

Trabalho final Mininet – C115

1) Considere uma topologia árvore com profundidade três e ramificação cinco

a) Com uso de linha de comando padrão do Mininet, crie a topologia considerando o endereço MAC padronizado, larguras de banda bw de 30 Mbps e controlador do Mininet (não precisa especificar)

b) Inspeccione informações das interfaces, endereços MAC, IP e portas através de linhas de comando

1 - Dump

```
mininet@mininet-vm: ~
*** Starting CLI:
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=1856>
<Host h2: h2-eth0:10.0.0.2 pid=1858>
<Host h3: h3-eth0:10.0.0.3 pid=1860>
<Host h4: h4-eth0:10.0.0.4 pid=1862>
<Host h5: h5-eth0:10.0.0.5 pid=1864>
<Host h6: h6-eth0:10.0.0.6 pid=1866>
<Host h7: h7-eth0:10.0.0.7 pid=1868>
<Host h8: h8-eth0:10.0.0.8 pid=1870>
<Host h9: h9-eth0:10.0.0.9 pid=1872>
<Host h10: h10-eth0:10.0.0.10 pid=1874>
<Host h11: h11-eth0:10.0.0.11 pid=1876>
<Host h12: h12-eth0:10.0.0.12 pid=1878>
<Host h13: h13-eth0:10.0.0.13 pid=1880>
<Host h14: h14-eth0:10.0.0.14 pid=1882>
<Host h15: h15-eth0:10.0.0.15 pid=1884>
<Host h16: h16-eth0:10.0.0.16 pid=1886>
<Host h17: h17-eth0:10.0.0.17 pid=1888>
<Host h18: h18-eth0:10.0.0.18 pid=1890>
<Host h19: h19-eth0:10.0.0.19 pid=1892>
<Host h20: h20-eth0:10.0.0.20 pid=1894>
<Host h21: h21-eth0:10.0.0.21 pid=1896>
<Host h22: h22-eth0:10.0.0.22 pid=1898>
<Host h23: h23-eth0:10.0.0.23 pid=1900>
<Host h24: h24-eth0:10.0.0.24 pid=1902>
<Host h25: h25-eth0:10.0.0.25 pid=1904>
<Host h26: h26-eth0:10.0.0.26 pid=1906>
<Host h27: h27-eth0:10.0.0.27 pid=1908>
<Host h28: h28-eth0:10.0.0.28 pid=1910>
<Host h29: h29-eth0:10.0.0.29 pid=1912>
<Host h30: h30-eth0:10.0.0.30 pid=1914>
<Host h31: h31-eth0:10.0.0.31 pid=1916>
<Host h32: h32-eth0:10.0.0.32 pid=1918>
<Host h33: h33-eth0:10.0.0.33 pid=1920>
<Host h34: h34-eth0:10.0.0.34 pid=1922>
<Host h35: h35-eth0:10.0.0.35 pid=1924>
<Host h36: h36-eth0:10.0.0.36 pid=1926>
<Host h37: h37-eth0:10.0.0.37 pid=1928>
<Host h38: h38-eth0:10.0.0.38 pid=1930>
<Host h39: h39-eth0:10.0.0.39 pid=1932>
<Host h40: h40-eth0:10.0.0.40 pid=1934>
<Host h41: h41-eth0:10.0.0.41 pid=1936>
<Host h42: h42-eth0:10.0.0.42 pid=1938>
```

```
mininet@mininet-vm: ~
<Host h43: h43-eth0:10.0.0.43 pid=1940>
<Host h44: h44-eth0:10.0.0.44 pid=1942>
<Host h45: h45-eth0:10.0.0.45 pid=1944>
<Host h46: h46-eth0:10.0.0.46 pid=1946>
<Host h47: h47-eth0:10.0.0.47 pid=1948>
