Secure Network Design and Implementation using MPLS VPN

MSc Research Project
Cybersecurity

Mayur Mohan Mungse
Student ID: X21188823

School of Computing
National College of Ireland

Supervisor: Prof. Imran Khan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Mr. Mayur Mohan Mungse		
Student ID:	X21188823		
Programme:	Master’s Cybersecurity	**Year:**	2023-24
Module:	Research Project		
Supervisor:	Prof. Imran Khan		
Submission Due Date:	14th December 2023		
Project Title:	Secure Network Design and Implementation using MPLS VPN		
Word Count:	**6426**	**Page Count: 36**	

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Mayur Mohan Mungse

Date: 14/12/2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Contents

Secure Network Design and Implementation of MPLS VPN

Mayur Mohan Mungse
X21188823

Abstract

Secure and effective communication networks are critical in today's data-driven, networked environment. Virtual private networks using multiprotocol label switching, or MPLS VPNs, have become a strong option for enterprises looking to create dependable and safe connections inside their network infrastructure. An overview of the main ideas and advantages of designing and implementing secure networks with MPLS VPN technology is given in this abstract. With the security and isolation characteristics of conventional VPNs combined with the scalability and flexibility of MPLS, MPLS VPNs provide an adaptable framework. This combination gives businesses a strong tool for dividing and securing communication channels across a common network architecture.

The importance of MPLS VPNs in creating a secure network architecture that satisfies the needs of contemporary enterprises is highlighted in this abstract. Using MPLS VPNs for secure network design and implementation has become essential to modern IT strategy, enabling enterprises to confidently and resiliently traverse the complicated world of connection.

Keywords : Multiprotocol Label Switching, Virtual Private Networks, Virtual Forwarding Table

1 Introduction

Designing and implementing a secure network infrastructure is of paramount importance in today's digital landscape. As businesses and organizations rely on data-driven operations, secure communication, and uninterrupted connectivity, the need for robust network security has never been more critical. In response to these demands, Multiprotocol Label Switching Virtual Private Networks (MPLS VPNs) have emerged as a versatile and powerful solution for creating secure and efficient networks. This comprehensive introduction explores the fundamental concepts, challenges, and benefits associated with secure network design and implementation using MPLS VPN technology. (Tony Li, 1999)

Because the digital era has tightly intertwined the digital world into our daily lives, network security has become a critical factor. The attack surface for malicious actors has grown as linked gadgets, the Internet of Things (IoT), cloud computing, and remote work have proliferated. As a result, data breaches, cyberattacks, and network outages have gotten more sophisticated, resulting in considerable financial losses and reputational damage. As a result, a solid strategy to network security is critical.A secure network begins with careful planning, taking into account factors such as network architecture, access control, encryption, and intrusion detection systems. These components must function in unison to protect data, assure communication confidentiality and integrity, and lay the groundwork for business continuity.

MPLS (Multiprotocol Label Switching) is a core technology in modern networking that efficiently routes and forwards data packets within a network. This section delves into MPLS, outlining its origins, basic principles, and benefits, with an emphasis on how it differs from traditional routing and switching. (Ghein, 2007)

VPNs (Virtual Private Networks) have long been used to secure communications over insecure networks. This section delves into VPNs, their significance in providing private and encrypted communication channels, and the many VPN protocols and encryption methods available. The combination of MPLS and VPN technologies resulted in MPLS VPNs, a potent combination of routing efficiency and secure communication. This section recounts the evolution of MPLS VPNs, emphasizing their motives and transition into a versatile networking solution.

Customer Edge (CE) and Provider Edge (PE) Routers, Label Distribution Protocol (LDP), and Label Switching Paths (LSPs) are critical components of MPLS VPNs. These components allow for communication, the exchange of label information, and the establishment of virtual circuits for controlled routing. (Hussain, 20 Dec 2004)

MPLS VPNs are classified into three types: Layer 3 MPLS VPNs, Layer 2 MPLS VPNs, and Virtual Private LAN Service (VPLS), each with its own set of use cases and specifications.

Traffic segmentation and isolation, Quality of Service (QoS) control, scalability, flexibility, redundancy, high availability, and robust security protocols are all advantages of MPLS VPNs in secure network design. MPLS VPN implementations in enterprise networks and by service providers are highlighted in real-world case studies.

These studies demonstrate how MPLS VPNs address specific difficulties while also providing secure and efficient network designs.
Planning and design, step-by-step implementation and deployment, and ongoing monitoring and maintenance are all important factors for MPLS VPN implementation.

Considerations for the future include the growing threat landscape, the convergence of Software-Defined Networking (SDN) and MPLS technology, and the possible role of Artificial Intelligence (AI) and Machine Learning (ML) in improving MPLS VPN security.

2 Research Question

What are the techniques and concerns involved in developing and deploying a secure Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) in a shared infrastructure scenario, to protect the integrity of data and confidentiality?

To enhance the security of Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs), a comprehensive approach involves various key measures. Authentication and access control are paramount, necessitating the implementation of robust mechanisms to verify the identity of users and devices accessing the MPLS VPN. Access control policies should be employed to meticulously restrict access, allowing entry only to authorized entities and preventing any compromise by unauthorized users.

Encryption plays a pivotal role in safeguarding data confidentiality over the MPLS VPN. By utilizing encryption protocols like IPsec, data transmitted through the network is securely encrypted, ensuring its confidentiality even in the event of interception. Another crucial aspect is securing the Label Distribution Protocol (LDP) to prevent unauthorized entities from distributing or modifying labels. This can be achieved through the implementation of authentication mechanisms within the LDP.

Efficient traffic segmentation and isolation are facilitated through the inherent capabilities of MPLS VPNs. This feature ensures that data from different VPNs or customers remains distinct, thereby enhancing confidentiality. Additionally, the implementation of Quality of Service (QoS) policies is essential for prioritizing and managing network traffic based on specific requirements. This ensures optimal performance while guaranteeing that critical data receives the necessary bandwidth.

Routing security is addressed by implementing robust measures to protect against route manipulation or hijacking. Techniques such as Route Target (RT) filtering and Route Distinguisher (RD) are employed to control and secure route distribution among Virtual Routing and Forwarding (VRF) instances. Firewalls and Intrusion Prevention Systems (IPS) are deployed to monitor and filter traffic entering and leaving the MPLS VPN, providing an additional layer of security against malicious activities and unauthorized access.

3 Related Work

As the Internet has grown in popularity and use, people's awareness of VPN technology has gradually increased. It becomes a crucial area of research for network security and Internet applications. VPN stands for virtual private network, a type of special-purpose data communication network that is set up on public networks by ISPs (Internet service providers) and other NSPs. Utilizing tunnel technology is its fundamental idea. To provide point-to-point connectivity, data is encapsulated using the tunnel protocol and a private tunnel is built using an existing public network, such as the Internet, PSTN, or ISDN. It seems to be a personal physical connection. (Yongming WEI', 2006).

Security is a primary concern for Internet VPNs, especially those utilizing the public Internet for transport. Unlike private line, frame relay, and ATM-based services, IP networks do not allocate dedicated physical or logical pipes to specific applications, protocols, users, or locations. To address these security challenges, the Internet Engineering Task Force (IETF) introduced IPsec (Internet Protocol Security), a robust approach designed to enhance Internet security. Originally integrated into the IPv6 protocol, IPsec has been adapted for use in IPv4 networks such as the Internet. It introduces a modular framework aimed at providing strong and reliable security for communications over IP networks. This architecture defines protocols for establishing, maintaining, and terminating secure communication channels, often referred to as "tunnels," employing specific protocols for IP packet encryption and authentication. An additional feature of IPsec is replay protection. This functionality involves dropping packets that the network identifies as duplicates of those previously received. IPsec supports encryption algorithms such as Data Encryption Standard (DES) and Triple DES (3DES). These encryption techniques utilize challenging-to-decode secret encoding and decoding keys. (Swallow, 1999)
A significant enhancement in the latest iteration of the IPsec proposal is the inclusion of Internet Key Exchange (IKE). IKE simplifies the process of assigning keys to devices requiring encrypted connections for communication. Users are advised to insist on utilizing the most recent drafts provided by their VPN service provider. In the realm of VPN security, IPSec and IKE are widely recognized as industry standards, enjoying extensive acceptance for their robust security protocols. Users are encouraged to stay updated with the latest

developments in VPN security by consulting the most recent drafts from their VPN providers. (Haeryong Lee)

As per (Bennan) Conventionally, policing serves as the mechanism to regulate the volume of traffic admitted at network edges, aiming to curtail the potential for congestion. Policing is particularly crucial for premium classes, as their performance in terms of delay and packet loss relies on restricting their traffic volumes to specified fractions of the available bandwidth on each link in the end-to-end path. The role of the policing element is to discard packets that surpass the contracted bandwidth. Ideally, this action should be infrequent, assuming that the customer accurately specifies the required bandwidth. Simply dropping any out-of-profile traffic could result in elevated levels of dropped packets. An alternative solution involves applying remarking to out-of-profile traffic. This entails labelling in-profile and out-of-profile packets differently, making their status visible to subsequent routers along the end-to-end path. If congestion levels within the premium class are manageable, both types of packets are forwarded. However, if packet dropping becomes necessary, out-of-profile traffic is prioritized for dropping over in-profile traffic. In this scheme, there is no assurance for the admission and remarking of out-of-profile packets.

3.1 Background of MPLS VPN

(EZE, 2018) Multi-Protocol Label Switching (MPLS) emerged from the Internet Engineering Task Force (IETF) to address the limitations of conventional IP networks. It has become a cornerstone for Internet Service Providers (ISPs), particularly in the core network, ensuring enhanced Quality of Service (QoS). MPLS is widely employed by telecommunications operators and ISPs for MPLS Virtual Private Networks (MPLS-VPNs), a technology facilitating the transport and routing of diverse network traffic through an MPLS backbone.

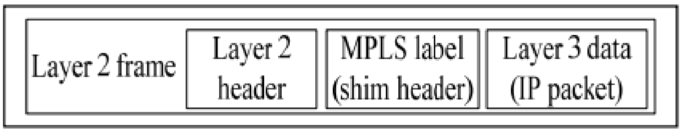
Within MPLS-VPNs, three prevalent types exist today: Point-to-Point, Layer 2, and Layer 3 MPLS-VPNs. Point-to-Point MPLS-VPNs utilize virtual leased lines, such as E1/T1 Ethernet and ATM, to establish point-to-point connectivity between two locations. Layer 2 MPLS-VPNs, also known as Virtual Private LAN Service (VPLS), provide a "switch in the cloud" service between LAN sites. Meanwhile, Layer 3 MPLS-VPNs leverage Virtual Routing and Forwarding (VRF) to segregate routing tables for individual VPN customers.

MPLS-VPNs can be implemented using two distinct approaches: Overlay and Peer-to-Peer. In the overlay method, the ISP allocates a dedicated circuit to the customer for service delivery. On the other hand, in the peer-to-peer method, the ISP establishes a peer relationship with the customer through Provider Edge (PE) routers. These PE routers are equipped with VRF instances, ensuring the isolation of each customer's routes from others.

MPLS, as outlined in RFC 3031, represents a layer 2 technology designed to enhance packet forwarding within traditional IP environments. In the conventional IP setting, routers forward packets based on the IP destination address in the header. Upon receiving a packet, a router conducts a recursive query by comparing the packet's IP destination address with its routing table for a longest-prefix-match. Subsequently, the router identifies the correct next hop and forwards the packet. However, this routing approach introduces significant delay due to the recursive query, and the router operates on a Per Hop Behavior (PHB) without scalability.

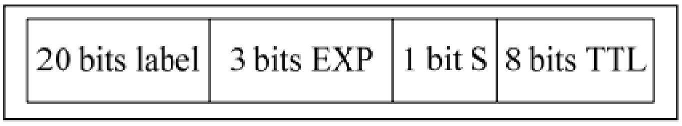
(Ming-Song Sun, 2012)Even though Policy Based Routing (PBR) can be implemented to enhance router behavior, it often leads to a substantial reduction in router performance. MPLS addresses these challenges by introducing a label between the Layer 2 header and

Layer 3 data. This label consists of a 20-bit label value, 3 bits for Experimental (EXP), 1 bit for bottom-of-stack, and 8 bits for Time-to-Live (TTL). The placement and structure of the MPLS label stack header are illustrated in Fig. 1 and Fig. 2.



Layer 2 frame	Layer 2 header	MPLS label (shim header)	Layer 3 data (IP packet)

Fig 1 Position of MPLS Label



20 bits label	3 bits EXP	1 bit S	8 bits TTL

Fig 2 MPLS Label stack header

3.2 Virtual forwarding in MPLS

A key element of (Ghein, 2007) MPLS (Multi-Protocol Label Switching) VPN (Virtual Private Network) systems is Virtual Routing and Forwarding (VRF). In order to enable the coexistence of numerous virtual networks on the same physical infrastructure, VRF is used to build segregated routing instances within routers. Every VRF has its own routing table, which allows traffic to be divided and segmented amongst many VPNs. Let's examine how VRF is used in MPLS VPN and cite some pertinent research as we can see in the figure 1.

1. Routing table isolation using VRF in MPLS VPN

On a single router, numerous routing tables can be created using VRF. Each client or VPN in the context of MPLS VPN is assigned a unique VRF, which guarantees that the routing data for one VPN is kept apart from that of other VPNs.

2. Deal with Segmentation of Space:

Address space segmentation is made possible via VRF. Since the VRF context isolates these addresses within each VPN, avoiding conflicts, different VPNs can use overlapping IP addresses.

3. Independence of the Customer:

With a unique VRF, every VPN user is independent and isolated. This guarantees the privacy of the network architecture and addressing scheme of the customer and prevents it from interfering with other VPNs using the same physical infrastructure.

4. Privacy and Security:

By maintaining the privacy of routing information within each VPN, VRF improves security. By preventing unintentional or malicious interference across VPNs, this isolation improves the MPLS network's overall security.

