

# TRABALHO MININET

NOME: Guilherme Felipe Ribeiro

MAT: 2042

CURSO: GEC

GITHUB: [https://github.com/Gpatinho/C115\\_Conceitos-e-Tecnologias-para-Dispositivos-Conectados](https://github.com/Gpatinho/C115_Conceitos-e-Tecnologias-para-Dispositivos-Conectados)

## OBJETIVO :

Criar uma **topologia linear com 6 switches**, configurar parâmetros (bw, MAC, controlador), **analisar interfaces e endereços**, e realizar **testes de conectividade (ping)** e **desempenho (iperf TCP)**.

## 1 - PREPARANDO O AMBIENTE:

### -Pré-requisitos:

- Ter a **VM Mininet** em execução.
- Ter **Putty** e **Xming** instalados no Windows.
- Estar conectado via Putty com **X11 forwarding habilitado**.

### -CONECT VIA PUTTY:

Inicie o **Xming** (deixe rodando em segundo plano).

- Abra o **Putty** → Vá em Connection → SSH → X11 → marque **Enable X11 forwarding**.
- Conecte-se ao IP da VM (exemplo: 192.168.56.101).
- Login:

```
{
  user: mininet
  password: mininet
}
```

## 2-CRIANDO TOPOLOGIA DE 6 SWITCHES:

- limpe qualquer topologia anterior: `sudo mn -c`
  - Crie a topologia linear, com largura de banda (bw) = 25 Mbps, endereços MAC padronizados, e controlador padrão do Mininet:  
`sudo mn --topo=linear,6 --mac --link=tc,bw=25`
- `--topo=linear,6` → cria 6 switches conectados em linha.
  - `--mac` → atribui endereços MAC automaticamente (padronizados).
  - `--link=tc,bw=25` → define banda de 25 Mbps entre os nós.
  - Controlador padrão do Mininet é automaticamente usado.

Adicione a imagem do prompt:

[illegible]

### 3 - INSPECIONAR INFORMAÇÕES DE REDE

Após o comando acima, você estará dentro do prompt do Mininet (mininet>).

- Listar nós (hosts e switches): `nodes`
- Ver conexões entre nós: `net`
- Ver detalhes de endereços e interfaces: `dump`
- Inspeccionar informações específicas de um nó: `h1 ifconfig -a`
- Para o switch 1 (s1): `s1 ifconfig -a`

Adicione a imagem do prompt:

```
{
```

Primeiro entramos com um **PINGALL** para testar a rede depois um `nodes`, `net`, `dump` e `h1 ifconfig -a`

```

*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6
h2 -> h1 h3 h4 h5 h6
h3 -> h1 h2 h4 h5 h6
h4 -> h1 h2 h3 h5 h6
h5 -> h1 h2 h3 h4 h6
h6 -> h1 h2 h3 h4 h5
*** Results: 0% dropped (30/30 received)
mininet> nodes
available nodes are:
c0 h1 h2 h3 h4 h5 h6 s1 s2 s3 s4 s5 s6
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s2-eth1
h3 h3-eth0:s3-eth1
h4 h4-eth0:s4-eth1
h5 h5-eth0:s5-eth1
h6 h6-eth0:s6-eth1
s1 lo: s1-eth1:h1-eth0 s1-eth2:s2-eth2
s2 lo: s2-eth1:h2-eth0 s2-eth2:s1-eth2 s2-eth3:s3-eth2
s3 lo: s3-eth1:h3-eth0 s3-eth2:s2-eth3 s3-eth3:s4-eth2
s4 lo: s4-eth1:h4-eth0 s4-eth2:s3-eth3 s4-eth3:s5-eth2
s5 lo: s5-eth1:h5-eth0 s5-eth2:s4-eth3 s5-eth3:s6-eth2
s6 lo: s6-eth1:h6-eth0 s6-eth2:s5-eth3
c0
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=21398>
<Host h2: h2-eth0:10.0.0.2 pid=21400>
<Host h3: h3-eth0:10.0.0.3 pid=21402>
<Host h4: h4-eth0:10.0.0.4 pid=21404>
<Host h5: h5-eth0:10.0.0.5 pid=21406>
<Host h6: h6-eth0:10.0.0.6 pid=21408>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=21413>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None,s2-eth3:None pid=21416>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None,s3-eth3:None pid=21419>
<OVSSwitch s4: lo:127.0.0.1,s4-eth1:None,s4-eth2:None,s4-eth3:None pid=21422>
<OVSSwitch s5: lo:127.0.0.1,s5-eth1:None,s5-eth2:None,s5-eth3:None pid=21425>
<OVSSwitch s6: lo:127.0.0.1,s6-eth1:None,s6-eth2:None pid=21428>
<Controller c0: 127.0.0.1:6653 pid=21391>
mininet> h1 ifconfig -a
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
    inet6 fe80::200:ff:fe00:1 prefixlen 64 scopeid 0x20<link>
    ether 00:00:00:00:00:01 txqueuelen 1000 (Ethernet)
    RX packets 147 bytes 10882 (10.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 33 bytes 2466 (2.4 KB)
    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
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    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

