

# ACN Mininet Assignment: BGP Path Hijacking

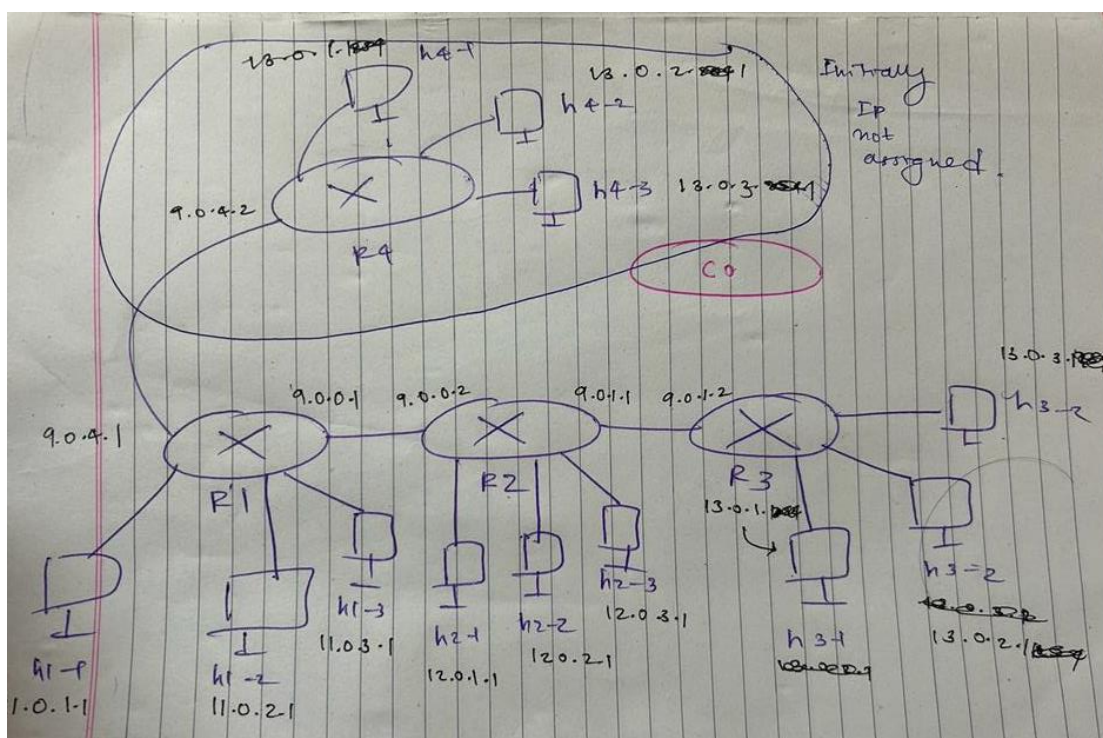
**Q1:** Draw the topology diagram used for this demo. How many hosts are there and how many Routers are present in the emulated network inside mininet? How many hosts are present in each subnet? (Hint: Each router here represents an autonomous system)

Ans:

```

mininet@mininet-vm: ~/bgp
File Edit Tabs Help
<Controller c0: 127.0.0.1:6633 pid=1959>
mininet> net
h1-1 h1-1-eth0:R1-eth1
h1-2 h1-2-eth0:R1-eth2
h1-3 h1-3-eth0:R1-eth3
h2-1 h2-1-eth0:R2-eth1
h2-2 h2-2-eth0:R2-eth2
h2-3 h2-3-eth0:R2-eth3
h3-1 h3-1-eth0:R3-eth1
h3-2 h3-2-eth0:R3-eth2
h3-3 h3-3-eth0:R3-eth3
h4-1 h4-1-eth0:R4-eth1
h4-2 h4-2-eth0:R4-eth2
h4-3 h4-3-eth0:R4-eth3
R1 R1-eth1:h1-1-eth0 R1-eth2:h1-2-eth0 R1-eth3:h1-3-eth0 R1-eth4:R2-eth4 R1-eth5
:R4-eth4
R2 R2-eth1:h2-1-eth0 R2-eth2:h2-2-eth0 R2-eth3:h2-3-eth0 R2-eth4:R1-eth4 R2-eth5
:R3-eth4
R3 R3-eth1:h3-1-eth0 R3-eth2:h3-2-eth0 R3-eth3:h3-3-eth0 R3-eth4:R2-eth5
R4 R4-eth1:h4-1-eth0 R4-eth2:h4-2-eth0 R4-eth3:h4-3-eth0 R4-eth4:R1-eth5
c0
mininet> links
*** Unknown command: links
mininet>
  
```

#Host=12  
#Routers=4  
#Host per subnet=3



Ans:

[illegible]

```

Node: R4
root@inet-ns1:~# ip netns exec r4 eth1 Link encap:Ethernet HWaddr 52:54:00:12:3d:f6:ca7a
  inet6 addr: fe80::5254:0012:3d:f6:ca7a ScopeLink
  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
  RX packets:18880 errors:0 dropped:0 overruns:0 frame:0
  TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:648 (648.0 B) TX bytes:648 (648.0 B)

r4-eth2 Link encap:Ethernet HWaddr f6:23:ac:3a:77:75:7d3a
  inet6 addr: fe80::f623:ac3a:7775:7d3a ScopeLink
  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
  RX packets:18880 errors:0 dropped:0 overruns:0 frame:0
  TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:648 (648.0 B) TX bytes:648 (648.0 B)

r4-eth3 Link encap:Ethernet HWaddr ac:1b:17:06:94:98
  inet6 addr: fe80::ac1b:1706:9498:0 ScopeLink
  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
  RX packets:18880 errors:0 dropped:0 overruns:0 frame:0
  TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:648 (648.0 B) TX bytes:648 (648.0 B)

r4-eth4 Link encap:Ethernet HWaddr 02:24:76:07:71:50
  inet6 addr: fe80::0224:7607:7150:0 ScopeLink
  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
  RX packets:18880 errors:0 dropped:0 overruns:0 frame:0
  TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:648 (648.0 B) TX bytes:648 (648.0 B)

root@inet-ns1:~# ip netns exec r4 eth1

