دانشکده مهندسی کامپیوتر

آزمایشگاه شبکه‌های کامپیوتری

گزارش کار آزمایش ۱

گروه ۴

علی صداقی ۹۷۵۲۱۳۷۸

دانیال بازمانده ۹۷۵۲۱۱۳۵

۱ بخش الف

پیاده‌سازی این بخش با تغییرات درون فایل lanTopolgy.py ایجاد شده است.

```
info( '*** Adding hosts\n' )
h1 = net.addHost( 'h1')
h2 = net.addHost( 'h2')
h3 = net.addHost( 'h3')
h4 = net.addHost( 'h4')


info( '*** Adding switch\n' )
s14 = net.addSwitch( 's14' )
s24 = net.addSwitch( 's24' )
s34 = net.addSwitch( 's34' )

info( '*** Creating links\n' )
net.addLink( h1, s14 )
net.addLink( h2, s24 )
net.addLink( h3, s34 )
net.addLink( h4, s14 )
net.addLink( h4, s24 )
net.addLink( h4, s34 )
```

```
#This is used to run commands on the hosts
info( '*** Starting terminals on hosts\n' )
h1.cmd('xterm -xrm "XTerm.vt100.allowTitleOps: false" -T h1 &')
h2.cmd('xterm -xrm "XTerm.vt100.allowTitleOps: false" -T h2 &')
h3.cmd('xterm -xrm "XTerm.vt100.allowTitleOps: false" -T h3 &')
h4.cmd('xterm -xrm "XTerm.vt100.allowTitleOps: false" -T h4 &')

info( '*** Running the command line interface\n' )
CLI( net )

info( '*** Closing the terminals on the hosts\n' )
h1.cmd("killall xterm")
h2.cmd("killall xterm")
h3.cmd("killall xterm")
h4.cmd("killall xterm")
```

از دستور scp برای انتقال این فایل از ماشین لوکال به ماشین مجازی استفاده می‌کنیم.

scp lanTopology.py mininet@192.168.83.3:/home/mininet/py-scripts

برنامه xterm و محیط گرافیکی را نیز از طریق دستور زیر نصب می‌کنیم:

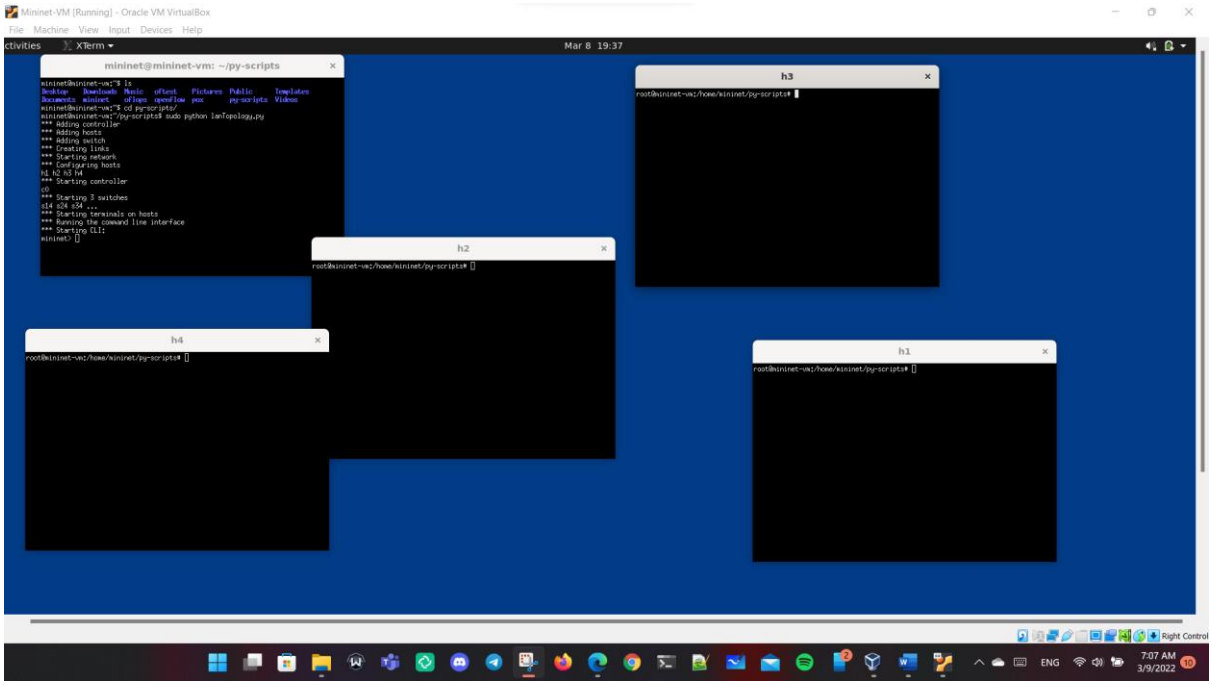
sudo apt-get update

sudo apt-get install xinit x11-xserver-utils lxde

sudo apt-get install xterm

شبکه درون فایل را از طریق دستور زیر اجرا می‌کنیم:

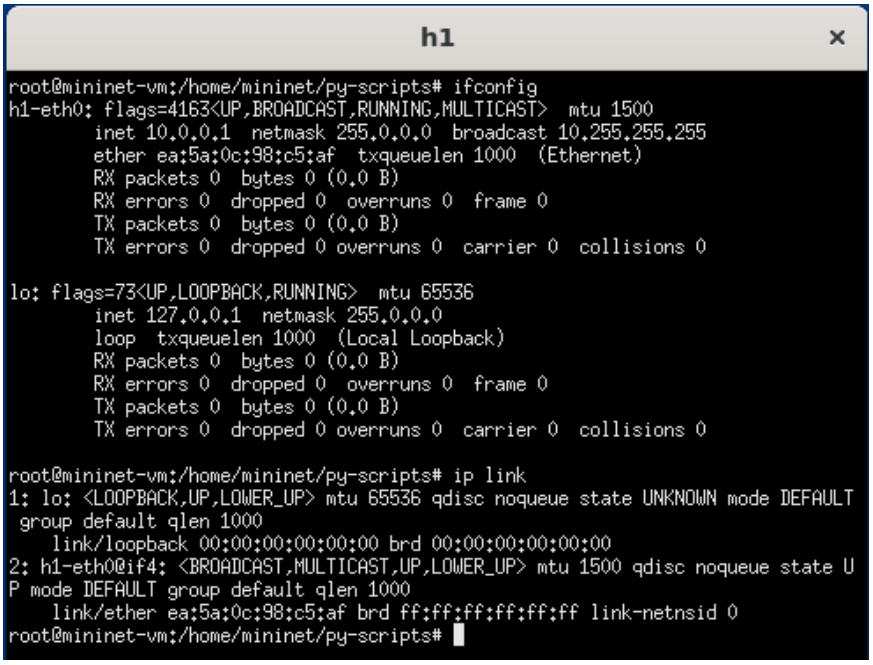
sudo mn --custom lanTopology.py --topo=lanTopology

