Relatório 03

Vinícius de Oliveira Peixoto Rodrigues (245294)

Março de 2023

1 Introdução

A tecnologia Wi-Fi é uma família de protocolos de rede sem fio, baseadas na série de padrões IEEE 802.11, que definem as especificações para implementação de tecnologias de rede wireless. As redes wireless são amplamente utilizadas para construir redes de pequeno porte em aplicações domésticas ou de negócios.

A ferramenta `mininet` fornece um conjunto de ferramentas (disponibilizadas como uma interface de linha de comando ou como uma interface em Python) que permite a construção de arquiteturas de rede simuladas de forma rápida e fácil. Um fork dessa ferramenta, chamado `mininet-wifi`, extende as funcionalidades da ferramenta original, implementando suporte para simulação de redes Wi-Fi. Este experimento busca explorar as funcionalidades da ferramenta `mininet-wifi`.

2 Objetivos

Este experimento básico tem como objetivo os seguintes pontos:

- Explorar as funcionalidades da ferramenta `mininet-wifi`
- Compreender os princípios básicos de funcionamento de arquiteturas de rede wireless
- Fazer uso do simulador para investigar a relação entre parâmetros físicos e a performance de redes sem fio

3 Metodologia

O experimento dividiu-se em sete etapas:

3.1 Primeiros passos

Foi realizada a criação de uma topologia wireless básica, com um AP e duas estações (`sta1` e `sta2`). Em seguida, foram realizados testes básicos de conexão

e desconexão entre as estações e o access point; finalmente, foram realizadas medições de largura de banda entre as duas estações.

3.2 Exploração dos parâmetros do simulador

Foi explorado o ajuste de parâmetros das estações no simulador, em particular o de posição das estações. Além disso, foi realizada uma análise dos diversos parâmetros de configuração disponíveis no simulador.

3.3 Visualização da topologia wireless

Nesta etapa, foram feitos testes com diferentes esquemas de plotagem de topologias wireless disponíveis no mininet-wifi.

3.4 Análise de frames MAC 802.11

Já aqui, foi investigado por meio da ferramenta Wireshark o formato de frames padrão 802.11, e como ele difere dos frames 802.3.

3.5 Estudo de modelos de propagação

Nesta etapa, foi investigada a influência de diferentes modelos teóricos de propagação de sinal na comunicação entre estações e APs Wi-Fi.

3.6 Análise da influência do RSSI na bandwidth

Em seguida, foi analisado, utilizando-se um dos modelos investigados na etapa interior, a influência do RSSI no throughput de comunicação entre duas estações na rede Wi-Fi, à medida que uma delas se distancia do AP.

3.7 Investigação do *Roaming* Wi-Fi

Finalmente, foi estudado um caso de simulação no `mininet-wifi` que demonstra o comportamento do *roaming* quando uma estação transita entre as regiões de cobertura de dois APs diferentes (mas com mesmo SSID).

4 Resultados e Discussão

Parte 1

> Qual é o atraso observado entre `sta1` e `sta2`? Houve perda de pacotes no canal? Justifique suas respostas de forma objetiva.

O atraso médio observado na saída do comando `ping` é de 9.8 segundos. Além disso, não foram observadas perdas de pacote.