<Host h48: h48-eth0:10.0.0.48 pid=1950>
<Host h49: h49-eth0:10.0.0.49 pid=1952>
<Host h50: h50-eth0:10.0.0.50 pid=1954>
<Host h51: h51-eth0:10.0.0.51 pid=1956>
<Host h52: h52-eth0:10.0.0.52 pid=1958>
<Host h53: h53-eth0:10.0.0.53 pid=1960>
<Host h54: h54-eth0:10.0.0.54 pid=1962>
<Host h55: h55-eth0:10.0.0.55 pid=1964>
<Host h56: h56-eth0:10.0.0.56 pid=1966>
<Host h57: h57-eth0:10.0.0.57 pid=1968>
<Host h58: h58-eth0:10.0.0.58 pid=1970>
<Host h59: h59-eth0:10.0.0.59 pid=1972>
<Host h60: h60-eth0:10.0.0.60 pid=1974>
<Host h61: h61-eth0:10.0.0.61 pid=1976>
<Host h62: h62-eth0:10.0.0.62 pid=1978>
<Host h63: h63-eth0:10.0.0.63 pid=1980>
<Host h64: h64-eth0:10.0.0.64 pid=1982>
<Host h65: h65-eth0:10.0.0.65 pid=1984>
<Host h66: h66-eth0:10.0.0.66 pid=1986>
<Host h67: h67-eth0:10.0.0.67 pid=1988>
<Host h68: h68-eth0:10.0.0.68 pid=1990>
<Host h69: h69-eth0:10.0.0.69 pid=1992>
<Host h70: h70-eth0:10.0.0.70 pid=1994>
<Host h71: h71-eth0:10.0.0.71 pid=1996>
<Host h72: h72-eth0:10.0.0.72 pid=1998>
<Host h73: h73-eth0:10.0.0.73 pid=2000>
<Host h74: h74-eth0:10.0.0.74 pid=2002>
<Host h75: h75-eth0:10.0.0.75 pid=2004>
<Host h76: h76-eth0:10.0.0.76 pid=2006>
<Host h77: h77-eth0:10.0.0.77 pid=2008>
<Host h78: h78-eth0:10.0.0.78 pid=2010>
<Host h79: h79-eth0:10.0.0.79 pid=2012>
<Host h80: h80-eth0:10.0.0.80 pid=2014>
<Host h81: h81-eth0:10.0.0.81 pid=2016>
<Host h82: h82-eth0:10.0.0.82 pid=2018>
<Host h83: h83-eth0:10.0.0.83 pid=2020>
<Host h84: h84-eth0:10.0.0.84 pid=2022>
<Host h85: h85-eth0:10.0.0.85 pid=2024>
<Host h86: h86-eth0:10.0.0.86 pid=2026>
```

```

mininet@mininet-vm: ~
<Host h87: h87-eth0:10.0.0.87 pid=2028>
<Host h88: h88-eth0:10.0.0.88 pid=2030>
<Host h89: h89-eth0:10.0.0.89 pid=2032>
<Host h90: h90-eth0:10.0.0.90 pid=2034>
<Host h91: h91-eth0:10.0.0.91 pid=2036>
<Host h92: h92-eth0:10.0.0.92 pid=2038>
<Host h93: h93-eth0:10.0.0.93 pid=2040>
<Host h94: h94-eth0:10.0.0.94 pid=2042>
<Host h95: h95-eth0:10.0.0.95 pid=2044>
<Host h96: h96-eth0:10.0.0.96 pid=2046>
<Host h97: h97-eth0:10.0.0.97 pid=2048>
<Host h98: h98-eth0:10.0.0.98 pid=2050>
<Host h99: h99-eth0:10.0.0.99 pid=2052>
<Host h100: h100-eth0:10.0.0.100 pid=2054>
<Host h101: h101-eth0:10.0.0.101 pid=2056>
<Host h102: h102-eth0:10.0.0.102 pid=2058>
<Host h103: h103-eth0:10.0.0.103 pid=2060>
<Host h104: h104-eth0:10.0.0.104 pid=2062>
<Host h105: h105-eth0:10.0.0.105 pid=2064>
<Host h106: h106-eth0:10.0.0.106 pid=2066>
