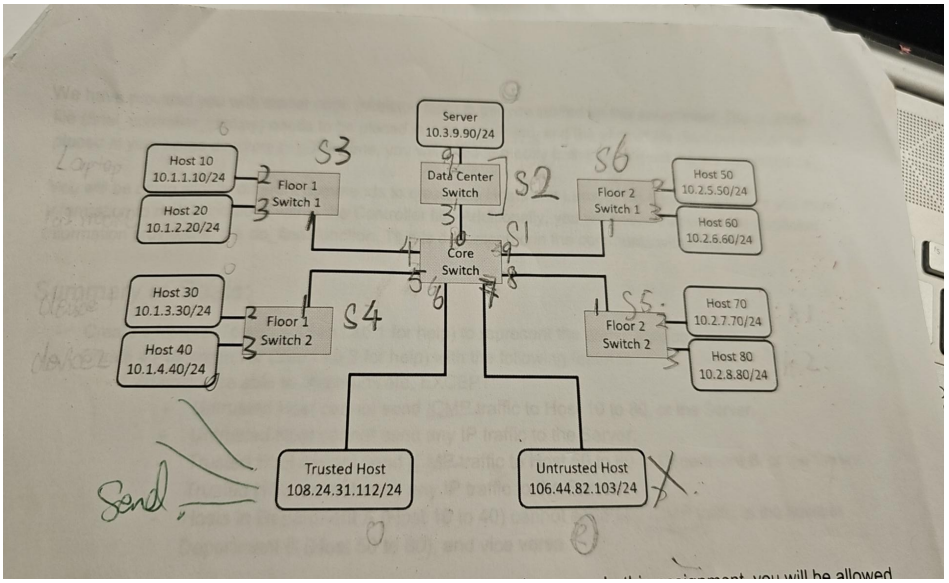
First of all, to test the network topology. I ran links within the mininet once I finished the set up of the topology, by connecting all the hosts and switches then I ran the “**links**” command to check to make sure they are linked correctly, and the result is shown below.

```
mininet@minine
File  Edit  Tabs  Help
mininet> links
s1-eth10<->s2-eth3 (OK OK)
s1-eth4<->s3-eth1 (OK OK)
s1-eth5<->s4-eth1 (OK OK)
s1-eth8<->s5-eth1 (OK OK)
s1-eth9<->s6-eth1 (OK OK)
s1-eth6<->trust-eth0 (OK OK)
s1-eth7<->untrust-eth0 (OK OK)
s2-eth9<->server-eth0 (OK OK)
s3-eth2<->h10-eth0 (OK OK)
s3-eth3<->h20-eth0 (OK OK)
s4-eth2<->h30-eth0 (OK OK)
s4-eth3<->h40-eth0 (OK OK)
s5-eth2<->h70-eth0 (OK OK)
s5-eth3<->h80-eth0 (OK OK)
s6-eth2<->h50-eth0 (OK OK)
s6-eth3<->h60-eth0 (OK OK)
mininet>
```



I created the topology based on the figure I labeled above. Upon closer examination, we can see that on the 1st line of the screenshot above, s1 which in my case is representing the core switch is connected to s2 (Data Center Switch) successfully. Besides, on the second line stated s1(core switch) is connected to s3(floor 1 switch 1), and the list goes on as follows. With that being said, we can conclude that the network’s topology is configured successfully as intended.

After that I test the “icmp” reachability using the pingall command. This test sends network ping requests to all connected devices to check their connectivity and response times. Most importantly to test out all the forwarding rules written on the lab specifications listed below