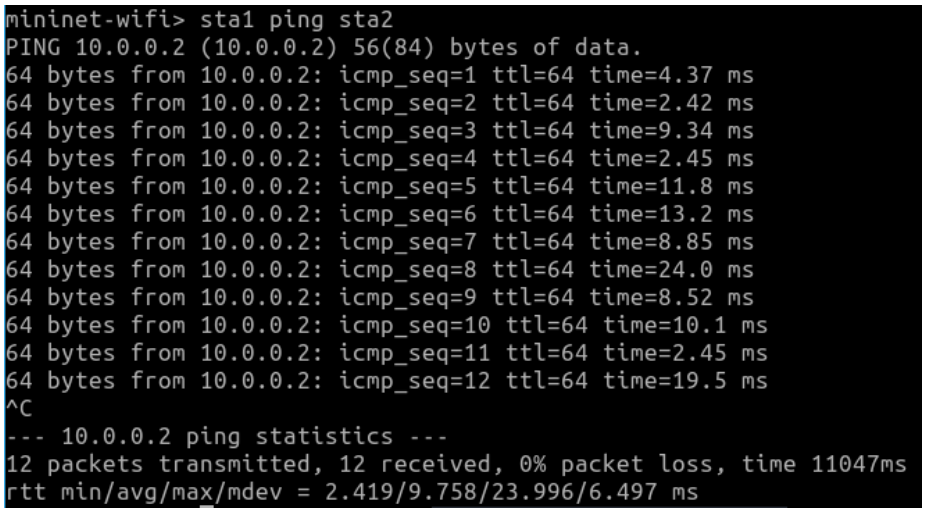

Figura 1: `ping` entre `sta1` e `sta2`.

Use a ferramenta iperf para avaliar a banda disponível (Mbps) entre sta1 e sta2.

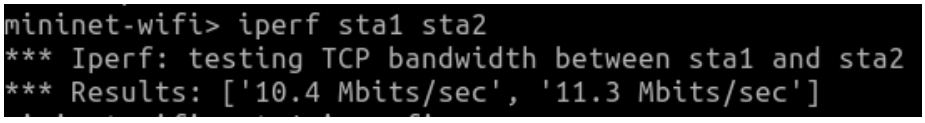

Figura 2: Teste de largura de banda entre `sta1` e `sta2`.

A largura de banda média entre as duas estações é de aproximadamente 11 Mbits/sec. É interessante observar que a largura de banda na prática é significativamente menor que o bitrate máximo negociado para os links `wlan` de cada estação.

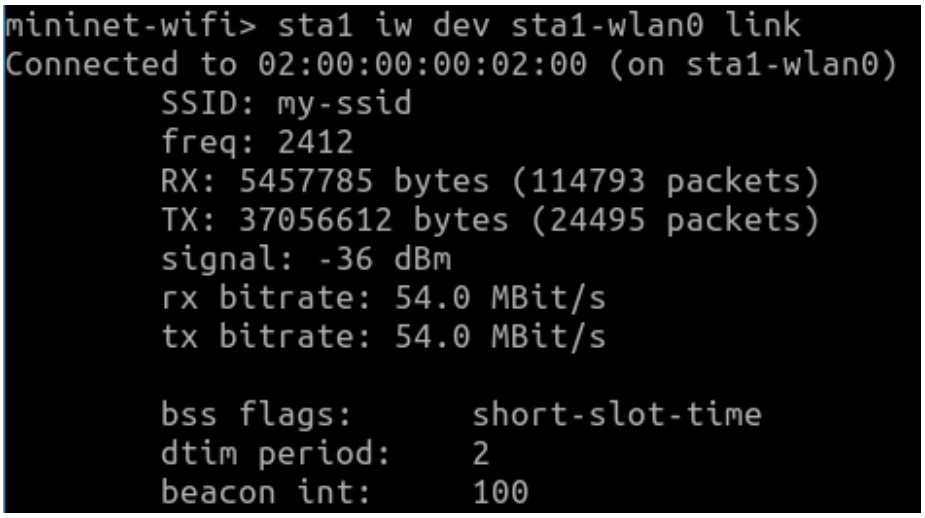

Figura 3: Status do link `wlan` da estação `sta1`, demonstrando velocidade negociada maior do que a medida pelo `iperf`.

4.1 Análise de quadros

Análise de quadros 802.11 gerados pelo simulador no Wireshark

Parte 2

> Identifique a posição dos nós `sta1` e `sta2`.

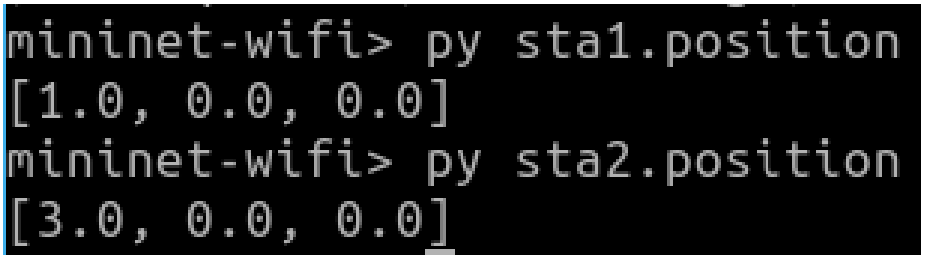

Figura 4: Posição das estações `sta1` e `sta2`.