5. Multiple VPN Support:

A router with VRF may support several VPNs at once. This is crucial for service providers, as they must provide VPN services to multiple customers while maintaining logical separation.

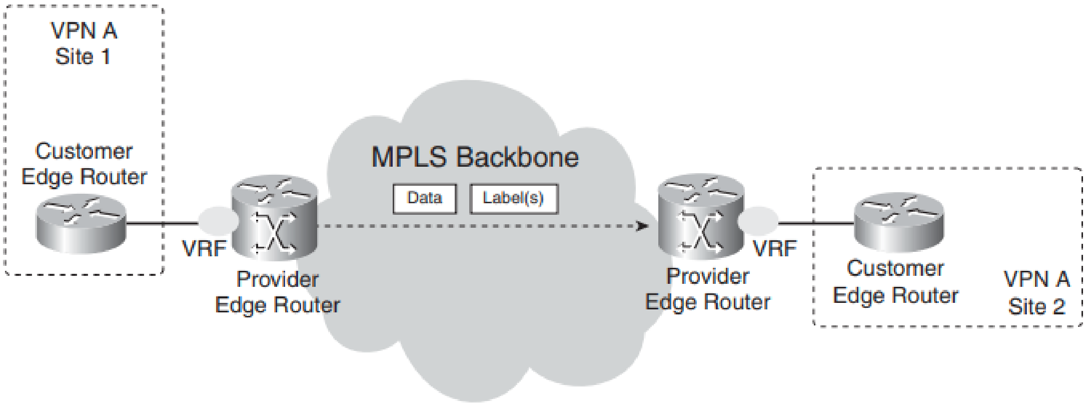


Fig 3. Understanding of VRF

The illustration represents a conceptual idea of how Multiprotocol Label Switching (MPLS) technology is used to set up an MPLS VPN, or Virtual Private Network. An MPLS VPN allows safe and isolated communication between isolated networks using a shared public network, such as the Internet.

As indicated in the illustration, VPN A consists of two client sites, Site 1 and Site 2, which are linked to the provider's network via client Edge (CE) routers. The provider's network is made up of Provider Edge (PE) routers, which communicate with the CE routers and serve as gateways to the public network.

To construct an MPLS VPN, each PE router creates a distinct Virtual Routing and Forwarding (VRF) table. The VRF table contains a mapping between IP addresses and labels, which are identifiers used to efficiently forward packets over the MPLS network.

When a packet arrives at a CE router, it is enveloped with an MPLS label that identifies its destination within the VPN. The CE router then transmits the packet to the matching PE router, which eliminates the label and uses the VRF database to find the next hop to the destination site.

PE routers in the provider's network use the Multiprotocol Border Gateway Protocol (MBGP) to exchange routing information. MBGP enables PE routers to advertise VPN routes to one another, allowing them to set up MPLS tunnels or Label Switched Paths (LSPs) to efficiently forward traffic between VPN sites.

4 Research Methodology

(Smith, 2003)The MPLS architecture is intricately designed to facilitate efficient packet handling, traffic management, and VPN services. Key components include Label Switch Routers (LSRs), responsible for forwarding decisions, and Label Edge Routers (LERs), which assign labels to packets. Protocols like Label Distribution Protocol (LDP) and Resource Reservation Protocol (RSVP) contribute to label distribution and traffic engineering. The architecture also incorporates elements like Forwarding Equivalence Class (FEC), Label Information Base (LIB), and Label Switching Database (LSDB) to organize and store forwarding information. Additionally, MPLS VPNs leverage Virtual Routing and Forwarding (VRF) instances for isolating routing tables, enabling the creation of multiple virtual private networks over a shared MPLS infrastructure. For further exploration, key papers include RFC 3031 for MPLS architecture and RFC 4364 for MPLS VPNs.

The architecture of Multi-Protocol Label Switching (MPLS) is a sophisticated and layered structure designed to facilitate efficient packet forwarding, traffic engineering, and the provision of Virtual Private Network (VPN) services. The following provides a detailed overview of the MPLS architecture, and suggested key papers for further exploration:

MPLS Architecture Overview:

Label Switch Routers (LSRs): Core routers in an MPLS network responsible for forwarding decisions based on labels. They construct the Label Information Base (LIB) to store label forwarding information.
Types include Ingress LSRs and Egress LSRs.

Label Edge Routers (LERs): Positioned at the ingress of an MPLS network, LERs assign labels to packets and encapsulate them with labels before entering the MPLS infrastructure.

Label Distribution Protocol (LDP): Facilitates label distribution among LSRs, establishing neighbour relationships and exchanging label binding information to construct label forwarding tables.

Resource Reservation Protocol (RSVP): Used in conjunction with MPLS for traffic engineering, enabling the explicit setup of routes for specific traffic flows to enhance network utilization.

Forwarding Equivalence Class (FEC): Represents a group of IP packets sharing similar characteristics, treated uniformly by the network. MPLS assigns labels to each FEC.

Label Information Base (LIB): Stored in LSRs, the LIB is a table containing label forwarding information, associating incoming labels with outgoing interfaces and labels.

Label Switching Database (LSDB): Maintained by LSRs, the LSDB is a database storing information about label bindings and network topology.
MPLS VPNs:

Enable the establishment of multiple virtual private networks over a shared MPLS infrastructure. Involves the use of Virtual Routing and Forwarding (VRF) instances to isolate routing tables for different VPNs.

Route Distinguisher (RD):

In (Route Distinguishers and Route Targets, 2013) MPLS VPNs, where IP address space is reused among isolated routing domains, route distinguishers play a crucial role. The primary objective is to keep track of routes belonging to different customers or VRFs, especially when overlapping IP address spaces are used. An RD is a unique number appended to each route within a VRF, identifying it as part of a specific customer or VRF. It is carried along with a route via MP-BGP (Multiprotocol BGP) when exchanging VPN routes with other PE (Provider Edge) routers. The RD consists of three fields: type, administrator, and value. Three defined formats can be used by a provider, and the choice of format is flexible for administrative purposes.

Route Targets (RT):
While route distinguishers maintain uniqueness among identical routes in different VRFs, route targets are employed to share routes among them. Route targets are in the form of an extended BGP community, resembling the structure of route distinguishers. Multiple route targets can be attached to routes within a VRF, controlling the import and export of routes between different VRFs. The route-target export command is used in VRF configuration to specify the route target for exported routes. For practical purposes, both route-target export and route-target import statements are often used together, allowing VRFs to learn about routes within each other.

Use of BGP:

(Tao, Chen, Liu, & She, 2023)BGP, or Border Gateway technology, is a routing information distribution technology that evaluates communication qualifications based on Multiprotocol extension and common features. The technique, which is specifically designed for Virtual Private Network (VPN) environments, works by permitting only particular Provider Edge Routers (PEs) linked to the VPN nodes to get the VPN's updated Forwarding Information Base (FIB).
Preassigned logic ports connected with distinct VPNs are used to differentiate VPN members. This proactive distribution of logic ports acts as a tool for categorizing and distinguishing VPN users. As a result, anyone attempting to get access to the Intranet or Extranet must give precise VPNJD (VPN Joining Details) as well as the necessary physical or logical port information.

5 Design Specification

(Jeyakumar, 2016) A careful evaluation of numerous elements is required while designing an MPLS VPN to ensure scalability, security, and efficient traffic routing. A high-level design definition for MPLS VPN is provided below:

1. **Network Topology**

A. Provider Edge (PE) Routers: Use PE routers to connect to client locations at the network's edge. Label assignment, customer routing, and VPN connectivity are all handled by PE routers.

B. Provider (P) Routers: These are internal routers that allow label switching within the MPLS network. P routers are not involved in VPN routing.
C. Customer Edge (CE) Routers: Customer-owned routers linked to PE routers that serve as the boundary between the customer and service provider networks.

D. Core MPLS Network: The internal MPLS infrastructure that connects PE and P routers and is in charge of label switching and forwarding.

2. **Routing protocol selection**

A. OSPF Process Configuration: Set up OSPF processes on PE routers so that they can participate in OSPF routing. OSPF router IDs and regions must be defined.

B. VRF OSPF Instances: On PE routers, create OSPF instances for each VRF. Give each VRF's OSPF instance a unique process ID.

C. Redistribution: To simplify communication between the MPLS VPN and OSPF domains, redistribute routes between OSPF and BGP. Control route redistribution carefully to avoid routing loops.

D. MPLS Configuration

A. MP BGP (Multiprotocol BGP): Configure MP BGP for VPN route exchange between Provide Edge routers, including RD and RT propagation.

B. IPsec Integration: Add an extra layer of security by integrating IPsec, which encrypts data in transit between customer sites.

E. VRF Configuration

A. VRF Creation: For each client, create a distinct VRF to provide dedicated routing instances and assure isolation.

B. Route Distinguisher (RD): Assign a distinct RD to each VRF in order to differentiate routes belonging to different customers. Make use of a mix of type, administrator, and value fields.

C. Route Target (RT): Use route targets to regulate the import and export of routes between VRFs, allowing for controlled customer communication.

6 Implementation

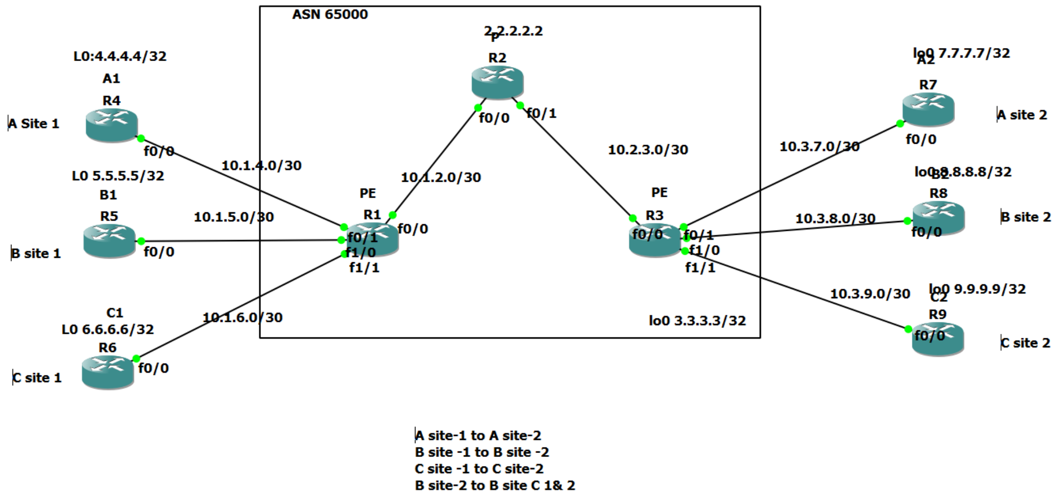


Fig 4. Network Topology for VPN sites

1. Network Topology:

We are using network topology show in fig which consist of 2 PE router on each router and 3 CE router each side which will act as different customer locations or different sites.

As we can see we have R2 router which will act as backbone of provide network in which provider will only forward the packet to the other site. It does not have any information stored in his database.

after that we have 2 provider edge routers R1 and R3 on which the MPLS VPN is formed and label will add on these routers. Then we have 3 sites on each site which act as company environments were the data is getting originated.

2. GNS 3 Setup.

GNS3 Configuration:

- GNS3 should be installed on your PC. Ascertain that the necessary Cisco IOS images are accessible for routers. Router images can be found on Marketplace of GNS3 (GNS3 Windows Install)

  https://www.gns3.com/software

- Create a GNS3 project and drag routers into it.

- Create the Topology and start the devices

3. IP addressing

Network Layer Routing: In MPLS networks, IP addresses are used for routing at the Network Layer (Layer 3). Each site or router in the MPLS architecture is given an IP address, which makes it easier to determine how MPLS-labelled packets are passed from one router to the next.Label Switching and Forwarding: MPLS use labels for efficient packet forwarding within the network, but IP addresses are critical in establishing the packet's ultimate destination. Labels are used for quick switching and forwarding, and when a packet arrives at its destination, the IP headers are checked for additional processing. (Padole, 2017)(Padole, 2017)

IP addresses are essential for end-to-end communication between devices at various MPLS locations. MPLS network routers rely on IP routing tables to determine the

Router	Interface	IP address	Subnet
R1	F0/0	10.1.2.1	255.255.255.0
	F0/1	10.1.4.2	255.255.255.0
	F1/0	10.1.5.2	255.255.255.0
	F1/1	10.1.6.2	255.255.255.0
R2	F0/0	10.1.2.2	255.255.255.0
	F0/1	10.2.3.1	255.255.255.0
R3	F0/0	10.2.3.2	255.255.255.0
	F0/1	10.3.7.1	255.255.255.0
	F1/0	10.3.8.1	255.255.255.0
	F1/1	10.3.9.1	255.255.255.0
R4	F0/0	10.1.4.2	255.255.255.0
R5	F0/0	10.1.5.1	255.255.255.0
R6	F0/0	10.1.6.1	255.255.255.0
R7	F0/0	10.3.7.2	255.255.255.0
R8	F0/0	10.3.8.2	255.255.255.0
R9	F0/0	10.3.9.2	255.255.255.0

4. System Requirements Analysis:

This analysis was conducted to determine what operational needs are needed in making VPN computer network simulations with Routing Protocol on Multiprotocol Label Switching (MPLS) networks covering hardware, software requirement

a. Hardware
- NoteBook with Intel Core i5 12th Gen
- 16 Gb Random Access Memory (RAM) capacity
- Hard drive with a capacity of 512 GB

b. Software
- Microsoft Windows 10
- GNS3 network simulation
- Lucid Chart for Network Topology
- Virtual Machines (Virtual Box, Qemu Emulator)
- CISCO ISO Program

5. Wireshark

Wireshark is a famous open-source network protocol analyzer that lets you capture and inspect data as it travels over a network in real time. It is frequently used in network administration, analysis, software development, and education.

