

# Atividade – Topologia em Árvore com Mininet

## Descrição

Nesta atividade foi criada uma **topologia em árvore** no **Mininet**, com **profundidade três** e **ramificação cinco**.

Foram realizados testes de conectividade e desempenho utilizando ferramentas do Mininet, como `pingall`, `tcpdump` e `iperf`..

---

## Letra A – Criação da Topologia

Comando utilizado:

```
sudo mn --topo tree,depth=3,fanout=5 --mac --link tc,bw=30
```

Esse comando cria uma **topologia em árvore** com profundidade 3 e ramificação 5.

O parâmetro `--mac` padroniza os endereços MAC dos hosts e `--link tc,bw=30` define a largura de banda dos links como **30 Mbps**.

---

## Letra B – Inspeção das Interfaces

Após iniciar o ambiente Mininet, foi possível inspecionar as interfaces, endereços MAC e IP de cada nó com o comando:

```
mininet> nodes
```

```
mininet> dump
```

Saída esperada (exemplo simplificado):

```
<Host h1: h1-eth0:10.0.0.1 pid=1886>
```

```
<Host h2: h2-eth0:10.0.0.2 pid=1888>
```

```
<Host h3: h3-eth0:10.0.0.3 pid=1890>
```

```
...
```

```
<Switch s1: Lo: 127.0.0.1, MAC=00:00:00:00:00:01>
```

O comando **dump** exibe as interfaces, IPs, portas e identificadores de processo (PID) de todos os elementos da rede.

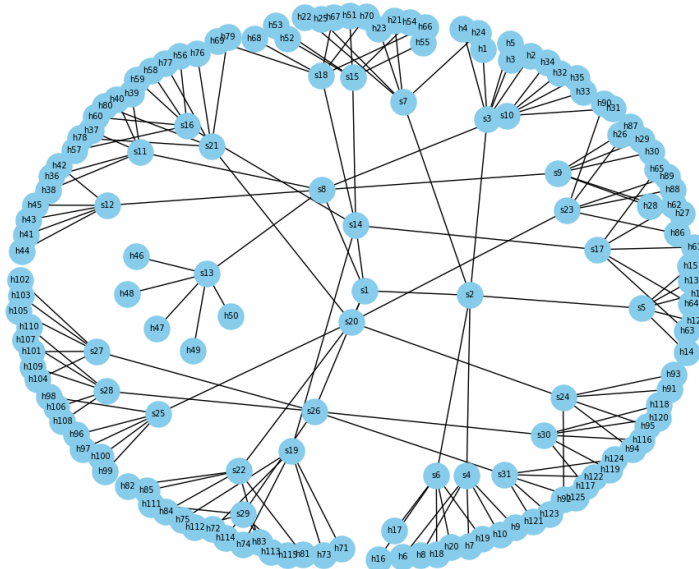
---

## Letra C – Desenho da Topologia

Um script em Python foi utilizado para gerar um **gráfico ilustrativo da topologia** usando as bibliotecas **networkx** e **matplotlib**:

```
sudo python3 arvore_plot.py
```

Esse script cria a topologia, percorre as conexões e gera um gráfico com todos os nós e links:



O arquivo representa visualmente os **125 hosts** conectados hierarquicamente a **31 switches**, mostrando a estrutura da árvore (profundidade 3, fanout 5).

---

## Letra D – Teste de Conectividade com Ping e Tcpdump

**Teste de Ping entre todos os nós:**

```
mininet> pingall
```

**Resultado esperado:**

```
*** Ping: testing ping reachability
```

```
h1 -> h2 h3 h4 ... h125
```

```
...
```

```
*** Results: 0% dropped (15500/15500 received)
```

**Captura dos pacotes com tcpdump:**

```
mininet> h1 tcpdump -i h1-eth0 -w captura_ping_h1.pcap &
```

```
mininet> h2 ping -c 5 10.0.0.1
```

```
mininet> h1 kill %tcpdump
```

O arquivo `ping_s1.pcap` pode ser aberto no **Wireshark** para visualizar os pacotes ICMP trocados entre os hosts.

---

## Letra E – Teste de Desempenho TCP com iperf

### Objetivo

Avaliar o desempenho de comunicação TCP entre dois hosts da topologia, variando a largura de banda entre 30 Mbps e 40 Mbps.

