Trabalho Prático 01
Redes de Computadores

Gabriel Miranda - 3857
Mariana Souza - 3898
Mateus Aparecido - 3858

20 de Setembro de 2021

Sumário

1. Preparo Inicial ……..…….....3

1.1 Instalação do Wireshark…….....3

2. Parte 1: Tecnologia Ethernet.. …….....3

2.1 Formato dos quadros Ethernet…….3

2.2 Envio de quadros Ethernet…….....4

2.3 Escolhendo Interface Ethernet…….5

2.4 ifconfig - endereço gateway……..……...……....6

2.5 Ping para o ip do gateway…….……...……6

2.6 Tráfego Ethernet……..…….……...7

2.7 Tráfego gerado pelo ICMP…….……..……...8

2.8 Análise Pacote 1……...……...9

2.9 Quadro Frame…….……..……..10

2.10 Quadro Ethernet II…….…….……..11

2.11 Destination, source e type……..……...12

2.12 Análise do Pacote 2…….……...……..14

2.13 Frame, Destination, source e type - pacote 2……...14

2.14 Endereço físico do Gateway….…….……..16

2.15 arp -a……..……..16

2.16 Comparação Endereço físico com os campos……...……..17

3. Parte 2: Tecnologia Wifi……..18

3.1 Formato dos quadros Wifi……18

3.2 Controle de Acesso ……...20

3.3 Atividade considerando a interface Wifi ……..……....21

3.4 Execução do Wireshark em modo monitor …..……..……...……28

4. Conclusão ……...……...…….29

5. Referência ……...……..30

1.Preparo Inicial

1.1 Instalando o wireshark

Para a instalação do software de rede foram necessários os seguintes comando no terminal do Ubuntu:

- Passo 1. Abra um terminal (use as teclas CTRL + ALT + T);
- Passo 2. Se ainda não tiver, adicione o repositório do programa com este comando;
- sudo add-apt-repository ppa:wireshark-dev/stable
- Passo 3. Atualize o gerenciador de pacotes com o comando: sudo apt-get update
- Passo 4. Agora use o comando para instalar o programa; sudo apt-get install wireshark

Para executar o wireshark basta ir no terminal e digitar "sudo wireshark". Pronto, agora é possível com esta ferramenta solucionar problemas de rede, inspecionar pacotes individuais e capturar pacotes de rede.

2. Parte 1: Tecnologia Ethernet

2.1 Formato dos quadros Ethernet

Os campos principais do quadro ETHERNET são:

- **Campos Preâmbulo e Delimitador:**
  Os campos Preâmbulo (7 bytes) e Delimitador de início de quadro (SFD), também chamado de início de quadro(1 byte), são usados para a sincronização entre os dispositivos de envio e recebimento. Esses primeiros oito bytes do quadro são usados para chamar a atenção dos nós receptores. Essencialmente, os primeiros bytes informam aos receptores para se prepararem para receber um novo quadro.

- **Campo Endereço MAC Destino:**
  Esse campo de 6 bytes é o identificador do destinatário desejado. Como você se lembrará, esse endereço é usado pela camada 2 para auxiliar os dispositivos a determinar se um quadro é endereçado a eles. O endereço no quadro é comparado ao endereço MAC no dispositivo. Se houver correspondência, o dispositivo aceitará o quadro.

- **Campo Endereço MAC origem:**
  Esse campo de 6 bytes identifica a placa de rede ou a interface de origem do quadro.

- **Campo Comprimento:**
  Em qualquer padrão IEEE 802.3 antes de 1997, o campo Comprimento define o comprimento exato do campo de dados do quadro. Isso é usado posteriormente como parte do FCS para garantir que a mensagem foi recebida corretamente. Caso contrário, o propósito do campo é descrever qual é o protocolo de camada superior existente. Se o valor de dois octetos for igual ou superior a 0x0600 hexadecimal ou 1536 decimal, o conteúdo do campo Dados será decodificado de acordo com o protocolo EtherType

indicado. Considerando que, se o valor for igual a ou menor que 0x05DC hexadecimal ou 1500 decimal, o campo Comprimento será usado para indicar o uso do formato de quadro IEEE 802.3. É assim que os quadros Ethernet II e 802.3 são diferenciados.

