

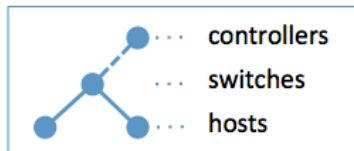
Conceitos e Tecnologias para Dispositivos Conectados (C115)

Prof. Samuel Baraldi Mafra



Mininet

```
> sudo mn
```



<http://mininet.org/>

- O Mininet é um emulador de rede que cria redes com servidores, switches, controladores e enlaces virtuais.
- O Mininet cria uma rede virtual realística rodando em um núcleo real (Kernel Linux), switches e códigos de aplicação em uma única máquina física ou virtual (Virtual Machine - VM) que pode ser nativa ou em nuvem (Cloud).
- O Mininet possui linha de comando própria (Command Line Interface - CLI) e APIs, que permitem a criação de rede e serviços, customização e compartilhamento com outros usuários, e também a implantação em hardware real.

- O Mininet é ideal para experimentos com OpenFlow e Redes Definidas por Software (Software Defined Networks - SDN).

O Mininet combina muitos dos melhores recursos de emuladores, testbeds de hardware e simuladores.

- Comparado com abordagens baseadas em virtualização de sistema completo, Mininet:
 - Arranca mais rápido : segundos em vez de minutos
 - Escalas maiores : centenas de hosts e switches vs. dígitos únicos
 - Oferece mais largura de banda : normalmente 2 Gbps de largura de banda total em hardware modesto
 - Instala facilmente : uma VM predefinida está disponível para ser executada em VMware ou VirtualBox para Mac / Win / Linux com ferramentas OpenFlow v1.0 já instaladas.

- Comparado com testbeds de hardware, Mininet
- é barato e está sempre disponível (mesmo antes dos prazos das conferências)
- é rapidamente reconfigurável e reiniciável.

- Comparado com simuladores, Mininet
- executa código real não modificado, incluindo código do aplicativo, código do kernel do sistema operacional e código do plano de controle (código do controlador OpenFlow e código Open vSwitch)
- conecta-se facilmente a redes reais
- oferece desempenho interativo - você pode digitar nele

- As principais limitações das redes emuladas no Mininet referem-se à capacidade de banda disponível e CPU que não podem exceder à capacidade e banda disponível no servidor onde o Mininet está instalado.
- Instalação de somente programas linux.

Instalação:

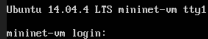
- Baixar a imagem Mininet ubuntu de acordo com o seu pc 32 ou 64 bits em
<https://github.com/mininet/mininet/releases>;
- Baixar e instalar o virtualbox em
<https://www.virtualbox.org/wiki/Downloads>
- Instalar o mininet na máquina virtual, sigam os passos deste vídeo: <https://youtu.be/TGmh6ppSS3M?list=FL8rCt095hxrCVZZRilHSuA>
- Instalar a interface gráfica usando Xming pelo link
<https://sourceforge.net/projects/xming/>;
- Instalar o Putty pelo link <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>
- Seguir as instruções do link
<https://youtu.be/RTlituHdp7M> para acessar a máquina virtual pelo putty e xterm

Comandos básicos em linux:

- sudo: Executar com privilégios de administrador;
- sudo apt-get install: instalar aplicativo;
- sudo apt-get update: Atualizar repositórios;
- ls: Mostrar pasta e arquivos;
- mkdir: Criar uma pasta;
- cd: ir até uma determinada pasta.
- cd ..: Retorna diretório acima.
- nano: edição de arquivos.

