

به نام خدا

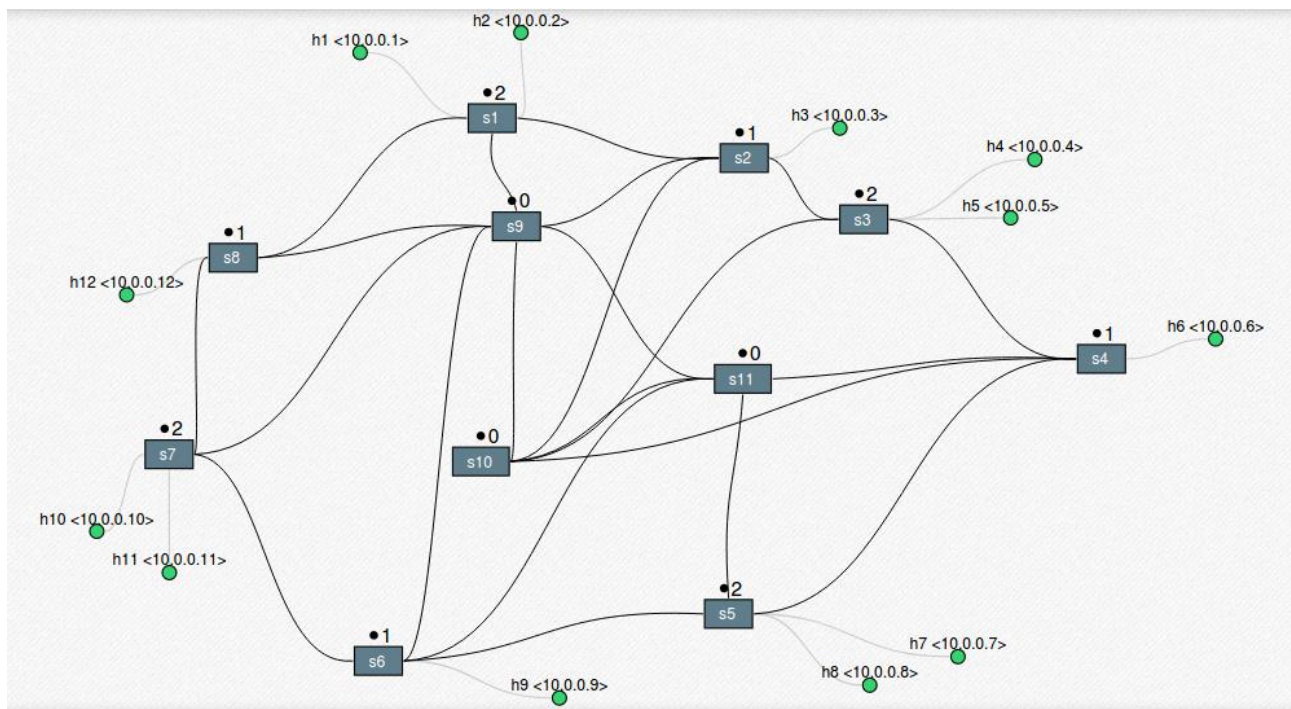
حدیث غفوری ۹۸۲۵۴۱۳

پروژه ۳ شبکه ۲

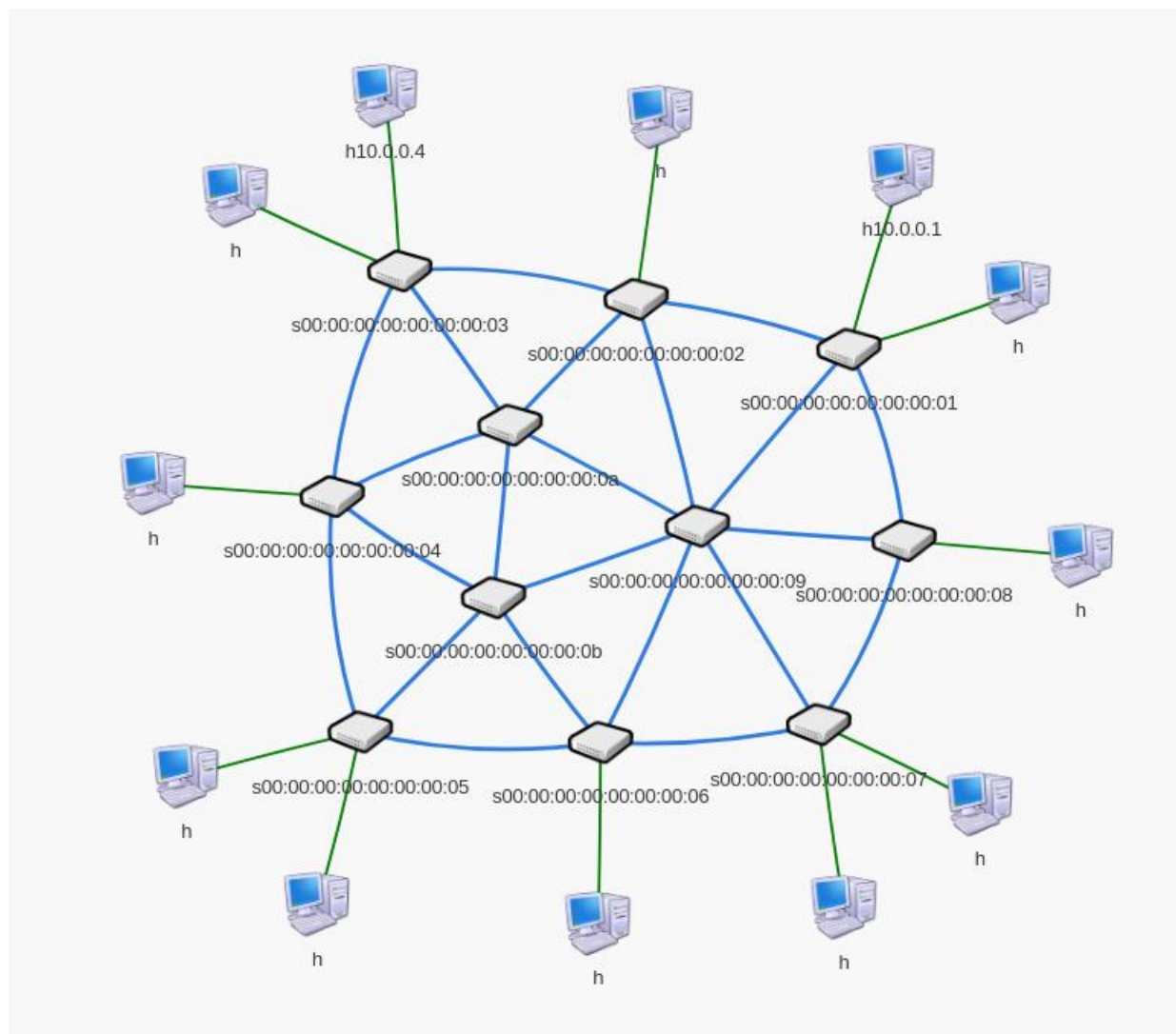
سوال ۱ تصویر توپولوژی ایجاد شده:

```
hadis@ubuntu: ~/Downloads/python_examples
hadis@ubuntu:~/Downloads/python_examples$ sudo mn --custom ex5.py --topo project --controller=remote
,ip=127.0.0.1,port=6653
[sudo] password for hadis:
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12
*** Adding switches:
s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11
*** Adding links:
(h1, s1) (h2, s1) (h3, s2) (h4, s3) (h5, s3) (h6, s4) (h7, s5) (h8, s5) (h9, s6) (h10, s7) (h11, s7)
(h12, s8) (s1, s2) (s1, s8) (s1, s9) (s2, s3) (s2, s9) (s2, s10) (s3, s4) (s3, s10) (s4, s5) (s4, s
10) (s4, s11) (s5, s6) (s5, s11) (s6, s7) (s6, s9) (s6, s11) (s7, s8) (s7, s9) (s8, s9) (s9, s10) (s
9, s11) (s10, s11)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12
*** Starting controller
c0
*** Starting 11 switches
s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 ...
*** Starting CLI:
mininet>
```