6. Router Configuration.

Will start with Basic Interface configuration for all router. (Cisco IOS Software Configuration Guide,)

R1

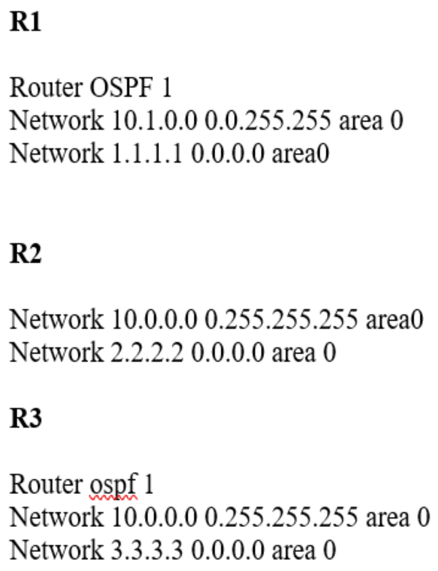
```
R1#config t (once we are logged into device will to config t mode to edit config)

Enter configuration commands, one per line. End with CNTL/Z.

R1(config)#int f0/0       (will select the interface on which we are assigning the ip address)
R1(config-if)#ip add 10.1.2.1 255.255.255.252  (Assign this IP address to the interface)
R1(config-if)#no shut


R1(config-if)#int f0/1
R1(config-if)#ip add 10.1.4.2 255.255.255.252 (Assign this IP address to the interface)
R1(config-if)#no shut


R1(config-if)#int f1/0
R1(config-if)#ip add 10.1.5.2 255.255.255.252 (Assign this IP address to the interface)
R1(config-if)#no shut
```

```
R1(config-if)#int f1/1
R1(config-if)#ip add 10.1.6.2 255.255.255.252 (Assign this IP address to the interface)
R1(config-if)#no shut
```

```
R1(config-if)#int lo0
R1(config-if)#ip add 1.1.1.1 255.255.255.255 (Assign this IP address to the interface)
```

Will follow same procedure to all the Routers.

- **OSPF configuration on edge routers (CISCO, 2016)**



R1

```
Router OSPF 1
Network 10.1.0.0 0.0.255.255 area 0
Network 1.1.1.1 0.0.0.0 area0
```

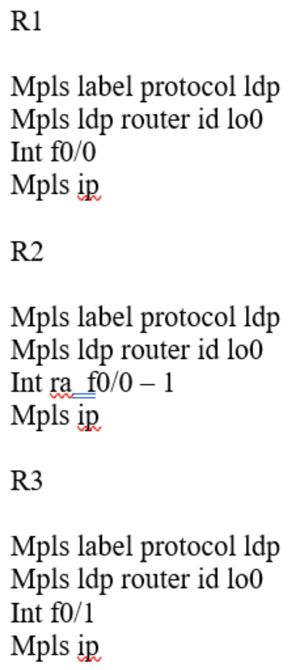
R2

```
Network 10.0.0.0 0.255.255.255 area0
Network 2.2.2.2 0.0.0.0 area 0
```

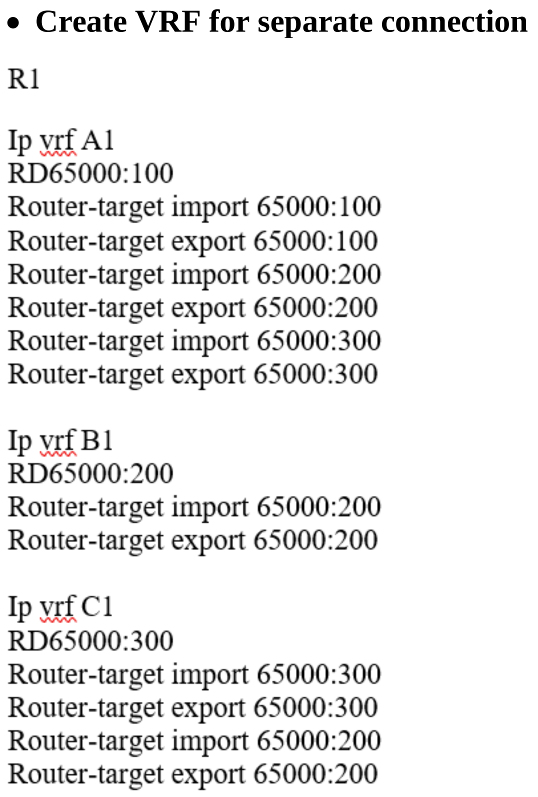
R3

```
Router ospf 1
Network 10.0.0.0 0.255.255.255 area 0
Network 3.3.3.3 0.0.0.0 area 0
```

- **MPLS configuration of edge routers   (MPLS Basic MPLS Configuration Guide, 2016)**

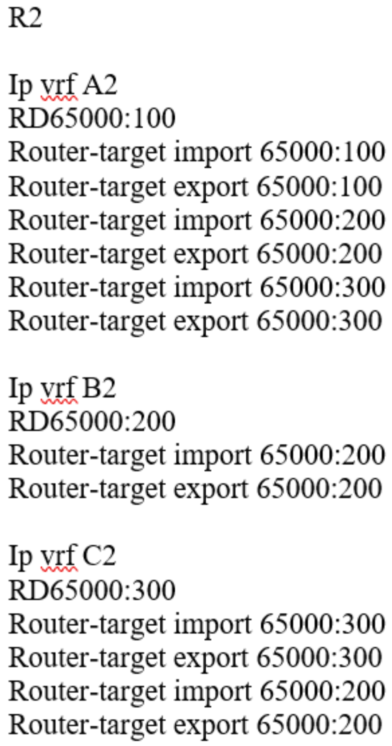


```
R1

Mpls label protocol ldp
Mpls ldp router id lo0
Int f0/0
Mpls ip

R2

Mpls label protocol ldp
Mpls ldp router id lo0
Int ra  f0/0 – 1
Mpls ip

R3

Mpls label protocol ldp
Mpls ldp router id lo0
Int f0/1
Mpls ip
```



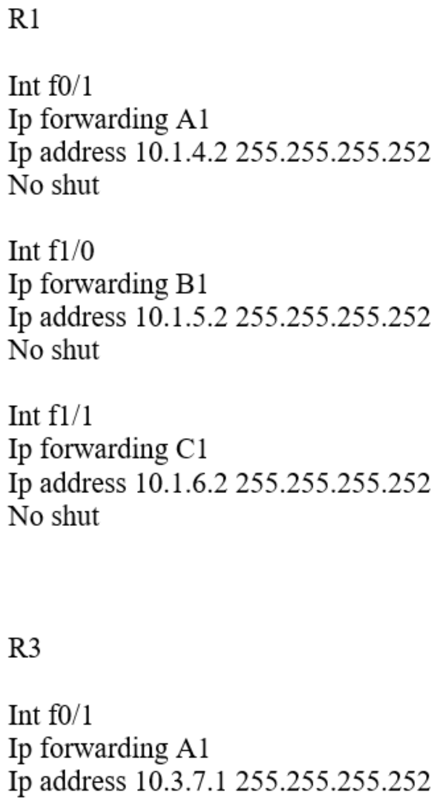
- **Create VRF for separate connection**

```
R1

Ip vrf A1
RD65000:100
Router-target import 65000:100
Router-target export 65000:100
Router-target import 65000:200
Router-target export 65000:200
Router-target import 65000:300
Router-target export 65000:300

Ip vrf B1
RD65000:200
Router-target import 65000:200
Router-target export 65000:200

Ip vrf C1
RD65000:300
Router-target import 65000:300
Router-target export 65000:300
Router-target import 65000:200
Router-target export 65000:200
```

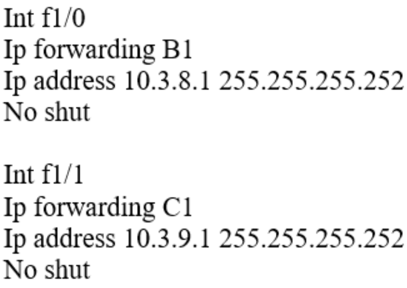


```
R2

Ip vrf A2
RD65000:100
Router-target import 65000:100
Router-target export 65000:100
Router-target import 65000:200
Router-target export 65000:200
Router-target import 65000:300
Router-target export 65000:300

Ip vrf B2
RD65000:200
Router-target import 65000:200
Router-target export 65000:200

Ip vrf C2
RD65000:300
Router-target import 65000:300
Router-target export 65000:300
Router-target import 65000:200
Router-target export 65000:200
```

- **VRF Forwarding config**

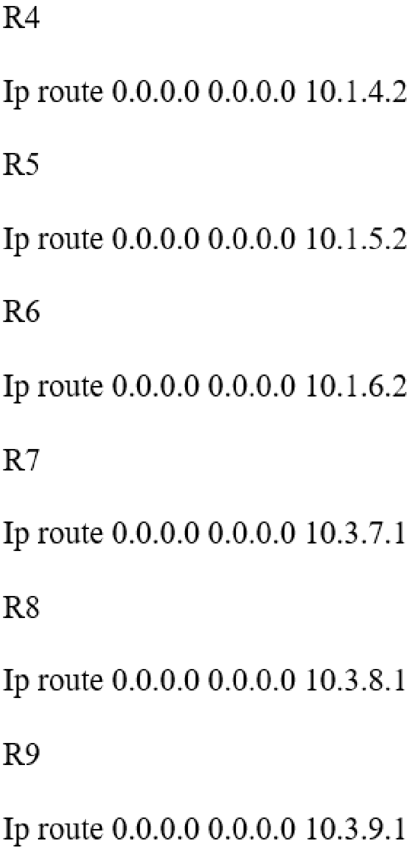


```
R1

Int f0/1
Ip forwarding A1
Ip address 10.1.4.2 255.255.255.252
No shut

Int f1/0
Ip forwarding B1
Ip address 10.1.5.2 255.255.255.252
No shut

Int f1/1
Ip forwarding C1
Ip address 10.1.6.2 255.255.255.252
No shut



R3

Int f0/1
Ip forwarding A1
Ip address 10.3.7.1 255.255.255.252
```



```
Int f1/0
Ip forwarding B1
Ip address 10.3.8.1 255.255.255.252
No shut

Int f1/1
Ip forwarding C1
Ip address 10.3.9.1 255.255.255.252
No shut
```

- **Static default Route toward PE routers**



```
R4

Ip route 0.0.0.0 0.0.0.0 10.1.4.2

R5

Ip route 0.0.0.0 0.0.0.0 10.1.5.2

R6

Ip route 0.0.0.0 0.0.0.0 10.1.6.2

R7

Ip route 0.0.0.0 0.0.0.0 10.3.7.1

R8

Ip route 0.0.0.0 0.0.0.0 10.3.8.1

R9

Ip route 0.0.0.0 0.0.0.0 10.3.9.1
```

R1

```
Ip route vrf A1 4.4.4.4 255.255.255.255 10.1.4.1
Ip route vrf B1 5.5.5.5 255.255.255.255 10.1.5.1
Ip route vrf C1 6.6.6.6 255.255.255.255 10.1.6.1
```

R2

```
Ip route vrf A2 7.7.7.7 255.255.255.255 10.3.7.2
Ip route vrf B2 8.8.8.8 255.255.255.255 10.3.8.2
Ip route vrf C2 9.9.9.9 255.255.255.255 10.3.9.2
```

- **BGP configuration (Configuring a Basic BGP Network, 2016)**

R1

```
Router BGP 65000
Neigbhour   3.3.3.3 remote-as 65000
Neigbhour 3.3.3.3 update-source lo0

Address-family vpnv4
Neigbhour 3.3.3.3 activate
Neigbhour 3.3.3.3 next hop-self
Neigbhour 3.3.3.3 send-community

Address-family ipv4 VRF A1
Redistribute static
Redistribute connected

Address-family Ipv4 vrf B1
Redistribute static
Redistribute connected

 Address-family ipv4 VRF C1
Redistribute static
Redistribute connected
```

R2

```
Router BGP 65000
Neigbhour  1.1.1.1 remote-as 65000
Neigbhour  1.1.1.1 update-source lo0

Address-family vpnv4
Neigbhour 1.1.1.1 activate
Neigbhour 1.1.1.1 next hop-self
Neigbhour 1.1.1.1 send-community

Address-family ipv4 VRF A1
Redistribute static
Redistribute connected
```

```
Address-family Ipv4 vrf B1
Redistribute static
Redistribute connected

Address-family ipv4 VRF C1
Redistribute static
Redistribute connected
```

7 Evaluation

MPLS VPNs are thoroughly examined, including their design, benefits, components, and essential deployment factors. It begins with an abstract that emphasizes the critical relevance of secure communication networks, particularly MPLS VPNs, in today's digital landscape. The abstract emphasizes MPLS VPNs' security and isolation capabilities, combining the benefits of traditional VPNs with the scalability and adaptability of MPLS.

The introductory portion looks into the expanding network security landscape, emphasizing the growing importance of robust network security in the context of modern issues. It introduces MPLS as a basic technology, explaining its essential concepts and advantages, particularly in comparison to traditional routing and switching.

The following sections provide a thorough overview of MPLS VPNs, including their origin, structure, variants, and benefits. It goes into detail about how MPLS VPNs help with traffic segmentation, Quality of Service (QoS) control, scalability, and flexibility. The abstract also mentions MPLS VPNs' practical applications, such as in enterprise networks and for service providers.