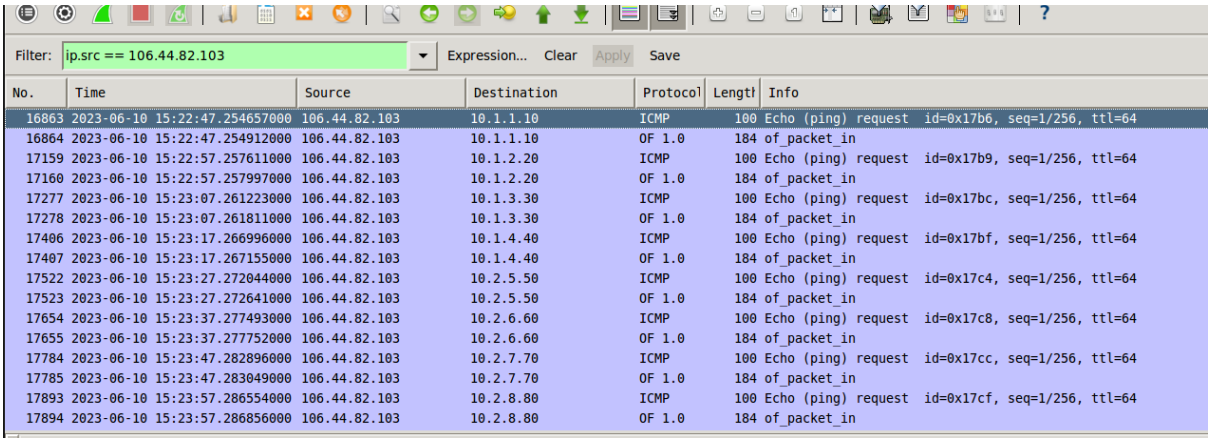
- Create a Mininet Topology (See Lab 1 for help) to represent the above topology.
- Create a Pox controller (See Lab 3 for help) with the following features:
    - All hosts are able to communicate, EXCEPT:
        - Untrusted Host cannot send ICMP traffic to Host 10 to 80, or the Server.
        - Untrusted Host cannot send any IP traffic to the Server.
        - Trusted Host cannot send ICMP traffic to Host 50 to 80 in Department B, or the Server.
        - Trusted Host cannot send any IP traffic to the Server.
        - Hosts in Department A (Host 10 to 40) cannot send any ICMP traffic to the hosts in Department B (Host 50 to 80), and vice versa.

By running the pingall command, the icmp echo request is sending icmp packets to every host in the topology.

```
File  Edit  Tabs  Help
s5-eth3<->h80-eth0 (OK OK)
s6-eth2<->h50-eth0 (OK OK)
s6-eth3<->h60-eth0 (OK OK)
mininet> pingall
*** Ping: testing ping reachability
h10 -> h20 h30 h40 X X X X server trust X
h20 -> h10 h30 h40 X X X X server trust X
h30 -> h10 h20 h40 X X X X server trust X
h40 -> h10 h20 h30 X X X X server trust X
h50 -> X X X X h60 h70 h80 server X X
h60 -> X X X X h50 h70 h80 server X X
h70 -> X X X X h50 h60 h80 server X X
h80 -> X X X X h50 h60 h70 server X X
server -> h10 h20 h30 h40 h50 h60 h70 h80 X X
trust -> h10 h20 h30 h40 X X X X X untrust
untrust -> X X X X X X X X X trust
*** Results: 54% dropped (50/110 received)
mininet>
```

An “x” on the pingall results tells you a packet has been blocked by a specific host and that is aligned with the sets of rules. Let’s take a look at the first line of the result (host 10), we can see there are four “x” illustrated within the spot of h50 h60 h70 h80 (host in B), and a “x” for the untrust server. This validates the rules implemented in the firewall have been configured accurately – Host in A can’t send ICMP traffic to Host in B as well as untrusted host can not send icmp traffic to host 10 to 80.