Inicialização

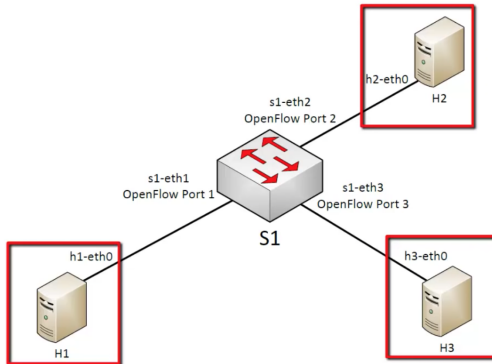
- Usuário e senha:
- Acesse a console e use as seguintes credenciais:
- user: mininet
- password: mininet

A screenshot of a terminal window with a black background and white text. The text shows the system booting into Ubuntu 14.04.4 LTS as the user 'mininet-vm' on the terminal 'tty1'. It then prompts for a login.

```
Ubuntu 14.04.4 LTS mininet-vm tty1
mininet-vm login:
```

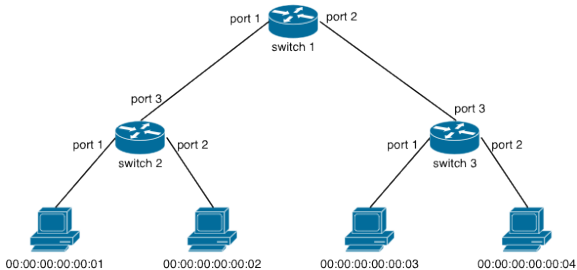
Topologias

`sudo mn --topo=single,3`

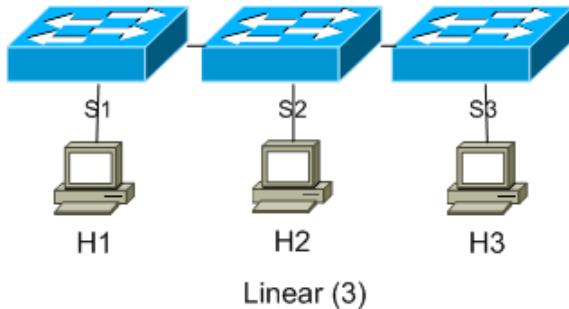


Provavelmente essa será usada pra atividade

`sudo mn --topo=tree,depth=2,fanout=2`



```
sudo mn --topo=linear,3
```

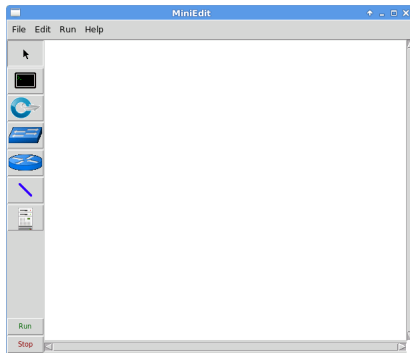


Controladores:

- o mininet inicializa seu controlador padrão e se conecta ao controlador em localhost: 6633.
- ovsc;
- nox;
- remote para rodar um controlador remoto como pox.
- sudo mn -controller=remote;

miniedit:

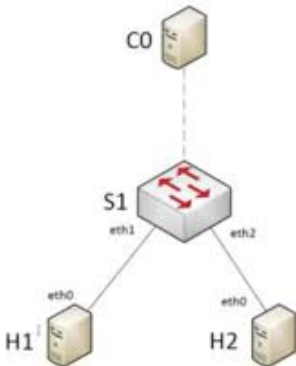
- Um editor GUI simples para Mininet.
- MiniEdit é uma ferramenta experimental criado para demonstrar como Mininet pode ser estendido;
- Importar código python.



Executar `sudo ~/mininet/examples/miniedit.py`

Mininet

- `sudo mn`: executa o mininet em uma configuração exemplo.



```
Ubuntu 14.04.4 LTS mininet-vm tty1
mininet-vm login: mininet
Password:
Last login: Tue Apr 13 18:41:12 PDT 2021 on tty1
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 4.2.0-27-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
mininet@mininet-vm:~$ sudo mn
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> mininet@mininet-vm:~$
```

Comandos básicos;

- nodes: Mostra os nós na rede;
- net : Verificar os enlaces;
 - As portas são simbolizadas por eth. No exemplo abaixo a porta 0 de h1 está conectada a porta 1 de s1.

```
mininet> nodes  
available nodes are:  
c0 h1 h2 s1
```

```
mininet> net  
h1 h1-eth0:s1-eth1  
h2 h2-eth0:s1-eth2  
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0  
c0
```

