# DESIGNING AND EVALUATING A RELIABLE NETWORK TOPOLOGY BY USING BGP

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES OF
ÇANKAYA UNIVERSITY

BY
ALİ MURAT KARAOĞLU

IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF
MASTER OF SCIENCE
IN
DEPARTMENT OF COMPUTER ENGINEERING

FEBRUARY 2017

Title of Thesis: **Designing and Evaluating a Reliable Network Topology by Using BGP**

Submitted by: **Ali Murat KARAOĞLU**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University

Prof. Dr. Halil Tanyer EYYUBOĞLU
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science

Prof. Dr. Erdoğan DOĞDU
Head of Department

This is to certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Murat SARAN
Supervisor

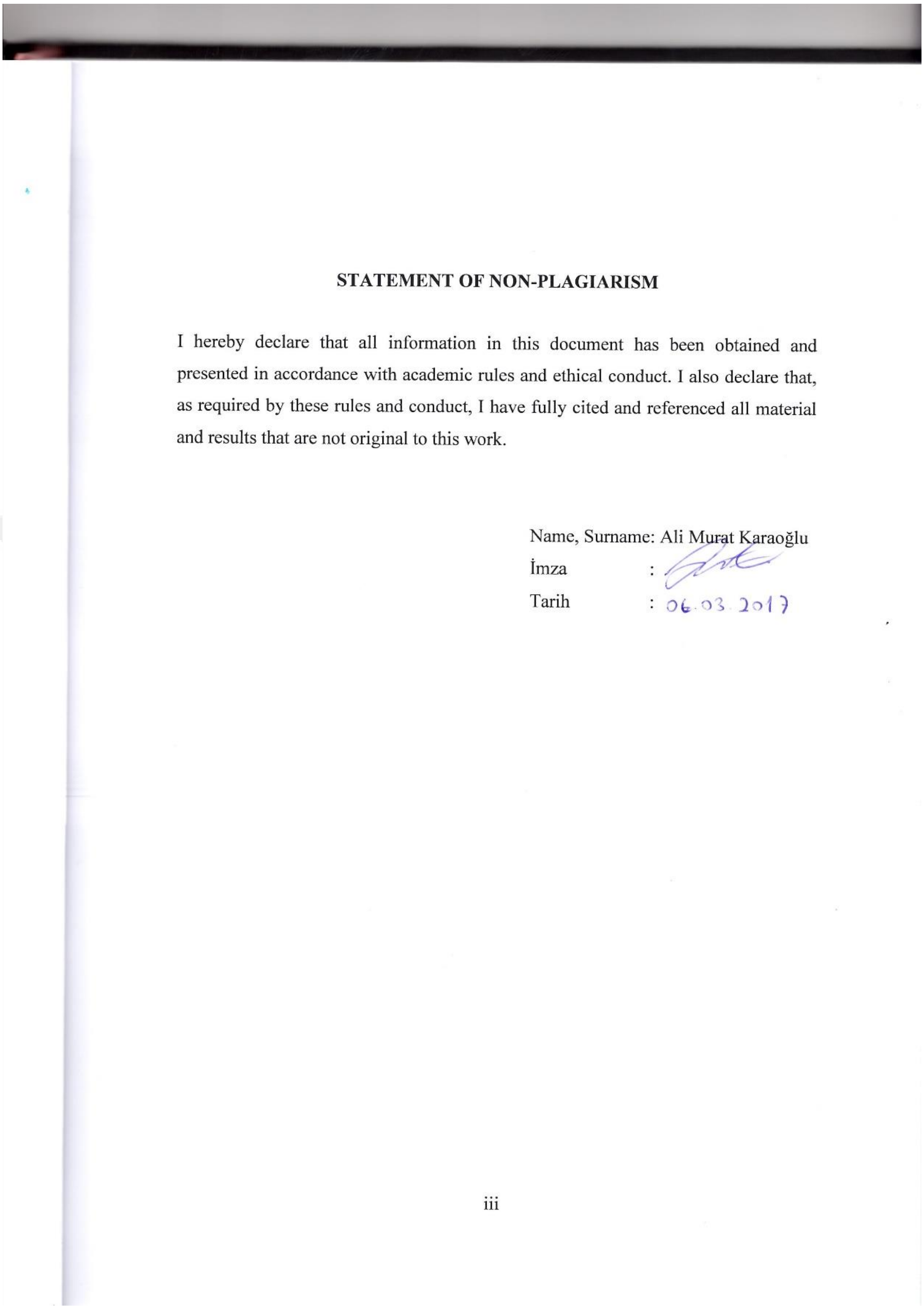
**Examination Date:** 07.02.2017

**Examining Committee Members**

Assist. Prof. Dr. Reza HASSANPOUR (Çankaya Univ.)

Assist. Prof. Dr. Murat SARAN (Çankaya Univ.)

Assist. Prof. Dr. Gökhan ŞENGÜL (Atılım Univ.)



## STATEMENT OF NON-PLAGIARISM

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Ali Murat Karaoğlu
İmza :
Tarih : 06.03.2017

# ABSTRACT

## DESIGNING AND EVALUATING A RELIABLE NETWORK TOPOLOGY BY USING BGP

Karaoğlu, Ali Murat

M.Sc., Department of Computer Engineering

Supervisor: Assist. Prof. Dr. Murat SARAN

February 2017, 97 pages

Border Gateway Protocol (BGP), an exterior gateway protocol, carries routing information between autonomous systems (AS) via Internet Service Providers (ISP) and the Internet. Alternative routing protocols are insufficient for use on the Internet as they are principally insufficient to cope with large routing tables. BGP is a dynamic routing protocol, and it is used by large enterprises to connect their sites via other ASs and the Internet. This study provides an implementation of BGP with some of the critical attributes using the open source network simulator, GNS3. The main aim of this study is to create a reliable and redundant topology for computer networks. In this study, route manipulation techniques are used in the network configuration process to provide redundancy. Moreover, a failover scenario is examined and tested in order to observe the effects of failovers on the routing paths. This thesis presents a useful BGP topology for real-life computer network settings.

**Keywords:** Routing, BGP, Configuring BGP, Route Manipulation, NAT-PAT, MPLS, VRF, GNS3

# ÖZ

## BGP PROTOKOLÜ İLE GÜVENİLİR BİR AĞ TOPOLOJİSİNİN DİZAYNI VE DEĞERLENDİRİLMESİ

Karaoğlu, Ali Murat

Yüksek Lisans, Bilgisayar Mühendisliği Anabilim Dalı

Tez Yöneticisi: Yrd. Doç. Dr. Murat SARAN

Şubat 2017, 97 sayfa

Border Gateway Protocol (BGP), dış ağ geçidi protokolü, yönlendirme bilgilerini Servis Sağlayıcılar ve İnternet üzerinden otonom sistemler arasında taşır. Diğer yönlendirme protokolleri İnternet ile birlikte kullanılmak için yetersiz kalmaktadır; çünkü onlar, büyük yönlendirme tablolarını yönetmekte yetersizdir. BGP dinamik bir yönlendirme protokolüdür ve büyük kurumsal firmalar tarafından ofislerinin farklı otonom sistemler ve İnternet üzerinden bağlantısının kurulması amacıyla kullanılır. Bu çalışmada BGP protokolünün bazı önemli özellikleri kullanılarak açık kaynak kodlu ağ simülatörü GNS3 üzerinden bir uygulaması sunulmaktadır. Çalışmanın ana amacı, bilgisayar ağları için güvenilir ve yedekli bir topoloji ortaya koymaktır. Yedekli yapının sağlanması amacı ile konfigürasyon aşamasında rota manipülasyon teknikleri kullanılmıştır. Ayrıca, problemlerin yönlendirme patikaları üzerindeki etkilerini gözlemlemek amacı ile bir hata senaryosu açıklanıp test edilmiştir. Bu tez gerçek hayattaki bilgisayar ağlarında uygulanabilecek kullanışlı bir BGP topolojisi sunmaktadır.

**Anahtar Kelimeler:** Yönlendirme, BGP, BGP konfigürasyonu, Rota manipülasyonu, NAT-PAT, MPLS, VRF, GNS3

## TABLE OF CONTENTS

# LIST OF FIGURES

## LIST OF TABLES

## ACKNOWLEDGEMENTS

I would like to thank you my whole family for supporting me during my thesis study. Moreover, I wish to thank you to my fiancée Sinem KARACABAY that she participated to the process.

I wish to specially thank to my thesis’ advisor Assist. Prof Dr. Murat SARAN by giving me the opportunity to study for this thesis.

# 1. INTRODUCTION

## 1.1 Overview

The routing concept appeared in the late 1950s [1]. At a time when computers were rarely established and only in few organizations. These computers needed to be linked together and consequently, routing was born. Linked computers create networks, which in turn create internetworks.

Networks can be categorized as Local Area Networks (LANs) and Wide Area Networks (WANs). Multiple users in a small area make constitute LANs, and LANs are the components of WANs. A LAN and a WAN are illustrated in Figures 1 and 2 below.

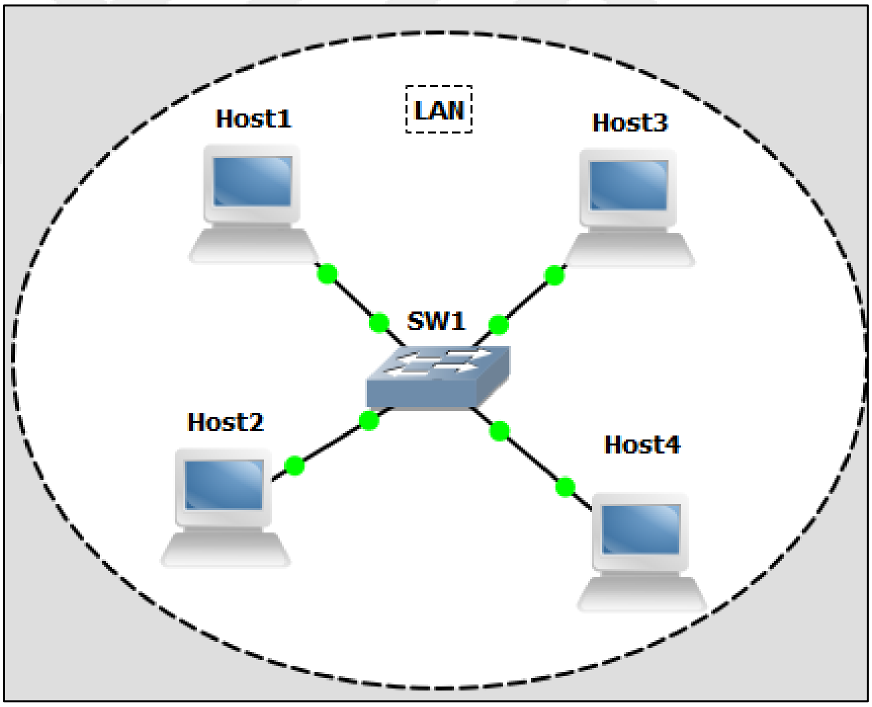


**Figure 1.** Local Area Networks (LANs)

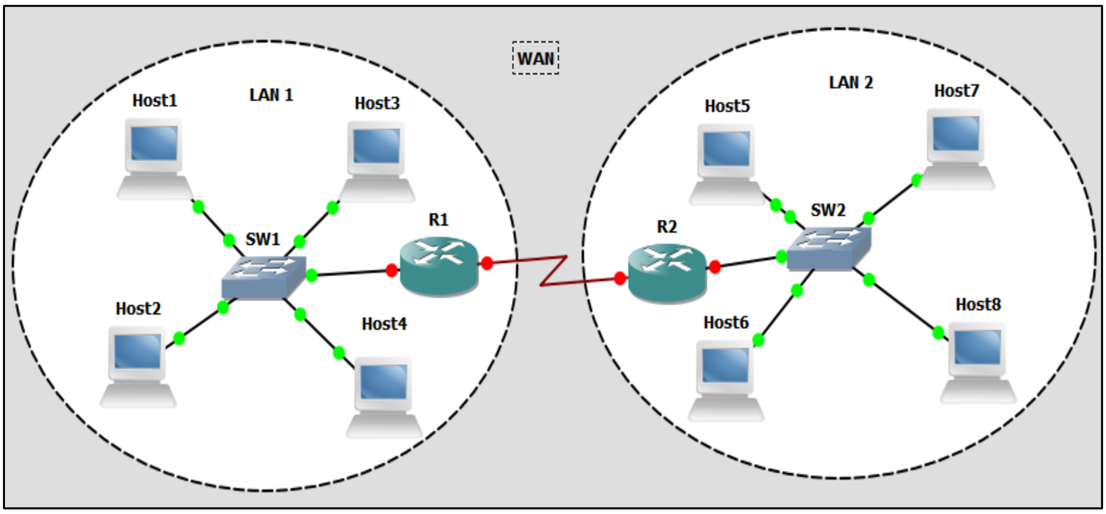


**Figure 2.**Wide Area Networks (WANs)

Bringing networks together can only be accomplished by routing. Static routing or dynamic routing protocols provide the solution. Static routing is used for simple networks. Because they do not know about network changes, network administrators must change the configuration for every network change. Dynamic routing protocols are used for large, complex networks. Network changes do not affect them. The most common dynamic routing protocols: include RIP (Routing Information Protocol), EIGRP (Enhanced Interior Gateway Routing Protocol), OSPF (Open Shortest Path First) and BGP (Border Gateway Protocol) [2]. Each protocol has its own capabilities and they are used in different scenarios. They will be reviewed in the following sections.

BGP is used to interconnect ASs (autonomous system) and to connect customer sites with ISPs (Internet Service Providers). The Internet is the largest computer network; therefore, the devices on the Internet advertise very many routing updates. BGP has a number of features to handle this. The alternative routing protocols are insufficient to solve such problems. Moreover, BGP easily adapts customer computer networks to the Internet, and it offers redundant topologies. There are many studies in the literature on BGP, and this study examines some of them and illustrates how to create a computer network with these characteristics.

### 1.2 Aim of the Study

BGP is an inter-autonomous protocol that carries information via service provider networks to other ASs. BGP has many features that can be configured. However, for real-life scenarios route manipulation techniques, MPLS and VRF are very important. The objective of the thesis is, to design and evaluate a reliable network topology using BGP. To do so, the open source network simulator software GNS3 will be used. Virtual routers will be configured for an example topology. To provide redundancy, route manipulation techniques will be used. Customer side routers and ISP routers will also be configured. Afterwards, clients in the customer side networks will be made to reach to the Internet via redundant gateways. At the end of the configuration and testing, an example implementation of a redundant and reliable BGP network which can be used in real-life will be presented.

There are many studies on BGP in the literature. However, practical studies are difficult to find. Most practical studies only partially examine the features of BGP. Consequently, this study focuses on how to design and configure a complete computer network with BGP. The topology and configurations of routers in this study can be easily used in real life and they can provide a different view to network administrators.

### 1.3 Research Question

This study focuses on finding answers to two main questions and three sub-questions.

- Why is the BGP protocol used in computer networks?
  - Can alternative routing protocols be used instead of BGP?
- How can a computer network be designed with BGP?
  - How are the routers configured for BGP?
  - Can route manipulation be used to provide redundancy in BGP configurations?

### 1.4 Organization of the Thesis

This thesis consists of five chapters: an introduction, a background, a presentation of the implementation and testing, results and the conclusion and future work.

The first chapter presents a preamble and the aim of the study. The second chapter introduces the background of the routing and dynamic routing protocols. In addition, an introduction to BGP, MPLS and VRF is given in this chapter. The third chapter is the implementation and testing chapter. In this chapter; how to configure BGP in routers is presented, in addition to the implementation, topology and testing. The fourth chapter provides results of the implementation. Finally, the fifth chapter presents the conclusion of the chapters and future work.

# 2. BACKGROUND

## 2.1.Introduction to Routing and Packet Forwarding

Routing carries information between a source and a destination. This can be performed by selecting the best paths in a network. Routers are responsible for the routing process. It determines the best path, regardless of a routing protocol, and installs it to the routing table followed by forwarding packets to the destination. When a packet reaches a router, the router finds the packet's IP header in order to obtain the destination address; then it checks the local routing table. If there is a match, it forwards the packet to the destination. If the router does not have a route, it uses the default route to forward the packet. If there is no default route in the routing table, the packet is discarded and an ICMP message is sent to the source to state that the destination is unreachable [3]. Routers perform the routing process in two ways. They can be configured to use static routes, or they can be configured to use dynamic routing protocols.

## 2.2. Static Routing

Static routes are manually entered routes by the network administrator. When a static route is configured, the router searches for a specific gateway (next-hop) or a specific local interface to forward packets. Static routes cannot react to changes in the network, and they always remain in the routing table even if a destination is unreachable. Consequently, when a topology change occurs, the network administrator has to modify the route. If there is no destination for a route, the router searches through its routing table for whether there is a default route. Default routes are also manually entered, and they are used for the remaining routes, which may have any destination. If no default route is configured, then the packet is discarded.

Static routing has a number of advantages, however, they are used for small sized networks; they are not appropriate for medium sized or large networks. Since when there is a topology change, the network administrator must modify the static route due to the fact that the router cannot be informed about the change by any protocol, and thereby making the network uncontrollable. Generally static routes are used as backup routes, or there are temporary link problem and bandwidth issues. Static routes make a network more secure, and they do not waste CPU resources to calculate routes, thereby requiring less memory. If the network size is small and has need for only static routes, it is easier to find a cheaper router or a switch with routing capabilities. Since routers are designed for dynamic routing, they are expensive devices.

In summary; static routes eliminate all traffic caused by routing updates; therefore, they are ideal for point-to-point WAN links, and can be used as backup when primary links fail. If an entire network has been configured with static routes, this can cause extreme administrative overhead, so it is better for use in small networks.

### 2.3.Dynamic Routing Protocols

Routers must use the same routing protocol to speak with each other. Routing protocols are the language of routers. Dynamic routing protocols are responsible for all the routing processes of a network. Responsibilities may include best path determination, finding another path if the best path becomes unusable, and routing table updates. The most specific difference with static routes is the reaction to topology changes. Therefore, there is no need to change routes manually. Each routing protocol uses an algorithm. These algorithms have either common issues or special issues. An algorithm must have the minimum following specifications: passing and receiving reachability information, obtaining routes and recording this information to their routing tables, and reacting and advertising topology changes. Moreover, dynamic routing protocols have some common issues including path determination, metrics, convergence, and load balancing.

All the interfaces of a router have an IP address, and the path determination process is carried out according to these addresses. If a router has more than one route to a destination, it accomplishes this with metric values [4]. Metric values are changeable to the routing protocols, bandwidth and delay values of a link. Convergence occurs when all the routing tables of a network are made consistent. Routing tables must be consistent; otherwise, routing loops may occur. To use the bandwidth efficiently, load balancing is used if there are proper circumstances, such as multiple paths and identical metric values to a destination.

Figure 3 shows two types of routing. Dynamic routing protocols are split as Interior Gateway Protocols (IGP), and Exterior Gateway Protocols (EGP). IGPs are dynamic routing protocols that are used within an autonomous system and EGPs are dynamic routing protocols that are used between different autonomous systems [5].

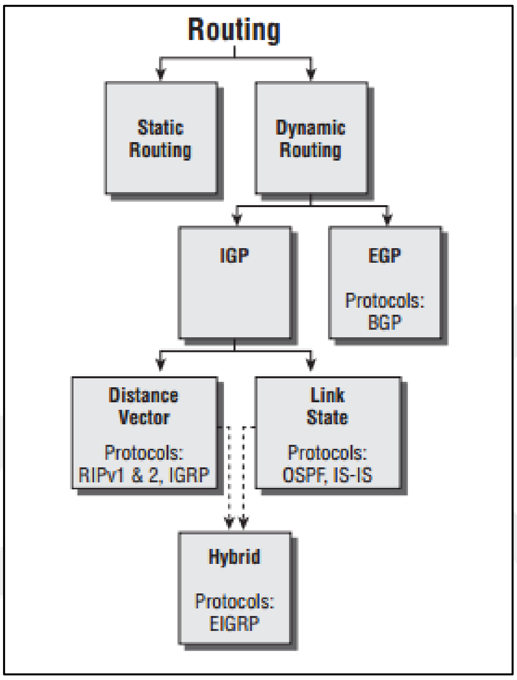


**Figure 3.** Routing Protocols

IGP’s are split into two different categories: Distance Vector Routing Protocols, and Link State Routing Protocols. These categories are based on how the best paths are calculated with their own algorithms.

### 2.3.1. Distance Vector Routing Protocols

Distance vector routing protocols are sometimes called Bellman-Ford or Ford-Fulkerson algorithms [5]. According to these algorithms, entire routing tables are sent periodically. Each hop on the path adds their distance vector to the routing table, after which it forwards it to the next hop routers as a broadcast. All the neighbor routers receive these update packets. Therefore, routers learn the network from their neighbor routers’ perspective, but they cannot know the entire topology. This is known as routing by rumor. For distance vector routing, routers actually share their routing tables with each other. RIPv1, RIPv2, and IGRP are distance vector routing protocols, which are support broadcasts; additionally, RIPv2 supports multicasts, and sends its updates as triggered updates, which makes routers send an update as soon as a topology change has occurred.

Periodic update packets, which include destination routes and distance to advertising routers (hops), are sent based on timers. Whenever a new router is added to the network, it requests the entire routing table of other routers. The other routers wait for the timer to expire, after which they send their routing tables. Periodic updates are not sent when a topology change occurs. They are always sent periodically, and this causes unnecessary traffic. Moreover, if a router is not informed about a failure of a link or a topology change immediately, routing loops may occur.

When a router receives update information, it adds its own vector (one more hop) and waits for the timer. After the timer expires, it broadcasts the update packet from all of its interfaces except for one interface, the one which previously received the update packet (split horizon). The timer controls the sending update packets and when a change occurs (new router, failed link, etc.), it takes time to inform the other routers. Consequently, the convergence of the network occurs slowly.

Distance vector routing protocols have a number of problems. Such as, the hop count limit. When the hop count limit (generally 15) is exceeded the packet is dropped, which may be a problem for complex networks. Periodic updates can cause high CPU utilization and link overhead. Distance vector routing protocols do not support VLSM (Variable Length Subnet Masking) and CIDR (Classless Inter Domain Routing). They do not carry mask information in the routing updates, and there can be a match problem between network addresses. The most important problem in routing protocols is convergence time. Due to slow convergence times, distance vector routing protocols are not appropriate for large networks.

### 2.3.2. Link State Routing Protocols

Link state routing protocols give the whole picture of the network to the routers. Each router builds some information about itself and forwards it to their directly connected neighbors; then, each router passes this information on to other routers without any change [5]. They do not share their routing tables, but they do share their neighbor tables and topology tables with other routers. Therefore, every router knows the link states of every other router and they calculate the routes separately. Finally, each router can see the entire topology of the network.

Link state routing protocols use E.W. Dijkstra's algorithm, which is known as the shortest path first algorithm [5]. When the convergence of the network finishes, every router has a complete topology of the network, after which they apply Dijkstra's algorithm to calculate the best paths to the destinations. OSPF (Open Shortest Path First), and IS-IS (Intermediate System to Intermediate System) are well known among link state routing protocols. Distance vector routing protocols calculate the best path based on distance, however link state routing protocols calculate best path based on bandwidth, delay, reliability, load, and MTU [2]. Consequently, a routing decision is more efficiently performed for link state routing protocols.

Link state routing protocols use triggered updates, known as Link-State Advertisements (LSAs), to inform routers about changes in a network. After the first convergence of the network, triggered updates are only sent when there is a topology change in the network. Additionally, update packets only include the changed portion of the information. These updates are sent as multicasts, unlike distance vector updates (broadcast). Thus, link state routing protocols have many advantages in addition to distance vector routing protocols, including lower CPU and memory consumption and less busy network traffic. We cannot discuss periodic updates for the link state routing protocols. Unlike distance vector routing protocols, link state routing protocols support classful routing, which means update packets include subnet mask information.

In summary, link state routing protocols have many more advantages over distance vector routing protocols, and they are used widely. However, distance vector protocols may be preferred under some circumstances, such as network size, when other protocols are used in the network, and according to customer requirements.

## 2.4.BGP (Border Gateway Protocol)

As an exterior gateway protocol (EGP), BGP is an inter-autonomous system routing protocol [2]. An autonomous system, which is a collection of networks under a single administrative domain, can contain different sets of routing protocols, and the BGP interconnects autonomous systems known as EBGP. Separately, BGP can be used within an autonomous system known as IBGP. Generally, BGP is used by Internet Service Providers (ISPs) and information is exchanged over the Internet or a cloud. Figure 4 illustrates EBGP and IBGP routers in different ASs.

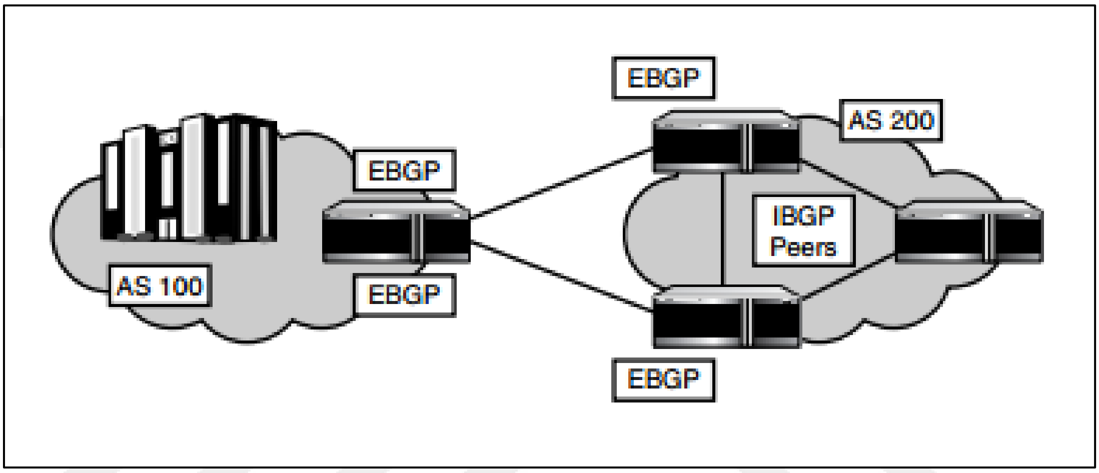


**Figure 4.** EBGP and IBGP [2]

The first version of BGP was introduced in 1989 in RFC 1105, and named BGP-1; the second version was introduced in 1990 in RFC 1163, and named BGP-2; the third version was introduced in 1991 in RFC 1267, and named BGP-3. The final and current version was introduced in 1995 in RFC 1771, and named BGP-4.

Enterprises can use single or multiple connections to reach the Internet. If a single connection is used, there is no need to use BGP; instead, a default route can be used, which means that all the traffic is routed through this route. If there are multiple connections to reach the Internet via single or multiple ISPs, BGP needs to be used as BGP has a number of attributes that select better paths over multiple paths to a single destination. Moreover, if the network is a transit area for other networks, BGP should be used. Consequently, BGP must be used for the Internet connection. Without BGP, it is not possible to transit to other autonomous systems or the Internet.

In a BGP router domain, after the first configuration of routers, all share their full routing tables; then the network becomes converged. Following that point, the routers do not send periodic updates; instead, they send only the changed portion of the network if there is a topology change and they only send optimal paths in these announcements. Additionally, BGP uses CIDR which reduces the possible overhead of the routing table.

### 2.4.1. BGP's Path Vector Functionality

BGP carries routing information with *path vector* functionality between autonomous systems. This functionality is based on AS paths; which are defined by AS numbers. The other routing protocols use metric values to carry routing information. However, BGP uses path vectors and these path vectors occur from path attributes. Path vectors hold very AS number in the network; therefore, the network topology is occurred AS by AS. Moreover, there are many more attributes to be explained in the following sections.

Path vector functionality prevents loops with AS path information. If a BGP router already knows an AS number in the network, it does not accept the related routing update to provide the loop-free mechanism. AS path information controls the routing behavior by identifying routing policies to manipulate path selection.

BGP announces paths, AS numbers and IP addresses. The paths can be selected by a network administrator by defining routing policies with some of the BGP attributes. Figure 5 illustrates path vector functionality.

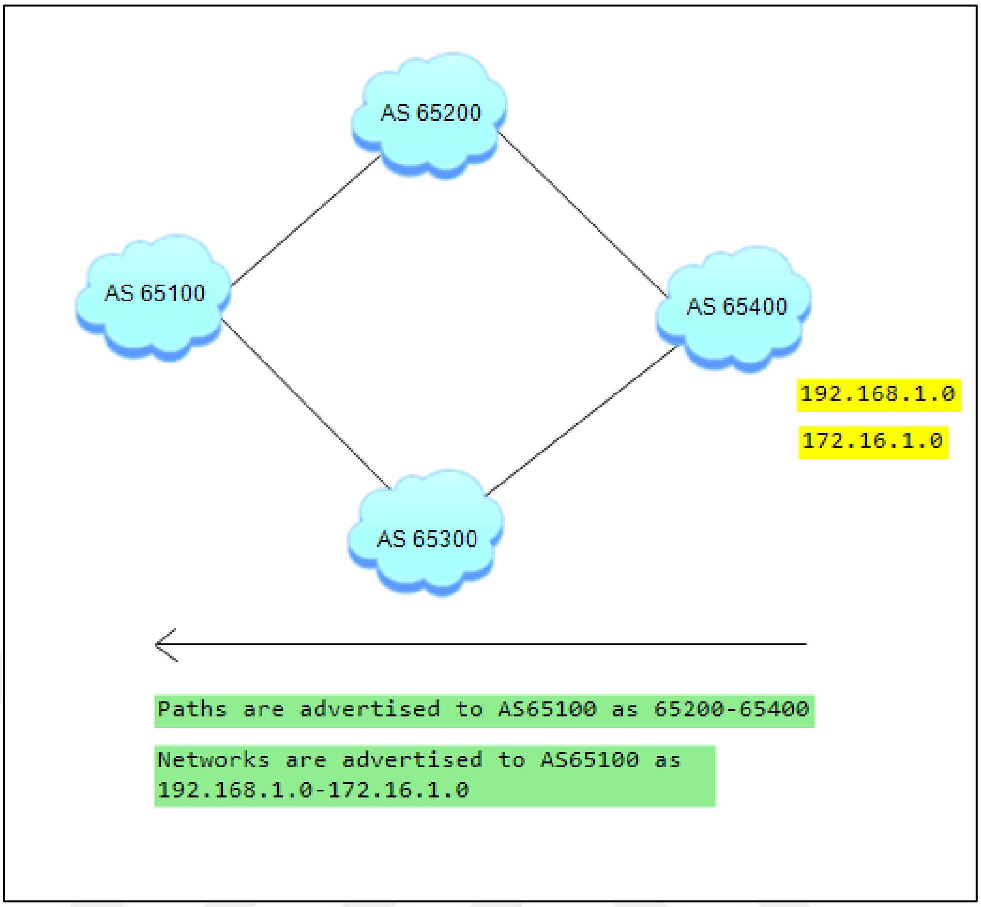

AS 65200
AS 65100
AS 65400
192.168.1.0
172.16.1.0
AS 65300
Paths are advertised to AS65100 as 65200-65400
Networks are advertised to AS65100 as 192.168.1.0-172.16.1.0

**Figure 5.** BGP Path Vector Function

The networks behind AS65400 are advertised to AS65100 via AS65200. This operation can also be performed via AS65300. The path vector function does this with some of the BGP attributes and BGP routing policies.

