MBM
Madan Bhandari Memorial College

Lab no: 11 ***Date: 2024/09/26***

Title: Dynamic Routing Implementation using BGP

Objectives:

- To implement BGP for network routing
- To configure and verify BGP routing across multiple routers

Background Theory:

Border Gateway Protocol (BGP) is a robust and scalable routing protocol used for inter-domain routing. It operates using path vector routing and exchanges routing information between autonomous systems (AS). Unlike OSPF, BGP is ideal for routing between large networks (e.g., ISPs) and is capable of handling multiple policies and preferences. BGP uses attributes to make routing decisions and is highly flexible, making it a preferred choice for large-scale network infrastructures.

Process for Dynamic Routing using BGP:

Step 1: Setup an environment with 2 routers, 2 switches and some desktops, ensuring all devices are connected properly.

Step 2: Manually assign IP addresses and subnet masks to the desktops, router interfaces, and the interface of each router.

Step 3: Access the CLI of the router and enable it using ***"en"*** command.

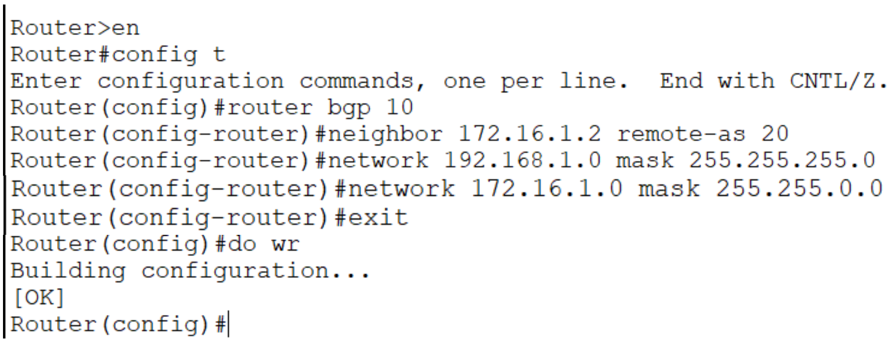
Step 4: Enter the global configuration mode with command ***"config t".***

Step 5: Access the BGP configuration mode and assign a autonomous system number (ASN) using the command: ***"router bgp [ASN]".***

Step 6: Configure the neighbor relationships by specifying the IP addresses of neighboring routers and their ASN using the command: ***"neighbor [IP-address] remote-as [ASN]".***

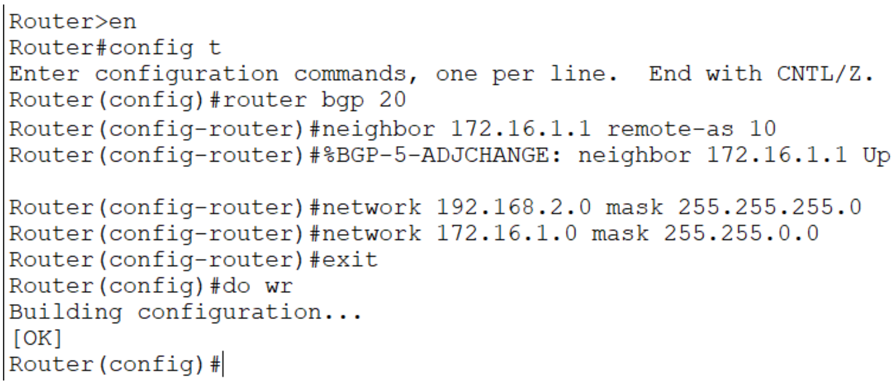
Step 7: Advertise the connected networks within the BGP routing table using the command: ***"network [network-address] mask [subnet-mask]".***

Step 8: Repeat the configuration steps on all routers, ensuring that neighbor relationships and network advertisements are consistent.



```
Router>en
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router bgp 10
Router(config-router)#neighbor 172.16.1.2 remote-as 20
Router(config-router)#network 192.168.1.0 mask 255.255.255.0
Router(config-router)#network 172.16.1.0 mask 255.255.0.0
Router(config-router)#exit
Router(config)#do wr
Building configuration...
[OK]
Router(config)#
```

Fig: BGP Configuration of Router1



```
Router>en
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router bgp 20
Router(config-router)#neighbor 172.16.1.1 remote-as 10
Router(config-router)#%BGP-5-ADJCHANGE: neighbor 172.16.1.1 Up

Router(config-router)#network 192.168.2.0 mask 255.255.255.0
Router(config-router)#network 172.16.1.0 mask 255.255.0.0
Router(config-router)#exit
Router(config)#do wr
Building configuration...
[OK]
Router(config)#
```

Fig: BGP Configuration of Router2

Observation and Findings:

Implementing BGP Protocol for dynamic routing using 2 routers, 2 switches and some desktops.