```

Entramos com o **s1 ifconfig -a**

```

mininet> s1 ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1280
    inet 192.168.114.139 netmask 255.255.240.0 broadcast 192.168.127.255
    inet6 fe80::215:5dff:febc:45a6 prefixlen 64 scopeid 0x20<link>
    ether 00:15:5d:bc:45:a6 txqueuelen 1000 (Ethernet)
    RX packets 239945 bytes 235859677 (235.8 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 24402 bytes 1779273 (1.7 MB)
    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
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 7301 bytes 871061 (871.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 7301 bytes 871061 (871.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ovs-system: flags=4098<BROADCAST,MULTICAST> mtu 1500
    ether 6a:d4:7d:ee:a6:6f 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

s1: flags=4098<BROADCAST,MULTICAST> mtu 1500
    ether 56:bb:40:91:5e:42 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

s2: flags=4098<BROADCAST,MULTICAST> mtu 1500
    ether d2:3c:ca:41:ea:41 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

s3: flags=4098<BROADCAST,MULTICAST> mtu 1500
    ether 7e:57:86:c2:f4:4b 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

s4: flags=4098<BROADCAST,MULTICAST> mtu 1500
    ether da:03:ae:f0:6e:41 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

s5: flags=4098<BROADCAST,MULTICAST> mtu 1500
    ether aa:cf:c1:10:eb:44 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

s6: flags=4098<BROADCAST,MULTICAST> mtu 1500

```

Proxima:



```

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s6: flags=4098<BROADCAST,MULTICAST> mtu 1500
    ether ea:dd:09:3b:92:41 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

s1-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::bc2f:95ff:fe84:8c7a prefixlen 64 scopeid 0x20<link>
    ether be:2f:95:84:8c:7a txqueuelen 1000 (Ethernet)
    RX packets 33 bytes 2466 (2.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 147 bytes 10882 (10.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s1-eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::d445:59ff:febe:ad73 prefixlen 64 scopeid 0x20<link>
    ether d6:45:59:be:ad:73 txqueuelen 1000 (Ethernet)
    RX packets 135 bytes 9946 (9.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 43 bytes 3202 (3.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s2-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::d4e9:9cff:fe14:a0f9 prefixlen 64 scopeid 0x20<link>
    ether d6:e9:9c:14:a0:f9 txqueuelen 1000 (Ethernet)
    RX packets 34 bytes 2556 (2.5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 146 bytes 10792 (10.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s2-eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::c864:8cff:fed5:356b prefixlen 64 scopeid 0x20<link>
    ether ca:64:8c:d5:35:6b txqueuelen 1000 (Ethernet)
    RX packets 43 bytes 3202 (3.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 135 bytes 9946 (9.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s2-eth3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::f09c:feff:fe98:40fa prefixlen 64 scopeid 0x20<link>
    ether f2:9c:fe:98:40:fa txqueuelen 1000 (Ethernet)
    RX packets 124 bytes 9168 (9.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 75 bytes 5534 (5.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s3-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::6028:e2ff:fedb:91c9 prefixlen 64 scopeid 0x20<link>
    ether 62:28:e2:db:91:c9 txqueuelen 1000 (Ethernet)
    RX packets 33 bytes 2466 (2.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 148 bytes 10972 (10.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s3-eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::8c63:6cff:fe75:5f50 prefixlen 64 scopeid 0x20<link>
    ether 8e:63:6c:75:5f:50 txqueuelen 1000 (Ethernet)
    RX packets 75 bytes 5534 (5.5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 124 bytes 9168 (9.1 KB)