```

```

root@mininet-vx:~/bgp# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:1b:7d:48
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:336 errors:0 dropped:0 overruns:0 frame:0
          TX packets:341 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:46057 (46.0 KB)  TX bytes:30186 (30.1 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:100 (100.0 B)  TX bytes:100 (100.0 B)

root@mininet-vx:~/bgp#

```

**Q3:** Check the reachability for host “h3-1” from at least three other hosts. Post screenshots as proof that you are able to communicate with “h3-1”. (hint: use xterm “hostname” to log in the host and test reachability)\

Ans:

```

Node: h2-1
root@mininet-vm:~/bgp# ping -c 5 11.0.1.1
PING 11.0.1.1 (11.0.1.1) 56(84) bytes of data:
64 bytes from 11.0.1.1: icmp_seq=1 ttl=62 time=0.065 ms
64 bytes from 11.0.1.1: icmp_seq=2 ttl=62 time=0.121 ms
64 bytes from 11.0.1.1: icmp_seq=3 ttl=62 time=0.063 ms
64 bytes from 11.0.1.1: icmp_seq=4 ttl=62 time=0.081 ms
64 bytes from 11.0.1.1: icmp_seq=5 ttl=62 time=0.059 ms

--- 11.0.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3997ms
rtt min/avg/max/mdev = 0.063/0.075/0.121/0.026 ms
root@mininet-vm:~/bgp#

```

```

Node: h3-1
root@mininet-vm:~/bgp# ping -c 5 11.0.1.1
PING 11.0.1.1 (11.0.1.1) 56(84) bytes of data:
64 bytes from 11.0.1.1: icmp_seq=1 ttl=61 time=0.065 ms
64 bytes from 11.0.1.1: icmp_seq=2 ttl=61 time=0.127 ms
64 bytes from 11.0.1.1: icmp_seq=3 ttl=61 time=0.071 ms
64 bytes from 11.0.1.1: icmp_seq=4 ttl=61 time=0.147 ms
64 bytes from 11.0.1.1: icmp_seq=5 ttl=61 time=0.113 ms

--- 11.0.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.065/0.104/0.147/0.033 ms
root@mininet-vm:~/bgp#

```

```

Node: h4-1
root@mininet-vm:~/bgp# ping -c 5 11.0.1.1
PING 11.0.1.1 (11.0.1.1) 56(84) bytes of data:
From 13.0.1.1 icmp_seq=1 Destination Host Unreachable
From 13.0.1.1 icmp_seq=2 Destination Host Unreachable
From 13.0.1.1 icmp_seq=3 Destination Host Unreachable
From 13.0.1.1 icmp_seq=4 Destination Host Unreachable
From 13.0.1.1 icmp_seq=5 Destination Host Unreachable

--- 11.0.1.1 ping statistics ---
5 packets transmitted, 0 received, +5 errors, 100% packet loss, time 4025ms
pipe 3
root@mininet-vm:~/bgp#

```

**Q4:** What do you see at the router R1 (AS1). Explain your interpretation of the entries in the BGP table with screenshots.

Ans: Based on the output I can interpret that if the destination IP is of current subnet, it will flow the packet onto the link which can be seen by the corresponding next hop entry 0.0.0.0 and if the destination network id are of subnet 2 and 3 it will pass it to the interface 4<sup>th</sup> of the subnet 2 : 9.0.0.2

```

mininet@mininet-vm: ~/bgp
File Edit Tabs Help
Escape character is '^['.

Hello, this is Quagga (version 0.99.22.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:
bgpd-R1> en
Password:
bgpd-R1# sh ip bgp
BGP table version is 0, local router ID is 9.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network        Next Hop        Metric LocPrf Weight Path
*> 11.0.0.0        0.0.0.0          0         32768 i
*> 12.0.0.0         9.0.0.2          0           2 i
*> 13.0.0.0         9.0.0.2          0           2 3 i

Total number of prefixes 3
bgpd-R1#

```

**Q5:** Perform the same for the router R2. Post screenshot. Are the entries in the routers different from each other? Why? What do they signify?

Ans: Based on the topological position of router R2, the entries of R1 and R2 will differ for each of the network ID.

If the packet belongs to subnet 2, it will put it on the link. On the other hand for subnet 1 and subnet 2 it will forward to their respective interface.

```
mininet@mininet-vm: ~/bgp
File Edit Tabs Help
Escape character is '^['.

Hello, this is Quagga (version 0.99.22.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:
bgpd-R2> en
Password:
bgpd-R2# sh ip bgp
BGP table version is 0, local router ID is 9.0.0.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network        Next Hop           Metric LocPrf Weight Path
*> 11.0.0.0        9.0.0.1             0         0 1 i
*> 12.0.0.0        0.0.0.0             0        32768 i
*> 13.0.0.0        9.0.1.2             0         0 3 i

Total number of prefixes 3
bgpd-R2#
```

**Q6.** Post contents of forwarding tables at R1 and R2 using “route -n” command by logging into respective routers. Explain the difference between R1’s BGP table and its forwarding table and how the BGP table is used to populate entries in the forwarding table of R1.