### 2.4.2. BGP Routing Policies

The BGP routing policy can be implemented based on autonomous systems. It covers all the networks that are related to an individual AS. A routing policy is applied to a network prefix. For instance, if a policy is applied to the 192.168.0.0/16 prefix, all the addresses from 192.168.0.1 to 192.168.255.255 are influenced.

BGP sees autonomous systems as a hop. Routers advertise related routes to other autonomous systems based on these criterion, and this is known as hop-by-hop routing. The BGP routing policy can be set for inbound to outbound routes. In other words, reaching neighbor autonomous systems can be controlled for a certain path; however, incoming routes from neighboring autonomous systems cannot be controlled. That limitation can be handled by some different capabilities of BGP, which is explained in the following sections.

To illustrate, the BGP routing policy (Figure 6) can be used:

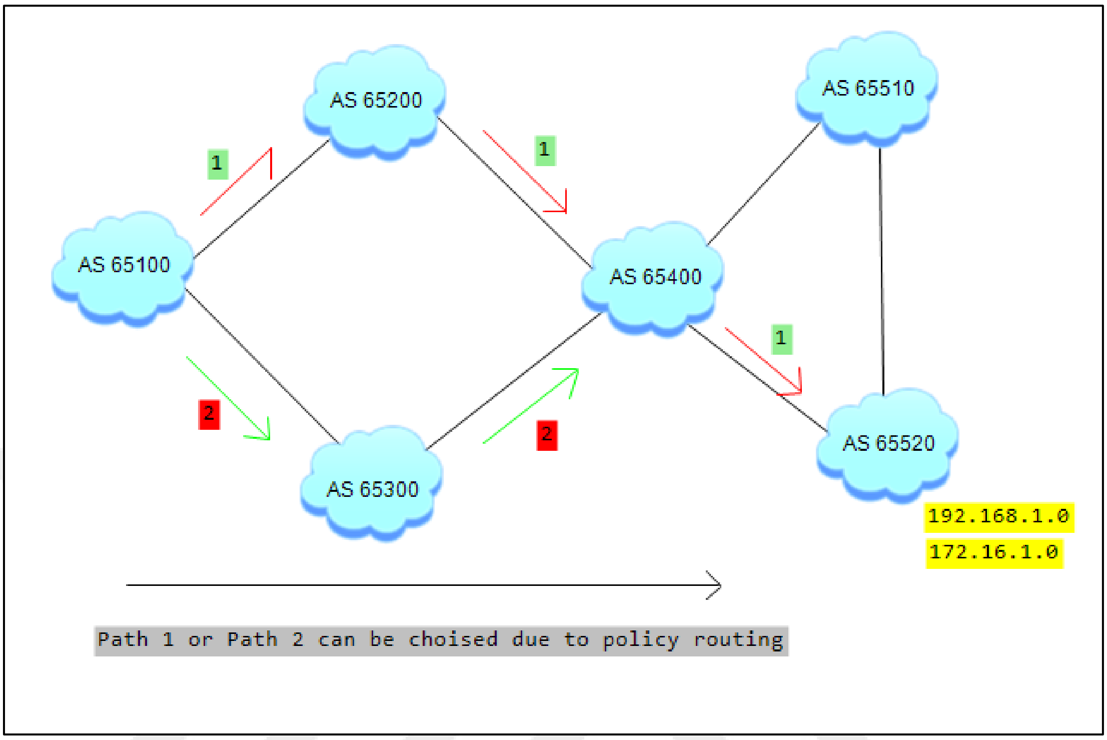


**Figure 6.** BGP routing policy

Path 1 or Path 2 can be selected at the edge router of 65100 AS by using policy routing. Moreover, AS 65400 advertises only Path 1 to AS 65200 and AS 65300 because Path 1 is the best least-cost route for AS 65400 to 65520. AS 65100 will use the path of 65200, 65400, 65520, 65300, 65400, 65520. It will not use 65510 path to reach the network behind AS 65520. AS 65100 can learn this path only if there is a failure at AS 65400 and AS 65520 (hop-by-hop routing).

### 2.4.3. BGP's Work Principles

To enable BGP on a router, first an autonomous system number must be configured. Then, a BGP speaker router is selected for this autonomous system and the peer router's IP addresses is configured. Afterwards, peer routers are configured with the autonomous system number and the BGP speaker router's IP address. The BGP speaker builds a TCP connection with the peers to share BGP information. With the establishment of the TCP connection, peer routers share their reachability information with the speaker to build routing tables. Finally, all the routers create a loop-free BGP network map.

When the peer routers finish exchanging routing tables, peers exchange the changes. There are now no longer any existing periodic updates. Hereafter, TCP controls the BGP connection, and it sends *keepalives* for a certain amount of time if there is no activity with the peer and speaker. If the TCP connection fails due to an error, the BGP session fails.

BGP routers separate the BGP information with tables. They store received routing update information, advertised routing update information, and the routing table. The BGP speakers inform their neighbors about any changes in the network by exchanging updates. For instance, if a peer router has become unavailable, the speaker informs the other peers with updates for removing the related route from their routing tables. Additionally, when the speaker finds a better path for a route, it informs the peers again with updates. The update includes the new path information and its attributes. Thus, the peer changes its routing table with the new route.

### 2.4.4. Message Types

BGP uses TCP to establish a relationship between routers. Consequently, BGP routers must implement TCP's three-way handshake operation over the TCP 179 port. TCP provides reliable connection for BGP peers. If UDP were to be used, we would not speak about reliable connections and there could be security vulnerability issues. BGP routers send unicast TCP messages to build relationships. There are four message types: Open, *Keepalive, Update*, and *Notification* [2]. These messages participated at the BGP header(as shown in Figure 7).

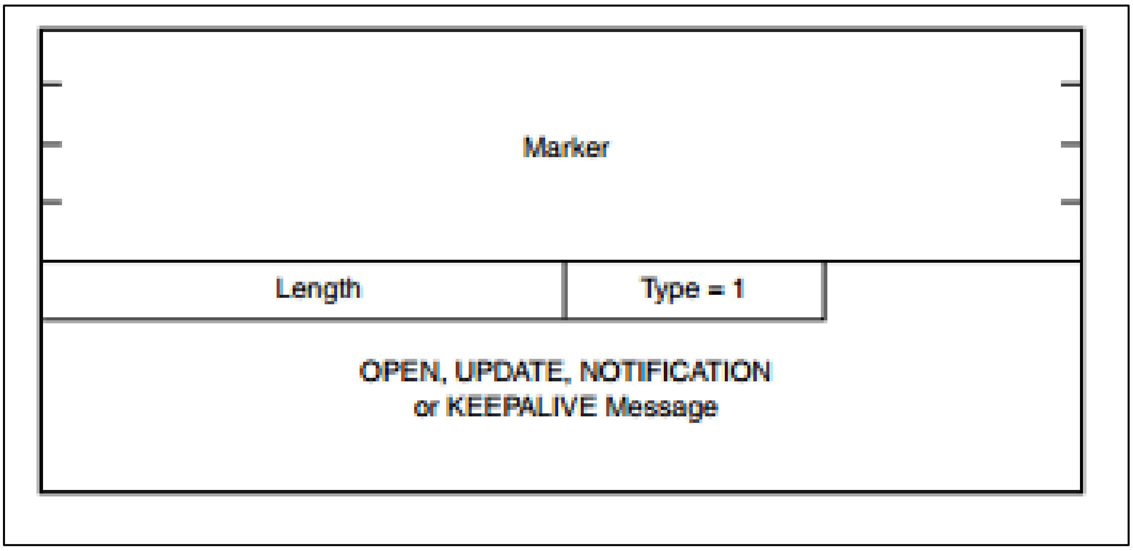


**Figure 7.** BGP Header [2]

The marker is a 16-byte field that specifies the peering connection and authentication. The length is a 2-byte field that specifies the BGP datagram and header. The type is a 1-byte field that specifies the message type.

After completing the TCP's three-way handshake operation, the TCP connection is established. Therefore, BGP peer routers send Open messages to each other. The Open message contains the BGP version number (which could be 2, 3, or 4), an autonomous system number, the hold time, the BGP identifier, and any optional parameters. The default value of the version number is 4. However, if a router sends another version number, this Open message is rejected and the other router is informed about the actual version number. Subsequently, the other router sends a new Open message with the related BGP version number. The AS number specifies whether the BGP session is either EBGP or IBGP. If both routers have identical numbers, it means the BGP session is IBGP. The Hold time is the amount of time during which a router receives an *Update* or *Keepalive* message. During Hold time, if a router does not receive any of these messages, the TCP connection fails. The BGP identifier is the IP address of the router. Generally, loopback interfaces are used for this function, which is the name of the BGP router in the AS. Optional parameters are used for a number of functions such as route refresh, multiprotocol support, and authentication. If a router has one of these functions, it negotiates with the other peer as to whether or not it has that function.

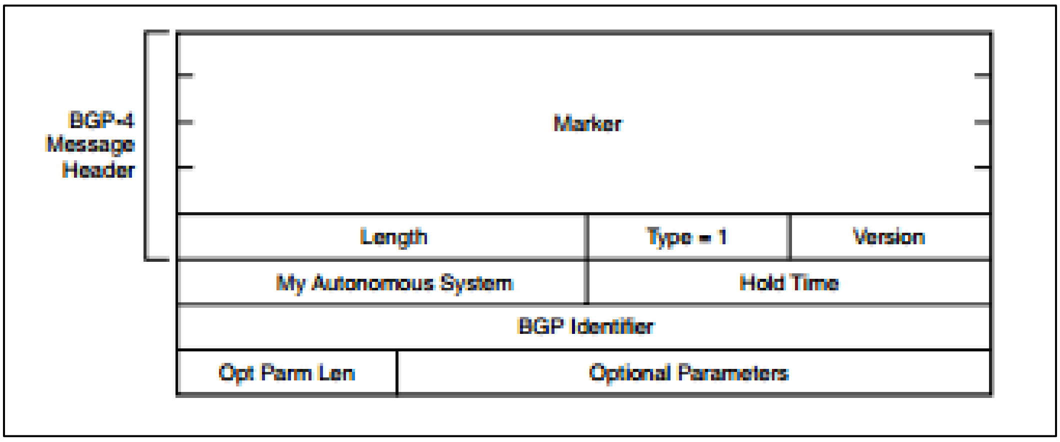
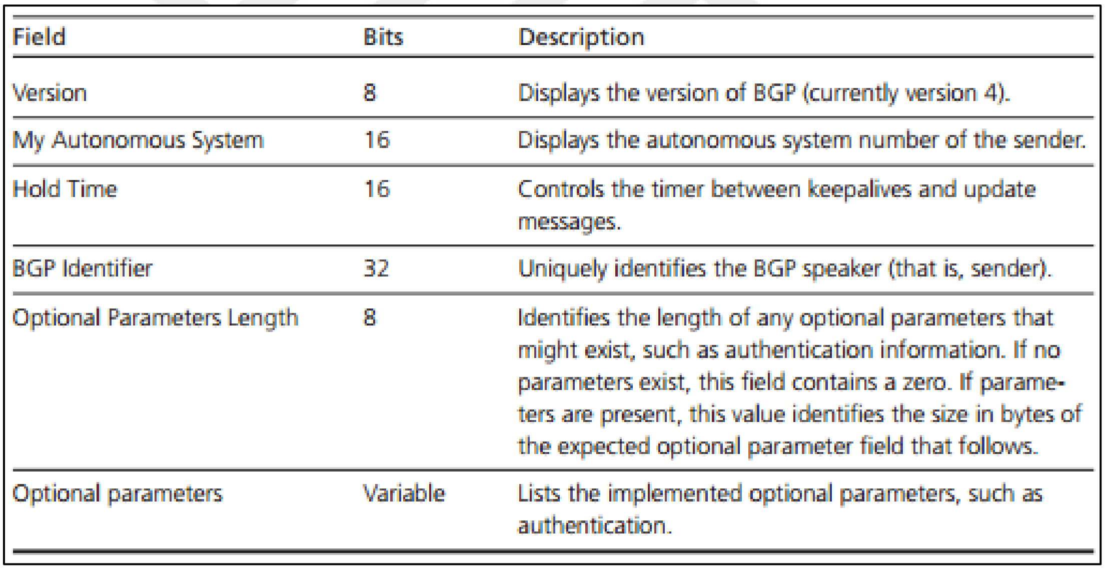


**Figure 8.** BGP Open Message [2]

The Open message is a type 1 message and it consists of six fields: Version, My Autonomous System, Hold Time, BGP Identifier, Optional Parameters Length, and Optional Parameters. Figure 8 illustrates an Open Message and Table 1 presents the Open Message Fields:

**Table 1.** BGP Open Message Fields [2]



| Field | Bits | Description |
|---|---|---|
| Version | 8 | Displays the version of BGP (currently version 4). |
| My Autonomous System | 16 | Displays the autonomous system number of the sender. |
| Hold Time | 16 | Controls the timer between keepalives and update messages. |
| BGP Identifier | 32 | Uniquely identifies the BGP speaker (that is, sender). |
| Optional Parameters Length | 8 | Identifies the length of any optional parameters that might exist, such as authentication information. If no parameters exist, this field contains a zero. If parameters are present, this value identifies the size in bytes of the expected optional parameter field that follows. |
| Optional parameters | Variable | Lists the implemented optional parameters, such as authentication. |

When the negotiation of the Open message finishes, a *Keepalive* message is sent periodically. When a *Keepalive* message is sent, the BGP peer connection is established. *Keepalive* messages are sent periodically to check the peer connection. Figure 9 illustrates the BGP *Keepalive* message:

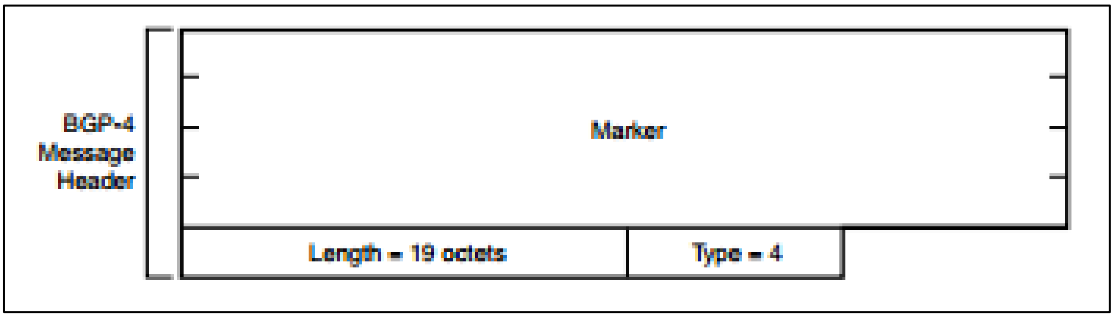


**Figure 9.** Keepalive Message [2]

The *Keepalive* message is a type 4 message. It consists of a BGP header and checks the BGP connection between peers.

The Update message (Figure 10) advertises newly added feasible routes and withdrawn routes from the routing table. Additionally, they contain the Network Layer Reachability Information (NLRI), Path Attributes, and Withdrawn Routes. NLRI holds the length (subnet mask information), and prefix for a number of routes that will be advertised. In spite of the multiple prefixes advertised in the NLRI field, the Update messages advertise a single route. However, this route may sign multiple destinations. The Path attributes help the router detect the shortest path, routing loops, and routing policy. Moreover, a network administrator can carry out route manipulation with these attributes. Withdrawn routes are the routes that have become unreachable and deleted from the routing table.

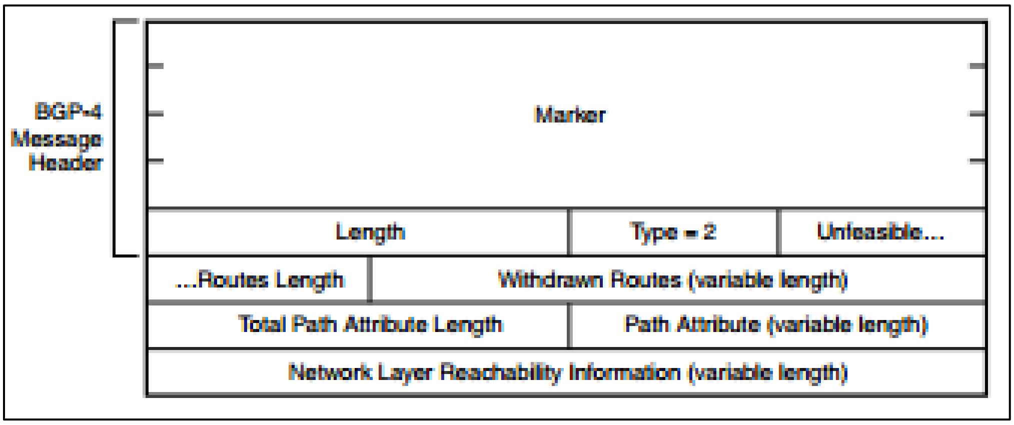
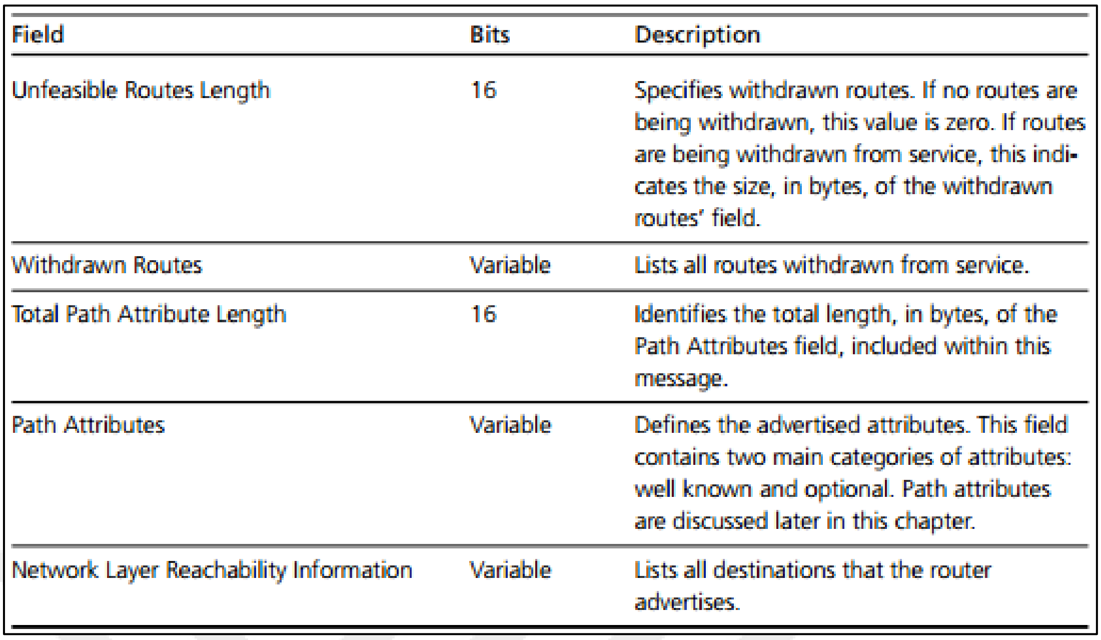

BGP-4
Message
Header
Marker
Length
Type = 2
Unfeasible...
...Routes Length
Withdrawn Routes (variable length)
Total Path Attribute Length
Path Attribute (variable length)
Network Layer Reachability Information (variable length)

**Figure 10.** BGP Update Message [2]

Unfeasible Routes Length, Withdrawn Routes, Total Path Attributes, Path Attributes, and Network Layer Reachability Information (NLRI) are included in the BGP Update Messages which are type 2 messages. Table 2 presents the BGP Update Message Fields:

**Table 2**. BGP Update Message Fields



| Field | Bits | Description |
|---|---|---|
| Unfeasible Routes Length | 16 | Specifies withdrawn routes. If no routes are being withdrawn, this value is zero. If routes are being withdrawn from service, this indicates the size, in bytes, of the withdrawn routes' field. |
| Withdrawn Routes | Variable | Lists all routes withdrawn from service. |
| Total Path Attribute Length | 16 | Identifies the total length, in bytes, of the Path Attributes field, included within this message. |
| Path Attributes | Variable | Defines the advertised attributes. This field contains two main categories of attributes: well known and optional. Path attributes are discussed later in this chapter. |
| Network Layer Reachability Information | Variable | Lists all destinations that the router advertises. |

The BGP Notification Messages are sent when an error occurs. For instance, after establishing a BGPv4 TCP connection, if a router receives an Open message with another version of BGP the Notification message is sent and the TCP connection closes. When the Notification message is sent, the TCP connection is always terminated. Figure 11 shows the BGP Notification Message:

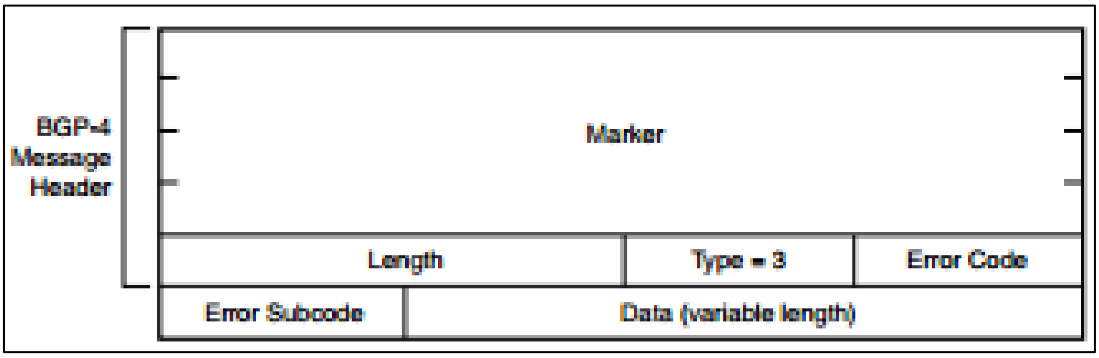
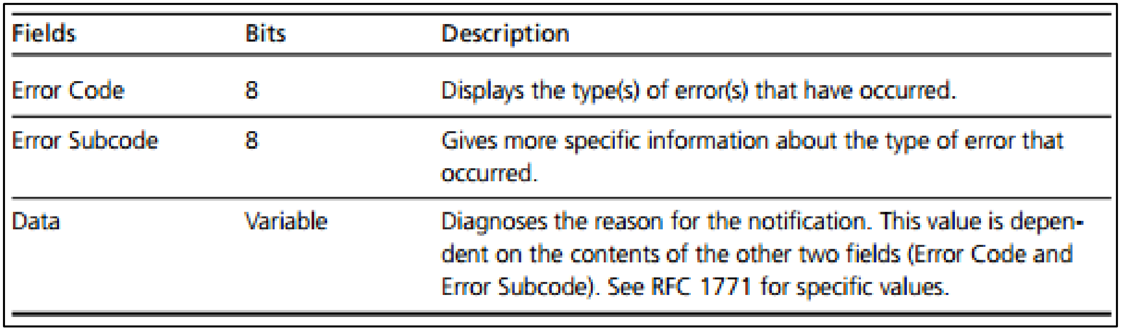


**Figure 11.** BGP Notification Message

The BGP Notification Message is a type 3 message that contains Error Code, Error Subcode, and Date fields. The BGP Notification Message Fields are presented in Table 3:

**Table 3.** BGP Notification Message Fields



| Fields | Bits | Description |
|---|---|---|
| Error Code | 8 | Displays the type(s) of error(s) that have occurred. |
| Error Subcode | 8 | Gives more specific information about the type of error that occurred. |
| Data | Variable | Diagnoses the reason for the notification. This value is dependent on the contents of the other two fields (Error Code and Error Subcode). See RFC 1771 for specific values. |

### 2.4.5. Finite State Machine and BGP States

BGP Finite State Machine summarizes BGP states [6]. It illustrates all the BGP sessions from the Idle to the Established state with certain messages, Figure 12 illustrates this process:

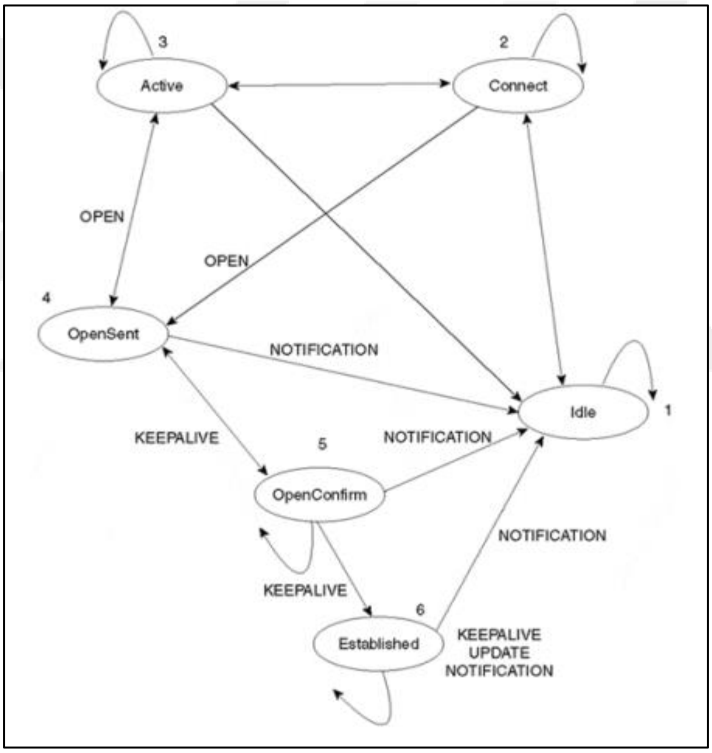


**Figure 12.** BGP Finite State Machine [22]

- **IDLE State**

Initially, BGP is in the *IDLE* state before establishing the BGP session. When the network administrator carries out the BGP configuration on a router BGP creates a start event. Then BGP initializes its resources, resets the *ConnectRetry* timer, starts the TCP transport connection, and listens if there is another start event which is initiated by another peer. If this operation is successful, BGP goes to the connect state. If there is an error, then BGP falls back to the idle state again and the router waits for the *ConnectRetry* timer to start the process again.

- **Connect State**

In this state, BGP waits for the TCP connection. If the TCP transport connection is successful, then the BGP state goes to the *OpenSent* state. If the TCP transport connection fails, the BGP state goes to the Active state. In this situation, if the *ConnectRetry* timer expires and if an error occurs before the TCP connection initiates, the BGP state goes back to idle. Otherwise, it attemps to go to the *OpenSent* state.

- **Active State**

In this state, BGP tries to establish a TCP connection with its neighbor. If the connection is successful, BGP resets the *ConnectRetry* timer, sends the OPEN message and goes into an *OpenSent* state. Otherwise, it falls back to the Connect state. Incidentally, BGP listens for whether other peers have started another TCP connection. If a peer attempts to establish a TCP connection with an unexpected IP address, the *ConnectRetry* timer is reset, and the TCP connection is refused. The BGP state remains in the Active state. During these operations, if another event occurs, the BGP state might fall back to the *Idle* state.

- **OpenSent State**

In this state, BGP waits for an Open message. When it is received, the Open message is checked for errors. For instance, the version number and AS number must match with the peer, and in the event of an incompatibility, an error Notification message is sent and the BGP state falls back to the *Idle* state. If the Open message is received without an error, BGP sends the *Keepalive* message, and it goes into the *OpenConfirm* state. Moreover, in this state, BGP checks whether the connection is an Internal BGP or External BGP connection by comparing AS numbers. If an error occurs, the Notification message is sent and BGP falls back to the *Idle* state.

- **OpenConfirm State**

The *Keepalive* message waits to go to the Established state. If a *Keepalive* message is received, the BGP goes into the Established state, and neighbor negotiation is completed. If a Notification message is received, the BGP falls back to the *Idle* state.

- **Established State**

  Established state means that the BGP peer connection has been established. Henceforth, *Update*, *Keepalive* and *Notification* messages can be exchanged. If an error occurs, a *Notification* message is sent and the BGP falls back to the *Idle* state. Additionally, if the hold timer expires, a disconnect notification is received from TCP, or a stop event is received and the BGP falls back to the Idle state again.

### 2.4.6. Path Attributes

BGP path attributes are used to influence best path decisions. Moreover, they keep track of path information, route preferences, next hop values, and aggregation information. A path attribute consists of an attribute type, an attribute length, and an attribute value. Attribute type is divided into 1-byte attribute flags and 1-byte attribute type code [3]. Figure 13 illustrates the Path Attribute:

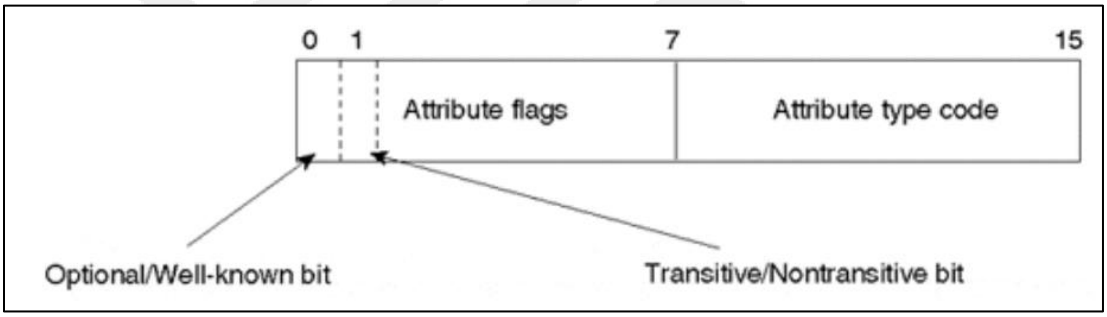


**Figure 13.** Path Attribute Type Format [22]

There are four categories for path attributes, namely well-known mandatory, well-known discretionary, optional transitive and optional nontransitive [2]. The first two bits of the attribute flags hold that information.

