EA080: Laboratório 03

Vinícius Esperança Mantovani,
247395.

Introdução)

Neste experimento, tem-se como objetivos o aprofundamento de conhecimentos relacionados às redes sem-fio, a introdução e familiarização com o *mininet-wifi* para tal fim e, a diferenciação do *mininet-wifi* comparado ao *mininet*.

Exercício 1)

Para iniciar o experimento, foi executado o comando "*sudo mn --wifi*" para iniciar uma rede de topologia apresentada na *Figura 1*. Em seguida, os comandos disponíveis do mininet-wifi foram reconhecidos por meio do comando help e, foram visualizados os nós da rede, usando-se "*nodes*". Assim, foram obtidas as *Figuras 2* e *3*.

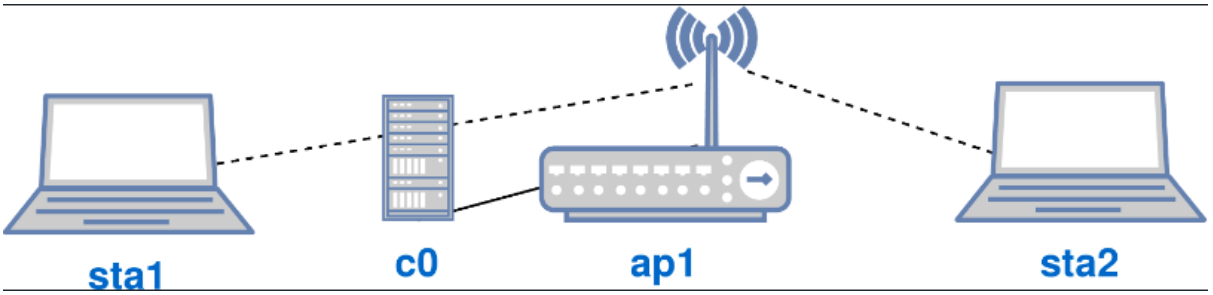

Figura 1: Topologia da rede wifi utilizada

```
Documented commands (type help <topic>):
========================================
EOF       exit   iperf     net      pingallfull   px    source  time
distance  gterm  iperfudp  nodes    pingpair      py    start   x
dpctl     help   link      noecho   pingpairfull  quit  stop    xterm
dump      intfs  links     pingall  ports         sh    switch
```

Figura 2: Comandos mostrados pelo comando "help"

```
mininet-wifi> nodes
available nodes are:
ap1 c0 sta1 sta2
```

Figura 3: Apresentação dos nós da rede por comando "nodes"

Após os passos iniciais, foram executados comandos para visualização de informações dos nós *sta1* e *sta2*, conforme se nota nas *Figuras 4* e *5* abaixo. Além disso, é exibido o resultado de um comando *ping* de *sta1* para *sta2*.