- **Campo Dados:**
  Esse campo (46 a 1500 bytes) contém os dados encapsulados de um nível superior, que é uma PDU genérica de Camada 3 ou, mais comumente, um pacote IPv4. Todos os quadros devem ter pelo menos 64 bytes de comprimento. Se um pacote pequeno for encapsulado, os bits adicionais chamados de pad serão usados para aumentar o tamanho do quadro até seu tamanho mínimo.

- **Campo Sequência de verificação de quadro:**
  O campo Sequência de verificação de quadro (FCS)(4 bytes) é usado para detectar erros em um quadro. Ele usa uma verificação de redundância cíclica (CRC). O dispositivo emissor inclui os resultados de uma CRC no campo de FCS do quadro. O dispositivo receptor recebe o quadro e gera uma CRC para buscar erros. Se o cálculo que não correspondem é a indicação de que os dados mudaram; portanto, o quadro será descartado. Uma mudança nos dados pode ser o resultado da interrupção dos sinais elétricos que representam os bits.

Campos do quadro ETHERNET com com suas quantidade de bytes:

IEEE 802.3

7	1	6	6	2	46 a 1500	4
Preâmbulo	Delimitador de Início de Quadro	Endereço Destino	Endereço Origem	Comprimento	802.2 Cabeçalho e Dados	Sequência de Verificação de Quadro

Imagem 1.

2.2 Envio de quadros Ethernet

Imagine uma empresa com correio interdepartamental em que uma pessoa pode enviar documentos a outra pessoa em sua organização privada / local. O conteúdo é colocado em um envelope interno e o remetente escreve seu nome e departamento no campo “De”, depois escreve o nome do destinatário e o departamento no campo “Para”.

Quando o envelope é enviado, a sala de correspondência reconhece o envelope de uso interno, lê o nome e o departamento de destino, usa um diretório para traduzir essas informações em um local físico (prédio / escritório) e entrega ao destinatário. O envelope nunca sai da organização privada / local e toda a movimentação é feita por recursos locais familiarizados com o meio ambiente.

Um envelope interno não pode ser enviado para fora da empresa porque o envelope não possui um endereço de correspondência. Para enviar o conteúdo a um escritório fora da área local, o envelope entre os escritórios deverá ser colocado dentro de um envelope postal e

etiquetado com um endereço postal adequado.

Um quadro Ethernet funciona de maneira semelhante. É um contêiner de dados com endereço de origem e destino para entregar informações, chamado de carga útil, entre dois locais na mesma rede. Em vez de um nome e departamento, os endereços de origem e de destino de um quadro são o endereço MAC (Media Access Controller) de um computador, tablet, telefone IP, dispositivo IoT, etc. Este é um número de ID único para cada dispositivo Ethernet em todo o mundo.

Os quadros são gerados na camada 2 da pilha TCP / IP pelo dispositivo de interface de rede com um tamanho de carga útil que depende do tipo de dados que está sendo transmitido. O quadro é enviado para a rede, onde um switch Ethernet verifica o endereço de destino do quadro em relação a uma tabela de pesquisa MAC em sua memória. A tabela de pesquisa informa ao switch qual porta física, ou seja, a porta RJ45, está associada ao dispositivo cujo endereço MAC corresponde ao endereço de destino do quadro.

2.3 Escolhendo interface Ethernet

Nesta parte será exibida a tela principal do wireshark que é similar a tela a seguir. Pode ter pequenas diferenças dependendo da versão. O processo principal para iniciar o uso do wireshark é a captura dos pacotes da rede. Você precisa identificar qual é a interface de rede utilizada. No caso da imagem abaixo, a parte em azul é a interface cabeada. Após escolhida a interface é só clicar.