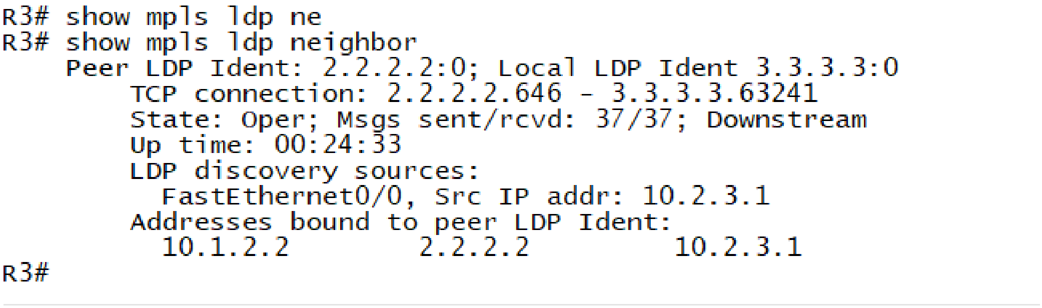
The paper next introduces terms such as route distinguisher (RD) and route target (RT) within MPLS VPNs, explaining their functions in preserving distinctness and promoting route sharing among Virtual Routing and Forwarding (VRF) instances. It backs up these claims with references to research publications and writers such as Haeryong Lee, EZE, Ghein, and Tao, Chen, Liu, and She.

The design specification section provides an overview of MPLS VPN design, including network topology, routing protocol selection, MPLS configuration, and VRF configuration. The following part introduces the GNS3 setup and configuration requirements, highlighting the importance of both hardware and software components in the effective simulation of MPLS VPNs. Although there is no dedicated evaluation section in the offered literature, it comprehensively addresses the underlying principles, components, and implementation issues connected to MPLS VPNs, giving an all-encompassing reference for understanding and installing this technology.

7.1 Experiment 1 Verification of Neighborships

Once the configuration is in place, we are going the try to check whether it’s working properly or not for that we are going to try multiple experiments for it.

The output presented here is from the command show mpls ldp bindings on a router (R3). The following command displays the MPLS Label Distribution Protocol (LDP) bindings on the router, demonstrating how labels for different prefixes are assigned and distributed. The following is an interpretation of the output:

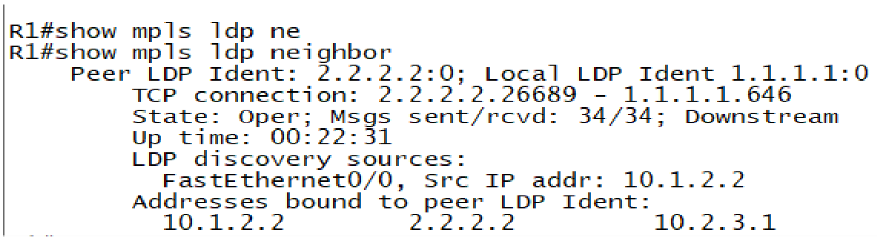


```
R3# show mpls ldp ne
R3# show mpls ldp neighbor
    Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 3.3.3.3:0
        TCP connection: 2.2.2.2.646 - 3.3.3.3.63241
        State: Oper; Msgs sent/rcvd: 37/37; Downstream
        Up time: 00:24:33
        LDP discovery sources:
          FastEthernet0/0, Src IP addr: 10.2.3.1
        Addresses bound to peer LDP Ident:
          10.1.2.2        2.2.2.2         10.2.3.1
R3#
```

Fig 5. MPLS neighbor details

Each "lib entry" represents an IP prefix, and the "local binding" shows the label assigned to that prefix by the local router. The "remote binding" displays the label assigned by the remote LSR (2.2.2.2) to the same prefix. The label values as well as the existence of "Implicit Null" show how MPLS labels are used for forwarding.

In the second entry (2.2.2.2/32), for example, the local router (R3) assigns Label 16 to this prefix whereas the remote router (LSR 2.2.2.2) assigns an Implicit Null label. When traffic is routed to the remote router, the label is replaced with an implied null label.

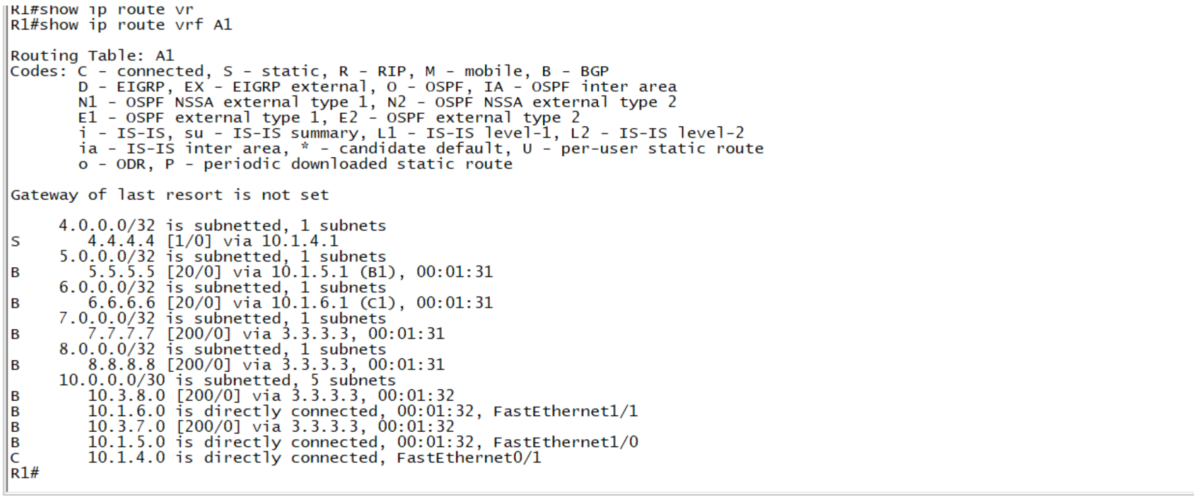


```
R1#show mpls ldp ne
R1#show mpls ldp neighbor
    Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 1.1.1.1:0
        TCP connection: 2.2.2.2.26689 - 1.1.1.1.646
        State: Oper; Msgs sent/rcvd: 34/34; Downstream
        Up time: 00:22:31
        LDP discovery sources:
          FastEthernet0/0, Src IP addr: 10.1.2.2
        Addresses bound to peer LDP Ident:
          10.1.2.2        2.2.2.2         10.2.3.1
```

Fig 6 MPLS LDP neighbor

Similarly, to the previous explanation, each "lib entry" represents a specific IP prefix, and the "local binding" shows the label assigned to that prefix by the local router. The "remote binding" displays the label assigned by the remote LSR (2.2.2.2) to the same prefix.

In the third entry (3.3.3.3/32), for example, the local router (R1) assigns Label 16 to this prefix, and the remote router (LSR 2.2.2.2) assigns Label 16. This means that the label is replaced with Label 16 when traffic is sent to the remote router.



```
R1#show ip route vr
R1#show ip route vrf A1

Routing Table: A1
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     4.0.0.0/32 is subnetted, 1 subnets
S       4.4.4.4 [1/0] via 10.1.4.1
     5.0.0.0/32 is subnetted, 1 subnets
B       5.5.5.5 [20/0] via 10.1.5.1 (B1), 00:01:31
     6.0.0.0/32 is subnetted, 1 subnets
B       6.6.6.6 [20/0] via 10.1.6.1 (C1), 00:01:31
     7.0.0.0/32 is subnetted, 1 subnets
B       7.7.7.7 [200/0] via 3.3.3.3, 00:01:31
     8.0.0.0/32 is subnetted, 1 subnets
B       8.8.8.8 [200/0] via 3.3.3.3, 00:01:31
     10.0.0.0/30 is subnetted, 5 subnets
B       10.3.8.0 [200/0] via 3.3.3.3, 00:01:32
B       10.1.6.0 is directly connected, 00:01:32, FastEthernet1/1
B       10.3.7.0 [200/0] via 3.3.3.3, 00:01:32
B       10.1.5.0 is directly connected, 00:01:32, FastEthernet1/0
C       10.1.4.0 is directly connected, FastEthernet0/1
R1#
```

Fig 7 VRF A1 details

7.2 Experiment 2 Verification VRF of the Routes.

The output of the "show mpls forwarding-table detail" command on router R3 provides detailed information about the MPLS forwarding table. Let's break down the key information presented in fig the output:

Local Label 16:

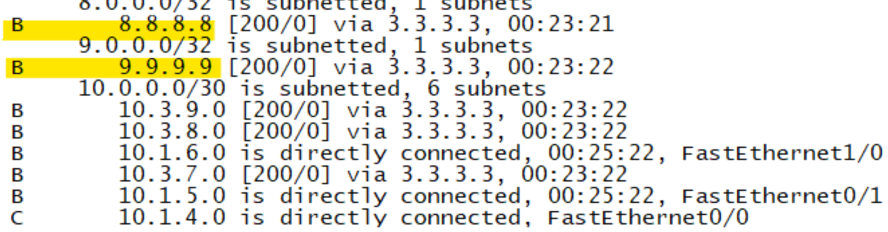
Outgoing Label/VC: Pop Label
Prefix: 2.2.2.2/32
Bytes Switched: 0
Outgoing Interface: Fa0/0
Next Hop: 10.2.3.1
Label Stack: {}
Encapsulation Information: MAC/Encaps=14/14, MRU=1504
No output feature configured
Local Label 17:

Outgoing Label/VC: Pop Label
Prefix: 10.1.2.0/30
Bytes Switched: 0
Outgoing Interface: Fa0/0
Next Hop: 10.2.3.1

```
Label Stack: {}
Encapsulation Information: MAC/Encaps=14/14, MRU=1504
No output feature configured
Local Label 18:

Outgoing Label/VC: 18
Prefix: 1.1.1.1/32
Bytes Switched: 0
Outgoing Interface: Fa0/0
Next Hop: 10.2.3.1
Label Stack: {18}
Encapsulation Information: MAC/Encaps=14/18, MRU=1500
No output feature configured
Local Label 19:

Outgoing Label/VC: No Label
Prefix: 7.7.7.7/32[V]
Bytes Switched: 0
Outgoing Interface: Fa0/1
Next Hop: 10.3.7.2
Label Stack: {}
Encapsulation Information: MAC/Encaps=14/14, MRU=1504
VPN Route: A2
No output feature configured
```

This output contains information on the MPLS labels, their associated prefixes, outgoing labels or actions, the outgoing interface, next hop information, label stack, encapsulation data, VPN route information, and the state of the output feature configuration. It can help you understand how MPLS forwards traffic based on the labels in the forwarding table.

The output of the "show ip route vrf A1" command on router R1 displays the routing table for the Virtual Routing and Forwarding (VRF) instance named A1. Let's analyze the information presented:



```
      8.0.0.0/32 is subnetted, 1 subnets
B        8.8.8.8 [200/0] via 3.3.3.3, 00:23:21
      9.0.0.0/32 is subnetted, 1 subnets
B        9.9.9.9 [200/0] via 3.3.3.3, 00:23:22
      10.0.0.0/30 is subnetted, 6 subnets
B        10.3.9.0 [200/0] via 3.3.3.3, 00:23:22
B        10.3.8.0 [200/0] via 3.3.3.3, 00:23:22
B        10.1.6.0 is directly connected, 00:25:22, FastEthernet1/0
B        10.3.7.0 [200/0] via 3.3.3.3, 00:23:22
B        10.1.5.0 is directly connected, 00:25:22, FastEthernet0/1
C        10.1.4.0 is directly connected, FastEthernet0/0
```

Fig 8 Route getting learn in VRF A1

The routing table contains information on the routes in VRF A1, including their sources, next hops, metrics, and other details dependent on the routing source. Which mean that A site 1 traffic can be reach to other sites. But if we check other VRF B1, C1. In the route table show that their data traffic can be reach to only Similar routing table will there for different VRF with different route as shown in Appendix.

7.3 Experiment 3 Reachability Test

We have created a reachability scenario as follow:

A site 1 to all site
B site 1 to B site 2
C site 1 to C site 2
B site 1 to B site 2 and C site 1 & 2
A site 1 to A site 2

To explain reachability scenarios using ping and traceroute between the mentioned sites (A site 1, A site 2, B site 1, B site 2, C site 1, and C site 2), we'll assume that these sites are interconnected through a network infrastructure.

A site 1 to all sites:

Ping: Use the ping command to check the reachability of A site 1 to A site 2, B site 1, B site 2, C site 1, and C site 2. Shown in fig 9
Traceroute: Employ the traceroute command to trace the path taken by packets from A site 1 to the mentioned sites.

B site 1 to B site 2:

Ping: Execute a ping from B site 1 to B site 2 to verify the direct reachability between them.
Traceroute: Use traceroute to trace the path between B site 1 and B site 2 shown in fig 11

C site 1 to C site 2:

Ping: Perform a ping from C site 1 to C site 2 to confirm the direct reachability between these two sites.
Traceroute: Use traceroute to trace the path between C site 1 and C site 2. Shown in fig 12

B site 1 to B site 2 and C site 1 & 2:

Ping: Use ping to check the reachability of B site 1 to B site 2, C site 1, and C site 2 individually.
Traceroute: Employ traceroute to trace the paths from B site 1 to B site 2, C site 1, and C site 2.

A site 1 to A site 2:

Ping: Use ping to confirm the reachability of A site 1 to A site 2.
Traceroute: Employ traceroute to trace the path between A site 1 and A site 2. Referee fig 10 and 12
These commands will provide insights into the network path taken by packets, as well as the round-trip time and potential issues such as packet loss. It's important to note that firewalls, routing policies, and network configurations can impact the results, so consider these factors when interpreting the output of ping and traceroute commands.

To perform implementation of QOS policies I have come with some of obstacles.

While GNS3 is an extremely capable network simulation tool, it has significant limits when it comes to implementing QoS standards for MPLS VPNs. Here are some restrictions you may encounter:

1. **Platform Support is Limited:**

GNS3 primarily emulates routers using Dynamips, and the router images provided may not completely implement all MPLS and QoS capabilities. GNS3 may lack several advanced MPLS and QoS functionalities seen in commercial routers.

2. **Constraints on Resources:**

GNS3 simulations consume a lot of resources, and executing sophisticated MPLS VPN scenarios with QoS settings might tax your computer's resources. Performance concerns may arise when simulating a large-scale network with several routers and QoS setups.

3. **Protocol Support is Limited:**

GNS3 may not support all MPLS and QoS protocols and features completely. Some QoS methods or MPLS extensions may not be available or may function differently than they would on physical hardware.