To be more rigorous, I opened wireshark

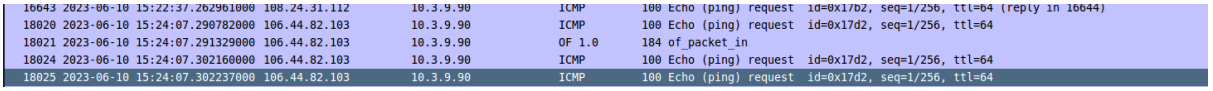


Filter: ip.src == 106.44.82.103 | Expression... | Clear | Apply | Save

No.	Time	Source	Destination	Protocol	Length	Info
16863	2023-06-10 15:22:47.254657000	106.44.82.103	10.1.1.10	ICMP	100	Echo (ping) request id=0x17b6, seq=1/256, ttl=64
16864	2023-06-10 15:22:47.254912000	106.44.82.103	10.1.1.10	OF 1.0	184	of_packet_in
17159	2023-06-10 15:22:57.257611000	106.44.82.103	10.1.2.20	ICMP	100	Echo (ping) request id=0x17b9, seq=1/256, ttl=64
17160	2023-06-10 15:22:57.257997000	106.44.82.103	10.1.2.20	OF 1.0	184	of_packet_in
17277	2023-06-10 15:23:07.261223000	106.44.82.103	10.1.3.30	ICMP	100	Echo (ping) request id=0x17bc, seq=1/256, ttl=64
17278	2023-06-10 15:23:07.261811000	106.44.82.103	10.1.3.30	OF 1.0	184	of_packet_in
17406	2023-06-10 15:23:17.266996000	106.44.82.103	10.1.4.40	ICMP	100	Echo (ping) request id=0x17bf, seq=1/256, ttl=64
17407	2023-06-10 15:23:17.267155000	106.44.82.103	10.1.4.40	OF 1.0	184	of_packet_in
17522	2023-06-10 15:23:27.272044000	106.44.82.103	10.2.5.50	ICMP	100	Echo (ping) request id=0x17c4, seq=1/256, ttl=64
17523	2023-06-10 15:23:27.272641000	106.44.82.103	10.2.5.50	OF 1.0	184	of_packet_in
17654	2023-06-10 15:23:37.277493000	106.44.82.103	10.2.6.60	ICMP	100	Echo (ping) request id=0x17c8, seq=1/256, ttl=64
17655	2023-06-10 15:23:37.277752000	106.44.82.103	10.2.6.60	OF 1.0	184	of_packet_in
17784	2023-06-10 15:23:47.282896000	106.44.82.103	10.2.7.70	ICMP	100	Echo (ping) request id=0x17cc, seq=1/256, ttl=64
17785	2023-06-10 15:23:47.283049000	106.44.82.103	10.2.7.70	OF 1.0	184	of_packet_in
17893	2023-06-10 15:23:57.286554000	106.44.82.103	10.2.8.80	ICMP	100	Echo (ping) request id=0x17cf, seq=1/256, ttl=64
17894	2023-06-10 15:23:57.286856000	106.44.82.103	10.2.8.80	OF 1.0	184	of_packet_in

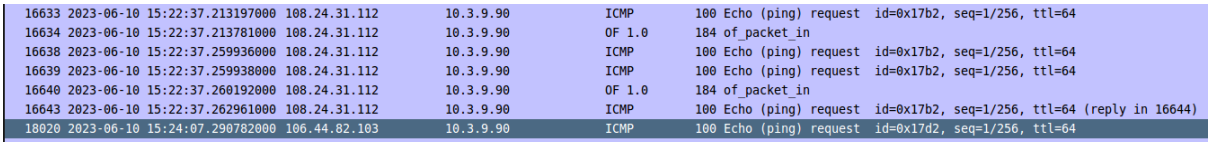
From the screenshot above we can see that the packet being sent from 106.44.82.103 (untrusted host) only has echo requests but no echo replies. According to the forwarding rules "Untrusted Host cannot send ICMP traffic to Host 10 to 80", those packets sent from the untrusted host successfully sent a request to the h10 to h80, but it did not receive a response back from host10 to 80. This could indicate the firewall is blocking ICMP traffic, and those packets will be dropped.

Let's move on to the next checkpoint – "Untrusted/Trust Host cannot send any traffic to Server"



16643	2023-06-10 15:22:37.262961000	108.24.31.112	10.3.9.90	ICMP	100	Echo (ping) request id=0x17b2, seq=1/256, ttl=64 (reply in 16644)
18020	2023-06-10 15:24:07.290782000	106.44.82.103	10.3.9.90	ICMP	100	Echo (ping) request id=0x17d2, seq=1/256, ttl=64
18021	2023-06-10 15:24:07.291329000	106.44.82.103	10.3.9.90	OF 1.0	184	of_packet_in
18024	2023-06-10 15:24:07.302160000	106.44.82.103	10.3.9.90	ICMP	100	Echo (ping) request id=0x17d2, seq=1/256, ttl=64
18025	2023-06-10 15:24:07.302237000	106.44.82.103	10.3.9.90	ICMP	100	Echo (ping) request id=0x17d2, seq=1/256, ttl=64