<Host h107: h107-eth0:10.0.0.107 pid=2068>
<Host h108: h108-eth0:10.0.0.108 pid=2070>
<Host h109: h109-eth0:10.0.0.109 pid=2072>
<Host h110: h110-eth0:10.0.0.110 pid=2074>
<Host h111: h111-eth0:10.0.0.111 pid=2076>
<Host h112: h112-eth0:10.0.0.112 pid=2078>
<Host h113: h113-eth0:10.0.0.113 pid=2080>
<Host h114: h114-eth0:10.0.0.114 pid=2082>
<Host h115: h115-eth0:10.0.0.115 pid=2084>
<Host h116: h116-eth0:10.0.0.116 pid=2086>
<Host h117: h117-eth0:10.0.0.117 pid=2088>
<Host h118: h118-eth0:10.0.0.118 pid=2090>
<Host h119: h119-eth0:10.0.0.119 pid=2092>
<Host h120: h120-eth0:10.0.0.120 pid=2094>
<Host h121: h121-eth0:10.0.0.121 pid=2096>
<Host h122: h122-eth0:10.0.0.122 pid=2098>
<Host h123: h123-eth0:10.0.0.123 pid=2100>
<Host h124: h124-eth0:10.0.0.124 pid=2102>
<Host h125: h125-eth0:10.0.0.125 pid=2104>
<OVSSwitch s1: lo:127.0.0.1, s1-eth1:None, s1-eth2:None, s1-eth3:None, s1-eth4:None, s1-eth5:None pid=2109>
<OVSSwitch s2: lo:127.0.0.1, s2-eth1:None, s2-eth2:None, s2-eth3:None, s2-eth4:None, s2-eth5:None, s2-eth6:None pid=2112>
<OVSSwitch s3: lo:127.0.0.1, s3-eth1:None, s3-eth2:None, s3-eth3:None, s3-eth4:None, s3-eth5:None, s3-eth6:None pid=2115>
<OVSSwitch s4: lo:127.0.0.1, s4-eth1:None, s4-eth2:None, s4-eth3:None, s4-eth4:None, s4-eth5:None, s4-eth6:None pid=2118>
<OVSSwitch s5: lo:127.0.0.1, s5-eth1:None, s5-eth2:None, s5-eth3:None, s5-eth4:None, s5-eth5:None, s5-eth6:None pid=2121>
<OVSSwitch s6: lo:127.0.0.1, s6-eth1:None, s6-eth2:None, s6-eth3:None, s6-eth4:None, s6-eth5:None, s6-eth6:None pid=2124>
<OVSSwitch s7: lo:127.0.0.1, s7-eth1:None, s7-eth2:None, s7-eth3:None, s7-eth4:None, s7-eth5:None, s7-eth6:None pid=2127>
<OVSSwitch s8: lo:127.0.0.1, s8-eth1:None, s8-eth2:None, s8-eth3:None, s8-eth4:None, s8-eth5:None, s8-eth6:None pid=2131>
<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 pid=2134>
<OVSSwitch s10: lo:127.0.0.1, s10-eth1:None, s10-eth2:None, s10-eth3:None, s10-eth4:None, s10-eth5:None, s10-eth6:None pid=2137>
<OVSSwitch s11: lo:127.0.0.1, s11-eth1:None, s11-eth2:None, s11-eth3:None, s11-eth4:None, s11-eth5:None, s11-eth6:None pid=2140>
<OVSSwitch s12: lo:127.0.0.1, s12-eth1:None, s12-eth2:None, s12-eth3:None, s12-eth4:None, s12-eth5:None, s12-eth6:None pid=2143>
<OVSSwitch s13: lo:127.0.0.1, s13-eth1:None, s13-eth2:None, s13-eth3:None, s13-eth4:None, s13-eth5:None, s13-eth6:None pid=2146>