Comandos básicos;

- dump: Verificando os endereços lógicos dos dispositivos da rede. Os nós são configurados em uma subrede padrão (default) de IP 10.0.0.0/8;

```
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=1586>
<Host h2: h2-eth0:10.0.0.2 pid=1588>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=1593>
<Controller c0: 127.0.0.1:6653 pid=1579>
```

Comandos básicos;

- h1 ifconfig -a: Mostra os dados do host h1;
- s1 ifconfig -a: Mostra os dados do switch s1;

```
mininet> h1 ifconfig -a
h1-eth0  Link encap:Ethernet  HWaddr 9c:31:96:b5:21:3a
         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)
```

HWaddr -> mac gerado aleatoriamente

```
s1
s1-eth1  Link encap:Ethernet  HWaddr 76:02:5f:d7:d4:44
         UP BROADCAST RUNNING  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:0
         RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

s2
s2-eth1  Link encap:Ethernet  HWaddr 3e:40:14:27:5d:41
         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:0
         RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

s3
s3-eth1  Link encap:Ethernet  HWaddr 1a:7d:b0:c8:10:11
         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:0
         RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

s1-eth1  Link encap:Ethernet  HWaddr 8a:60:7d:ce:04:f9
         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)

s1-eth2  Link encap:Ethernet  HWaddr c2:e3:95:b6:a7:1b
         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)
```

Comandos básicos;

- pingall: Faz ping com todos os nós da rede;
- h1 ping h2:
- h1 ping -c 5 h2: 5 pings

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
```

```
mininet> h1 ping 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=1.19 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.926 ns
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.070 ns
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.615 ns
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.067 ns

64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.053 ns
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=0.922 ns
^C
-- 10.0.0.2 ping statistics --
7 packets transmitted, 7 received, 0% packet loss, time 6002ms
rtt min/avg/max/ndev = 0.053/0.549/1.192/0.448 ms
```

Comandos básicos;

- link h1 s1 down: derrubar link;
- link h1 s1 up: conectar link.
- sudo mn -c: Limpa a última conexão. [Executar um exit antes](#)

```
mininet> link h1 s1 down
mininet> pingall
*** Ping: testing ping reachability
h1 -> X
h2 -> X
*** Results: 100% dropped (0/2 received)
mininet> link h1 s1 up
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
```

Comandos básicos;

- `sudo mn --link tc,bw=[bandwidth],delay=[delay_in_millisecond];` Fazer uma conexão com largura de banda e atraso definidos.
- `sudo mn --link tc,bw=10,delay=10ms`
Executar esse comando todo, menos o delay para definir a bw

Testes Executar o 'sudo mn -c' antes dos teste

- sudo mn --test pingpair: Este comando cria a topologia Mininet e realiza um teste de ping entre os nós e encerra;

```
mininet@mininet-vm:~$ sudo mn --test pingpair
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 5.269 seconds
```

Testes

- `sudo mn --test iperf`: Este comando cria a topologia Mininet, executa um servidor iperf em um host, executa um cliente iperf no segundo host e analisa a largura de banda alcançada.

```
mininet@mininet-vn:~$ sudo mn --test iperf
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['38.4 Gbits/sec', '38.5 Gbits/sec']
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 11.108 seconds
mininet@mininet-vn:~$ _
```

Testes

- `sudo mn --test none`: Mostra o tempo para configurar e terminar uma topologia;

```
mininet@mininet-vm:~$ sudo mn --test none
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 0.491 seconds
mininet@mininet-vm:~$
```

caso a interface gráfica não carregue usar o comando `sudo -E mn`

Opções:

- `sudo mn --mac`: Fixa a geração dos endereços MAC para valores mais fáceis de analisar;
- Fazer teste com `ifconfig`: `h1 ifconfig`

```
mininet@mininet-vm:~$ sudo mn --mac
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> h1 ifconfig
h1-eth0  Link encap:Ethernet  HWaddr ee:cfa6:e0:77:44
          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>
```