(untrusted host cannot send any traffic to server)

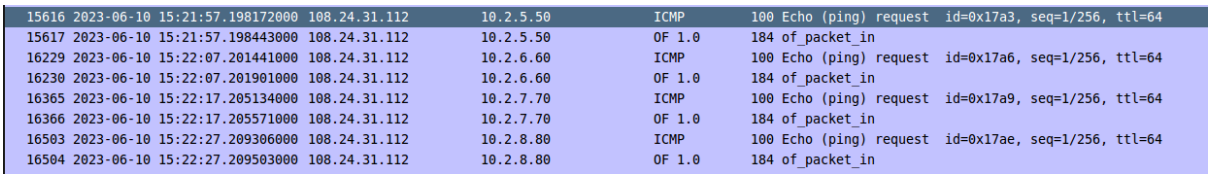


16633	2023-06-10 15:22:37.213197000	108.24.31.112	10.3.9.90	ICMP	100	Echo (ping) request id=0x17b2, seq=1/256, ttl=64
16634	2023-06-10 15:22:37.213781000	108.24.31.112	10.3.9.90	OF 1.0	184	of_packet_in
16638	2023-06-10 15:22:37.259936000	108.24.31.112	10.3.9.90	ICMP	100	Echo (ping) request id=0x17b2, seq=1/256, ttl=64
16639	2023-06-10 15:22:37.259938000	108.24.31.112	10.3.9.90	ICMP	100	Echo (ping) request id=0x17b2, seq=1/256, ttl=64
16640	2023-06-10 15:22:37.260192000	108.24.31.112	10.3.9.90	OF 1.0	184	of_packet_in
16643	2023-06-10 15:22:37.262961000	108.24.31.112	10.3.9.90	ICMP	100	Echo (ping) request id=0x17b2, seq=1/256, ttl=64 (reply in 16644)
18020	2023-06-10 15:24:07.290782000	106.44.82.103	10.3.9.90	ICMP	100	Echo (ping) request id=0x17d2, seq=1/256, ttl=64

(trusted host cannot send any traffic to server)

Similar to logics above we are only seeing echo replies but no echo reply from both untrusted/trust hosts as server as the destination, therefore we can conclude the traffic is being blocked by the firewall and the packet going to the server from the untrust/trust host will be dropped.

Go on to the next "Trusted Host cannot send ICMP traffic to Host 50 to 80"



15616	2023-06-10 15:21:57.198172000	108.24.31.112	10.2.5.50	ICMP	100	Echo (ping) request id=0x17a3, seq=1/256, ttl=64
15617	2023-06-10 15:21:57.198443000	108.24.31.112	10.2.5.50	OF 1.0	184	of_packet_in
16229	2023-06-10 15:22:07.201441000	108.24.31.112	10.2.6.60	ICMP	100	Echo (ping) request id=0x17a6, seq=1/256, ttl=64
16230	2023-06-10 15:22:07.201901000	108.24.31.112	10.2.6.60	OF 1.0	184	of_packet_in
16365	2023-06-10 15:22:17.205134000	108.24.31.112	10.2.7.70	ICMP	100	Echo (ping) request id=0x17a9, seq=1/256, ttl=64
16366	2023-06-10 15:22:17.205571000	108.24.31.112	10.2.7.70	OF 1.0	184	of_packet_in
16503	2023-06-10 15:22:27.209306000	108.24.31.112	10.2.8.80	ICMP	100	Echo (ping) request id=0x17ae, seq=1/256, ttl=64
16504	2023-06-10 15:22:27.209503000	108.24.31.112	10.2.8.80	OF 1.0	184	of_packet_in

Same reasoning as above, there are only echo requests going between the trusted host and the host in department B (h50 to 80). Indicating the firewall is blocking ICMP traffic, and those packets will be dropped.