<OVSSwitch s14: lo:127.0.0.1, s14-eth1:None, s14-eth2:None, s14-eth3:None, s14-eth4:None, s14-eth5:None, s14-eth6:None pid=2149>
<OVSSwitch s15: lo:127.0.0.1, s15-eth1:None, s15-eth2:None, s15-eth3:None, s15-eth4:None, s15-eth5:None, s15-eth6:None pid=2152>
<OVSSwitch s16: lo:127.0.0.1, s16-eth1:None, s16-eth2:None, s16-eth3:None, s16-eth4:None, s16-eth5:None, s16-eth6:None pid=2155>
<OVSSwitch s17: lo:127.0.0.1, s17-eth1:None, s17-eth2:None, s17-eth3:None, s17-eth4:None, s17-eth5:None, s17-eth6:None pid=2158>
<OVSSwitch s18: lo:127.0.0.1, s18-eth1:None, s18-eth2:None, s18-eth3:None, s18-eth4:None, s18-eth5:None, s18-eth6:None pid=2161>
<OVSSwitch s19: lo:127.0.0.1, s19-eth1:None, s19-eth2:None, s19-eth3:None, s19-eth4:None, s19-eth5:None, s19-eth6:None pid=2164>
<OVSSwitch s20: lo:127.0.0.1, s20-eth1:None, s20-eth2:None, s20-eth3:None, s20-eth4:None, s20-eth5:None, s20-eth6:None pid=2167>
<OVSSwitch s21: lo:127.0.0.1, s21-eth1:None, s21-eth2:None, s21-eth3:None, s21-eth4:None, s21-eth5:None, s21-eth6:None pid=2170>
<OVSSwitch s22: lo:127.0.0.1, s22-eth1:None, s22-eth2:None, s22-eth3:None, s22-eth4:None, s22-eth5:None, s22-eth6:None pid=2173>
<OVSSwitch s23: lo:127.0.0.1, s23-eth1:None, s23-eth2:None, s23-eth3:None, s23-eth4:None, s23-eth5:None, s23-eth6:None pid=2176>
<OVSSwitch s24: lo:127.0.0.1, s24-eth1:None, s24-eth2:None, s24-eth3:None, s24-eth4:None, s24-eth5:None, s24-eth6:None pid=2179>
<OVSSwitch s25: lo:127.0.0.1, s25-eth1:None, s25-eth2:None, s25-eth3:None, s25-eth4:None, s25-eth5:None, s25-eth6:None pid=2182>
<OVSSwitch s26: lo:127.0.0.1, s26-eth1:None, s26-eth2:None, s26-eth3:None, s26-eth4:None, s26-eth5:None, s26-eth6:None pid=2185>
<OVSSwitch s27: lo:127.0.0.1, s27-eth1:None, s27-eth2:None, s27-eth3:None, s27-eth4:None, s27-eth5:None, s27-eth6:None pid=2188>
<OVSSwitch s28: lo:127.0.0.1, s28-eth1:None, s28-eth2:None, s28-eth3:None, s28-eth4:None, s28-eth5:None, s28-eth6:None pid=2191>
<OVSSwitch s29: lo:127.0.0.1, s29-eth1:None, s29-eth2:None, s29-eth3:None, s29-eth4:None, s29-eth5:None, s29-eth6:None pid=2194>
<OVSSwitch s30: lo:127.0.0.1, s30-eth1:None, s30-eth2:None, s30-eth3:None, s30-eth4:None, s30-eth5:None, s30-eth6:None pid=2197>
<OVSSwitch s31: lo:127.0.0.1, s31-eth1:None, s31-eth2:None, s31-eth3:None, s31-eth4:None, s31-eth5:None, s31-eth6:None pid=2200>
<Controller c0: 127.0.0.1:6653 pid=1849>
mininet> 

