priority low code-points 110;
            }
            forwarding-class VOICE_EF {
                loss-priority low code-points 101;
            }
        }
        inet-precedence traffic_classifier_pe_devices {
            forwarding-class DATA_BE {
                loss-priority low code-points 000;
            }
            forwarding-class VIDEO_AF {
                loss-priority low code-points 010;
            }
            forwarding-class VOICE_EF {
                loss-priority low code-points 101;
            }
            forwarding-class NETWORK_CONTROL {
                loss-priority low code-points 110;
            }
        }
    }
    forwarding-classes {
        queue 0 DATA_BE priority low;
        queue 1 VIDEO_AF priority low;
        queue 2 VOICE_EF priority high;
        queue 3 NETWORK_CONTROL priority high;
        queue 4 test priority high;
    }
    interfaces {
        ge-* {
            scheduler-map traffic-scheduler-map;
            unit * {
                classifiers {
                    exp traffic_classifier_p_devices;
                }
            }
        }
    }
}
rewrite-rules {
    exp traffic_rewrite_rule;
}
}
}
rewrite-rules {
    exp traffic_rewrite_rule {
        forwarding-class DATA_BE {
            loss-priority low code-point 000;
        }
        forwarding-class VIDEO_AF {
            loss-priority low code-point 010;
        }
        forwarding-class NETWORK_CONTROL {
            loss-priority low code-point 110;
        }
        forwarding-class VOICE_EF {
            loss-priority low code-point 101;
        }
    }
}
}
scheduler-maps {
    traffic-scheduler-map {
        forwarding-class DATA_BE scheduler data;
        forwarding-class VOICE_EF scheduler voice;
        forwarding-class VIDEO_AF scheduler video;
        forwarding-class NETWORK_CONTROL scheduler network;
    }
}
schedulers {
    voice {
        transmit-rate percent 15;
        buffer-size percent 0;
        priority high;
    }
    video {
        transmit-rate percent 40;
        buffer-size percent 40;
        priority low;
    }
    network {
        transmit-rate percent 10;
        buffer-size percent 10;
        priority high;
    }
    data {
        transmit-rate {
            remainder;
        }
        buffer-size {
            remainder;
        }
        priority low;
    }
}
security {
    forwarding-options {

family {
  mpls {
    mode packet-based;
  }
}

P-11

system {
  host-name P-11;
  root-authentication {
    encrypted-password "$1$HbwHNY.o$o8YQRduNh3HniuhX01WDo."; ## SECRET-DATA
  }
  services {
    ssh;
    web-management {
      http {
        interface ge-0/0/0.0;
      }
    }
  }
  syslog {
    user * {
      any emergency;
    }
    file messages {
      any any; authorization info;
    }
    file interactive-commands {
      interactive-commands any;
    }
  }
  license {
    autoupdate {
      url https://ae1.juniper.net/junos/key_retrieval;
    }
  }
}

interfaces {
  ge-0/0/0 {
    unit 0 {
      family inet {
        address 10.210.30.20/24;
      }
    }
  }
  ge-0/0/1 {
    speed 100m;
    link-mode full-duplex;
    gigether-options {
      no-auto-negotiation;
    }
    unit 0 {
      family inet {
        address 10.10.10.86/30;
      }
    }
  }
}
MPLS & QoS in Virtual Environments

```plaintext
point-to-point;
level 1 disable;
}
interface ge-0/0/2.0 {
  point-to-point;
  level 1 disable;
}
interface ge-0/0/3.0 {
  point-to-point;
  level 1 disable;
}
interface lo0.0 {
  level 1 disable;
}

class-of-service {
  classifiers {
    exp traffic_classifier_p_devices {
      forwarding-class DATA_BE {
        loss-priority low code-points 000;
      }
      forwarding-class VIDEO_AF {
        loss-priority low code-points 010;
      }
      forwarding-class NETWORK_CONTROL {
        loss-priority low code-points 110;
      }
      forwarding-class VOICE_EF {
        loss-priority low code-points 101;
      }
    }
    inet-precedence traffic_classifier_pe_devices {
      forwarding-class DATA_BE {
        loss-priority low code-points 000;
      }
      forwarding-class VIDEO_AF {
        loss-priority low code-points 010;
      }
      forwarding-class VOICE_EF {
        loss-priority low code-points 101;
      }
      forwarding-class NETWORK_CONTROL {
        loss-priority low code-points 110;
      }
    }
  }
  forwarding-classes {
    queue 0 DATA_BE priority low;
    queue 1 VIDEO_AF priority low;
    queue 2 VOICE_EF priority high;
    queue 3 NETWORK_CONTROL priority high;
    queue 4 test priority high;
  }
  interfaces {
    ge-* {
      scheduler-map traffic-scheduler-map;
      unit * {
        classifiers {
        }
      }
    }
  }
```