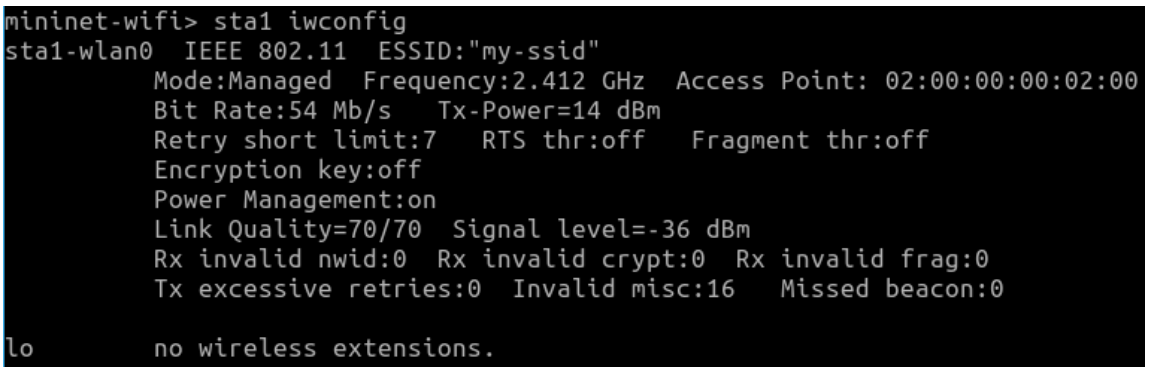

Figura 4: Informações a respeito de sta1 dadas pelo comando iwconfig nesse nó

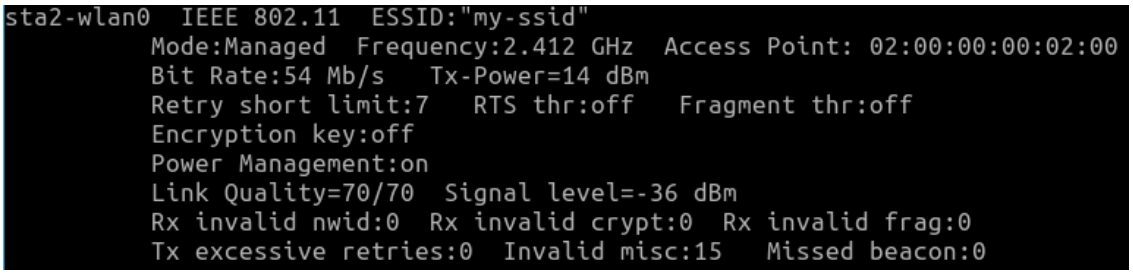

Figura 5: Informações a respeito de sta1 dadas pelo comando iwconfig nesse nó

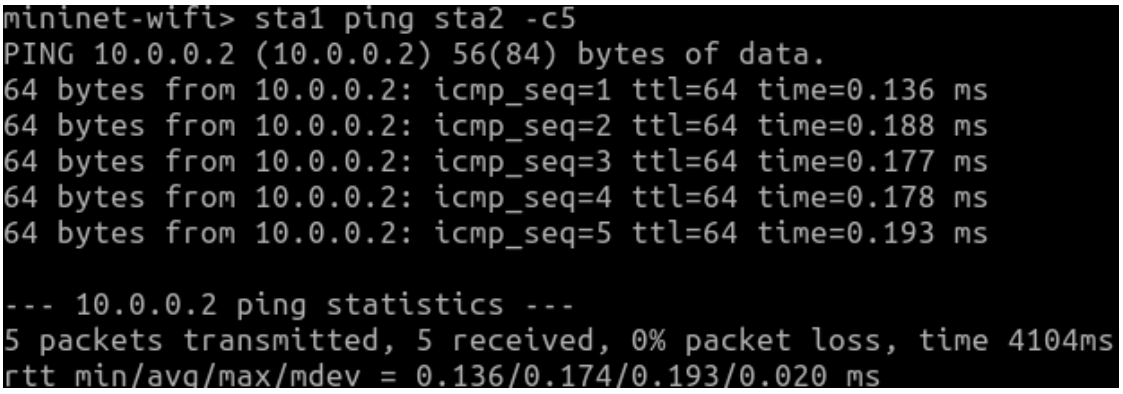

Figura 6: Resultado do ping de sta1 para sta2

Como alternativa para o acesso às informações sobre os nós, pôde-se usar, a critério de conhecimento, o comando "*sta1 iw dev sta1-wlan0 info*" cuja resposta é apresentada na *Figura 7* que se segue. No entanto, deste último modo, o número de informações apresentadas é menor que no primeiro.

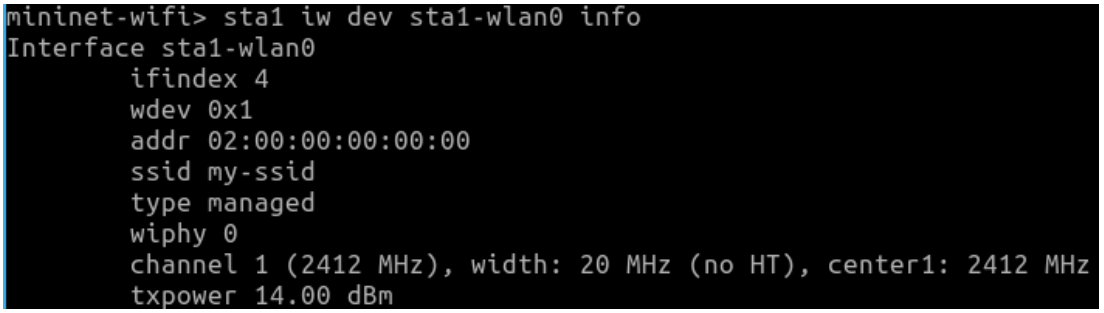

Figura 7: Resultado do comando sta1 iw dev sta1-wlan0 info

Em sequência ao exposto, o nó *sta1* foi desconectado do ponto de acesso, utilizando-se o comando "*sta1 iw dev sta1-wlan0 disconnect*" e, sua desconexão foi verificada por meio dos comandos *iwconfig* exposto anteriormente e de um *ping* de *sta1* para *sta2*, conforme se vê na seguinte figura. Nesta figura, após o comando de desconexão, na resposta ao *iwconfig*, tem-se a resposta à seção "*Access Point*" como "Not-Associated"

```
mininet-wifi> sta1 iw dev sta1-wlan0 disconnect
mininet-wifi> sta1 iwconfig
sta1-wlan0  IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:on

lo        no wireless extensions.

mininet-wifi> sta1 ping sta2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable
From 10.0.0.1 icmp_seq=2 Destination Host Unreachable
From 10.0.0.1 icmp_seq=3 Destination Host Unreachable
From 10.0.0.1 icmp_seq=4 Destination Host Unreachable
From 10.0.0.1 icmp_seq=5 Destination Host Unreachable
From 10.0.0.1 icmp_seq=6 Destination Host Unreachable
^C
--- 10.0.0.2 ping statistics ---
7 packets transmitted, 0 received, +6 errors, 100% packet loss, time 6123ms
pipe 4
```

Figura 8: Desconexão e verificação do nó sta1

Para continuar o experimento, após o teste de desconexão, o nó foi reconectado para permitir o progresso nas atividades, usando-se o comando "*sta1 iw dev sta1-wlan0 connect my-ssid*". Logo após isso, foi verificada a reconexão, do mesmo modo como se verificou a desconexão, tudo conforme a figura abaixo.

```
mininet-wifi> sta1 iw dev sta1-wlan0 connect my-ssid
mininet-wifi> sta1 iwconfig
sta1-wlan0  IEEE 802.11  ESSID:"my-ssid"
          Mode:Managed  Frequency:2.412 GHz  Access Point: 02:00:00:00:02:00
          Bit Rate:1 Mb/s   Tx-Power=14 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:on
          Link Quality=70/70  Signal level=-36 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:2   Missed beacon:0

lo        no wireless extensions.

mininet-wifi> sta1 ping sta2
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=3.11 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.180 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=1.08 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.192 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.171 ms
^C
--- 10.0.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4039ms
rtt min/avg/max/mdev = 0.171/0.945/3.107/1.135 ms
```

Figura 9: Reconexão e verificação do nó sta1

Executando um comando *ping* com trinta repetições, tem-se o que se mostra na *Figura 10*, na qual é possível perceber que não há perda de pacotes na situação em que a rede se encontra e que o atraso médio é de *0,130 ms*.

```
--- 10.0.0.2 ping statistics ---
30 packets transmitted, 30 received, 0% packet loss, time 29696ms
rtt min/avg/max/mdev = 0.076/0.130/0.239/0.037 ms
```