```

Proxima:

```

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s3-eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::8c63:6cff:fe75:5f50 prefixlen 64 scopeid 0x20<link>
    ether 8e:63:6c:75:5f:50 txqueuelen 1000 (Ethernet)
    RX packets 75 bytes 5534 (5.5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 124 bytes 9168 (9.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s3-eth3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::4cbf:39ff:feeb:a256 prefixlen 64 scopeid 0x20<link>
    ether 4e:bf:39:cb:a2:56 txqueuelen 1000 (Ethernet)
    RX packets 105 bytes 7786 (7.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 103 bytes 7610 (7.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s4-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::4cd6:6fff:fead:ff44 prefixlen 64 scopeid 0x20<link>
    ether 4e:d6:06:ad:ff:44 txqueuelen 1000 (Ethernet)
    RX packets 33 bytes 2466 (2.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 147 bytes 10886 (10.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s4-eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::8441:eaff:fe48:3e4f prefixlen 64 scopeid 0x20<link>
    ether 86:41:ea:48:3e:4f txqueuelen 1000 (Ethernet)
    RX packets 103 bytes 7610 (7.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 105 bytes 7786 (7.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s4-eth3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::4c99:d4ff:fee1:f44c prefixlen 64 scopeid 0x20<link>
    ether 4e:99:d4:e1:f4:4c txqueuelen 1000 (Ethernet)
    RX packets 77 bytes 5710 (5.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 123 bytes 9082 (9.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s5-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::4c75:4dff:fed2:c11a prefixlen 64 scopeid 0x20<link>
    ether 4e:75:4d:d2:c1:1a txqueuelen 1000 (Ethernet)
    RX packets 33 bytes 2466 (2.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 150 bytes 11148 (11.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s5-eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::d827:59ff:fe4e:1ff prefixlen 64 scopeid 0x20<link>
    ether da:27:59:4e:01:ff txqueuelen 1000 (Ethernet)
    RX packets 123 bytes 9082 (9.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 77 bytes 5710 (5.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s5-eth3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::986f:75ff:fe76:8af2 prefixlen 64 scopeid 0x20<link>
    ether 9a:6f:75:76:8a:f2 txqueuelen 1000 (Ethernet)
    RX packets 43 bytes 3202 (3.2 KB)

```

Proxima:

```

s5-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::4c75:4dff:fed2:c11a prefixlen 64 scopeid 0x20<link>
    ether 4e:75:4d:d2:c1:1a txqueuelen 1000 (Ethernet)
    RX packets 33 bytes 2466 (2.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 150 bytes 11148 (11.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s5-eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::d827:59ff:fe4e:1ff prefixlen 64 scopeid 0x20<link>
    ether da:27:59:4e:01:ff txqueuelen 1000 (Ethernet)
    RX packets 123 bytes 9082 (9.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 77 bytes 5710 (5.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s5-eth3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::986f:75ff:fe76:8af2 prefixlen 64 scopeid 0x20<link>
    ether 9a:6f:75:76:8a:f2 txqueuelen 1000 (Ethernet)
    RX packets 43 bytes 3202 (3.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 136 bytes 10036 (10.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s6-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::b85c:81ff:fe75:bff3 prefixlen 64 scopeid 0x20<link>
    ether ba:5c:81:75:bf:f3 txqueuelen 1000 (Ethernet)
    RX packets 34 bytes 2552 (2.5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 147 bytes 10882 (10.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

s6-eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::10f4:b3ff:fe00:702c prefixlen 64 scopeid 0x20<link>
    ether 12:f4:b3:00:70:2c txqueuelen 1000 (Ethernet)
    RX packets 136 bytes 10036 (10.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 43 bytes 3202 (3.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

mininet>

```

1

#### 4 - TESTE DE CONECTIVIDADE (PING)

Esse comando verifica se todos os hosts conseguem se comunicar entre si.