4. **Real-world Traffic Characteristics Are Missing:**

GNS3 is a simulator, and the traffic generated within the simulated network may not fully mirror real-world traffic characteristics. This can have an effect on the precision of QoS testing and validation.

5. **Absence of Hardware-Specific Behavior:**

GNS3 does not precisely imitate the hardware behavior of individual routers or switches. In some circumstances, hardware-specific QoS behavior may be lacking, resulting in discrepancies in outcomes when compared to real-world deployments.

8

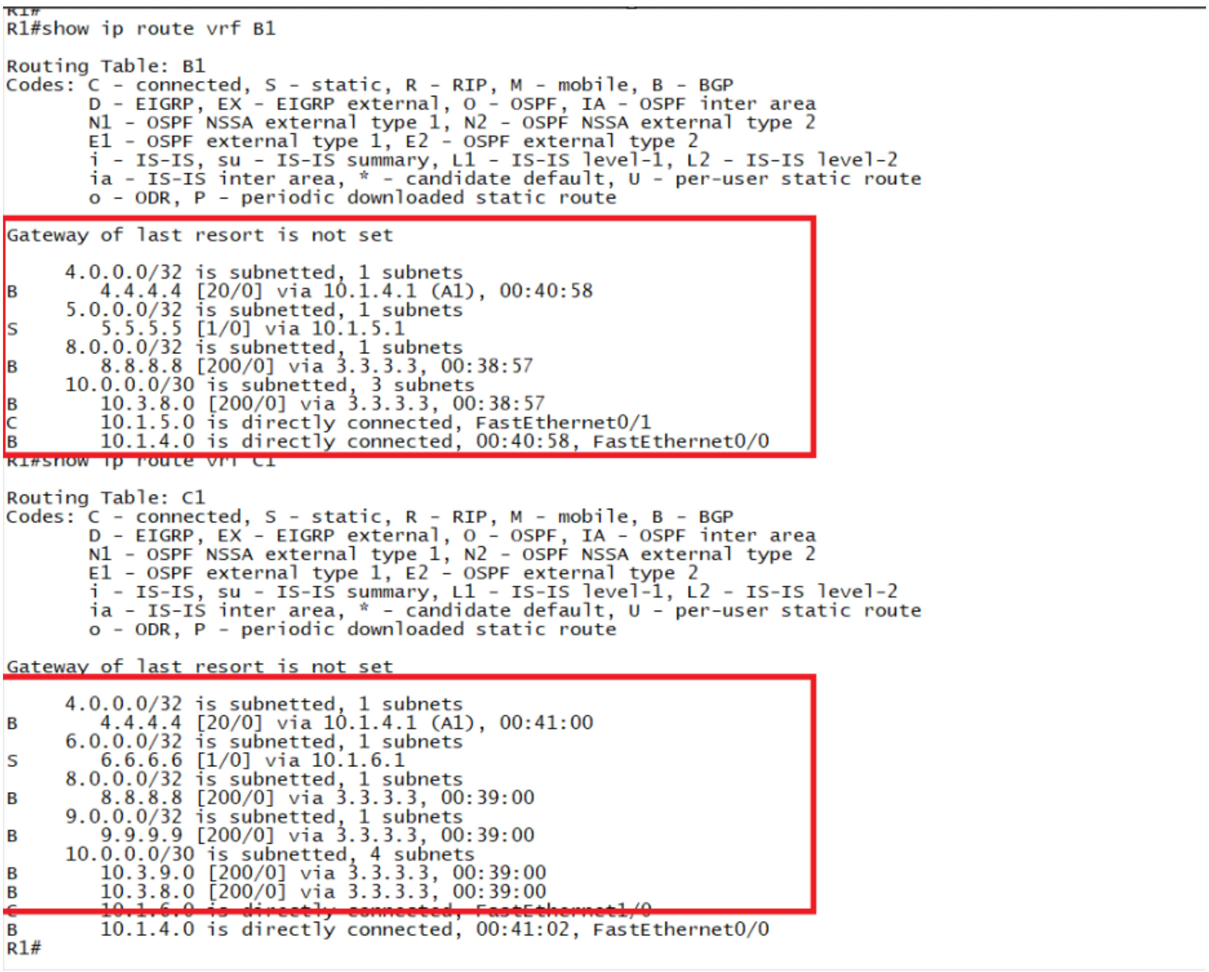
9 Conclusion and Future Work

In conclusion, the investigation of MPLS VPNs has highlighted their critical role in modern networking, combining solid security with excellent data routing. The growing importance of network security in the face of changing cyber threats necessitates the adoption of secure network design concepts. With its core concepts and benefits over traditional routing and switching, MPLS technology emerges as a critical facilitator of safe and scalable network infrastructures. The structure of MPLS VPNs, which involves the interplay of Customer Edge (CE) and Provider Edge (PE) routers, Label Distribution Protocol (LDP), and Label Switching Paths (LSPs), highlights its adaptability in establishing traffic segmentation and controlled routing. The discussion of MPLS VPN versions, including Layer 3, Layer 2, and Virtual Private LAN Service (VPLS), demonstrates MPLS's adaptability in satisfying a variety of networking requirements.

The benefits of MPLS VPNs, which include traffic segmentation, Quality of Service (QoS) control, scalability, and centralized management, highlight their significance in satisfying the complex needs of modern companies and service providers. Case studies demonstrate successful MPLS VPN implementations in a variety of contexts. The section on upcoming trends and technology forecasts the dynamic landscape of cybersecurity threats in the future. The possible integration of Software-Defined Networking (SDN) with MPLS, as well as the engagement of Artificial Intelligence (AI) and Machine Learning (ML), presents promising opportunities for improving MPLS VPN security.

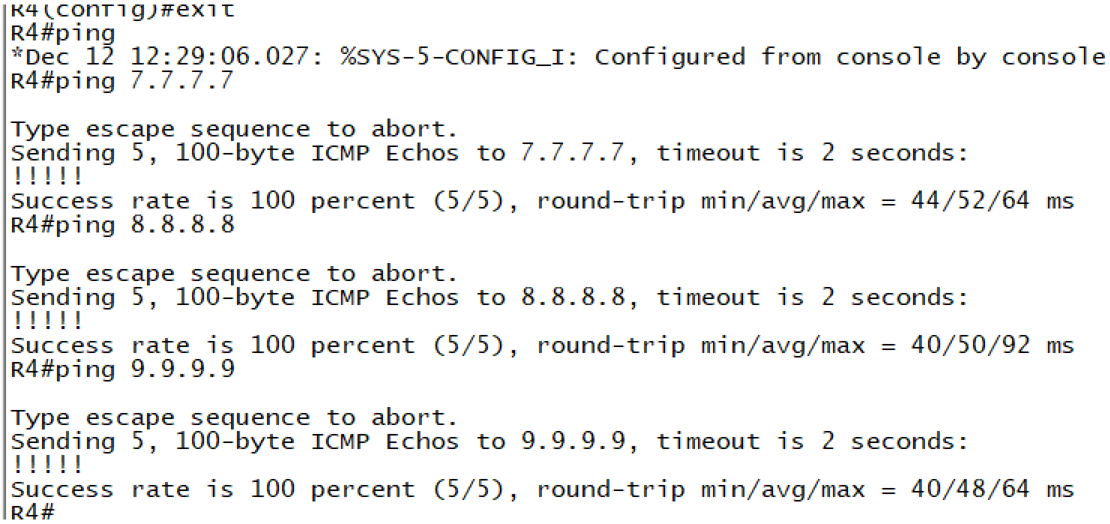
Prospective projects in this area could explore deeper into the growing threat landscape and practical applications of AI/ML in MPLS VPN security. Furthermore, continued study and adaptation of MPLS VPNs to future technologies will be critical in ensuring their effectiveness in an ever-changing digital ecosystem. MPLS VPNs continue to be a strong and expanding basis in the domain of network security as enterprises seek secure, scalable, and efficient network solutions.

10 Appendix



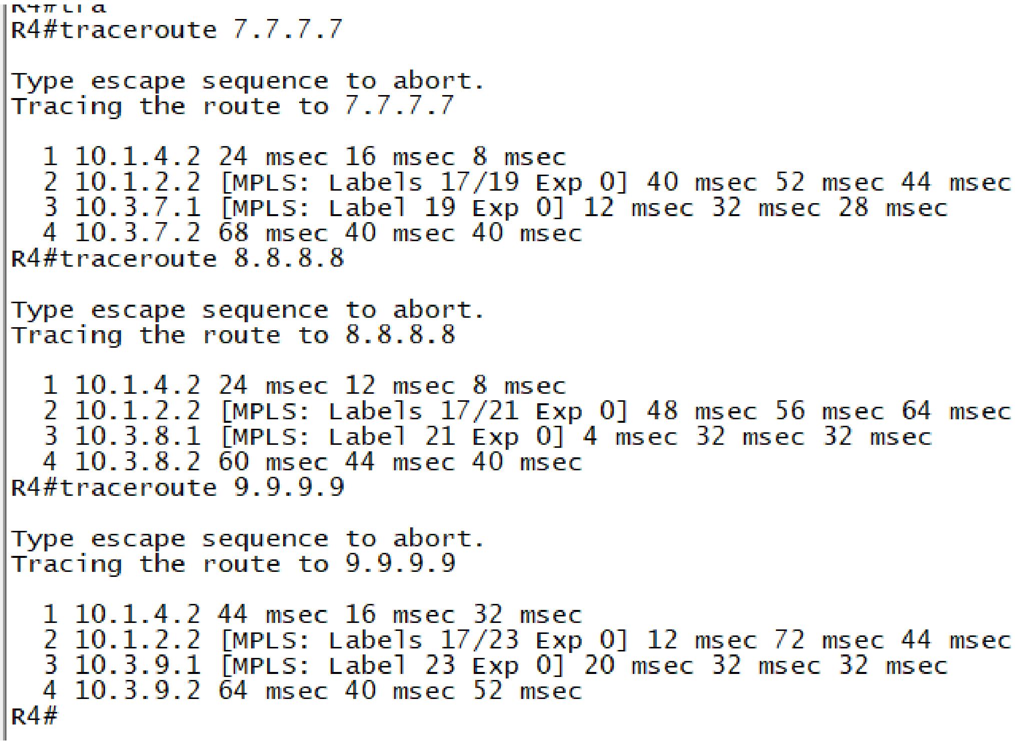
```
R1#
R1#show ip route vrf B1

Routing Table: B1
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     4.0.0.0/32 is subnetted, 1 subnets
B       4.4.4.4 [20/0] via 10.1.4.1 (A1), 00:40:58
     5.0.0.0/32 is subnetted, 1 subnets
S       5.5.5.5 [1/0] via 10.1.5.1
     8.0.0.0/32 is subnetted, 1 subnets
B       8.8.8.8 [200/0] via 3.3.3.3, 00:38:57
     10.0.0.0/30 is subnetted, 3 subnets
B       10.3.8.0 [200/0] via 3.3.3.3, 00:38:57
C       10.1.5.0 is directly connected, FastEthernet0/1
B       10.1.4.0 is directly connected, 00:40:58, FastEthernet0/0
R1#show ip route vrf C1

Routing Table: C1
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     4.0.0.0/32 is subnetted, 1 subnets
B       4.4.4.4 [20/0] via 10.1.4.1 (A1), 00:41:00
     6.0.0.0/32 is subnetted, 1 subnets
S       6.6.6.6 [1/0] via 10.1.6.1
     8.0.0.0/32 is subnetted, 1 subnets
B       8.8.8.8 [200/0] via 3.3.3.3, 00:39:00
     9.0.0.0/32 is subnetted, 1 subnets
B       9.9.9.9 [200/0] via 3.3.3.3, 00:39:00
     10.0.0.0/30 is subnetted, 4 subnets
B       10.3.9.0 [200/0] via 3.3.3.3, 00:39:00
B       10.3.8.0 [200/0] via 3.3.3.3, 00:39:00
C       10.1.6.0 is directly connected, FastEthernet1/0
B       10.1.4.0 is directly connected, 00:41:02, FastEthernet0/0
R1#
```

Fig 8 Routes getting learn in VRF B1 & C1



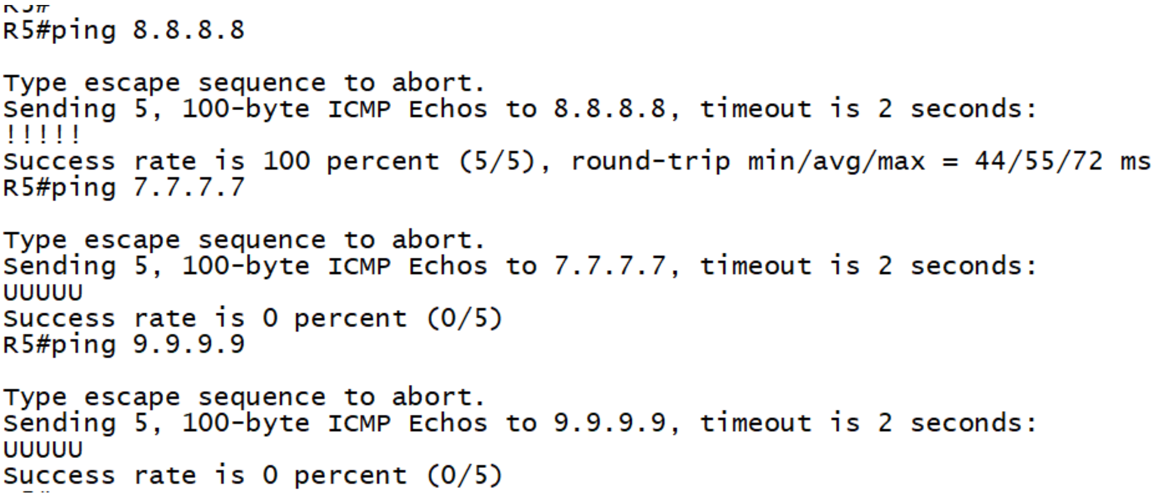
```
R4(config)#exit
R4#ping
*Dec 12 12:29:06.027: %SYS-5-CONFIG_I: Configured from console by console
R4#ping 7.7.7.7

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 7.7.7.7, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/52/64 ms
R4#ping 8.8.8.8

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.8, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/50/92 ms
R4#ping 9.9.9.9

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 9.9.9.9, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/48/64 ms
R4#
```