Figura 10: Resposta para comando ping com 30 repetições

Em seguida, para avaliar a banda disponível entre as duas estações, *sta1* a *sta2*, foi executado o comando "*iperf sta1 sta2*". Tal comando gerou a resposta da *Figura 11*, abaixo, o qual mostra um throughput médio de 54,2 Mbps.

```
mininet-wifi> iperf sta1 sta2
*** Iperf: testing TCP bandwidth between sta1 and sta2
*** Results: ['50.1 Mbits/sec', '58.3 Mbits/sec']
```

Figura 11: Resposta para comando iperf

Exercício 2)

Neste exercício, de começo, foi necessário sair da *CLI* do mininet-wifi para iniciar a mesma topologia mas com argumentos não anteriormente utilizados. Isso se deu com o comando "*sudo mn --wifi --position --link=wmediumd*", o qual impõe, por meio de "*position*", posições iniciais para os nós da topologia e habilita, por meio de "*--link=wmediumd*" o simulador de enlace wireless *wmediumd.* Feito isso, foi utilizado o comando "*distance sta1 sta2*" para verificar a distância entre as duas estações *sta1* e *sta2*, cujo valor foi de 2 metros, tudo conforme a figura abaixo.

```
wifi@wifi-virtualbox:~/EA080-2S2021/lab3$ sudo mn --wifi
d
[sudo] password for wifi:
*** Adding stations:
sta1 sta2
*** Adding access points:
ap1
*** Configuring wifi nodes...
*** Connecting to wmediumd server /var/run/wmediumd.sock
*** Creating network
*** Adding controller
*** Adding hosts:

*** Adding switches:

*** Adding links:
(sta1, ap1) (sta2, ap1)
*** Starting controller(s)
c0
*** Starting L2 nodes
ap1 ...
*** Starting CLI:
mininet-wifi> distance sta1 sta2
The distance between sta1 and sta2 is 2.0 meters
```

Figura 12: Geração da topologia com novos parâmetros e verificação da distância entre estações

Para conhecimento de informações a respeito da estação *sta1*, foi comandado "*py sta1.params*", cuja saída permite a verificação de alguns parâmetros da estação (Figura 13). Além disso, como um adicional desse comando, pôde-se usar filtros de parâmetros como "*['ip']*" e "*['freq']*", conforme as *Figuras 14* e *15* a seguir.

```
mininet-wifi> py sta1.params
{'ip': '10.0.0.1/8', 'ip6': '2001:0:0:0:0:0:0:1/64', 'channel': 1, 'freq': 2.4, 'b
and': 20, 'mode': 'g', 'isStation': True, 'wlan': ['sta1-wlan0']}
```

Figura 13: Saída do comando params

```
mininet-wifi> py sta1.params['ip']
10.0.0.1/8
```

Figura 14: IP da estação sta1 dado pelo comando params

```
mininet-wifi> py sta1.params['freq']
2.4
```

Figura 15: Frequência portadora da comunicação

Investigando, agora, a lista de parâmetros apresentada pelo comando *params*, tem-se o seguinte (tudo com relação ao *sta1* neste caso):

- ip → Apresenta o endereço IPv4 atual da interface sem-fio do dispositivo;
- ip6 → Apresenta o endereço IPv6 da interface sem-fio do dispositivo;
- channel → Apresenta o canal dentro da frequência em que se estabelecem as comunicações da rede;
- freq → Apresenta a frequência portadora da rede;
- band → Apresenta a largura de banda em Mbps do link;
- mode → Apresenta o modo de configuração operacional do dispositivo na rede *wifi*;
- isStation → Indica se o nó é uma estação;
- wlan → Apresenta a interface wireless do dispositivo.

Dando continuidade ao procedimento deste exercício, foram executados os comandos "*py sta1.ports*" para identificar as portas da estação *sta1*, "*py sta1.wintfs*" para checar as interfaces sem-fio e, "*py sta1.position*" e "*py sta2.position*" para conhecer as posições de *sta1* e *sta2* respectivamente. Desse modo, sabe-se que *sta1* está no ponto (1, 0, 0) e *sta2* está em (3, 0, 0), tudo conforme a figura que segue.

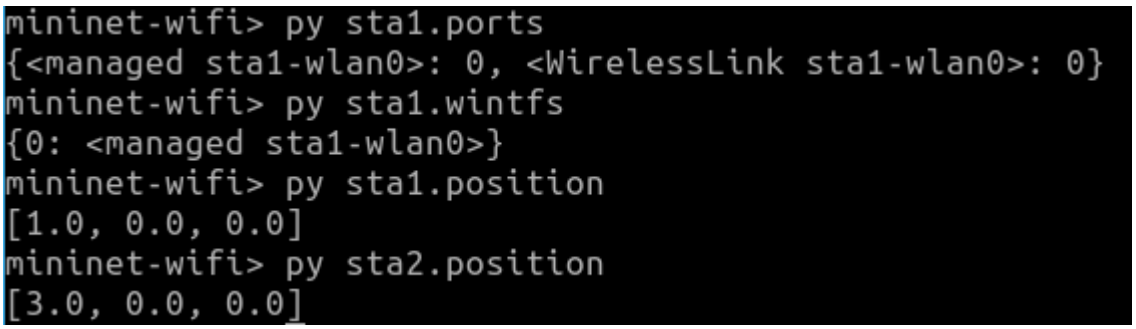

Figura 16: Saída de comandos de porta, de interfaces e de posições citados

Seguindo, foi alterada a posição do *sta1* e medida a distância entre os dois nós *sta1* e *sta2*, conforme a figura a seguir:

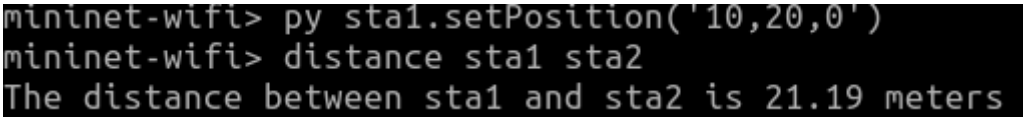

Figura 17: Saída do teste de distância entre sta1 e sta2 após mudança de posição

Por fim, apenas como modo de conhecimento, foi executado o comando "*ap1 ip link*", que gerou o que se apresenta na imagem a seguir:

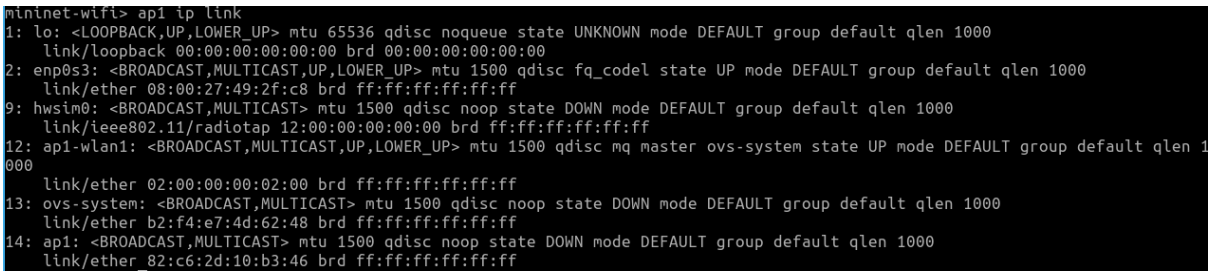

Figura 18: Saída do comando ap1 ip link

Exercício 3)

Seguindo o procedimento do laboratório, foi usado o *script python* "*position.py*", presente na pasta "*mininet-wifi/examples*", para plotar dois gráficos, um em três dimensões e outro em duas, ambos da topologia usada para as análises anteriores. Desse modo, obteve-se o que se apresenta nas duas figuras a seguir.

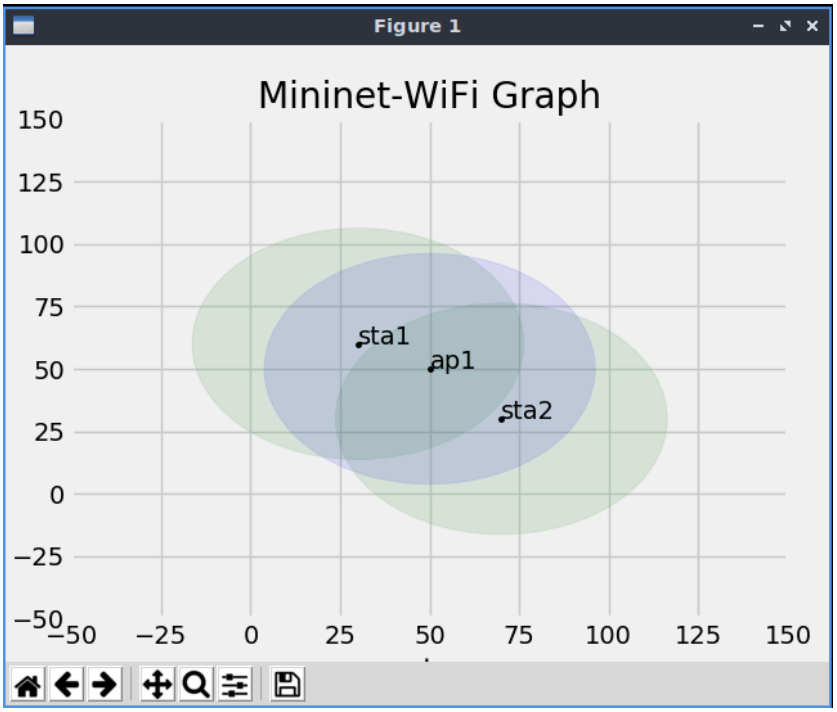

Figura 19: Gráfico da topologia em 2D

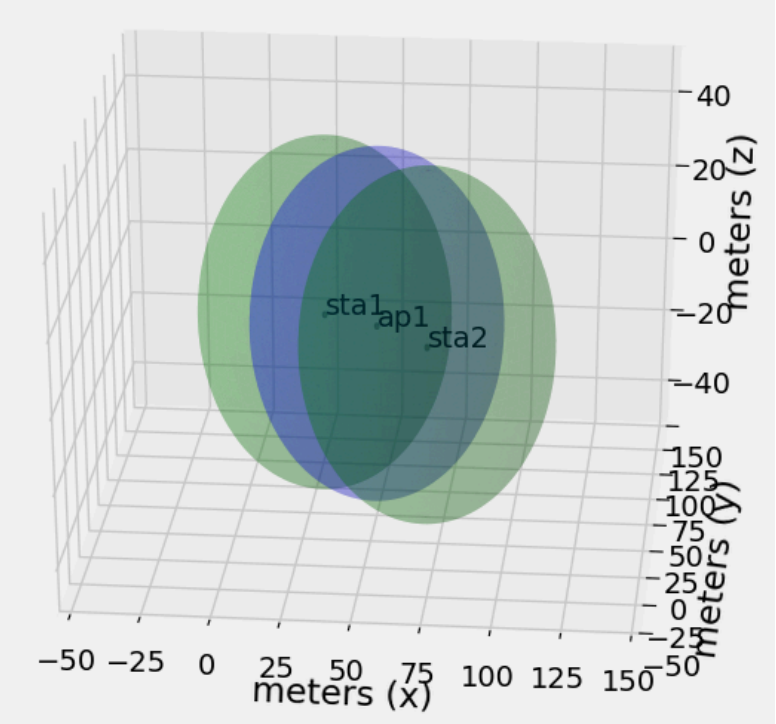

Figura 20: Gráfico da topologia em 3D

Exercício 4)

Prosseguindo, para possibilitar a análise de pacotes wireless por meio do wireshark, considerando a mesma topologia usada no exercício 2, foram usados os seguintes comandos na CLI do mininet:

1) *sta1 iw dev sta1-wlan0 interface add sta1-mon type monitor*
2) *sta1 ifconfig sta1-mon up*

Tais comandos, foram responsáveis por criar uma interface do tipo monitor na estação 1, de forma a permitir a captura dos pacotes *wireless* antes de serem encapsulados em pacotes *ethernet* falsos. Com isso, foi possível efetuar uma captura por *wireshark* de pacotes resultantes de um *ping* de sta2 para sta1, conforme se apresenta na figura abaixo.

```
98 9.272307… 10.0.0.2    10.0.0.1    ICMP  134 Echo (ping) request  id=0x0e8d, seq=4/1024…
99 9.272563… 10.0.0.1    10.0.0.2    ICMP  129 Echo (ping) reply    id=0x0e8d, seq=4/1024…
```

Figura 21: Pacotes wireless capturados em sta1

Além disso, foram capturados pacotes em uma conexão ethernet convencional feita na topologia dada pelo comando "sudo mn --mac", obtendo-se o que se apresenta a seguir.

```
3 0.0026… 10.0.0.2    10.0.0.1    ICMP  98 Echo (ping) request  id=0x0fd0, se…
4 0.0027… 10.0.0.1    10.0.0.2    ICMP  98 Echo (ping) reply    id=0x0fd0, se…
```

Figura 22: Pacotes ethernet capturados em h1

Feito isso, foi conduzida uma análise a respeito das diferenças no número de endereços MAC em cada caso (*ethernet* e *wireless*). Para tanto, cabe visualizar as duas figuras que se seguem, as quais apresentam tais MACs respectivamente do pacote *wireless* e do *ethernet*.

```
Receiver address: 02:00:00:00:00:00 (02:00:00:00:00:00)
Transmitter address: 02:00:00:00:02:00 (02:00:00:00:02:00)
Destination address: 02:00:00:00:00:00 (02:00:00:00:00:00)
Source address: 02:00:00:00:01:00 (02:00:00:00:01:00)
```

Figura 23: endereços MAC no pacote wireless "request" do ping

```
Destination: 00:00:00_00:00:01 (00:00:00:00:00:01)
Source: 00:00:00_00:00:02 (00:00:00:00:00:02)
```

Figura 24: endereços MAC no pacote ethernet "request" do ping

Observando-se tais figuras, é possível notar que o número de MACs que participam do pacote *wireless* excede aquele do pacote *ethernet* em dois. Isso ocorre porque, para além dos MACs do dispositivo que envia o pacote e daquele que é o destinatário, existem os MACs dos dispositivos de recepção e transmissão do pacote. Isso se faz evidente neste experimento, pois o MAC do transmissor coincide com o MAC do ponto de acesso *ap1*, conforme se espera, uma vez que é ele o responsável por receber o pacote de *sta2* e transmiti-lo a *sta1* na rede sem fio em que esta estação está presente. Ademais, no caso presente, nota-se que o MAC do receptor é o mesmo que o MAC do destinatário, também conforme esperado, uma vez que o MAC escrito nessa seção (*Receiver address*) é sempre o

MAC da próxima estação que deve receber o pacote e, como *sta1* é a última estação, tal MAC no pacote que chega a ela é o seu próprio MAC.

Por fim, para se detalhar melhor o que foi exposto acima com relação aos *Receiver* e *Transmitter address*, foram capturados pacotes em *ap1* após a criação de interface de monitoramento nesse ponto de acesso e, obteve-se o seguinte para os endereços MAC em um pacote:

```
Receiver address: 02:00:00:00:02:00 (02:00:00:00:02:00)
Transmitter address: 02:00:00:00:01:00 (02:00:00:00:01:00)
Destination address: 02:00:00:00:00:00 (02:00:00:00:00:00)
Source address: 02:00:00:00:01:00 (02:00:00:00:01:00)
```

Figura 25: endereços MAC no pacote wireless "request" do ping recebido por ap1

Na figura apresentada (25), nota-se que, conforme se esperava, o *Receiver address* foi alterado pelo endereço do ponto de acesso *ap1* e o *Transmitter address* foi trocado pelo MAC da estação *sta2*. Isso se deu, conforme o que se explicou anteriormente, por conta do fato de esses endereços serem alterados a cada estação, de modo que o *Receiver address* apresenta o endereço da próxima estação que deve receber o pacote e o *Transmitter* indica o MAC da estação que transmite o pacote para a rede sem fio. Desse modo, como o *ap1* recebe, seu MAC é o apresentado em *Receiver* e como o pacote em questão veio de *sta2*, é o MAC dessa estação que aparece como *Transmitter address.*

Exercício 5)

Prosseguindo na execução do experimento, foi executado o *script python* "*propagation_model.py*". Tal *script* conforme se nota na *Figura 26* a seguir, cria uma topologia com três estações finais (*sta1, sta2* e *sta3*) e um *access point* (*ap1*). Desse modo, esperar-se-ia que as três estações estivessem associadas ao ponto de acesso *ap1*. No entanto, conforme se nota ao avaliar as conexões de cada estação por meio de *iwconfig* (*Figuras 27, 28 e 29*), a estação *sta3* não está associada a *ap1*.

```
mininet-wifi> nodes
available nodes are:
ap1 sta1 sta2 sta3
```

Figura 26: Nós na topologia do script propagation_model.py

```
root@wifi-virtualbox:/home/wifi/EA080-2S2021/lab3# iwconfig
sta1-wlan0  IEEE 802.11  ESSID:"my-ssid"
          Mode:Managed  Frequency:5.18 GHz  Access Point: 02:00:00:00:03:00
```

Figura 27: Resultado de iwconfig na estação 1

```
root@wifi-virtualbox:/home/wifi/EA080-2S2021/lab3# iwconfig
lo        no wireless extensions.

sta2-wlan0  IEEE 802.11  ESSID:"my-ssid"
          Mode:Managed  Frequency:5.18 GHz  Access Point: 02:00:00:00:03:00
```