```
mininet@mininet-vm:~$ sudo mn --mac
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> h1 ifconfig
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)

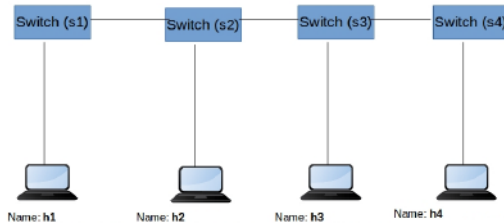
mininet>
```

Análise com uso de Xming e Putty

- Abrir a máquina virtual;
- Abrir o xming;
- Abrir o putty.

- xterm h1: Abre o terminal do usuário h1;
 - Este terminal funciona como um terminal linux.
- wireshark &: Abre o wireshark.
- sudo ~ /mininet/examples/miniedit.py: Abre o miniedit

```
sudo mn --topo=linear,4 --link tc,bw=10
```



- Comando iperf: Utilizado para fazer análises de desempenho no mininet.
 - xterm h1 h2
 - servidor TCP h2: `iperf -s -p 5555 -i 1` definindo h2 como server
 - Cliente h1:
 - `iperf -c 10.0.0.2 -p 5555 -i 1 -t 15` h1 como cliente
- - c significa modo cliente.
- -i significa intervalo de relatório
- -p porta do servidor
- -t significa duração do teste em segundos

Node: h1

root@mininet-vw:~# iperf -c 10.0.0.2 -p 5555 -i 1 -t 15

Client connecting to 10.0.0.2, TCP port 5555
TCP window size: 85.3 KByte (default)

[ID]	Interval	Transfer	Bandwidth
[23]	0.0- 1.0 sec	1.50 MBytes	12.6 Mbits/sec
[23]	1.0- 2.0 sec	1.12 MBytes	9.44 Mbits/sec
[23]	2.0- 3.0 sec	1.12 MBytes	9.44 Mbits/sec
[23]	3.0- 4.0 sec	1.12 MBytes	9.44 Mbits/sec
[23]	4.0- 5.0 sec	1.25 MBytes	10.5 Mbits/sec
[23]	5.0- 6.0 sec	1.12 MBytes	9.44 Mbits/sec
[23]	6.0- 7.0 sec	1.12 MBytes	9.44 Mbits/sec
[23]	7.0- 8.0 sec	1.12 MBytes	9.44 Mbits/sec
[23]	8.0- 9.0 sec	1.00 MBytes	8.39 Mbits/sec
[23]	9.0-10.0 sec	1.12 MBytes	9.44 Mbits/sec
[23]	10.0-11.0 sec	1.25 MBytes	10.5 Mbits/sec
[23]	11.0-12.0 sec	1.12 MBytes	9.44 Mbits/sec
[23]	12.0-13.0 sec	1.12 MBytes	9.44 Mbits/sec
[23]	13.0-14.0 sec	1.12 MBytes	9.44 Mbits/sec
[23]	14.0-15.0 sec	1.12 MBytes	9.44 Mbits/sec
[23]	0.0-15.1 sec	17.5 MBytes	9.71 Mbits/sec

root@mininet-vw:~#

Node: h2

root@mininet-vw:~# iperf -s -p 5555 -i 1

Server listening on TCP port 5555
TCP window size: 85.3 KByte (default)

[ID]	Interval	Transfer	Bandwidth
[24]	0.0- 1.0 sec	1.14 MBytes	9.59 Mbits/sec
[24]	1.0- 2.0 sec	1.14 MBytes	9.55 Mbits/sec
[24]	2.0- 3.0 sec	1.14 MBytes	9.57 Mbits/sec
[24]	3.0- 4.0 sec	1.14 MBytes	9.56 Mbits/sec
[24]	4.0- 5.0 sec	1.14 MBytes	9.59 Mbits/sec
[24]	5.0- 6.0 sec	1.14 MBytes	9.52 Mbits/sec
[24]	6.0- 7.0 sec	1.15 MBytes	9.63 Mbits/sec
[24]	7.0- 8.0 sec	1.14 MBytes	9.52 Mbits/sec
[24]	8.0- 9.0 sec	1.14 MBytes	9.52 Mbits/sec
[24]	9.0-10.0 sec	1.15 MBytes	9.63 Mbits/sec
[24]	10.0-11.0 sec	1.14 MBytes	9.52 Mbits/sec
[24]	11.0-12.0 sec	1.14 MBytes	9.59 Mbits/sec
[24]	12.0-13.0 sec	1.14 MBytes	9.57 Mbits/sec
[24]	13.0-14.0 sec	1.14 MBytes	9.52 Mbits/sec
[24]	14.0-15.0 sec	1.14 MBytes	9.59 Mbits/sec
[24]	0.0-15.3 sec	17.5 MBytes	9.56 Mbits/sec

root@mininet-vw:~#

- UDP
- xterm h1 h2
 - servidor UDP h2: `iperf -u -s -p 5555 -i 1`
 - Cliente h1: `iperf -u-c 10.0.0.2 -p 5555 -i 1 -t 15`
- u refere-se a UDP.

Node: h2