Fig 9 Router R4 is able to ping the A,B,C sites 2



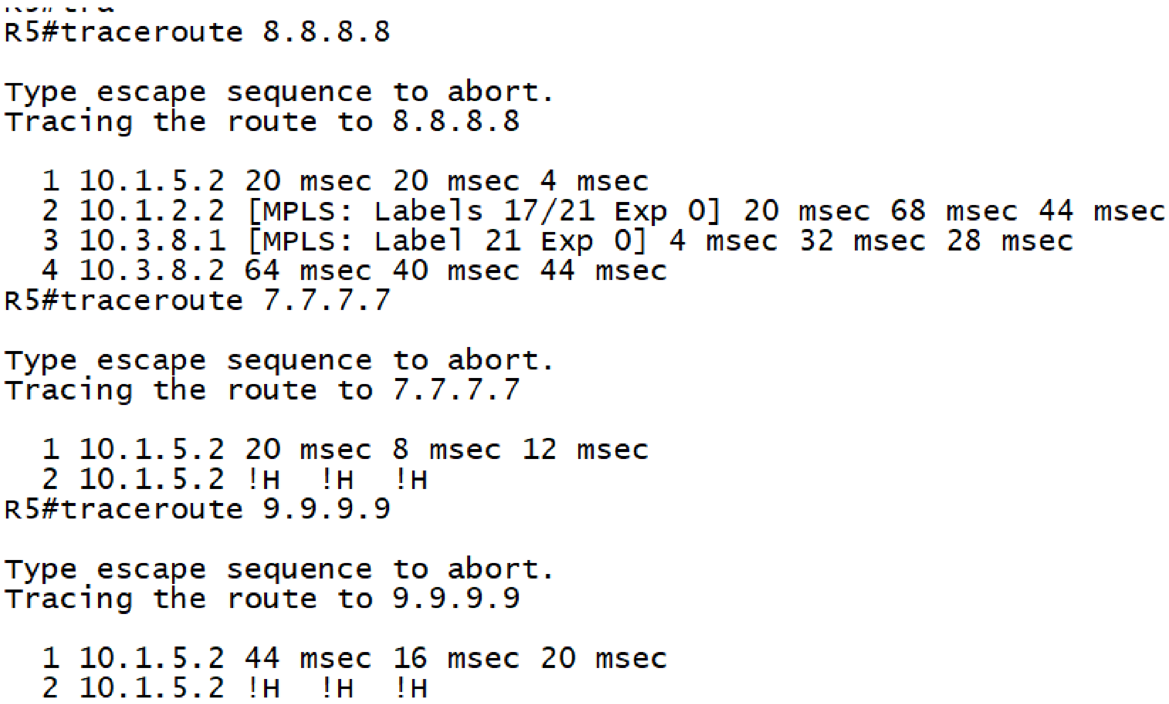
```
R4#traceroute 7.7.7.7

Type escape sequence to abort.
Tracing the route to 7.7.7.7

  1 10.1.4.2 24 msec 16 msec 8 msec
  2 10.1.2.2 [MPLS: Labels 17/19 Exp 0] 40 msec 52 msec 44 msec
  3 10.3.7.1 [MPLS: Label 19 Exp 0] 12 msec 32 msec 28 msec
  4 10.3.7.2 68 msec 40 msec 40 msec
R4#traceroute 8.8.8.8

Type escape sequence to abort.
Tracing the route to 8.8.8.8

  1 10.1.4.2 24 msec 12 msec 8 msec
  2 10.1.2.2 [MPLS: Labels 17/21 Exp 0] 48 msec 56 msec 64 msec
  3 10.3.8.1 [MPLS: Label 21 Exp 0] 4 msec 32 msec 32 msec
  4 10.3.8.2 60 msec 44 msec 40 msec
R4#traceroute 9.9.9.9

Type escape sequence to abort.
Tracing the route to 9.9.9.9

  1 10.1.4.2 44 msec 16 msec 32 msec
  2 10.1.2.2 [MPLS: Labels 17/23 Exp 0] 12 msec 72 msec 44 msec
  3 10.3.9.1 [MPLS: Label 23 Exp 0] 20 msec 32 msec 32 msec
  4 10.3.9.2 64 msec 40 msec 52 msec
R4#
```

Fig 10 Traceroute result of Site 2



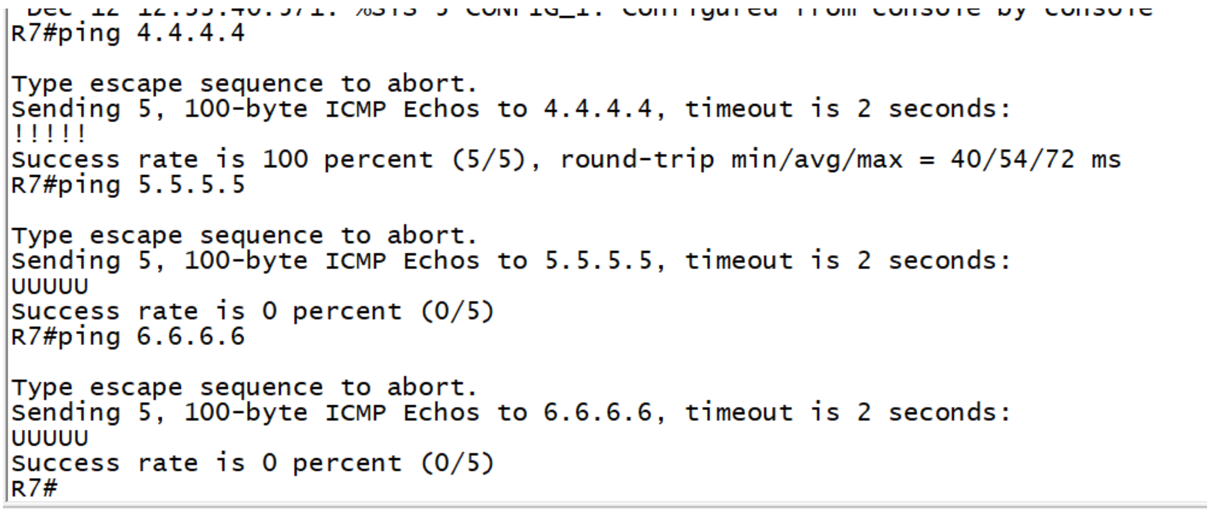
```
R5#ping 8.8.8.8

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.8, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/55/72 ms
R5#ping 7.7.7.7

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 7.7.7.7, timeout is 2 seconds:
UUUUU
Success rate is 0 percent (0/5)
R5#ping 9.9.9.9

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 9.9.9.9, timeout is 2 seconds:
UUUUU
Success rate is 0 percent (0/5)
```

Fig 11 Ping result of Router 5 that can only reach to B site 2



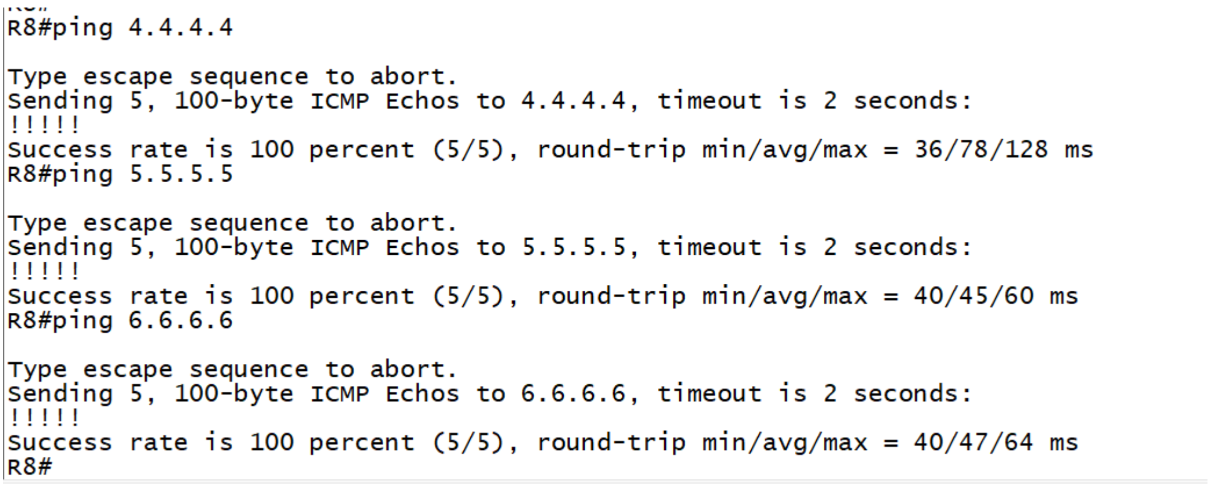
```
R5#traceroute 8.8.8.8

Type escape sequence to abort.
Tracing the route to 8.8.8.8

  1 10.1.5.2 20 msec 20 msec 4 msec
  2 10.1.2.2 [MPLS: Labels 17/21 Exp 0] 20 msec 68 msec 44 msec
  3 10.3.8.1 [MPLS: Label 21 Exp 0] 4 msec 32 msec 28 msec
  4 10.3.8.2 64 msec 40 msec 44 msec
R5#traceroute 7.7.7.7

Type escape sequence to abort.
Tracing the route to 7.7.7.7

  1 10.1.5.2 20 msec 8 msec 12 msec
  2 10.1.5.2 !H  !H  !H
R5#traceroute 9.9.9.9

Type escape sequence to abort.
Tracing the route to 9.9.9.9

  1 10.1.5.2 44 msec 16 msec 20 msec
  2 10.1.5.2 !H  !H  !H
```

Fig 12 Traceroute results



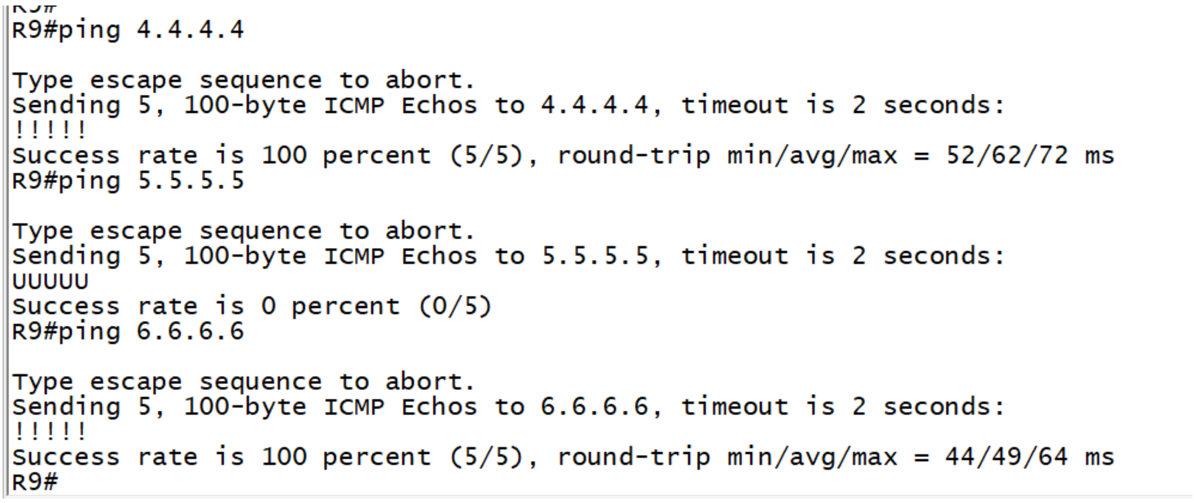
```
R7#ping 4.4.4.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/54/72 ms
R7#ping 5.5.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
UUUUU
Success rate is 0 percent (0/5)
R7#ping 6.6.6.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 6.6.6.6, timeout is 2 seconds:
UUUUU
Success rate is 0 percent (0/5)
R7#
```

Fig 13 Router 7 trying to reach site -1



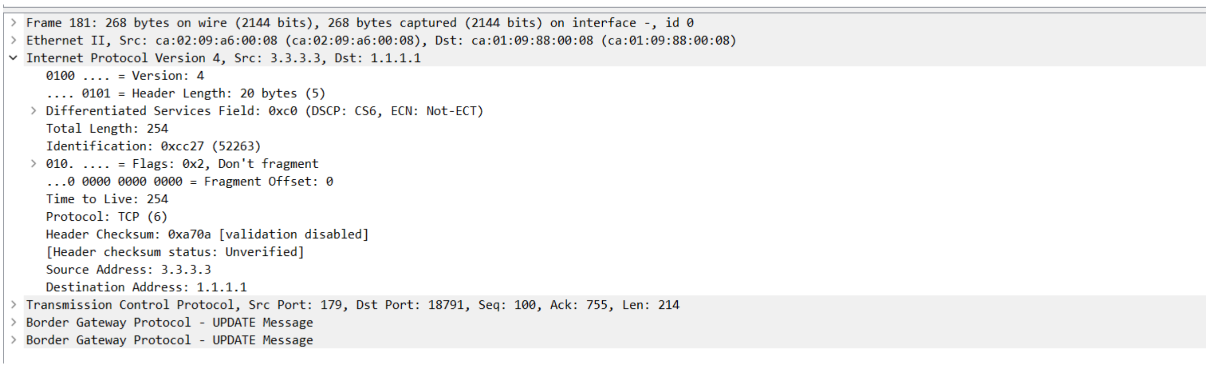
```
R8#ping 4.4.4.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/78/128 ms
R8#ping 5.5.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/45/60 ms
R8#ping 6.6.6.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 6.6.6.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/47/64 ms
R8#
```