The first bit (bit 0) is 0, meaning that the attribute is well known, 1 means that the attribute is optional. The second bit (bit 1) is 0, meaning that the optional attribute is non-transitive; 1 means that the optional attribute is transitive. Additionally, well-known attributes are always transitive. Therefore, the second bit is set to 1 for well-known attributes. The third bit (bit 2) is an optional transitive attribute, and if it is set to 0, the attribute is complete; if it is set to 1, the attribute is partial. The fourth bit (bit 3) is related to the attribute length, and if it is set to 0, the attribute length is 1 byte; if it is set to 1, the attribute length is 2 bytes. The remaining bits (from 4 to 7) of the attribute flags for future use are set to 0.

Well-known mandatory attributes must be involved in BGP update packets. If this attribute is missing, a Notification message is created, and then the BGP session is terminated. The BGP speakers must process these attributes.

Well-known discretionary attributes may or may not be involved in the BGP update packets. If they are involved, every BGP router must recognize that attribute, and the BGP speakers must process these attributes.

If a BGP speaker receives an optional transitive attribute, it does not have to recognize it, and it passes it. When the attribute is not recognized, a transitive flag is searched and if the flag is set, the attribute is transitive. Otherwise, the attribute is optional non-transitive. If the attribute is optional nontransitive, it is not passed to the other BGP peers.

Table 4 below presents the attribute names and type codes:

**Table 4**. Attribute Type Codes [22]

| Attribute Number | Attribute Name | Category/Type Code | Related RFC/Internet Draft |
|---|---|---|---|
| 1 | ORIGIN | Well-known mandatory, Type code 1 | RFC 1771 |
| 2 | AS_PATH | Well-known mandatory, Type code 2 | RFC 1771 |
| 3 | NEXT_HOP | Well-known mandatory, Type code 3 | RFC 1771 |
| 4 | MULTI_EXIT_DISCRIMINATOR | Optional nontransitive, Type code 4 | RFC 1771 |
| 5 | LOCAL_PREF | Well-known discretionary, Type code 5 | RFC 1771 |
| 6 | ATOMIC_AGGREGATE | Well-known discretionary, Type code 6 | RFC 1771 |
| 7 | AGGREGATOR | Optional transitive, Type code 7 | RFC 1771 |
| 8 | COMMUNITY | Optional transitive, Type code 8 | RFC 1997 |
| 9 | ORIGINATOR_ID | Optional nontransitive, Type code 9 | RFC 1966 |

| | | | |
|---|---|---|---|
| 10 | Cluster List | Optional nontransitive, Type code 10 | RFC 1966 |
| 11 | DPA | Destination Point Attribute for BGP | Expired Internet Draft |
| 12 | Advertiser | BGP/IDRP Route Server | RFC 1863 |
| 13 | RCID_PATH/CLUSTER_ID | BGP/IDRP Route Server | RFC 1863 |
| 14 | Multıiprotocol Reachable NLRI | Optional nontransitive, Type code 14 | RFC 2283 |
| 15 | Multıiprotocol Unreachable NLRI | Optional nontransitive, Type code 15 | RFC 2283 |
| 16 | Extended Communities | | draft-ramachandra-bgp-ext-communities-00.txt, "work in progress" |
| 256 | | Reserved for development | |

- **Origin**

The *Origin* attribute is a well-known mandatory attribute that defines how the route was learned and placed into the routing table [2]. If there are multiple routes, the origin of the route can be selected with that attribute. The *origin* attribute can take three values, namely IGP, EGP, and Incomplete. If the routing update comes from the same AS, the *Origin* takes the “i” value in the BGP table, which makes it the origin IGP. If the routing update is learned from outside the AS, the *Origin* takes the “e” value in the BGP table, which makes it the origin EGP. If the routing update is learned via the redistribution of a route, it makes the origin Incomplete and the *Origin* takes the “?” value in the BGP table.

- **AS_Path**

The *AS_Path* attribute is a well-known mandatory attribute which prevents the BGP network from routing loops [2]. When a routing update passes through any ASs, each AS marks the routing update’s *AS_Path* with their unique numbers. Therefore, ASs rejects the routing update when they receive the same older routing update (Figure 4).

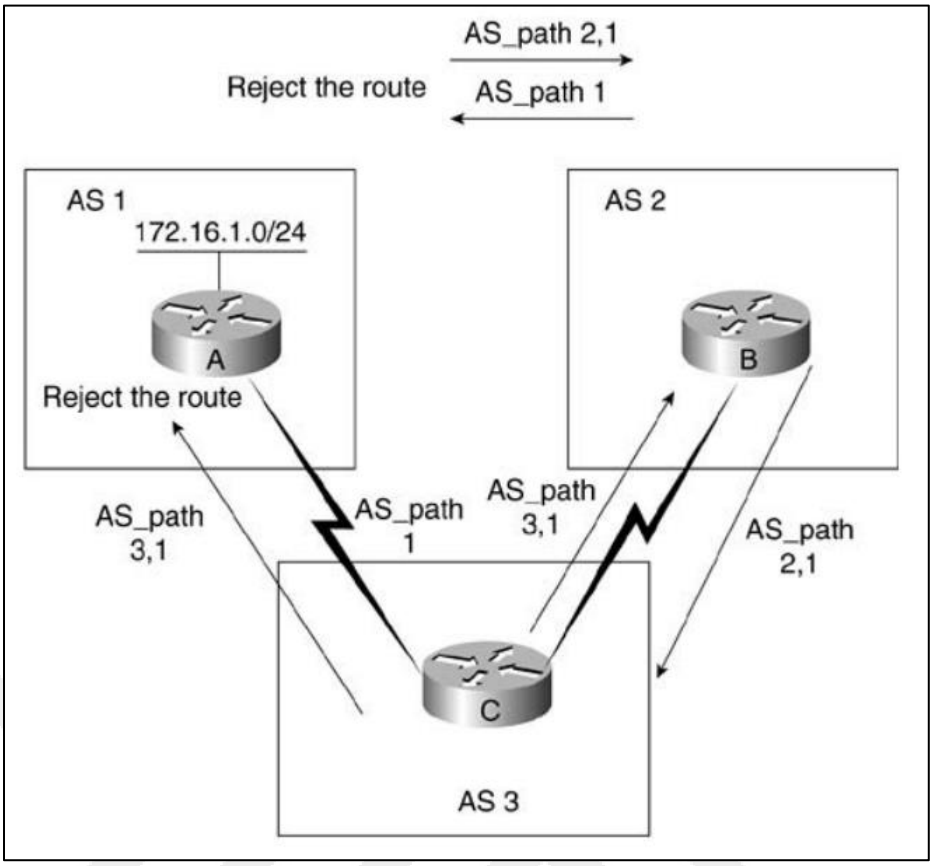


**Figure 14.** AS_Path Attribute [3]

In the figure, AS1 advertises the 172.16.1.0 route to AS2 and AS3. The *AS_Path* attribute of this route receives the {1} value. When AS2 and AS3 advertise this route to AS1, they mark the *AS_Path* attribute with their unique values. In this scenario, the *AS_Path* attribute receives {2,1}, and {3,1}. Consequently, if AS1 receives that update again, it recognizes the update from the *AS_Path* attribute values and rejects the update.

- **Next_Hop**

*Next_Hop* is a well-known mandatory attribute that identifies the IP address of the advertising router [2]. If the advertising router is in a different AS, *Next_Hop* will be the IP address of the peer in the other AS. If the advertising router is in the same AS, *Next_Hop* will be the IP address of the advertising router in the same AS.

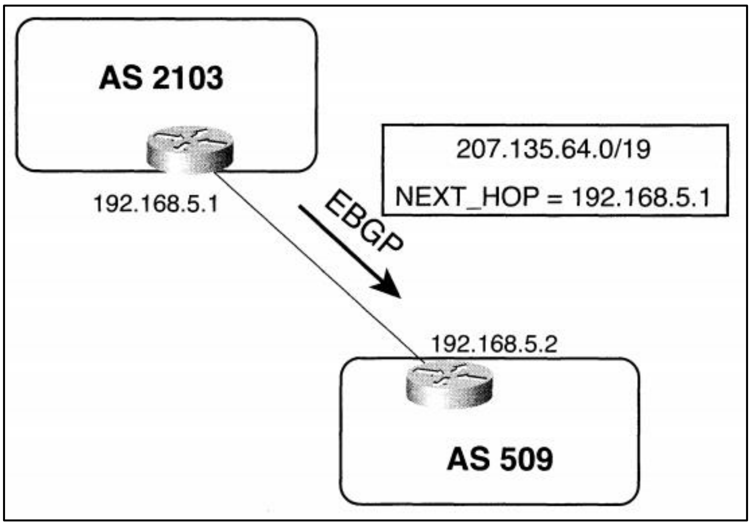


**Figure 15.** Next_Hop Attribute - 1 [6]

According to Figure 15, the advertising router is in AS2103, and the other router uses the IP address of the advertising router to access the network behind it.

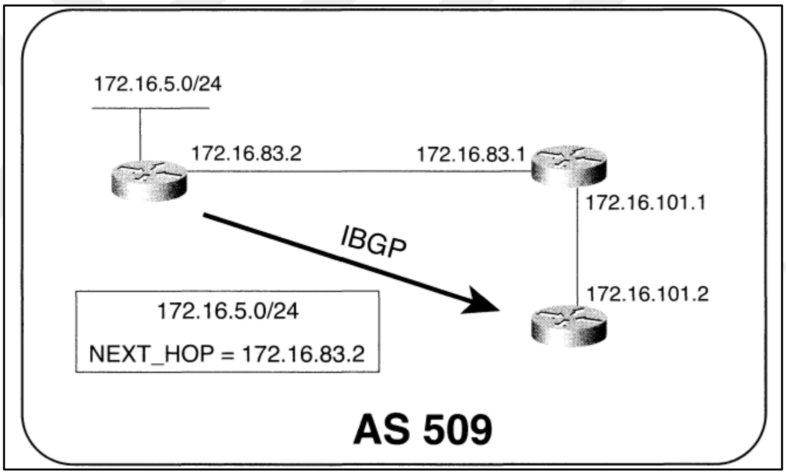


**Figure 16.** Next_Hop Attribute - 2 [6]

According to Figure 16, every router is in the same AS, and the receiving router knows the originating router's IP address to access the network behind it.

- **Multi_Exit_Discriminator**

*Multi_Exit_Discriminator (MED)* is an optional nontransitive attribute [2]. If there are multiple ingress points to reach an AS, the *MED* attribute informs the other AS about the preferred route. *MED* attribute is equal to the metric value. Routers prefer the route that a router has lower *MED* value.

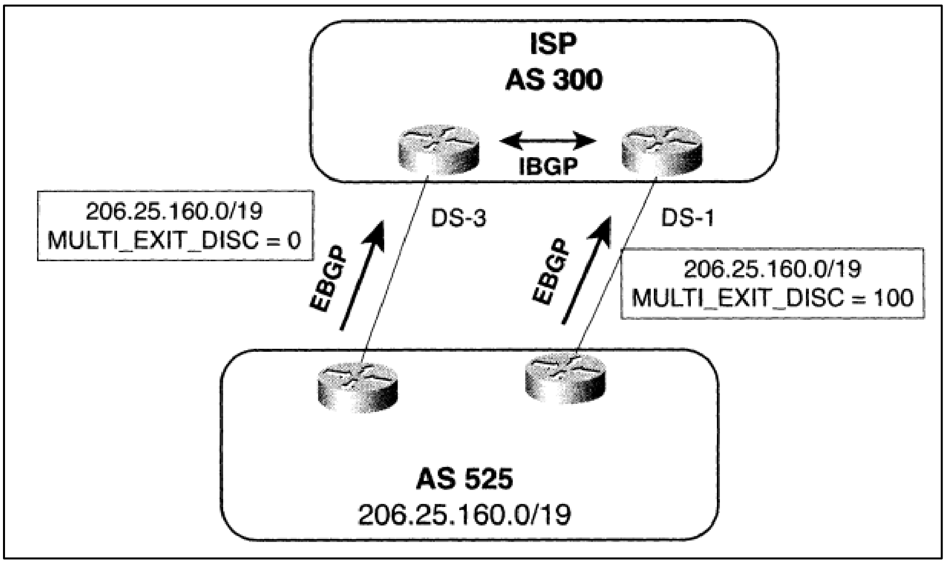


**Figure 17.** MED Attribute [6]

In the Figure 17, AS525 will prefer the DS-3 router to reach AS300 as it has a lower *MED* value.

- **Local_Pref**

*Local_Pref* is a well-known discretionary attribute [2]. It is not carried to other ASs, and it is used in the local AS. If there are multiple exit points to the outside from the local AS, the *Local_Pref* attribute is used to select a route. A Higher *Local_Pref* value is preferred. The *Local_Pref* attribute is illustrated in Figure 18. Router A and router B receive advertisement packets from router C and router D about the 172.16.1.0/24 network in AS200. When a router from AS100 wants to reach the 172.16.1.0/24 network, it will use router B as it has a higher *Local_Pref* value.

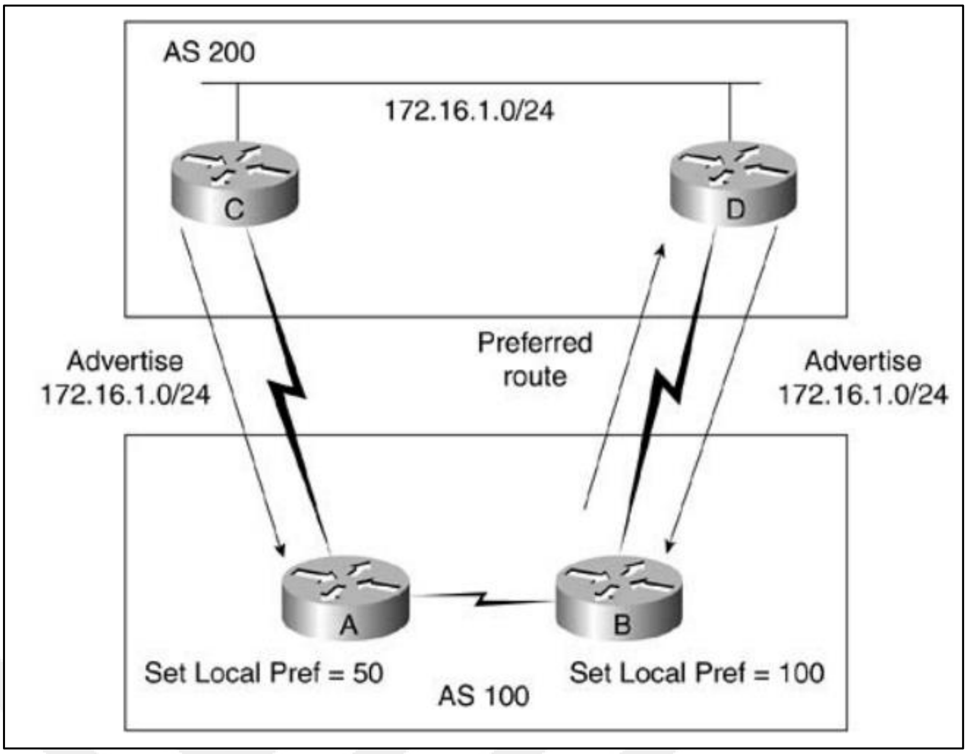


**Figure 18.** Local_Pref Attribute [3]

- **Atomic_Aggregate**

*Atomic_Aggregate* is a well-known discretionary attribute [2] that is used when route summarization is configured. BGP summarizes more specific routes into less specific aggregated routes. When this occurs, some of the routes can be lost. *Atomic_Aggregate* attribute prevents routers from losing some of this path information.

- **Aggregator**

*Aggregator* is an optional transitive attribute [2] that is used only if the *Atomic_Aggregate* attribute is set. *Aggregator* detects where the route summarization originates and identifies its AS.

- **Community**

*Community* is an optional transitive attribute [2]. With this attribute, some of the routers can be defined as a group (community), and a number of the attributes or policies can be applied to that group, including *Local_Pref* and *MED* attributes. Moreover, there are three predefined community attributes, namely no-export, no-advertise, and internet. These attributes are illustrated in the following figures:

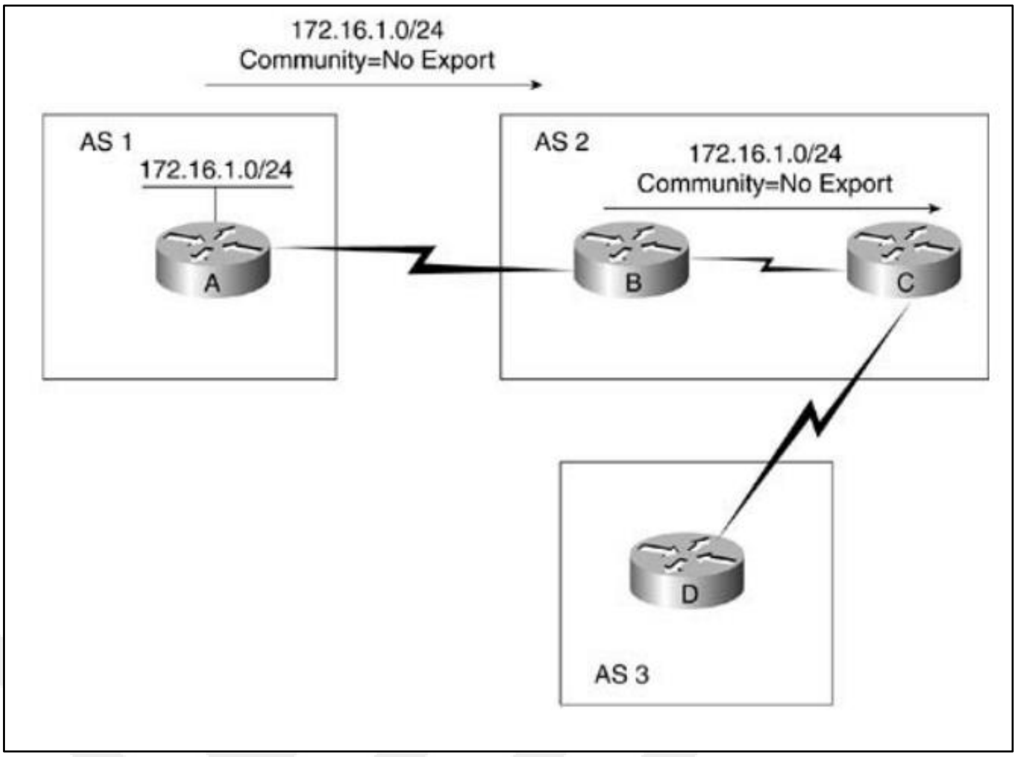


**Figure 19.** Community with No Export [3]

In Figure 19, the 172.16.1.0/24 network is advertised from AS1 to AS2 with the no-export community. The 172.16.1.0/24 network is propagated within AS2; however, it is not sent to AS3. The no-export community attribute prevents the update from being advertised to AS3.

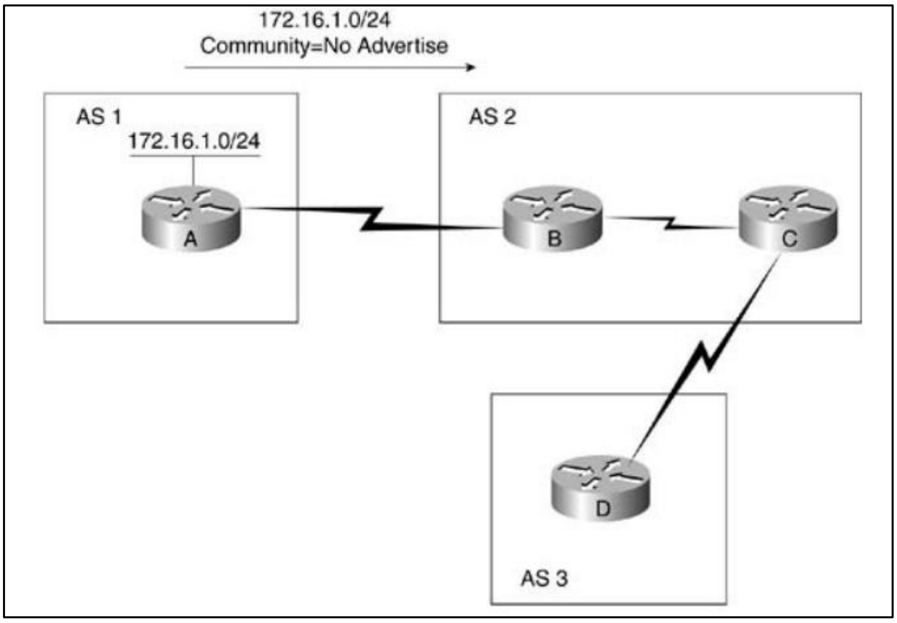


**Figure 20.** Community with No Advertise [3]

In Figure 20, the 172.16.1.0/24 network is advertised from AS1 to AS2 with the no-export community. Router B cannot send this update to any other router because of the no-advertise community.

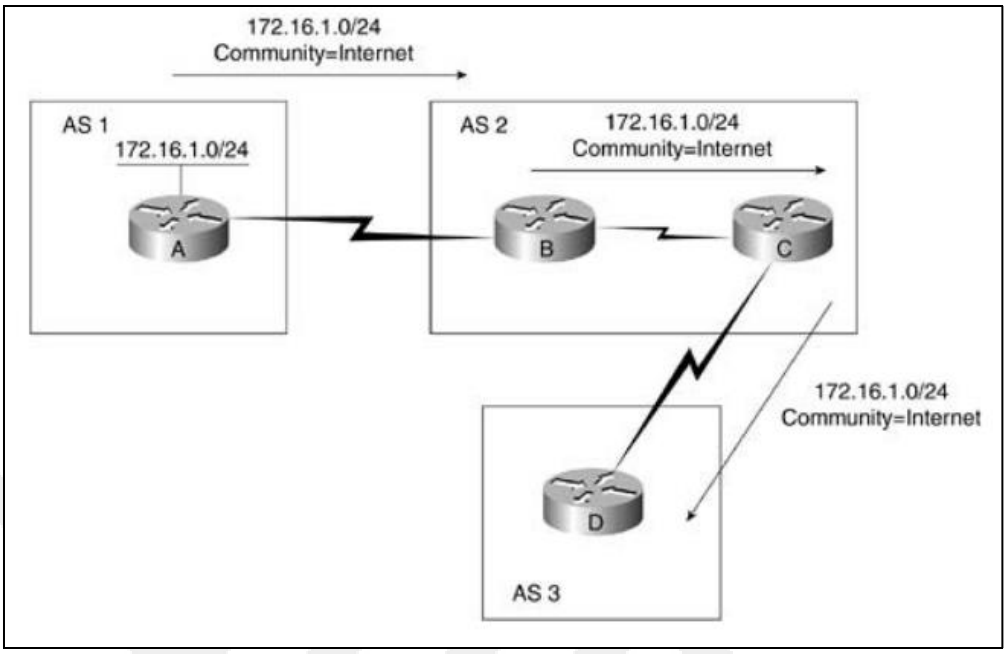


**Figure 21.** Community with Internet [3]

In Figure 21, the 172.16.1.0/24 network is advertised from AS1 to AS2 with the Internet community. AS2 will propagate the update to every Internet community member. There are no limitations for the Internet community.

- **Originator_ID**

*Originator_ID* is an optional nontransitive attribute [2]. It is used only if route reflectors are configured, and it is a loop prevention mechanism. *Originator_ID* is the router ID of the originator route, and route reflectors create them. When route reflectors see their own *Originator_IDs* in an update, they assume that there is a routing loop and ignore the route.

- **Cluster List**

*Cluster List* is an optional nontransitive attribute [2]. It is used only if the route reflectors are configured. This is a loop prevention mechanism. The BGP routers add a cluster ID to updates, which produces the *Cluster List*. The list includes all cluster IDs (all the router’s IDs in the path) through a path. When a route reflector receives an update, and if it sees its local cluster ID the route is ignored.

### 2.4.7. BGP Path Selection

BGP routers make routing decisions according to a number of rules that are stored in a BGP Routing Information Database (RIB). The RIB is divided into three parts: *Adj-RIBs-In, Loc-Rib, Adj-RIBs-Out. Adj-RIBs-In* stores the routing information that come from peers as routing updates, which are feasible routes. *Loc-Rib* stores the routing information that the BGP speaker’s local routing policies as an ingress direction to the router, and *Loc-Rib* applies its policies to *Adj-RIBs-In’s* routes. *Adj-RIBs-Out* stores the routing information that the BGP speaker’s routing information to be advertised to its peers [6].

BGP path selection consists of three phases. In Phase 1, the BGP speaker calculates the preference degree for an update that comes from its peer. In Phase 2, the best route is selected from all available routes to a certain destination, and then it is installed to *Loc-RIB*. In Phase 3, appropriate routes are installed to the *Adj-RIBs-Out* to be advertised to its peers.

According to BGP’s Path Attributes, the path selection process is performed in the following order [3]:

1. The route selected is that which has the highest *Weight* value (*Weight* attribute is a Cisco specific attribute; and can be thought of as a metric value).
2. If the weights are equal, the highest *Local_Pref* is selected.
3. If the *Local_Pref* value is equal, the route is preferred to that which was originated locally.
4. If *Local_Pref* is identical and no route is originated, the *AS_Path* that is preferred is that which has the shortest value.
5. If the *AS_Path* length is equal, the lowest Origin code is preferred.
6. If the *AS_Path* is identical, the lowest *MED* is preferred. *MED* is only valid for the routes that are in the same AS.
7. If the routes have an identical *MED* value, external paths are preferred over internal paths.
8. If the routes are still equal, the route with the shortest path (lowest IGP metric to the Next_Hop) to the BGP, *Next_Hop* is selected.

9. If still there is an equality and maximum-paths command (known as multipath) activated, the route is load balanced with equal cost in the *Loc-RIB.*
10. If multipath is not enabled, the lowest BGP *router ID* (lowest IP address) is preferred.

### 2.5.Introduction to MPLS

MPLS (Multiprotocol Label Switching) uses labels to forward packets. To forward a packet, MPLS looks at the labels that are attached to the IP Packets, instead of the destination IP address [7]. MPLS blends Layer 2 switching technologies and Layer 3 routing technologies to increase stability and performance in a network [8].

In an MPLS network, routers are known as LSRs (Label Switched Router) and the paths between LSRs are called LSPs (Label Switched Path). When an IP packet reaches an MPLS network, the first router that takes the packet is called the Edge LSR and it assigns a label to the packet and forwards it to the next LSR. LSRs do not perform routing lookups; instead, they forward packet based labels. Each LSR in the MPLS domain adds their own label when they receive a packet. At the end of the LSP, an Edge LSR receives the packet again and it removes the label; then it performs a routing lookup to forward the packet (as illustrated in Figure 22).

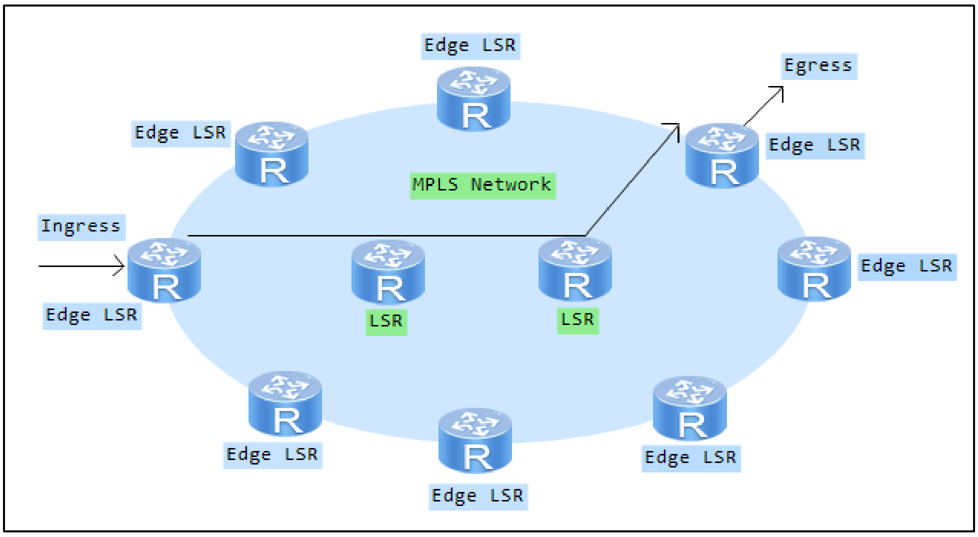


**Figure 22.** An MPLS Network [8]

Traditional IP routing has a number of limitations rather than MPLS. Every device in a network distributes routing information and because of the routing protocols, packet forwarding is carried out based only on the destination address and every hop in the network performs a routing lookup [9]. These drawbacks create performance and scalability issues. MPLS removes routing lookups; therefore, with the exception of Edge LSRs, many of the devices in the MPLS domain do not perform the routing lookup. The routing lookup affects network performance because each router shares their forwarding tables with other routers, and a forwarding table can be extremely large in a large network.

MPLS has the advantage of eliminating BGP based routers in a service provider network. In a BGP network, the customer side runs BGP in the routers and the service provider side runs BGP in its routers. The service provider routers need to hold prefix information of the customer in addition to the prefixes of the Internet, which can cause an enormous routing table cost in each router, thereby causing the routers to be insufficient in memory and CPU resources. Additionally, the performance of the network can be reduced significantly. If MPLS is used at the service provider side, only edge routers of the service provider must run BGP. MPLS forwards packets based on the labels, so there is no need for routing lookup on MPLS routers. In the MPLS domain, LSRs inform other routers about how to go to the Edge LSR. Edge The LSR holds the BGP information; consequently, any of the routers should run BGP in the MPLS domain except for Edge LSRs [7].

## 2.6. MPLS VPN

VPN technology separates and interconnects different networks. Some organizations can have large enterprise networks and in such scenarios, different VPNs need to be interconnected. Moreover, these connections may need to be performed over the Internet. MPLS VPNs make this possible (Figure 23).