تصویر ایجاد شده در topology visualizer



تصویر ایجاد شده در محیط topology در فلودلایت



سوال ۳

هر دو میزبان با یک دیگر ارتباط دارند و میتوانیم هردو هاست در توپولوژی را باهم پینگ کنیم. زیرا همه از طریق سویچ ها به هم وصل شده اند. مثل مثال های زیر:

```
mininet> h2 ping h8
PING 10.0.0.8 (10.0.0.8) 56(84) bytes of data.
64 bytes from 10.0.0.8: icmp_seq=1 ttl=64 time=11.7 ms
64 bytes from 10.0.0.8: icmp_seq=2 ttl=64 time=0.348 ms
64 bytes from 10.0.0.8: icmp_seq=3 ttl=64 time=0.050 ms
64 bytes from 10.0.0.8: icmp_seq=4 ttl=64 time=0.056 ms
^C
--- 10.0.0.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3051ms
rtt min/avg/max/mdev = 0.050/3.053/11.761/5.029 ms
mininet> h5 ping h11
PING 10.0.0.11 (10.0.0.11) 56(84) bytes of data.
64 bytes from 10.0.0.11: icmp_seq=1 ttl=64 time=16.9 ms
64 bytes from 10.0.0.11: icmp_seq=2 ttl=64 time=0.250 ms
64 bytes from 10.0.0.11: icmp_seq=3 ttl=64 time=0.055 ms
64 bytes from 10.0.0.11: icmp_seq=4 ttl=64 time=0.071 ms
^C
--- 10.0.0.11 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3045ms
rtt min/avg/max/mdev = 0.055/4.335/16.967/7.293 ms
mininet>
```

```
hadi@ubuntu: ~/Downloads/python_examples
mininet> h1 ifconfig
h1 eth0: Link encap:Ethernet HWaddr 5a:3a:84:82:fd:41
      inet addr:10.0.0.1 Bcast:10.255.255.255 Mask:255.0.0.0
      inet6 addr: fe80::583a:34ff:fe82:fd41/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:888363 errors:0 dropped:170 overruns:0 frame:0
      TX packets:4312 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:114251870 (114.2 MB) TX bytes:370704 (370.7 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:1000
      RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

mininet>
```

```
hadis@ubuntu: ~/Downloads/python_examples
mininet> h4 ifconfig
h4-eth0: Link encap:Ethernet HWaddr 1e:1f:25:86:d7:96
        inet addr:10.0.0.4 Bcast:10.255.255.255 Mask:255.0.0.0
        inet6 addr: fe80::1c1f:25ff:fe86:d796/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:952823 errors:0 dropped:174 overruns:0 frame:0
        TX packets:4455 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:122423051 (122.4 MB) TX bytes:383002 (383.0 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:1000
        RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

mininet> 
```