Fig 14 : Router 8 ping results



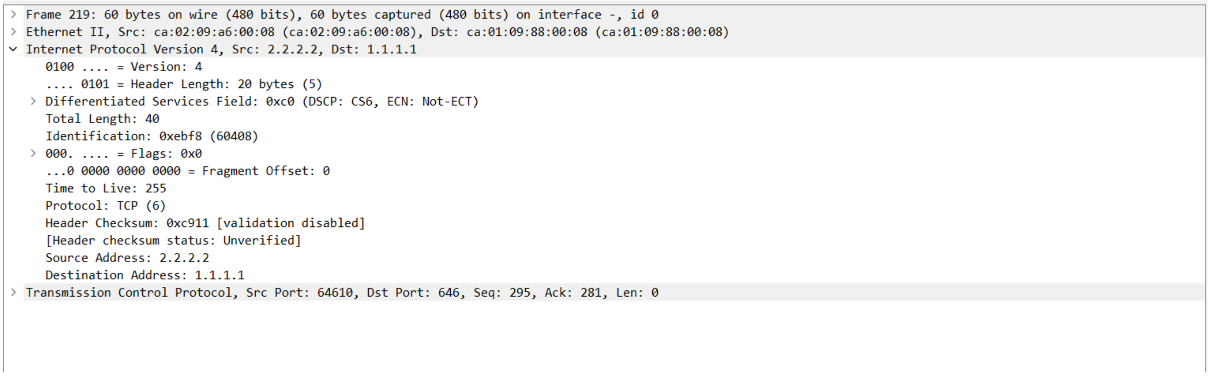
```
R9#ping 4.4.4.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/62/72 ms
R9#ping 5.5.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
UUUUU
Success rate is 0 percent (0/5)
R9#ping 6.6.6.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 6.6.6.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/49/64 ms
R9#
```

Fig 15: router 9 ping results



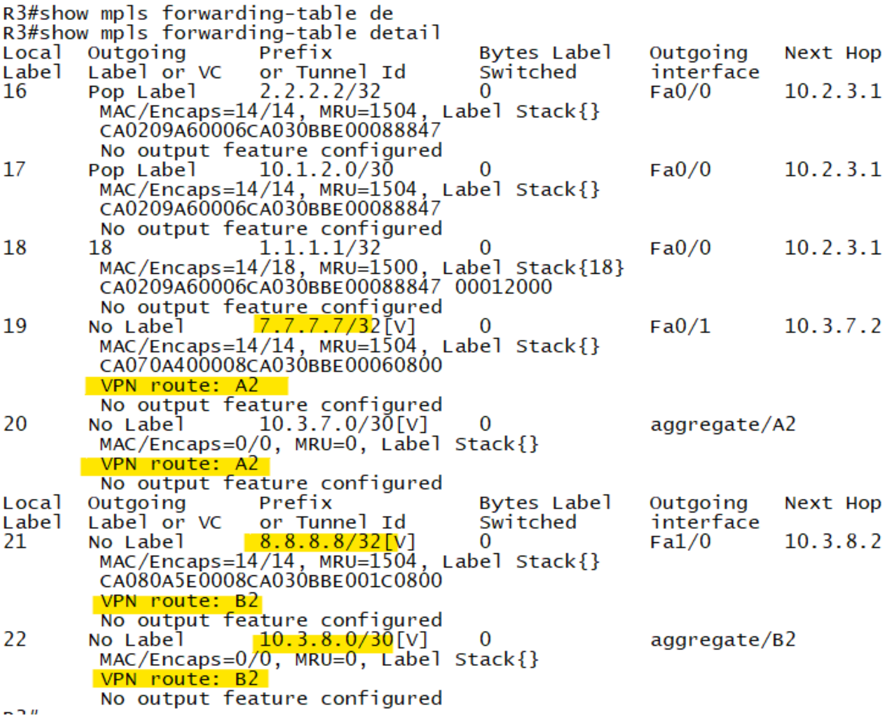
```
> Frame 181: 268 bytes on wire (2144 bits), 268 bytes captured (2144 bits) on interface -, id 0
> Ethernet II, Src: ca:02:09:a6:00:08 (ca:02:09:a6:00:08), Dst: ca:01:09:88:00:08 (ca:01:09:88:00:08)
v Internet Protocol Version 4, Src: 3.3.3.3, Dst: 1.1.1.1
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
    Total Length: 254
    Identification: 0xcc27 (52263)
  > 010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 254
    Protocol: TCP (6)
    Header Checksum: 0xa70a [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 3.3.3.3
    Destination Address: 1.1.1.1
> Transmission Control Protocol, Src Port: 179, Dst Port: 18791, Seq: 100, Ack: 755, Len: 214
> Border Gateway Protocol - UPDATE Message
> Border Gateway Protocol - UPDATE Message
```

Fig 16 Wireshark Captures will be forming the BGP neighborship Router 1



```
> Frame 219: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface -, id 0
> Ethernet II, Src: ca:02:09:a6:00:08 (ca:02:09:a6:00:08), Dst: ca:01:09:88:00:08 (ca:01:09:88:00:08)
v Internet Protocol Version 4, Src: 2.2.2.2, Dst: 1.1.1.1
     0100 .... = Version: 4
     .... 0101 = Header Length: 20 bytes (5)
   > Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
     Total Length: 40
     Identification: 0xebf8 (60408)
   > 000. .... = Flags: 0x0
     ...0 0000 0000 0000 = Fragment Offset: 0
     Time to Live: 255
     Protocol: TCP (6)
     Header Checksum: 0xc911 [validation disabled]
     [Header checksum status: Unverified]
     Source Address: 2.2.2.2
     Destination Address: 1.1.1.1
> Transmission Control Protocol, Src Port: 64610, Dst Port: 646, Seq: 295, Ack: 281, Len: 0
```

Fig 17 Wireshark Captures will be forming the BGP neighborship Router 3



```
R3#show mpls forwarding-table de
R3#show mpls forwarding-table detail
Local  Outgoing      Prefix            Bytes Label   Outgoing   Next Hop
Label  Label or VC   or Tunnel Id      Switched      interface
16     Pop Label     2.2.2.2/32        0             Fa0/0      10.2.3.1
        MAC/Encaps=14/14, MRU=1504, Label Stack{}
        CA0209A60006CA030BBE00088847
        No output feature configured
17     Pop Label     10.1.2.0/30       0             Fa0/0      10.2.3.1
        MAC/Encaps=14/14, MRU=1504, Label Stack{}
        CA0209A60006CA030BBE00088847
        No output feature configured
18     18            1.1.1.1/32        0             Fa0/0      10.2.3.1
        MAC/Encaps=14/18, MRU=1500, Label Stack{18}
        CA0209A60006CA030BBE00088847 00012000
        No output feature configured
19     No Label      7.7.7.7/32[V]     0             Fa0/1      10.3.7.2
        MAC/Encaps=14/14, MRU=1504, Label Stack{}
        CA070A400008CA030BBE00060800
        VPN route: A2
        No output feature configured
20     No Label      10.3.7.0/30[V]    0             aggregate/A2
        MAC/Encaps=0/0, MRU=0, Label Stack{}
        VPN route: A2
        No output feature configured
Local  Outgoing      Prefix            Bytes Label   Outgoing   Next Hop
Label  Label or VC   or Tunnel Id      Switched      interface
21     No Label      8.8.8.8/32[V]     0             Fa1/0      10.3.8.2
        MAC/Encaps=14/14, MRU=1504, Label Stack{}
        CA080A5E0008CA030BBE001C0800
        VPN route: B2
        No output feature configured
22     No Label      10.3.8.0/30[V]    0             aggregate/B2
        MAC/Encaps=0/0, MRU=0, Label Stack{}
        VPN route: B2
        No output feature configured
```

Fig 18 MPLS forwarding table

Router configuation

R2

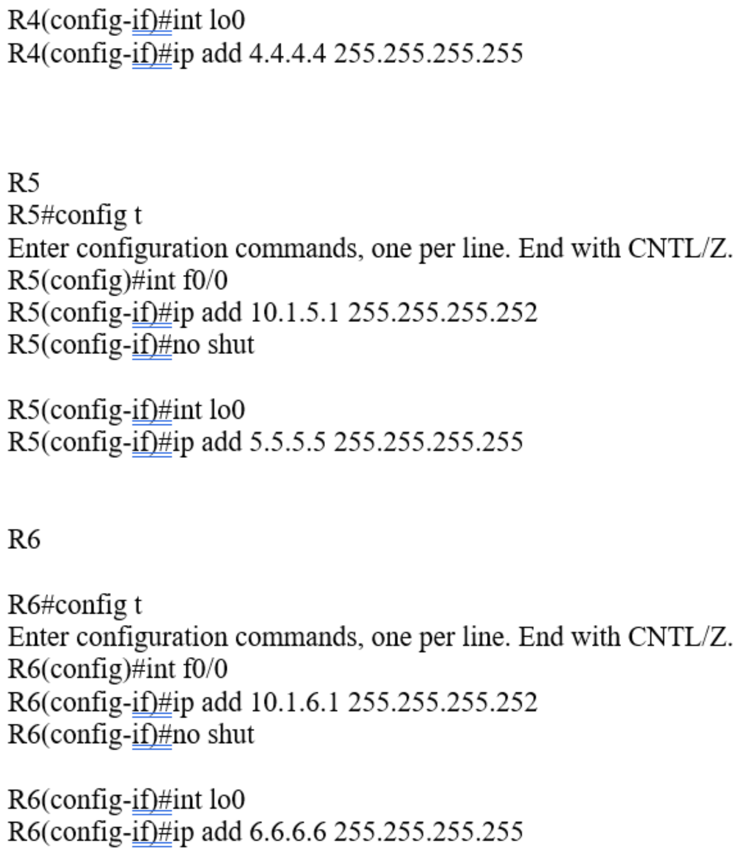
```
R2#config t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#int f0/0
R2(config-if)#ip add 10.1.2.2 255.255.255.252
R2(config-if)#no shut

R2(config-if)#int f0/1
R2(config-if)#ip add 10.2.3.1 255.255.255.252
R2(config-if)#no shut


R2(config-if)#int lo0
R2(config-if)#ip add 2.2.2.2 255.255.255.255
```

R3

```
R3#config t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#int f0/0
R3(config-if)#ip add 10.2.3.2 255.255.255.252
R3(config-if)#no shut

R3(config-if)#int f0/1
R3(config-if)#ip add 10.3.7.1 255.255.255.252
R3(config-if)#no shut

R3(config-if)#int f1/0
R3(config-if)#ip add 10.3.8.1 255.255.255.252
R3(config-if)#no shut

R3(config-if)#int f1/1
R3(config-if)#ip add 10.3.9.1 255.255.255.252
R3(config-if)#no shut

R3(config-if)#int lo0
R3(config-if)#ip add 3.3.3.3 255.255.255.255
```

R4

```
R4#config t
Enter configuration commands, one per line. End with CNTL/Z.
R4(config)#int f0/0
R4(config-if)#ip add 10.1.4.1 255.255.255.252
R4(config-if)#no shut
```



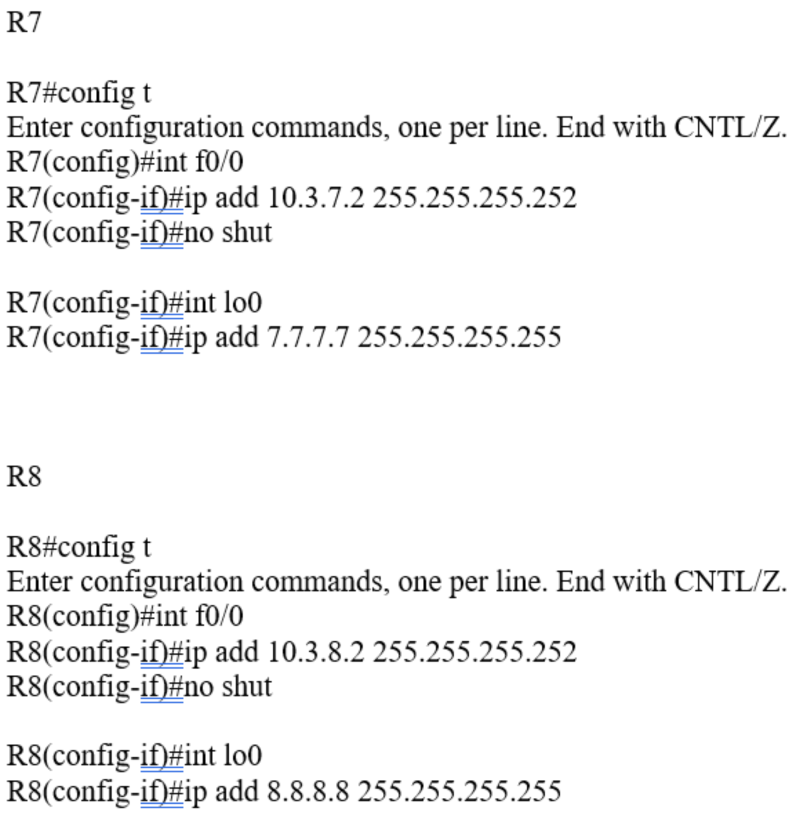
```
R4(config-if)#int lo0
R4(config-if)#ip add 4.4.4.4 255.255.255.255
```

R5

```
R5#config t
Enter configuration commands, one per line. End with CNTL/Z.
R5(config)#int f0/0
R5(config-if)#ip add 10.1.5.1 255.255.255.252
R5(config-if)#no shut

R5(config-if)#int lo0
R5(config-if)#ip add 5.5.5.5 255.255.255.255
```

R6

```
R6#config t
Enter configuration commands, one per line. End with CNTL/Z.
R6(config)#int f0/0
R6(config-if)#ip add 10.1.6.1 255.255.255.252
R6(config-if)#no shut

R6(config-if)#int lo0
R6(config-if)#ip add 6.6.6.6 255.255.255.255
```



