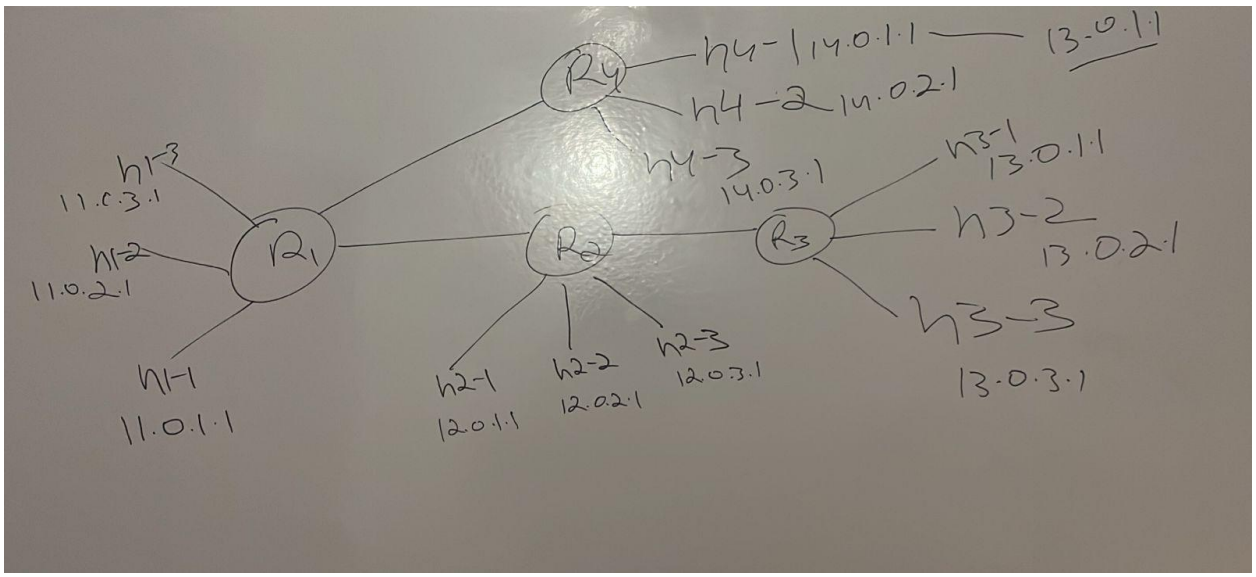
Mininet Assignment Report

Sanyam Kaul - CS23MTECH14011
Bhargav Patel - CS23MTECH11026
Arnab Ghosh - CS23MTECH11025

Ans 1.



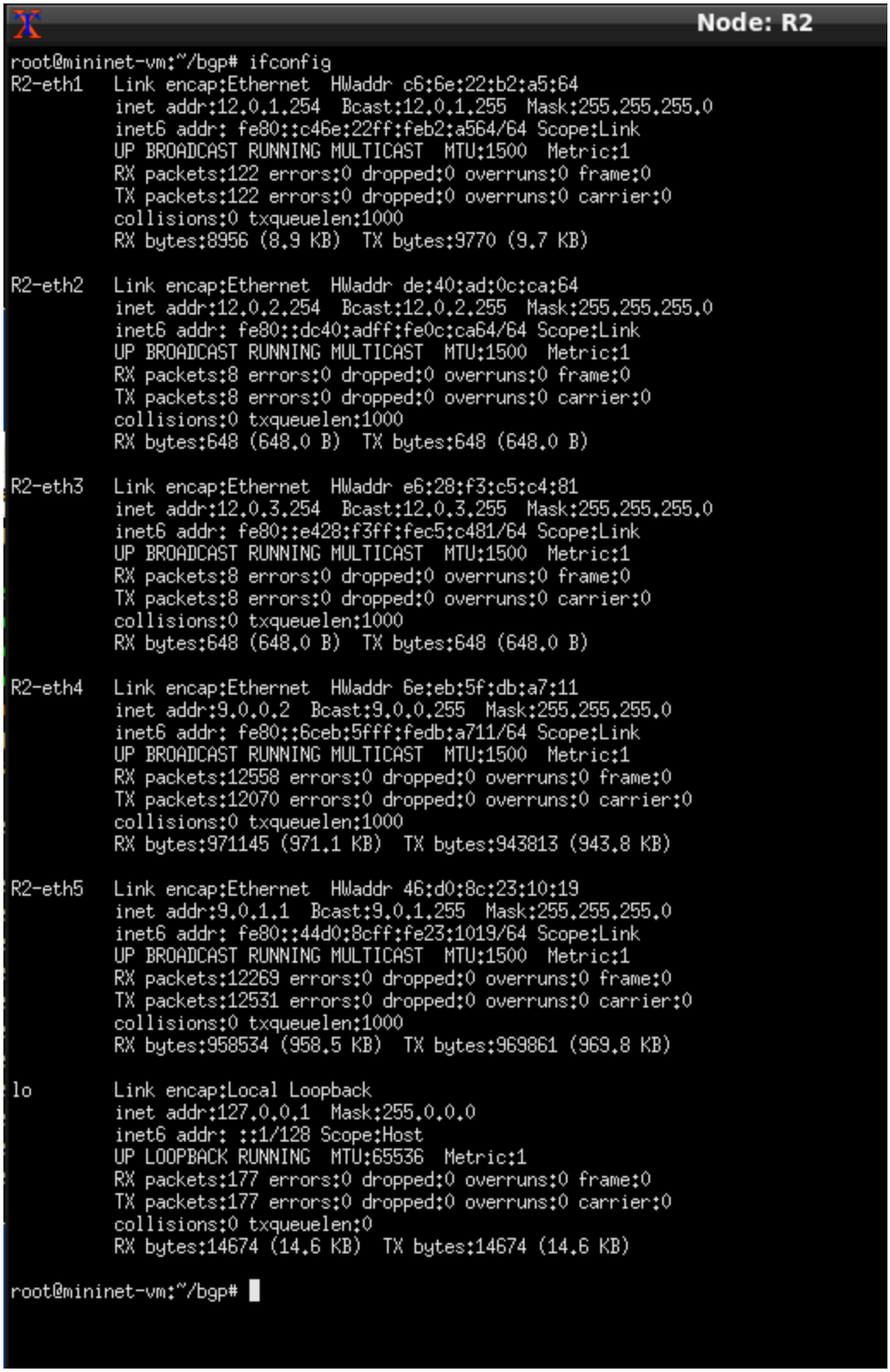
4 routers represent 4 subnets - R1, R2, R3 and R4
Each subnet has 3 hosts as represented in the diagram above

Ans 2.

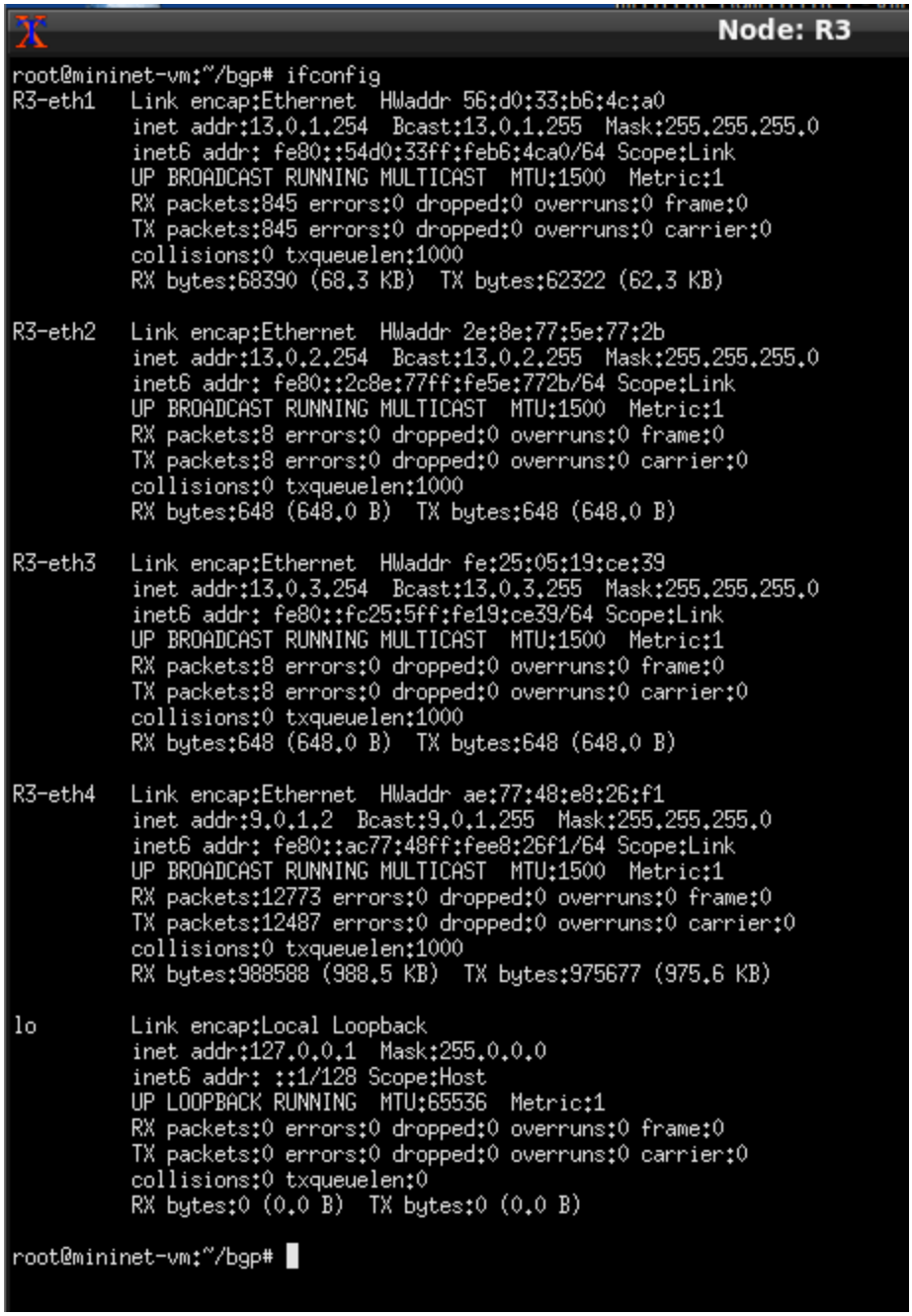
Snapshots of Interfaces with IP address info: -

R1 -

R2 -


Node: R2
root@mininet-vm:~/bgp# ifconfig
R2-eth1 Link encap:Ethernet HWaddr c6:6e:22:b2:a5:64
inet addr:12.0.1.254 Bcast:12.0.1.255 Mask:255.255.255.0
inet6 addr: fe80::c46e:22ff:feb2:a564/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:122 errors:0 dropped:0 overruns:0 frame:0
TX packets:122 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:8956 (8.9 KB) TX bytes:9770 (9.7 KB)
R2-eth2 Link encap:Ethernet HWaddr de:40:ad:0c:ca:64
inet addr:12.0.2.254 Bcast:12.0.2.255 Mask:255.255.255.0
inet6 addr: fe80::dc40:adff:fe0c:ca64/64 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)
R2-eth3 Link encap:Ethernet HWaddr e6:28:f3:c5:c4:81
inet addr:12.0.3.254 Bcast:12.0.3.255 Mask:255.255.255.0
inet6 addr: fe80::e428:f3ff:fec5:c481/64 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)
R2-eth4 Link encap:Ethernet HWaddr 6e:eb:5f:db:a7:11
inet addr:9.0.0.2 Bcast:9.0.0.255 Mask:255.255.255.0
inet6 addr: fe80::6ceb:5fff:fedb:a711/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:12558 errors:0 dropped:0 overruns:0 frame:0
TX packets:12070 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:971145 (971.1 KB) TX bytes:943813 (943.8 KB)
R2-eth5 Link encap:Ethernet HWaddr 46:d0:8c:23:10:19
inet addr:9.0.1.1 Bcast:9.0.1.255 Mask:255.255.255.0
inet6 addr: fe80::44d0:8cff:fe23:1019/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:12269 errors:0 dropped:0 overruns:0 frame:0
TX packets:12531 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:958534 (958.5 KB) TX bytes:969861 (969.8 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:177 errors:0 dropped:0 overruns:0 frame:0
TX packets:177 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:14674 (14.6 KB) TX bytes:14674 (14.6 KB)
root@mininet-vm:~/bgp#