ارتباط دو هاست h1,h4 به کمک xterm

"Node: h4"	"Node: h1"
<pre>root@ubuntu:~/Downloads/python_examples# ping 10.0.0.1 PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data: From 10.0.0.4 icmp_seq=1 Destination Host Unreachable From 10.0.0.4 icmp_seq=2 Destination Host Unreachable From 10.0.0.4 icmp_seq=3 Destination Host Unreachable From 10.0.0.4 icmp_seq=4 Destination Host Unreachable From 10.0.0.4 icmp_seq=5 Destination Host Unreachable From 10.0.0.4 icmp_seq=6 Destination Host Unreachable From 10.0.0.4 icmp_seq=7 Destination Host Unreachable From 10.0.0.4 icmp_seq=8 Destination Host Unreachable From 10.0.0.4 icmp_seq=9 Destination Host Unreachable ^C --- 10.0.0.1 ping statistics --- 10 packets transmitted, 0 received, +9 errors, 100% packet loss, time 9196ms pipe 4 root@ubuntu:~/Downloads/python_examples# </pre>	<pre>root@ubuntu:~/Downloads/python_examples# ping 10.0.0.4 PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data: From 10.0.0.1 icmp_seq=1 Destination Host Unreachable From 10.0.0.1 icmp_seq=2 Destination Host Unreachable From 10.0.0.1 icmp_seq=3 Destination Host Unreachable From 10.0.0.1 icmp_seq=4 Destination Host Unreachable From 10.0.0.1 icmp_seq=5 Destination Host Unreachable From 10.0.0.1 icmp_seq=6 Destination Host Unreachable From 10.0.0.1 icmp_seq=7 Destination Host Unreachable From 10.0.0.1 icmp_seq=8 Destination Host Unreachable From 10.0.0.1 icmp_seq=9 Destination Host Unreachable ^C --- 10.0.0.4 ping statistics --- 10 packets transmitted, 0 received, +9 errors, 100% packet loss, time 9223ms pipe 4 root@ubuntu:~/Downloads/python_examples# </pre>

Display all the nodes controller, switch and hosts of the topology

```
mininet> nodes
available nodes are:
c0 h1 h10 h11 h12 h2 h3 h4 h5 h6 h7 h8 h9 s1 s10 s11 s2 s3 s4 s5 s6 s7 s8 s9
mininet>
```

Show the links connection.

```
hadis@ubuntu: ~/Downloads/python_examples
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
h3 h3-eth0:s2-eth1
h4 h4-eth0:s3-eth1
h5 h5-eth0:s3-eth2
h6 h6-eth0:s4-eth1
h7 h7-eth0:s5-eth1
h8 h8-eth0:s5-eth2
h9 h9-eth0:s6-eth1
h10 h10-eth0:s7-eth1
h11 h11-eth0:s7-eth2
h12 h12-eth0:s8-eth1
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0 s1-eth3:s2-eth2 s1-eth4:s8-eth2 s1-eth5:s9-eth1
s2 lo: s2-eth1:h3-eth0 s2-eth2:s1-eth3 s2-eth3:s3-eth3 s2-eth4:s9-eth2 s2-eth5:s10-eth1
s3 lo: s3-eth1:h4-eth0 s3-eth2:h5-eth0 s3-eth3:s2-eth3 s3-eth4:s10-eth2 s3-eth5:s4-eth2
s4 lo: s4-eth1:h6-eth0 s4-eth2:s3-eth5 s4-eth3:s5-eth3 s4-eth4:s10-eth3 s4-eth5:s11-eth1
s5 lo: s5-eth1:h7-eth0 s5-eth2:h8-eth0 s5-eth3:s4-eth3 s5-eth4:s11-eth2 s5-eth5:s6-eth2
s6 lo: s6-eth1:h9-eth0 s6-eth2:s5-eth5 s6-eth3:s7-eth3 s6-eth4:s9-eth3 s6-eth5:s11-eth3
s7 lo: s7-eth1:h10-eth0 s7-eth2:h11-eth0 s7-eth3:s6-eth3 s7-eth4:s8-eth3 s7-eth5:s9-eth4
s8 lo: s8-eth1:h12-eth0 s8-eth2:s1-eth4 s8-eth3:s7-eth4 s8-eth4:s9-eth5
s9 lo: s9-eth1:s1-eth5 s9-eth2:s2-eth4 s9-eth3:s6-eth4 s9-eth4:s7-eth5 s9-eth5:s8-eth4 s9-eth6:s10-eth4 s9-eth7:s11-eth4
s10 lo: s10-eth1:s2-eth5 s10-eth2:s3-eth4 s10-eth3:s4-eth4 s10-eth4:s9-eth6 s10-eth5:s11-eth5
s11 lo: s11-eth1:s4-eth5 s11-eth2:s5-eth4 s11-eth3:s6-eth5 s11-eth4:s9-eth7 s11-eth5:s10-eth5
c0
mininet>
```

1. میزبان h1 با استفاده از رابط شبکه h1-eth0 به سوئیچ S1 از طریق رابط s1-eth1 متصل می شود.
2. میزبان h2 با استفاده از رابط شبکه h2-eth0 به سوئیچ S1 از طریق رابط s1-eth2 متصل می شود.
3. به همین ترتیب ارتباط همه ی هاست ها با سوئیچ ها را نشان میدهد.
4. سوئیچ s1
 - a. دارای یک رابط Loopback lo.

- b. از طریق رابط s1-eth1 به h1-eth0 متصل می شود.
- c. از طریق رابط s1-eth2 به h2-eth0 متصل می شود.
- d. از طریق رابط s1-eth3 به S2-eth2 متصل می شود.
- e. از طریق رابط s1-eth4 به S8-eth2 متصل می شود.
- f. از طریق رابط s1-eth5 به S9-eth1 متصل می شود.

و به همین ترتیب همه ی ارتباط ها مشخص میشود.

خروجی dump

Display all the information about the nodes

هر هاست را به همراه ایپی اختصاص یافته و ایدی پروسس و رابط ان نشان میدهد.

نمایش اطلاعات سویچ مثل داشتن یک loopback interface و ایدی پروسس.