- Testar conectividade entre todos os nós: **Pingall**



- Ping específico entre dois hosts: `h1 ping -c 4 h6` (ex: entre h1 e h6)

Adicione a imagem do prompt:

{

Testando a conectividade e pingando apenas 4

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6
h2 -> h1 h3 h4 h5 h6
h3 -> h1 h2 h4 h5 h6
h4 -> h1 h2 h3 h5 h6
h5 -> h1 h2 h3 h4 h6
h6 -> h1 h2 h3 h4 h5
*** Results: 0% dropped (30/30 received)
mininet> h1 ping -c 4 h6
PING 10.0.0.6 (10.0.0.6) 56(84) bytes of data.
64 bytes from 10.0.0.6: icmp_seq=1 ttl=64 time=7.41 ms
64 bytes from 10.0.0.6: icmp_seq=2 ttl=64 time=0.275 ms
64 bytes from 10.0.0.6: icmp_seq=3 ttl=64 time=0.163 ms
64 bytes from 10.0.0.6: icmp_seq=4 ttl=64 time=0.209 ms

--- 10.0.0.6 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3032ms
rtt min/avg/max/mdev = 0.163/2.015/7.414/3.117 ms
mininet>
```

}

## 5 - TESTE DE DESEMPENHO (IPERF TCP)

Cenário:

- h1 será o servidor TCP na porta 5555
- h2 será o cliente TCP
- Teste de 15 segundos com relatório a cada 1 segundo

- Abra terminais separados para h1 e h2: `xterm h1 h2`

- No terminal do h1 (servidor): `iperf -s -p 5555 -i 1`

Isso cria um **servidor TCP** na porta 5555 e mostra relatórios a cada segundo.

- No terminal do h2 (cliente): `iperf -c 10.0.0.1 -p 5555 -i 1 -t 15`

**Explicando:**

- -c 10.0.0.1 → IP do servidor (h1).
- -p 5555 → porta de comunicação.
- -i 1 → relatório a cada 1 segundo.
- -t 15 → duração total do teste (15 segundos).

**Adicione a imagem do prompt:**

{

Testando o desempenho e criando de acordo com as especificações sem usar GUI apenas no prompt mininet:

```
mininet> h1 iperf -s -p 5555 -i 1 &
mininet> h2 iperf -c 10.0.0.1 -p 5555 -i 1 -t 15
-----
Client connecting to 10.0.0.1, TCP port 5555
TCP window size: 110 KByte (default)
-----
[ 3] local 10.0.0.2 port 42082 connected with 10.0.0.1 port 5555
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0- 1.0 sec  3.12 MBytes 26.2 Mbits/sec
[ 3] 1.0- 2.0 sec  3.00 MBytes 25.2 Mbits/sec
[ 3] 2.0- 3.0 sec  2.88 MBytes 24.1 Mbits/sec
[ 3] 3.0- 4.0 sec  2.88 MBytes 24.1 Mbits/sec
[ 3] 4.0- 5.0 sec  2.88 MBytes 24.1 Mbits/sec
[ 3] 5.0- 6.0 sec  2.88 MBytes 24.1 Mbits/sec
[ 3] 6.0- 7.0 sec  2.75 MBytes 23.1 Mbits/sec
[ 3] 7.0- 8.0 sec  2.75 MBytes 23.1 Mbits/sec
[ 3] 8.0- 9.0 sec  3.00 MBytes 25.2 Mbits/sec
[ 3] 9.0-10.0 sec  2.88 MBytes 24.1 Mbits/sec
[ 3] 10.0-11.0 sec 2.75 MBytes 23.1 Mbits/sec
[ 3] 11.0-12.0 sec 2.75 MBytes 23.1 Mbits/sec
[ 3] 12.0-13.0 sec 3.00 MBytes 25.2 Mbits/sec
[ 3] 13.0-14.0 sec 2.88 MBytes 24.1 Mbits/sec
[ 3] 14.0-15.0 sec 2.75 MBytes 23.1 Mbits/sec
[ 3] 0.0-15.0 sec 43.1 MBytes 24.1 Mbits/sec
mininet>
```

### RESULTADO ESPERADO :

O terminal do h2 mostrará a taxa de transferência média e instantânea (em Mbps) durante os 15 segundos.

O terminal do h1 mostrará os pacotes recebidos e o throughput total.

**Adicione a imagem do prompt:**

{