R3 -


Node: R3
root@mininet-vm:~/bgp# ifconfig
R3-eth1 Link encap:Ethernet HWaddr 56:d0:33:b6:4c:a0
inet addr:13.0.1.254 Bcast:13.0.1.255 Mask:255.255.255.0
inet6 addr: fe80::54d0:33ff:feb6:4ca0/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:845 errors:0 dropped:0 overruns:0 frame:0
TX packets:845 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:68390 (68.3 KB) TX bytes:62322 (62.3 KB)
R3-eth2 Link encap:Ethernet HWaddr 2e:8e:77:5e:77:2b
inet addr:13.0.2.254 Bcast:13.0.2.255 Mask:255.255.255.0
inet6 addr: fe80::2c8e:77ff:fe5e:772b/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)
R3-eth3 Link encap:Ethernet HWaddr fe:25:05:19:ce:39
inet addr:13.0.3.254 Bcast:13.0.3.255 Mask:255.255.255.0
inet6 addr: fe80::fc25:5ff:fe19:ce39/64 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)
R3-eth4 Link encap:Ethernet HWaddr ae:77:48:e8:26:f1
inet addr:9.0.1.2 Bcast:9.0.1.255 Mask:255.255.255.0
inet6 addr: fe80::ac77:48ff:fee8:26f1/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:12773 errors:0 dropped:0 overruns:0 frame:0
TX packets:12487 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:988588 (988.5 KB) TX bytes:975677 (975.6 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-vm:~/bgp#

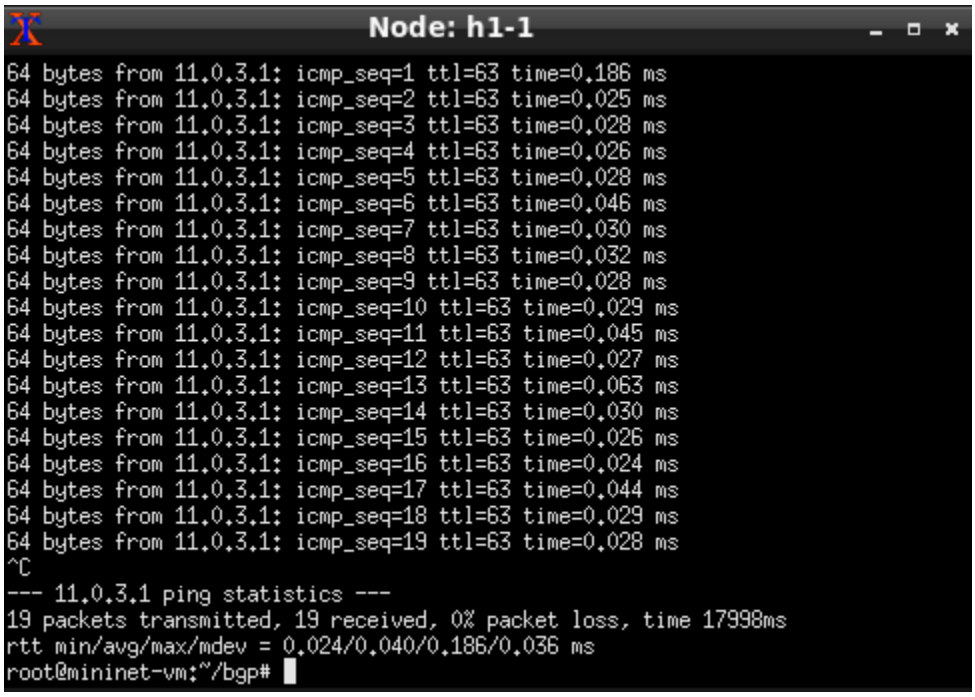
R4 -

Hosts' IP addresses are mentioned in the diagram in the previous answer

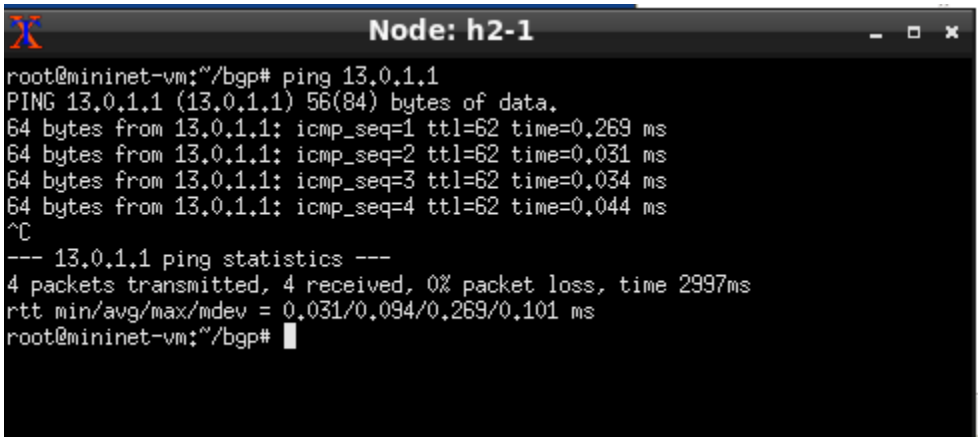
Hosts of AS4 do not have any IP address at this time. They are assigned IP addresses when we run the .'start_rogue.py' script

Ans 3.

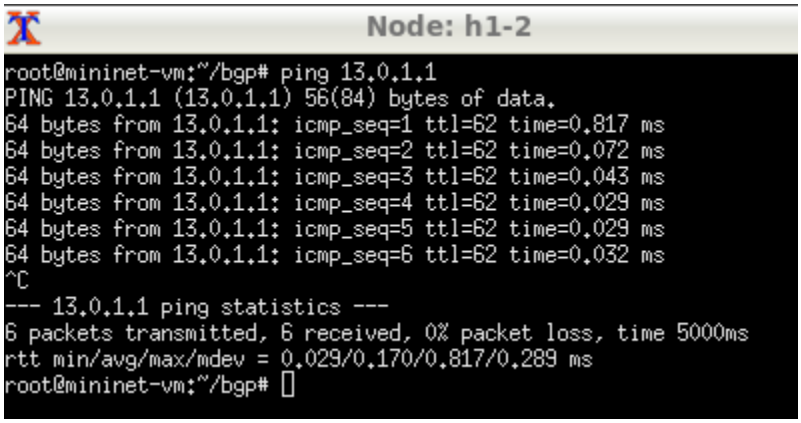
h3-1 is reachable from h1-1



H3-1 reachable from h2-1



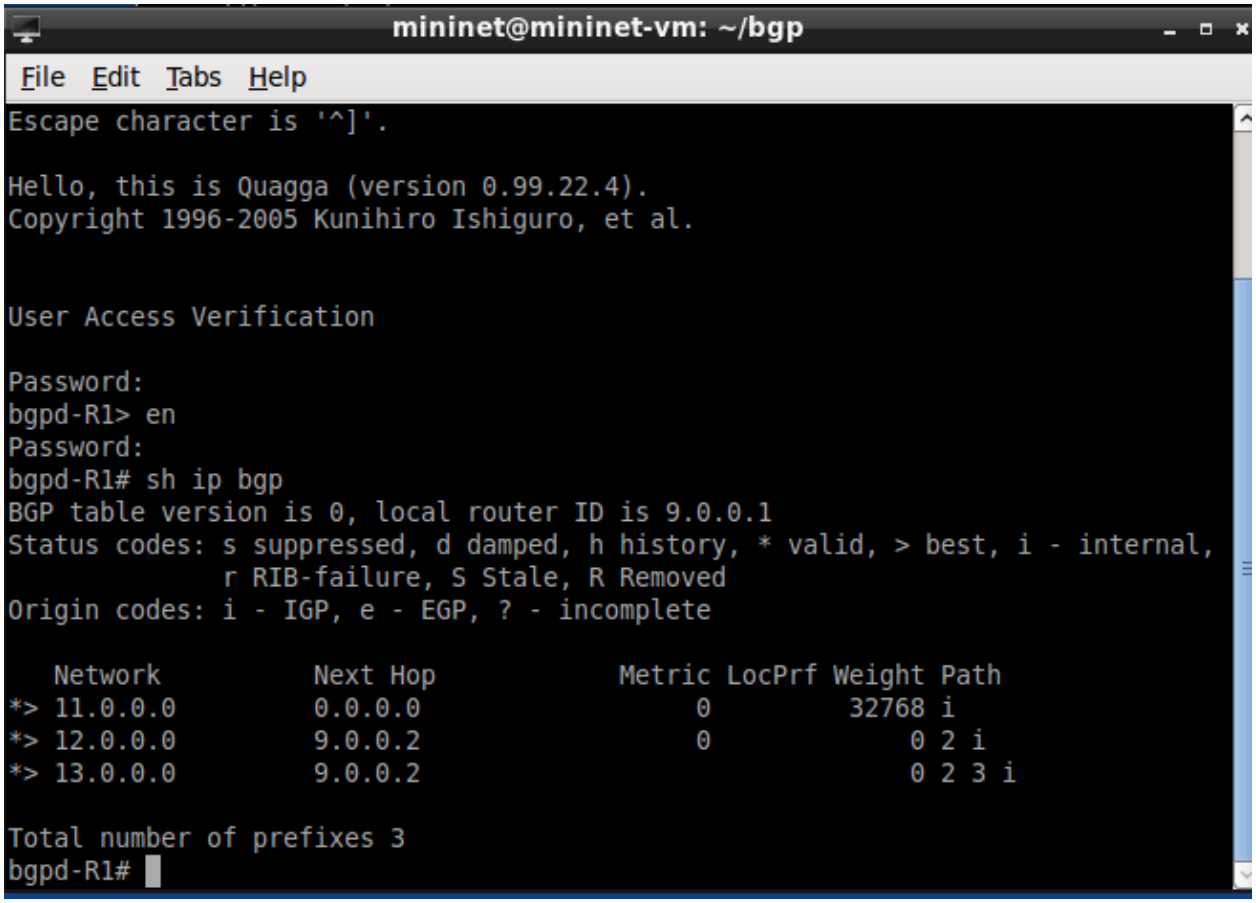
H3-1 reachable from h1-2



Ans4

The BGP tables provide information about the learned routes to different hosts in different ASes