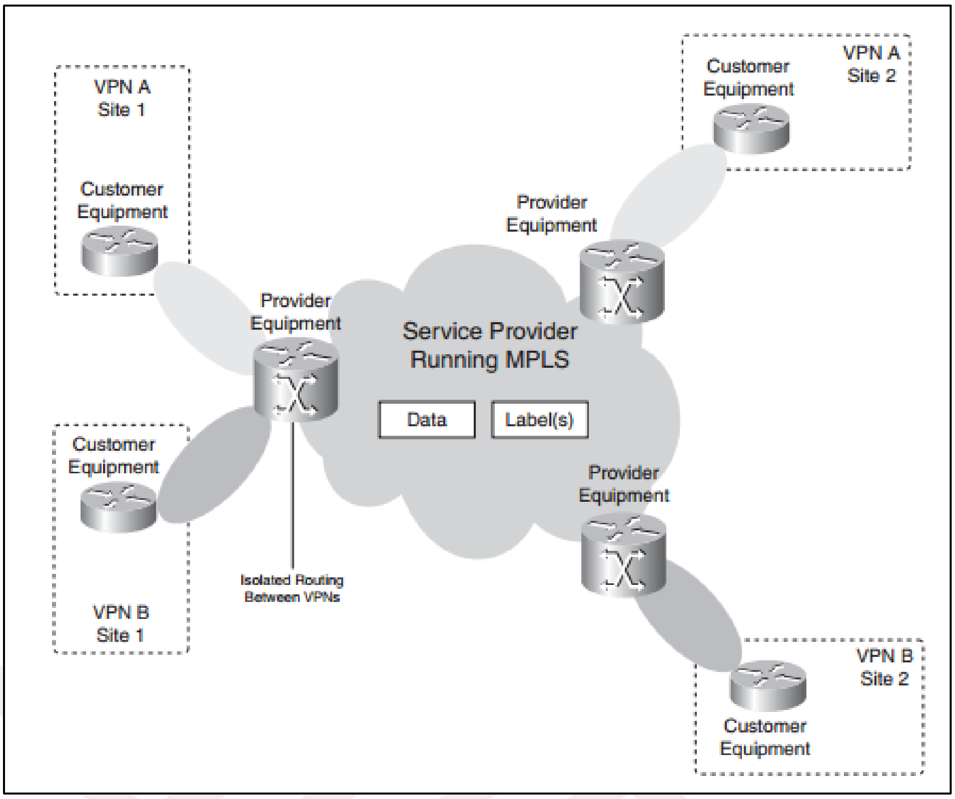
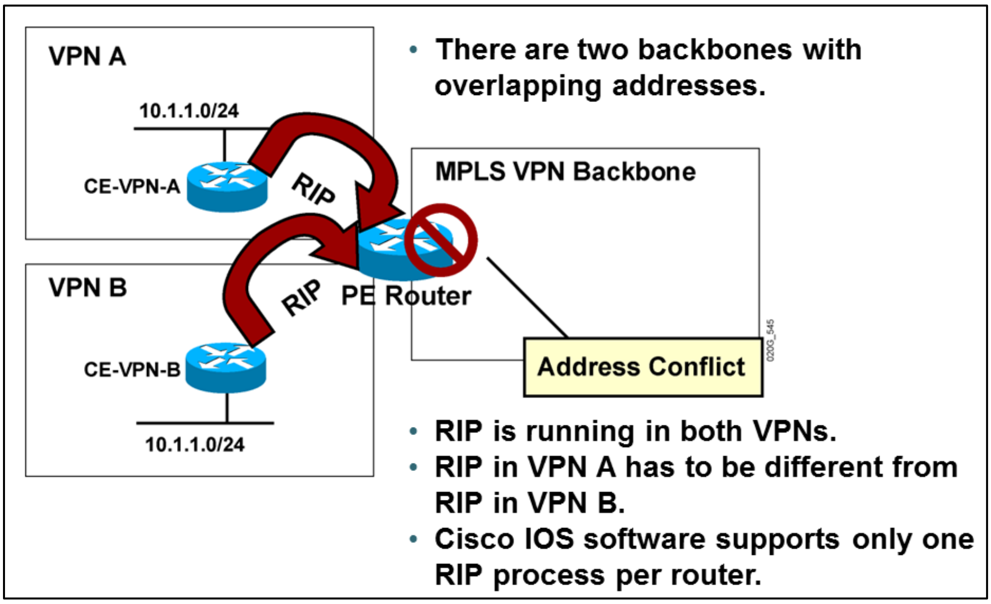


**Figure 23.** An MPLS VPN Model [7]

In an MPLS VPN, an organization, devices of a customer are named as C (customer router) and CE (customer edge router), service provider devices are named as P (provider router) and PE (provider edge router). All of these must support MPLS technology. These routers accomplish MPLS VPN tasks with VRF technology.

### 2.7. VRF (Virtual Routing Forwarding)

VRF is the routing and forwarding instance for a set of sites with identical connectivity requirements [7]. VRF provides its own routing table other than a global routing table. Therefore, the IP addresses of customer networks can be carried separately. Different customer networks may have identical IP addresses as they are private. However, they may need to be carried via Provider routers (P and PE). VRF prevents possible engagement of a private network’s IP addresses. Provider and Customer routers have independent routing tables for each VRF.



- There are two backbones with overlapping addresses.

- RIP is running in both VPNs.
- RIP in VPN A has to be different from RIP in VPN B.
- Cisco IOS software supports only one RIP process per router.

**Figure 24.** VPN process without VRF [23]

Figure 24 illustrates the VPN process with a traditional routing protocol. There is a conflict with IP addresses due to the router not being able to separate them. Like Cisco routers, different vendors avoid this conflict with VRF. A VPN interface can be a physical, sub-interface, or loopback interface, each of which is associated with a VRF, PE and CE router interface that can have only one VRF. Thus, VRFs are separated, and this means routing tables are separated.

The "ip vrf" command creates the VRF; then the "ip vrf forwarding" command must be inserted under the related interface for association with the VRF and the interface. This is a small example of it:

```
ip vrf Cankaya_Merkez // TO CREATE A VRF

interface FastEthernet1/0.100
 ip vrf forwarding Cankaya_Balgat // TO INCLUDE THIS INTERFACE
TO RELATED VRF
 ip address 100.99.88.21 255.255.255.252
!
```

The VRF routing table can be checked by "show ip vrf ***" command.

BGP takes IPv4 prefixes from the customer side, and carries it to the other side of the customer via the provider side. When there are multiple customers, the IPv4 prefixes may overlap. Consequently, the RD (route distinguisher) separates each of the VRF prefixes. They can be used under VRFs and they have two formats: “AS number:nn” and “IP address:nn”. The following are examples of VRFs:

```
ip vrf Cankaya_Merkez // TO CREATE A VRF
 rd 65130:2 // TO SPECIFY THE VRF AND TO MAKE THE ROUTES OF
THAT VRF UNIQUE
```

RDs indicate VRF prefixes uniquely. The communication of the same customer’ sides can be performed with RDs. What if there is a requirement for communication between different customers? RTs (route target) make this possible. RTs state which routes will import from a PE to a VRF or which routes will export from a VRF to a PE. The following is an example of the use of RTs:

```
ip vrf Cankaya_Merkez // TO CREATE A VRF
 rd 65130:2 // TO SPECIFY THE VRF AND TO MAKE THE ROUTES OF
THAT VRF UNIQUE
 route-target import 65130:130 // TO LEAK THE ROUTES FROM
OTHER VRF's TO THAT VRF
!
ip vrf Cankaya_Balgat // TO CREATE A VRF
 rd 65130:1 // TO SPECIFY THE VRF AND TO MAKE THE ROUTES OF
THAT VRF UNIQUE
 route-target export 65130:130 // TO LEAK THE ROUTES FROM THAT
VRF TO OTHER VRF's
!
```

To sum up, the routes in an MPLS VPN network, known as VPNv4 routes, can look similar to people in a city; a VRF that can look similar to a bus; and PE and CE can look similar to stations. With these images, BGP can be seen as a municipality. Figure 25 below illustrates such a scenario:

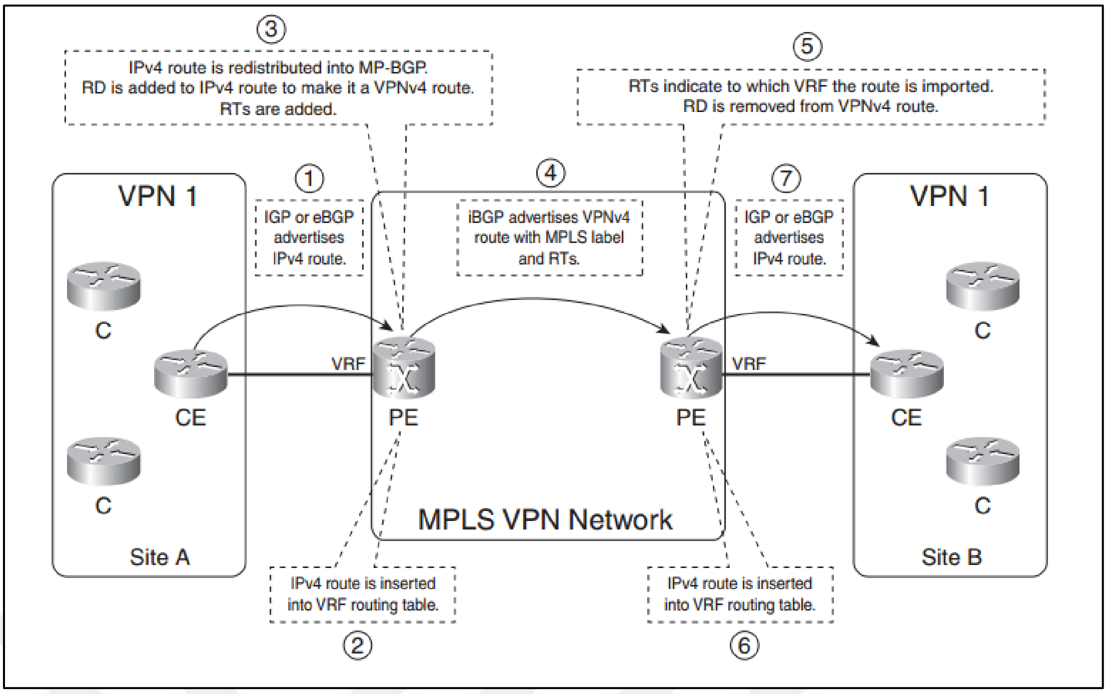

③
IPv4 route is redistributed into MP-BGP.
RD is added to IPv4 route to make it a VPNv4 route.
RTs are added.
⑤
RTs indicate to which VRF the route is imported.
RD is removed from VPNv4 route.
VPN 1
①
IGP or eBGP advertises IPv4 route.
④
iBGP advertises VPNv4 route with MPLS label and RTs.
⑦
IGP or eBGP advertises IPv4 route.
VPN 1
C
C
CE
VRF
PE
PE
VRF
CE
C
C
MPLS VPN Network
Site A
Site B
IPv4 route is inserted into VRF routing table.
IPv4 route is inserted into VRF routing table.
②
⑥

**Figure 25.** Route propagation in an MPLS VPN Network [7]

## 2.8.Related Works

There are numerous studies on BGP in the literature. BGP has many features and attributes that can be studied. BGP is a path vector protocol and there have been many studies about that particular functionality. Furthermore, the security of BGP has been attracting much attention from researchers. Some of the related studies are examined below.

Kotronis et al. proposed a framework that reduces the convergence time of BGP [10]. In order to do that so, they combined multiple ASs as clusters wherein the routing decision of clusters is made centrally. Their objectives are “to support hybrid BGP-SDN experiments with multiple ASs using real router software, and to demonstrate the effect of centralization on IDR (inter-domain routing) convergence times” [10]. Moreover, they endeavored to show the effect of centralized routing decisions on IDR (inter domain routing) convergence time. An SDN controller was designed to complete the experiments. Experimental results showed that IDR centralization improved the convergence time for small SDN cluster deployments. Additionally, they tested the convergence time for a route withdrawal, which yielded results such that the convergence time was linearly reduced for that scenario. However, route failover and announcement scenarios were not successful [10].

When the BGP is misconfigured its routing policies could be conflicted, which can cause persistent route flapping. Consequently, D. Zhang et al. “presented an automated tool for verifying the convergence property of BGP by using the SPIN model checker” [11]. They developed a SPIN library with unspecified routing policies for a specific BGP instance, which can be used by users to create a network topology to test their scenarios. The user provides the topology and a proper routing policy and the SPIN library simulates the BGP instance and gives the verification results. According to their work, different ASs can have their own routing policies. Therefore, their convergence properties can be divergent. They compared the BGP convergence on different topologies. As a result, BGP with default routing policies was shown to be convergent for fewer than 16 nodes for any BGP network topology.

BGP routers trust other BGP routers and they do not have mechanisms to detect malicious updates and routes. Rough BGP routers can announce their IP address blocks and pull traffic to themselves. Thus, J. Yun and J. Song proposed Secure AS_PATH BGP (SAPBGP) to monitor BGP’s AS_PATH attribute to prevent AS path hijacking. In their method, “SAPBGP constructs its own policy based database by collecting RIPE NCC repository and checks the AS_PATH attribute in BGP update messages whether the ASs listed in the AS_PATH attributes are actually connected or not” [12]. To accomplish this, they compared live BGP streams to their policy-based database by collecting the RIPE NCC repository. Therefore, they validated the AS_PATH attributes to handle a huge number of live BGP streams in real time [12]. SAPBGP can compare BGP updates to its policy based database on a daily basis.

Z. Duan et al. analyzed BGP’s convergence mechanism [13]. The convergence of BGP may be slow due to path exploration, and Z. Duan et al. developed a path vector algorithm to reduce BGP’s convergence time. After a failure of BGP, the configured network convergence can take nearly fifteen minutes due to the exploring of the paths. There can be enormous numbers of paths and each of them should be explored after a failure. When a path is withdrawn, dependency on that path leads to slower convergence. Consequently, Z. Duan et al. proposed a simple and novel mechanism, namely forward edge sequence numbers, to annotate routing updates with path dependency information so as to address the path exploration problem effectively. The path exploration problem occurs after a failure in the BGP network. During convergence, BGP needs to detect the withdrawn routes and paths, and as a result, BGP convergence time stretches. Z. Duan et al. developed a path vector routing protocol, which they called EPIC [13], which is an enhanced path vector protocol that decreases the path exploration problem. According to their test results, BGP's convergence time can be reduced, and their method is more successful than the other existing solutions.

X. Zhao et al. examined how transient routing loops occur in BGP, and the major factors [14]. They analyzed the duration of the loops by creating them and examining the reasons for them. They saw that the duration closes the BGP's convergence time and BGP's Minimum Route Advertisement Interval Timer (MRAI) value. Moreover, they analyzed four BGP enhancement mechanisms: Sender Side loop detection, Withdrawal Rate Limiting (WRATE), Assertion approach, and Ghost Flushing. According to their work, routing loops may occur during BGP's convergence period, which they illustrated with a 110-node BGP topology. In that topology, the convergence time of BGP was 527 seconds, and 86% of the packets encountered a routing loop. Their simulation results showed that, "both the Assertion and Ghost Flushing approaches are effective in speeding up routing convergence and reducing transient loops, however WRATE enhancement may significantly lengthen transient loop duration compared to the standard BGP without WRATE" [14]. Additionally, they showed that BGP's poison reverse attribute, which prevents loops by blocking the previous paths of the packets that came previously, may be inadequate when detecting and preventing loops as they may not be detected until the packet reaches every node [14].

S. Singh et al. proposed a method to solve security issues on BGP [15]. Attacks may occur for BGP, such as during the TCP connection stage, the misconfiguration of BGP routers, and whenever wrong AS number announcements occur. Rough BGP speakers may advertise wrong routes and AS numbers to redirect other BGP speakers to undesired destinations. This can cause the Internet connection to go down or it can slow down network traffic. To prevent these kinds of attack S.

Singh et al. provided a solution. According to their solution, BGP speakers perform a one-time authentication process during TCP session establishment. To do this, they used a symmetric key to secure TCP session establishment. Before establishment of TCP sessions between BGP peers, “each BGP router generates a private key using the most secure cyclic shifting algorithm” [15]. In cyclic shifting algorithms, both BGP speakers agree first on large prime numbers and on using complex operations on prime numbers to generate symmetric keys; then SHA-1 (secure hash algorithm) generates hash codes for the same. Only the hash code of the symmetric key is exchanged through the optional field of the OPEN message during the TCP session establishment process. At the receiving end, the same operations are performed, and hash code is verified with a receiving the hash code. If both hash codes match, then a secure TCP session between BGP peers is established [15]. Their work showed that the overall delay is reduced to make BGP secure and reliable. However, the establishment of a TCP connection is faced with a 1.5ms delay due to key generation and verification. When the key is generated and the TCP connection is established, malicious attackers cannot inject false routes and AS numbers. “When session terminates by BGP speakers and reconnect with neighbors, BGP speaker requires certificate id, private key and hash value of key for new session establishment” [15].

W. Hamzeh and A. Hafid introduced a novel method that separates the BGP routing table on routers [17]. Their method enhances the routing table lookup speed and scalability of the RIB (Routing Information Base). Current BGP routing tables have been growing exponentially. And to prevent this, CIDR was introduced, which aggregates the routes to the most possible single prefix. This slowed the growth of the routing tables for several years; however, it has become insufficient. The recent work on that problem has not offered a distributed architecture for the control plane. According to the IETF ForCES RFC [16], the control plane holds the RIB (the RIB holds reachability information and maps them to the FIB (Forwarding Information Base)), and the forwarding plane forwards the packets. W. Hamzeh and A. Hafid’s work has provided a novel method to partition the BGP routing table at multiple controller cards. Their method is known as DRTP (Distributed Router Table Partitioning). In that method, routing lookups and updates are performed in parallel on those multiple controller cards. The partitioning of the routing table method offers scalability, which in turn provides efficient memory usage, and better routing table lookup speeds. Moreover, their method removes the bottleneck at the BGP RIB with multiple controller cards. “The proposed DRTP increases the scalability of the BGP routing table by making a horizontal division on the prefix length and a vertical division on the prefix range of the entire table and the lookup performance by exploiting parallelism and distributing adequately the load on multiple controller cards. The overhead introduced by DRTP to process the insertion and deletion of prefixes, is minimal” [17]. Their experimental results proved that DRTP improved the performance of the BGP, and the scalability of the BGP routing table.

Narasimhan and Latchman presented an implementation work on BGP route manipulation techniques by using a GNS3 network simulator [14]. Their work illustrates how to build a multi-homed network by using some of the BGP attributes, including Local-preference, AS-path, MED, weight, prefix lists, and community lists. These attributes provided a policy-based route selection. Therefore, a path to an ISP from multiple ISPs is selected according to the policies. Moreover, they discussed Man in the Middle attacks in BGP networks by injecting fake AS numbers.

N. Zidan and M. Hamarsheh described a BGP implementation by using GNS3 network simulator [19]. They compared some of the BGP attributes (Local-preference, AS-path, MED) with no policy to see routing selections in a simple topology. According to their work, routing selection can be manipulated with BGP attributes.

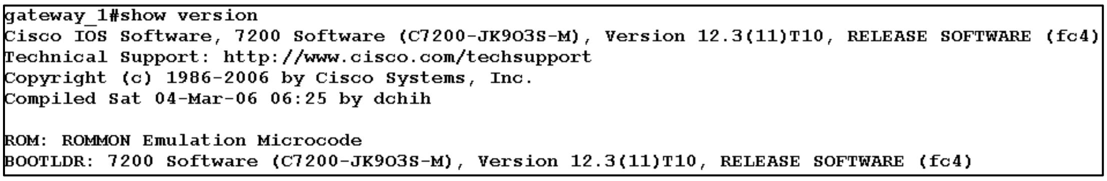
# 3. IMPLEMENTATION AND TESTING

## 3.1. System Overview

BGP is used to carry information over the Internet between different AS's. There is no way to accomplish this with IGP protocols. BGP is a scalable and robust protocol [20] that has many different features to carry information. BGP can be used in huge areas such as universities, government organizations, banks and ISP's (Internet Service Providers). If a BGP is used by an organization, there must be very critical data traffic flow and there cannot be any interruption of traffic. To illustrate, if a bank is subject to traffic interruptions, it can lose large amounts of money. Consequently, building a reliable network topology is extremely important.

To build such a reliable network topology, GNS3 software version 1.5.2 is used. GNS3 is a tool to build, design and test a network in a risk-free virtual environment [21]. The application is free of charge and it can be downloaded from "https://www.gns3.com/software".

In our methodology, we used thirteen virtual routers, two clients and one server to act as the Internet. All of these nodes used Cisco's IOS software. In our test environment, we used the C7200-JK903S-M Version 12.3(11) T10 router software. This can be seen on the virtual router(Figure 26):



```
gateway_1#show version
Cisco IOS Software, 7200 Software (C7200-JK9O3S-M), Version 12.3(11)T10, RELEASE SOFTWARE (fc4)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Sat 04-Mar-06 06:25 by dchih

ROM: ROMMON Emulation Microcode
BOOTLDR: 7200 Software (C7200-JK9O3S-M), Version 12.3(11)T10, RELEASE SOFTWARE (fc4)
```

**Figure 26**. Virtual Router Software Version

Different versions of router software can be used to perform the test. Moreover, the tools and software of different vendors can also be used, including Huawei's eNSP tool and VRP software. Different software versions and the software of different vendors may require different configuration commands. For instance, to execute configuration commands on Cisco routers, the "configure terminal" command must be used. On the other hand on Huawei's routers, the "system-view" command must be used. There are slight differences between the software of different vendors;furthermore, very few differences can be seen in the different software from the same vendor.

In the following sections, the configuration commands and other features that they are used in the testing environment will be explained in detail. Moreover, the network topology, the IP diagram, the router configuration and an explanation of the topology will be given.

## 3.2. BGP Configuration

In this section, some of the BGP features will be explained. These are used in the example of a BGP network topology in this thesis.

- **Neighbor and Neighbor Activate:**

To create a BGP neighborhood with a router or a group of routers in the same or different the AS neighbor command is used under “router bgp”:

```
router bgp 20000
neighbor INTERNAL peer-group // TO CREATE NEIGHBORSHIP WITH A
GROUP
 neighbor INTERNAL remote-as 20000 // TO SPECIFY THE AS NUMBER
OF THE ROUTERS IN THE GROUP
neighbor 100.3.3.3 peer-group INTERNAL // TO INCLUDE THE
ROUTER IN THE GROUP
neighbor 100.5.5.5 peer-group INTERNAL
neighbor 102.99.88.26 remote-as 65130 // TO CREATE BGP
NEIGHBORSHIP
```

To activate the neighbors, “neighbor activate” command is used under the related address family:

```
address-family ipv4 // TO BUILD IPv4 BGP
 neighbor INTERNAL activate // TO ACTIVATE BGP NEIGHBORSHIP
FOR THE ADDRESS-FAMILY
 neighbor 100.3.3.3 peer-group INTERNAL // TO INCLUDE THE
ROUTER IN THE GROUP FOR THE ADDRESS-FAMILY
 neighbor 100.5.5.5 peer-group INTERNAL
neighbor 102.99.88.26 activate // TO ACTIVATE BGP NEIGHBORSHIP
```

- **Router-id:**

To specify the router in a BGP domain, router-id is used. If the command is not used, BGP assigns the loopback interface address of the router as the default router-id, if there is no loopback interface, the highest IP address is used:

```
router bgp 20000
 bgp router-id 100.4.4.4 // IDENTIFIES THE ROUTER IN THE BGP
DOMAIN
```

- **Password:**

A password can be set for BGP members to prevent unauthorized access to BGP traffic. When a password set in a BGP domain, the TCP connection starts between BGP members only if the password matches. Otherwise, the neighborhood cannot be set. The command should be inserted into the neighbor command:

```
neighbor INTERNAL password cankaya // IF A ROUTER IN THE GROUP
DOES NOT HAVE THE PASSWORD, IT CANNOT MAKE THE TCP CONNECTION
```

- **Update-source:**

The Update source command creates a TCP connection with the specified IGP neighbor via that interface. Generally, loopback interfaces is used for such an operation and it is used when there are multiple paths to the same destination:

```
neighbor INTERNAL update-source Loopback0 // MAKES THE IGP
NEIGHBORSHIP VIA LOOPBACK0 INTERFACE
```

- **Next-hop-self:**

A BGP router takes advertised routes from its neighbors, and it may not know how to reach every advertised route. If it has no direct connection to these routes, it cannot add them to its routing table. Now, a neighbor router must distribute these routes. When the “next-hop-self” command is used, the router uses that neighbor to reach the advertised routes. The configuration is performed under “router bgp”:

```
neighbor 100.8.8.8 update-source Loopback0 // MAKES THE IGP
NEIGHBORSHIP VIA LOOPBACK0 INTERFACE
```

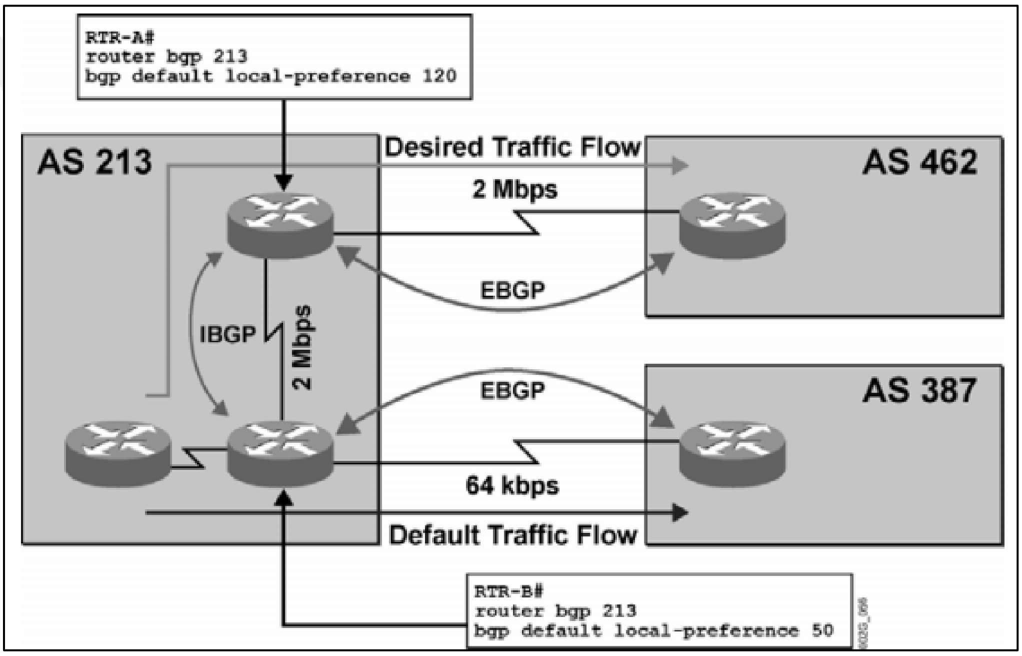
- **Synchronization:**

When the routes from an AS are carried via another AS, IGP needs to learn every route. In older ASs, IGP learned its advertised routes via synchronization. Now, in modern ASs IGPs cannot learn every route that comes from BGP as they can learn only a limited number of routes [20]. Consequently, synchronization should be disabled. A “no synchronization” command should be inserted under “router bgp”:

```
no synchronization // TO PREVENT ADVERTISING UNNECESSARY
ROUTES INTO IBGP
```

- **Local-preference:**

The local preference feature allows routers to give priority to their paths towards outside of the AS. Default local preference value of a router is 100, higher value means preference.



```
RTR-A#
router bgp 213
bgp default local-preference 120
```

```
RTR-B#
router bgp 213
bgp default local-preference 50
```

**Figure 27**. Default local-preference [24]

In Figure 27 above, the default local-preference value of Router A makes other routers to prefer Router A to reach other ASs. Local preference should be configured under the “router bgp”:

```
bgp default local-preference 101 // TELLS THE OTHER ROUTERS
THAT THERE IS A PRIORITY
```

- **Redistribution:**

To redistribute the routes that are learned from other protocols, such as OSPF and static routes, the “redistribute” command is used. With this command these routes are injected into the BGP process. The command should be inserted under “router bgp” or an address-family:

```
redistribute static // ADVERTISES THE STATICALLY CONFIGURED
ROUTES INTO BGP
 redistribute ospf 1 // ADVERTISES THE OSPF ROUTES INTO BGP
```

- **Address-family:**

Address families are used in the BGP configuration as BGP is multiprotocol. BGP carries the unicast or multicast addresses, IPv4 or IPv6 addresses, VPNv4 addresses, etc. Different types of traffic may need to be carried to a particular neighbor. Address-family interfaces accomplish that task. To activate address-family interfaces the “no bgp default ipv4-unicast” command must be inserted under “router bgp” first:

```
router bgp 20000
 no bgp default ipv4-unicast // TO STATE AN ADDRESS-FAMILY
SPECIFICALLY
address-family ipv4 // TO BUILD IPv4 BGP
```

- **Auto-summary:**

If the IGP learned routes are redistributed into BGP, automatic summarization needs to be disabled. Automatic summarization compresses the subnets to their classful boundary. This can lead to wrong routes in the routing table. When automatic summarization is disabled, exact subnets can be seen in the routing table. Automatic summarization can be disabled under “router bgp” or an address-family:

```
no auto-summary // TO ADVERTISE SUBNETS CERTAINLY
```

- **Aggregation:**

Aggregation means summarization in BGP. Aggregation summarizes IGP routes and redistributes them into BGP. Aggregation can be performed under “router bgp” or an address-family:

```
aggregate-address 100.0.0.0 255.0.0.0 summary-only // TO
SUMMARIZE SPECIFIED SUBNET
 aggregate-address 200.0.0.0 255.0.0.0 summary-only
```

- **Route-reflector:**

If an AS does not have a full mesh design, routing loops may occur. Route reflectors prevent routing loops in non-full-mesh IBGP designs. If a route comes from a route reflector client or EBGP, it is sent to every other IBGP neighbors. If a route comes from a non-client, it is sent only to the route reflector clients. Moreover, there can be multiple route reflectors to provide redundancy in an AS, so they form a cluster. Multiple route reflectors are stated with cluster-id.