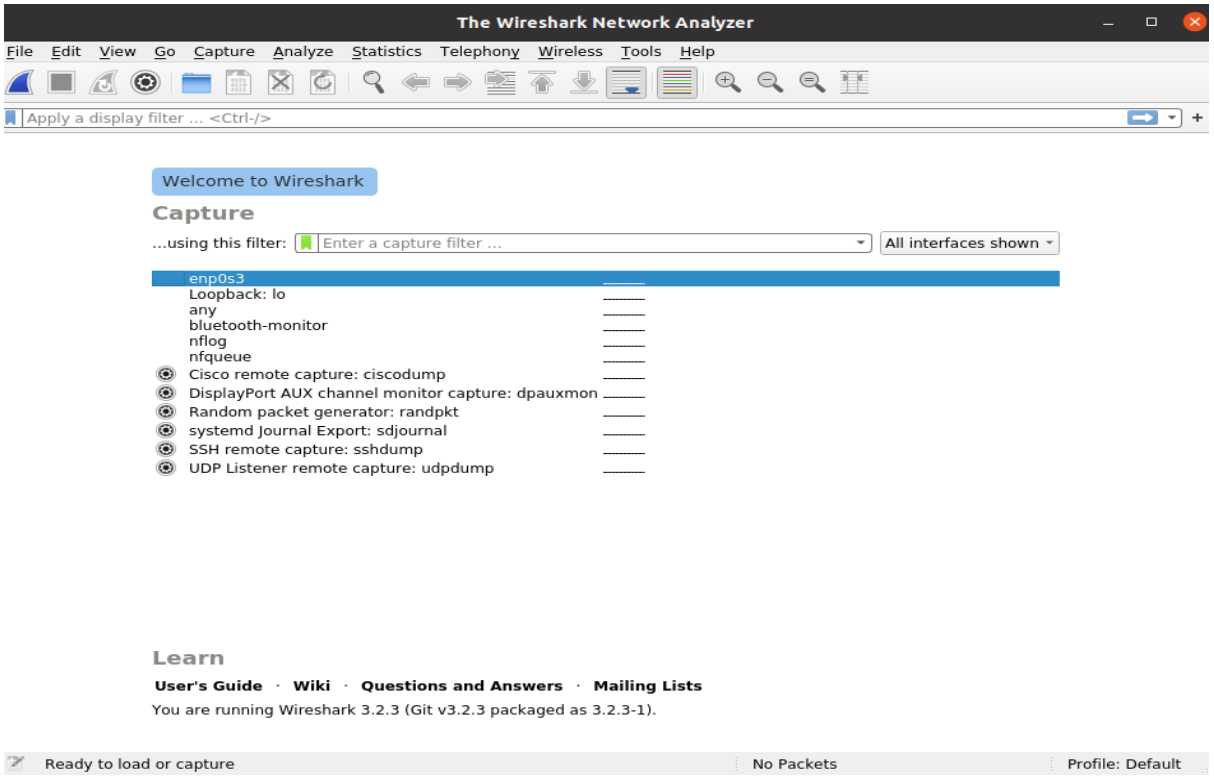


Imagem 2.

2.4 ifconfig - endereço gateway

Utilizando o comando “ifconfig” e o comando “ip route show” para achar o gateway padrão, como mostrado na imagem abaixo. Diante disso, foi utilizado o segundo comando para ter certeza que estava-se pegando o gateway correto. Para prosseguir com a prática foi escolhido o gateway padrão. Dessa forma, antes de prosseguir, é importante ressaltar o que é o gateway padrão. O gateway padrão normalmente é expresso como uma variação do mesmo endereço para o endereço IPv4, ele normalmente é o endereço IP do roteador que conecta-se a

rede interna com uma rede externa (internet). O gateway padrão é o dispositivo que roteia o tráfego da rede local para dispositivos de rede remota.

```
gabriel@gabriel-VirtualBox: ~
gabriel@gabriel-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::9bcb:a327:4672:85a8  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:0a:05:71  txqueuelen 1000  (Ethernet)
        RX packets 3151  bytes 2601414 (2.6 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2429  bytes 394676 (394.6 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  (Loopback Local)
        RX packets 388  bytes 31901 (31.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 388  bytes 31901 (31.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

gabriel@gabriel-VirtualBox:~$ ip route show
default via 10.0.2.2 dev enp0s3 proto dhcp metric 100
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100
169.254.0.0/16 dev enp0s3 scope link metric 1000
```

Imagem 3.