Finally let's test Host 10 to 40 cannot send ICMP traffic to Host 50 to 80

97	2023-06-10 15:14:15.932444000	10.1.1.10	10.2.5.50	OF 1.0	184	of_packet_in
99	2023-06-10 15:14:15.933287000	10.1.1.10	10.2.5.50	ICMP	100	Echo (ping) request id=0x16d1, seq=1/256, ttl=64
100	2023-06-10 15:14:15.933322000	10.1.1.10	10.2.5.50	ICMP	100	Echo (ping) request id=0x16d1, seq=1/256, ttl=64
101	2023-06-10 15:14:15.933574000	10.1.1.10	10.2.5.50	OF 1.0	184	of_packet_in
642	2023-06-10 15:14:25.937640000	10.1.1.10	10.2.6.60	ICMP	100	Echo (ping) request id=0x16d5, seq=1/256, ttl=64
643	2023-06-10 15:14:25.937839000	10.1.1.10	10.2.6.60	OF 1.0	184	of_packet_in
647	2023-06-10 15:14:25.988467000	10.1.1.10	10.2.6.60	ICMP	100	Echo (ping) request id=0x16d5, seq=1/256, ttl=64
648	2023-06-10 15:14:25.988469000	10.1.1.10	10.2.6.60	ICMP	100	Echo (ping) request id=0x16d5, seq=1/256, ttl=64
649	2023-06-10 15:14:25.988813000	10.1.1.10	10.2.6.60	OF 1.0	184	of_packet_in
779	2023-06-10 15:14:35.942618000	10.1.1.10	10.2.7.70	ICMP	100	Echo (ping) request id=0x16da, seq=1/256, ttl=64
780	2023-06-10 15:14:35.942759000	10.1.1.10	10.2.7.70	OF 1.0	184	of_packet_in
783	2023-06-10 15:14:35.965991000	10.1.1.10	10.2.7.70	ICMP	100	Echo (ping) request id=0x16da, seq=1/256, ttl=64
784	2023-06-10 15:14:35.965994000	10.1.1.10	10.2.7.70	ICMP	100	Echo (ping) request id=0x16da, seq=1/256, ttl=64
785	2023-06-10 15:14:35.966313000	10.1.1.10	10.2.7.70	OF 1.0	184	of_packet_in
914	2023-06-10 15:14:45.948511000	10.1.1.10	10.2.8.80	ICMP	100	Echo (ping) request id=0x16dd, seq=1/256, ttl=64
915	2023-06-10 15:14:45.949024000	10.1.1.10	10.2.8.80	OF 1.0	184	of_packet_in
919	2023-06-10 15:14:45.993814000	10.1.1.10	10.2.8.80	ICMP	100	Echo (ping) request id=0x16dd, seq=1/256, ttl=64
920	2023-06-10 15:14:45.993901000	10.1.1.10	10.2.8.80	ICMP	100	Echo (ping) request id=0x16dd, seq=1/256, ttl=64
921	2023-06-10 15:14:45.994132000	10.1.1.10	10.2.8.80	OF 1.0	184	of_packet_in

(host 10 can not send to h50 to 80)

1537	2023-06-10 15:15:06.094854000	10.1.2.20	10.1.4.40	ICMP	100	Echo (ping) request id=0x16e7, seq=1/256, ttl=64 (reply in 1538)
1550	2023-06-10 15:15:06.103280000	10.1.2.20	10.2.5.50	ICMP	100	Echo (ping) request id=0x16e8, seq=1/256, ttl=64
1551	2023-06-10 15:15:06.103644000	10.1.2.20	10.2.5.50	OF 1.0	184	of_packet_in
1553	2023-06-10 15:15:06.105890000	10.1.2.20	10.2.5.50	ICMP	100	Echo (ping) request id=0x16e8, seq=1/256, ttl=64
1554	2023-06-10 15:15:06.105927000	10.1.2.20	10.2.5.50	ICMP	100	Echo (ping) request id=0x16e8, seq=1/256, ttl=64
1555	2023-06-10 15:15:06.106168000	10.1.2.20	10.2.5.50	OF 1.0	184	of_packet_in
2106	2023-06-10 15:15:16.109422000	10.1.2.20	10.2.6.60	ICMP	100	Echo (ping) request id=0x16eb, seq=1/256, ttl=64
2107	2023-06-10 15:15:16.109913000	10.1.2.20	10.2.6.60	OF 1.0	184	of_packet_in
2111	2023-06-10 15:15:16.152788000	10.1.2.20	10.2.6.60	ICMP	100	Echo (ping) request id=0x16eb, seq=1/256, ttl=64
2112	2023-06-10 15:15:16.152790000	10.1.2.20	10.2.6.60	ICMP	100	Echo (ping) request id=0x16eb, seq=1/256, ttl=64
2113	2023-06-10 15:15:16.153261000	10.1.2.20	10.2.6.60	OF 1.0	184	of_packet_in
2248	2023-06-10 15:15:26.114166000	10.1.2.20	10.2.7.70	ICMP	100	Echo (ping) request id=0x16ef, seq=1/256, ttl=64
2249	2023-06-10 15:15:26.114791000	10.1.2.20	10.2.7.70	OF 1.0	184	of_packet_in
2252	2023-06-10 15:15:26.137713000	10.1.2.20	10.2.7.70	ICMP	100	Echo (ping) request id=0x16ef, seq=1/256, ttl=64
2253	2023-06-10 15:15:26.137718000	10.1.2.20	10.2.7.70	ICMP	100	Echo (ping) request id=0x16ef, seq=1/256, ttl=64
2254	2023-06-10 15:15:26.137818000	10.1.2.20	10.2.7.70	OF 1.0	184	of_packet_in
2383	2023-06-10 15:15:36.117033000	10.1.2.20	10.2.8.80	ICMP	100	Echo (ping) request id=0x16f4, seq=1/256, ttl=64
2384	2023-06-10 15:15:36.117490000	10.1.2.20	10.2.8.80	OF 1.0	184	of_packet_in