```

mininet> h2 iperf -c 10.0.0.1 -p 5555 -i 1 -t 15 > relatorio_iperf_h1h2.txt
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s2-eth1
h3 h3-eth0:s3-eth1
h4 h4-eth0:s4-eth1
h5 h5-eth0:s5-eth1
h6 h6-eth0:s6-eth1
s1 lo: s1-eth1:h1-eth0 s1-eth2:s2-eth2
s2 lo: s2-eth1:h2-eth0 s2-eth2:s1-eth2 s2-eth3:s3-eth2
s3 lo: s3-eth1:h3-eth0 s3-eth2:s2-eth3 s3-eth3:s4-eth2
s4 lo: s4-eth1:h4-eth0 s4-eth2:s3-eth3 s4-eth3:s5-eth2
s5 lo: s5-eth1:h5-eth0 s5-eth2:s4-eth3 s5-eth3:s6-eth2
s6 lo: s6-eth1:h6-eth0 s6-eth2:s5-eth3
c0
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=21398>
<Host h2: h2-eth0:10.0.0.2 pid=21400>
<Host h3: h3-eth0:10.0.0.3 pid=21402>
<Host h4: h4-eth0:10.0.0.4 pid=21404>
<Host h5: h5-eth0:10.0.0.5 pid=21406>
<Host h6: h6-eth0:10.0.0.6 pid=21408>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=21413>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None,s2-eth3:None pid=21416>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None,s3-eth3:None pid=21419>
<OVSSwitch s4: lo:127.0.0.1,s4-eth1:None,s4-eth2:None,s4-eth3:None pid=21422>
<OVSSwitch s5: lo:127.0.0.1,s5-eth1:None,s5-eth2:None,s5-eth3:None pid=21425>
<OVSSwitch s6: lo:127.0.0.1,s6-eth1:None,s6-eth2:None pid=21428>
<Controller c0: 127.0.0.1:6653 pid=21391>
mininet>

```

## 6 – RELATORIOS E ENCERRAMENTO

- Ver estatísticas adicionais: `net, dump`

- Salvar resultados (opcional): `iperf -c 10.0.0.1 -p 5555 -i 1 -t 15 > relatorio_iperf.txt`

Você pode redirecionar a saída para um arquivo:

- Encerrar e limpar tudo: `sudo mn -c`

Adicione a imagem do prompt:

```
{
```

```

mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 11 links
.....
*** Stopping 6 switches
s1 s2 s3 s4 s5 s6
*** Stopping 6 hosts
h1 h2 h3 h4 h5 h6
*** Done
completed in 1825.149 seconds
unix2004@DESKTOP-C1PDWV7:~/patinho/mininet$ sudo mn -c
[sudo] password for unix2004:
*** Removing excess controllers/ofprotocols/ofdatapaths/pings/noxes
killall controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd ovs-controller ovs-testcontroller udbwtest mnexec ivs ryu-manager 2> /dev/null
killall -9 controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd ovs-controller ovs-testcontroller udbwtest mnexec ivs ryu-manager 2> /dev/null
pkill -9 -f "sudo mnexec"
*** Removing junk from /tmp
rm -f /tmp/vconn* /tmp/vlogs* /tmp/*.out /tmp/*.log
*** Removing old X11 tunnels
*** Removing excess kernel datapaths
ps ax | egrep -o 'dp[0-9]+' | sed 's/dp/nl:/'
*** Removing OVS datapaths
ovs-vsctl --timeout=1 list-br
ovs-vsctl --timeout=1 list-br
*** Removing all links of the pattern foo-ethX
ip link show | egrep -o '([_[:alnum:]]+-eth[[:digit:]]+)'
ip link show
*** Killing stale mininet node processes
pkill -9 -f mininet:
*** Shutting down stale tunnels
pkill -9 -f Tunnel=Ethernet
pkill -9 -f .ssh/mn
rm -f ~/.ssh/mn/*
*** Cleanup complete.
unix2004@DESKTOP-C1PDWV7:~/patinho/mininet$

```

1

## VISUALIZAÇÃO E DIAGNOSTICOS:

Para capturar pacotes e verificar fluxos:

wireshark &

sudo ovs-ofctl dump-flows s1

Adicione a imagem do prompt:

{

O **Open vSwitch (OVS)**, que o Mininet usa, já tem comandos poderosos de inspeção e estatísticas via terminal.

Você consegue ver tudo que interessa sobre **fluxos, pacotes e portas** diretamente pelo **ovs-ofctl** e **ovs-vsctl**.

**o ideal é abrir outro terminal Bash** (ou outra aba) **para monitorar os fluxos e estatísticas**, enquanto o seu teste (iperf, ping, etc.) roda dentro do terminal onde o **Mininet** está ativo.

O **Mininet** roda de forma interativa dentro de um terminal —

então, se você quiser rodar comandos de diagnóstico do sistema (como **ovs-ofctl dump-flows s1**), **precisa sair ou usar outro terminal**.

Como **ovs-ofctl** e **ovs-vsctl** são comandos do **sistema host (Ubuntu/WSL)** e não do shell do Mininet, eles devem ser executados **fora** do prompt **mininet>**.