ایپی و ایدی پروسس و پورت اجرایی کنترلر را هم نشان میدهد.

```
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=2782>
<Host h2: h2-eth0:10.0.0.2 pid=2784>
<Host h3: h3-eth0:10.0.0.3 pid=2786>
<Host h4: h4-eth0:10.0.0.4 pid=2788>
<Host h5: h5-eth0:10.0.0.5 pid=2790>
<Host h6: h6-eth0:10.0.0.6 pid=2792>
<Host h7: h7-eth0:10.0.0.7 pid=2794>
<Host h8: h8-eth0:10.0.0.8 pid=2796>
<Host h9: h9-eth0:10.0.0.9 pid=2798>
<Host h10: h10-eth0:10.0.0.10 pid=2800>
<Host h11: h11-eth0:10.0.0.11 pid=2802>
<Host h12: h12-eth0:10.0.0.12 pid=2804>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None,s1-eth4:None,s1-eth5:None pid=2809>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None,s2-eth3:None,s2-eth4:None,s2-eth5:None pid=2812>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None,s3-eth3:None,s3-eth4:None,s3-eth5:None pid=2815>
<OVSSwitch s4: lo:127.0.0.1,s4-eth1:None,s4-eth2:None,s4-eth3:None,s4-eth4:None,s4-eth5:None pid=2818>
<OVSSwitch s5: lo:127.0.0.1,s5-eth1:None,s5-eth2:None,s5-eth3:None,s5-eth4:None,s5-eth5:None pid=2821>
<OVSSwitch s6: lo:127.0.0.1,s6-eth1:None,s6-eth2:None,s6-eth3:None,s6-eth4:None,s6-eth5:None pid=2824>
<OVSSwitch s7: lo:127.0.0.1,s7-eth1:None,s7-eth2:None,s7-eth3:None,s7-eth4:None,s7-eth5:None pid=2827>
<OVSSwitch s8: lo:127.0.0.1,s8-eth1:None,s8-eth2:None,s8-eth3:None,s8-eth4:None pid=2830>
<OVSSwitch s9: lo:127.0.0.1,s9-eth1:None,s9-eth2:None,s9-eth3:None,s9-eth4:None,s9-eth5:None,s9-eth6
```

```

<OVSSwitch s8: lo:127.0.0.1,s8-eth1:None,s8-eth2:None,s8-eth3:None,s8-eth4:None pid=2830>
<OVSSwitch s9: lo:127.0.0.1,s9-eth1:None,s9-eth2:None,s9-eth3:None,s9-eth4:None,s9-eth5:None,s9-eth6:None,s9-eth7:None pid=2833>
<OVSSwitch s10: lo:127.0.0.1,s10-eth1:None,s10-eth2:None,s10-eth3:None,s10-eth4:None,s10-eth5:None pid=2836>
<OVSSwitch s11: lo:127.0.0.1,s11-eth1:None,s11-eth2:None,s11-eth3:None,s11-eth4:None,s11-eth5:None pid=2839>
<RemoteController{'ip': '127.0.0.1', 'port': 6653} c0: 127.0.0.1:6653 pid=2775>
mininet> links
h1-eth0<->s1-eth1 (OK OK)
h2-eth0<->s1-eth2 (OK OK)
h3-eth0<->s2-eth1 (OK OK)
h4-eth0<->s3-eth1 (OK OK)
h5-eth0<->s3-eth2 (OK OK)
h6-eth0<->s4-eth1 (OK OK)
h7-eth0<->s5-eth1 (OK OK)
h8-eth0<->s5-eth2 (OK OK)
h9-eth0<->s6-eth1 (OK OK)
h10-eth0<->s7-eth1 (OK OK)
h11-eth0<->s7-eth2 (OK OK)
h12-eth0<->s8-eth1 (OK OK)
s1-eth3<->s2-eth2 (OK OK)
s1-eth4<->s8-eth2 (OK OK)
s1-eth5<->s9-eth1 (OK OK)
s2-eth3<->s3-eth3 (OK OK)
s2-eth4<->s9-eth2 (OK OK)
s2-eth5<->s10-eth1 (OK OK)
s3-eth5<->s4-eth2 (OK OK)
s3-eth4<->s10-eth2 (OK OK)
s4-eth3<->s5-eth3 (OK OK)

```

```

h5-eth0<->s3-eth2 (OK OK)
h6-eth0<->s4-eth1 (OK OK)
h7-eth0<->s5-eth1 (OK OK)
h8-eth0<->s5-eth2 (OK OK)
h9-eth0<->s6-eth1 (OK OK)
h10-eth0<->s7-eth1 (OK OK)
h11-eth0<->s7-eth2 (OK OK)
h12-eth0<->s8-eth1 (OK OK)
s1-eth3<->s2-eth2 (OK OK)
s1-eth4<->s8-eth2 (OK OK)
s1-eth5<->s9-eth1 (OK OK)
s2-eth3<->s3-eth3 (OK OK)
s2-eth4<->s9-eth2 (OK OK)
s2-eth5<->s10-eth1 (OK OK)
s3-eth5<->s4-eth2 (OK OK)
s3-eth4<->s10-eth2 (OK OK)
s4-eth3<->s5-eth3 (OK OK)
s4-eth4<->s10-eth3 (OK OK)
s4-eth5<->s11-eth1 (OK OK)
s5-eth5<->s6-eth2 (OK OK)
s5-eth4<->s11-eth2 (OK OK)
s6-eth3<->s7-eth3 (OK OK)
s6-eth4<->s9-eth3 (OK OK)
s6-eth5<->s11-eth3 (OK OK)
s7-eth4<->s8-eth3 (OK OK)
s7-eth5<->s9-eth4 (OK OK)
s8-eth4<->s9-eth5 (OK OK)
s9-eth6<->s10-eth4 (OK OK)
s9-eth7<->s11-eth4 (OK OK)
s10-eth5<->s11-eth5 (OK OK)
mininet>