> Investigue a lista de parâmetros de `sta1` (`py sta1.params`). Explique resumidamente qual informação do nó `sta1` cada item da lista representa.

Os parâmetros encontrados são:

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

Figura 5: Parâmetros de `sta1`.

- `ip/ip6`: o endereço IPv4/Ipv6 atribuído à interface `wlan` da estação
- `channel`: o canal (isto é, faixa de frequências de operação) de acordo com a especificação 802.11g
- `band`: a banda (2.4 GHz) de comunicação wireless
- `freq`: a frequência da comunicação wireless
- `mode`: qual versão específica do protocolo 802.11 (neste caso, 802.11g) na qual o canal está operando
- `isStation`: variável para identificação de estações Wi-Fi
- `wlan`: indica qual interface de rede (`sta1-wlan0`) está sendo usada para a conexão Wi-Fi

4.2 Parte 3

> Plote um gráfico em duas e outro em três dimensões.

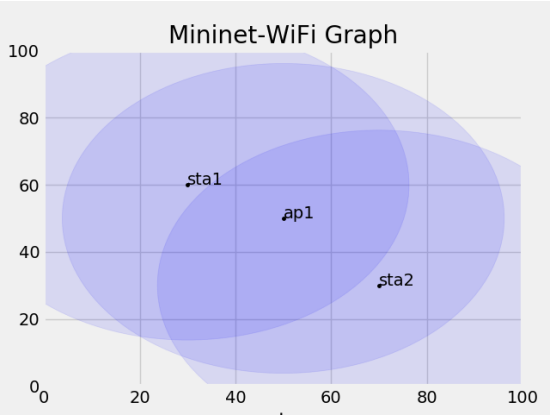

Figura 6: Gráfico 2D gerado.

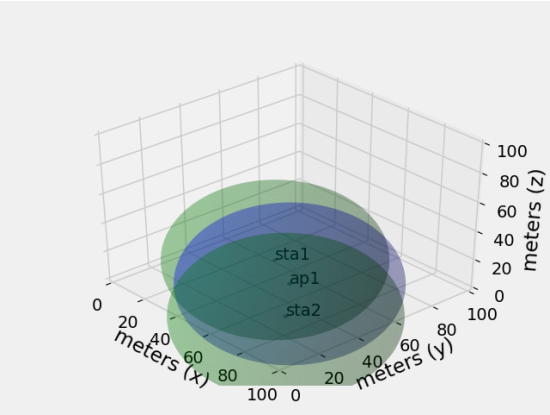

Figura 7: Gráfico 3D gerado.

4.3 Parte 4

> Explique as principais diferenças e funções em relação ao número de endereços MAC contidos nos quadros do tipo 802.11 e 802.3.

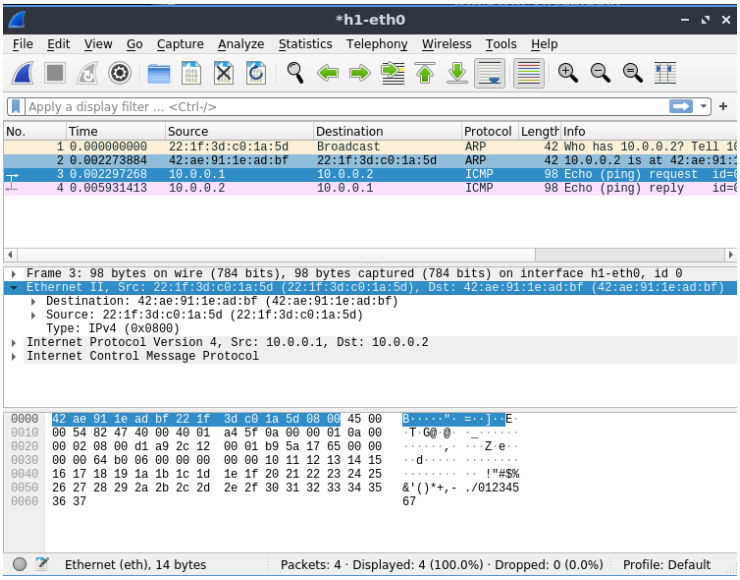

Figura 8: Frame MAC numa conexão *wired*.