Amar Madhavan
exp traffic_classifier_p_devices;
}
rewrite-rules {
  exp traffic_rewrite_rule;
}
}
}
rewrite-rules {
  exp traffic_rewrite_rule {
    forwarding-class DATA_BE {
      loss-priority low code-point 000;
    }
    forwarding-class VIDEO_AF {
      loss-priority low code-point 010;
    }
    forwarding-class NETWORK_CONTROL {
      loss-priority low code-point 110;
    }
    forwarding-class VOICE_EF {
      loss-priority low code-point 101;
    }
  }
}
scheduler-maps {
  traffic-scheduler-map {
    forwarding-class DATA_BE scheduler data;
    forwarding-class VOICE_EF scheduler voice;
    forwarding-class VIDEO_AF scheduler video;
    forwarding-class NETWORK_CONTROL scheduler network;
  }
}
schedulers {
  video {
    transmit-rate percent 40;
    buffer-size percent 40;
    priority low;
  }
  network {
    transmit-rate percent 10;
    buffer-size percent 10;
    priority high;
  }
  data {
    transmit-rate {
      remainder;
    }
    buffer-size {
      remainder;
    }
    priority low;
  }
  voice {
    transmit-rate percent 15;
    buffer-size percent 0;
    priority high;
  }
}
MPLS & QoS in Virtual Environments

security {
    forwarding-options {
        family {
            mpls {
                mode packet-based;
            }
            ...
        }
    }
}

PE-2

system {
    host-name PE-2;
    root-authentication {
        encrypted-password "$1$HbwHNy.o$08YQRduNh3HniuhX01WDo."; ## SECRET-DATA
    }
    services {
        ssh;
        web-management {
            http {
                interface ge-0/0/0.0;
            }
            ...
        }
    }
    syslog {
        user * {
            any emergency;
        }
        file messages {
            any any;
            authorization info;
        }
        file interactive-commands {
            interactive-commands any;
        }
        license {
            autoupdate {
                url https://ae1.juniper.net/junos/key_retrieval;
            }
        }
    }
    interfaces {
        ge-0/0/0 {
            unit 0 {
                family inet {
                    address 10.210.30.3/24;
                }
            }
        }
        ge-0/0/1 {
            speed 100m;
            link-mode full-duplex;
            gigether-options {
                no-auto-negotiation;
            }
            unit 0 {
                ...
            }
        }
    }
}
family inet {
    address 20.20.20.5/30;
}
family iso;
}ge-0/0/2 {
    speed 100m;
    link-mode full-duplex;
    gigether-options {
        no-auto-negotiation;
    }
    unit 0 {
        family inet {
            address 10.10.10.74/30;
        }
        family iso;
        family mpls;
    }
}ge-0/0/3 {
    speed 100m;
    link-mode full-duplex;
    gigether-options {
        no-auto-negotiation;
    }
    unit 0 {
        family inet {
            address 10.10.10.82/30;
        }
        family iso;
        family mpls;
    }
}ge-0/0/4 {
    speed 100m;
    link-mode full-duplex;
    gigether-options {
        no-auto-negotiation;
    }
    unit 0 {
        family inet {
            address 10.10.10.90/30;
        }
        family iso;
        family mpls;
    }
}lo0 {
    unit 0 {
        family inet {
            address 10.10.255.200/32;
        }
        family iso {
            address 49.2222.0100.1025.5200.00;
        }
        family mpls;
    }
}
MPLS & QoS in Virtual Environments