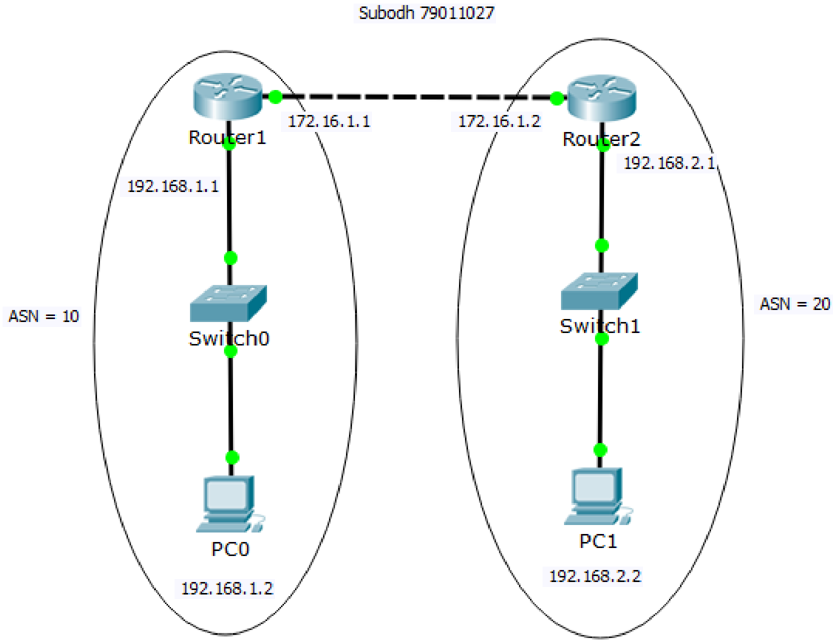
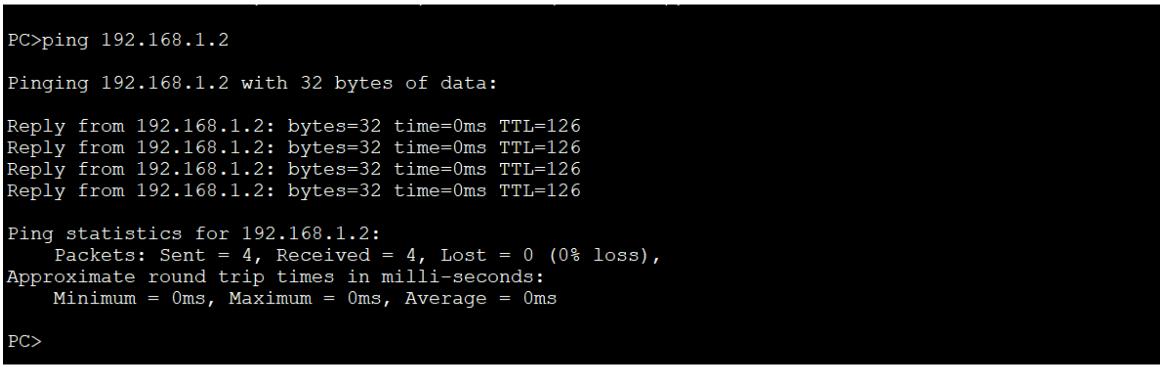


Fig: Dynamic Routing using BGP

Output:



```
PC>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time=0ms TTL=126
Reply from 192.168.1.2: bytes=32 time=0ms TTL=126
Reply from 192.168.1.2: bytes=32 time=0ms TTL=126
Reply from 192.168.1.2: bytes=32 time=0ms TTL=126

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>
```

Discussions:

The routers were configured to implement dynamic routing using Border Gateway Protocol (BGP) for managing network traffic between different subnets. The process involved assigning IP addresses to router interfaces and desktops, enabling BGP on the routers, and specifying the networks to be advertised. BGP established peer connections between routers using their physical interface IP addresses, exchanging routing information to populate the routing tables dynamically. Successful ping tests between devices confirmed that the BGP-based dynamic routes were correctly implemented, enabling seamless communication between networks. This highlights BGP's capability to efficiently manage routing policies and interconnect networks without requiring loopback interfaces.

Conclusion:

Implementing dynamic routing using BGP allowed for efficient and automated management of network traffic between subnets. The successful connectivity tests validated that BGP was configured correctly, ensuring adaptive and policy-driven data flow across the networks while minimizing manual intervention. Even without loopback interfaces, BGP's path-vector mechanism and scalability proved effective for managing routing in a distributed network environment.