```

2 – h1 config -a

```

mininet> h1 ifconfig -a
h1-eth0  Link encap:Ethernet HWaddr 00:00:00:00:00:01
          inet addr:10.0.0.1 Bcast:10.255.255.255 Mask:255.0.0.0
                  UP BROADCAST RUNNING MULTICAST MTU:1500 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)
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: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)
mininet> 

```

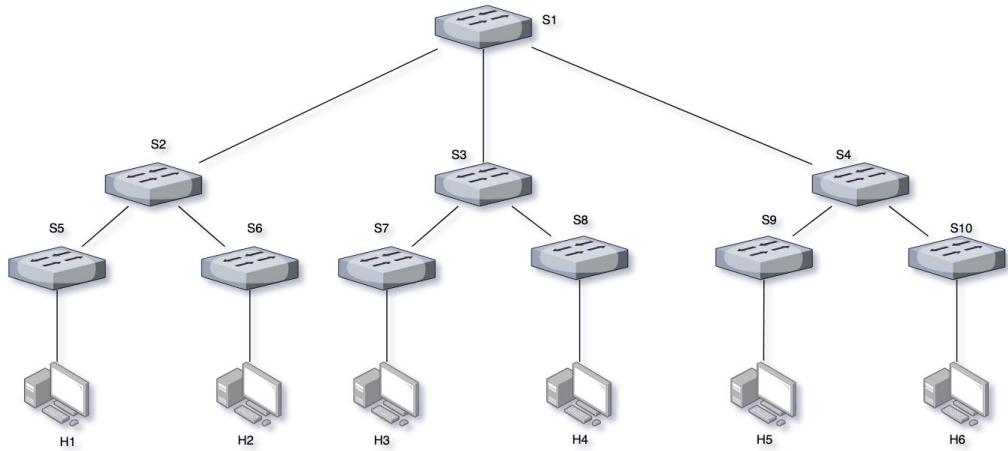
3 – sh ovs-ofctl show s1

```

mininet> sh ovs-ofctl show s1
OFPT_FEATURES_REPLY (xid=0x2): dpid:0000000000000001
n_tables:254, n_buffers:256
capabilities: FLOW_STATS TABLE_STATS PORT_STATS QUEUE_STATS ARP_MATCH_IP
actions: OUTPUT SET VLAN_VID SET VLAN_PCP STRIP_VLAN SET_DL_SRC SET_DL_DST SET_NW_SRC SET_NW_DST SET_NW_TOS SET_TP_SRC SET_TP_DST ENQUEUE
1(s1-eth1): addr:5:a:d:c:c:a:47
    config: 0
    state: 0
    current: 10GB-FD COPPER
    speed: 10000 Mbps now, 0 Mbps max
2(s1-eth2): addr:62:f:a:1:71:f5:38
    config: 0
    state: 0
    current: 10GB-FD COPPER
    speed: 10000 Mbps now, 0 Mbps max
3(s1-eth3): addr:9:a:8:c:fe:a:66
    config: 0
    state: 0
    current: 10GB-FD COPPER
    speed: 10000 Mbps now, 0 Mbps max
4(s1-eth4): addr:d2:69:3:a:32:23:bb
    config: 0
    state: 0
    current: 10GB-FD COPPER
    speed: 10000 Mbps now, 0 Mbps max
5(s1-eth5): addr:9:e:16:c:7:53:ce:6e
    config: 0
    state: 0
    current: 10GB-FD COPPER
    speed: 10000 Mbps now, 0 Mbps max
LOCAL(s1): addr:9:e:16:0:2:0:a:4c
    config: 0
    state: 0
    current: 0 Mbps now, 0 Mbps max
OFPT_GET_CONFIG_REPLY (xid=0x4): frags:normal miss_send_len=0
mininet> 

```

c - Crie um desenho ilustrativo da topologia com todas as informações obtidas no item anterior



d - Execute testes de ping entre os diferentes nós, mostre os pacotes chegando nos nós com uso do comando tcpdump.