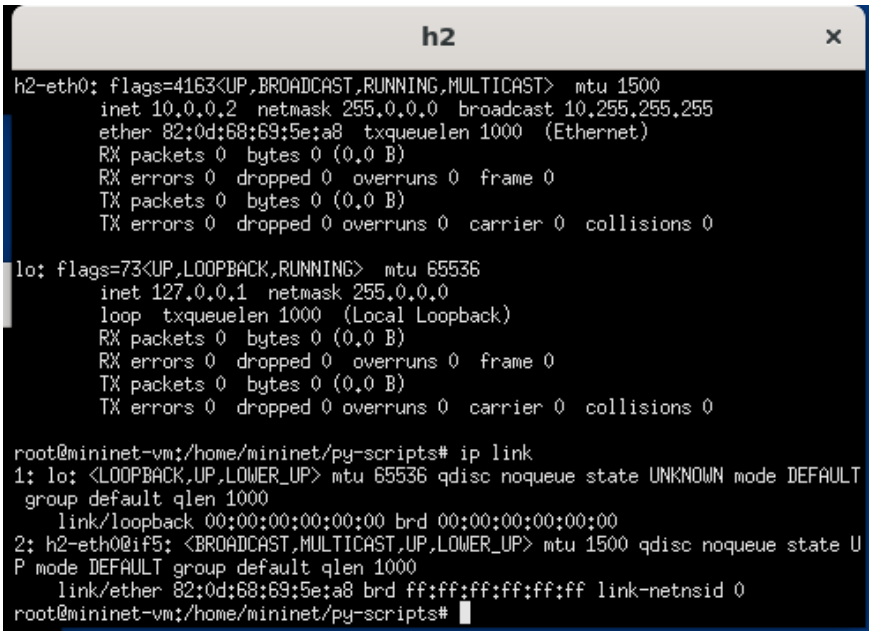


۲ سوال ب

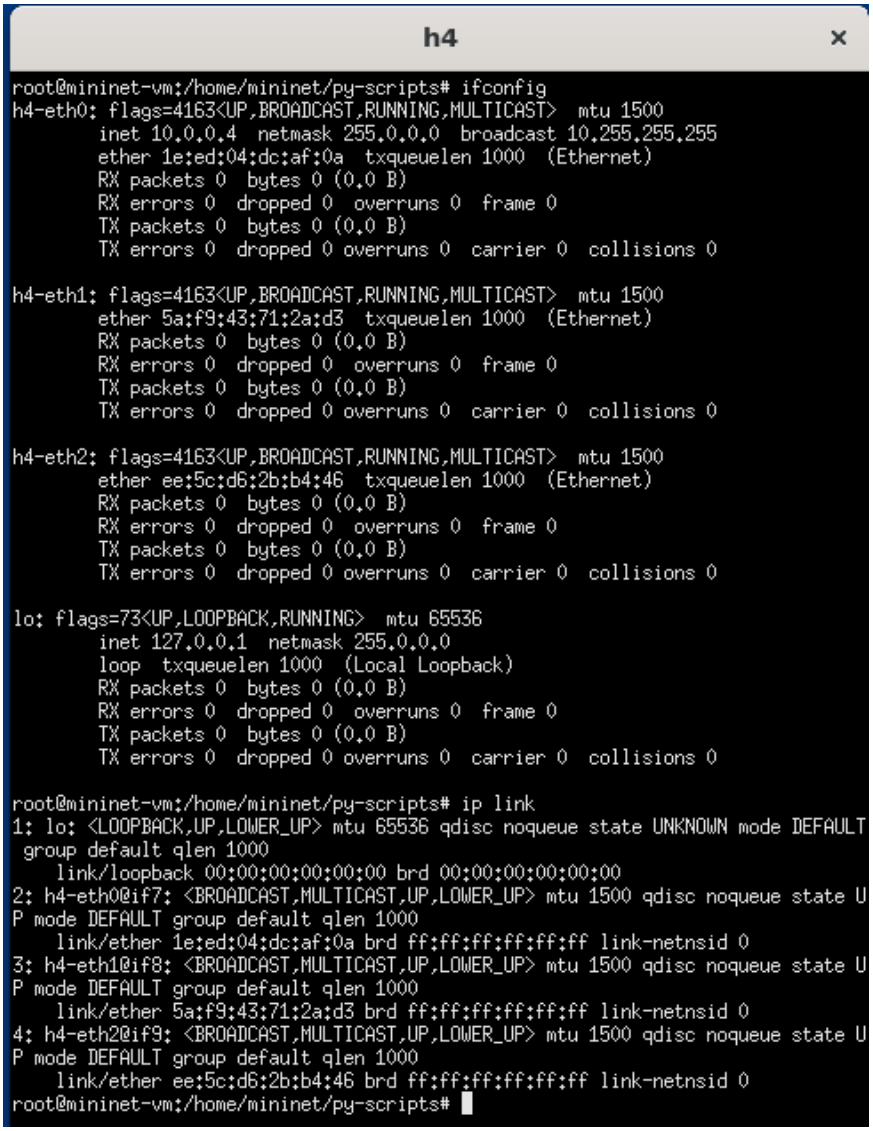
دستورات ip link و ifconfig را روی همه هاست‌ها اجرا می‌کنیم.