Um frame 802.11 contém, além dos endereços MAC `source` e `destination` encontrados no 802.3, um terceiro endereço MAC pertencente ao *access point*.

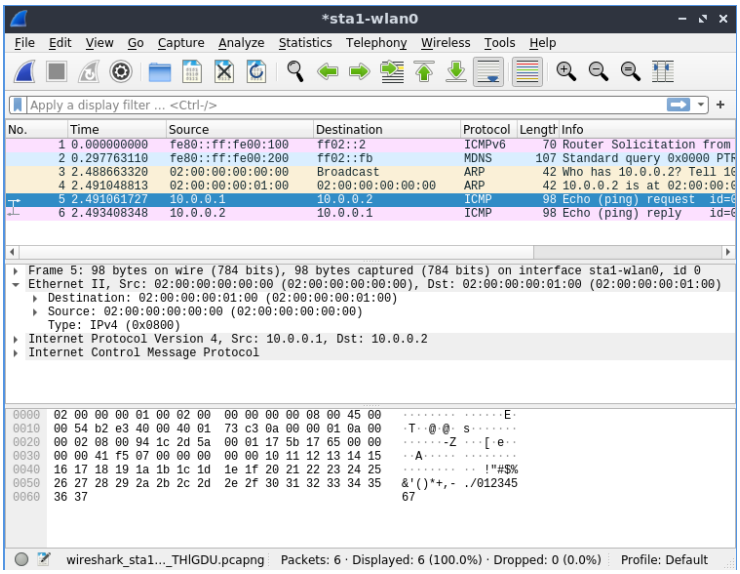

Figura 9: Frame MAC numa conexão *wireless*.

As Figuras 8 e 9 mostram, contudo, que na captura de pacotes do Wireshark os frames MAC com fio e sem fio são idênticos; isso acontece porque a interface *wireless* do `mininet-wifi` está rodando em modo *managed*, o que significa que o driver da interface no kernel "reorganiza"os frames MAC dos pacotes que entram e saem, retirando os campos específicos do 802.11 e substituindo pelos equivalentes no 802.3; não foi possível rodar a interface simultaneamente em modo monitor e managed para ilustrar as diferenças de frames no 802.11 e 802.3.

4.4 Parte 5

> Um dos nós parece estar incomunicável. Qual é ele e por qual motivo o nó está incomunicável? Ele encontra-se associado ao AP1?

Na Figura 10, é possível ver que a estação `sta3` se encontra inalcançável, visto que não está associada ao AP `ap1`. Isso acontece, como ilustrado no gráfico, porque `sta3` está fora do alcance de `ap1` (razão pela qual o AP aparece no scan de `sta1`, mas não de `sta3`).

> O script já vem pré-configurado com um modelo de propagação. Identifique-o.

A Figura 11 mostra o snippet de código relevante, onde é configurado o

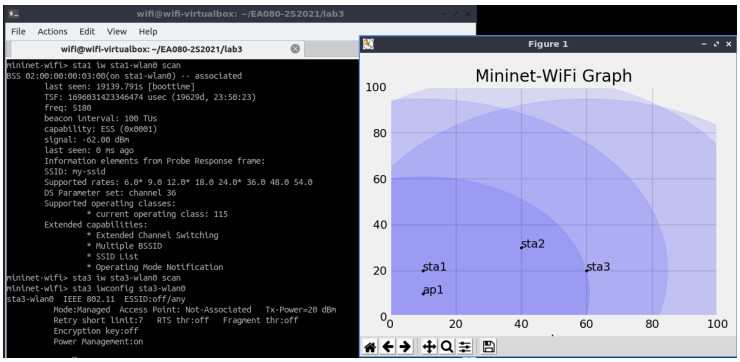

Figura 10: Diagnóstico da estação `sta3`, que se encontra incomunicável.

```
info("*** Configuring Propagation Model\n")
net.setPropagationModel(model="logDistance", exp=4)
```