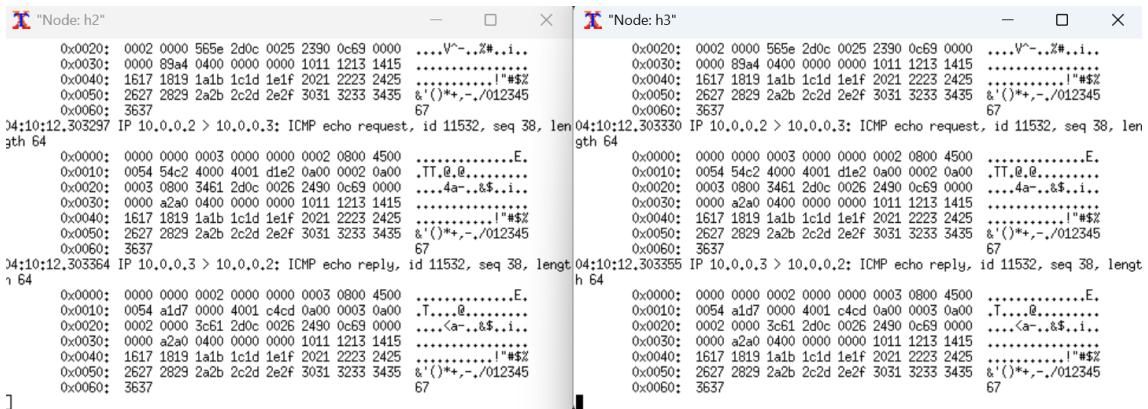
```
mininet> xterm h2 mininet> xterm h3
```

```
X "Node: h2"                                     X "Node: h3"
root@mininet-vm:~# tcpdump -XX -n -i h2-eth0      root@mininet-vm:~# tcpdump -XX -n -i h3-eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on h2-eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
[
```

```

mininet> h2 ping h3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=23.0 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=11.5 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=0.138 ms
64 bytes from 10.0.0.3: icmp_seq=4 ttl=64 time=0.127 ms
64 bytes from 10.0.0.3: icmp_seq=5 ttl=64 time=0.137 ms
64 bytes from 10.0.0.3: icmp_seq=6 ttl=64 time=0.084 ms
64 bytes from 10.0.0.3: icmp_seq=7 ttl=64 time=3.77 ms
64 bytes from 10.0.0.3: icmp_seq=8 ttl=64 time=0.097 ms
64 bytes from 10.0.0.3: icmp_seq=9 ttl=64 time=0.070 ms
64 bytes from 10.0.0.3: icmp_seq=10 ttl=64 time=0.144 ms
64 bytes from 10.0.0.3: icmp_seq=11 ttl=64 time=0.153 ms
64 bytes from 10.0.0.3: icmp_seq=12 ttl=64 time=0.167 ms
64 bytes from 10.0.0.3: icmp_seq=13 ttl=64 time=0.105 ms
64 bytes from 10.0.0.3: icmp_seq=14 ttl=64 time=0.091 ms
64 bytes from 10.0.0.3: icmp_seq=15 ttl=64 time=0.073 ms
64 bytes from 10.0.0.3: icmp_seq=16 ttl=64 time=0.072 ms
64 bytes from 10.0.0.3: icmp_seq=17 ttl=64 time=0.094 ms
64 bytes from 10.0.0.3: icmp_seq=18 ttl=64 time=0.073 ms
64 bytes from 10.0.0.3: icmp_seq=19 ttl=64 time=0.135 ms
64 bytes from 10.0.0.3: icmp_seq=20 ttl=64 time=0.098 ms
64 bytes from 10.0.0.3: icmp_seq=21 ttl=64 time=0.070 ms
64 bytes from 10.0.0.3: icmp_seq=22 ttl=64 time=0.077 ms
64 bytes from 10.0.0.3: icmp_seq=23 ttl=64 time=0.095 ms
64 bytes from 10.0.0.3: icmp_seq=24 ttl=64 time=0.101 ms
64 bytes from 10.0.0.3: icmp_seq=25 ttl=64 time=0.085 ms
64 bytes from 10.0.0.3: icmp_seq=26 ttl=64 time=0.052 ms
64 bytes from 10.0.0.3: icmp_seq=27 ttl=64 time=0.094 ms
64 bytes from 10.0.0.3: icmp_seq=28 ttl=64 time=0.141 ms
64 bytes from 10.0.0.3: icmp_seq=29 ttl=64 time=0.046 ms
64 bytes from 10.0.0.3: icmp_seq=30 ttl=64 time=0.048 ms
64 bytes from 10.0.0.3: icmp_seq=31 ttl=64 time=0.066 ms
64 bytes from 10.0.0.3: icmp_seq=32 ttl=64 time=0.298 ms
64 bytes from 10.0.0.3: icmp_seq=33 ttl=64 time=0.056 ms
64 bytes from 10.0.0.3: icmp_seq=34 ttl=64 time=0.070 ms
64 bytes from 10.0.0.3: icmp_seq=35 ttl=64 time=0.095 ms
64 bytes from 10.0.0.3: icmp_seq=36 ttl=64 time=0.062 ms
64 bytes from 10.0.0.3: icmp_seq=37 ttl=64 time=0.151 ms
64 bytes from 10.0.0.3: icmp_seq=38 ttl=64 time=0.098 ms
^C
--- 10.0.0.3 ping statistics ---
38 packets transmitted, 38 received, 0% packet loss, time 37012ms
rtt min/avg/max/mdev = 0.046/1.104/23.073/4.084 ms

```



e - Especifique que o host 1 na porta 5555 vai ser um servidor TCP e o host 2 um cliente e execute testes de iperf, considere um relatório por segundo com teste de 20 segundos. Faça os testes para larguras de banda bw de 30 e 40 Mbps (Necessário reconstruir a topologia para os outros valores).

```
mininet> xterm h1  
mininet> xterm h2  
mininet>
```