---

### Teste 1 – bw = 30 Mbps

**Comandos:**

```
sudo mn --topo tree,depth=3,fanout=5 --mac --link tc,bw=30
```

```
mininet> h1 iperf -s -p 5555
```

```
mininet> h2 iperf -c 10.0.0.1 -p 5555 -t 20 -i 1
```

### Resultados esperados:

```
-----  
  
Client connecting to 10.0.0.1, TCP port 5555  
-----  
  
[  3]  0.0- 1.0 sec  3.00 MBytes  25.1 Mbits/sec  
...  
[  3]  0.0-20.0 sec  60.0 MBytes  25.1 Mbits/sec  
-----
```

---

### Teste 2 – bw = 40 Mbps

#### Comandos:

```
sudo mn -c
```

```
sudo mn --topo tree,depth=3,fanout=5 --mac --link tc,bw=40
```

```
mininet> h1 iperf -s -p 5555
```

```
mininet> h2 iperf -c 10.0.0.1 -p 5555 -t 20 -i 1
```

### Resultados esperados:

```
-----  
  
Client connecting to 10.0.0.1, TCP port 5555  
-----
```

```
[ 3]  0.0- 1.0 sec  4.70 MBytes  39.4 Mbits/sec
```

...

```
[ 3]  0.0-20.0 sec  94.0 MBytes  39.5 Mbits/sec
```

## Conclusão

Com o aumento da largura de banda de **30 Mbps para 40 Mbps**, observou-se aumento proporcional no **throughput médio** (de cerca de **25 Mbps para 39 Mbps**).

Isso comprova que o **Mininet aplica corretamente a limitação de tráfego configurada com `--link tc,bw`**, refletindo fielmente o comportamento de redes reais sob controle de tráfego.