Ans: Each router will construct its forwarding table using their respective BGP tables. BGP table contains the information about neighbouring autonomous system whereas forwarding table used to do destination-based routing. Here BGP table, tells how it can go to destination asynchronous system.

```
Node: R1
root@mininet-vm:~/bgp# route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
9.0.0.0 0.0.0.0 255.255.255.0 U 0 0 0 R1-eth4
9.0.4.0 0.0.0.0 255.255.255.0 U 0 0 0 R1-eth5
11.0.1.0 0.0.0.0 255.255.255.0 U 0 0 0 R1-eth1
11.0.2.0 0.0.0.0 255.255.255.0 U 0 0 0 R1-eth2
12.0.3.0 0.0.0.0 255.255.255.0 U 0 0 0 R1-eth3
12.0.0.0 9.0.0.2 255.0.0.0 UG 0 0 0 R1-eth4
13.0.0.0 9.0.0.2 255.0.0.0 UG 0 0 0 R1-eth4
root@mininet-vm:~/bgp#
```

```
Node: R2
root@mininet-vm:~/bgp# route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
9.0.0.0 0.0.0.0 255.255.255.0 U 0 0 0 R2-eth4
9.0.1.0 0.0.0.0 255.255.255.0 U 0 0 0 R2-eth5
11.0.0.0 9.0.0.1 255.0.0.0 UG 0 0 0 R2-eth4
12.0.1.0 0.0.0.0 255.255.255.0 U 0 0 0 R2-eth1
12.0.2.0 0.0.0.0 255.255.255.0 U 0 0 0 R2-eth2
12.0.3.0 0.0.0.0 255.255.255.0 U 0 0 0 R2-eth3
13.0.0.0 9.0.1.2 255.0.0.0 UG 0 0 0 R2-eth5
root@mininet-vm:~/bgp#
```



**Q7:** Open wireshark and listen to an interface (you have to choose the appropriate one). Post screenshots of the HTTP GET requests and the response you received. This should correspond to the output seen on the terminal window. (hint: to open wireshark: xterm into a host and type “sudo wireshark &”; ignore any error messages that pop up. Then choose an interface and listen to it.)

Ans:

The first screenshot shows a terminal window titled 'mininet@mininet-vm: ~/bgp'. The user has run 'cd bgp' and './website.sh'. The output shows a series of 'Fri Dec 1 09:45:57 PST 2023 -- <h1>Default web server</h1>' messages, indicating a web server is running and responding to requests.

The second screenshot shows a Wireshark window titled 'Mininet-Tutorial [Running] - Oracle VM VirtualBox'. It is capturing traffic on interface 'eth0'. The packet list shows several HTTP requests and responses. The packet details pane shows the structure of an HTTP GET request and its corresponding response, including the status line '200 OK' and the response body.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	11.0.1.1	13.0.1.1	TCP	76	34187 > http [SYN] Seq=0 Win=29288 Len=0 MSS=1460 SACK_PERM=1 TSval=544961 TSecr=0 WS=512
2	0.00028800	13.0.1.1	11.0.1.1	TCP	76	http > 34187 [SYN, ACK] Seq=0 Ack=1 Win=29696 Len=0 MSS=1460 SACK_PERM=1 TSval=544961 TSecr=544961 WS=512
3	0.00035600	11.0.1.1	13.0.1.1	TCP	68	34187 > http [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=544961 TSecr=544961
4	0.00143600	11.0.1.1	13.0.1.1	HTTP	140	GET / HTTP/1.1
5	0.00155600	13.0.1.1	11.0.1.1	TCP	68	http > 34187 [ACK] Seq=1 Ack=73 Win=29184 Len=0 TSval=544961 TSecr=544961
6	0.00268800	13.0.1.1	11.0.1.1	TCP	85	[TCP segment of a reassembled PDU]
7	0.00279600	11.0.1.1	13.0.1.1	TCP	68	34187 > http [ACK] Seq=73 Ack=18 Win=29696 Len=0 TSval=544961 TSecr=544961
8	0.00285600	13.0.1.1	11.0.1.1	TCP	165	[TCP segment of a reassembled PDU]
9	0.00285600	11.0.1.1	13.0.1.1	TCP	68	34187 > http [ACK] Seq=73 Ack=55 Win=29696 Len=0 TSval=544961 TSecr=544961
10	0.00303600	13.0.1.1	11.0.1.1	TCP	165	[TCP segment of a reassembled PDU]
11	0.00304000	11.0.1.1	13.0.1.1	TCP	68	34187 > http [ACK] Seq=73 Ack=92 Win=29696 Len=0 TSval=544961 TSecr=544961
12	0.00314000	13.0.1.1	11.0.1.1	TCP	93	[TCP segment of a reassembled PDU]
13	0.00315600	11.0.1.1	13.0.1.1	TCP	68	34187 > http [ACK] Seq=73 Ack=117 Win=29696 Len=0 TSval=544961 TSecr=544961
14	0.00324000	13.0.1.1	11.0.1.1	TCP	76	[TCP segment of a reassembled PDU]
15	0.00325600	11.0.1.1	13.0.1.1	TCP	68	34187 > http [ACK] Seq=73 Ack=119 Win=29696 Len=0 TSval=544961 TSecr=544961
16	0.00334000	13.0.1.1	11.0.1.1	HTTP	96	Continuation or non-HTTP traffic
17	0.00335600	11.0.1.1	13.0.1.1	TCP	68	34187 > http [ACK] Seq=73 Ack=147 Win=29696 Len=0 TSval=544961 TSecr=544961
18	0.00354000	13.0.1.1	11.0.1.1	TCP	68	http > 34187 [FIN, ACK] Seq=147 Ack=73 Win=29184 Len=0 TSval=544961 TSecr=544961
19	0.00446000	11.0.1.1	13.0.1.1	TCP	68	34187 > http [FIN, ACK] Seq=73 Ack=148 Win=29696 Len=0 TSval=544961 TSecr=544961
20	0.00450600	13.0.1.1	11.0.1.1	TCP	68	http > 34187 [ACK] Seq=148 Ack=74 Win=29184 Len=0 TSval=544961 TSecr=544961
21	0.00501800	11.0.1.1	13.0.1.1	TCP	76	34188 > http [SYN] Seq=0 Win=29288 Len=0 MSS=1460 SACK_PERM=1 TSval=545229 TSecr=0 WS=512

**Q8:** Modify `website.sh` (call it `website2.sh`) by choosing one of the hosts in AS2 to send GET requests to the webserver running on h3-1. Post a screenshot of CLI output and wireshark log as the proof.