Amar Madhavan

130

```plaintext
} routing-options {
  router-id 10.210.255.200;
  autonomous-system 100;
  forwarding-table {
    export load-bal;
  }
}

protocols {
  rsvp {
    interface all;
  }
  mpls {
    auto-policing {
      class all drop;
    }
    icmp-tunneling;
    no-cspf;
    oam {
      bfd-liveness-detection {
        minimum-interval 300;
        multiplier 2;
        failure-action teardown;
      }
    }
  }
  label-switched-path PE-2-to-PE-1 {
    to 10.10.255.100;
    bandwidth 10m;
    fast-reroute {
      hop-limit 15;
    }
    primary Best_Path;
    secondary Failover_Path {
      standby;
    }
  }
  path Best_Path {
    10.10.10.89 strict;
    10.10.10.33 strict;
    10.10.10.5 strict;
  }
  path Failover_Path {
    10.10.10.81 strict;
    10.10.10.77 strict;
    10.10.10.61 strict;
    10.10.10.53 strict;
    10.10.10.49 strict;
    10.10.10.17 strict;
    10.10.10.13 strict;
    10.10.10.1 strict;
  }
  interface all;
}

bgp {
  group internal {
    type internal;
    local-address 10.10.255.200;
    family inet {
      unicast;
    }
  }
```
family inet-vpn {
    unicast;
}
export nhs;
neighbor 10.10.255.100;
}
}
isis {
    interface ge-0/0/2.0 {
        point-to-point;
        level 1 disable;
    }
    interface ge-0/0/3.0 {
        point-to-point;
        level 1 disable;
    }
    interface ge-0/0/4.0 {
        point-to-point;
        level 1 disable;
    }
    interface lo0.0 {
        level 1 disable;
    }
}
}
policy-options {
    policy-statement load-bal {
        then {
            load-balance per-packet;
        }
    }
    policy-statement nhs {
        then {
            next-hop self;
        }
    }
    policy-statement vrf-export-policy {
        term 1 {
            from protocol [ bgp direct ];
            then {
                community add customer-a;
                accept;
            }
        }
        term 2 {
            then reject;
        }
    }
    policy-statement vrf-import-policy {
        term 1 {
            from {
                protocol bgp;
                community customer-a;
            }
            then accept;
        }
        term 2 {
            then reject;
        }
    }
}
community customer-a members target:100:200;

class-of-service {
  classifiers {
    exp traffic_classifier_p_devices {
      forwarding-class DATA_BE {
        loss-priority low code-points 000;
      }
      forwarding-class VIDEO_AF {
        loss-priority low code-points 010;
      }
      forwarding-class NETWORK_CONTROL {
        loss-priority low code-points 110;
      }
      forwarding-class VOICE_EF {
        loss-priority low code-points 101;
      }
    }
    inet-precedence traffic_classifier_pe_devices {
      forwarding-class DATA_BE {
        loss-priority low code-points 000;
      }
      forwarding-class VIDEO_AF {
        loss-priority low code-points 010;
      }
      forwarding-class VOICE_EF {
        loss-priority low code-points 101;
      }
      forwarding-class NETWORK_CONTROL {
        loss-priority low code-points 110;
      }
    }
  }
  forwarding-classes {
    queue 0 DATA_BE priority low;
    queue 1 VIDEO_AF priority low;
    queue 2 VOICE_EF priority high;
    queue 3 NETWORK_CONTROL priority high;
    queue 4 test priority high;
  }
  interfaces {
    ge-* {
      scheduler-map traffic-scheduler-map;
      unit * {
        classifiers {
          exp traffic_classifier_p_devices;
        }
        rewrite-rules {
          exp traffic_rewrite_rule;
        }
      }
    }
    ge-0/0/1 {
      scheduler-map traffic-scheduler-map;
      unit 0 {
        classifiers {
          inet-precedence traffic_classifier_pe_devices;
        }
      }
    }
  }
}
rewrite-rules {
    exp traffic_rewrite_rule {
        forwarding-class DATA_BE {
            loss-priority low code-point 000;
        }
        forwarding-class VIDEO_AF {
            loss-priority low code-point 010;
        }
        forwarding-class NETWORK_CONTROL {
            loss-priority low code-point 110;
        }
        forwarding-class VOICE_EF {
            loss-priority low code-point 101;
        }
    }
}