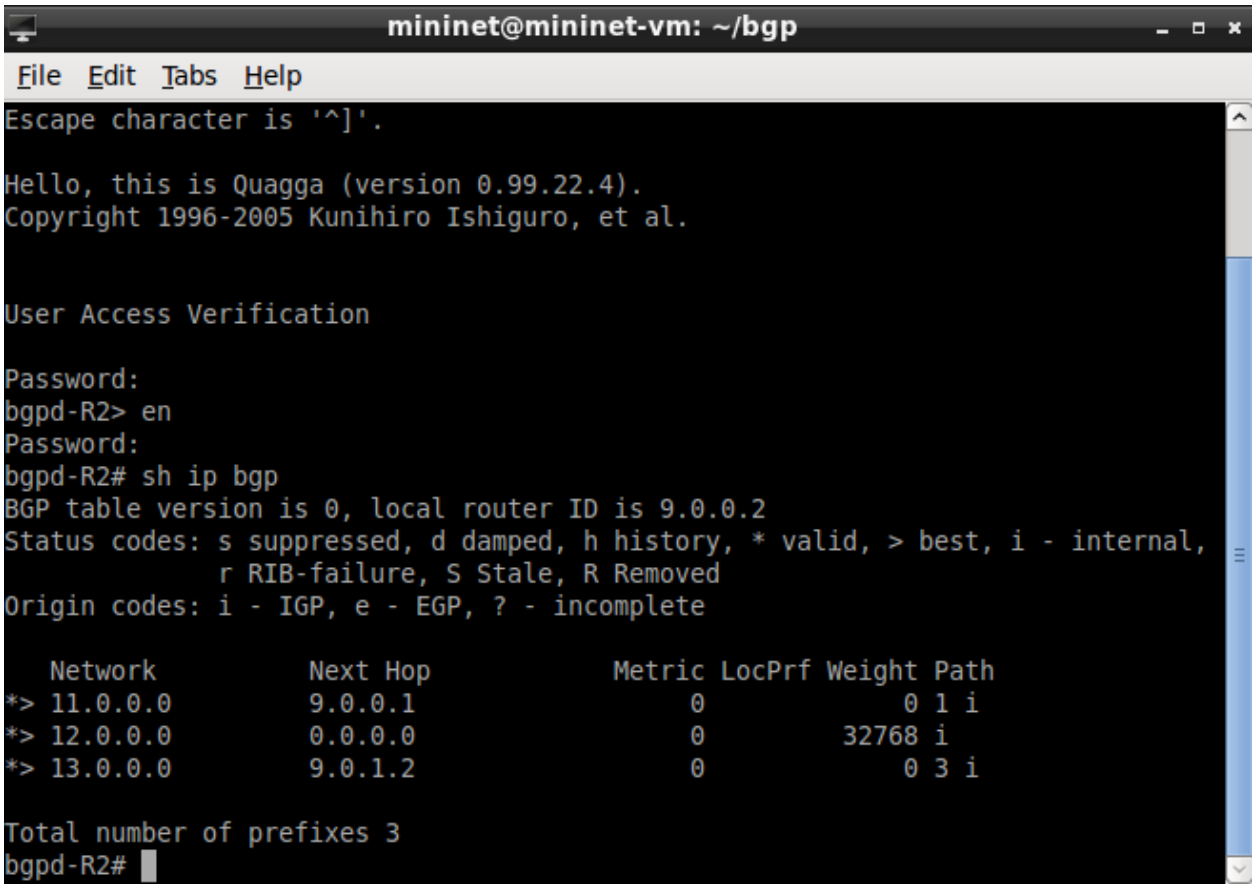
R1 BGP table



A brief explanation of the different fields of the table:-

Network - This is the network address of the destination network in CIDR notation
Next Hop - This field provides the IP address of the next hop router to reach the destination network
Metric - This indicates the cost of the path (a lower cost is preferred)
LocPrf - Stands for Local Preference. It is a value used within the AS to influence the outbound routing decisions
Weight - A CISCO-specific parameter indicating the local preference for a route
Path - It shows the route's path through different ASes. The 'i' value means that the route was learned from an interior gateway protocol. For example, as per the table if a packet has to be delivered to a host in AS3 then first R1 will deliver it to R2 (Ip - 9.0.0.2), then R@ will deliver it to R3 and finally the internal route from R3 to host will be learned at R3 only. That's why the weighted path says '2 3 i'

Ans 5

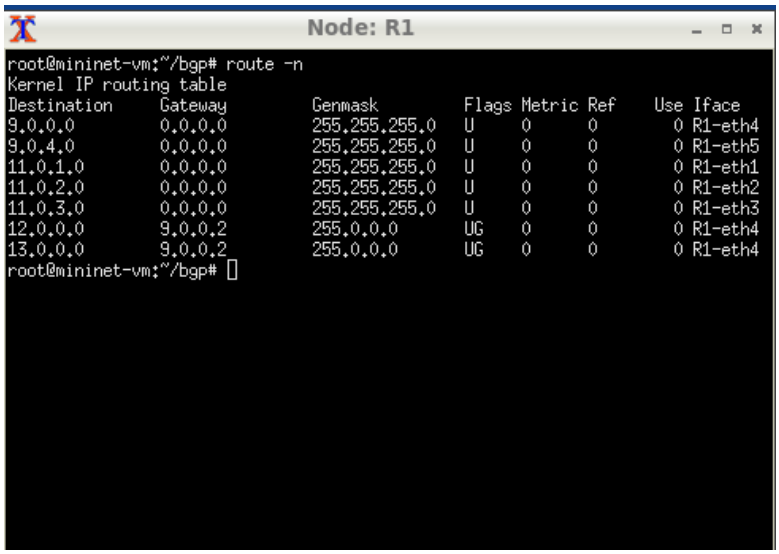


The next hop, weight and Path values are different for Network CIDRs as compared to R1
Since 12.0.0.0 is the local AS of R2 that's why its Next Hop is 0.0.0.0 and Path is 'i'.
In the case of R1 11.0.0.0 is the local AS, so its next hop is 0.0.0.0 and Path is 'i'.

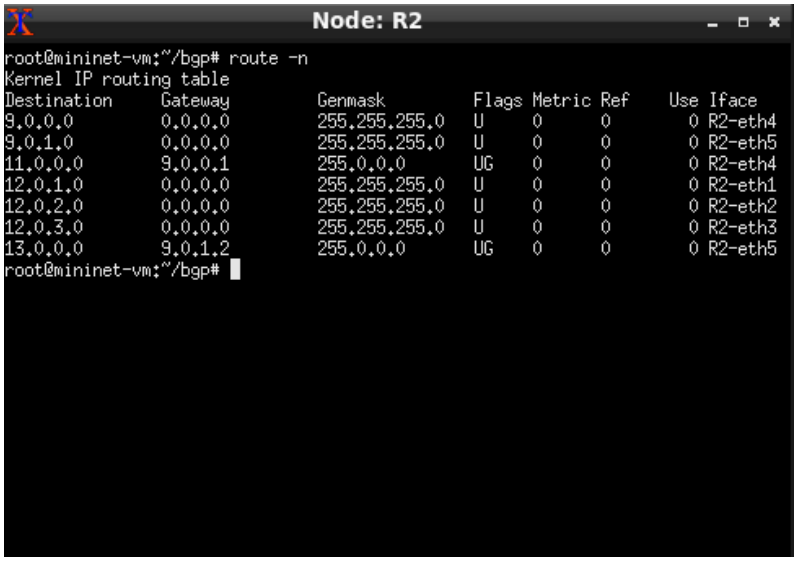
Also, note that the Path to go to 11.0.0.0 is ‘1 i’ which means that the packet will be delivered to R1 in the next hop (Next hop - 9.0.0.1) and the Path to go to 13.0.0.0 is ‘3 i’ which means that the packet will be delivered to R3 in next hop (9.0.1.2)

Ans 6

R1 forwarding table



R2 forwarding table

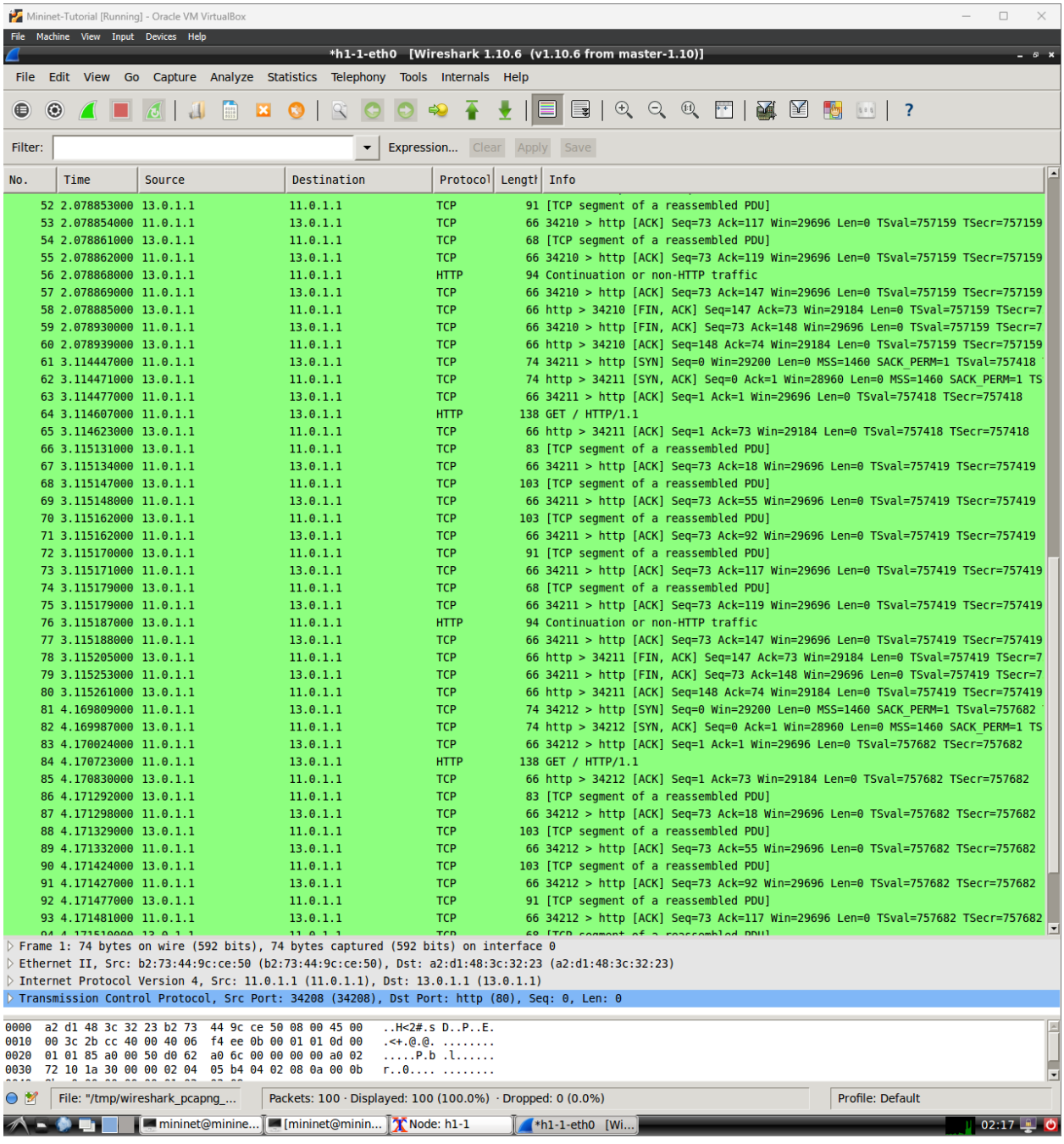


The forwarding table is used locally the the router. We can see the routing table of R1 and infer that it is giving details about which interface the packet will be forwarded to if it is supposed to