Figura 28: Resultado de iwconfig na estação 2

```
root@wifi-virtualbox:/home/wifi/EA080-2S2021/lab3# iwconfig
lo        no wireless extensions.

sta3-wlan0  IEEE 802.11  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
```

Figura 29: Resultado de iwconfig na estação 3

Então, pode-se concluir que *sta3* está incomunicável com as demais estações, uma vez que não está ligada a nenhum ponto de acesso. Sendo assim, buscando-se um motivo para tal fato, pôde-se usar o gráfico *plotado* pelo próprio *script* em sua execução (*Figura 30*), no qual se nota que *sta3* está muito próxima do limite de alcance do sinal de *ap1*. Isso poderia, em primeiro momento, justificar a incomunicabilidade de *sta3* e, conforme análise apresentada a diante, isso de fato explica o problema.

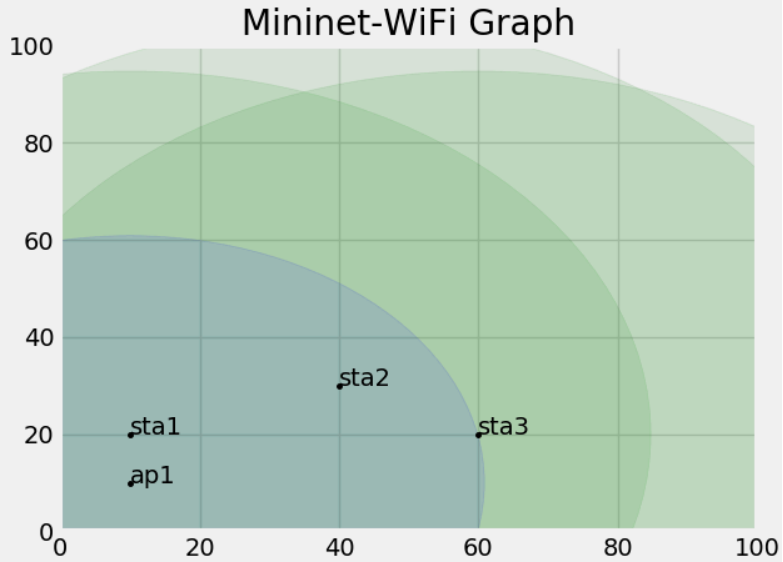

Figura 30: Gráfico da topologia do script propagation_model.py

Com isso, pode-se notar que o problema de *sta3* não estar associado ao *ap1* ocorre por causa do fato de que o sinal de *ap1* não alcança, por pouco, a estação em questão (sinal que chega a estação é cortado pelo threshold do mininet, nível menor que *-92 dBm*). Desse modo, a estação é incapaz de se comunicar com *ap1* e, por consequência, com qualquer estação da topologia apresentada.

Seguindo, ao se analisar o *script*, pode-se notar que o modelo de propagação configurado para a emulação é o *Log Distance*, conforme se nota na linha a seguir:

"*net.setPropagationModel(model="logDistance", exp=4)*"

Assim, pode-se notar que o nível do sinal recebido é bastante alto, uma vez que o modelo pode ser descrito pela seguinte equação:

$$[P_L(d)]dB = [P_L(d_0)]dB + 10\ n\ log_{10}\left(\frac{d}{d_0}\right) + \chi$$
$$for\ d_f \leq d_0 \leq d \qquad (1)$$

Figura 31: Equação do modelo Log Distance de propagação

De acordo com tal equação, pode-se perceber que o *Path Loss*, ou seja, a redução do sinal entre transmissor e receptor, é proporcional ao aumento da distância, de modo que, quanto mais próximo a estação do transmissor, maior a intensidade do sinal que chega até essa estação, pois menor é a perda até ela. Dessa maneira, como *sta1* está bastante próxima de *ap1*, pode-se concluir que o sinal

de *ap1* que chega na estação 1 tem um nível muito próximo do sinal que é emitido pelo *access point*. Finalmente, como evidência de que o nível de sinal é bastante bom, tem-se o que se apresenta na *Figuras 32 e 33* abaixo, nas quais são apresentados qualidade do *link* e nível de sinal de *sta1* e *sta2* respectivamente.

Figura 32: medidas de qualidade de enlace e nível de sinal de *ap1* em *sta1*

Figura 33: medidas de qualidade de enlace e nível de sinal de *ap1* em *sta2*

Nestas duas últimas figuras, pode-se perceber que o valor de nível de sinal para *sta2* é menor que o de *sta1*, indicando que *sta1* recebe um sinal mais potente (mais distante de 0 mW), enquanto que *sta2* recebe um sinal de potência menor. Além disso, é explícito nas imagens que a qualidade do enlace com *sta1* é muito maior que a com *sta2*, pois a intensidade que chega em *sta1* é consideravelmente maior por essa estação estar mais próxima de *ap1*.

Na sequência disso, para analisar um segundo caso de modelo, foi substituído o modelo pelo modelo "*friss"* disponível no emulador. Para tanto, a linha em que se define o modelo foi alterada para "*net.setPropagationModel(model="friis")*" de modo que a perda do sistema (parâmetro *sL* presente na definição do método no site do mininet-wifi, Figura 34) foi mantida a mesma para se analisar a diferença dos sinais entre modelos.