Figura 11: Modelo de propagação selecionado no código.

modelo Log-distance path loss, onde a potência do sinal cai exponencialmente:

$$\frac{P_{\text{Rx}}}{P_{\text{Tx}}} \sim \frac{1}{d^{\gamma}}$$

> Identifique e descreva o nível de sinal percebido por sta1. Depois substitua o modelo de propagação do script por um outro suportado pelo Mininet-WiFi e então configure esse modelo no script.

```
mininet-wifi> sta1 iwconfig sta1-wlan0
sta1-wlan0  IEEE 802.11  ESSID:"my-ssid"
          Mode:Managed  Frequency:5.18 GHz  Access Point: 02:00:00:00:03:00
          Bit Rate:54 Mb/s   Tx-Power=14 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:on
          Link Quality=48/70  Signal level=-62 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:22   Missed beacon:0
```

Figura 12: Nível de sinal para o modelo log-distance.

O outro modelo de propagação escolhido foi o Friis path loss, no qual a potência do sinal cai com o quadrado da distância:

$$\frac{P_{\text{Rx}}}{P_{\text{Tx}}} \sim \frac{1}{d^{2}}$$

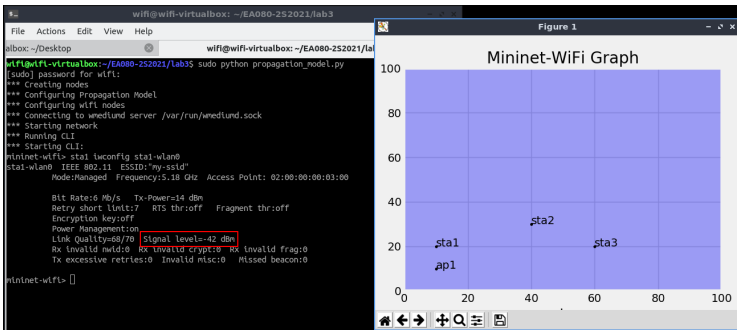

Figura 13: Nível de sinal para o modelo Friis.

> Há diferenças entre o nível de sinal antes e após a mudança do modelo de propagação? Justifique o resultado obtido.

Sim: percebe-se que o nível de sinal aumentou com o uso do modelo de propagação Friis (-62dB $\rightarrow -42$dB).

4.5 Parte 6

> Com o script `propagation_model.py` e utilizando Iperf, monte um gráfico ou tabela que relacione a largura de banda entre dois nós (sta) com a distância x entre eles, assim como exemplificado na figura. Monte também um gráfico ou tabela que relacione o nível de sinal à distância.

Para essa etapa, foi utilizado o modelo Fiis de propagação. O script `propagation_model.py` (enviado em anexo juntamente ao relatório) foi modificado para aumentar gradualmente a distância entre as estações e coletar dados de RSS e throughput.

Na Figura 14, apesar das variações aleatórias introduzidas pela simulação do `mininet-wifi`, é possível ver que a diminuição do RSS devido ao distanciamento entre `sta1` e `sta2` faz com que o throughput entre da comunicação entre as estações seja deteriorado.

4.6 Parte 7

> A comunicação entre sta1 e sta2 se manteve ininterrupta? Por quê? O que aconteceu com sta1?

Não; a Figura 15 mostra o momento em que houve uma desconexão quando `sta1` saiu da área de alcance do `ap1` e entrou na área de cobertura do `ap2`, desconectando-se de um para conectar-se ao outro.