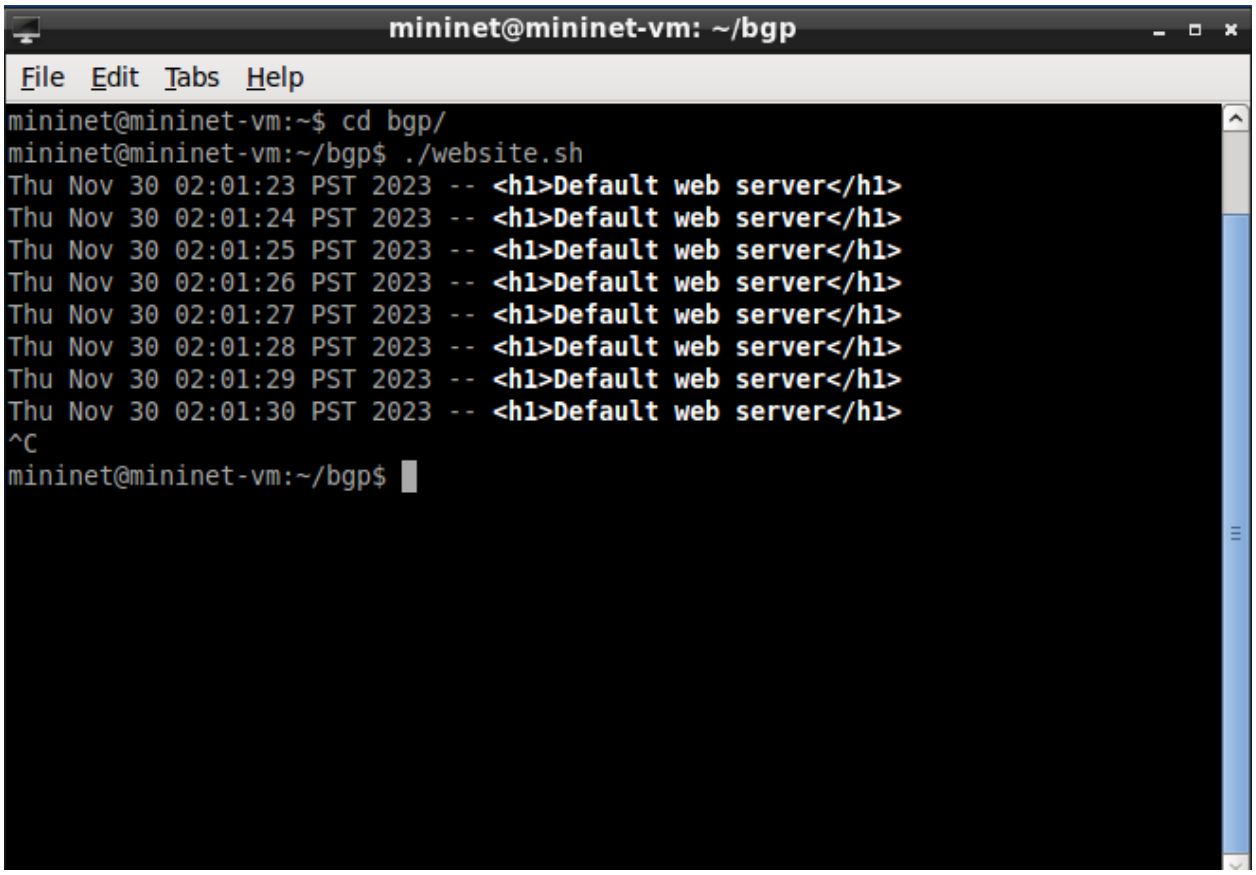
be delivered to a particular IP address. The IP address can be of a host connected with R1 or is present in a different AS.
The interface information about the hosts connected to R1 is filled in the Forwarding table by Intra-Routing protocol which could be OSPF, IS-IS, RIP etc. The interface information about the network addresses of other ASes is filled in the Forwarding table by BGP

Ans 7

Wireshark snapshot depicting TCP and HTTP packets flowing in the network when h1-1 is sending GET requests to the server running on host h3-1: -

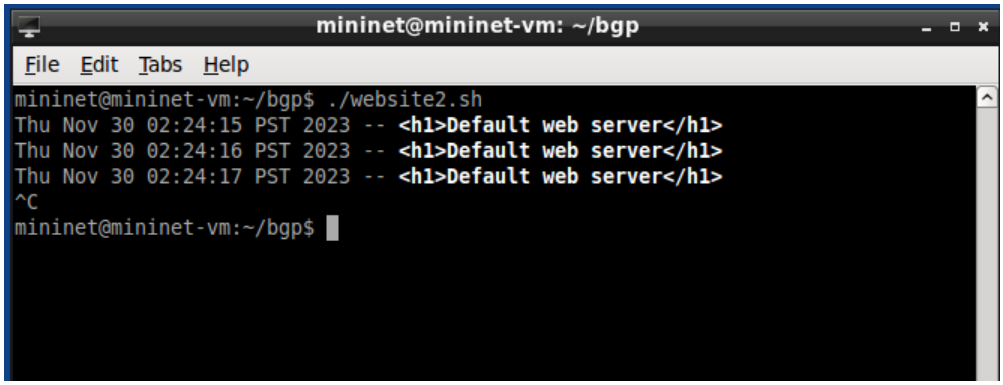


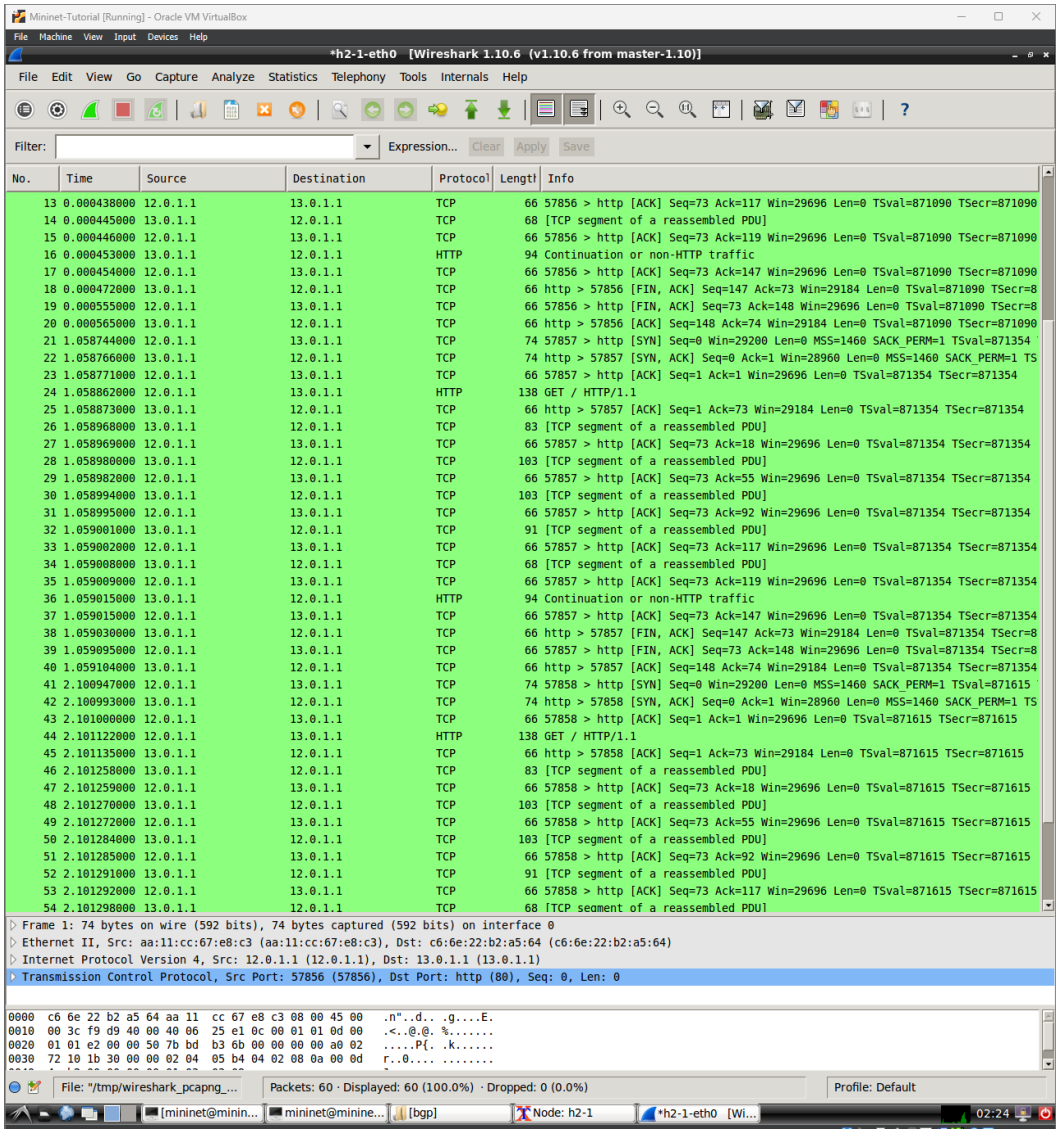
Terminal snapshot: -



Ans 8

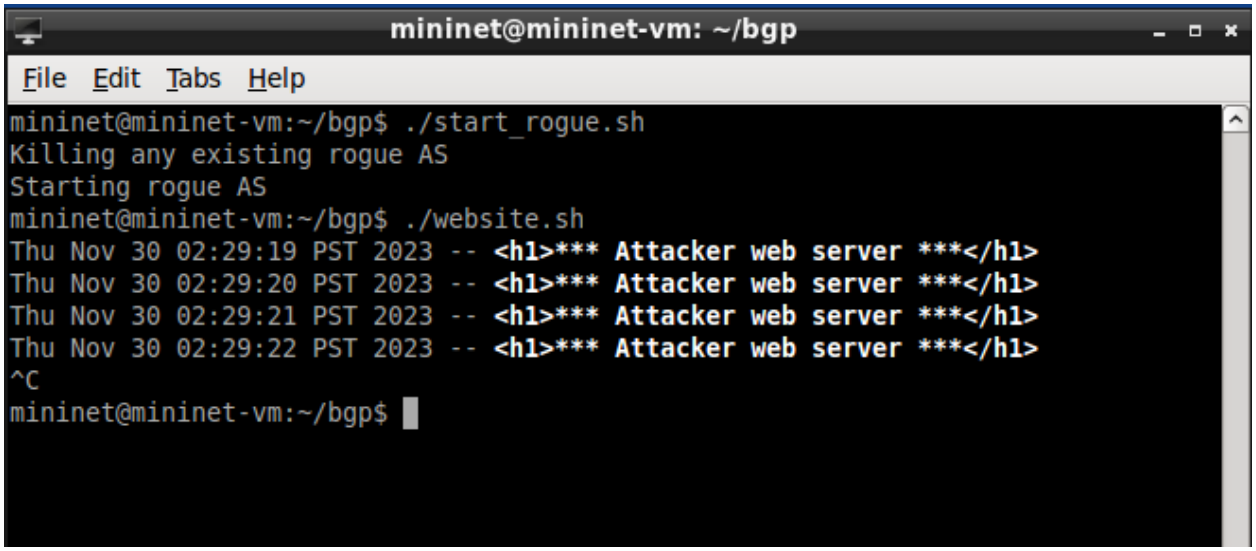
GET request being sent from h2-1 to h3-1: -