```

```

root@ubuntu:~/Downloads/python_examples# iperf -s -p 5566 -i 1
-----
Server listening on TCP port 5566
TCP window size: 85.3 KByte (default)
-----
[ 54] local 10.0.0.4 port 5566 connected with 10.0.0.1 port 51074
[ ID] Interval      Transfer    Bandwidth
[ 54] 0.0- 1.0 sec  1.45 GBytes 12.5 Gbits/sec
[ 54] 1.0- 2.0 sec  1.44 GBytes 12.3 Gbits/sec
[ 54] 2.0- 3.0 sec  1.44 GBytes 12.4 Gbits/sec
[ 54] 3.0- 4.0 sec  1.42 GBytes 12.2 Gbits/sec
[ 54] 4.0- 5.0 sec  1.44 GBytes 12.4 Gbits/sec
[ 54] 5.0- 6.0 sec  1.35 GBytes 11.6 Gbits/sec
[ 54] 6.0- 7.0 sec  1.46 GBytes 12.5 Gbits/sec
[ 54] 7.0- 8.0 sec  1.39 GBytes 11.9 Gbits/sec
[ 54] 8.0- 9.0 sec  1.50 GBytes 12.9 Gbits/sec
[ 54] 9.0-10.0 sec  1.66 GBytes 14.3 Gbits/sec
[ 54] 10.0-11.0 sec 1.37 GBytes 11.8 Gbits/sec
[ 54] 11.0-12.0 sec 1.41 GBytes 12.1 Gbits/sec
[ 54] 12.0-13.0 sec 1.59 GBytes 13.6 Gbits/sec
[ 54] 13.0-14.0 sec 1.36 GBytes 11.7 Gbits/sec
[ 54] 0.0-15.0 sec 21.7 GBytes 12.5 Gbits/sec
[  0]

```

```

root@ubuntu:~/Downloads/python_examples# iperf -c 10.0.0.4 -p 5566 -t 15
-----
Client connecting to 10.0.0.4, TCP port 5566
TCP window size: 85.3 KByte (default)
-----
[ 53] local 10.0.0.1 port 51074 connected with 10.0.0.4 port 5566
[ ID] Interval      Transfer    Bandwidth
[ 53] 0.0-15.0 sec  21.7 GBytes 12.4 Gbits/sec
root@ubuntu:~/Downloads/python_examples#

```

`iperf -s -p 5566 -i 1`

Start the TCP server (-s) at h4 with port 5566 (-p). Also, monitor the results every one second (-i). the default setting is using TCP.

`iperf -c 10.0.0.4 -p 5566 -t 15`

Start the TCP client (-c) at h1. Also, set the transmission duration (-t) to 15 seconds. Note: after -c, you need to specify the server ip address which is 10.0.0.4.

نتایج

در h1 (کلاینت):

از ثانیه ی صفر تا ۱۵ میانگین گذردهی 12.4Gbps بوده است.

در h4 (سرور):

Throughput را در هرثانیه نشان میدهد.

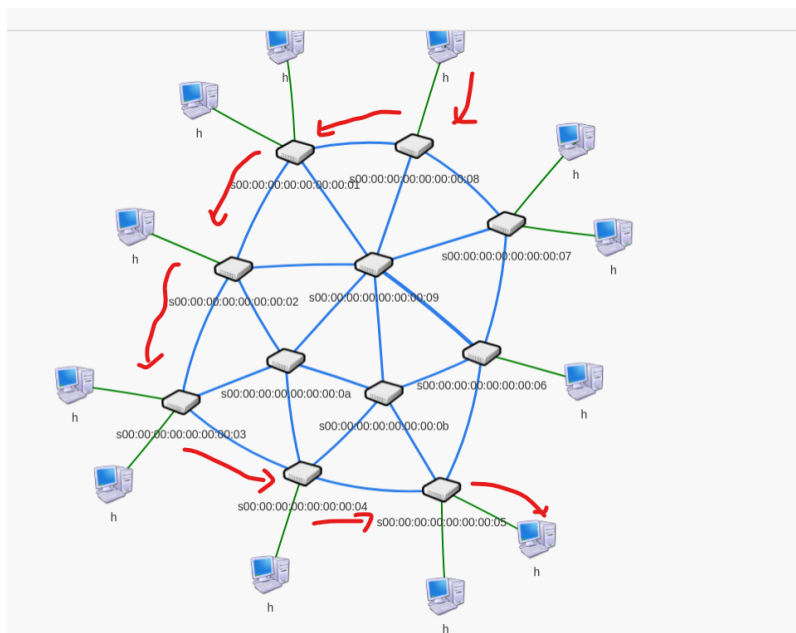
همچنین چون از TCP استفاده میشود یعنی پکت ها به صورت پنجره پنجره منتقل میشوند و در اینجا هم سائز پنجره مشخص شده که 85.3kbyte است.

سوال 6

مسیرهای دلخواه تعریف شده

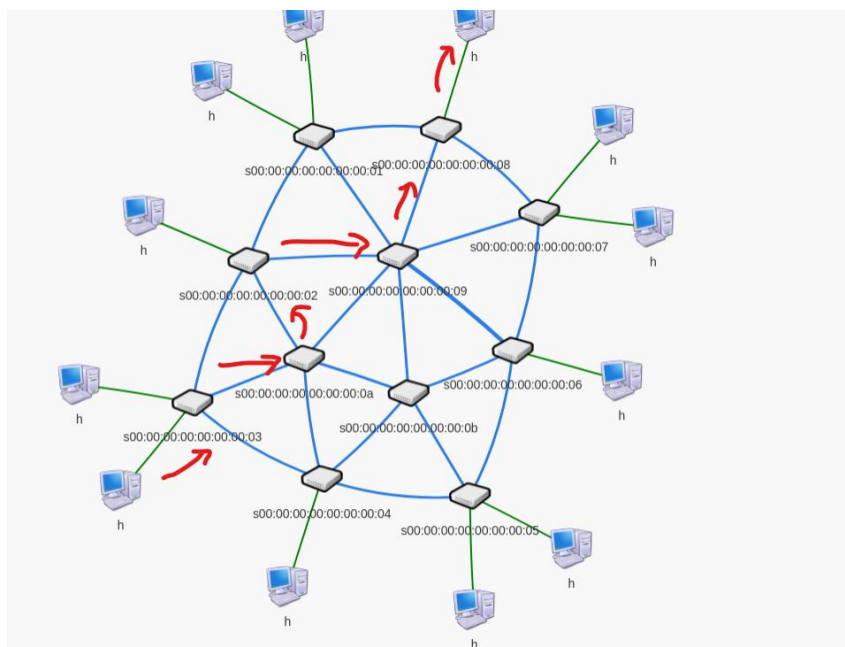
مسیر 1:

path1 = h12->s8->s1->s2->s3->s4->s5->h8



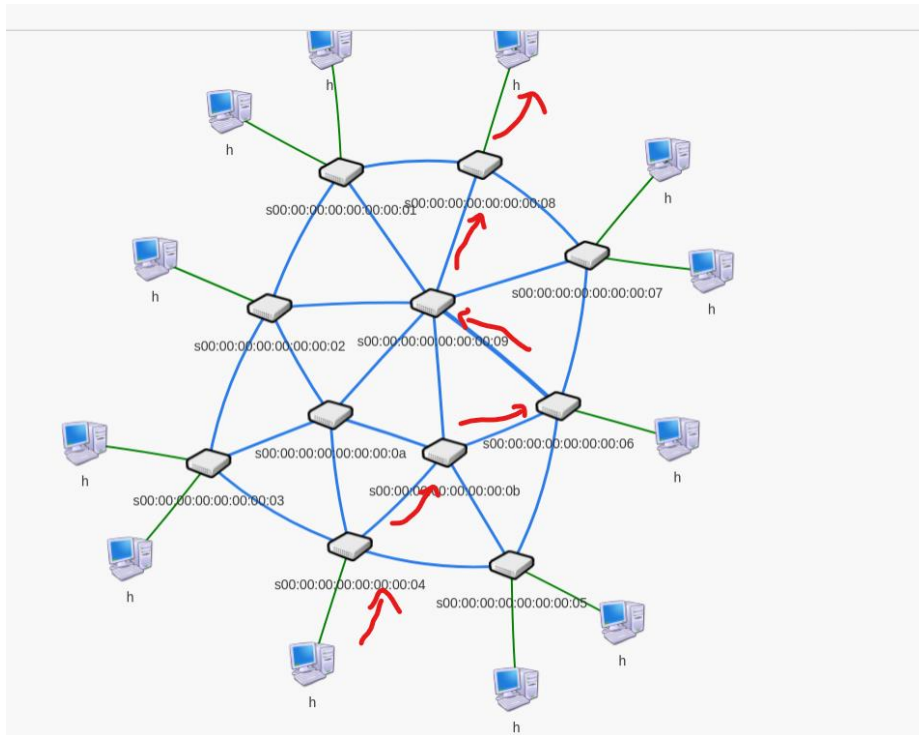
مسیر 2:

path2 = h5->s3->s10->s2->s9->s1->s8->s7->h10



مسیر 3:

path3 = h6->s4->s0b->s6->s9->s8->h12



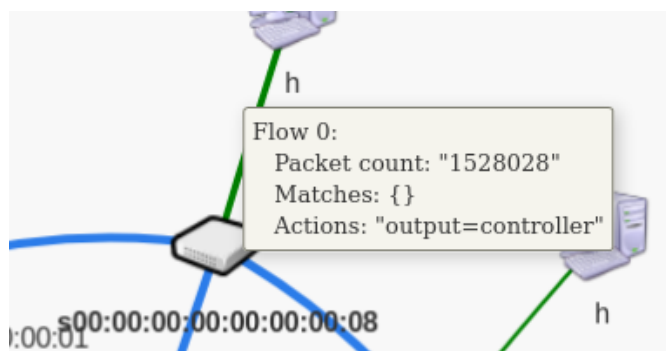
سوال 7

فایل های flow را باید در ترمینال اجرا کنیم. (هر فایل برای یک مسیر دلخواه است). فایل ها را در فولدر جواب ها قرار دادم.