مشاهده می‌کنیم تمامی اینترفیس‌ها در مد UP هستند و نیازی به زدن دستور توسط ما نیست.






h3
root@mininet-vm:/home/mininet/py-scripts# ifconfig
h3-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 10.0.0.3 netmask 255.0.0.0 broadcast 10.255.255.255
ether 9e:7c:15:bb:c2:9a txqueuelen 1000 (Ethernet)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
loop txqueuelen 1000 (Local Loopback)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
root@mininet-vm:/home/mininet/py-scripts# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
group default qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: h3-eth0@if6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state U
P mode DEFAULT group default qlen 1000
link/ether 9e:7c:15:bb:c2:9a brd ff:ff:ff:ff:ff:ff link-netnsid 0
root@mininet-vm:/home/mininet/py-scripts#

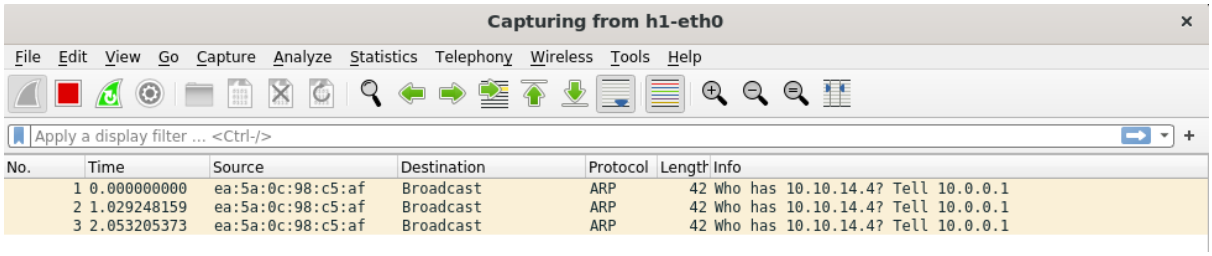

h4
root@mininet-vm:/home/mininet/py-scripts# ifconfig
h4-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 10.0.0.4 netmask 255.0.0.0 broadcast 10.255.255.255
ether 1e:ed:04:dc:af:0a txqueuelen 1000 (Ethernet)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
h4-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
ether 5a:f9:43:71:2a:d3 txqueuelen 1000 (Ethernet)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
h4-eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
ether ee:5c:d6:2b:b4:46 txqueuelen 1000 (Ethernet)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
loop txqueuelen 1000 (Local Loopback)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
root@mininet-vm:/home/mininet/py-scripts# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
group default qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: h4-eth0@if7: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state U
P mode DEFAULT group default qlen 1000
link/ether 1e:ed:04:dc:af:0a brd ff:ff:ff:ff:ff:ff link-netnsid 0
3: h4-eth1@if8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state U
P mode DEFAULT group default qlen 1000
link/ether 5a:f9:43:71:2a:d3 brd ff:ff:ff:ff:ff:ff link-netnsid 0
4: h4-eth2@if9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state U
P mode DEFAULT group default qlen 1000
link/ether ee:5c:d6:2b:b4:46 brd ff:ff:ff:ff:ff:ff link-netnsid 0
root@mininet-vm:/home/mininet/py-scripts#

روی هاست h1 برنامه Wireshark را اجرا می‌کنیم و به اینترفیس h1-eth0 آن گوش می‌دهیم. یک بسته پینگ نیز از h1 به آدرس 10.10.14.4 ارسال می‌کنیم.

```
root@mininet-vm:/home/mininet/py-scripts# sudo wireshark
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
^Z
[1]+  Stopped                 sudo wireshark
root@mininet-vm:/home/mininet/py-scripts# bg
[1]+ sudo wireshark &
root@mininet-vm:/home/mininet/py-scripts# ping 10.10.14.4 -c 1
PING 10.10.14.4 (10.10.14.4) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable

--- 10.10.14.4 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

root@mininet-vm:/home/mininet/py-scripts#
```



سوال ۱)

درخواست ICMP (Request) ارسال نمی‌شود.

جدول ARP در h1 آدرس فیزیکی معادل IP داده شده را ندارد.

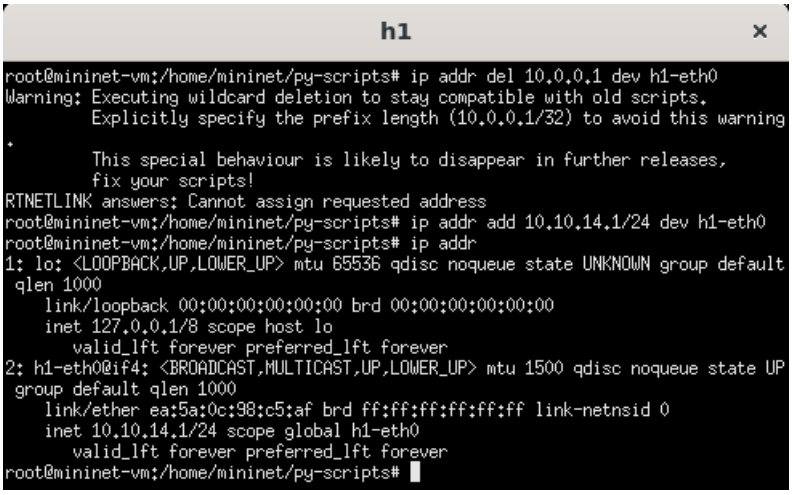
در ابتدا درخواست ARP (Request) فرستاده می‌شود و از آنجایی که چنین IP ای در شبکه موجود نیست، پاسخی دریافت نمی‌شود.