R7

```
R7#config t
Enter configuration commands, one per line. End with CNTL/Z.
R7(config)#int f0/0
R7(config-if)#ip add 10.3.7.2 255.255.255.252
R7(config-if)#no shut

R7(config-if)#int lo0
R7(config-if)#ip add 7.7.7.7 255.255.255.255
```

R8

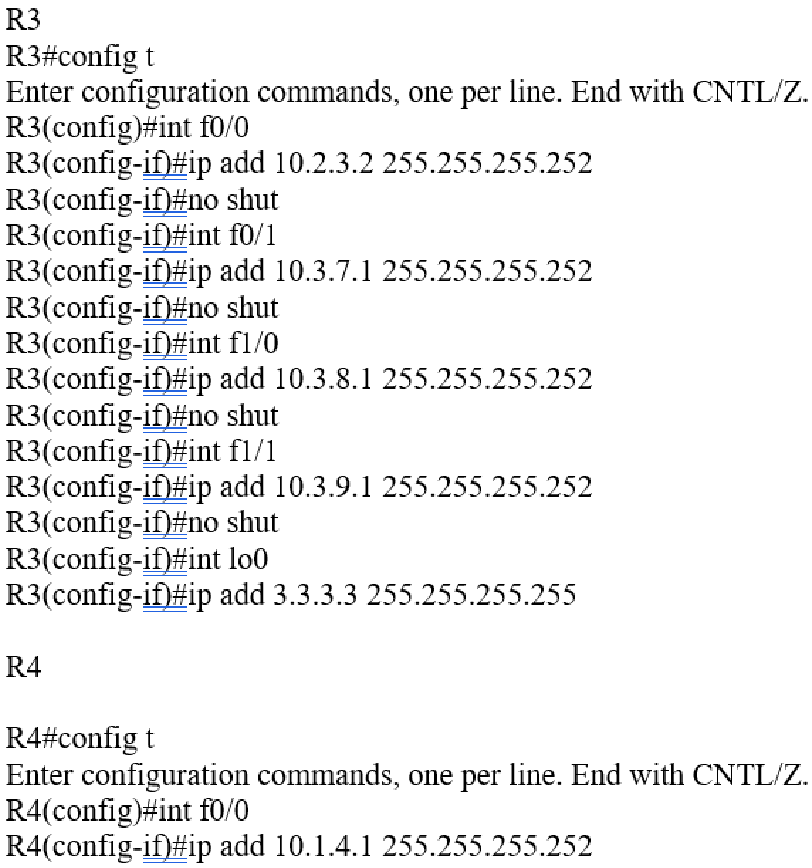
```
R8#config t
Enter configuration commands, one per line. End with CNTL/Z.
R8(config)#int f0/0
R8(config-if)#ip add 10.3.8.2 255.255.255.252
R8(config-if)#no shut

R8(config-if)#int lo0
R8(config-if)#ip add 8.8.8.8 255.255.255.255
```

R9

```
R9#config t
Enter configuration commands, one per line. End with CNTL/Z.
R9(config)#int f0/0
R9(config-if)#ip add 10.3.9.2 255.255.255.252
R9(config-if)#no shut

R9(config-if)#int lo0
R9(config-if)#ip add 9.9.9.9 255.255.255.255
```

R2

```
R2#config t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#int f0/0
R2(config-if)#ip add 10.1.2.2 255.255.255.252
R2(config-if)#no shut
R2(config-if)#int f0/1
R2(config-if)#ip add 10.2.3.1 255.255.255.252
R2(config-if)#no shut
R2(config-if)#int lo0
R2(config-if)#ip add 2.2.2.2 255.255.255.255
```

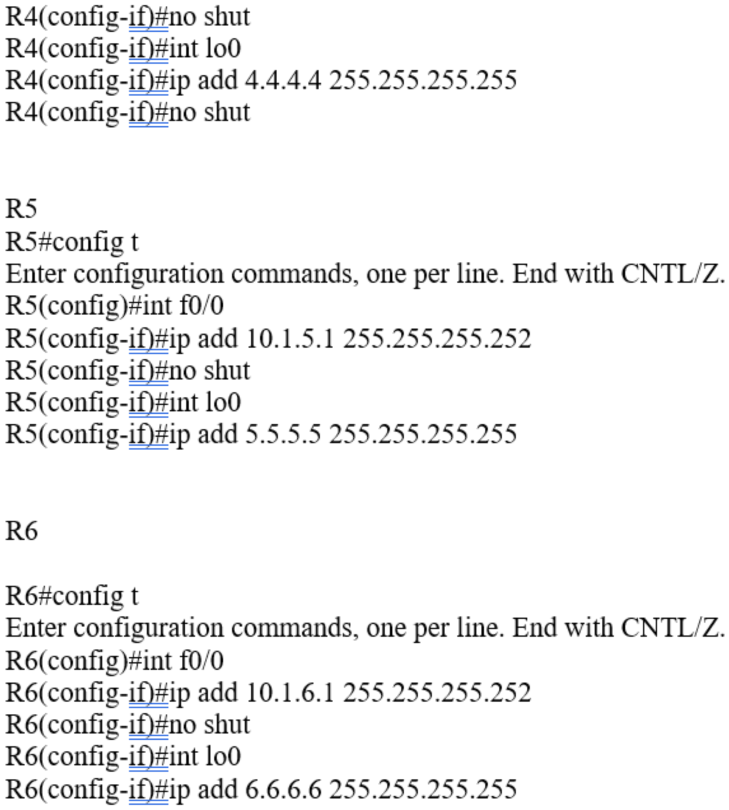


R3

```
R3#config t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#int f0/0
R3(config-if)#ip add 10.2.3.2 255.255.255.252
R3(config-if)#no shut
R3(config-if)#int f0/1
R3(config-if)#ip add 10.3.7.1 255.255.255.252
R3(config-if)#no shut
R3(config-if)#int f1/0
R3(config-if)#ip add 10.3.8.1 255.255.255.252
R3(config-if)#no shut
R3(config-if)#int f1/1
R3(config-if)#ip add 10.3.9.1 255.255.255.252
R3(config-if)#no shut
R3(config-if)#int lo0
R3(config-if)#ip add 3.3.3.3 255.255.255.255
```

R4

```
R4#config t
Enter configuration commands, one per line. End with CNTL/Z.
R4(config)#int f0/0
R4(config-if)#ip add 10.1.4.1 255.255.255.252
```



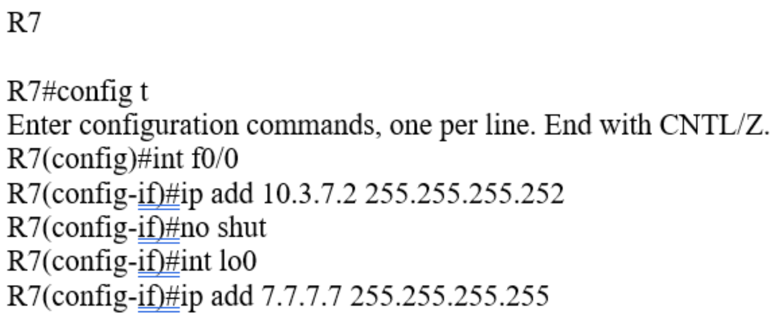
```
R4(config-if)#no shut
R4(config-if)#int lo0
R4(config-if)#ip add 4.4.4.4 255.255.255.255
R4(config-if)#no shut
```

R5

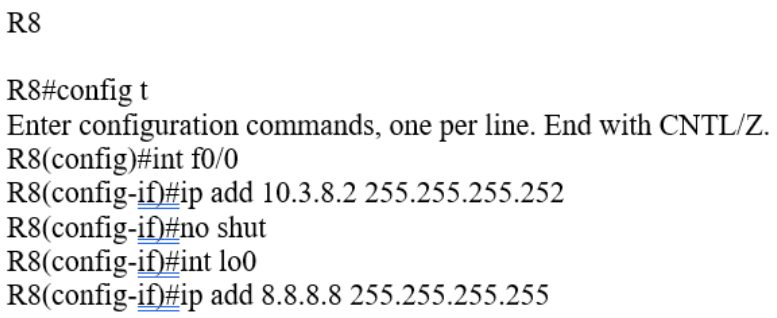
```
R5#config t
Enter configuration commands, one per line. End with CNTL/Z.
R5(config)#int f0/0
R5(config-if)#ip add 10.1.5.1 255.255.255.252
R5(config-if)#no shut
R5(config-if)#int lo0
R5(config-if)#ip add 5.5.5.5 255.255.255.255
```

R6

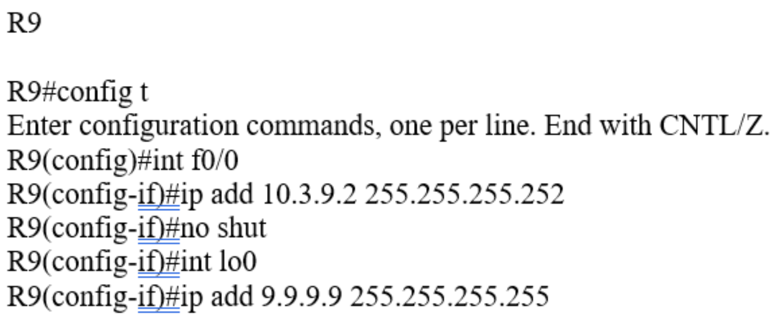
```
R6#config t
Enter configuration commands, one per line. End with CNTL/Z.
R6(config)#int f0/0
R6(config-if)#ip add 10.1.6.1 255.255.255.252
R6(config-if)#no shut
R6(config-if)#int lo0
R6(config-if)#ip add 6.6.6.6 255.255.255.255
```



```
R7

R7#config t
Enter configuration commands, one per line. End with CNTL/Z.
R7(config)#int f0/0
R7(config-if)#ip add 10.3.7.2 255.255.255.252
R7(config-if)#no shut
R7(config-if)#int lo0
R7(config-if)#ip add 7.7.7.7 255.255.255.255
```



```
R8

R8#config t
Enter configuration commands, one per line. End with CNTL/Z.
R8(config)#int f0/0
R8(config-if)#ip add 10.3.8.2 255.255.255.252
R8(config-if)#no shut
R8(config-if)#int lo0
R8(config-if)#ip add 8.8.8.8 255.255.255.255
```



```
R9

R9#config t
Enter configuration commands, one per line. End with CNTL/Z.
R9(config)#int f0/0
R9(config-if)#ip add 10.3.9.2 255.255.255.252
R9(config-if)#no shut
R9(config-if)#int lo0
R9(config-if)#ip add 9.9.9.9 255.255.255.255
```

11 References

Bennan, M. E.-A. (March 2008). Efficient QoS implementation for MPLS VPN. *22nd International Conference on Advanced Information Networking and Applications*, (pp. 259-260).

BGP, C. (2016). *IP Routing: BGP Configuration Guide*.

CISCO. (2016, February 23). *IP Routing: OSPF Configuration Guide*. Retrieved from https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_ospf/configuration/xe-16/iro-xe-16-book/iro-cfg.html: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_ospf/configuration/xe-16/iro-xe-16-book/iro-cfg.html

Cisco IOS Software Configuration Guide,. (n.d.). Cisco.

Configuring a Basic BGP Network. (2016). Retrieved from Cisco: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_bgp/configuration/xe-16/irg-xe-16-book/configuring-a-basic-bgp-network.html

EZE, M. O. (2018). SECURITY VULNERABILITY ON MULTI- PROTOCOL LABEL SWITCHING. *International Journal of Engineering, Science and Mathematics*, (pp. 50-52).

Ghein, L. D. (2007). *MPLS Fundamentals.* Cisco .

GNS3 Windows Install. (n.d.). Retrieved from GNS3: https://docs.gns3.com/docs/getting-started/installation/windows/

Haeryong Lee, J. H. (n.d.). End-To-End QoS Architecture for VPNs: MPLS VPN Deployment in backbone network. (p. 480). Electronics and Telecommunications Research Institute .

Hussain, I. (20 Dec 2004). Overview of MPLS Technology and Traffic . *2004 International Networking and Communication Conference.* IEEE.

Jeyakumar, S. Y. (2016). *Design of Traffic Engineered MPLS VPN for Protected Traffic using GNS Simulator.* IEEE.

Ming-Song Sun, W.-H. W. (2012). *Engineering Analysis and Research of MPLS VPN.* IEEE.

MPLS Basic MPLS Configuration Guide. (2016, Feb 10). Retrieved from Cisco: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/mp_basic/configuration/xe-16/mp-basic-xe-16-book/multiprotocol-label-switching-mpls-on-cisco-routers.html

Padole, D. M. (2017). An Insight into IP Addressing. *ORIENTAL JOURNAL OF COMPUTER SCIENCE & TECHNOLOGY.*

Route Distinguishers and Route Targets. (2013, June 10). Retrieved from Packetlife: https://packetlife.net/blog/2013/jun/10/route-distinguishers-and-route-targets/

Route Distinguishers and Route Targets. (2013, June 10). Retrieved from Packetlife: https://packetlife.net/blog/2013/jun/10/route-distinguishers-and-route-targets/

Smith, S. (2003). *Introduction to MPLS.* Cisco.

Swallow, G. (1999). MPLS Advantages for Traffic Engineering . *IEEE Communications Magazine*, (pp. 54 - 57).

Tao, J., Chen, H., Liu, H., & She, X. (2023). Experimental Design and Teaching Research of a MPLS VPN Network Based on BGP . *IEEE 12th International Conference on Educational and Information Technology*, (p. 290). Chongqing.

Tony Li, P. N. (1999). MPLS and the Evolving Internet Architecture. *IEEE Communications Magazine* (pp. 38-41). IEEEE.

Yongming WEI', Z. F. (2006). A MPLS and VPN Based eGovernment System. *Proceedings of the 15th International Conference on Computing* (pp. 1-2). IEEE.