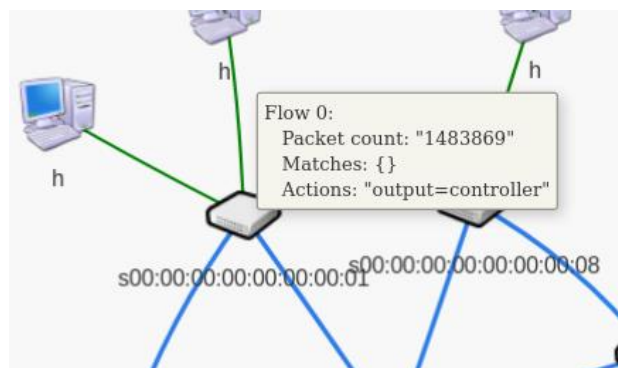
توضیحات مربوط به مسیر path1

قبل از اجرای فایل پایتون flow1

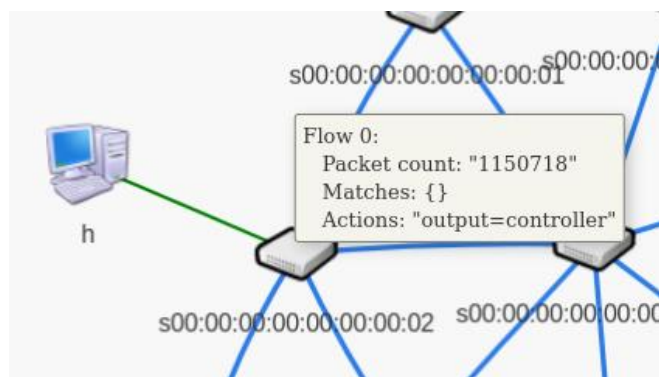
وضعیت سوئیچ ۸



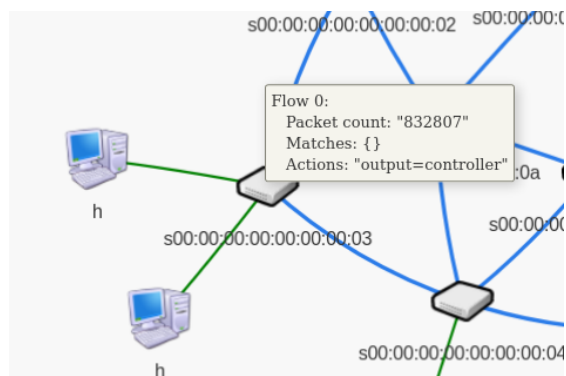
وضعیت سویچ ۱



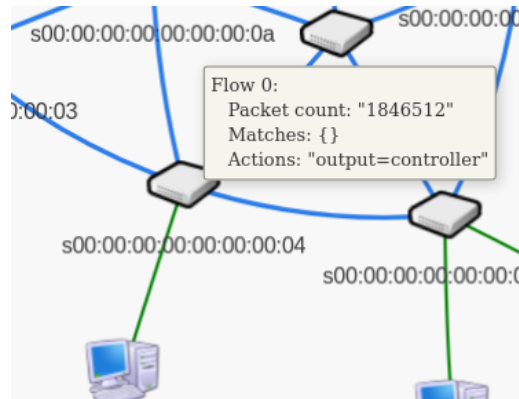
وضعیت سویچ ۲



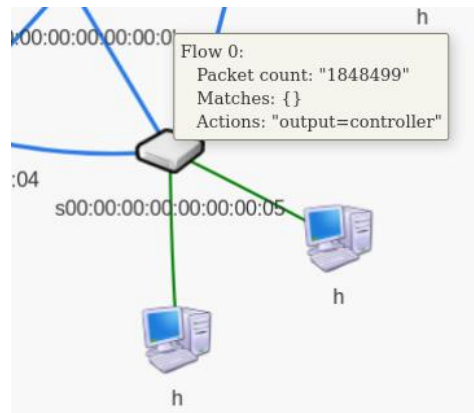
وضعیت سویچ ۳



وضعیت سویچ ۴



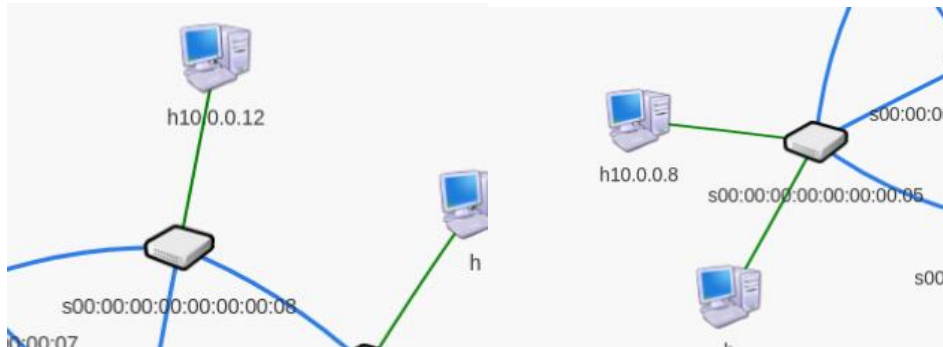
وضعیت سویچ ۵



بعد از اجرای فایل پایتون flow1

[illegible]

همان طور که مشخص است دو هاست که در path1 مشخص کردیم در توپولوژی ایبی گرفتند.



سپس باید دو هاست مبدا و مقصد را پینگ کنیم:

```
mininet>
mininet> h12 ping h8
PING 10.0.0.8 (10.0.0.8) 56(84) bytes of data.
64 bytes from 10.0.0.8: icmp_seq=1 ttl=64 time=56.9 ms
64 bytes from 10.0.0.8: icmp_seq=2 ttl=64 time=0.332 ms
64 bytes from 10.0.0.8: icmp_seq=3 ttl=64 time=0.066 ms
64 bytes from 10.0.0.8: icmp_seq=4 ttl=64 time=0.045 ms
64 bytes from 10.0.0.8: icmp_seq=5 ttl=64 time=0.068 ms
64 bytes from 10.0.0.8: icmp_seq=6 ttl=64 time=0.048 ms
64 bytes from 10.0.0.8: icmp_seq=7 ttl=64 time=0.088 ms
64 bytes from 10.0.0.8: icmp_seq=8 ttl=64 time=0.044 ms
64 bytes from 10.0.0.8: icmp_seq=9 ttl=64 time=0.047 ms
64 bytes from 10.0.0.8: icmp_seq=10 ttl=64 time=0.061 ms
64 bytes from 10.0.0.8: icmp_seq=11 ttl=64 time=0.053 ms
64 bytes from 10.0.0.8: icmp_seq=12 ttl=64 time=0.047 ms
64 bytes from 10.0.0.8: icmp_seq=13 ttl=64 time=0.055 ms
64 bytes from 10.0.0.8: icmp_seq=14 ttl=64 time=0.050 ms
64 bytes from 10.0.0.8: icmp_seq=15 ttl=64 time=0.036 ms
64 bytes from 10.0.0.8: icmp_seq=16 ttl=64 time=0.064 ms
64 bytes from 10.0.0.8: icmp_seq=17 ttl=64 time=0.066 ms
```

وضعیت سویچ ۸

چون داریم پینگ میکنیم یعنی پروتوکل icmp از سمت فرستنده یک request به مقصد میفرستد و مقصد هم باید reply کند پس دوتا flow ایجاد میشود که مبدا در اولی 10.0.0.12 است و مقصد 10.0.0.8 است. ولی در دومی برعکس میشود.