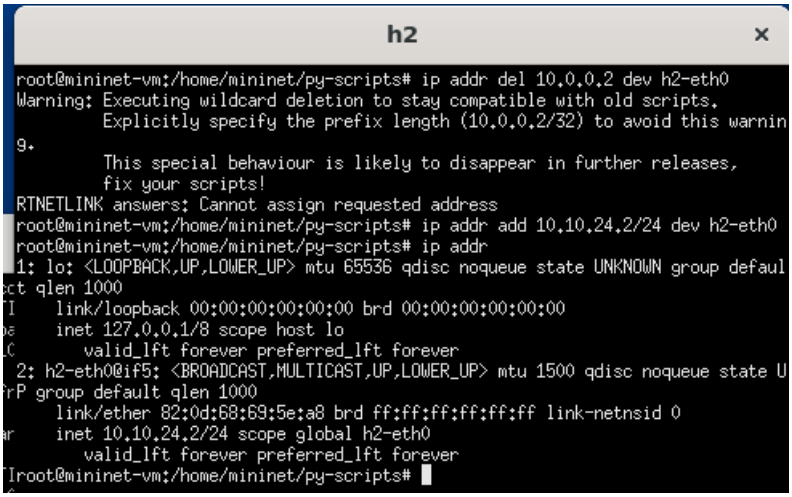
یعنی هیچ reply ARP ارسال نمی‌شود و دریافت نیز نمی‌شود.

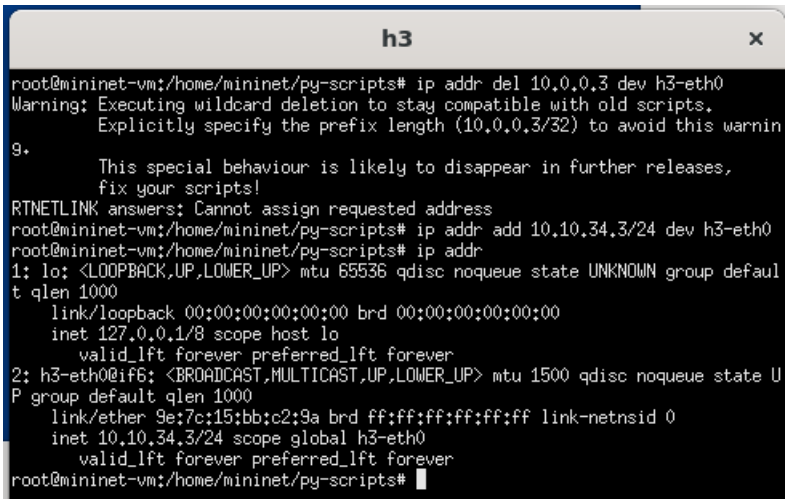
```
root@mininet-vm:/home/mininet/py-scripts# arp -a
? (10.0.0.4) at 1e:ed:04:dc:af:0a [ether] on h1-eth0
? (10.10.14.4) at <incomplete> on h1-eth0
root@mininet-vm:/home/mininet/py-scripts#
```

۳ سوال ج

دستورات ip addr del و ip addr add را روی هر ۴ هاست به صورت زیر اجرا می‌کنیم.