```

Server listening on UDP port 5555
Receiving 1470 byte datagrams
UDP buffer size: 208 KByte (default)
-----
[ 23] local 10.0.0.2 port 5555 connected with 10.0.0.1 port 35952
[ 10] Interval      Transfer      Bandwidth      Jitter      Lost/Total Datagrams
[ 23] 0.0- 1.0 sec  128 KBytes  1.05 Mbits/sec  0.011 ms    0/ 89 (0%)
[ 23] 1.0- 2.0 sec  128 KBytes  1.05 Mbits/sec  0.009 ms    0/ 89 (0%)
[ 23] 2.0- 3.0 sec  128 KBytes  1.05 Mbits/sec  0.011 ms    0/ 89 (0%)
[ 23] 3.0- 4.0 sec  128 KBytes  1.05 Mbits/sec  0.007 ms    0/ 89 (0%)
[ 23] 4.0- 5.0 sec  129 KBytes  1.06 Mbits/sec  0.018 ms    0/ 90 (0%)
[ 23] 5.0- 6.0 sec  128 KBytes  1.05 Mbits/sec  0.012 ms    0/ 89 (0%)
[ 23] 6.0- 7.0 sec  128 KBytes  1.05 Mbits/sec  0.010 ms    0/ 89 (0%)
[ 23] 7.0- 8.0 sec  128 KBytes  1.05 Mbits/sec  0.014 ms    0/ 89 (0%)
[ 23] 8.0- 9.0 sec  128 KBytes  1.05 Mbits/sec  0.014 ms    0/ 89 (0%)
[ 23] 9.0-10.0 sec  129 KBytes  1.06 Mbits/sec  0.009 ms    0/ 90 (0%)
[ 23] 10.0-11.0 sec 128 KBytes  1.05 Mbits/sec  0.013 ms    0/ 89 (0%)
[ 23] 11.0-12.0 sec 128 KBytes  1.05 Mbits/sec  0.015 ms    0/ 89 (0%)
[ 23] 12.0-13.0 sec 128 KBytes  1.05 Mbits/sec  0.014 ms    0/ 89 (0%)
[ 23] 13.0-14.0 sec 128 KBytes  1.05 Mbits/sec  0.007 ms    0/ 89 (0%)
[ 23] 14.0-15.0 sec 128 KBytes  1.05 Mbits/sec  0.007 ms    0/ 89 (0%)
[ 23] 0.0-15.0 sec 1.88 MBytes  1.05 Mbits/sec  0.009 ms    0/ 1339 (0%)
root@mininet-vn:~#

```

Node: h1