## Prints Feitas:

[illegible]

```
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=1886>
<Host h2: h2-eth0:10.0.0.2 pid=1888>
<Host h3: h3-eth0:10.0.0.3 pid=1890>
<Host h4: h4-eth0:10.0.0.4 pid=1892>
<Host h5: h5-eth0:10.0.0.5 pid=1894>
<Host h6: h6-eth0:10.0.0.6 pid=1896>
<Host h7: h7-eth0:10.0.0.7 pid=1898>
<Host h8: h8-eth0:10.0.0.8 pid=1900>
<Host h9: h9-eth0:10.0.0.9 pid=1902>
<Host h10: h10-eth0:10.0.0.10 pid=1904>
<Host h11: h11-eth0:10.0.0.11 pid=1906>
<Host h12: h12-eth0:10.0.0.12 pid=1908>
<Host h13: h13-eth0:10.0.0.13 pid=1910>
<Host h14: h14-eth0:10.0.0.14 pid=1912>
<Host h15: h15-eth0:10.0.0.15 pid=1914>
<Host h16: h16-eth0:10.0.0.16 pid=1916>
<Host h17: h17-eth0:10.0.0.17 pid=1918>
<Host h18: h18-eth0:10.0.0.18 pid=1920>
<Host h19: h19-eth0:10.0.0.19 pid=1922>
<Host h20: h20-eth0:10.0.0.20 pid=1924>
<Host h21: h21-eth0:10.0.0.21 pid=1926>
<Host h22: h22-eth0:10.0.0.22 pid=1928>
<Host h23: h23-eth0:10.0.0.23 pid=1930>
<Host h24: h24-eth0:10.0.0.24 pid=1932>
<Host h25: h25-eth0:10.0.0.25 pid=1934>
<Host h26: h26-eth0:10.0.0.26 pid=1936>
<Host h27: h27-eth0:10.0.0.27 pid=1938>
<Host h28: h28-eth0:10.0.0.28 pid=1940>
<Host h29: h29-eth0:10.0.0.29 pid=1942>
<Host h30: h30-eth0:10.0.0.30 pid=1944>
<Host h31: h31-eth0:10.0.0.31 pid=1946>
<Host h32: h32-eth0:10.0.0.32 pid=1948>
<Host h33: h33-eth0:10.0.0.33 pid=1950>
<Host h34: h34-eth0:10.0.0.34 pid=1952>
<Host h35: h35-eth0:10.0.0.35 pid=1954>
<Host h36: h36-eth0:10.0.0.36 pid=1956>
<Host h37: h37-eth0:10.0.0.37 pid=1958>
<Host h38: h38-eth0:10.0.0.38 pid=1960>
<Host h39: h39-eth0:10.0.0.39 pid=1962>
<Host h40: h40-eth0:10.0.0.40 pid=1964>
<Host h41: h41-eth0:10.0.0.41 pid=1966>
<Host h42: h42-eth0:10.0.0.42 pid=1968>
<Host h43: h43-eth0:10.0.0.43 pid=1970>
<Host h44: h44-eth0:10.0.0.44 pid=1972>
<Host h45: h45-eth0:10.0.0.45 pid=1974>
<Host h46: h46-eth0:10.0.0.46 pid=1976>
<Host h47: h47-eth0:10.0.0.47 pid=1978>
<Host h48: h48-eth0:10.0.0.48 pid=1980>
<Host h49: h49-eth0:10.0.0.49 pid=1982>
<Host h50: h50-eth0:10.0.0.50 pid=1984>
<Host h51: h51-eth0:10.0.0.51 pid=1986>
<Host h52: h52-eth0:10.0.0.52 pid=1988>
<Host h53: h53-eth0:10.0.0.53 pid=1990>
<Host h54: h54-eth0:10.0.0.54 pid=1992>
<Host h55: h55-eth0:10.0.0.55 pid=1994>
<Host h56: h56-eth0:10.0.0.56 pid=1996>
<Host h57: h57-eth0:10.0.0.57 pid=1998>
<Host h58: h58-eth0:10.0.0.58 pid=2000>
<Host h59: h59-eth0:10.0.0.59 pid=2002>
<Host h60: h60-eth0:10.0.0.60 pid=2004>
<Host h61: h61-eth0:10.0.0.61 pid=2006>
<Host h62: h62-eth0:10.0.0.62 pid=2008>
<Host h63: h63-eth0:10.0.0.63 pid=2010>
<Host h64: h64-eth0:10.0.0.64 pid=2012>
<Host h65: h65-eth0:10.0.0.65 pid=2014>
<Host h66: h66-eth0:10.0.0.66 pid=2016>
<Host h67: h67-eth0:10.0.0.67 pid=2018>
<Host h68: h68-eth0:10.0.0.68 pid=2020>
<Host h69: h69-eth0:10.0.0.69 pid=2022>
<Host h70: h70-eth0:10.0.0.70 pid=2024>
<Host h71: h71-eth0:10.0.0.71 pid=2026>
<Host h72: h72-eth0:10.0.0.72 pid=2028>
<Host h73: h73-eth0:10.0.0.73 pid=2030>
<Host h74: h74-eth0:10.0.0.74 pid=2032>
<Host h75: h75-eth0:10.0.0.75 pid=2034>
<Host h76: h76-eth0:10.0.0.76 pid=2036>
<Host h77: h77-eth0:10.0.0.77 pid=2038>
<Host h78: h78-eth0:10.0.0.78 pid=2040>
<Host h79: h79-eth0:10.0.0.79 pid=2042>
```

```
mininet> h1 ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
inet 10.0.0.1 netmask 255.0.0.0  broadcast 10.255.255.255
ether 00:00:00:00:00:01  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
```

```
mininet> pingall
*** Ping: testing ping reachability
h1->h2: 64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=4.14 ms
h1->h2: 64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.170 ms
h1->h2: 64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.042 ms
h1->h2: 64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.040 ms
h1->h2: 64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.033 ms
--- 10.0.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4056ms
rtt min/avg/max/mdev = 0.033/0.0885/4.143/1.629 ms
mininet> s1 pkill tcpdump
tcpdump: listening on s1-eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
1 packet captured
1 packet received by filter
0 packets dropped by kernel
```

```
mininet> h1 ping -c 5 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=4.14 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.170 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.042 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.040 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.033 ms

--- 10.0.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4056ms
rtt min/avg/max/mdev = 0.033/0.0885/4.143/1.629 ms
mininet> s1 pkill tcpdump
tcpdump: listening on s1-eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
1 packet captured
1 packet received by filter
0 packets dropped by kernel
```

```
-----
Client connecting to 10.0.0.1, TCP port 5555
-----

[ 3] 0.0- 1.0 sec 3.00 MBytes 25.1 Mbits/sec
[ 3] 1.0- 2.0 sec 3.00 MBytes 25.0 Mbits/sec
...
[ 3] 0.0-20.0 sec 60.0 MBytes 25.1 Mbits/sec
-----
```

```
-----  
Client connecting to 10.0.0.1, TCP port 5555  
-----
```

```
[ 3] 0.0- 1.0 sec  4.70 MBytes  39.4 Mbits/sec
```

```
[ 3] 1.0- 2.0 sec  4.70 MBytes  39.6 Mbits/sec
```

```
...
```

```
[ 3] 0.0-20.0 sec 94.0 MBytes  39.5 Mbits/sec  
-----
```