Ans 9

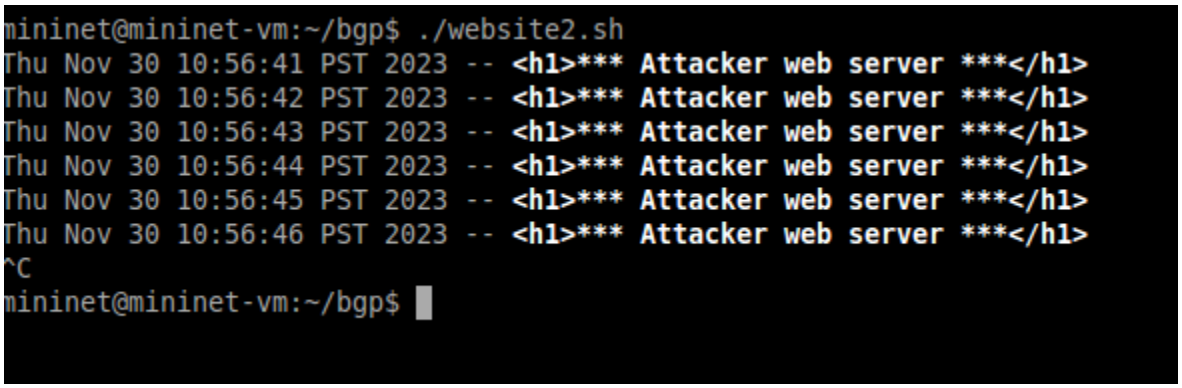
GET requests going to attacker host (h4-1) instead of original hos when I am running the website.sh: -



The Attacker has been able to fool the network into believing that the original web server is running in his host but it's a fake server

Ans 10

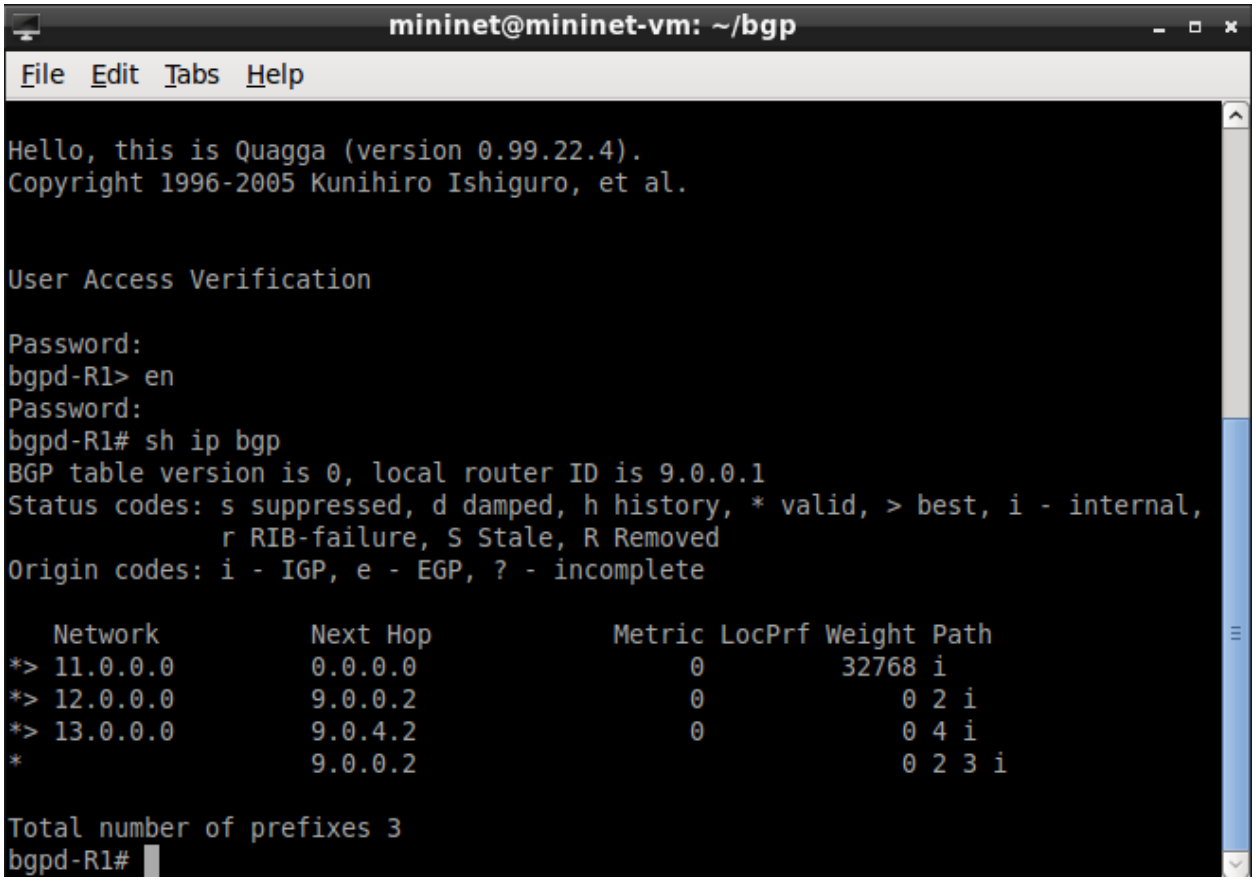
Changed host to h1-2 in website2.sh and the GET requests are still going to the attacker web server: -



When I changed the host to h3-1 and used it, the requests were going to the correct server. Looks, like all the hosts in AS1 have been fooled into believing that the path to the webserver has been updated.

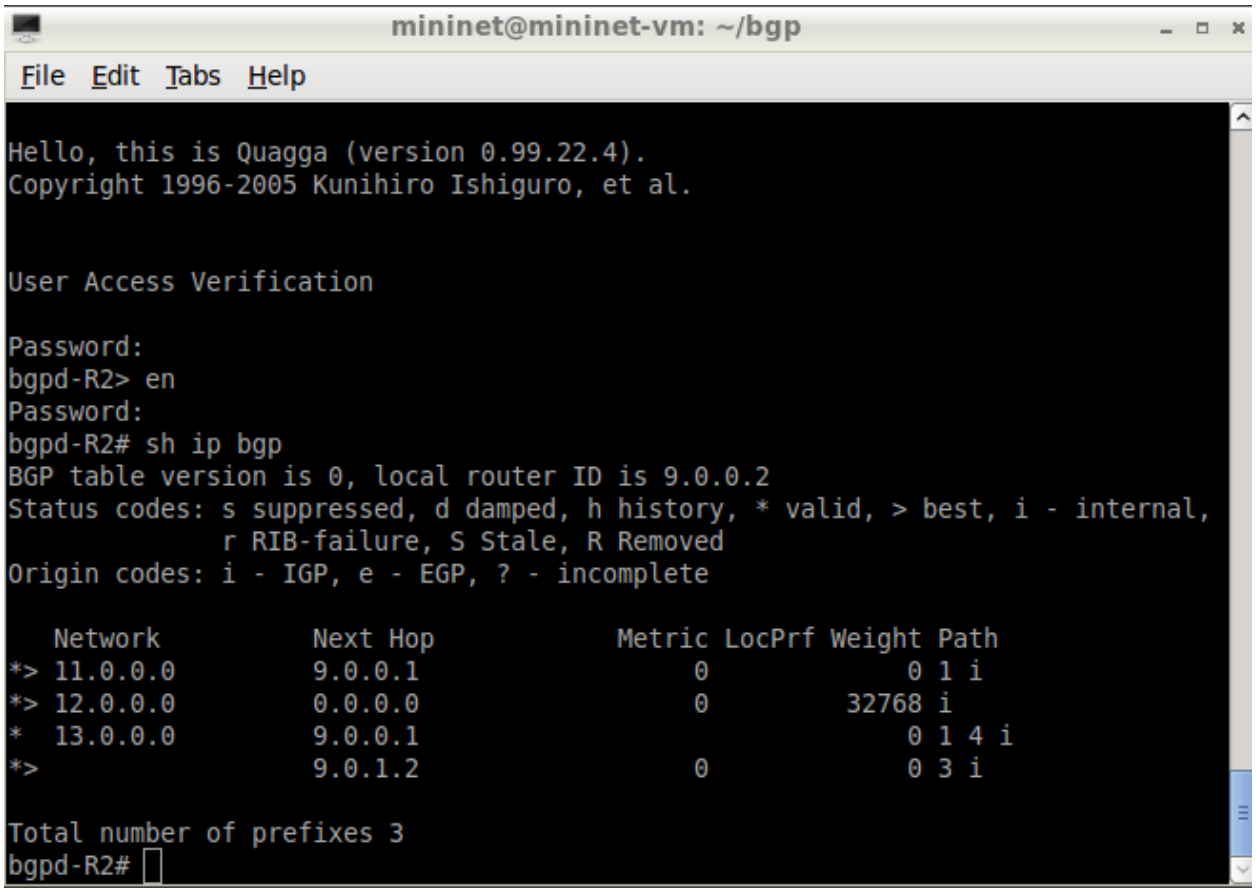
Ans 11

BGP table to R1 after running start_rogue.sh : -



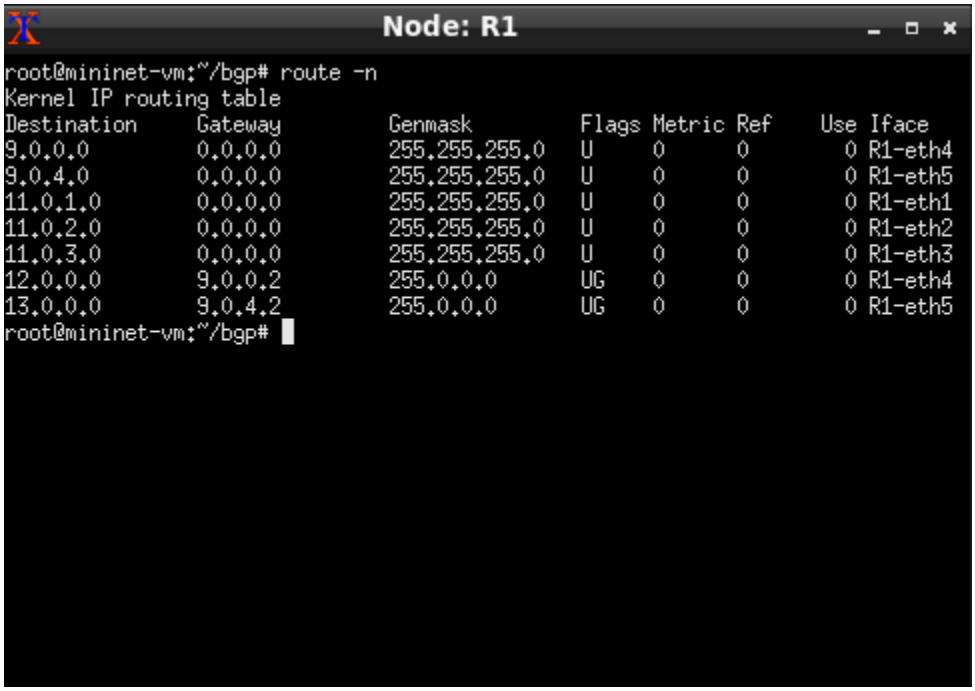
We can see in the table that now there are 2 paths to reach 13.0.0.0 which is AS3. One is the old path - ‘2 3 i’ for which the next hop is R2. The other is the new path - ‘4 i’ for which the next hop is R4. Since the latter is the shortest path, all hosts of R1 will be using this new path to reach h3-1