Ans:

```

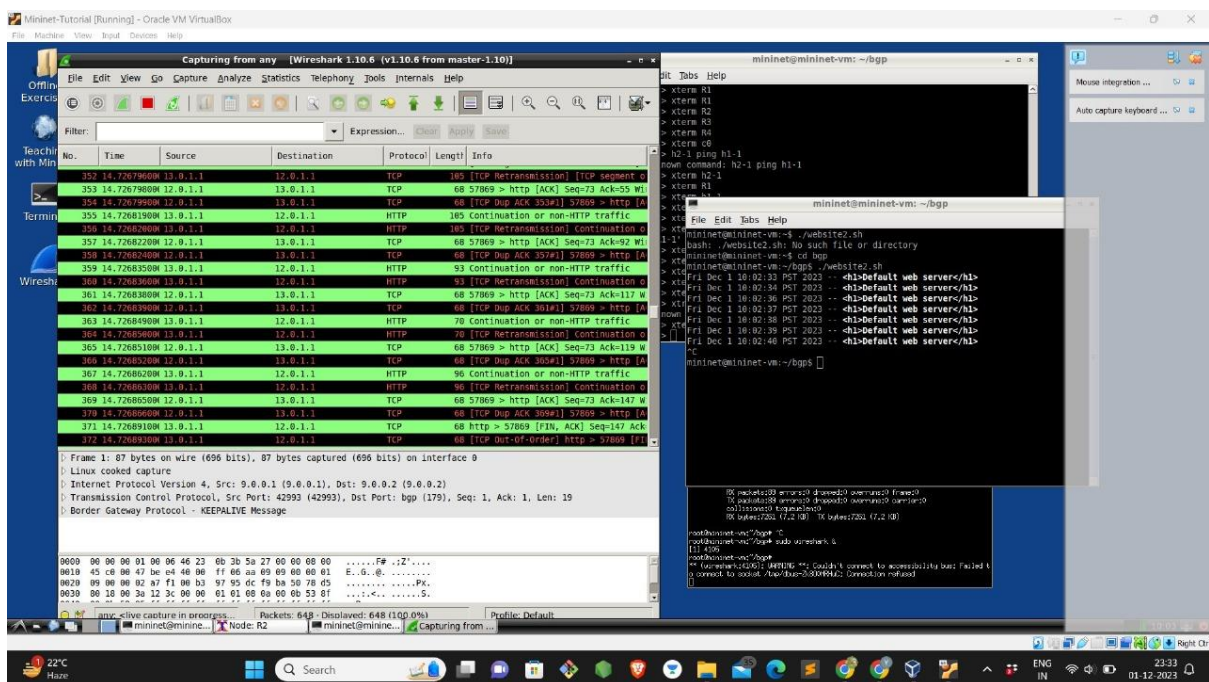
Node: h1-1

/bin/bash

node=$(h1-h2-1)
bold=$(tput bold)
normal=$(tput sgr0)

while true; do
    out=$(sudo python run.py --node $node --cmd "curl -s 13.0.1.1")
    date=$(date)
    echo $date -- $bold$out$normal
    sleep 1
done

root@mininet-vms:/bgp#
root@mininet-vms:/bgp#
  
```



**Q9:** Do you see any change in the CLI output where you ran `website.sh`? If yes, post the screenshot. If not, post the screenshot. What do you think has happened?

Ans:

```
mininet@mininet-vm: ~/bgp
File Edit Tabs Help
mininet@mininet-vm:~/bgp$ ./website.sh
Fri Dec 1 10:07:19 PST 2023 -- <h1>*** Attacker web server ***</h1>
Fri Dec 1 10:07:20 PST 2023 -- <h1>*** Attacker web server ***</h1>
Fri Dec 1 10:07:21 PST 2023 -- <h1>*** Attacker web server ***</h1>
Fri Dec 1 10:07:22 PST 2023 -- <h1>*** Attacker web server ***</h1>
Fri Dec 1 10:07:23 PST 2023 -- <h1>*** Attacker web server ***</h1>
Fri Dec 1 10:07:24 PST 2023 -- <h1>*** Attacker web server ***</h1>
Fri Dec 1 10:07:25 PST 2023 -- <h1>*** Attacker web server ***</h1>
^Z
[1]+  Stopped                  ./website.sh
mininet@mininet-vm:~/bgp$
```

After running, start\_rogue.sh, malicious AS4 will become the part of the network. So, it will advertise its existence to its autonomous system, as a result of which when AS1 receives the notification that it can reach AS3 in a less path, it will update its respective forwarding and BGP table. For reference the updated BGP table at R1 is shown below:

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 11.0.0.0	0.0.0.0	0		32768	i
*> 12.0.0.0	9.0.0.2	0		0	2 i
*> 13.0.0.0	9.0.4.2	0		0	4 i
*	9.0.0.2			0	2 3 i

**Q10:** Do you see any change in the CLI output where you ran the modified script, website2.sh? If yes, post the screenshot. If not, post the screenshot. What do you think has happened?