(host 20 can not send to h50 to 80)

3029	2023-06-10 15:15:56.236317000	10.1.3.30	10.2.5.50	OF 1.0	184	of_packet_in
3031	2023-06-10 15:15:56.239969000	10.1.3.30	10.2.5.50	ICMP	100	Echo (ping) request id=0x16ff, seq=1/256, ttl=64
3032	2023-06-10 15:15:56.240176000	10.1.3.30	10.2.5.50	ICMP	100	Echo (ping) request id=0x16ff, seq=1/256, ttl=64
3033	2023-06-10 15:15:56.240268000	10.1.3.30	10.2.5.50	OF 1.0	184	of_packet_in
3479	2023-06-10 15:16:06.242196000	10.1.3.30	10.2.6.60	ICMP	100	Echo (ping) request id=0x1702, seq=1/256, ttl=64
3480	2023-06-10 15:16:06.242392000	10.1.3.30	10.2.6.60	OF 1.0	184	of_packet_in
3483	2023-06-10 15:16:06.268859000	10.1.3.30	10.2.6.60	ICMP	100	Echo (ping) request id=0x1702, seq=1/256, ttl=64
3484	2023-06-10 15:16:06.268943000	10.1.3.30	10.2.6.60	ICMP	100	Echo (ping) request id=0x1702, seq=1/256, ttl=64
3485	2023-06-10 15:16:06.269305000	10.1.3.30	10.2.6.60	OF 1.0	184	of_packet_in
3614	2023-06-10 15:16:16.245923000	10.1.3.30	10.2.7.70	ICMP	100	Echo (ping) request id=0x1705, seq=1/256, ttl=64
3615	2023-06-10 15:16:16.246169000	10.1.3.30	10.2.7.70	OF 1.0	184	of_packet_in
3618	2023-06-10 15:16:16.260287000	10.1.3.30	10.2.7.70	ICMP	100	Echo (ping) request id=0x1705, seq=1/256, ttl=64
3619	2023-06-10 15:16:16.260320000	10.1.3.30	10.2.7.70	ICMP	100	Echo (ping) request id=0x1705, seq=1/256, ttl=64
3620	2023-06-10 15:16:16.260422000	10.1.3.30	10.2.7.70	OF 1.0	184	of_packet_in
3731	2023-06-10 15:16:26.248675000	10.1.3.30	10.2.8.80	ICMP	100	Echo (ping) request id=0x1709, seq=1/256, ttl=64
3732	2023-06-10 15:16:26.249199000	10.1.3.30	10.2.8.80	OF 1.0	184	of_packet_in
3735	2023-06-10 15:16:26.286643000	10.1.3.30	10.2.8.80	ICMP	100	Echo (ping) request id=0x1709, seq=1/256, ttl=64
3736	2023-06-10 15:16:26.286645000	10.1.3.30	10.2.8.80	ICMP	100	Echo (ping) request id=0x1709, seq=1/256, ttl=64
3737	2023-06-10 15:16:26.305000000	10.1.3.30	10.2.8.80	OF 1.0	184	of_packet_in