BGP table to R2 after running start_rogue.sh : -



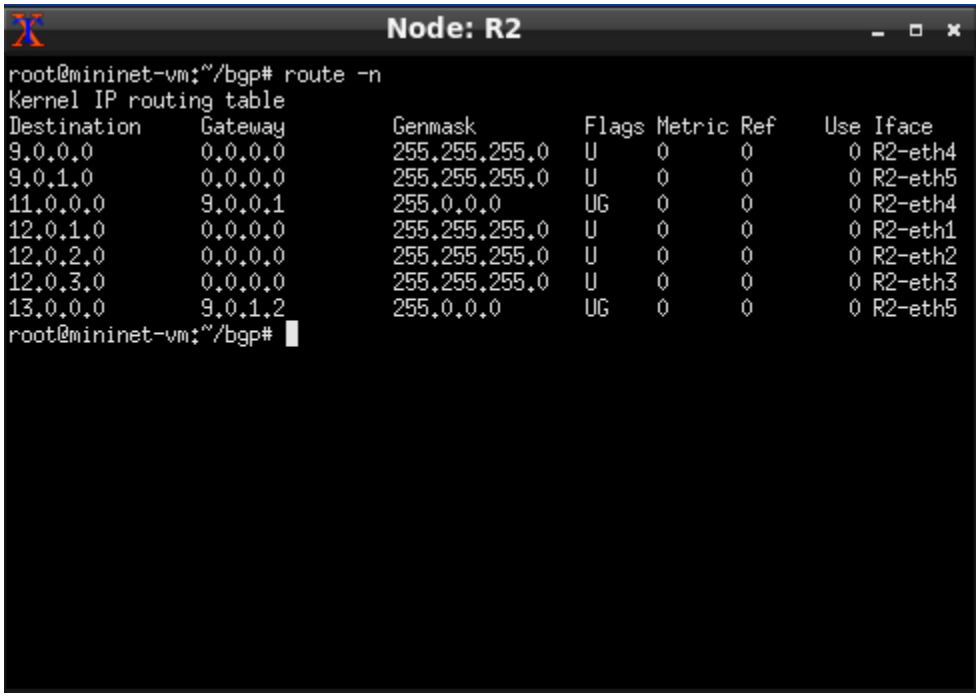
We can see here that for R2 path - ‘3 i’ is still the shortest path to reach h3-1. This is the reason why requests from hosts of R2 are going to the correct server

Forwarding table to R1 after running start_rogue.sh : -



Note that the Gateway for 13.0.0.0 is not set to 9.0.4.2. This is the gateway which connects R1 to R4. Earlier the gateway being used for 12.0.0.0 was 9.0.0.2 which connects R1 to R2.

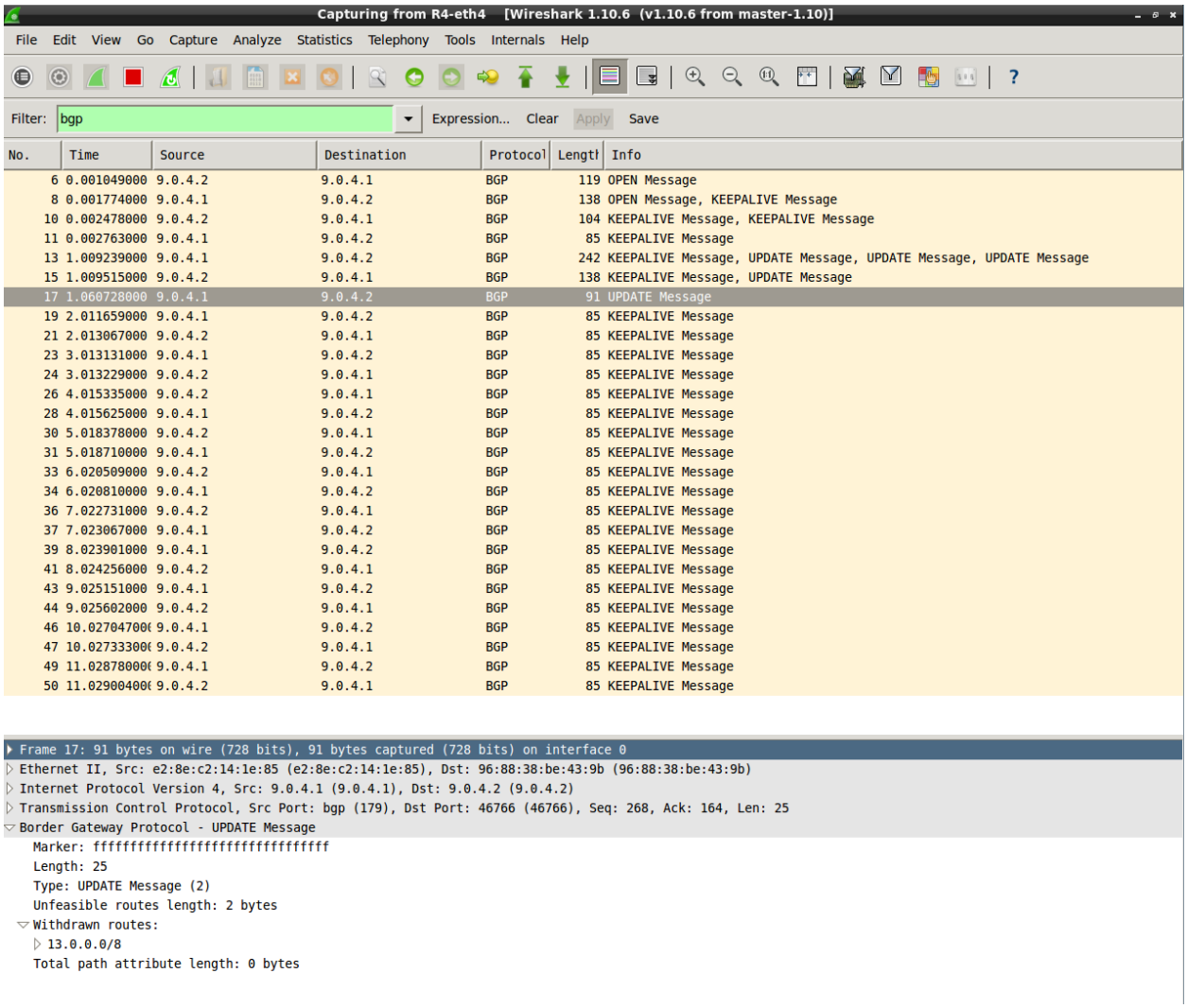
Forwarding table to R2 after running start_rogue.sh : -



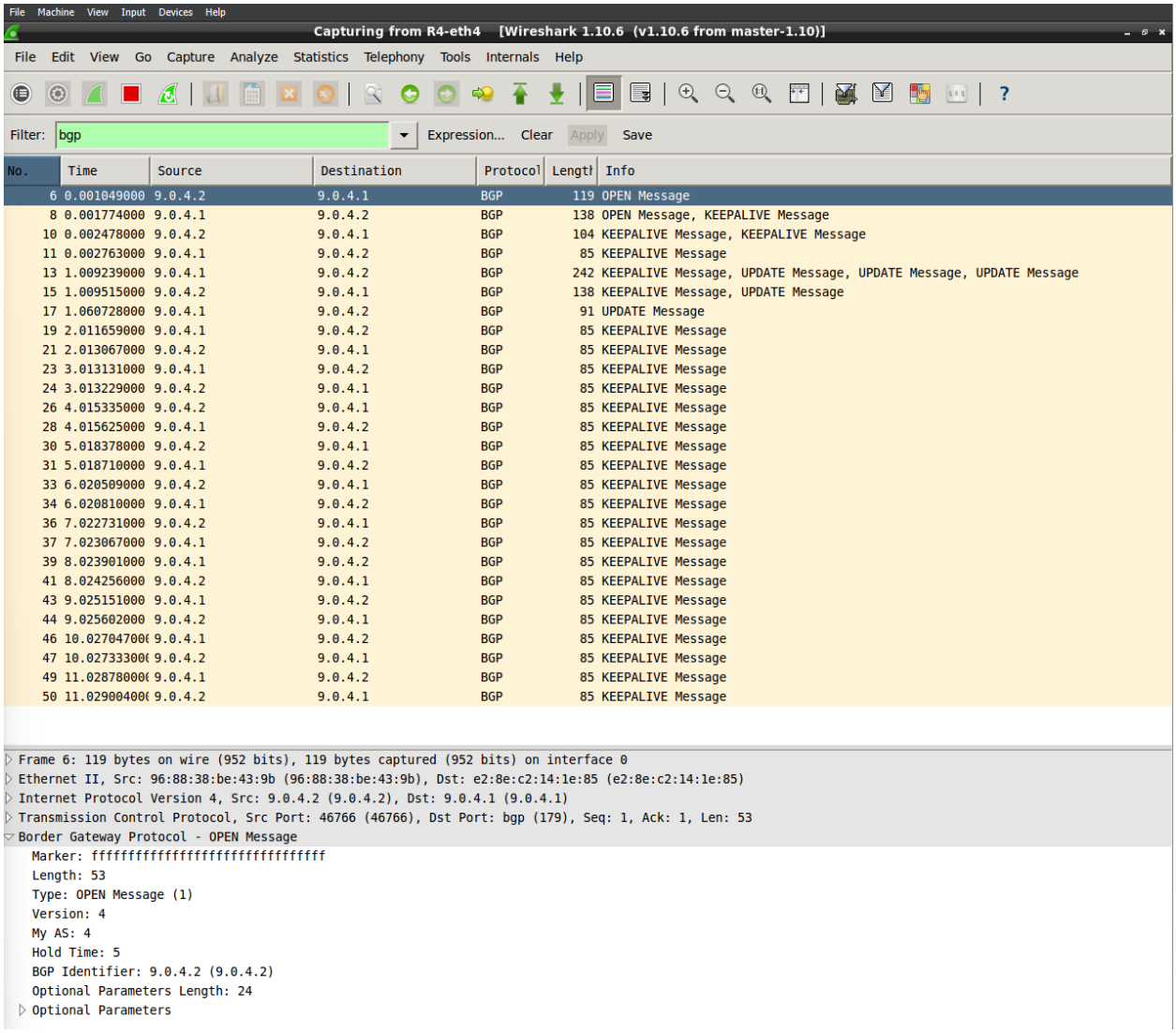
Note here that the gateway for 13.0.0.0 in the case of R2 is still 9.0.1.2 which connects R2 to R3