Ans: There would be no change in the forwarding table, but there would be change in the BGP table. As autonomous system 2 would be notified about the reachability of network 13.0.0 via AS1. But since AS\_path via that route is longer than the previous AS\_path so it won't change its forwarding table.

```
mininet@mininet-vm: ~/bgp
File Edit Tabs Help
mininet@mininet-vm:~/bgp$ ./website2.sh
Fri Dec 1 10:09:32 PST 2023 -- <h1>Default web server</h1>
Fri Dec 1 10:09:33 PST 2023 -- <h1>Default web server</h1>
Fri Dec 1 10:09:34 PST 2023 -- <h1>Default web server</h1>
Fri Dec 1 10:09:35 PST 2023 -- <h1>Default web server</h1>
Fri Dec 1 10:09:36 PST 2023 -- <h1>Default web server</h1>
Fri Dec 1 10:09:37 PST 2023 -- <h1>Default web server</h1>
Fri Dec 1 10:09:39 PST 2023 -- <h1>Default web server</h1>
^C
mininet@mininet-vm:~/bgp$
```

```
mininet@mininet-vm: ~/bgp
File Edit Tabs Help
Hello, this is Quagga (version 0.99.22.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:
bgpd-R2> en
Password:
bgpd-R2# sh ip bgp
BGP table version is 0, local router ID is 9.0.0.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network        Next Hop        Metric LocPrf Weight Path
*> 11.0.0.0        9.0.0.1          0         0 1 i
*> 12.0.0.0        0.0.0.0          0         0 32768 i
* 13.0.0.0        9.0.0.1          0         0 1 4 i
*>                 9.0.1.2          0         0 3 i

Total number of prefixes 3
bgpd-R2#
```

**Q11:** Log into the routers R1, R2. Are their BGP tables and forwarding tables different from before? If so, what is the difference? What has happened after bogus BGP advertisements by AS4 at AS1 and AS2?

Ans: Yes, BGP tables of both are different. Forwarding table of R1 differs while R2 will remain same. When bogus BGP advertisement happened by AS4 at AS1 and AS2. AS2 will not have a change in its forwarding table as for it to reach network 13.0.0.1 is just a one hop journey whereas the forwarding table of AS1 will have a change as it finds a shorter tour then previous to reach the same network tho it's faulty.

```
Node: R1
root@mininet-vm:~/bgp# route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
9.0.0.0 0.0.0.0 255.255.255.0 U 0 0 0 R1-eth4
9.0.4.0 0.0.0.0 255.255.255.0 U 0 0 0 R1-eth5
11.0.1.0 0.0.0.0 255.255.255.0 U 0 0 0 R1-eth1
11.0.2.0 0.0.0.0 255.255.255.0 U 0 0 0 R1-eth2
11.0.3.0 0.0.0.0 255.255.255.0 U 0 0 0 R1-eth3
12.0.0.0 9.0.0.2 255.0.0.0 UG 0 0 0 R1-eth4
13.0.0.0 9.0.4.2 255.0.0.0 UG 0 0 0 R1-eth5
root@mininet-vm:~/bgp#
```

```
Node: R2
root@mininet-vm:~/bgp# route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
9.0.0.0 0.0.0.0 255.255.255.0 U 0 0 0 R2-eth4
9.0.1.0 0.0.0.0 255.255.255.0 U 0 0 0 R2-eth5
11.0.0.0 9.0.0.1 255.0.0.0 UG 0 0 0 R2-eth4
12.0.1.0 0.0.0.0 255.255.255.0 U 0 0 0 R2-eth1
12.0.2.0 0.0.0.0 255.255.255.0 U 0 0 0 R2-eth2
12.0.3.0 0.0.0.0 255.255.255.0 U 0 0 0 R2-eth3
13.0.0.0 9.0.1.2 255.0.0.0 UG 0 0 0 R2-eth5
root@mininet-vm:~/bgp#
```



**Q12:** But this time open the xterm of the appropriate hosts and listen to the appropriate interfaces (figure out these interfaces) on wireshark in order to listen to the traffic. Now run the `start_rogue.sh` script. Do you see any BGP message sequence in the wireshark captures? Pin point which BGP message contains the rogue BGP update and post the screenshot. Expand the packet and post the screenshot. Explain the message contents, especially prefixes being advertised. Correlate this message with the screenshot taken earlier.

Ans:

## 1) Bogus AS4 advertising reachability of network 13.0.0.0 to AS1

The screenshot shows a Wireshark capture on interface eth0. The filter is set to `bgp`. The packet list shows a sequence of BGP messages. The packet details pane shows the structure of a BGP OPEN message (Frame 119):

- Marker: ffffffffffffffffffffffff
- Length: 33
- Type: OPEN Message (1)
- Version: 4
- My AS: 4
- Hold Time: 5
- BGP Identifier: 9.0.4.2 (9.0.4.2)
- Optional Parameters Length: 24
- Optional Parameters

The packet bytes pane shows the raw data of the BGP message.