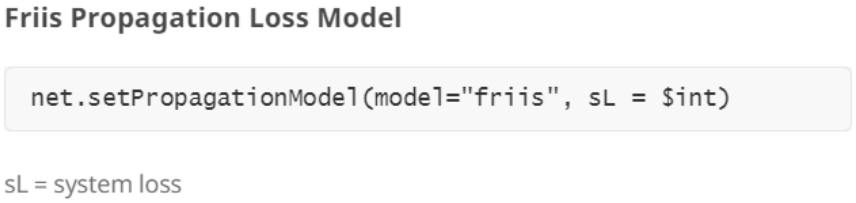

Figura 34: Apresentação do método de definição de modelo de propagação para o caso friis

Feita a alteração no *script*, obteve-se o seguinte gráfico da topologia:

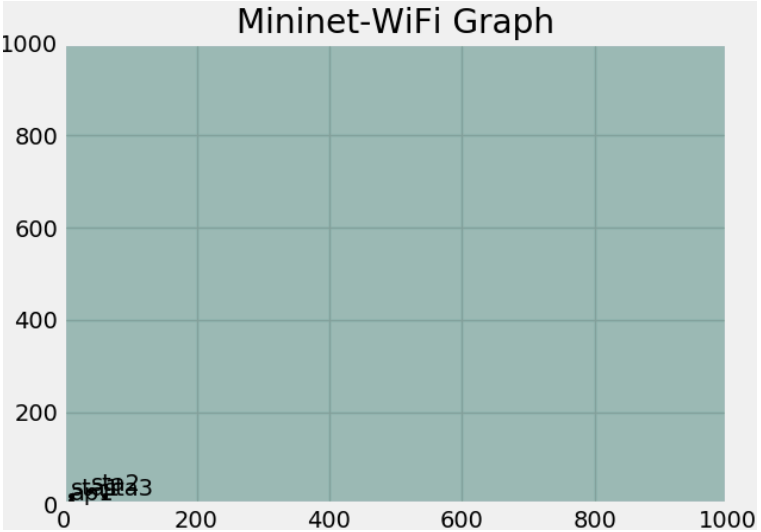

Figura 35: Gráfico da topologia com modelo Friss de propagação

Conforme se nota, obteve-se um sinal de alcance extremamente amplo, o que indica que talvez o *system loss* seja o único fator de perda do modelo ou o preponderante sobre os demais fatores. Logo, após uma análise de valores, aumentando esse parâmetro para 2600, obteve-se algo semelhante ao que se tinha com *Log Distance* para o alcance do sinal de *ap1*, conforme o gráfico a seguir:

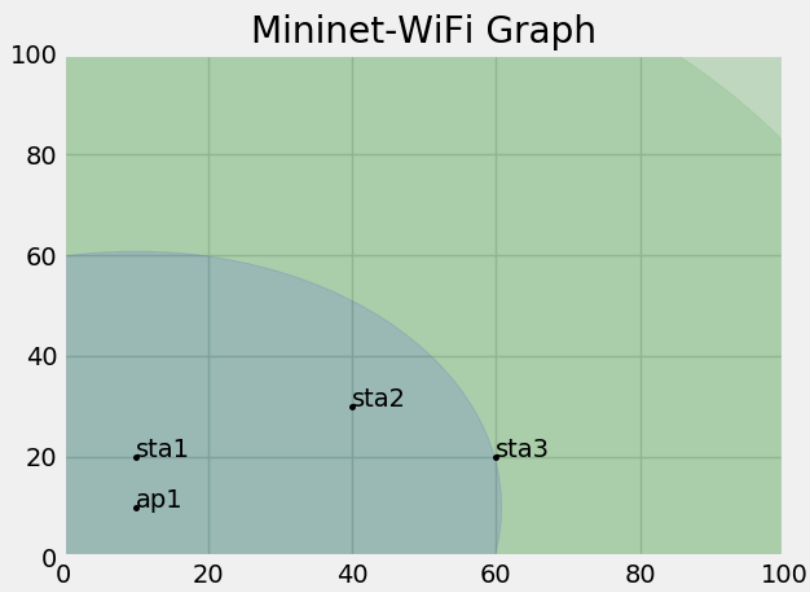

Figura 36: Gráfico da topologia com modelo Friss de propagação com sL = 2600

Por fim, nota-se dos dois últimos gráficos, que para o modelo de propagação *Friss*, ao contrário do modelo *Log Distance*, o valor de *system loss* é extremamente determinante, de modo que não atribuí-lo na chamada do método gera alcances absurdamente grandes dos sinais das estações. Além disso, percebe-se que, mesmo com um valor de *system loss* que busca igualar o comportamento do sinal de *ap1* entre os modelos, os sinais de *sta1*, *sta2* e *sta3* se mantêm absurdamente maiores que aqueles no modelo *Log Distance*, o que indica uma diferença do efeitos de cada modelo entre estações finais e *access points*. Isso ocorre, pela diferença matemática entre os dois modelos, que define comportamentos emulados, por consequência, também diferentes, o que indica que, para se tentar retratar a realidade, deve-se buscar o modelo mais adequado para a situação que se deseja representar, uma vez que cada um deles pode demonstrar comportamentos mais semelhantes a diferentes situações reais que os demais representariam.

Exercício 6)

Usando novamente o modelo *Log Distance* do mesmo modo como feito anteriormente, mas com *sta1* e *sta2,* de início, respectivamente nas posições *(11, 10, 0)* e *(12, 10, 0)*, utilizando-se iperf e alterando a posição de *sta2* subsequentemente, traçou-se o gráfico de largura de banda entre estações (*sta1* e *sta2*) em função da distância entre elas, conforme a *Figura 37* que se segue.