(host 30 can not send to h50 to 80)

4439	2023-06-10 15:16:46.387273000	10.1.4.40	10.2.5.50	ICMP	100	Echo (ping) request id=0x1716, seq=1/256, ttl=64
4440	2023-06-10 15:16:46.387722000	10.1.4.40	10.2.5.50	OF 1.0	184	of_packet_in
4442	2023-06-10 15:16:46.390942000	10.1.4.40	10.2.5.50	ICMP	100	Echo (ping) request id=0x1716, seq=1/256, ttl=64
4443	2023-06-10 15:16:46.390985000	10.1.4.40	10.2.5.50	ICMP	100	Echo (ping) request id=0x1716, seq=1/256, ttl=64
4444	2023-06-10 15:16:46.391377000	10.1.4.40	10.2.5.50	OF 1.0	184	of_packet_in
5002	2023-06-10 15:16:56.393255000	10.1.4.40	10.2.6.60	ICMP	100	Echo (ping) request id=0x1719, seq=1/256, ttl=64
5003	2023-06-10 15:16:56.393749000	10.1.4.40	10.2.6.60	OF 1.0	184	of_packet_in
5006	2023-06-10 15:16:56.432745000	10.1.4.40	10.2.6.60	ICMP	100	Echo (ping) request id=0x1719, seq=1/256, ttl=64
5007	2023-06-10 15:16:56.432825000	10.1.4.40	10.2.6.60	ICMP	100	Echo (ping) request id=0x1719, seq=1/256, ttl=64
5008	2023-06-10 15:16:56.433078000	10.1.4.40	10.2.6.60	OF 1.0	184	of_packet_in
5137	2023-06-10 15:17:06.398298000	10.1.4.40	10.2.7.70	ICMP	100	Echo (ping) request id=0x171f, seq=1/256, ttl=64
5138	2023-06-10 15:17:06.398455000	10.1.4.40	10.2.7.70	OF 1.0	184	of_packet_in
5141	2023-06-10 15:17:06.399780000	10.1.4.40	10.2.7.70	ICMP	100	Echo (ping) request id=0x171f, seq=1/256, ttl=64
5142	2023-06-10 15:17:06.399781000	10.1.4.40	10.2.7.70	ICMP	100	Echo (ping) request id=0x171f, seq=1/256, ttl=64
5143	2023-06-10 15:17:06.400232000	10.1.4.40	10.2.7.70	OF 1.0	184	of_packet_in
5280	2023-06-10 15:17:16.403945000	10.1.4.40	10.2.8.80	ICMP	100	Echo (ping) request id=0x1722, seq=1/256, ttl=64
5281	2023-06-10 15:17:16.404764000	10.1.4.40	10.2.8.80	OF 1.0	184	of_packet_in
5285	2023-06-10 15:17:16.414561000	10.1.4.40	10.2.8.80	ICMP	100	Echo (ping) request id=0x1722, seq=1/256, ttl=64
5286	2023-06-10 15:17:16.414563000	10.1.4.40	10.2.8.80	ICMP	100	Echo (ping) request id=0x1722, seq=1/256, ttl=64

(host 40 can not send to h50 to 80)

Same reasoning as above, there are only echo requests going between the host in department A (h10 to 40) and the host in department B (h50 to 80). Indicating the firewall is blocking ICMP traffic, and those packets will be dropped.

In terms of ip traffic we can use iperf to test the connectivity between them.

```
mininet> iperf h10 h40
*** Iperf: testing TCP bandwidth between h10 and h40
*** Results: ['25.6 Gbits/sec', '25.7 Gbits/sec']
```

Based on the screenshot above we can tell there is successful communication between h10 and h40. Since h10 and h40 both belong to the same department and no firewall is present to block the traffic.

However, things will go differently when they are not in the same department which in our case a set of restrictions are applied in between. As a result we can't get anything out of this.

```
mininet> iperf h10 h80
*** Iperf: testing TCP bandwidth between h10 and h80
```