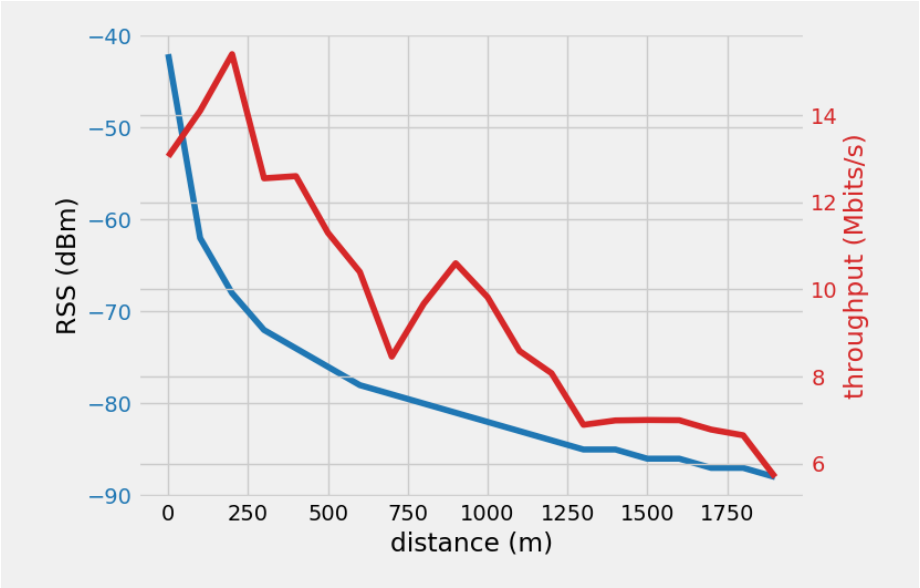

Figura 14: Influência do RSS no throughput entre as estações.

```
64 bytes from 10.0.0.3: icmp_seq=5 ttl=64 time=4.52 ms
Exception in thread wifiParameters:
Traceback (most recent call last):
  File "/usr/lib/python3.8/threading.py", line 932, in _bootstrap_inner
    self.run()
  File "/usr/lib/python3.8/threading.py", line 870, in run
    self._target(*self._args, **self._kwargs)
  File "/usr/local/lib/python3.8/dist-packages/mininet_wifi-2.6-py3.8.egg/mn_wifi/mobility.py", line 171, in parameters
    self.config_links(mob_nodes)
  File "/usr/local/lib/python3.8/dist-packages/mininet_wifi-2.6-py3.8.egg/mn_wifi/mobility.py", line 196, in config_links
    ack = self.check_in_range(intf, ap_intf)
  File "/usr/local/lib/python3.8/dist-packages/mininet_wifi-2.6-py3.8.egg/mn_wifi/mobility.py", line 139, in check_in_range
    self.ap_out_of_range(intf, ap_intf)
  File "/usr/local/lib/python3.8/dist-packages/mininet_wifi-2.6-py3.8.egg/mn_wifi/mobility.py", line 105, in ap_out_of_range
    intf.disconnect(ap_intf)
  File "/usr/local/lib/python3.8/dist-packages/mininet_wifi-2.6-py3.8.egg/mn_wifi/link.py", line 549, in disconnect
    self.iwdev_cmd('{} disconnect'.format(self.name))
  File "/usr/local/lib/python3.8/dist-packages/mininet_wifi-2.6-py3.8.egg/mn_wifi/link.py", line 114, in iwdev_cmd
    return self.cmd('iw dev', *args)
  File "/usr/local/lib/python3.8/dist-packages/mininet/link.py", line 70, in cmd
    return self.node.cmd( *args, **kwargs )
  File "/usr/local/lib/python3.8/dist-packages/mininet/node.py", line 386, in cmd
    self.sendCmd( *args, **kwargs )
  File "/usr/local/lib/python3.8/dist-packages/mininet/node.py", line 303, in sendCmd
    assert self.shell and not self.waiting
AssertionError
64 bytes from 10.0.0.3: icmp_seq=6 ttl=64 time=11.6 ms
64 bytes from 10.0.0.3: icmp_seq=7 ttl=64 time=5.12 ms
```

Figura 15: Desconexão durante o processo de movimentação de `sta1` entre a cobertura dos APs.

> Qual o nome do processo que a mobilidade ocasionou em sta1 ? Comente como ele funciona.

O nome do processo é *roaming*; essencialmente, as estações costumam definir

um *roaming threshold* para o RSSI com o AP atual; quando o RSSI se torna ruim o suficiente de modo a passar desse limite, a estação começa a fazer *scans* de modo a procurar outros APs com o mesmo SSID do atual. A estação escolhe então o AP alternativo com maior RSSI, e se conecta a ele.