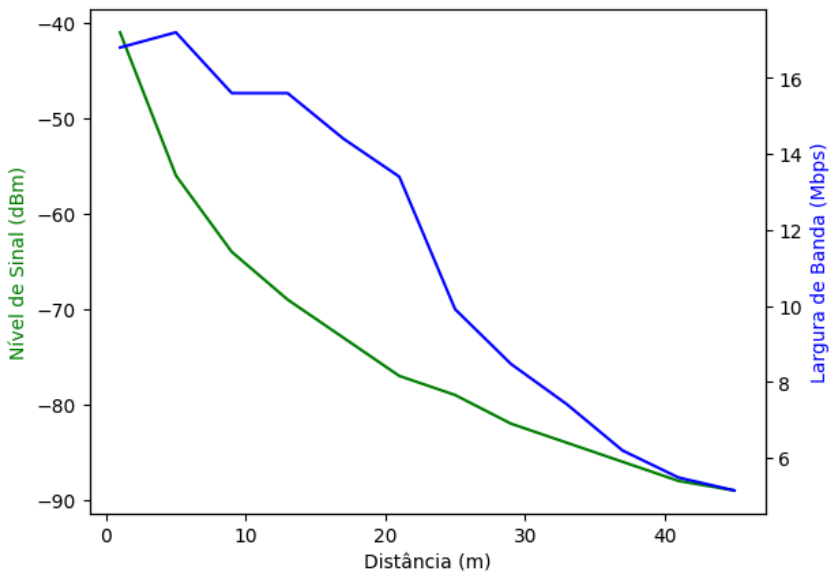

Figura 37: Gráfico de nível de sinal e largura de banda em função da distância entre duas estações

Para análise do gráfico acima, cabe ressaltar que as amostras foram tomadas de 4 em 4 metros de distância, começando em 1 e terminando em 45 metros, pois a partir de 49 o nível de sinal fica menor que *-92dBm* e, por isso, o *mininet* corta a comunicação. Finalmente, vale destacar que o gráfico foi feito usando *matplotlib* com o código abaixo, no qual se pode observar todos os valores obtidos. Além disso, em algumas das distâncias, foi necessário coletar dados de largura de banda mais de uma vez e calcular a média, pois os valores para tais distâncias oscilavam muito.

```python
import matplotlib.pyplot as plt
import numpy as np

xpoints = np.array([1, 5, 9, 13, 17, 21, 25, 29, 33, 37, 41, 45])
ypoints = np.array([16.8, 17.2, 15.6, 15.6, 14.4, 13.4, 9.91, 8.47, 7.41,
6.19, 5.48, 5.14])
y2points = np.array([-41, -56, -64, -69, -73, -77, -79, -82, -84, -86, -88,
-89])

fig, ax1 = plt.subplots()

ax2 = ax1.twinx()
ax1.plot(xpoints, y2points, 'g-')
ax2.plot(xpoints, ypoints, 'b-')

ax1.set_xlabel('Distância (m)')
ax1.set_ylabel('Nível de Sinal (dBm)', color='g')
ax2.set_ylabel('Largura de Banda (Mbps)', color='b')

plt.show()
```

Exercício 7)

Para este exercício, foi alterado o *script* "*handover.py*" como forma de facilitar a análise da conectividade entre *sta1* e *sta2*. Para tanto, foram multiplicados todos os valores de tempo na configuração da mobilidade no *script*, de modo que o movimento ficou mais lento e, por consequência, mais fácil de ser analisado. Feito isso, o programa foi executado e, o mais rápido possível, foi iniciado um comando *ping* de *sta1* para *sta2*. Deste comando *ping*, que perdurou até um pouco depois do fim da mobilidade, obteve-se o resultado a seguir.

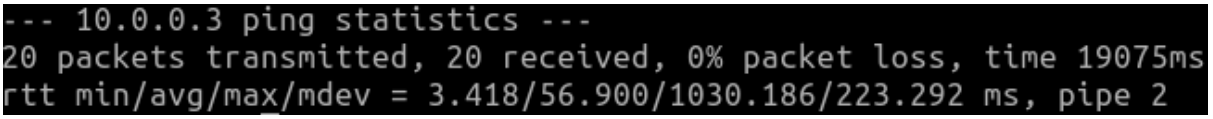

Figura 38: Resultado de ping de sta1 para sta2 durante mobilidade das estações

Dessa figura, é possível concluir que a conectividade não foi perdida em momento nenhum entre as duas estações. No entanto, existe um momento, na saída de *sta1* dos limites do sinal de *ap1*, em que ocorre um erro no comando *ping* (problema de comunicação), pois a estação *sta1* passou a depender de *ap2* para ser localizado e receber os pacotes. Isso porque, sem sinal de *ap1*, a estação *sta1* passou a receber sinal apenas de *ap2* e, por consequência, é de *ap2* que passa a receber pacotes enviados a ela. Logo, a partir de tal momento, *ap1* passa a endereçar os pacotes, cujos destinos são *sta1*, para *ap2*, enviando-lhes por ethernet para este *access point*, que por sua vez os envia para *sta1*. Vale ressaltar ainda, que essa alteração ocorre quando *sta1* sem acesso a *ap1* passa a usar o *access point 2* para se comunicar com o *access point 1* ao qual está associado e conseguir, consequentemente, voltar a se comunicar com outras estações.

Feita a análise anterior, pode-se afirmar que esse processo, com nome igual ao do *script*, chama-se *handover* e é caracterizado pela passagem da estação a um novo *access point* de sinal melhor que o *access point* ao qual a estação estava associada anteriormente (No caso presente, a troca ocorre pela falta de conectabilidade da estação com o *access point* anterior). Sendo assim, ele funciona com um escaneamento periódico da estação por *access points* acessíveis, de modo que não há troca de associação, mas o *access point* ao qual a estação está associada passa a redirecionar pacotes para o *ap* com o qual a estação está em contato no momento. Além disso, o processo em questão pode ocorrer quando a estação sente um sinal melhor de um ou outro *access point* ou quando ocorre perda do sinal com o *access point* associado à estação e ela encontra um *access point* diferente que esteja acessível.

Conclusão)

Por fim, pode-se concluir que o objetivo de aprofundamento em conceitos relacionados às redes sem fio foi muito bem sucedido. De mesmo modo, o almejo de se aprender a usar o mininet-wifi também foi satisfeito, de modo que não só se pode aprender a usá-lo, mas também se pode entender um pouco do funcionamento dos modelos matemáticos usados por ele para emulação de redes.