Ans 12

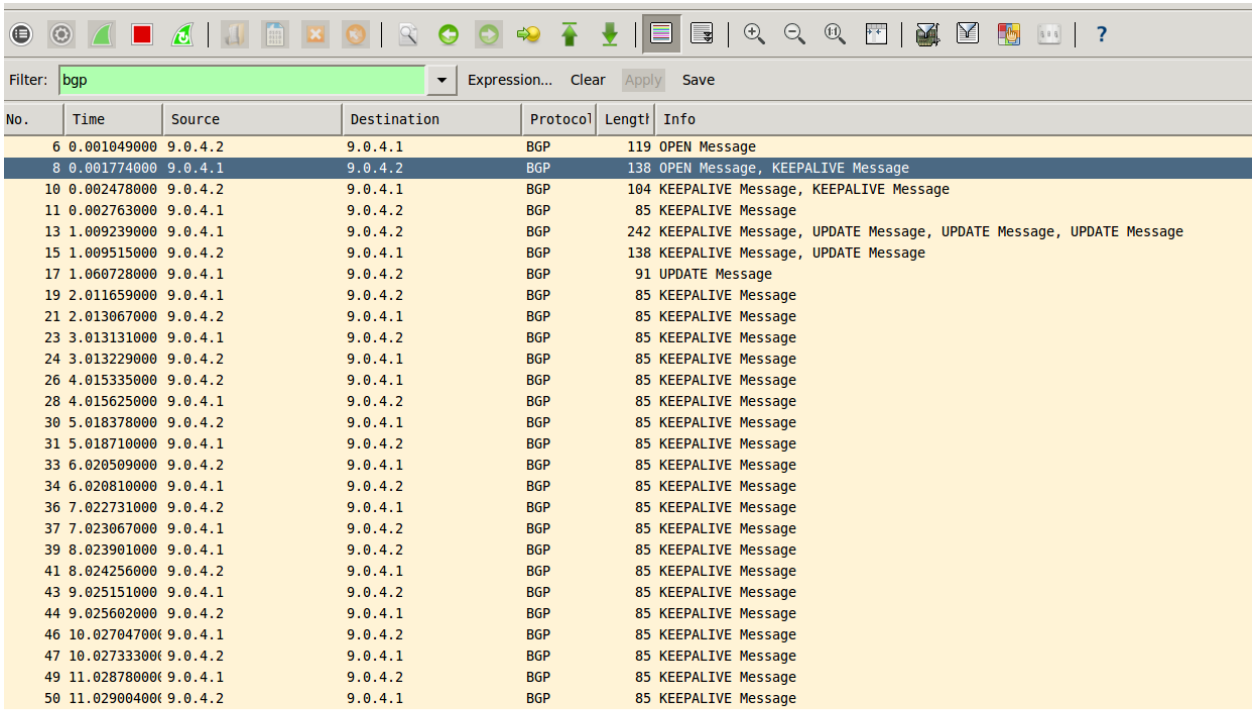
Sniffed packets at the interface connecting R4 and R1 and can see that the BGP advertisements of R4 are being sent to and accepted by R1: -



In the opened packet, we can see that R1 is telling R2 that it has withdrawn the old path to go to 13.0.0.0/8 . We can see the BGP OPEN and BGP Keep alive packets as well which are used to start the BGP transactions keep the connection alive.

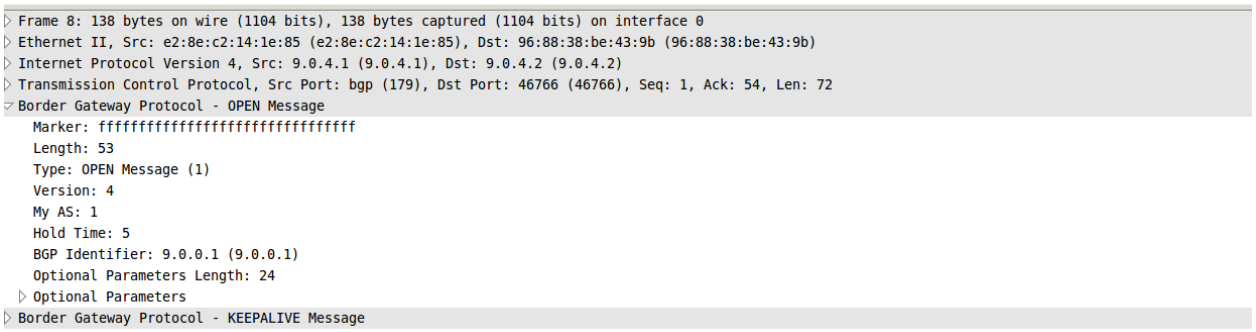


Here, in teh BGP open packet sent by R2 to R1, we can see that R4 is telling R1 that his AS is 4 and IP is 9.0.4.2



Filter: bgp Expression... Clear Apply Save

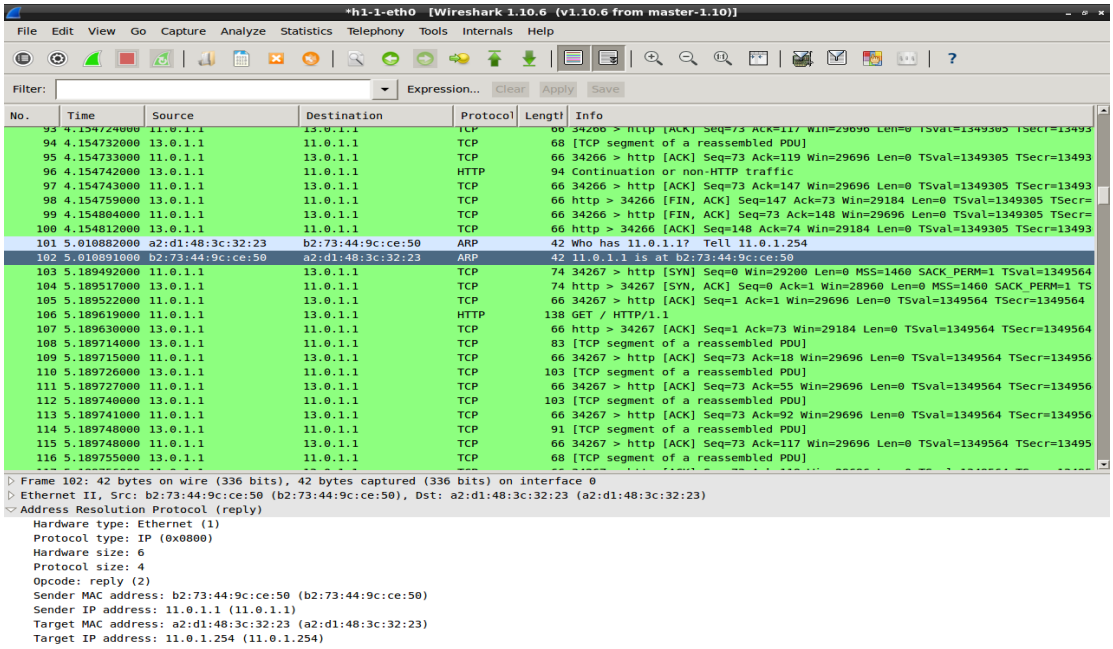
No.	Time	Source	Destination	Protocol	Length	Info
6	0.001049000	9.0.4.2	9.0.4.1	BGP	119	OPEN Message
8	0.001774000	9.0.4.1	9.0.4.2	BGP	138	OPEN Message, KEEPALIVE Message
10	0.002478000	9.0.4.2	9.0.4.1	BGP	104	KEEPALIVE Message, KEEPALIVE Message
11	0.002763000	9.0.4.1	9.0.4.2	BGP	85	KEEPALIVE Message
13	1.009239000	9.0.4.1	9.0.4.2	BGP	242	KEEPALIVE Message, UPDATE Message, UPDATE Message, UPDATE Message
15	1.009515000	9.0.4.2	9.0.4.1	BGP	138	KEEPALIVE Message, UPDATE Message
17	1.060728000	9.0.4.1	9.0.4.2	BGP	91	UPDATE Message
19	2.011659000	9.0.4.1	9.0.4.2	BGP	85	KEEPALIVE Message
21	2.013067000	9.0.4.2	9.0.4.1	BGP	85	KEEPALIVE Message
23	3.013131000	9.0.4.1	9.0.4.2	BGP	85	KEEPALIVE Message
24	3.013229000	9.0.4.2	9.0.4.1	BGP	85	KEEPALIVE Message
26	4.015335000	9.0.4.2	9.0.4.1	BGP	85	KEEPALIVE Message
28	4.015625000	9.0.4.1	9.0.4.2	BGP	85	KEEPALIVE Message
30	5.018378000	9.0.4.2	9.0.4.1	BGP	85	KEEPALIVE Message
31	5.018710000	9.0.4.1	9.0.4.2	BGP	85	KEEPALIVE Message
33	6.020509000	9.0.4.2	9.0.4.1	BGP	85	KEEPALIVE Message
34	6.020810000	9.0.4.1	9.0.4.2	BGP	85	KEEPALIVE Message
36	7.022731000	9.0.4.2	9.0.4.1	BGP	85	KEEPALIVE Message
37	7.023067000	9.0.4.1	9.0.4.2	BGP	85	KEEPALIVE Message
39	8.023901000	9.0.4.1	9.0.4.2	BGP	85	KEEPALIVE Message
41	8.024256000	9.0.4.2	9.0.4.1	BGP	85	KEEPALIVE Message
43	9.025151000	9.0.4.1	9.0.4.2	BGP	85	KEEPALIVE Message
44	9.025602000	9.0.4.2	9.0.4.1	BGP	85	KEEPALIVE Message
46	10.02704700(	9.0.4.1	9.0.4.2	BGP	85	KEEPALIVE Message
47	10.02733300(	9.0.4.2	9.0.4.1	BGP	85	KEEPALIVE Message
49	11.02878000(	9.0.4.1	9.0.4.2	BGP	85	KEEPALIVE Message
50	11.02900400(	9.0.4.2	9.0.4.1	BGP	85	KEEPALIVE Message



```
▷ Frame 8: 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits) on interface 0
▷ Ethernet II, Src: e2:8e:c2:14:1e:85 (e2:8e:c2:14:1e:85), Dst: 96:88:38:be:43:9b (96:88:38:be:43:9b)
▷ Internet Protocol Version 4, Src: 9.0.4.1 (9.0.4.1), Dst: 9.0.4.2 (9.0.4.2)
▷ Transmission Control Protocol, Src Port: bgp (179), Dst Port: 46766 (46766), Seq: 1, Ack: 54, Len: 72
▽ Border Gateway Protocol - OPEN Message
    Marker: ffffffffffffffffffffffffffffffff
    Length: 53
    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
▷ Border Gateway Protocol - KEEPALIVE Message
```

R1 has replied with its BGP OPEN message and has told R4 that its AS is 1. After this the paths have been updated as shown in the first snapshot of this answer.

Also, I can see the ARP request response happening at the time of running start_rogue.sh, when the GET requests were going to h3-1. This indicates that the IP address which was of h3-1 was advertised by h4-1 and that's why the ARP request was triggered when the GET request packet reached R4 for the first time: -



Ans 13

In an ideal scenario, all the packets from R1 intended to go to a host in R3 are forwarded to R2. R2 then forwards them to R3 and R3 forwards them to the specific host based on the IP address and MAC address. When R4 joins the network it connects itself to R1. Then every host of R4 advertises the IP address used by the hosts of R3. h4-1 advertises the IP address of h3-1, h4-2 advertises the address of h3-2 and so on. This updates the BGP and forwarding table of R1 because R1 thinks that now there is a shorter path to reach the hosts of R3.
This is why all the traffic going from R1 to R3 through R2 now starts to go to R4.

Ans 14

I ran ping multiple times on h3-1 IP address - 13.0.1.1 and found that after running the start_rogue.sh script the pings are taking less RTT on average as compared to the pings which were sent before running the start_rogue.sh script. This makes sense, as this IP is now being advertised by h4-1 which is the attacker's host. H4-1 is just 1 hop away from R1 whereas h3-1 is 2 hops away. So the BGP and forwarding tables have been updated for this IP and the shorter path has been updated. This is the reason the requests are going to the attacker host now.