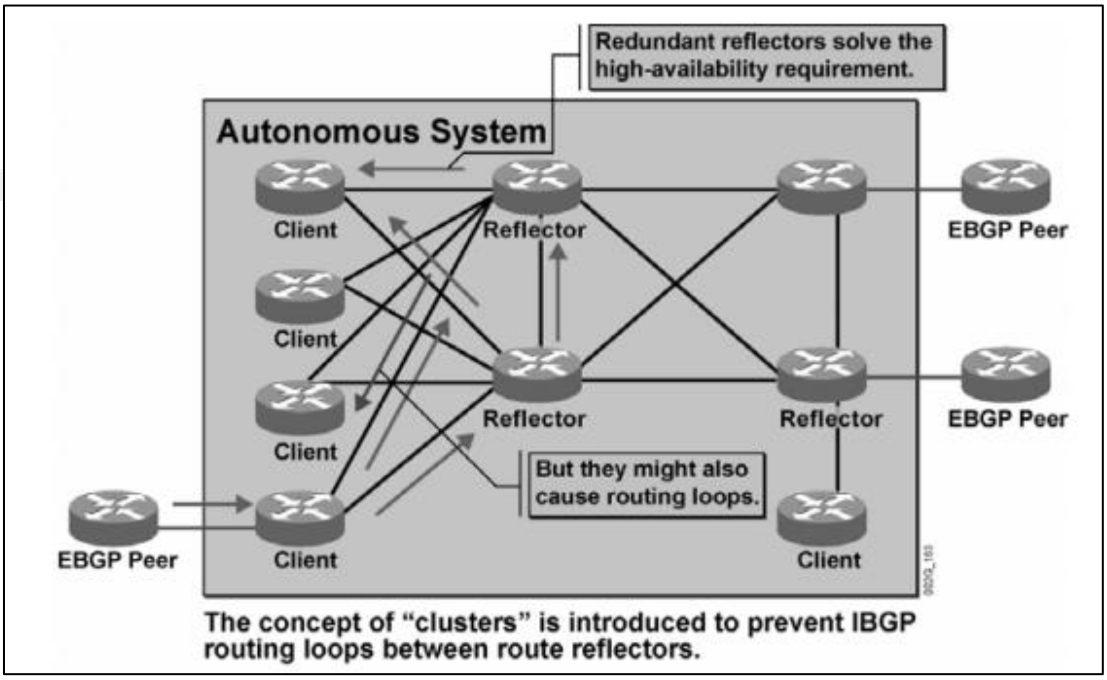

Redundant reflectors solve the high-availability requirement.
Autonomous System
Client
Reflector
EBGP Peer
Client
Client
Reflector
Reflector
EBGP Peer
But they might also cause routing loops.
EBGP Peer
Client
Client
The concept of "clusters" is introduced to prevent IBGP routing loops between route reflectors.

**Figure 28**. Route-Reflectors [25]

The Figure 28 illustrates redundant route reflectors in an AS. Route reflectors must be configured with the same cluster-id, and their clients must be specified on the route reflectors. Route reflector clients need no configuration for that feature. For instance:

```
router bgp 20000
bgp cluster-id 100.5.5.5 // TO INCLUDE MULTIPLE ROUTE-
REFLECTORS TO THE SAME CLUSTER
address-family ipv4 // TO BUILD IPv4 BGP
neighbor INTERNAL route-reflector-client // TO ADVERTISE THE
ROUTES THAT LEARNED FROM IBGP TO OTHER IBGP NEIGHBORS (WITHOUT
THIS ALL THE ROUTERS IN THE IGP MUST BE FULLY MESHED)
```

- **Community:**

Communities are used to assign a number of routing policies to certain groups of routers. When communities are used with a route-map, some routing decisions can be applied to the related community group. In the topology this thesis, communities are only used for working with VPNv4 address families.

```
address-family vpnv4 // TO BUILD VPNv4 BGP
neighbor INTERNAL send-community extended // NECESSARY FOR
WORKING VPNv4 ADDRESS-FAMILY (SOME POLICIES CAN BE APPLIED
WITH COMMUNITIES)
```

- **Soft-reconfiguration:**

If there is a routing policy change in the AS, the BGP session must be reset for the new policy. For that, a hard reset or a soft reset can be applied. A hard reset is not recommended, because it resets the entire BGP session. During the BGP convergence process, negative effects can occur. Therefore, a soft reset offers a better solution. If the soft reset is applied only to routing tables, the BGP session remains unchanged. The “neighbor soft-reconfiguration” command stores any routing policy changes; then the “clear ip bgp” command performs the soft reset.

```
address-family ipv4 // TO BUILD IPv4 BGP
neighbor 200.99.88.14 soft-reconfiguration inbound // TO FORCE
CLEAR BGP PROCESS FOR THAT NEIGHBOR (AFTER A POLICY CHANGE,
CLEARING BGP PROCESS TO TAKE AFFECT IS REQURIED)
```

- **Default-originate:**

“default-originate” command is used for sending default route (0.0.0.0/0) to a specified neighbor. The router advertises default route with that command:

```
neighbor 100.99.88.21 default-originate // INJECTS DEFAULT
ROUTE FROM THAT ROUTER TO THE NEIGHBOR
```

- **Route-map:**

Route-maps are complex access-lists. They start with a route-map name, and then match conditions are applied under the route-maps. Match conditions are performed with statements. A route map can have multiple statements, and each statement has a sequence number. Therefore, multiple match conditions may be applied with a single route map. Route-map conditions are written down under global configuration mode, following this they are applied to a neighbor under the BGP configuration mode. To illustrate:

```
route-map MED permit 10 // TO CREATE A POLICY TO A SPECIFIC
ADDRESS
 set metric 101
router bgp 65130
address-family ipv4 // TO BUILD IPv4 BGP
neighbor 100.99.88.25 route-map MED out // APPLIES THE ROUTE-
MAP POLICY TO EGRESS DIRECTION
```

- **NAT and PAT:**

The NAT function allows the router to translate private IP addresses into public IP addresses. The PAT function redirects connection requests, which come from outside to any related devices in the private network according to related port numbers. To create NAT, first an access-list needs to be defined in order to allow certain IP address blocks. To illustrate:

```
“access-list 199 permit ip any any”
```

The access-list is configured under global configuration mode, and this access-list allows all the IP addresses, that need to NAT translation, according to IP protocol.

```
“ip nat inside source list 199 interface FastEthernet0/0.102
overload”
```

The command above applies the NAT translation to IP addresses that match the condition of access-list 199 for the “FastEthernet0/0.102” interface. Additionally, the “overload” command performs the PAT. For that configuration, the FastEthernet0/0.102 interface should be outbound interface. Furthermore, for the NAT and PAT operations inbound and outbound interfaces need to be cleared. For instance,

```
interface FastEthernet0/0.100 // SUBINTERFACE: IF THERE IS A
NEED TO CONNECT MORE THAN ONE NETWORK FROM SAME PHYSICAL
INTERFACE
 encapsulation dot1Q 100 // DOT1Q OR ISL (ISL IS CISCO
PROTOCOL)
 ip address 100.99.88.22 255.255.255.252
ip nat inside // STATES INBOUND NAT INTERFACE
!
interface FastEthernet0/0.101
```

```
 encapsulation dot1Q 101
 ip address 101.99.88.22 255.255.255.252
ip nat inside // STATES INBOUND NAT INTERFACE
!
interface FastEthernet0/0.102
 encapsulation dot1Q 102
 ip address 102.99.88.22 255.255.255.252
 ip nat outside // STATES OUTBOUND NAT INTERFACE
!
```

### 3.3. Design and System Architecture
### (A Reliable and Redundant BGP Network Topology)

In this section topology of the BGP implementation, IP diagrams and configurations of the routers will be given. All the router configurations and connections are presented at Appendix A.

- **Topology:**

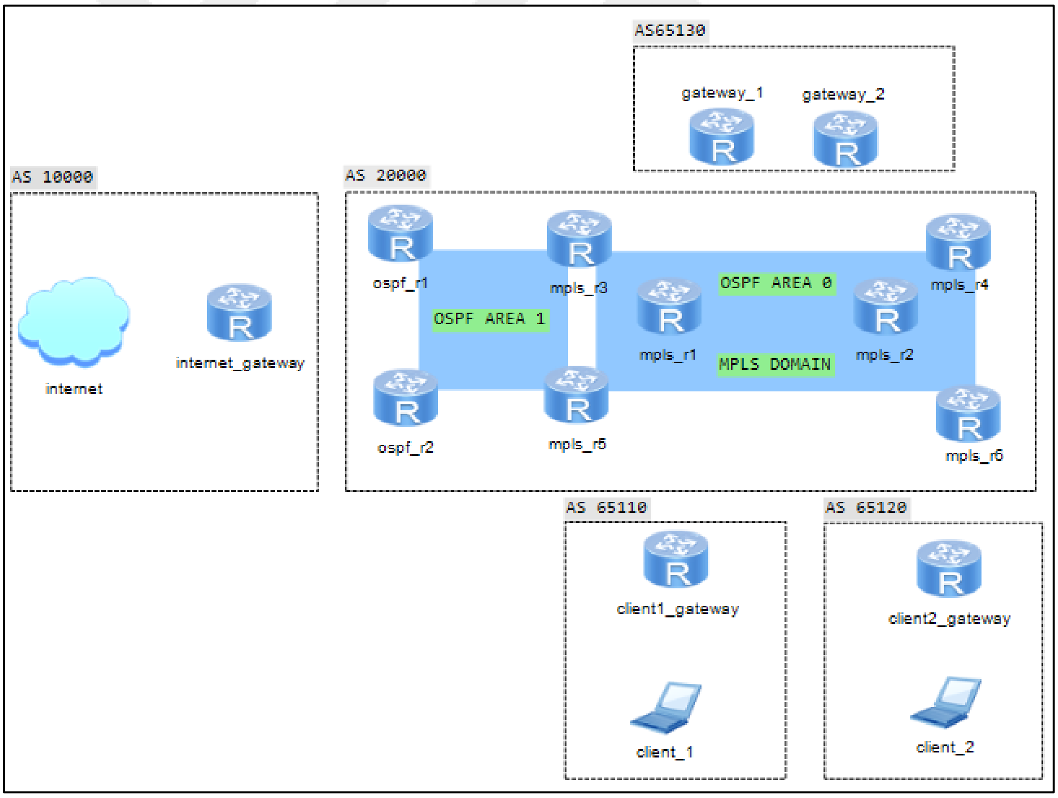


**Figure 29**. Network Topology

According to the topology above in Figure 29, there are two client sites, two gateways, a service provider, and the Internet. AS65110 and AS65120 represent the client sites, and AS65130 holds two redundant gateways for the client sites. AS20000 is the Service Provider network that provides BGP and MPLS services to the client sites for the Internet connection. AS10000 represents the Internet.

When an end user in the AS65110 or AS65120 attempts to connect to the Internet, its connection request reaches the gateway router at the same AS first. Then, the request is forwarded to the Service Provider. The Service Provider may have a complex network; however, this situation is unimportant for the client sites. According to the topology, in the Service Provider network, the MPLS routers carry the traffic to the gateways for the client sites. The gateway routers establish an EBGP connection with the edge routers of the Service Provider. Gateway routers in the AS65130 provide NAT and PAT services for end users. If the Gateway_1 router becomes unavailable, the Gateway_2 router takes all the traffic and it provides a NAT and PAT service. After translating an IP address from a Gateway router in the AS65130, it forwards the traffic to AS20000 and the routers of that AS pass the traffic to the AS10000 (Internet).

An example of a redundant BGP scenario is illustrated in the topology. This scenario can be easily implemented and used in real life.

- **IP Diagram:**

The IP addresses for every interface of the routers are illustrated below:

**AS65110 and AS65120:**

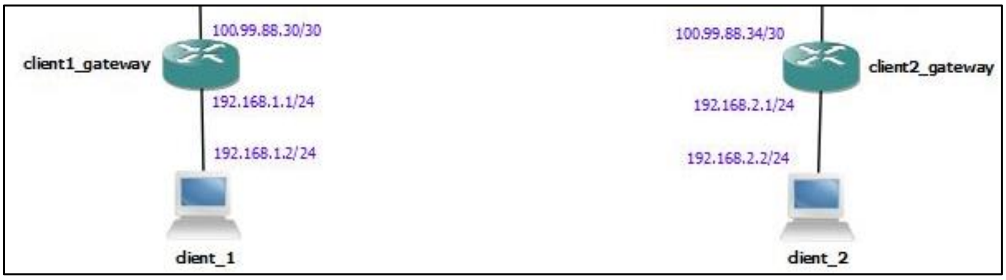


**Figure 30**. IP Diagram of AS65110 and AS65120

Figure 30 shows the IP Diagram of AS65110 and AS65120.

**AS20000 and AS65130:**

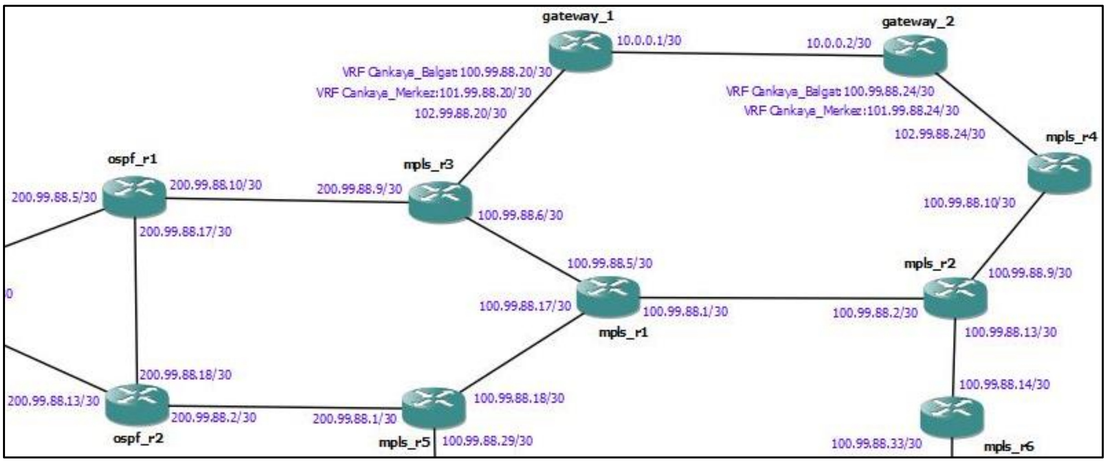


**Figure 31**. IP Diagram of AS20000 and AS65130

Figure 31 shows the IP Diagram of AS20000 and AS65130.

**AS10000:**

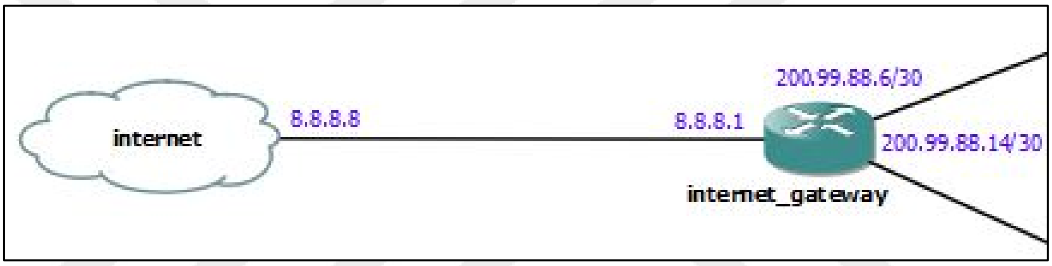
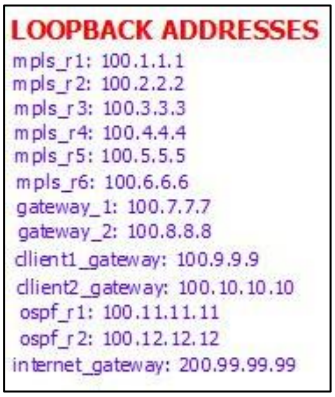


**Figure 32**. IP Diagram of AS10000

Figure 32 shows the IP Diagram of AS10000

- **Loopback Addresses:**



LOOPBACK ADDRESSES
mpls_r1: 100.1.1.1
mpls_r2: 100.2.2.2
mpls_r3: 100.3.3.3
mpls_r4: 100.4.4.4
mpls_r5: 100.5.5.5
mpls_r6: 100.6.6.6
gateway_1: 100.7.7.7
gateway_2: 100.8.8.8
dlient1_gateway: 100.9.9.9
dlient2_gateway: 100.10.10.10
ospf_r1: 100.11.11.11
ospf_r2: 100.12.12.12
internet_gateway: 200.99.99.99

**Figure 33**. Loopback Addresses

Figure 33 shows the Loopback Addresses of the routers.

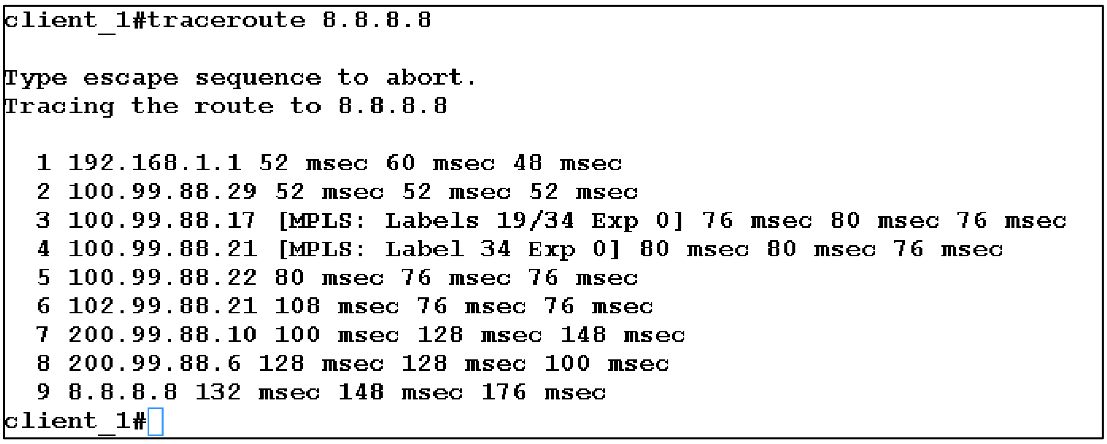
# 4. RESULTS

## 4.1. Testing the Environment

There are two clients in the topology, one of which is in AS65110 and the order in AS65120. To ensure that BGP and MPLS configurations are successful, ping and traceroute tests are performed on client_1 and client_2. If the ping and traceroute reach to the Internet (8.8.8.8), the connection test will be successful. Moreover, redundancy will be tested. There are two gateways, namely gateway_1 and gateway_2. Gateway_1 carries out NAT operations to take clients to the Internet. Gateway_1 router's BGP interfaces are shut down to perform the redundancy test. When gateway_1 is shut down, there will be a BGP convergence time, and then the gateway_2 router will take control. Traceroute outputs will show the path of the packets.

## 4.2. Connection Tests

**From client_1:**



```
client_1#traceroute 8.8.8.8

Type escape sequence to abort.
Tracing the route to 8.8.8.8

  1 192.168.1.1 52 msec 60 msec 48 msec
  2 100.99.88.29 52 msec 52 msec 52 msec
  3 100.99.88.17 [MPLS: Labels 19/34 Exp 0] 76 msec 80 msec 76 msec
  4 100.99.88.21 [MPLS: Label 34 Exp 0] 80 msec 80 msec 76 msec
  5 100.99.88.22 80 msec 76 msec 76 msec
  6 102.99.88.21 108 msec 76 msec 76 msec
  7 200.99.88.10 100 msec 128 msec 148 msec
  8 200.99.88.6 128 msec 128 msec 100 msec
  9 8.8.8.8 132 msec 148 msec 176 msec
client_1#
```

**Figure 34**. Reachability test to the Internet from Client_1

In Figure 34, Client_1 has successfully reached the Internet via gateway_1, and as illustrated in Figure 35, the packets followed that route:

1: client1_gateway

2: mpls_r5

3: mpls_r1

4: mpls_r3

5: gateway_1

6: mpls_r3

7: ospf_r1

8: internet_gateway

9: internet

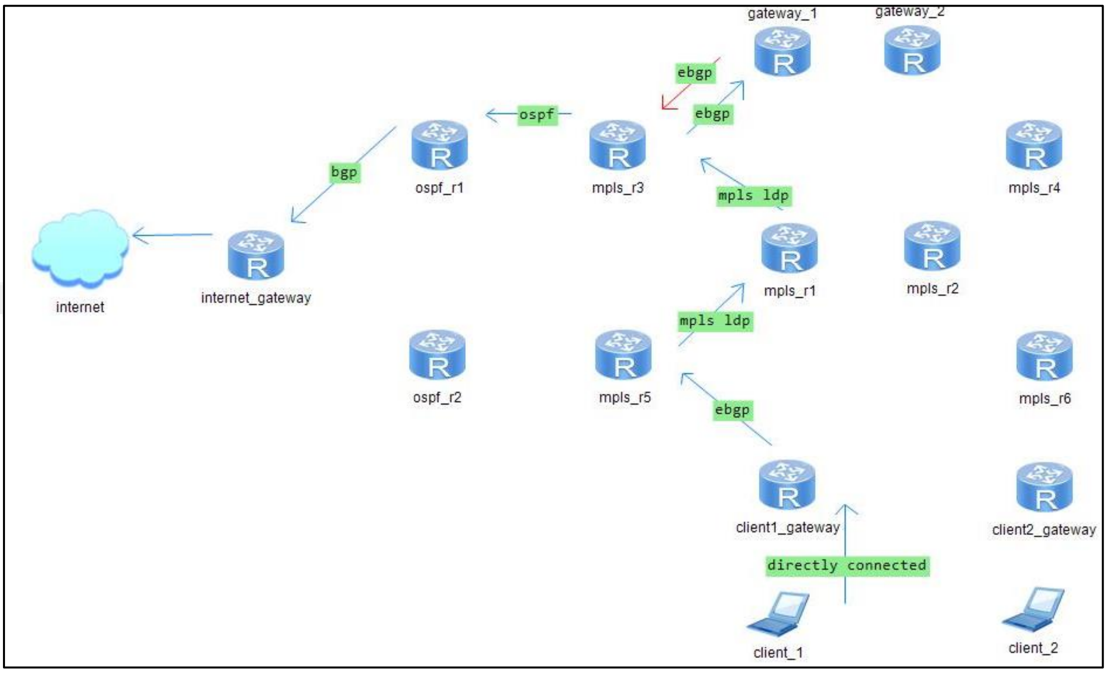
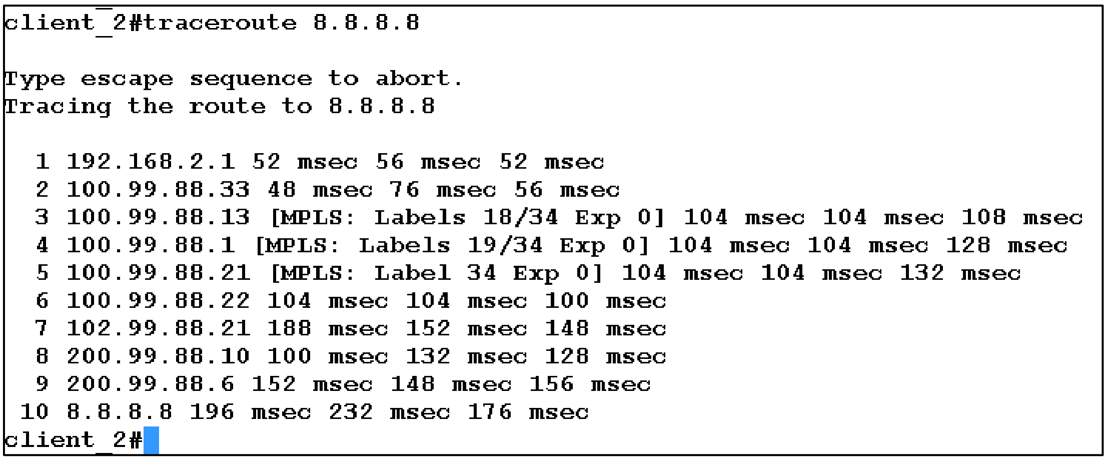


**Figure 35**. Path of the packets from client_1 to the internet via gateway_1

**From client_2:**



```
client_2#traceroute 8.8.8.8

Type escape sequence to abort.
Tracing the route to 8.8.8.8

  1 192.168.2.1 52 msec 56 msec 52 msec
  2 100.99.88.33 48 msec 76 msec 56 msec
  3 100.99.88.13 [MPLS: Labels 18/34 Exp 0] 104 msec 104 msec 108 msec
  4 100.99.88.1 [MPLS: Labels 19/34 Exp 0] 104 msec 104 msec 128 msec
  5 100.99.88.21 [MPLS: Label 34 Exp 0] 104 msec 104 msec 132 msec
  6 100.99.88.22 104 msec 104 msec 100 msec
  7 102.99.88.21 188 msec 152 msec 148 msec
  8 200.99.88.10 100 msec 132 msec 128 msec
  9 200.99.88.6 152 msec 148 msec 156 msec
 10 8.8.8.8 196 msec 232 msec 176 msec
client_2#
```

**Figure 36**. Reachability test to the Internet from Client_2

In Figure 36, Client_2 has successfully reached to the internet via gateway_1. The packets have followed that route and it is illustrated in Figure 37:

1: client2_gateway

2: mpls_r6

3: mpls_r2

4: mpls_r1

5: mpls_r3

6: gateway_1

7: mpls_r3

8: odpf_r1

9: internet_gateway

10: intenet

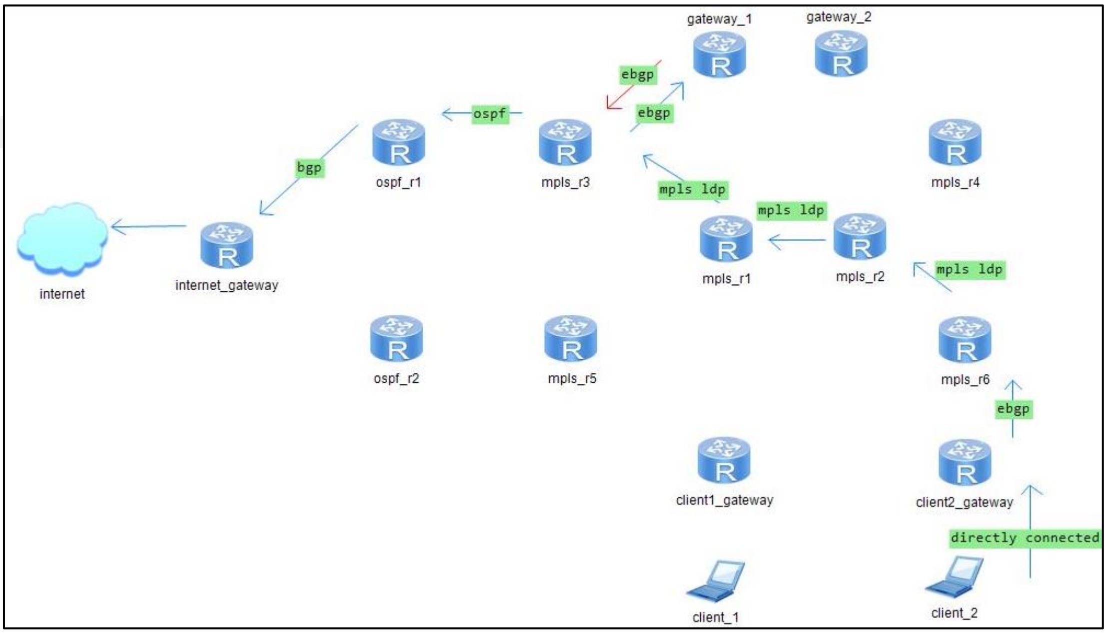
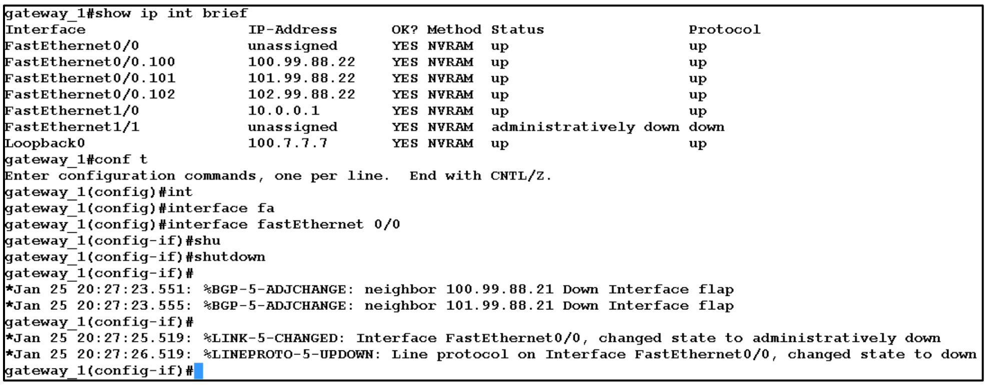


**Figure 37**. Path of the packets from client_2 to the internet via gateway_1

## 4.3. Redundancy Tests

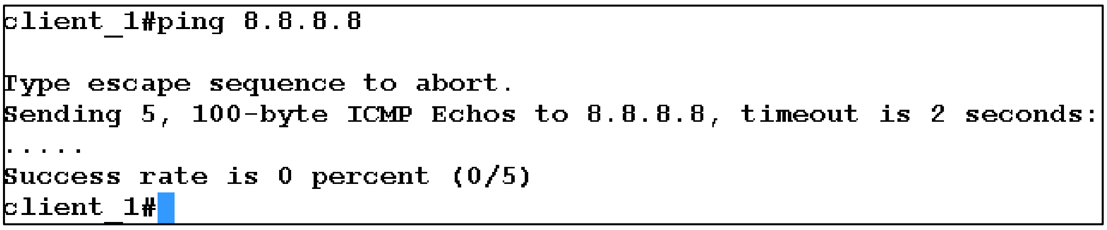
As the first step, gateway_1 is shut down (Figure 38):



```
gateway_1#show ip int brief
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            unassigned      YES NVRAM  up                    up
FastEthernet0/0.100        100.99.88.22    YES NVRAM  up                    up
FastEthernet0/0.101        101.99.88.22    YES NVRAM  up                    up
FastEthernet0/0.102        102.99.88.22    YES NVRAM  up                    up
FastEthernet1/0            10.0.0.1        YES NVRAM  up                    up
FastEthernet1/1            unassigned      YES NVRAM  administratively down down
Loopback0                  100.7.7.7       YES NVRAM  up                    up
gateway_1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
gateway_1(config)#int
gateway_1(config)#interface fa
gateway_1(config)#interface fastEthernet 0/0
gateway_1(config-if)#shu
gateway_1(config-if)#shutdown
gateway_1(config-if)#
*Jan 25 20:27:23.551: %BGP-5-ADJCHANGE: neighbor 100.99.88.21 Down Interface flap
*Jan 25 20:27:23.555: %BGP-5-ADJCHANGE: neighbor 101.99.88.21 Down Interface flap
gateway_1(config-if)#
*Jan 25 20:27:25.519: %LINK-5-CHANGED: Interface FastEthernet0/0, changed state to administratively down
*Jan 25 20:27:26.519: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to down
gateway_1(config-if)#
```