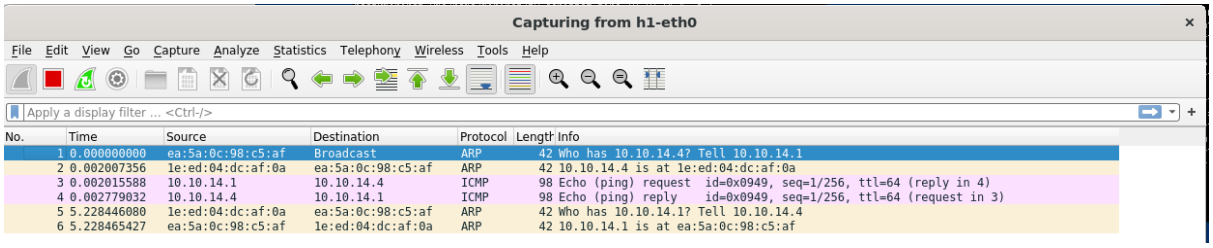


```
h4
root@mininet-vm:/home/mininet/py-scripts# ip addr del 10.0.0.4 dev h4-eth0
Warning: Executing wildcard deletion to stay compatible with old scripts.
         Explicitly specify the prefix length (10.0.0.4/32) to avoid this warnin
g.
         This special behaviour is likely to disappear in further releases,
         fix your scripts!
root@mininet-vm:/home/mininet/py-scripts# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: h4-eth0@if7: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state U
P group default qlen 1000
    link/ether 1e:ed:04:dc:af:0a brd ff:ff:ff:ff:ff:ff link-netnsid 0
3: h4-eth1@if8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state U
P group default qlen 1000
    link/ether 5a:f9:43:71:2a:d3 brd ff:ff:ff:ff:ff:ff link-netnsid 0
4: h4-eth2@if9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state U
P group default qlen 1000
    link/ether ee:5c:d6:2b:b4:46 brd ff:ff:ff:ff:ff:ff link-netnsid 0
root@mininet-vm:/home/mininet/py-scripts# ip addr add 10.10.14.4/24 dev h4-eth0
root@mininet-vm:/home/mininet/py-scripts# ip addr add 10.10.24.4/24 dev h4-eth1
root@mininet-vm:/home/mininet/py-scripts# ip addr add 10.10.34.4/24 dev h4-eth2
root@mininet-vm:/home/mininet/py-scripts# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: h4-eth0@if7: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state U
P group default qlen 1000
    link/ether 1e:ed:04:dc:af:0a brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.10.14.4/24 scope global h4-eth0
       valid_lft forever preferred_lft forever
3: h4-eth1@if8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state U
P group default qlen 1000
    link/ether 5a:f9:43:71:2a:d3 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.10.24.4/24 scope global h4-eth1
       valid_lft forever preferred_lft forever
4: h4-eth2@if9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state U
P group default qlen 1000
    link/ether ee:5c:d6:2b:b4:46 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.10.34.4/24 scope global h4-eth2
       valid_lft forever preferred_lft forever
root@mininet-vm:/home/mininet/py-scripts#
```

یک بسته پینگ از مبدا h1 به 10.10.14.4 ارسال می‌کنیم.

```
h1
root@mininet-vm:/home/mininet/py-scripts# sudo wireshark
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
^Z
[1]+  Stopped                 sudo wireshark
root@mininet-vm:/home/mininet/py-scripts# bg
[1]+ sudo wireshark &
root@mininet-vm:/home/mininet/py-scripts# ping 10.10.14.4 -c 1
PING 10.10.14.4 (10.10.14.4) 56(84) bytes of data.
64 bytes from 10.10.14.4: icmp_seq=1 ttl=64 time=2.81 ms

--- 10.10.14.4 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.814/2.814/2.814/0.000 ms
root@mininet-vm:/home/mininet/py-scripts# arp -a
? (10.10.14.4) at 1e:ed:04:dc:af:0a [ether] on h1-eth0
root@mininet-vm:/home/mininet/py-scripts#
```

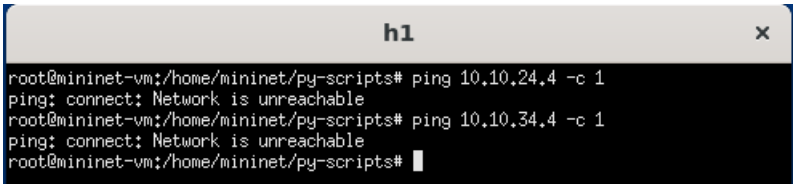


سوال ۲)

بله Wireshark توانسته هر دو بسته ARP Reply و ICMP Request را روی اینترفیس h1-eth0 انجام دهد. در واقع هاست h1 موفق شده آدرس واقعی h4 را پیدا کند.