Ans 15

Original code in bgp.py: -

```
def getIP(hostname):
    AS, idx = hostname.replace('h', '').split('-')
    AS = int(AS)
    if AS == 4:
        AS = 3
    ip = '%s.0.%s.1/24' % (10+AS, idx)
    return ip


def getGateway(hostname):
    AS, idx = hostname.replace('h', '').split('-')
    AS = int(AS)
    # This condition gives AS4 the same IP range as AS3 so it can be an
    # attacker.
    if AS == 4:
        AS = 3
    gw = '%s.0.%s.254' % (10+AS, idx)
    return gw
```

Modified code: -

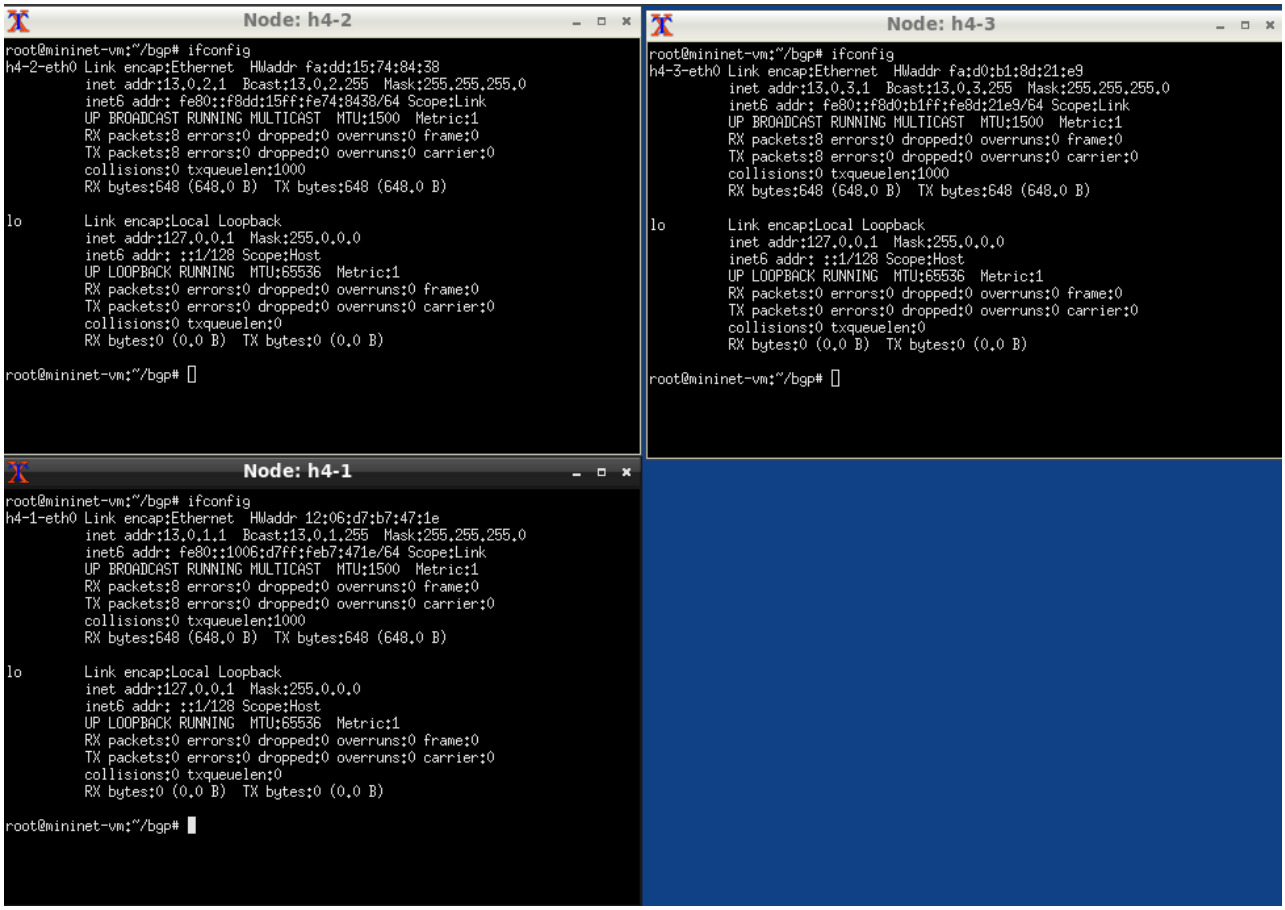
```
def getIP(hostname):
    AS, idx = hostname.replace('h', '').split('-')
    AS = int(AS)
    if hostname == 'h4-1':
        AS = 3
        idx = 1
    ip = '%s.0.%s.1/24' % (10+AS, idx)
    return ip


def getGateway(hostname):
    AS, idx = hostname.replace('h', '').split('-')
    AS = int(AS)
    # This condition gives AS4 the same IP range as AS3 so it can be an
    # attacker.
    if hostname == 'h4-1':
        AS = 3
        idx = 1
    gw = '%s.0.%s.254' % (10+AS, idx)
    return gw
```

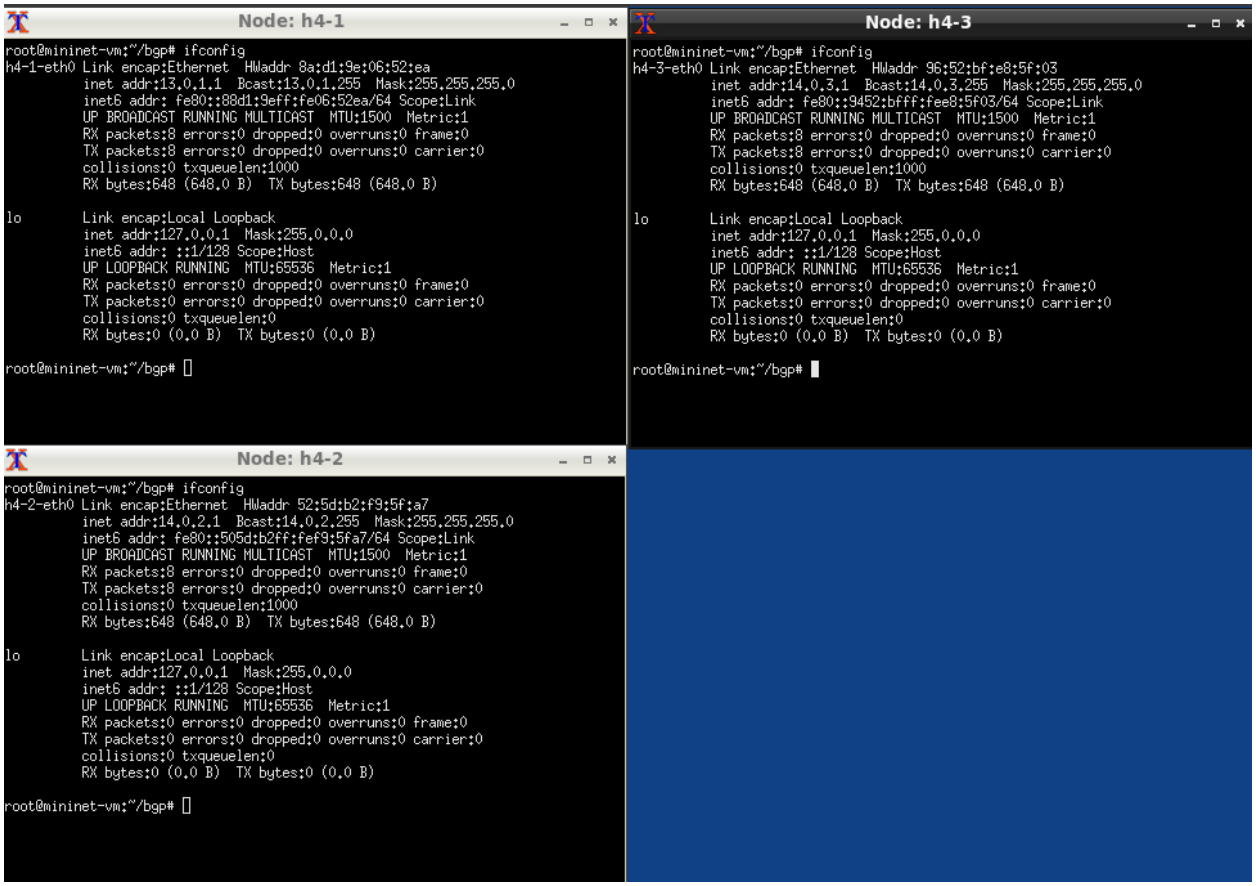
This will give the IP of h3-1 to h4-1. The IP addresses of h4-2 and h4-3 will be new and not the same as h3-2 and h3-3.

h4-2 new IP - 14.0.2.1
h4-3 new IP - 14.0.3.1

IPs of R4 hosts when bgp.py is used to set up topology: -



IPs of R4 hosts when bgp_modified.py is used to set-up topology: -



Next, we need to change the inet address of R4 router's interfaces which connect with h4-2 and h4-3

For this we need to update the bgpd-R4.conf and zebra-R4.conf files.

Original bgpd-conf file: -

```
! -*- 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 13.0.0.0/8
  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

log file /tmp/R4-bgpd.log

debug bgp as4
debug bgp events
debug bgp filters
debug bgp fsm
debug bgp keepalives
debug bgp updates

!
log stdout
```

Modified bgpd-conf file: -

```
! -*- 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 13.0.1.0/24
  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

log file /tmp/R4-bgpd.log

debug bgp as4
debug bgp events
debug bgp filters
debug bgp fsm
debug bgp keepalives
debug bgp updates

!
log stdout
```

2 network IPs are added for R4 now - one is 13.0.1.0/24 which is used for hacking and the other is 14.0.0.0/8 which will be used for the remaining hosts.

Original zebra-R4.conf file: -

```
! -*- 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 13.0.2.254/24

interface R4-eth3
  ip address 13.0.3.254/24

!

interface R4-eth4
  ip address 9.0.4.2/24

log file /tmp/R4.log
```

Modified zebra-R4.conf file: -

```
! -*- 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
  ip address 14.0.3.254/24

!

interface R4-eth4
  ip address 9.0.4.2/24

log file /tmp/R4.log
```

Note that R4-eth2 and R4-eth3 are now updated to connect with the updated Ip addresses of h4-2 and h4-3 hosts.

R4 ifconfig details after making the changes: -

Note that R4-eth1 has inet address - 13.0.1.254 where has the inet addresses of all other interfaces are starting with 14 now.

Updated bgp table of R1: -

```
   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 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
bgpd-R1#
```

Note here that now there are 2 new entries - 13.0.1.0/24 and 14.0.0.0 in the table. The paths for both of these entries are - ‘4 i’. Hence, all the requests to 13.0.1.1 Ip address will go to R4 and eventually to attacker web server. Also, all the requests to remaining R4 hosts will also go to R4 as they will get matched with the 14.0.0.0 network IP.

Note that, all the request to R3 hosts (apart from h1-1) will go to R3 only and the same has been highlighted by the path of 13.0.0.0 network IP which is - ‘2 3 i’

Longest Prefix matching will come into play here and the requests to 13.0.1.1 (h1-1) will go to R4 as they will be matched with 13.0.1.0/24

Updated routing table of 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        9.0.0.2         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-vm:~/bgp#
```

Note that everypacket which get matched with 13.0.0.0 will be sent to R1-eth4 (R2).
Every packet which get matched with 13.0.1.0 (longest prefix matching) and 14.0.0.0 will be sent to R1-eth5(R4)

ANTI-PLAGIARISM Statement <Include it in your report>

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.

Sanyam Kaul - CS23MTECH14011
Bhargav Patel - CS23MTECH11026
Arnab Ghosh - CS23MTECH11025
Date: 1/12/2023

References:

- https://github.com/mininet/mininet/wiki/BGP-Path-Hijacking-Attack-Demo
- https://bitbucket.org/jvimal/bgp/src/master/