scheduler-maps {
    traffic-scheduler-map {
        forwarding-class DATA_BE scheduler data;
        forwarding-class VOICE_EF scheduler voice;
        forwarding-class VIDEO_AF scheduler video;
        forwarding-class NETWORK_CONTROL scheduler network;
    }
}
schedulers {
    voice {
        transmit-rate percent 15;
        buffer-size percent 0;
        priority high;
    }
    video {
        transmit-rate percent 40;
        buffer-size percent 40;
        priority low;
    }
    network {
        transmit-rate percent 10;
        buffer-size percent 10;
        priority high;
    }
    data {
        transmit-rate {
            remainder;
        }
        buffer-size {
            remainder;
        }
        priority low;
    }
}
security {
forwarding-options {
    family {
        mpls {
            mode packet-based;
        }
    }
}

routing-instances {
    CE-A {
        instance-type vrf;
        interface ge-0/0/1.0;
        route-distinguisher 10.10.255.200:1;
        vrf-import vrf-import-policy;
        vrf-export vrf-export-policy;
        vrf-table-label;
        protocols {
            bgp {
                group external {
                    peer-as 50;
                    as-override;
                    neighbor 20.20.20.6;
                }
            }
        }
    }
}

CE-2

system {
    host-name CE-2;
    root-authentication {
        encrypted-password "$1$GQtlaQSc$xhd6oMzDiN4j8vTWynIdx/$"; # SECRET-DATA
    }
    services {
        ftp;
        ssh;
        telnet;
        web-management {
            http {
                interface ge-0/0/0.0;
            }
        }
    }
    syslog {
        user * {
            any emergency;
        }
        file messages {
            any any;
            authorization info;
        }
        file interactive-commands {
            interactive-commands any;
        }
    }
}
license {
autoupdate {
    url https://ae1.juniper.net/junos/key_retrieval;
} } } } interfaces { ge-0/0/0 { unit 0 { family inet { address 10.210.30.8/24; } } } ge-0/0/1 { unit 0 { family inet { address 20.20.20.6/30; } } } } ge-0/0/4 { unit 0 { family inet { address 10.10.30.1/30; } } } } lo0 { unit 0 { family inet { address 20.20.255.2/32; } } } } } } routing-options { static { route 60.60.60.60/32 receive; route 0.0.0.0/0 receive; } router-id 20.20.255.2; autonomous-system 50; } } protocols { bgp { group external { type external; export test-route; peer-as 100; neighbor 20.20.20.5; } } } } policy-options { policy-statement test-route { term 1 { from protocol [ static direct ]; then accept; } } }
security {
    forwarding-options {
        family {
            mpls {
                mode packet-based;
            }
        }
    }
}

services {
    rpm;
}
References


Maini, K. S. (2002). *An MPLS implementation study (MS Thesis).*


Tan, C. C. (2002). *Performance analysis of voice traffic in MPLS communication networks (MS Thesis).*