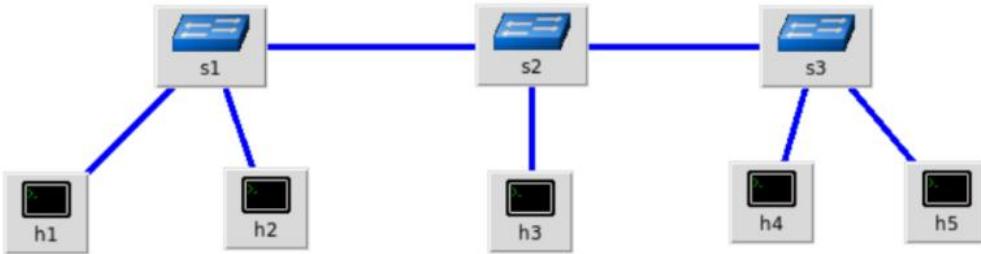
```

Node: h1
root@mininet-vm:~# iperf -s -p 5555 -i 1
Server listening on TCP port 5555
TCP window size: 85.3 KByte (default)
[320] local 10.0.0.1 port 5555 connected with 10.0.0.2 port 44468
[ ID] Interval Transfer Bandwidth
[320] 0.0- 1.0 sec 4.30 MBytes 36.1 Mbits/sec
[320] 1.0- 2.0 sec 4.32 MBytes 36.2 Mbits/sec
[320] 2.0- 3.0 sec 4.30 MBytes 36.1 Mbits/sec
[320] 3.0- 4.0 sec 4.32 MBytes 36.3 Mbits/sec
[320] 4.0- 5.0 sec 4.19 MBytes 35.2 Mbits/sec
[320] 5.0- 6.0 sec 4.40 MBytes 36.9 Mbits/sec
[320] 6.0- 7.0 sec 4.33 MBytes 36.4 Mbits/sec
[320] 7.0- 8.0 sec 4.31 MBytes 36.2 Mbits/sec
[320] 8.0- 9.0 sec 4.29 MBytes 36.0 Mbits/sec
[320] 9.0-10.0 sec 4.33 MBytes 36.3 Mbits/sec
[320] 10.0-11.0 sec 4.33 MBytes 36.4 Mbits/sec
[320] 11.0-12.0 sec 4.30 MBytes 36.1 Mbits/sec
[320] 12.0-13.0 sec 4.31 MBytes 36.2 Mbits/sec
[320] 13.0-14.0 sec 4.31 MBytes 36.2 Mbits/sec
[320] 14.0-15.0 sec 4.24 MBytes 35.6 Mbits/sec
[320] 0.0-15.5 sec 66.9 MBytes 36.1 Mbits/sec

Node: h2
root@mininet-vm:~# iperf -c 10.0.0.1 -p 5555 -i 1 -t 20
Client connecting to 10.0.0.1, TCP port 5555
TCP window size: 85.3 KByte (default)
[319] local 10.0.0.2 port 44468 connected with 10.0.0.1 port 5555
[ ID] Interval Transfer Bandwidth
[319] 0.0- 1.0 sec 4.88 MBytes 40.9 Mbits/sec
[319] 1.0- 2.0 sec 4.25 MBytes 35.7 Mbits/sec
[319] 2.0- 3.0 sec 4.50 MBytes 37.7 Mbits/sec
[319] 3.0- 4.0 sec 4.12 MBytes 34.6 Mbits/sec
[319] 4.0- 5.0 sec 4.25 MBytes 35.7 Mbits/sec
[319] 5.0- 6.0 sec 4.50 MBytes 37.7 Mbits/sec
[319] 6.0- 7.0 sec 4.25 MBytes 35.7 Mbits/sec
[319] 7.0- 8.0 sec 4.38 MBytes 36.7 Mbits/sec
[319] 8.0- 9.0 sec 4.38 MBytes 36.7 Mbits/sec
[319] 9.0-10.0 sec 4.38 MBytes 36.7 Mbits/sec
[319] 10.0-11.0 sec 4.38 MBytes 36.7 Mbits/sec
[319] 11.0-12.0 sec 4.25 MBytes 35.7 Mbits/sec
[319] 12.0-13.0 sec 4.38 MBytes 36.7 Mbits/sec
[319] 13.0-14.0 sec 4.38 MBytes 36.7 Mbits/sec
[319] 14.0-15.0 sec 4.00 MBytes 33.6 Mbits/sec
^C[319] 0.0-15.4 sec 66.9 MBytes 36.5 Mbits/sec
root@mininet-vm:~#

```

2) Crie um código Python para a topologia customizada abaixo:



a - Com uso de linha de comando padrão do Mininet, crie a topologia customizada considerando o endereço MAC padronizado e controlador manual

```

mininet@mininet-vm:~/trabalho$ sudo python TrabalhoMininet-final.py
*** Creating network
*** Adding hosts:
h1 h2 h3 h4 h5
*** Adding switches:
s1 s2 s3
*** Adding links:
(h1, s1) (h2, s1) (h3, s2) (h4, s3) (h5, s3) (s1, s2) (s2, s3)
*** Configuring hosts
h1 h2 h3 h4 h5
*** Starting controller

*** Starting 3 switches
s1 s2 s3 ...