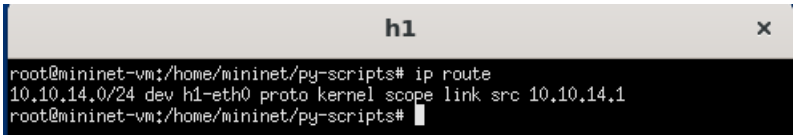
۴ بخش د

سوال ۳)



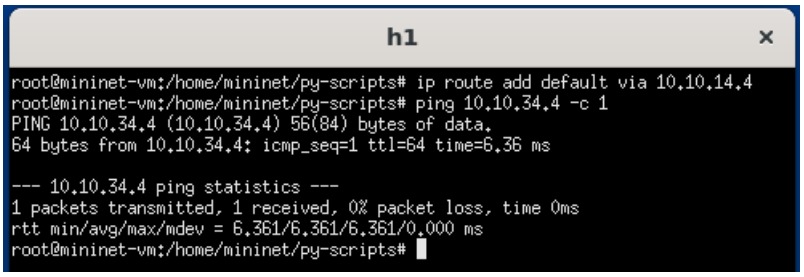
خیر کار نمی‌کند. زیرا میان این دو آدرس هیچ اینترفیسی وجود ندارد. h4 نیز نمی‌تواند پیام‌های ICMP را Forward کند.

سوال ۴)



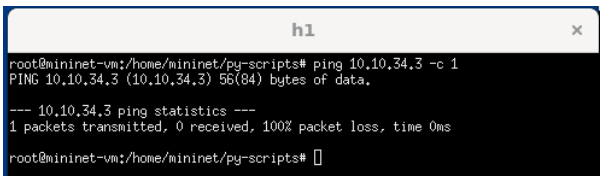
در اینجا دو مورد وجود دارد. یکی 10.10.14.0/24 که به عنوان مبدا هر بسته ایست که اینترفیسی برای ارسال آن تعیین نشده. دومی نیز 10.10.14.1 است که بیانگر h1-eth0 است.

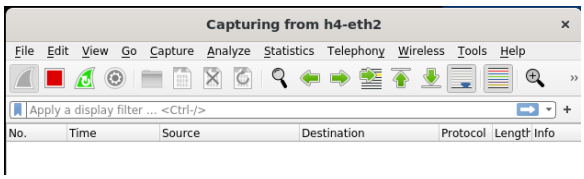
سوال ۵)

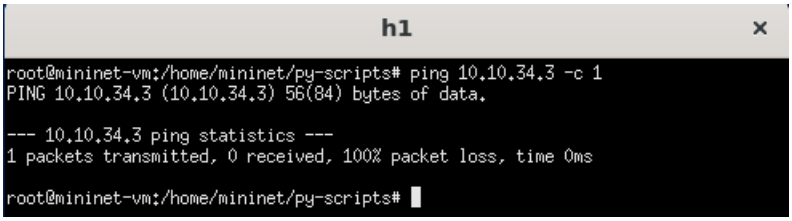


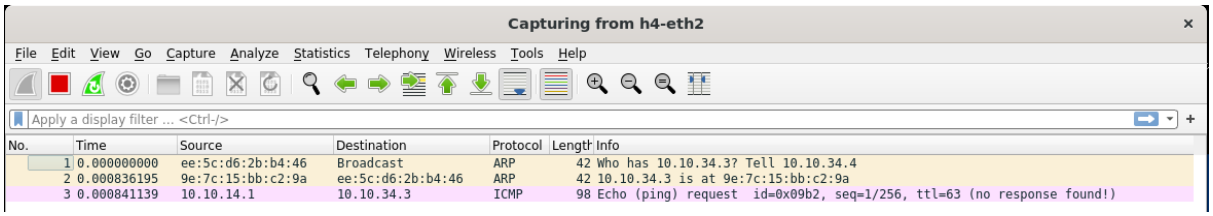
این بار اینترفیس eth2 در هاست h4 از طریق h1 قابل دسترسی است. زیرا Default Gateway را در h1 برابر اینترفیسی قرار دادیم که با اینترفیس مقصد در ارتباط است.