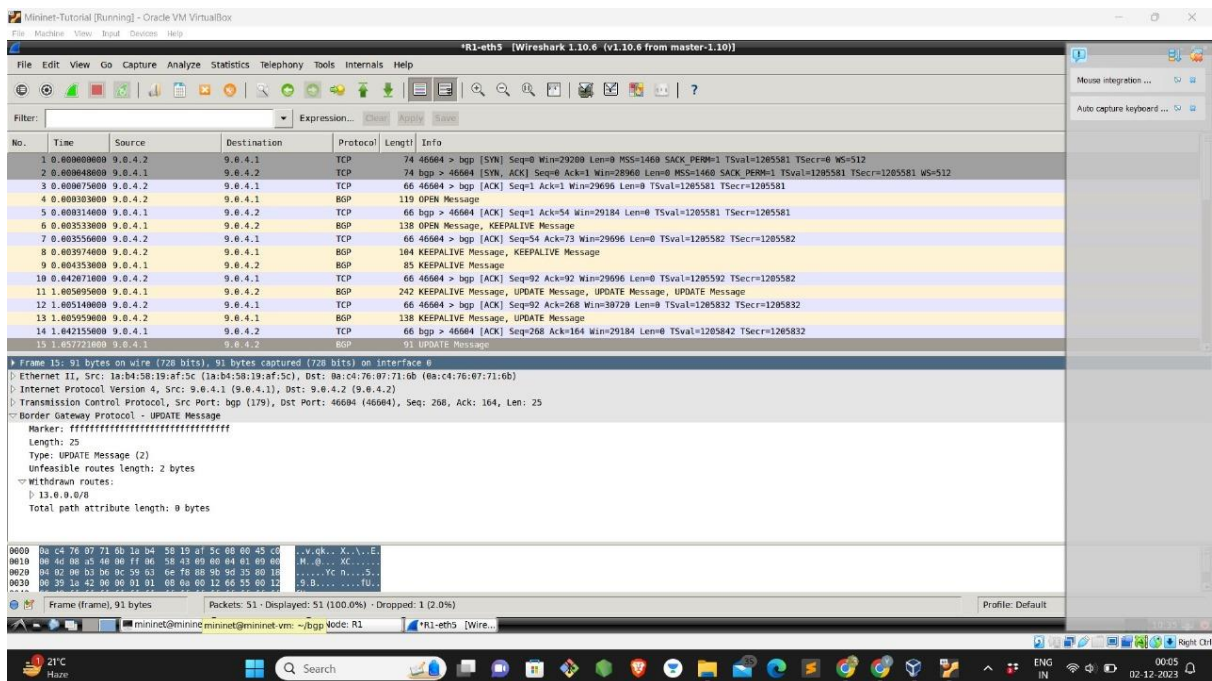
## 2) AS1 also advertise AS\_PATH to AS4

The screenshot shows a Wireshark capture on interface eth0. The filter is set to `bgp`. The packet list shows a sequence of BGP messages. The packet details pane shows the structure of a BGP OPEN message (Frame 6):

- Marker: ffffffffffffffffffffffff
- Length: 33
- Type: OPEN Message (1)
- Version: 4
- My AS: 1
- Hold Time: 5
- BGP Identifier: 9.0.0.1 (9.0.0.1)
- Optional Parameters Length: 24
- Optional Parameters

The packet bytes pane shows the raw data of the BGP message.

### 3) AS1 finds shortest path and updates its BGP table to reach 13.0.0.0



**Q13:** Now put the sequence of events together and explain in clear steps what has occurred from start to finish. Is rogue AS succeeded in fooling the hosts (and then directing them to a fake website running at the hijacked host/web server) present in all other ASs or only a subset of them? List out the hosts that got fooled by the rogue AS.

**Ans:** When AS4 becomes the part of network, it will have impacts on the BGP tables of entire network router but only the forwarding table of R1 will have a change because it thinks AS4 as AS3 and finds it one hop away. As a result of which all the hosts of AS1 which are willing to communicate AS3 will now be redirected to bogus AS4. But AS2's and AS3's forwarding table will not have any changes as they already know shortest path to reach AS3 and so only a part of the network(AS1) will be poisoned because of inclusion of bogus AS4.

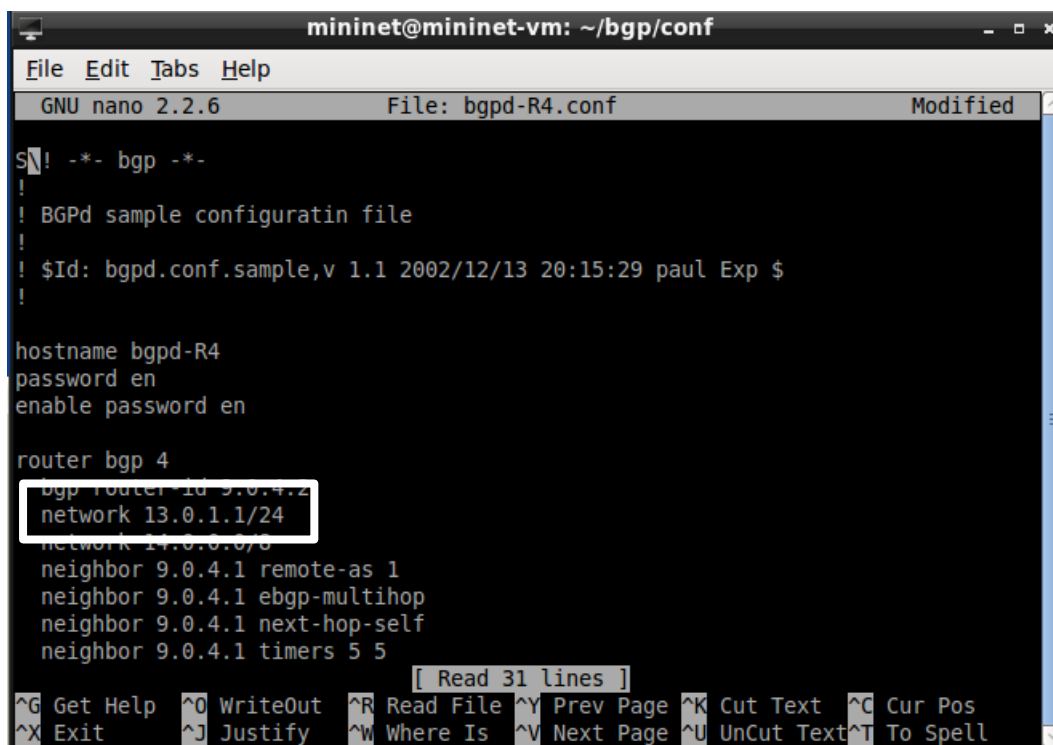
**Q14.** When hosts present in AS1 ping hosts in AS3, observe RTT before running `start_rogue.sh` script and after running `start_rogue.sh` script. Do you find any difference, explain. Did the rogue AS (AS4) hijack all the hosts in AS3 or a subset of them?