```
mininet> h2 iperf -c 10.0.0.1 -p 5555 -i 1 -t 15 > relatorio_iperf_h1h2.txt
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s2-eth1
h3 h3-eth0:s3-eth1
h4 h4-eth0:s4-eth1
h5 h5-eth0:s5-eth1
h6 h6-eth0:s6-eth1
s1 lo: s1-eth1:h1-eth0 s1-eth2:s2-eth2
s2 lo: s2-eth1:h2-eth0 s2-eth2:s1-eth2 s2-eth3:s3-eth2
s3 lo: s3-eth1:h3-eth0 s3-eth2:s2-eth2 s3-eth3:s4-eth2
s4 lo: s4-eth1:h4-eth0 s4-eth2:s3-eth3 s4-eth3:s5-eth2
s5 lo: s5-eth1:h5-eth0 s5-eth2:s4-eth3 s5-eth3:s6-eth2
s6 lo: s6-eth1:h6-eth0 s6-eth2:s5-eth3
o0
mininet> dump
OHost h1: h1-eth0:10.0.0.1 pid=21390>
OHost h2: h2-eth0:10.0.0.2 pid=21400>
OHost h3: h3-eth0:10.0.0.3 pid=21402>
OHost h4: h4-eth0:10.0.0.4 pid=21404>
OHost h5: h5-eth0:10.0.0.5 pid=21406>
OHost h6: h6-eth0:10.0.0.6 pid=21408>
OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=21413>
OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None,s2-eth3:None pid=21416>
OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None,s3-eth3:None pid=21419>
OVSSwitch s4: lo:127.0.0.1,s4-eth1:None,s4-eth2:None,s4-eth3:None pid=21422>
OVSSwitch s5: lo:127.0.0.1,s5-eth1:None,s5-eth2:None,s5-eth3:None pid=21425>
OVSSwitch s6: lo:127.0.0.1,s6-eth1:None,s6-eth2:None,s6-eth3:None pid=21428>
Controller c0: 127.0.0.1:6653 pid=21391>
mininet> h2 iperf -c 10.0.0.1 -p 5555 -i 1 -t 15
-----
Client connecting to 10.0.0.1, TCP port 5555
TCP window size: 128 KByte (default)
-----
[ ] local 10.0.0.2 port 44514 connected with 10.0.0.1 port 5555
[0] Interval Transfer Bandwidth
[ ] 0.0- 1.0 sec 3.25 MBytes 27.3 Mbits/sec
[ ] 1.0- 2.0 sec 2.75 MBytes 23.1 Mbits/sec
[ ] 2.0- 3.0 sec 3.00 MBytes 25.2 Mbits/sec
[ ] 3.0- 4.0 sec 2.75 MBytes 23.1 Mbits/sec
[ ] 4.0- 5.0 sec 2.88 MBytes 24.1 Mbits/sec
[ ] 5.0- 6.0 sec 2.88 MBytes 24.1 Mbits/sec
[ ] 6.0- 7.0 sec 2.88 MBytes 24.1 Mbits/sec
[ ] 7.0- 8.0 sec 2.75 MBytes 23.1 Mbits/sec
[ ] 8.0- 9.0 sec 2.88 MBytes 24.1 Mbits/sec
[ ] 9.0-10.0 sec 2.88 MBytes 24.1 Mbits/sec
[ ] 10.0-11.0 sec 2.88 MBytes 24.1 Mbits/sec
[ ] 11.0-12.0 sec 2.75 MBytes 23.1 Mbits/sec
[ ] 12.0-13.0 sec 3.00 MBytes 25.2 Mbits/sec
[ ] 13.0-14.0 sec 2.75 MBytes 23.1 Mbits/sec
[ ] 14.0-15.0 sec 2.88 MBytes 24.1 Mbits/sec
[ ] 0.0-15.0 sec 43.1 MBytes 24.1 Mbits/sec
mininet>
```