```

UDP buffer size: 208 KByte (default)
-----
[ 23] local 10.0.0.1 port 35952 connected with 10.0.0.2 port 5555
[ 10] Interval      Transfer      Bandwidth
[ 23] 0.0- 1.0 sec  128 KBytes  1.06 Mbits/sec
[ 23] 1.0- 2.0 sec  128 KBytes  1.05 Mbits/sec
[ 23] 2.0- 3.0 sec  128 KBytes  1.05 Mbits/sec
[ 23] 3.0- 4.0 sec  128 KBytes  1.05 Mbits/sec
[ 23] 4.0- 5.0 sec  128 KBytes  1.05 Mbits/sec
[ 23] 5.0- 6.0 sec  128 KBytes  1.05 Mbits/sec
[ 23] 6.0- 7.0 sec  129 KBytes  1.06 Mbits/sec
[ 23] 7.0- 8.0 sec  128 KBytes  1.05 Mbits/sec
[ 23] 8.0- 9.0 sec  128 KBytes  1.05 Mbits/sec
[ 23] 9.0-10.0 sec  128 KBytes  1.05 Mbits/sec
[ 23] 10.0-11.0 sec 128 KBytes  1.05 Mbits/sec
[ 23] 11.0-12.0 sec 128 KBytes  1.05 Mbits/sec
[ 23] 12.0-13.0 sec 129 KBytes  1.06 Mbits/sec
[ 23] 13.0-14.0 sec 128 KBytes  1.05 Mbits/sec
[ 23] 14.0-15.0 sec 128 KBytes  1.05 Mbits/sec
[ 23] 0.0-15.0 sec 1.88 MBytes  1.05 Mbits/sec
[ 23] Sent 1339 datagrams
[ 23] Server Report:
[ 23] 0.0-15.0 sec  1.88 MBytes  1.05 Mbits/sec  0.009 ms    0/ 1339 (0%)
root@mininet-vn:~#

```

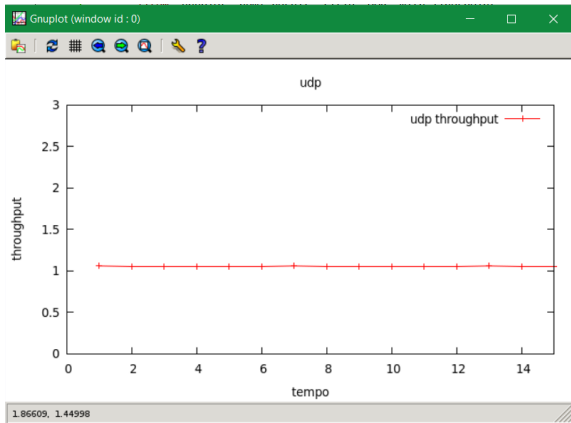
gnuplot

- Obter gráficos;
- `sudo apt-get update`
- `sudo apt-get install gnuplot-x11`
- Confirmar com y (yes) quando questionado.

<http://www.gnuplot.info/>

- `sudo mn --topo=linear,4 --link tc,bw=10`
- `xterm h1 h2`
 - servidor UDP h2: `iperf -u -s -p 5555 -i 1`
 - Cliente h1: `iperf -u-c 10.0.0.2 -p 5555 -i 1 -t 15 > result`
 - `cat result`
 - `cat result | grep sec | head -15 | tr - " " | awk '{print $4,$8}' > new_result`
 - pegue o arquivo result;
 - procure pelas linhas com sec;
 - pegue as 15 primeiras
 - retire as aspas
 - imprima as colunas 4 e 8 em new_result
 - `cat new_result`

- gnuplot
- plot "new_result" title "UDP" with linespoint
- set title "udp"
- replot
- set xlabel "tempo"
- set ylabel "throughput"
- replot
- set xrange[0:15]
- set yrange[0:3]
- replot



Trabalho: Considere uma topologia tree com profundidade (depth=4) e ramificação (fanout=2).

- Com uso de linha de comando padrão do Mininet, crie a topologia considerando o endereço MAC padronizado, larguras de banda bw de 25Mbps e controlador do Mininet (não precisa especificar);
- Inspeccione informações das interfaces, endereços MAC, IP e portas através de linhas de comando;
- Execute testes de ping entre os diferentes nós.
- 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 25 segundos. Faça os testes para larguras de banda bw de 10 Mbps