**Ans:** It was clearly seen that the RTT before inclusion of bogus RS4 is more than after its inclusion. This is because of the simple fact that before for AS1 to reach AS3 it was two hop away and when bogus AS4 that mimics AS3 comes into the picture then for AS1 to reach AS4 or say AS3 is just one hop tour.

**Q15.** This is an advanced task. Modify the scripts given in the code base in a way the rogue attacker (AS4) only hijacks the host "h3-1" but not other hosts present in AS3. Explain how to launch this targeted BGP path hijack attack on the target host "h3-1" and demonstrate it with step-by-step instructions with screenshots

**Ans:**

## Changes to bgpd-R4.conf



The screenshot shows a terminal window titled "mininet@mininet-vm: ~/bgp/conf". The window displays the configuration file "bgpd-R4.conf" using the GNU nano 2.2.6 editor. The configuration includes a sample header, hostname "bgpd-R4", password "en", and BGP settings for router "4". The "network 13.0.1.1/24" line is highlighted with a white box. A status bar at the bottom indicates "[ Read 31 lines ]" and provides keyboard shortcuts for various editor functions.

```
mininet@mininet-vm: ~/bgp/conf
File Edit Tabs Help
GNU nano 2.2.6 File: bgpd-R4.conf Modified

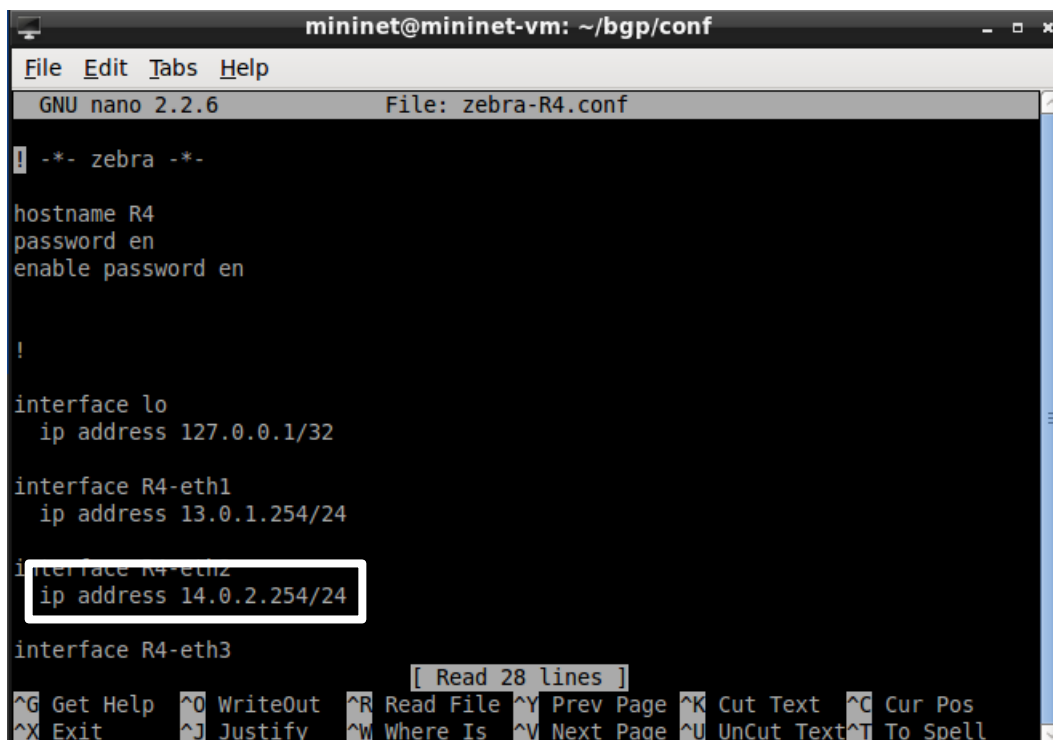
S! ! *- bgp *-
! BGPd sample configuratin file
! $Id: bgpd.conf.sample,v 1.1 2002/12/13 20:15:29 paul Exp $
!

hostname bgpd-R4
password en
enable password en

router bgp 4
  bgp router-id 9.0.4.2
  network 13.0.1.1/24
  network 14.0.0.0/8
  neighbor 9.0.4.1 remote-as 1
  neighbor 9.0.4.1 ebgp-multihop
  neighbor 9.0.4.1 next-hop-self
  neighbor 9.0.4.1 timers 5 5

[ Read 31 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

## Changes to zebre-R4.conf



The screenshot shows a terminal window titled "mininet@mininet-vm: ~/bgp/conf". The window displays the configuration file "zebra-R4.conf" using the GNU nano 2.2.6 editor. The configuration includes a sample header, hostname "R4", password "en", and interface settings for "lo", "R4-eth1", "R4-eth2", and "R4-eth3". The "ip address 14.0.2.254/24" line for "R4-eth2" is highlighted with a white box. A status bar at the bottom indicates "[ Read 28 lines ]" and provides keyboard shortcuts for various editor functions.