**Figure 38**. Shutting down the Gateway_1 Router

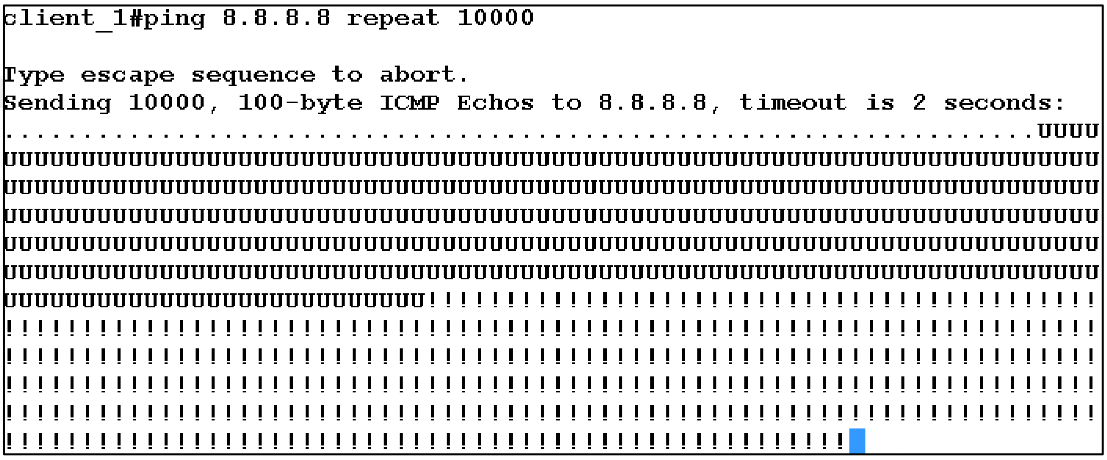
After that, a ping test is initiated by client_1:



```
client_1#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 0 percent (0/5)
client_1#
```

**Figure 39.** Connection lose

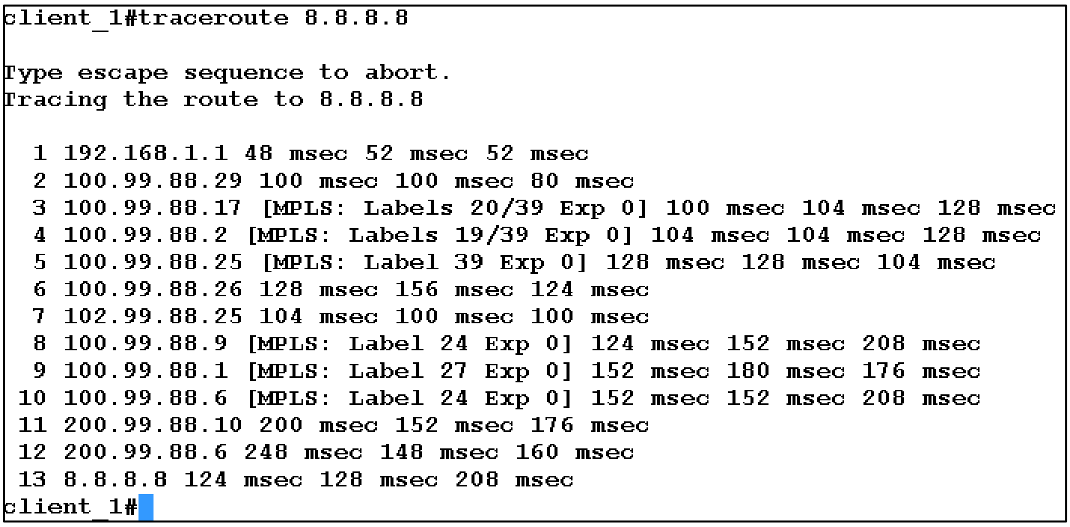
Figure 39 above shows that the connection is lost. Consequently, ping packets cannot reach the Internet. The expectation here is that after the BGP convergence time gateway_2 will take control.



```
client_1#ping 8.8.8.8 repeat 10000

Type escape sequence to abort.
Sending 10000, 100-byte ICMP Echos to 8.8.8.8, timeout is 2 seconds:
..................................................................UUUU
UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
UUUUUUUUUUUUUUUUUUUUUUUUUUU!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

**Figure 40**. Connection turns back

Figure 40 shows how the packets can now reach the Internet. At the beginning of the ping, connectivity loss can be seen as gateway_1 is unavailable, and there is convergence time for BGP. After that time, connectivity returns, because gateway_2 has taken the control. The figures below show these processes:



```
client_1#traceroute 8.8.8.8

Type escape sequence to abort.
Tracing the route to 8.8.8.8

  1 192.168.1.1 48 msec 52 msec 52 msec
  2 100.99.88.29 100 msec 100 msec 80 msec
  3 100.99.88.17 [MPLS: Labels 20/39 Exp 0] 100 msec 104 msec 128 msec
  4 100.99.88.2 [MPLS: Labels 19/39 Exp 0] 104 msec 104 msec 128 msec
  5 100.99.88.25 [MPLS: Label 39 Exp 0] 128 msec 128 msec 104 msec
  6 100.99.88.26 128 msec 156 msec 124 msec
  7 102.99.88.25 104 msec 100 msec 100 msec
  8 100.99.88.9 [MPLS: Label 24 Exp 0] 124 msec 152 msec 208 msec
  9 100.99.88.1 [MPLS: Label 27 Exp 0] 152 msec 180 msec 176 msec
 10 100.99.88.6 [MPLS: Label 24 Exp 0] 152 msec 152 msec 208 msec
 11 200.99.88.10 200 msec 152 msec 176 msec
 12 200.99.88.6 248 msec 148 msec 160 msec
 13 8.8.8.8 124 msec 128 msec 208 msec
client_1#
```

**Figure 41**. Reachability test to the Internet from Client_1 via Gateway_2

In Figure 41, Client_1 has successfully reached to the internet via gateway_2. The packets have followed that route and it is illustrated in Figure 42:

1: client1_gateway

2: mpls_r5

3: mpls_r1

4: mpls_r2

5: mpls_r4

6: gateway_2

7: mpls_r4

8: mpls_r2

9: mpls_r1

10: mpls_r3

11: ospf_r1

12: internet_gateway

13: internet

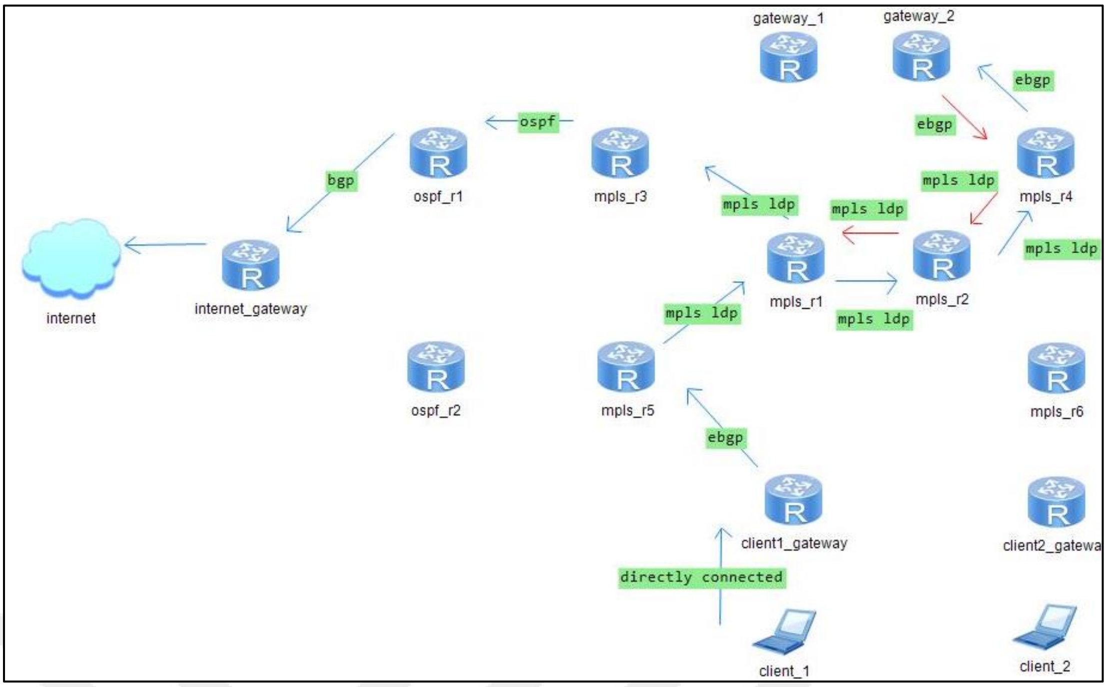
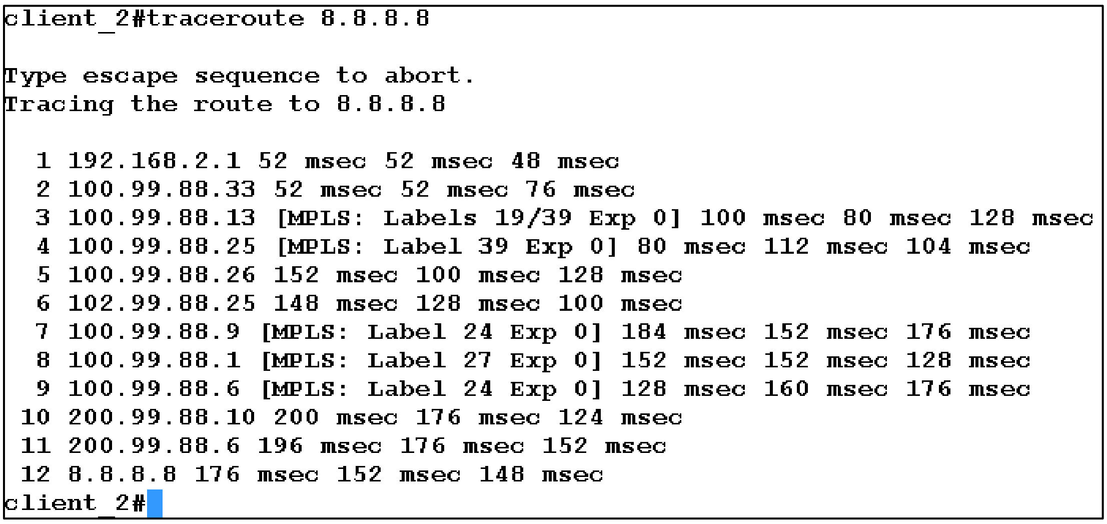


**Figure 42**. Path of the packets from client_1 to the internet via gateway_2

**From client_2:**



```
client_2#traceroute 8.8.8.8

Type escape sequence to abort.
Tracing the route to 8.8.8.8

  1 192.168.2.1 52 msec 52 msec 48 msec
  2 100.99.88.33 52 msec 52 msec 76 msec
  3 100.99.88.13 [MPLS: Labels 19/39 Exp 0] 100 msec 80 msec 128 msec
  4 100.99.88.25 [MPLS: Label 39 Exp 0] 80 msec 112 msec 104 msec
  5 100.99.88.26 152 msec 100 msec 128 msec
  6 102.99.88.25 148 msec 128 msec 100 msec
  7 100.99.88.9 [MPLS: Label 24 Exp 0] 184 msec 152 msec 176 msec
  8 100.99.88.1 [MPLS: Label 27 Exp 0] 152 msec 152 msec 128 msec
  9 100.99.88.6 [MPLS: Label 24 Exp 0] 128 msec 160 msec 176 msec
 10 200.99.88.10 200 msec 176 msec 124 msec
 11 200.99.88.6 196 msec 176 msec 152 msec
 12 8.8.8.8 176 msec 152 msec 148 msec
client_2#
```

**Figure 43**. Reachability test to the Internet from Client_2 via Gateway_2

In Figure 43, Client_2 has successfully reached to the internet via gateway_2. The packets have followed that route and it is illustrated in Figure 44:

1: client2_gateway

2: mpls_r6

3: mpls_r2

4: mpls_r4

5: gateway_2

6: mpls_r4

7: mpls_r2

8: mpls_r1

9: mpls_r3

10: ospf_r1

11: internet_gateway

12: internet

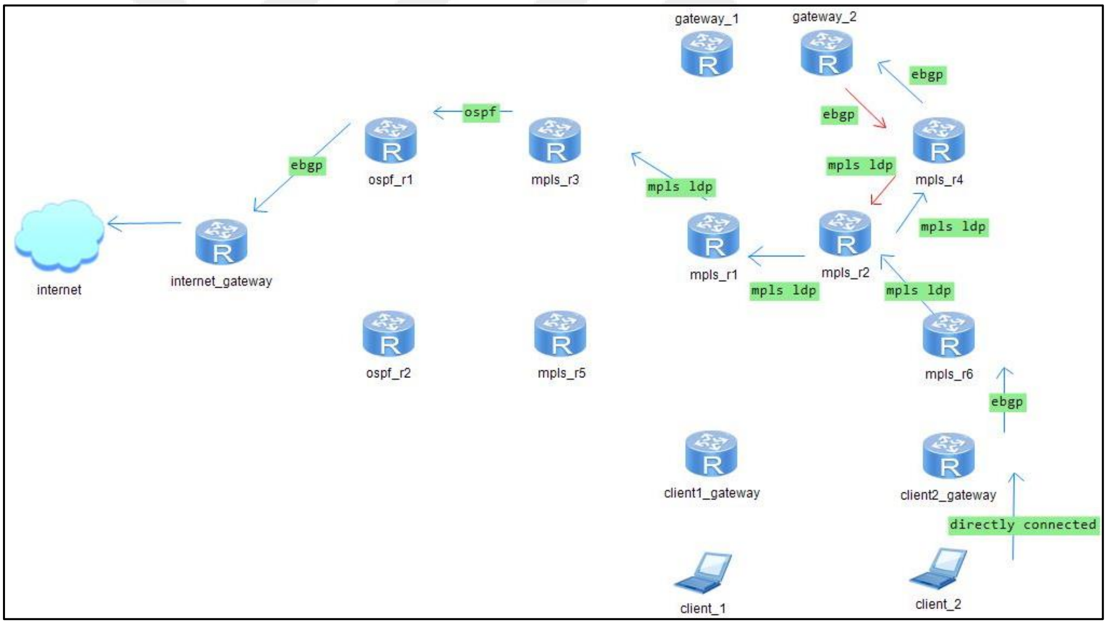
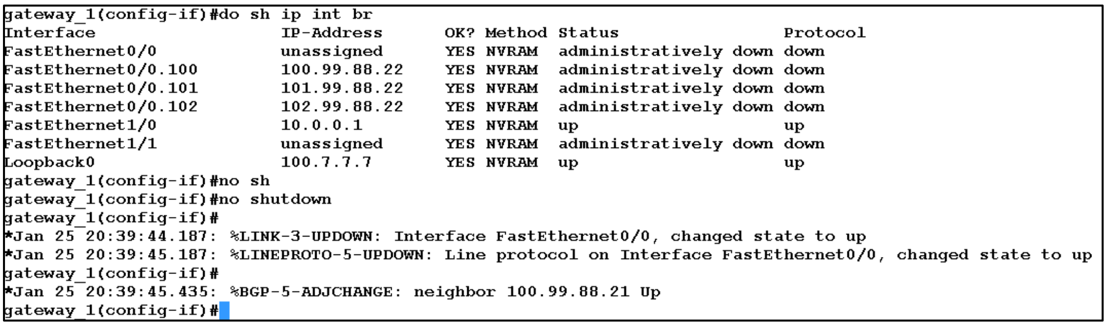


**Figure 44**. Path of the packets from client_2 to the internet via gateway_2

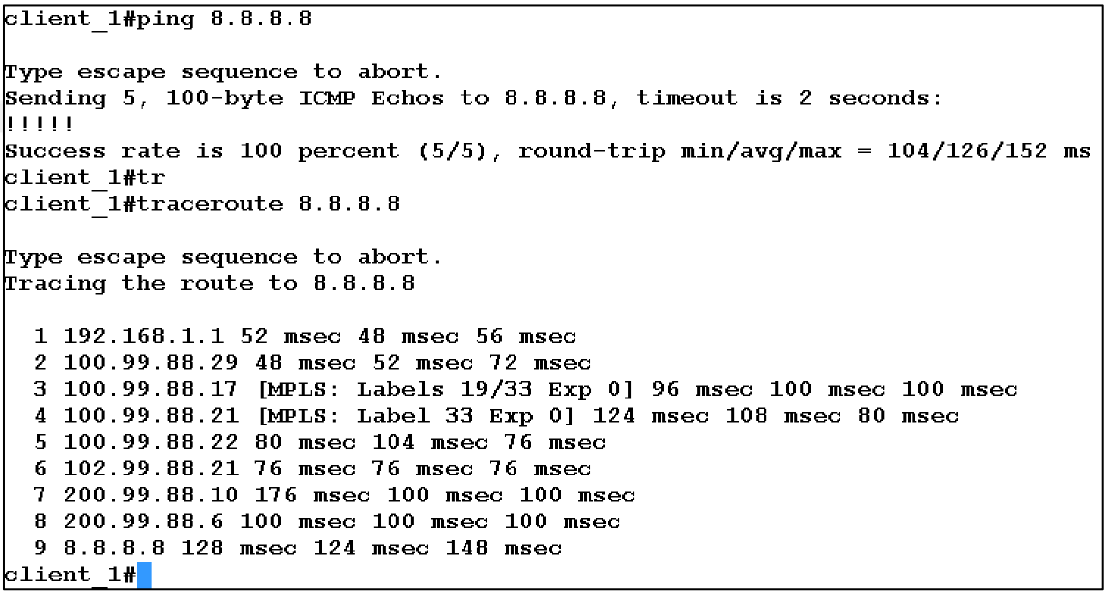
The figures above show that the redundancy tests are successful. When gateway_1 becomes available, it will take control to itself again. The ping and traceroute outputs below prove so in the following figures:



```
gateway_1(config-if)#do sh ip int br
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            unassigned      YES NVRAM  administratively down down
FastEthernet0/0.100        100.99.88.22    YES NVRAM  administratively down down
FastEthernet0/0.101        101.99.88.22    YES NVRAM  administratively down down
FastEthernet0/0.102        102.99.88.22    YES NVRAM  administratively down down
FastEthernet1/0            10.0.0.1        YES NVRAM  up                    up
FastEthernet1/1            unassigned      YES NVRAM  administratively down down
Loopback0                  100.7.7.7       YES NVRAM  up                    up
gateway_1(config-if)#no sh
gateway_1(config-if)#no shutdown
gateway_1(config-if)#
*Jan 25 20:39:44.187: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Jan 25 20:39:45.187: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
gateway_1(config-if)#
*Jan 25 20:39:45.435: %BGP-5-ADJCHANGE: neighbor 100.99.88.21 Up
gateway_1(config-if)#
```

**Figure 45**. Making the gateway_1 available

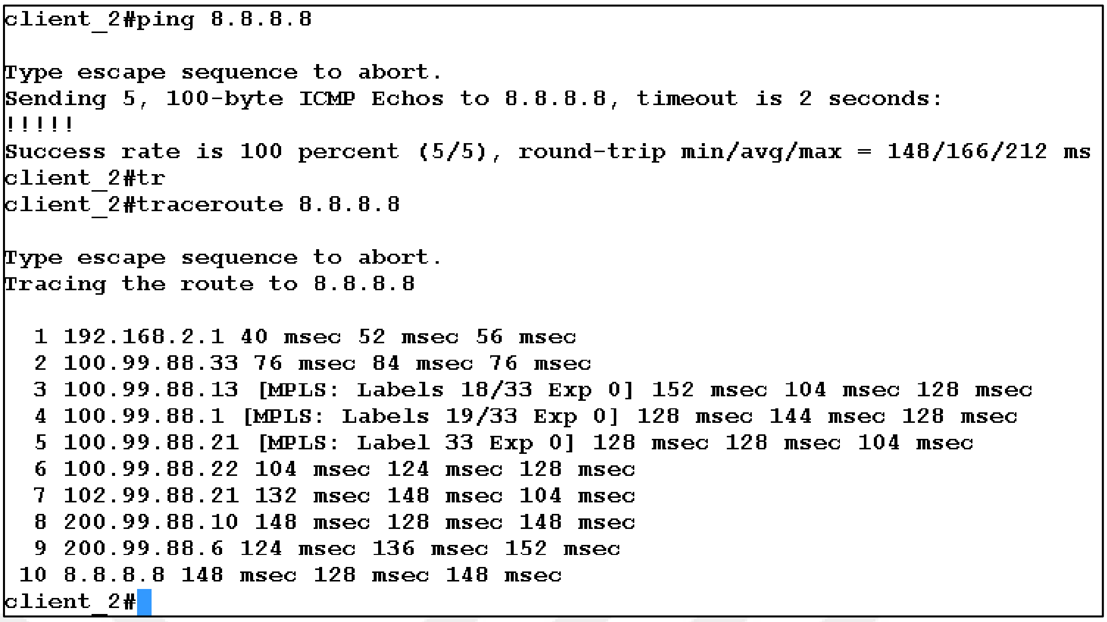
In the Figure 45, the Gateway_1 router is activated again.



```
client_1#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 = 104/126/152 ms
client_1#tr
client_1#traceroute 8.8.8.8

Type escape sequence to abort.
Tracing the route to 8.8.8.8

  1 192.168.1.1 52 msec 48 msec 56 msec
  2 100.99.88.29 48 msec 52 msec 72 msec
  3 100.99.88.17 [MPLS: Labels 19/33 Exp 0] 96 msec 100 msec 100 msec
  4 100.99.88.21 [MPLS: Label 33 Exp 0] 124 msec 108 msec 80 msec
  5 100.99.88.22 80 msec 104 msec 76 msec
  6 102.99.88.21 76 msec 76 msec 76 msec
  7 200.99.88.10 176 msec 100 msec 100 msec
  8 200.99.88.6 100 msec 100 msec 100 msec
  9 8.8.8.8 128 msec 124 msec 148 msec
client_1#
```

**Figure 46**. Reachability control from client_1 to the internet via gateway_1

Figure 46 shows that the Gateway_1 router has regained control.



```
client_2#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 = 148/166/212 ms
client_2#tr
client_2#traceroute 8.8.8.8

Type escape sequence to abort.
Tracing the route to 8.8.8.8

  1 192.168.2.1 40 msec 52 msec 56 msec
  2 100.99.88.33 76 msec 84 msec 76 msec
  3 100.99.88.13 [MPLS: Labels 18/33 Exp 0] 152 msec 104 msec 128 msec
  4 100.99.88.1 [MPLS: Labels 19/33 Exp 0] 128 msec 144 msec 128 msec
  5 100.99.88.21 [MPLS: Label 33 Exp 0] 128 msec 128 msec 104 msec
  6 100.99.88.22 104 msec 124 msec 128 msec
  7 102.99.88.21 132 msec 148 msec 104 msec
  8 200.99.88.10 148 msec 128 msec 148 msec
  9 200.99.88.6 124 msec 136 msec 152 msec
 10 8.8.8.8 148 msec 128 msec 148 msec
client_2#
```

**Figure 47**. Reachability control from client_2 to the Internet via gateway_1

Figure 47 shows that the Gateway_1 router has regained control.

To sum up, tests have proved that a reliable BGP network topology can be provided with BGP and MPLS. With the exception of edge routers, MPLS is a useful protocol to eliminate routing lookup in the MPLS domain by providing smaller routing tables. Moreover, VRF provides clear end separate routing tables and OSPF provides a routing process for IGP networks. In that example of topology, the most important and useful BGP features were used and all of them can be used in real-life modern networks.

# 5. CONCLUSION AND FUTURE WORK

BGP is the routing protocol that interconnects ASs via the Internet and how alternative routing protocols are insufficient when carrying information via the Internet. BGP must be used for enterprise networks while connecting ISPs. In this thesis routing protocol concepts are introduced and the BGP protocol is deeply discussed. BGP attributes, MPLS, VRF are described. An implementation of a BGP network is built with GNS3 Network Simulator for real-life scenarios.

This study has answered the research questions and it provides guidance for these questions in the implementation section. BGP is used to interconnect different ASs via the Internet; alternative routing protocols cannot be used instead of BGP. The implementation shows how a BGP computer network should be designed and how the routers should be configured. In addition, the study has showed that route implementation techniques can be used to create a redundant BGP computer network. The following Table 5 lists the Research Questions and their answers briefly:

**Table 5**. Research Questions and Answers

| Research Question | Result |
|---|---|
| Why is the BGP protocol used in computer networks? | BGP is used to interconnect different ASs via Internet. |
| Can alternative routing protocols be used instead of BGP? | Alternative routing protocols are not capable to connect different ASs. Because; they are insufficient, due to enormous routing updates from Internet. |
| How can a computer network be designed with BGP? | The implementation shows that how should be a BGP computer network design. Redundancy is very important while designing a network. |
| How are the routers configured for BGP? | BGP configuration has given and exampled in Testing and Appendices sections. |
| Can route manipulation be used to provide redundancy in BGP configurations? | Yes, route manipulation techniques can be used for creating a redundant BGP computer network. |

In the implementation, EBGP connections are created between different ASs, such as ISPs and customer networks. BGP is configured with MPLS and a redundant, reliable network topology is illustrated. For the most part, redundancy is ignored in real-life scenarios due to economic reasons, configuration and maintenance complexities. This study showed that a redundant network is vital as traffic in a network should not be interrupted in large enterprises. To illustrate, a national bank manages billions of dollars and its computer network performs this operation. When a problem occurs in the network, the Bank may easily lose large amounts of money and many people will be victims of this. Consequently, redundancy in a computer network becomes extremely important in order to avoid such potentially devastating losses.

Redundancy is provided with route manipulation. The “metric” attribute of BGP is used for redundancy. While every router is working, packets select the path to the Internet via gateway_1. However, when gateway_1 is down, the packets select the path to the Internet via gateway_2. Narasimhan and Latchman [18] used route manipulation techniques in their study titled *“Introduction to The Border Gateway Protocol – Case Study using GNS3”*. Likewise, Zidan and Hamarsheh [19] used route manipulation techniques in their study titled “*Implementation of Border Gateway Protocol (BGP) Attributes*”. Both works illustrated that route manipulation is successful for forcing the routers to certain path selection. This thesis also shows that, route manipulation with the BGP “metric” attribute proved to be successful. Additionally, Narasimhan and Latchman discussed Man in the Middle Attacks for BGP networks in their study. A password mechanism is illustrated in this thesis to prevent undesirable BGP connections for IBGP sessions. For future work, a multi-homed BGP scenario can be added to a redundant BGP network.

## REFERENCES

[1] Cisco Systems Inc., *IP Routing Fundamentals*. USA: Cisco Press, 1999.

[2] Heather Osterloh, *TCP/IP Primer Plus*. Indianapolis: SAMS, 2002.

[3] Cisco Systems Inc., *Internetworking Technologies Handbook*. Fourth Ed. USA: Cisco Press, 2003.

[4] Cisco Systems Inc., *Internetworking Technology Overview*. USA: Cisco Press, 1999.

[5] Jeff Doyle, *CCIE Professional Development: Routing TCP/IP*, vol 1, USA: Cisco Press, 1998.

[6] Jeff Doyle, *CCIE Professional Development: Routing TCP/IP*, vol 2, USA: Cisco Press, 1998.

[7] Luc De Ghein, *MPLS Fundamentals*. USA: Cisco Press, 2007.

[8] Cisco Press Advanced MPLS Design and Implementation, 2002.

[9] Cisco Systems Inc., *Implementing Cisco MPLS*, v2.1, USA: Cisco Press, 2004.

[10] V. Kotronis et al., “Evaluating the Effect of Centralization on Routing Convergence on a Hybrid BGP-SDN Emulation Framework”, 2014.

[11] D. Zhang et al., “Modeling and analyzing the convergence property of the BGP routing protocol in SPIN”, 2014.

[12] J. Yun and J. Song, “Enhancing Secure AS Path BGP (SAPBGP) for Efficient Comparison”, 2015.

[13] Z. Duan et al., “Limiting Path Exploration in BGP”, 2005.

[14] D. Pei et al., “A Study of BGP Path Vector Route Looping Behavior”, 2004.

[15] Divan G. Raimagia et al., “To Make Trust Relationship Between BGP Speakers with Help of Secure Private Key”, 2012.

[16] *Forwarding and Control Element Separation*, RFC 3746, 2004.

[17] W. Hamzeh and A. Hafid, “A Distributed Parallel Approach for BGP Routing Table Partitioning in Next Generation Routers”, 2010.

[18] S. Narasimhan and H. Latchman, “Introduction to The Border Gateway Protocol – Case Study using GNS3”, Dept. Compt. Eng., Univ. Florida, Gainesville, 2011.

[19] N. Zidan and M. Hamarsheh, “Implementation of Border Gateway Protocol (BGP) Attributes”, 2016.

[20] Cisco Systems Inc., *Configuring BGP on Cisco Routers (BGP)*, v3.2, USA: Cisco Press, 2005.

[21] *GNS3 Software*, Available: https://www.gns3.com

[22] Cisco Press, S. Halabi and D. McPherson, Internet Routing Architectures, Second Edition, 2000.

[23] Cisco Systems Inc., “VPN process without VRF”, USA: Cisco Press, 2006.

[24] Cisco Systems Inc., *Configuring BGP on Cisco Routers*, v3.2, vol 1, USA: Cisco Press, 2005.

[25] Cisco Systems Inc., *Configuring BGP on Cisco Routers*, v3.2, vol 2, USA: Cisco Press, 2005.

# APPENDICES

## APPENDIX A

### Configuration of the Routers

The configurations of the routers are below:

**Client1_gateway:**

```
Building configuration...