همان طور که در شکل مشخص است پورت های ورودی و خروجی هم نشان داده شده از انجایی که هر flow table شامل دو بخش match,action است در هر flow این دومورد نشان داده شده.

مثلا در flow0: عملی که انجام میشود خروج از پورت ۲ است. تعداد پکت های ارسالی از فرستنده به گیرنده هم مشخص شده که این تعداد برابر با تعداد پکت ها در flow1 است که جای مبدا و مقصد عوض شده است.

Eth_dest,eth_src هم ادرس های MAC هاست های مبدا و مقصد هستند.وضعیت بقیه سویچ ها هم به همین صورت بررسی میشود.

```
Flow 0:
  Packet count: "244"
  Matches: {"in_port": "1", "eth_type": "0x800", "ipv4_src": "10.0.0.12", "ipv4_dst": "10.0.0.8"}
  Actions: "output=2"
Flow 1:
  Packet count: "244"
  Matches: {"in_port": "3", "eth_dst": "ce:69:27:b7:11:5c", "eth_src": "da:f1:2b:fe:9e:0b", "eth_type": "0x800", "ipv4_src": "10.0.0.8", "ipv4_dst": "10.0.0.12"}
  Actions: "output=1"
Flow 2:
  Packet count: "1"
  Matches: {"in_port": "4", "eth_dst": "ce:69:27:b7:11:5c", "eth_src": "da:f1:2b:fe:9e:0b", "eth_type": "0x806"}
  Actions: "output=1"
Flow 3:
  Packet count: "0"
  Matches: {"in_port": "1", "eth_dst": "da:f1:2b:fe:9e:0b", "eth_src": "ce:69:27:b7:11:5c", "eth_type": "0x806"}
  Actions: "output=4"
Flow 4:
  Packet count: "1528166"
  Matches: {}
  Actions: "output=controller"
```

وضعیت سویچ ۱

```
Flow 0:
  Packet count: "632"
  Matches: {"in_port": "4", "eth_type": "0x800", "ipv4_src": "10.0.0.12", "ipv4_dst": "10.0.0.8"}
  Actions: "output=3"
Flow 1:
  Packet count: "1484088"
  Matches: {}
  Actions: "output=controller"
```

وضعیت سویچ ۲

```
Flow 0:
  Packet count: "632"
  Matches: {"in_port": "2", "eth_type": "0x800", "ipv4_src": "10.0.0.12", "ipv4_dst": "10.0.0.8"}
  Actions: "output=3"
Flow 1:
  Packet count: "1750054"
  Matches: {}
  Actions: "output=controller"
```


وضعیت سویچ ۳

```
Flow 0:
  Packet count: "632"
  Matches: {"in_port": "3", "eth_type": "0x800", "ipv4_src": "10.0.0.12", "ipv4_dst": "10.0.0.8"}
  Actions: "output=5"
Flow 1:
  Packet count: "1547407"
  Matches: {}
  Actions: "output=controller"
```

وضعیت سویچ ۴

```
Flow 0:
  Packet count: "632"
  Matches: {"in_port": "2", "eth_type": "0x800", "ipv4_src": "10.0.0.12", "ipv4_dst": "10.0.0.8"}
  Actions: "output=3"
Flow 1:
  Packet count: "1847336"
  Matches: {}
  Actions: "output=controller"
```

وضعیت سویچ ۵

```
Flow 0:
  Packet count: "233"
  Matches: {"in_port": "3", "eth_type": "0x800", "ipv4_src": "10.0.0.12", "ipv4_dst": "10.0.0.8"}
  Actions: "output=2"
Flow 1:
  Packet count: "232"
  Matches: {"in_port": "2", "eth_dst": "ce:69:27:b7:11:5c", "eth_src": "da:f1:2b:fe:9e:0b", "eth_type": "0x800", "ipv4_src": "10.0.0.8", "ipv4_dst": "10.0.0.12"}
  Actions: "output=5"
Flow 2:
  Packet count: "1848614"
  Matches: {}
  Actions: "output=controller"
```

مشاهده ی اطلاعات در سوییچ ها

سوییچ 8 (هاست مبدا به آن متصل است)

i Flow Table									
Show 10 entries		Search:							
Table No.	Bkt Count	Byte	Duration(s)	Priority	IdleTimeoutSec	HardTimeoutSec	Flags	Instructions	
0x0	403	39494	515	32768	0	0		SEND_FLOW_REM	output=2
0x0	403	39494	412	1	5	0			output=1
0x0	1528203	309803767	11645	0	0	0			output=controller

Showing 1 to 3 of 3 entries

Previous 1 Next

سوییچ 5 (هاست مقصد به آن متصل است)

i Flow Table									
Show 10 entries		Search:							
Table No.	Bkt Count	Byte	Duration(s)	Priority	IdleTimeoutSec	HardTimeoutSec	Flags	Instructions	
0x0	487	47726	601	32768	0	0		SEND_FLOW_REM	output=2
0x0	486	47628	498	1	5	0			output=5
0x0	1848676	374855596	11732	0	0	0			output=controller

Showing 1 to 3 of 3 entries

Previous 1 Next

همان طور که در اطلاعات سوییچ ها مشاهده میکنیم، در سطر اول entry ای قرار میگیرد که بالاترین اولویت را دارد که همانی است که ما تعریفش کردیم. همان طور که مشخص است تعداد پکت های ارسالی در سوییچ ۸ و تعداد پکت های دریافتی در سوییچ ۵ افزایش یافته اند زیرا ما بین دو هاستی که به این سوییچ ها وصل بودند ارتباط ایجاد کردیم (h12 ping h8) و در آخرین ستون هردو جدول هم پورت خروجی یا همان action ای که باید انجام شود را نشان میدهد.