```
mininet@mininet-vm: ~/bgp/conf
File Edit Tabs Help
GNU nano 2.2.6 File: zebra-R4.conf

! *- zebra *-

hostname R4
password en
enable password en

!

interface lo
  ip address 127.0.0.1/32

interface R4-eth1
  ip address 13.0.1.254/24

interface R4-eth2
  ip address 14.0.2.254/24

interface R4-eth3

[ Read 28 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

## Changes to R4's interface IP's

```

File Machine View Input Devices Help
X
root@mininet-vml:/bpg# ifconfig
R4-eth1  Link encap:Ethernet  HWaddr d6:a2:7c:24:fd:47
         inet addr:13.0.1.254  Bcast:13.0.1.255  Mask:255.255.255.0
         inet6 addr: fe80::d4a2:7cff:fe24:fd47/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:8 errors:0 dropped:0 overruns:0 frame:0
         TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:648 (648.0 B)  TX bytes:648 (648.0 B)

R4-eth2  Link encap:Ethernet  HWaddr fa:8f:13:0b:de:35
         inet addr:14.0.2.254  Bcast:14.0.2.255  Mask:255.255.255.0
         inet6 addr: fe80::f88f:13ff:fe0b:de35/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:8 errors:0 dropped:0 overruns:0 frame:0
         TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:648 (648.0 B)  TX bytes:648 (648.0 B)

R4-eth3  Link encap:Ethernet  HWaddr 4e:9a:88:e2:53:03
         inet addr:14.0.3.254  Bcast:14.0.3.255  Mask:255.255.255.0
         inet6 addr: fe80::4c9a:88ff:fee2:5303/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:8 errors:0 dropped:0 overruns:0 frame:0
         TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:648 (648.0 B)  TX bytes:648 (648.0 B)

R4-eth4  Link encap:Ethernet  HWaddr d6:92:e9:a9:77:34
         inet addr:9.0.4.2  Bcast:9.0.4.255  Mask:255.255.255.0
         inet6 addr: fe80::d492:e9ff:fea9:7734/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:95 errors:0 dropped:0 overruns:0 frame:0
         TX packets:84 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:7276 (7.2 KB)  TX bytes:6594 (6.5 KB)

lo       Link encap:Local Loopback
         inet addr:127.0.0.1  Mask:255.0.0.0
         inet6 addr: ::1/128 Scope:Host
         UP LOOPBACK RUNNING  MTU:65536  Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@mininet-vml:/bpg#

```

## Change to the R1 BGP and Forwarding Tables

```

Network      Next Hop      Metric LocPrf Weight Path
*> 11.0.0.0   0.0.0.0        0         32768 i
*> 12.0.0.0   9.0.0.2        0         0 2 i
*> 13.0.0.0   9.0.0.2        0         0 2 3 i
*> 13.0.1.0/24 9.0.4.2        0         0 4 i
*> 14.0.0.0   9.0.4.2        0         0 4 i

Total number of prefixes 5

```

```

root@mininet-vml:/bpg# route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
0.0.0.0 0.0.0.0 255.255.255.0 U 0 0 0 R1-eth4
9.0.4.0 0.0.0.0 255.255.255.0 U 0 0 0 R1-eth5
11.0.1.0 0.0.0.0 255.255.255.0 U 0 0 0 R1-eth1
11.0.2.0 0.0.0.0 255.255.255.0 U 0 0 0 R1-eth2
11.0.3.0 0.0.0.0 255.255.255.0 U 0 0 0 R1-eth3
12.0.0.0 9.0.0.2 255.0.0.0 UG 0 0 0 R1-eth4
13.0.0.0 9.0.0.2 255.0.0.0 UG 0 0 0 R1-eth4
13.0.1.0 9.0.4.2 255.255.255.0 UG 0 0 0 R1-eth5
14.0.0.0 9.0.4.2 255.0.0.0 UG 0 0 0 R1-eth5

root@mininet-vml:/bpg#

```



### Output comparison:

While trying to hit 13.0.1.1 from host H1-1

```
mininet@mininet-vm:~/bgp$ ./website.sh
Sat Dec 2 02:16:57 PST 2023 -- <h1>*** Attacker web server ***</h1>
Sat Dec 2 02:16:58 PST 2023 -- <h1>*** Attacker web server ***</h1>
Sat Dec 2 02:16:59 PST 2023 -- <h1>*** Attacker web server ***</h1>
^C
```

While trying to hit 13.0.2.1 from host H1-1

```
mininet@mininet-vm:~/bgp$ ^C
mininet@mininet-vm:~/bgp$ ./website3.sh
Sat Dec 2 02:17:18 PST 2023 -- <h1>Hey!! from host h3-2</h1>
Sat Dec 2 02:17:19 PST 2023 -- <h1>Hey!! from host h3-2</h1>
^C
mininet@mininet-vm:~/bgp$ ^C
```

# ANTI-PLAGIARISM Statement

*We certify that this assignment/report is our own work, based on our personal study and/or research and that we have acknowledged all material and sources used in its preparation, whether they be books, articles, packages, datasets, reports, lecture notes, and any other kind of document, electronic or personal communication. We also certify that this assignment/report has not previously been submitted for assessment/project in any other course lab, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that we have not copied in part or whole or otherwise plagiarized the work of other students and/or persons. Additionally, we acknowledge that we may have used AI tools, such as language models (e.g., ChatGPT, Bard), for assistance in generating and refining my assignment, and we have made all reasonable efforts to ensure that such usage complies with the academic integrity policies set for the course. I pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, we understand our responsibility to report honour violations by other students if we become aware of it.*

Name: Yug Patel, CS23MTECH14019  
Somya jain, CS23MTECH12011  
Anil Kumar, CS23MTECH13001

Date:02/12/2023

Signature: Yug Patel, CS23MTECH14019  
Somya jain, CS23MTECH12011  
Anil Kumar, CS23MTECH13001