سوال ۶)









حل مشکل:

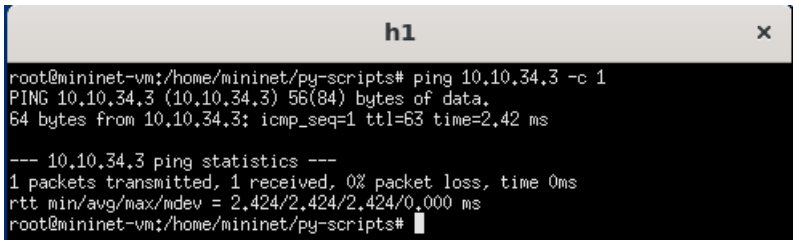
بایستی Default Gateway بر روی h2 و h3 را نیز تغییر دهیم.

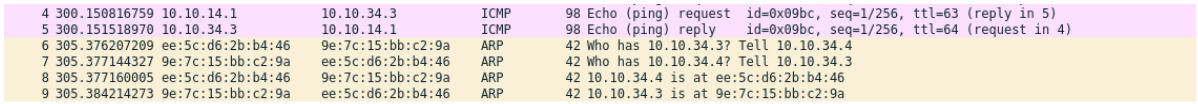
روی h2:

ip route add default via 10.10.24.4

روی h3:

ip route add default via 10.10.34.4

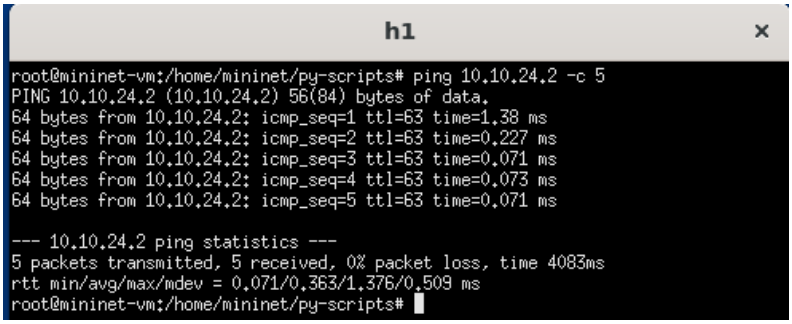




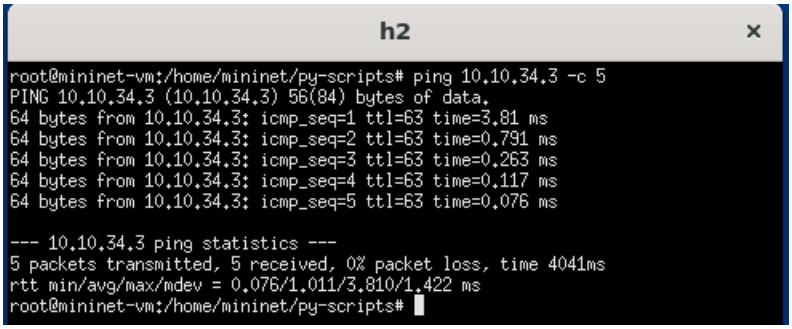
همانطور که مشاهده می‌شود آدرس پیدا شد و پینگ گرفته شد.

سوال ۷) بایستی در هر هاست Default Gatewayها را تعریف کنیم تا به h4 (مرکزی) متصل شوند.

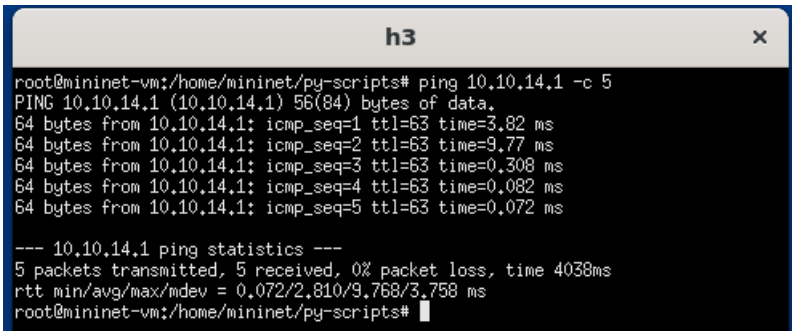
h2 از طریق h1: مقدار RTT: 0.363



h3 از طریق h2: مقدار RTT: 1.011



h1 از طریق h3: مقدار RTT: 2.810



با توجه به وجود تقارن در شبکه مقادیر RTT نباید اختلاف زیادی با هم داشته باشند. که این مورد در تصاویر نیز مشاهده می‌شود (با شک)