Current configuration : 1354 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname client1_gateway
!
boot-start-marker
boot-end-marker
!
no aaa new-model
ip subnet-zero
no ip icmp rate-limit unreachable
!
ip tcp synwait-time 5
ip cef
no ip domain lookup
!
ip ips po max-events 100
no ftp-server write-enable
!
no crypto isakmp ccm
!
interface Loopback0
 ip address 100.9.9.9 255.255.255.255
!
interface FastEthernet0/0
 ip address 100.99.88.30 255.255.255.252
 no ip redirects
 no ip proxy-arp
 duplex full
 speed auto
 no shutdown
!
interface FastEthernet0/1
 ip address 192.168.1.1 255.255.255.0
 duplex full
 speed auto
 no shutdown
!
router ospf 1
router-id 100.9.9.9 // IDENTIFIES THE ROUTER IN THE OSPF
DOMAIN
network 192.0.0.0 0.255.255.255 area 0 // INCLUDES 192.0.0.0
NETWORK FOR OSPF AREA 0
default-information originate // ADVERTISES THE DEFAULT ROUTE
INTO OSPF AREA 1
```

```
!
router bgp 65110
 no synchronization // TO PREVENT ADVERTISING UNNECESSARY
ROUTES INTO IBGP
 bgp router-id 100.9.9.9  // IDENTIFIES THE ROUTER IN THE BGP
DOMAIN
 bgp log-neighbor-changes
 redistribute ospf 1 // ADVERTISES THE OSPF ROUTES INTO BGP
 neighbor 100.99.88.29 remote-as 20000 // TO CREATE BGP
NEIGHBORSHIP
 no auto-summary
!
ip classless
!
no ip http server
no ip http secure-server
!
no cdp log mismatch duplex
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
!
End
```

**Client2_gateway:**

```
Building configuration...

Current configuration : 1364 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname client2_gateway
!
boot-start-marker
boot-end-marker
!
no aaa new-model
ip subnet-zero
no ip icmp rate-limit unreachable
!
ip tcp synwait-time 5
ip cef
```

```
no ip domain lookup
!
ip ips po max-events 100
no ftp-server write-enable
!
no crypto isakmp ccm
!
interface Loopback0
 ip address 100.10.10.10 255.255.255.255
!
interface FastEthernet0/0
 ip address 100.99.88.34 255.255.255.252
 no ip redirects
 no ip proxy-arp
 duplex full
 speed auto
 no shutdown
!
interface FastEthernet0/1
 ip address 192.168.2.1 255.255.255.0
 duplex full
 speed auto
 no shutdown
!
router ospf 1
router-id 100.10.10.10 // IDENTIFIES THE ROUTER IN THE OSPF
DOMAIN
network 192.0.0.0 0.255.255.255 area 0 // INCLUDES 192.0.0.0
NETWORK FOR OSPF AREA 0
default-information originate // ADVERTISES THE DEFAULT ROUTE
INTO OSPF AREA 1
!
router bgp 65120
 no synchronization // TO PREVENT ADVERTISING UNNECESSARY
ROUTES INTO IBGP
 bgp router-id 100.10.10.10 // IDENTIFIES THE ROUTER IN THE
BGP DOMAIN
 bgp log-neighbor-changes
 redistribute ospf 1 // ADVERTISES THE OSPF ROUTES INTO BGP
 neighbor 100.99.88.33 remote-as 20000 // TO CREATE BGP
NEIGHBORSHIP
 no auto-summary
!
ip classless
!
no ip http server
no ip http secure-server
!
no cdp log mismatch duplex
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
```

```
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
!
end
```

**mpls_r1:**

```
Building configuration...

Current configuration : 1296 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname mpls_r1
!
boot-start-marker
boot-end-marker
!
no aaa new-model
ip subnet-zero
no ip icmp rate-limit unreachable
!
ip tcp synwait-time 5
ip cef
no ip domain lookup
!
ip ips po max-events 100
mpls label protocol ldp //CHOOSES MPLS LABEL PROTOCOL: LDP OR
TDP (TDP IS CISCO PROTOCOL)
tag-switching tdp router-id Loopback0 // TO BUILD LDP
NEIGHBORSHIP VIA Loopback0 INTERFACE (IN THE NEWER IOS'es
"mpls ldp router-id loopback0")
no ftp-server write-enable
!
no crypto isakmp ccm
!
interface Loopback0
 ip address 100.1.1.1 255.255.255.0
!
interface FastEthernet0/0
 ip address 100.99.88.1 255.255.255.252
 duplex full
 tag-switching ip // TO INCLUDE THIS INTERFACE IN THE MPLS
DOMAIN
 no shutdown
!
interface FastEthernet1/0
 ip address 100.99.88.5 255.255.255.252
 duplex full
 speed auto
 tag-switching ip // TO INCLUDE THIS INTERFACE IN THE MPLS
DOMAIN
 no shutdown
!
```

```
interface FastEthernet1/1
 ip address 100.99.88.17 255.255.255.252
 duplex full
 speed auto
 tag-switching ip // TO INCLUDE THIS INTERFACE IN THE MPLS
DOMAIN
 no shutdown
!
router ospf 1
 router-id 100.1.1.1 // IDENTIFIES THE ROUTER IN THE OSPF
DOMAIN
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0 // INCLUDES ALL THE
SUBNETS FOR OSPF ROUTING
!
ip classless
!
no ip http server
no ip http secure-server
!
no cdp log mismatch duplex
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
!
end
```

**mpls_r2:**

```
Building configuration...

Current configuration : 1298 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname mpls_r2
!
boot-start-marker
boot-end-marker
!
no aaa new-model
ip subnet-zero
no ip icmp rate-limit unreachable
```

```
!
ip tcp synwait-time 5
ip cef
no ip domain lookup
!
ip ips po max-events 100
mpls label protocol ldp //CHOOSES MPLS LABEL PROTOCOL: LDP OR
TDP (TDP IS CISCO PROTOCOL)
tag-switching tdp router-id Loopback0 // TO BUILD LDP
NEIGHBORSHIP VIA Loopback0 INTERFACE (IN THE NEWER IOS'es
"mpls ldp router-id loopback0")
no ftp-server write-enable
!
no crypto isakmp ccm
!
interface Loopback0
 ip address 100.2.2.2 255.255.255.255
!
interface FastEthernet0/0
 ip address 100.99.88.2 255.255.255.252
 duplex full
 tag-switching ip // TO INCLUDE THIS INTERFACE IN THE MPLS
DOMAIN
 no shutdown
!
interface FastEthernet1/0
 ip address 100.99.88.9 255.255.255.252
 duplex full
 speed auto
 tag-switching ip // TO INCLUDE THIS INTERFACE IN THE MPLS
DOMAIN
 no shutdown
!
interface FastEthernet1/1
 ip address 100.99.88.13 255.255.255.252
 duplex full
 speed auto
 tag-switching ip // TO INCLUDE THIS INTERFACE IN THE MPLS
DOMAIN
 no shutdown
!
router ospf 1
 router-id 100.2.2.2 // IDENTIFIES THE ROUTER IN THE OSPF
DOMAIN
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0 // INCLUDES ALL THE
SUBNETS FOR OSPF ROUTING
!
ip classless
!
no ip http server
no ip http secure-server
!
no cdp log mismatch duplex
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
```

```
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
!
End
```

**mpls_r3:**

```
Building configuration...

Current configuration : 3574 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname mpls_r3
!
boot-start-marker
boot-end-marker
!
no aaa new-model
ip subnet-zero
no ip icmp rate-limit unreachable
!
ip tcp synwait-time 5
ip cef
no ip domain lookup
!
ip vrf Cankaya_Merkez // TO CREATE A VRF
 rd 65130:2 // TO SPECIFY THE VRF AND TO MAKE THE ROUTES OF
THAT VRF UNIQUE
 route-target import 65130:130 // TO LEAK THE ROUTES FROM
OTHER VRF's TO THAT VRF
 route-target import 65130:110
 route-target import 65130:120
!
ip vrf Cankaya_Balgat // TO CREATE A VRF
 rd 65130:1 // TO SPECIFY THE VRF AND TO MAKE THE ROUTES OF
THAT VRF UNIQUE
 route-target export 65130:130 // TO LEAK THE ROUTES FROM THAT
VRF TO OTHER VRF's
!
ip ips po max-events 100
mpls label protocol ldp //CHOOSES MPLS LABEL PROTOCOL: LDP OR
TDP (TDP IS CISCO PROTOCOL)
tag-switching tdp router-id Loopback0 // TO BUILD LDP
NEIGHBORSHIP VIA Loopback0 INTERFACE (IN THE NEWER IOS'es
"mpls ldp router-id loopback0")
no ftp-server write-enable
!
no crypto isakmp ccm
```

```
!
interface Loopback0
 ip address 100.3.3.3 255.255.255.255
!
interface FastEthernet0/0
 ip address 100.99.88.6 255.255.255.252
 duplex full
 tag-switching ip // TO INCLUDE THIS INTERFACE IN THE MPLS
DOMAIN
 no shutdown
!
interface FastEthernet1/0
 no ip address
 duplex full
 speed auto
 no shutdown
!
interface FastEthernet1/0.100
 encapsulation dot1Q 100 // DOT1Q OR ISL (ISL IS CISCO
PROTOCOL)
 ip vrf forwarding Cankaya_Balgat // TO INCLUDE THIS INTERFACE
TO RELATED VRF
 ip address 100.99.88.21 255.255.255.252
 no ip redirects
 no ip proxy-arp
!
interface FastEthernet1/0.101
 encapsulation dot1Q 101 // DOT1Q OR ISL (ISL IS CISCO
PROTOCOL)
 ip vrf forwarding Cankaya_Merkez // TO INCLUDE THIS INTERFACE
TO RELATED VRF
 ip address 101.99.88.21 255.255.255.252
!
interface FastEthernet1/0.102
 encapsulation dot1Q 102 // DOT1Q OR ISL (ISL IS CISCO
PROTOCOL)
 ip address 102.99.88.21 255.255.255.252
!
interface FastEthernet1/1
 ip address 200.99.88.9 255.255.255.252
 duplex full
 speed auto
 no shutdown
!
router ospf 1
 router-id 100.3.3.3 // IDENTIFIES THE ROUTER IN THE OSPF
DOMAIN
 log-adjacency-changes
 network 100.0.0.0 0.255.255.255 area 0 // INCLUDES 100.0.0.0
NETWORK FOR OSPF AREA 0
 network 200.0.0.0 0.255.255.255 area 1 // INCLUDES 200.0.0.0
NETWORK FOR OSPF AREA 1
!
router bgp 20000
 bgp router-id 100.3.3.3 // IDENTIFIES THE ROUTER IN THE BGP
DOMAIN
 no bgp default ipv4-unicast // TO STATE AN ADDRESS-FAMILY
SPECIFICALLY
 bgp cluster-id 100.3.3.3 // TO INCLUDE MULTIPLE ROUTE-
REFLECTORS TO THE SAME CLUSTER
 bgp log-neighbor-changes
 neighbor INTERNAL peer-group // TO CREATE NEIGHBORSHIP WITH A
GROUP
 neighbor INTERNAL remote-as 20000 // TO SPECIFY THE AS NUMBER
OF THE ROUTERS IN THE GROUP
 neighbor INTERNAL password cankaya // IF A ROUTER IN THE
GROUP DOES NOT HAVE THE PASSWORD, IT CANNOT MAKE THE TCP
CONNECTION
 neighbor INTERNAL update-source Loopback0 // MAKES THE IGP
NEIGHBORSHIP VIA LOOPBACK0 INTERFACE
 neighbor 100.4.4.4 peer-group INTERNAL  // TO INCLUDE THE
ROUTER IN THE GROUP
 neighbor 100.5.5.5 peer-group INTERNAL
 neighbor 100.6.6.6 peer-group INTERNAL
 neighbor 102.99.88.22 remote-as 65130 // TO CREATE BGP
NEIGHBORSHIP
 neighbor 200.11.11.11 peer-group INTERNAL
 neighbor 200.12.12.12 peer-group INTERNAL
 !
 address-family ipv4 // TO BUILD IPv4 BGP
 neighbor INTERNAL activate // TO ACTIVATE BGP NEIGHBORSHIP
FOR THE ADDRESS-FAMILY
 neighbor INTERNAL route-reflector-client // TO ADVERTISE THE
ROUTES THAT LEARNED FROM IBGP TO OTHER IBGP NEIGHBORS (WITHOUT
THIS ALL THE ROUTERS IN THE IGP MUST BE FULLY MESHED)
 neighbor 100.4.4.4 peer-group INTERNAL // TO INCLUDE THE
ROUTER IN THE GROUP FOR THE ADDRESS-FAMILY
 neighbor 100.5.5.5 peer-group INTERNAL
 neighbor 100.6.6.6 peer-group INTERNAL
 neighbor 102.99.88.22 activate // TO ACTIVATE BGP
NEIGHBORSHIP
 neighbor 200.11.11.11 peer-group INTERNAL
 neighbor 200.12.12.12 peer-group INTERNAL
 no auto-summary // TO ADVERTISE SUBNETS CERTAINLY
 no synchronization // TO PREVENT ADVERTISING UNNECESSARY
ROUTES INTO IBGP
 network 100.3.3.3 mask 255.255.255.255 // TO ADVERTISE THE
SPECIFIED SUBNET
 network 102.99.88.20 mask 255.255.255.252
 exit-address-family
 !
 address-family vpnv4 // TO BUILD VPNv4 BGP
 neighbor INTERNAL activate // TO ACTIVATE BGP NEIGHBORSHIP
FOR THE ADDRESS-FAMILY
 neighbor INTERNAL route-reflector-client // TO ADVERTISE THE
ROUTES THAT LEARNED FROM IBGP TO OTHER IBGP NEIGHBORS (WITHOUT
THIS ALL THE ROUTERS IN THE IGP MUST BE FULLY MESHED)
 neighbor INTERNAL send-community extended // NECESSARY FOR
WORKING VPNv4 ADDRESS-FAMILY (SOME POLICIES CAN BE APPLIED
WITH COMMUNITIES)
 neighbor 100.4.4.4 peer-group INTERNAL // TO INCLUDE THE
ROUTER IN THE GROUP FOR THE ADDRESS-FAMILY
 neighbor 100.5.5.5 peer-group INTERNAL
 neighbor 100.6.6.6 peer-group INTERNAL
 exit-address-family
 !
 address-family ipv4 vrf Cankaya_Balgat // TO CREATE AN
ADDRESS-FAMILY FOR RELATED VRF
 neighbor 100.99.88.22 remote-as 65130 // TO CREATE BGP
NEIGHBORSHIP
 neighbor 100.99.88.22 activate // TO ACTIVATE BGP
NEIGHBORSHIP
```

```
 no auto-summary // TO ADVERTISE SUBNETS CERTAINLY
 no synchronization // TO PREVENT ADVERTISING UNNECESSARY
ROUTES INTO IBGP
 network 100.99.88.20 mask 255.255.255.252 // TO ADVERTISE THE
SPECIFIED SUBNET
 exit-address-family
 !
 address-family ipv4 vrf Cankaya_Merkez // TO CREATE AN
ADDRESS-FAMILY FOR RELATED VRF
 neighbor 101.99.88.22 remote-as 65130 // TO CREATE BGP
NEIGHBORSHIP
 neighbor 101.99.88.22 activate // TO ACTIVATE BGP
NEIGHBORSHIP
 no auto-summary // TO ADVERTISE SUBNETS CERTAINLY
 no synchronization // TO PREVENT ADVERTISING UNNECESSARY
ROUTES INTO IBGP
 network 101.99.88.20 mask 255.255.255.252 // TO ADVERTISE THE
SPECIFIED SUBNET
 exit-address-family
!
ip classless
!
no ip http server
no ip http secure-server
!
no cdp log mismatch duplex
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
!
End
```

**mpls_r4:**

```
Building configuration...

Current configuration : 3421 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname mpls_r4
!
boot-start-marker
```

```
boot-end-marker
!
no aaa new-model
ip subnet-zero
no ip icmp rate-limit unreachable
!
ip tcp synwait-time 5
ip cef
no ip domain lookup
!
ip vrf Cankaya_Merkez // TO CREATE A VRF
 rd 65130:2 // TO SPECIFY THE VRF AND TO MAKE THE ROUTES OF
THAT VRF UNIQUE
 route-target import 65130:110 // TO LEAK THE ROUTES FROM
OTHER VRF's TO THAT VRF
 route-target import 65130:120
 route-target import 65130:130
!
ip vrf Cankaya_Balgat // TO CREATE A VRF
 rd 65130:1 // TO SPECIFY THE VRF AND TO MAKE THE ROUTES OF
THAT VRF UNIQUE
 route-target export 65130:130 // TO LEAK THE ROUTES FROM THAT
VRF TO OTHER VRF's
!
ip ips po max-events 100
mpls label protocol ldp //CHOOSES MPLS LABEL PROTOCOL: LDP OR
TDP (TDP IS CISCO PROTOCOL)
tag-switching tdp router-id Loopback0 // TO BUILD LDP
NEIGHBORSHIP VIA Loopback0 INTERFACE (IN THE NEWER IOS'es
"mpls ldp router-id loopback0")
no ftp-server write-enable
!
no crypto isakmp ccm
!
interface Loopback0
 ip address 100.4.4.4 255.255.255.255
!
interface FastEthernet0/0
 ip address 100.99.88.10 255.255.255.252
 duplex full
 tag-switching ip // TO INCLUDE THIS INTERFACE IN THE MPLS
DOMAIN
 no shutdown
!
interface FastEthernet1/0
 no ip address
 duplex full
 speed auto
 no shutdown
!
interface FastEthernet1/0.100
 encapsulation dot1Q 100 // DOT1Q OR ISL (ISL IS CISCO
PROTOCOL)
 ip vrf forwarding Cankaya_Balgat // TO INCLUDE THIS INTERFACE
TO RELATED VRF
 ip address 100.99.88.25 255.255.255.252
 no ip redirects
 no ip proxy-arp
!
interface FastEthernet1/0.101
```

```
 encapsulation dot1Q 101 // DOT1Q OR ISL (ISL IS CISCO
PROTOCOL)
 ip vrf forwarding Cankaya_Merkez // TO INCLUDE THIS INTERFACE
TO RELATED VRF
 ip address 101.99.88.25 255.255.255.252
!
interface FastEthernet1/0.102
 encapsulation dot1Q 102 // DOT1Q OR ISL (ISL IS CISCO
PROTOCOL)
 ip address 102.99.88.25 255.255.255.252
!
router ospf 1
 router-id 100.4.4.4 // IDENTIFIES THE ROUTER IN THE OSPF
DOMAIN
 log-adjacency-changes
 network 100.0.0.0 0.255.255.255 area 0 // INCLUDES 100.0.0.0
NETWORK FOR OSPF ROUTING
!
router bgp 20000
 bgp router-id 100.4.4.4 // IDENTIFIES THE ROUTER IN THE BGP
DOMAIN
 no bgp default ipv4-unicast // TO STATE AN ADDRESS-FAMILY
SPECIFICALLY
 bgp log-neighbor-changes
 neighbor INTERNAL peer-group // TO CREATE NEIGHBORSHIP WITH A
GROUP
 neighbor INTERNAL remote-as 20000 // TO SPECIFY THE AS NUMBER
OF THE ROUTERS IN THE GROUP
 neighbor INTERNAL password cankaya // IF A ROUTER IN THE
GROUP DOES NOT HAVE THE PASSWORD, IT CANNOT MAKE THE TCP
CONNECTION
 neighbor INTERNAL update-source Loopback0 // MAKES THE IGP
NEIGHBORSHIP VIA LOOPBACK0 INTERFACE
 neighbor 100.3.3.3 peer-group INTERNAL // TO INCLUDE THE
ROUTER IN THE GROUP
 neighbor 100.5.5.5 peer-group INTERNAL
 neighbor 100.6.6.6 peer-group INTERNAL
 neighbor 102.99.88.26 remote-as 65130 // TO CREATE BGP
NEIGHBORSHIP
 !
 address-family ipv4 // TO BUILD IPv4 BGP
 neighbor INTERNAL activate // TO ACTIVATE BGP NEIGHBORSHIP
FOR THE ADDRESS-FAMILY
 neighbor 100.3.3.3 peer-group INTERNAL // TO INCLUDE THE
ROUTER IN THE GROUP FOR THE ADDRESS-FAMILY
 neighbor 100.5.5.5 peer-group INTERNAL
 neighbor 100.6.6.6 peer-group INTERNAL
 neighbor 102.99.88.26 activate // TO ACTIVATE BGP
NEIGHBORSHIP
 no auto-summary // TO ADVERTISE SUBNETS CERTAINLY
 no synchronization // TO PREVENT ADVERTISING UNNECESSARY
ROUTES INTO IBGP
 network 100.3.3.3 mask 255.255.255.255 // TO ADVERTISE THE
SPECIFIED SUBNET
 network 100.4.4.4 mask 255.255.255.255
 network 102.99.88.24 mask 255.255.255.252
 exit-address-family
 !
 address-family vpnv4 // TO BUILD VPNv4 BGP
 neighbor INTERNAL activate  // TO ACTIVATE BGP NEIGHBORSHIP
FOR THE ADDRESS-FAMILY
```

```
 neighbor INTERNAL route-reflector-client // TO ADVERTISE THE
ROUTES THAT LEARNED FROM IBGP TO OTHER IBGP NEIGHBORS (WITHOUT
THIS ALL THE ROUTERS IN THE IGP MUST BE FULLY MESHED)
 neighbor INTERNAL send-community extended // NECESSARY FOR
WORKING VPNv4 ADDRESS-FAMILY (SOME POLICIES CAN BE APPLIED
WITH COMMUNITIES)
 neighbor 100.3.3.3 peer-group INTERNAL // TO INCLUDE THE
ROUTER IN THE GROUP FOR THE ADDRESS-FAMILY
 neighbor 100.5.5.5 peer-group INTERNAL
 neighbor 100.6.6.6 peer-group INTERNAL
 exit-address-family
 !
 address-family ipv4 vrf Cankaya_Balgat // TO CREATE AN
ADDRESS-FAMILY FOR RELATED VRF
 neighbor 100.99.88.26 remote-as 65130 // TO CREATE BGP
NEIGHBORSHIP
 neighbor 100.99.88.26 activate // TO ACTIVATE BGP
NEIGHBORSHIP
 no auto-summary // TO ADVERTISE SUBNETS CERTAINLY
 no synchronization // TO PREVENT ADVERTISING UNNECESSARY
ROUTES INTO IBGP
 network 100.99.88.24 mask 255.255.255.252  // TO ADVERTISE
THE SPECIFIED SUBNET
 exit-address-family
 !
 address-family ipv4 vrf Cankaya_Merkez // TO CREATE AN
ADDRESS-FAMILY FOR RELATED VRF
 neighbor 101.99.88.26 remote-as 65130 // TO CREATE BGP
NEIGHBORSHIP
 neighbor 101.99.88.26 activate // TO ACTIVATE BGP
NEIGHBORSHIP
 no auto-summary  // TO ADVERTISE SUBNETS CERTAINLY
 no synchronization // TO PREVENT ADVERTISING UNNECESSARY
ROUTES INTO IBGP
 network 101.99.88.24 mask 255.255.255.252  // TO ADVERTISE
THE SPECIFIED SUBNET
 exit-address-family
!
ip classless
!
no ip http server
no ip http secure-server
!
no cdp log mismatch duplex
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
```

```
 login
!
end
```

**mpls_r5:**

```
Building configuration...

Current configuration : 2856 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname mpls_r5
!
boot-start-marker
boot-end-marker
!
no aaa new-model
ip subnet-zero
no ip icmp rate-limit unreachable
!
ip tcp synwait-time 5
ip cef
no ip domain lookup
!
ip vrf Cankaya // TO CREATE A VRF
 rd 65130:3 // TO SPECIFY THE VRF AND TO MAKE THE ROUTES OF
THAT VRF UNIQUE
 route-target export 65130:110 // TO LEAK THE ROUTES FROM THAT
VRF TO OTHER VRF's
 route-target import 65130:130 // TO LEAK THE ROUTES FROM
OTHER VRF's TO THAT VRF
!
ip ips po max-events 100
mpls label protocol ldp //CHOOSES MPLS LABEL PROTOCOL: LDP OR
TDP (TDP IS CISCO PROTOCOL)
tag-switching tdp router-id Loopback0 // TO BUILD LDP
NEIGHBORSHIP VIA Loopback0 INTERFACE (IN THE NEWER IOS'es
"mpls ldp router-id loopback0")
no ftp-server write-enable
!
no crypto isakmp ccm
!
interface Loopback0
 ip address 100.5.5.5 255.255.255.255
!
interface FastEthernet0/0
 ip address 100.99.88.18 255.255.255.252
 duplex full
 tag-switching ip // TO INCLUDE THIS INTERFACE IN THE MPLS
DOMAIN
 no shutdown
!
interface FastEthernet1/0
 ip vrf forwarding Cankaya // TO INCLUDE THIS INTERFACE TO
RELATED VRF
 ip address 100.99.88.29 255.255.255.252
 no ip redirects
 no ip proxy-arp
 duplex full
 speed auto
 no shutdown
!
interface FastEthernet1/1
 ip address 200.99.88.1 255.255.255.252
 duplex full
 speed auto
 tag-switching ip
 no shutdown
!
router ospf 1
 router-id 100.5.5.5 // IDENTIFIES THE ROUTER IN THE OSPF
DOMAIN
 log-adjacency-changes
 network 100.0.0.0 0.255.255.255 area 0 // INCLUDES 100.0.0.0
NETWORK FOR OSPF AREA 0
 network 200.0.0.0 0.255.255.255 area 1 // INCLUDES 200.0.0.0
NETWORK FOR OSPF AREA 1
!
router bgp 20000
 bgp router-id 100.5.5.5 // IDENTIFIES THE ROUTER IN THE BGP
DOMAIN
 no bgp default ipv4-unicast // TO STATE AN ADDRESS-FAMILY
SPECIFICALLY
 bgp cluster-id 100.5.5.5 // TO INCLUDE MULTIPLE ROUTE-
REFLECTORS TO THE SAME CLUSTER
 bgp log-neighbor-changes
 neighbor INTERNAL peer-group // TO CREATE NEIGHBORSHIP WITH A
GROUP
 neighbor INTERNAL remote-as 20000 // TO SPECIFY THE AS NUMBER
OF THE ROUTERS IN THE GROUP
 neighbor INTERNAL password cankaya // IF A ROUTER IN THE
GROUP DOES NOT HAVE THE PASSWORD, IT CANNOT MAKE THE TCP
CONNECTION
 neighbor INTERNAL update-source Loopback0 // MAKES THE IGP
NEIGHBORSHIP VIA LOOPBACK0 INTERFACE
 neighbor 100.3.3.3 peer-group INTERNAL // TO INCLUDE THE
ROUTER IN THE GROUP
 neighbor 100.4.4.4 peer-group INTERNAL
 neighbor 100.6.6.6 peer-group INTERNAL
 neighbor 200.11.11.11 peer-group INTERNAL
 neighbor 200.12.12.12 peer-group INTERNAL
 !
 address-family ipv4 // TO BUILD IPv4 BGP
 neighbor INTERNAL activate // TO ACTIVATE BGP NEIGHBORSHIP
FOR THE ADDRESS-FAMILY
 neighbor INTERNAL route-reflector-client // TO ADVERTISE THE
ROUTES THAT LEARNED FROM IBGP TO OTHER IBGP NEIGHBORS (WITHOUT
THIS ALL THE ROUTERS IN THE IGP MUST BE FULLY MESHED)
 neighbor 100.3.3.3 peer-group INTERNAL // TO INCLUDE THE
ROUTER IN THE GROUP FOR THE ADDRESS-FAMILY
 neighbor 100.4.4.4 peer-group INTERNAL
 neighbor 100.6.6.6 peer-group INTERNAL
 neighbor 200.11.11.11 peer-group INTERNAL
 neighbor 200.12.12.12 peer-group INTERNAL
 no auto-summary // TO ADVERTISE SUBNETS CERTAINLY
 no synchronization // TO PREVENT ADVERTISING UNNECESSARY
ROUTES INTO IBGP
```

```
 aggregate-address 100.0.0.0 255.0.0.0 summary-only // TO
SUMMARIZE SPECIFIED SUBNET
 aggregate-address 200.0.0.0 255.0.0.0 summary-only
 exit-address-family
 !
 address-family vpnv4 // TO BUILD VPNv4 BGP
 neighbor INTERNAL activate // TO ACTIVATE BGP NEIGHBORSHIP
FOR THE ADDRESS-FAMILY
 neighbor INTERNAL route-reflector-client // TO ADVERTISE THE
ROUTES THAT LEARNED FROM IBGP TO OTHER IBGP NEIGHBORS (WITHOUT
THIS ALL THE ROUTERS IN THE IGP MUST BE FULLY MESHED)
 neighbor INTERNAL send-community extended // NECESSARY FOR
WORKING VPNv4 ADDRESS-FAMILY (SOME POLICIES CAN BE APPLIED
WITH COMMUNITIES)
 neighbor 100.3.3.3 peer-group INTERNAL // TO INCLUDE THE
ROUTER IN THE GROUP FOR THE ADDRESS-FAMILY
 neighbor 100.4.4.4 peer-group INTERNAL
 neighbor 100.6.6.6 peer-group INTERNAL
 exit-address-family
 !
 address-family ipv4 vrf Cankaya // TO CREATE AN ADDRESS-
FAMILY FOR RELATED VRF
 neighbor 100.99.88.30 remote-as 65110 // TO CREATE BGP
NEIGHBORSHIP
 neighbor 100.99.88.30 activate // TO ACTIVATE BGP
NEIGHBORSHIP
 no auto-summary // TO ADVERTISE SUBNETS CERTAINLY
 no synchronization // TO PREVENT ADVERTISING UNNECESSARY
ROUTES INTO IBGP
 exit-address-family
!
ip classless
!
no ip http server
no ip http secure-server
!
no cdp log mismatch duplex
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
!
End
```

**mpls_r6:**