```

b - Inspecione informações das interfaces, endereços MAC, IP e portas através de linhas de comando;

```

mininet> nodes
available nodes are:
h1 h2 h3 h4 h5 s1 s2 s3

```

```

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
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0 s1-eth3:s2-eth2
s2 lo: s2-eth1:h3-eth0 s2-eth2:s1-eth3 s2-eth3:s3-eth3
s3 lo: s3-eth1:h4-eth0 s3-eth2:h5-eth0 s3-eth3:s2-eth3

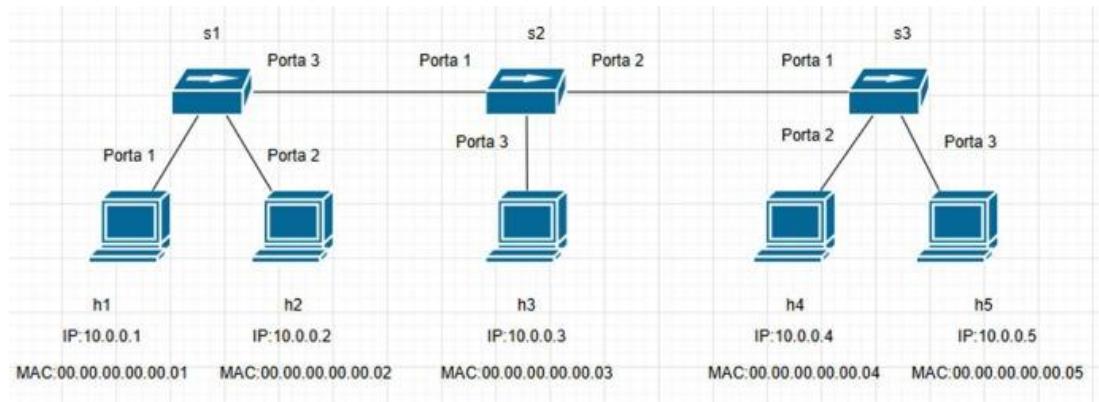
```

```

mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=3511>
<Host h2: h2-eth0:10.0.0.2 pid=3513>
<Host h3: h3-eth0:10.0.0.3 pid=3515>
<Host h4: h4-eth0:10.0.0.4 pid=3517>
<Host h5: h5-eth0:10.0.0.5 pid=3519>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None pid=3524>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None,s2-eth3:None pid=3527>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None,s3-eth3:None pid=3530>

```

c - Crie um desenho ilustrativo da topologia com todas as informações obtidas no item anterior



d - Faça testes de ping considerando os switches normais;

```

==== Testando conectividade (ping all) ====
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5
h2 -> h1 h3 h4 h5
h3 -> h1 h2 h4 h5
h4 -> h1 h2 h3 h5
h5 -> h1 h2 h3 h4
*** Results: 0% dropped (20/20 received)

```

e - Apague as regras anteriores e crie regras baseadas em endereços MAC para alguns nós.

```
mininet> sh ovs-ofctl del-flows s1  
mininet> sh ovs-ofctl del-flows s2  
mininet> sh ovs-ofctl del-flows s3
```

```
mininet> sh ovs-ofctl add-flow s1 dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:03,actions=output:3  
mininet> sh ovs-ofctl add-flow s2 dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:03,actions=output:1  
mininet> sh ovs-ofctl add-flow s2 dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:01,actions=output:2  
mininet> sh ovs-ofctl add-flow s1 dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:01,actions=output:1
```

f - Faça testes de ping para demonstrar que as regras foram bem implementadas.

```
mininet> pingall  
*** Ping: testing ping reachability  
h1 -> X h3 X X  
h2 -> X X X X  
h3 -> h1 X X X  
h4 -> X X X X  
h5 -> X X X X  
*** Results: 90% dropped (2/20 received)
```