```
unix2004@DESKTOP-CIPDMV7: ~$ sudo ovs-ofctl dump-flows s1
Every 1.0s: sudo ovs-ofctl dump-flows s1
DESKTOP-CIPDMV7: Tue Oct 21 13:44:23 2025

NXST_FLOW reply (xid=0x4):
cookie=0x0, duration=15.549s, table=0, n_packets=3497, n_bytes=230910, idle_timeout=60, idle_age=0, priority=65535, tcp,in_port=1,vlan_tci=0x0000,dl_src=00:00:00:00:01,dl_dst=00:00:00:00:02,nw_src=10.0.0.1,nw_dst=10.0.0.2,nw_tos=0,tp_src=5555,tp_dst=44514 actions=output:2
cookie=0x0, duration=15.538s, table=0, n_packets=3486, n_bytes=45435436, idle_timeout=60, idle_age=0, priority=65535, tcp,in_port=2,vlan_tci=0x0000,dl_src=00:00:00:00:02,dl_dst=00:00:00:00:01,nw_src=10.0.0.2,nw_dst=10.0.0.1,nw_tos=0,tp_src=44514,tp_dst=5555 actions=output:1
cookie=0x0, duration=10.437s, table=0, n_packets=0, n_bytes=0, idle_timeout=60, idle_age=10, priority=65535, arp,in_port=1,vlan_tci=0x0000,dl_src=00:00:00:00:01,dl_dst=00:00:00:00:02,arp_spa=10.0.0.1,arp_tpa=10.0.0.2,arp_op=1 actions=output:2
cookie=0x0, duration=10.416s, table=0, n_packets=0, n_bytes=0, idle_timeout=60, idle_age=10, priority=65535, arp,in_port=2,vlan_tci=0x0000,dl_src=00:00:00:00:02,dl_dst=00:00:00:00:01,arp_spa=10.0.0.2,arp_tpa=10.0.0.1,arp_op=2 actions=output:1
```

a imagem mostra **exatamente o comportamento esperado** de um teste de desempenho TCP (via `iperf`) e de monitoramento de fluxos (`ovs-ofctl`) em uma topologia Mininet.

“ `cookie=0x0, duration=15.549s, table=0, n_packets=3497, n_bytes=230910, idle_timeout=60, priority=65535, tcp,in_port=1,tp_src=5555,tp_dst=44514,actions=output:2` “

## Interpretação dos principais campos:

Campo	Significado
<code>duration=15.549s</code>	Regra ativa há 15 segundos (duração do teste)
<code>n_packets=3497</code>	Número de pacotes que passaram por esta regra
<code>n_bytes=230910</code>	Total de bytes transmitidos
<code>in_port=1</code>	Interface de entrada (porta física ou virtual)
<code>actions=output:2</code>	Ação do switch (encaminhar pela porta 2)
<code>priority=65535</code>	Prioridade da regra (alta, geralmente padrão para fluxos instalados)



## Resumo Rápido de Comandos

Etapa	Comando	Função
Limpar rede	<code>sudo mn -c</code>	Limpa topologia antiga
Criar topologia	<code>sudo mn --topo=linear,6 --mac --link=tc,bw=25</code>	Cria rede linear com 6 switches
Ver nós	<code>nodes</code>	Lista hosts e switches
Ver conexões	<code>net</code>	Mostra ligações entre os nós
Inspecionar rede	<code>dump</code>	Mostra IPs e interfaces
Testar ping	<code>pingall</code>	Testa conectividade entre todos
Servidor TCP	<code>iperf -s -p 5555 -i 1</code>	Roda no h1
Cliente TCP	<code>iperf -c 10.0.0.1 -p 5555 -i 1 -t 15</code>	Roda no h2
Fechar tudo	<code>exit e sudo mn -c</code>	Sai e limpa a simulação