```
Building configuration...
```

```
Current configuration : 2637 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname mpls_r6
!
boot-start-marker
boot-end-marker
!
no aaa new-model
ip subnet-zero
no ip icmp rate-limit unreachable
!
ip tcp synwait-time 5
ip cef
no ip domain lookup
!
ip vrf Cankaya // TO CREATE A VRF
 rd 65130:4 // TO SPECIFY THE VRF AND TO MAKE THE ROUTES OF
THAT VRF UNIQUE
 route-target export 65130:120 // TO LEAK THE ROUTES FROM THAT
VRF TO OTHER VRF's
 route-target import 65130:130 // TO LEAK THE ROUTES FROM
OTHER VRF's TO THAT VRF
!
ip ips po max-events 100
mpls label protocol ldp //CHOOSES MPLS LABEL PROTOCOL: LDP OR
TDP (TDP IS CISCO PROTOCOL)
tag-switching tdp router-id Loopback0 // TO BUILD LDP
NEIGHBORSHIP VIA Loopback0 INTERFACE (IN THE NEWER IOS'es
"mpls ldp router-id loopback0")
no ftp-server write-enable
!
no crypto isakmp ccm
!
interface Loopback0
 ip address 100.6.6.6 255.255.255.255
!
interface FastEthernet0/0
 ip address 100.99.88.14 255.255.255.252
 duplex full
 tag-switching ip // TO INCLUDE THIS INTERFACE IN THE MPLS
DOMAIN
 no shutdown
!
interface FastEthernet0/1
 ip vrf forwarding Cankaya // TO INCLUDE THIS INTERFACE TO
RELATED VRF
 ip address 100.99.88.33 255.255. 255.252
 no ip redirects
 no ip proxy-arp
 duplex full
 speed auto
 no shutdown
!
router ospf 1
```

```
 router-id 100.6.6.6 // IDENTIFIES THE ROUTER IN THE OSPF
DOMAIN
 log-adjacency-changes
 network 100.0.0.0 0.255.255.255 area 0 // INCLUDES 100.0.0.0
NETWORK FOR OSPF AREA 0
!
router bgp 20000
 bgp router-id 100.6.6.6 // IDENTIFIES THE ROUTER IN THE BGP
DOMAIN
 no bgp default ipv4-unicast // TO STATE AN ADDRESS-FAMILY
SPECIFICALLY
 bgp log-neighbor-changes
 neighbor INTERNAL peer-group // TO CREATE NEIGHBORSHIP WITH A
GROUP
 neighbor INTERNAL remote-as 20000 // TO SPECIFY THE AS NUMBER
OF THE ROUTERS IN THE GROUP
 neighbor INTERNAL password cankaya // IF A ROUTER IN THE
GROUP DOES NOT HAVE THE PASSWORD, IT CANNOT MAKE THE TCP
CONNECTION
 neighbor INTERNAL update-source Loopback0 // MAKES THE IGP
NEIGHBORSHIP VIA LOOPBACK0 INTERFACE
 neighbor 100.3.3.3 peer-group INTERNAL // TO INCLUDE THE
ROUTER IN THE GROUP
 neighbor 100.4.4.4 peer-group INTERNAL
 neighbor 100.5.5.5 peer-group INTERNAL
 !
 address-family ipv4 // TO BUILD IPv4 BGP
 neighbor INTERNAL activate // TO ACTIVATE BGP NEIGHBORSHIP
FOR THE ADDRESS-FAMILY
 neighbor 100.3.3.3 peer-group INTERNAL // TO INCLUDE THE
ROUTER IN THE GROUP FOR THE ADDRESS-FAMILY
 neighbor 100.4.4.4 peer-group INTERNAL
 neighbor 100.5.5.5 peer-group INTERNAL
 no auto-summary // TO ADVERTISE SUBNETS CERTAINLY
 no synchronization // TO PREVENT ADVERTISING UNNECESSARY
ROUTES INTO IBGP
 aggregate-address 100.0.0.0 255.0.0.0 summary-only // TO
SUMMARIZE SPECIFIED SUBNET
 exit-address-family
 !
 address-family vpnv4 // TO BUILD VPNv4 BGP
 neighbor INTERNAL activate // TO ACTIVATE BGP NEIGHBORSHIP
FOR THE ADDRESS-FAMILY
 neighbor INTERNAL route-reflector-client // TO ADVERTISE THE
ROUTES THAT LEARNED FROM IBGP TO OTHER IBGP NEIGHBORS (WITHOUT
THIS ALL THE ROUTERS IN THE IGP MUST BE FULLY MESHED)
 neighbor INTERNAL send-community extended // NECESSARY FOR
WORKING VPNv4 ADDRESS-FAMILY (SOME POLICIES CAN BE APPLIED
WITH COMMUNITIES)
 neighbor 100.3.3.3 peer-group INTERNAL // TO INCLUDE THE
ROUTER IN THE GROUP FOR THE ADDRESS-FAMILY
 neighbor 100.4.4.4 peer-group INTERNAL
 neighbor 100.5.5.5 peer-group INTERNAL
 exit-address-family
 !
 address-family ipv4 vrf Cankaya // TO CREATE AN ADDRESS-
FAMILY FOR RELATED VRF
 neighbor 100.99.88.34 remote-as 65120 // TO CREATE BGP
NEIGHBORSHIP
 neighbor 100.99.88.34 activate // TO ACTIVATE BGP
NEIGHBORSHIP
```

```
 no auto-summary  // TO ADVERTISE SUBNETS CERTAINLY
 no synchronization  // TO PREVENT ADVERTISING UNNECESSARY
ROUTES INTO IBGP
 exit-address-family
!
ip classless
!
no ip http server
no ip http secure-server
!
no cdp log mismatch duplex
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
!
end
```

**ospf_r1:**

```
Building configuration...

Current configuration : 2278 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ospf_r1
!
boot-start-marker
boot-end-marker
!
no aaa new-model
ip subnet-zero
no ip icmp rate-limit unreachable
!
ip tcp synwait-time 5
ip cef
no ip domain lookup
!
ip ips po max-events 100
no ftp-server write-enable
!
no crypto isakmp ccm
!
```

```
interface Loopback0
 ip address 200.11.11.11 255.255.255.255
!
interface FastEthernet0/0
 ip address 200.99.88.10 255.255.255.252
 duplex full
 no shutdown
!
interface FastEthernet1/0
 ip address 200.99.88.5 255.255.255.252
 duplex full
 speed auto
 no shutdown
!
interface FastEthernet1/1
 ip address 200.99.88.17 255.255.255.252
 duplex full
 speed auto
 no shutdown
!
router ospf 1
 router-id 200.11.11.11 // IDENTIFIES THE ROUTER IN THE OSPF
DOMAIN
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 1 // INCLUDES ALL THE
SUBNETS FOR OSPF ROUTING
!
router bgp 20000
 bgp router-id 200.11.11.11 // IDENTIFIES THE ROUTER IN THE
BGP DOMAIN
 bgp log-neighbor-changes
 neighbor INTERNAL peer-group // TO CREATE NEIGHBORSHIP WITH A
GROUP
 neighbor INTERNAL remote-as 20000 // TO SPECIFY THE AS NUMBER
OF THE ROUTERS IN THE GROUP
 neighbor INTERNAL password cankaya // IF A ROUTER IN THE
GROUP DOES NOT HAVE THE PASSWORD, IT CANNOT MAKE THE TCP
CONNECTION
 neighbor INTERNAL update-source Loopback0 // MAKES THE IGP
NEIGHBORSHIP VIA LOOPBACK0 INTERFACE
 neighbor 100.3.3.3 peer-group INTERNAL // TO INCLUDE THE
ROUTER IN THE GROUP
 neighbor 100.5.5.5 peer-group INTERNAL
 neighbor 200.99.88.6 remote-as 10000 // TO CREATE BGP
NEIGHBORSHIP
  !
 address-family ipv4 // TO BUILD IPv4 BGP
 neighbor INTERNAL activate // TO ACTIVATE BGP NEIGHBORSHIP
 neighbor INTERNAL next-hop-self // ADVERTISES THE ROUTES
BEHIND THE GROUP
 neighbor INTERNAL route-map setLocalPref out // APPLIES THE
ROUTE-MAP POLICY TO EGRESS DIRECTION
 neighbor 100.3.3.3 peer-group INTERNAL  // TO INCLUDE THE
ROUTER IN THE GROUP FOR THE ADDRESS-FAMILY
 neighbor 100.5.5.5 peer-group INTERNAL
 neighbor 200.99.88.6 activate // TO ACTIVATE BGP NEIGHBORSHIP
 neighbor 200.99.88.6 soft-reconfiguration inbound // TO FORCE
CLEAR BGP PROCESS FOR THAT NEIGHBOR (AFTER A POLICY CHANGE,
CLEARING BGP PROCESS TO TAKE AFFECT IS REQURIED)
 neighbor 200.99.88.6 route-map MED out // APPLIES THE ROUTE-
MAP POLICY TO EGRESS DIRECTION
```

```
 no auto-summary // TO ADVERTISE SUBNETS CERTAINLY
 no synchronization // TO PREVENT ADVERTISING UNNECESSARY
ROUTES INTO IBGP
 network 200.99.88.8 mask 255.255.255.252 // TO ADVERTISE THE
SPECIFIED SUBNET
 exit-address-family
!
ip classless
!
no ip http server
no ip http secure-server
!
no cdp log mismatch duplex
!
route-map setLocalPref permit 5 // TO APPLY THE POLICY FOR THE
SUBNET
 set local-preference 102
!
route-map setLocalPref permit 10
!
route-map MED permit 10
 set metric 100
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
!
End
```

**ospf_r2:**

```
Building configuration...

Current configuration : 2236 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ospf_r2
!
boot-start-marker
boot-end-marker
!
no aaa new-model
ip subnet-zero
```

```
no ip icmp rate-limit unreachable
!
ip tcp synwait-time 5
ip cef
no ip domain lookup
!
ip ips po max-events 100
no ftp-server write-enable
!
no crypto isakmp ccm
!
interface Loopback0
 ip address 200.12.12.12 255.255.255.255
!
interface FastEthernet0/0
 ip address 200.99.88.2 255.255.255.252
 duplex full
 no shutdown
!
interface FastEthernet1/0
 ip address 200.99.88.13 255.255.255.252
 duplex full
 speed auto
 no shutdown
!
interface FastEthernet1/1
 ip address 200.99.88.18 255.255.255.252
 duplex full
 speed auto
 no shutdown
!
router ospf 1
 router-id 200.12.12.12 // IDENTIFIES THE ROUTER IN THE OSPF
DOMAIN
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 1 // INCLUDES ALL THE
SUBNETS FOR OSPF ROUTING
!
router bgp 20000
 bgp router-id 200.12.12.12 // IDENTIFIES THE ROUTER IN THE
BGP DOMAIN
 bgp default local-preference 101 // TO GIVE THE ROUTER
PREFERENCE
 bgp log-neighbor-changes
 neighbor INTERNAL peer-group // TO CREATE NEIGHBORSHIP WITH A
GROUP
 neighbor INTERNAL remote-as 20000 // TO SPECIFY THE AS NUMBER
OF THE ROUTERS IN THE GROUP
 neighbor INTERNAL password cankaya // IF A ROUTER IN THE
GROUP DOES NOT HAVE THE PASSWORD, IT CANNOT MAKE THE TCP
CONNECTION
 neighbor INTERNAL update-source Loopback0 // MAKES THE IGP
NEIGHBORSHIP VIA LOOPBACK0 INTERFACE
 neighbor 100.3.3.3 peer-group INTERNAL // TO INCLUDE THE
ROUTER IN THE GROUP
 neighbor 100.5.5.5 peer-group INTERNAL
 neighbor 200.99.88.14 remote-as 10000 // TO CREATE BGP
NEIGHBORSHIP
 !
 address-family ipv4 // TO BUILD IPv4 BGP
 neighbor INTERNAL activate // TO ACTIVATE BGP NEIGHBORSHIP
 neighbor INTERNAL next-hop-self // ADVERTISES THE ROUTES
BEHIND THE GROUP
 neighbor INTERNAL route-map setLocalPref out // APPLIES THE
ROUTE-MAP POLICY TO EGRESS DIRECTION
 neighbor 100.3.3.3 peer-group INTERNAL // TO INCLUDE THE
ROUTER IN THE GROUP FOR THE ADDRESS-FAMILY
 neighbor 100.5.5.5 peer-group INTERNAL
 neighbor 200.99.88.14 activate // TO ACTIVATE BGP
NEIGHBORSHIP
 neighbor 200.99.88.14 soft-reconfiguration inbound // TO
FORCE CLEAR BGP PROCESS FOR THAT NEIGHBOR (AFTER A POLICY
CHANGE, CLEARING BGP PROCESS TO TAKE AFFECT IS REQURIED)
 neighbor 200.99.88.14 route-map MED out // APPLIES THE ROUTE-
MAP POLICY TO EGRESS DIRECTION
 no auto-summary // TO ADVERTISE SUBNETS CERTAINLY
 no synchronization // TO PREVENT ADVERTISING UNNECESSARY
ROUTES INTO IBGP
 exit-address-family
!
ip classless
!
no ip http server
no ip http secure-server
!
no cdp log mismatch duplex
!
route-map setLocalPref permit 10 // TO APPLY THE POLICY FOR
THE SUBNET
 set local-preference 100
!
route-map MED permit 5
 set metric 100
!
route-map MED permit 10
 set metric 200
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
!
end
```

**internet_gateway:**

```
Building configuration...

Current configuration : 1708 bytes
```

```
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname internet_gateway
!
boot-start-marker
boot-end-marker
!
no aaa new-model
ip subnet-zero
no ip icmp rate-limit unreachable
!
ip tcp synwait-time 5
ip cef
no ip domain lookup
!
ip ips po max-events 100
no ftp-server write-enable
!
no crypto isakmp ccm
!
interface Loopback0
 ip address 200.99.99.99 255.255.255.255
!
interface FastEthernet0/0
 ip address 8.8.8.1 255.255.255.0
 duplex full
 no shutdown
!
interface FastEthernet1/0
 ip address 200.99.88.6 255.255.255.252
 duplex full
 speed auto
 no shutdown
!
interface FastEthernet1/1
 ip address 200.99.88.14 255.255.255.252
 duplex full
 speed auto
 no shutdown
!
router bgp 10000
 bgp router-id 200.99.99.99 // IDENTIFIES THE ROUTER IN THE
BGP DOMAIN
 bgp log-neighbor-changes
 neighbor 200.99.88.5 remote-as 20000 // TO CREATE BGP
NEIGHBORSHIP
 neighbor 200.99.88.13 remote-as 20000
 !
 address-family ipv4 // TO BUILD IPv4 BGP
 neighbor 200.99.88.5 activate // TO ACTIVATE BGP NEIGHBORSHIP
 neighbor 200.99.88.13 activate
 no auto-summary // TO ADVERTISE SUBNETS CERTAINLY
 no synchronization  // TO PREVENT ADVERTISING UNNECESSARY
ROUTES INTO IBGP
 network 8.8.8.0 mask 255.255.255.0 // TO ADVERTISE THE
SPECIFIED SUBNET
 exit-address-family
```

```
!
ip classless
!
no ip http server
no ip http secure-server
!
no cdp log mismatch duplex
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
!
end
```

**gateway_1:**

```
Building configuration...

Current configuration : 2586 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname gateway_1
!
boot-start-marker
boot-end-marker
!
no aaa new-model
ip subnet-zero
no ip icmp rate-limit unreachable
!
ip tcp synwait-time 5
ip cef
no ip domain lookup
!
ip ips po max-events 100
no ftp-server write-enable
!
no crypto isakmp ccm
!
interface Loopback0
 ip address 100.7.7.7 255.255.255.255
!
interface FastEthernet0/0
```

```
 no ip address
 duplex full
 no shutdown
!
interface FastEthernet0/0.100 // SUBINTERFACE: IF THERE IS A
NEED TO CONNECT MORE THAN ONE NETWORK FROM SAME PHYSICAL
INTERFACE
 encapsulation dot1Q 100 // DOT1Q OR ISL (ISL IS CISCO
PROTOCOL)
 ip address 100.99.88.22 255.255.255.252
 no ip redirects
 no ip proxy-arp
 ip nat inside // STATES INBOUND NAT INTERFACE
 ip virtual-reassembly
!
interface FastEthernet0/0.101
 encapsulation dot1Q 101
 ip address 101.99.88.22 255.255.255.252
 no ip redirects
 no ip proxy-arp
 ip nat inside // STATES INBOUND NAT INTERFACE
 ip virtual-reassembly
!
interface FastEthernet0/0.102
 encapsulation dot1Q 102
 ip address 102.99.88.22 255.255.255.252
 ip nat outside // STATES OUTBOUND NAT INTERFACE
 ip virtual-reassembly
!
interface FastEthernet1/0
 ip address 10.0.0.1 255.255.255.252
 ip nat inside // STATES INBOUND NAT INTERFACE
 ip virtual-reassembly
 duplex full
 speed auto
 no shutdown
!
router ospf 1
 router-id 100.7.7.7 // IDENTIFIES THE ROUTER IN THE OSPF
DOMAIN
 log-adjacency-changes
 redistribute bgp 65130 subnets // ADVERTISES THE ROUTES THAT
LEARNED FROM BGP INTO OSPF NETWORK
 network 10.0.0.0 0.255.255.255 area 0 // INCLUDES 10.0.0.0
NETWORK FOR OSPF ROUTING
 default-information originate // ADVERTISES THE DEFAULT ROUTE
INTO OSPF AREA 1
!
router bgp 65130
 no synchronization // TO PREVENT ADVERTISING UNNECESSARY
ROUTES INTO IBGP
 bgp router-id 100.7.7.7 // IDENTIFIES THE ROUTER IN THE BGP
DOMAIN
 bgp default local-preference 101 // TELLS THE OTHER ROUTERS
THAT THERE IS A PRIORITY
 bgp log-neighbor-changes
 redistribute static // ADVERTISES THE STATICALLY CONFIGURED
ROUTES INTO BGP
 redistribute ospf 1 // ADVERTISES THE OSPF ROUTES INTO BGP
 neighbor 100.99.88.21 remote-as 20000 // CREATES NEIGBORSHIP
 neighbor 100.99.88.21 default-originate // INJECTS DEFAULT
ROUTE FROM THAT ROUTER TO THE NEIGHBOR
 neighbor 100.99.88.21 route-map MED out // APPLIES THE ROUTE-
MAP POLICY TO EGRESS DIRECTION
 neighbor 101.99.88.21 remote-as 20000  // TO CREATE BGP
NEIGHBORSHIP
 neighbor 102.99.88.21 remote-as 20000
 neighbor 100.8.8.8 remote-as 65130
 neighbor 100.8.8.8 update-source Loopback0 // MAKES THE IGP
NEIGHBORSHIP VIA LOOPBACK0 INTERFACE
 neighbor 100.8.8.8 next-hop-self  // ADVERTISES THE ROUTES
BEHIND THIS NEIGHBOR
 no auto-summary // TO ADVERTISE SUBNETS CERTAINLY
!
ip classless
ip route 0.0.0.0 0.0.0.0 102.99.88.21
!
no ip http server
no ip http secure-server
!
ip nat inside source list 199 interface FastEthernet0/0.102
overload // MAKES NAT DYNAMICALLY. ALL THE PRIVATE IP
ADDRESSES BEHIND THIS INTERFACE GETS SAME PUBLIC IP ADDRESS
(OVERLOAD:PAT)
!
access-list 199 permit ip any any
no cdp log mismatch duplex
!
route-map MED permit 10 // TO CREATE A POLICY TO A SPECIFIC
ADDRESS
 set metric 100
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
!
end
```

**gateway_2:**

```
Building configuration...

Current configuration : 2655 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
```

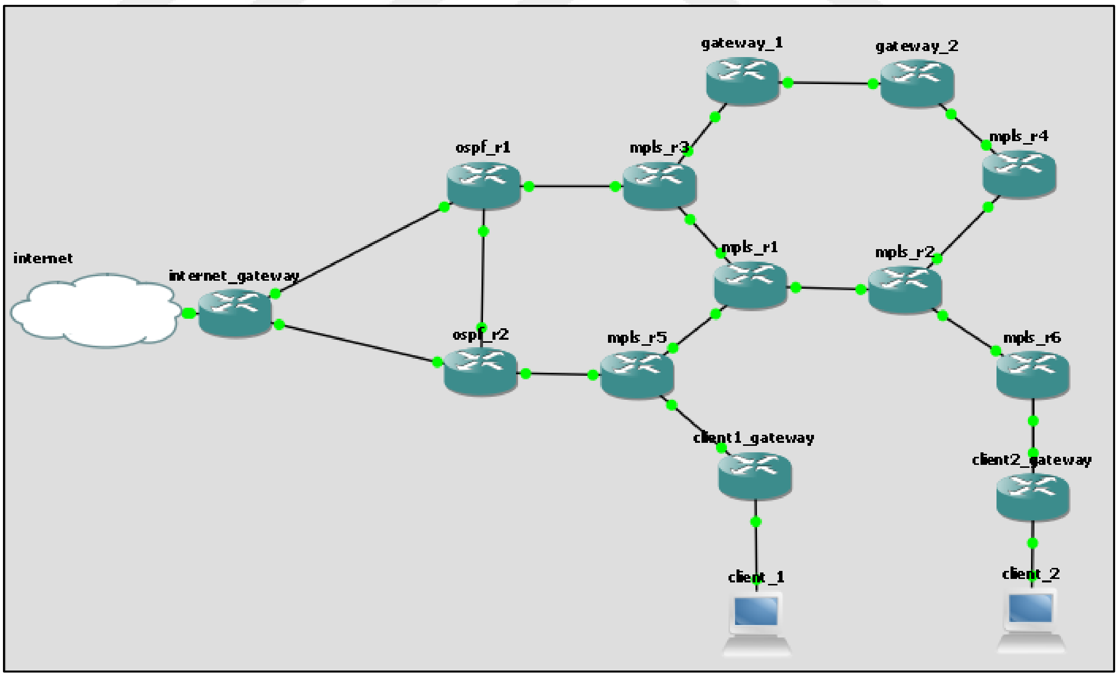
```
no service password-encryption
!
hostname gateway_2
!
boot-start-marker
boot-end-marker
!
no aaa new-model
ip subnet-zero
no ip icmp rate-limit unreachable
!
ip tcp synwait-time 5
ip cef
no ip domain lookup
!
!
ip ips po max-events 100
no ftp-server write-enable
!
no crypto isakmp ccm
!
interface Loopback0
 ip address 100.8.8.8 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 duplex full
 no shutdown
!
interface FastEthernet0/0.100 // SUBINTERFACE: IF THERE IS A
NEED TO CONNECT MORE THAN ONE NETWORK FROM SAME PHYSICAL
INTERFACE
 encapsulation dot1Q 100 // DOT1Q OR ISL (ISL IS CISCO
PROTOCOL)
 ip address 100.99.88.26 255.255.255.252
 no ip redirects
 no ip proxy-arp
 ip nat inside // STATES INBOUND NAT INTERFACE
 ip virtual-reassembly
!
interface FastEthernet0/0.101
 encapsulation dot1Q 101
 ip address 101.99.88.26 255.255.255.252
 no ip redirects
 no ip proxy-arp
 ip nat inside // STATES INBOUND NAT INTERFACE
 ip virtual-reassembly
!
interface FastEthernet0/0.102
 encapsulation dot1Q 102
 ip address 102.99.88.26 255.255.255.252
 ip nat outside // STATES OUTBOUND NAT INTERFACE
 ip virtual-reassembly
!
interface FastEthernet1/0
 ip address 10.0.0.2 255.255.255.252
 ip nat inside // STATES INBOUND NAT INTERFACE
 ip virtual-reassembly
 duplex full
 speed auto
 no shutdown
```

```
!
router ospf 1
 router-id 100.8.8.8 // IDENTIFIES THE ROUTER IN THE OSPF
DOMAIN
 log-adjacency-changes
 redistribute bgp 65130 subnets // ADVERTISES THE ROUTES THAT
LEARNED FROM BGP INTO OSPF NETWORK
 network 10.0.0.0 0.255.255.255 area 0 // INCLUDES 10.0.0.0
NETWORK FOR OSPF ROUTING
 default-information originate // ADVERTISES THE DEFAULT ROUTE
INTO OSPF AREA 1
!
router bgp 65130
 bgp router-id 100.8.8.8 // IDENTIFIES THE ROUTER IN THE BGP
DOMAIN
 no bgp default ipv4-unicast // TO STATE AN ADDRESS-FAMILY
SPECIFICALLY
 bgp log-neighbor-changes
 neighbor 100.99.88.25 remote-as 20000 // TO CREATE BGP
NEIGHBORSHIP
 neighbor 101.99.88.25 remote-as 20000
 neighbor 102.99.88.25 remote-as 20000
 neighbor 100.7.7.7 remote-as 65130
 neighbor 100.7.7.7 update-source Loopback0 // MAKES THE IGP
NEIGHBORSHIP VIA LOOPBACK0 INTERFACE
 !
 address-family ipv4 // TO BUILD IPv4 BGP
 redistribute static // ADVERTISES THE STATICALLY CONFIGURED
ROUTES INTO BGP
 redistribute ospf 1 // ADVERTISES THE OSPF ROUTES INTO BGP
 neighbor 100.99.88.25 activate
 neighbor 100.99.88.25 default-originate route-map MED //
INJECTS DEFAULT ROUTE FROM THAT ROUTER TO THE NEIGHBOR IF THE
ROUTE-MAP POLICY
 neighbor 100.99.88.25 route-map MED out // APPLIES THE ROUTE-
MAP POLICY TO EGRESS DIRECTION
 neighbor 101.99.88.25 activate // TO ACTIVATE BGP
NEIGHBORSHIP
 neighbor 102.99.88.25 activate
 neighbor 100.7.7.7 activate
 neighbor 100.7.7.7 next-hop-self // ADVERTISES THE ROUTES
BEHIND THIS NEIGHBOR
 no auto-summary // TO ADVERTISE SUBNETS CERTAINLY
 no synchronization // TO PREVENT ADVERTISING UNNECESSARY
ROUTES INTO IBGP
 exit-address-family
!
ip classless
ip route 0.0.0.0 0.0.0.0 102.99.88.25
!
no ip http server
no ip http secure-server
!
ip nat inside source list 199 interface FastEthernet0/0.102
overload // MAKES NAT DYNAMICALLY. ALL THE PRIVATE IP
ADDRESSES BEHIND THIS INTERFACE GETS SAME PUBLIC IP ADDRESS
(OVERLOAD:PAT)
!
access-list 199 permit ip any any
no cdp log mismatch duplex
!
```

```
route-map MED permit 10 // TO CREATE A POLICY TO A SPECIFIC ADDRESS
 set metric 101
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
!
end
```

## APPENDIX B

### Connection of Routers



# CURRICULUM VITAE

***Ali Murat KARAOGLU***

*21.sk 4/4, postal code:06510, Emek Çankaya/ANKARA, TURKEY*

*E-Mail:* *alimuratkaraoglu@gmail.com*

*Phone: +90 501 911 7347*

---

## *Personal Information*

***Date of Birth:*** *13.11.1987*
***Education:*** *Master's Degree*
***Driving Licence:*** *B (2006)*
***Marital Status:*** *Single*
***Military Obligation:*** *Done*
***Smoking:*** *No*

## *Experience*

***Penta Teknoloji*** *Jan'2016–Present*

***Sales Support Specialist – Huawei PreSales***

- *Doing PreSales activities and trying to create new sales opportunities,*
- *Making IP product configurations and offering network design solutions,*
- *Doing POC tests and giving product training to partners,*
- *Writing new RFP documents and replying puplished RFPs,*
- *Comparing different vendors' products and making price analysis about them, and making end-user visits.*

***Sentim Bilişim*** *Jul'2013–Jan'2016*

***Network Engineer***

- *Providing onsite support, determining problems and troubleshooting,*

- *Setting up networking devices (routing, switching, wireless, collaboration ) due to new projects,*
- *Working with several vendor technologies such as: Cisco, Huawei, HP, Aruba, Allied Telesis, Enterasys, Citrix,*
- *Following new technologies.*

***Servus Bilgisayar A.Ş*** *Feb'2013–Jul'2013*

***Pre-Sales Consultant***

- *Making product configurations to obtain their cost sheets,*
- *Determining technical specifications,*
- *Organizing technical details between engineers and account managers for submitting to the customers.*

***Servus Bilgisayar A.Ş*** *Jan'2012 – Feb'2013*

***Network Specialist***

- *Providing onsite support, determining problems and troubleshooting,*
- *Installing and configuring computer systems using different build technologies,*
- *Analyzing hardware problems on many devices, such as:*
  - Switches and Routers: Cisco, Allied Telesis, Enterasys, HP
- *Answering incoming calls from clients and giving remote support on related technologies.*

***Servus Bilgisayar A.Ş*** *Mar'2011 - Jan'2012*

***Network Infrastructure Team Leader***

- *Managing all the field processes on the related project,*
- *Directing field technicians during operations,*
- *Determining and providing materials,*

- *Meeting the needs of the technicians,*
- *Managing client expectations,*

***Six Flags St.Louis U.S.A*** *Jun’2008-Sep’2008*

***Photographer***

- *Taking pictures of the entertainment park’s customers,*
- *Editing the pictures on an appropriate software,*
- *Selling the pictures*

## *Major Projects*

*Fatih – Switching and Network Infrastructure*
*Turkish Naval Forces – R&S Support*
*Sca Yildiz – Cisco Wireless Project*
*Wavin – Cisco Video Conference*
*Turkish Armed Forces - Switching and Network Infrastructure*
*Vakifbank – R&S Support*

## *Trainings*

***Cisco CCNA Training*** *Oct’2009 – Apr’2013*
*METU Continuing Education Center*

***Cisco CCNP Training*** *Jul’2013 – Oct’2013*
*Academytech*

***Cisco CCIE Training*** *Dec’2014 – May’2015*
*CLIGuru*

***Huawei HCNP Training*** *Sep’2014*
*Huawei*

## *Certifications*

***CCNA*** *Nov’2010– Nov’2013*
*Cisco*

***CCNP*** *Nov’2014-Present*
*Cisco (Valid through Nov 2017)*

***CCIE (R&S# 50186)*** *Sep’2015-Present*
*Cisco (Valid through Sep 2017)*

***CCA-N*** *Apr’2013-Present*
*Citrix (Valid through Apr 2017)*

***MCTS*** *Oct’2012-Present*
*Microsoft:*

- *Windows 7 Configuring,*
- *Windows Server 2008 Active Directory,*
- *Windows Server 2008 Applications.*

## *Education*

***Master’s Degree*** *Jun’2012 – Present*

*Computer Engineering, Cankaya University*
*(Studying for thesis currently )*

***Undergraduate*** *Sep’2005 – Jun’2010*

*Physics, Ankara University*

***High School*** *Sep’2001 – Jun’2004*

*Bahcelievler Deneme High School*

## *Interests*

*Travelling, Soccer, Bowling, Fishing, Theater, Cinema, PC Games.*