2.5 Ping para o ip do gateway

O ping é um comando que serve para testar a conectividade entre equipamentos de uma rede. Ele basicamente envia dados para esses aparelhos e fica aguardando as respostas. Se o equipamento responder, significa que está ativo. Logo, foi usado esse comando para o gateway padrão para ver se este está ativo e mandando respostas. Dessa forma, o comando ping permite verificar, de maneira simples, se o computador está conectado a internet. Ao executar o comando para o gateway, o computador envia pacotes de informações para um host de destino e verifica se há respostas Além disso , ele mede a latência, ou seja, o tempo para a comunicação com outro ponto.

```
gabriel@gabriel-VirtualBox: ~
gabriel@gabriel-VirtualBox:~$ ping 10.0.2.2
PING 10.0.2.2 (10.0.2.2) 56(84) bytes of data.
64 bytes de 10.0.2.2: icmp_seq=1 ttl=64 tempo=0.417 ms
64 bytes de 10.0.2.2: icmp_seq=2 ttl=64 tempo=0.583 ms
64 bytes de 10.0.2.2: icmp_seq=3 ttl=64 tempo=0.595 ms
64 bytes de 10.0.2.2: icmp_seq=4 ttl=64 tempo=0.722 ms
64 bytes de 10.0.2.2: icmp_seq=5 ttl=64 tempo=0.866 ms
64 bytes de 10.0.2.2: icmp_seq=6 ttl=64 tempo=0.815 ms
64 bytes de 10.0.2.2: icmp_seq=7 ttl=64 tempo=0.808 ms
64 bytes de 10.0.2.2: icmp_seq=8 ttl=64 tempo=0.866 ms
64 bytes de 10.0.2.2: icmp_seq=9 ttl=64 tempo=0.336 ms
64 bytes de 10.0.2.2: icmp_seq=10 ttl=64 tempo=0.572 ms
64 bytes de 10.0.2.2: icmp_seq=11 ttl=64 tempo=0.832 ms
64 bytes de 10.0.2.2: icmp_seq=12 ttl=64 tempo=0.731 ms
64 bytes de 10.0.2.2: icmp_seq=13 ttl=64 tempo=0.949 ms
64 bytes de 10.0.2.2: icmp_seq=14 ttl=64 tempo=0.952 ms
64 bytes de 10.0.2.2: icmp_seq=15 ttl=64 tempo=0.860 ms
64 bytes de 10.0.2.2: icmp_seq=16 ttl=64 tempo=0.658 ms
64 bytes de 10.0.2.2: icmp_seq=17 ttl=64 tempo=0.870 ms
64 bytes de 10.0.2.2: icmp_seq=18 ttl=64 tempo=0.379 ms
64 bytes de 10.0.2.2: icmp_seq=19 ttl=64 tempo=0.928 ms
64 bytes de 10.0.2.2: icmp_seq=20 ttl=64 tempo=0.747 ms
64 bytes de 10.0.2.2: icmp_seq=21 ttl=64 tempo=0.982 ms
64 bytes de 10.0.2.2: icmp_seq=22 ttl=64 tempo=0.674 ms
```

Imagem 4.

Não é pedido para mostrar, mas estas são as estatísticas de ping

```
--- 10.0.2.2 estatísticas de ping ---
1495 pacotes transmitidos, 1495 recebidos, 0% perda de pacote, tempo 1524005ms
rtt min/avg/max/mdev = 0.118/0.578/1.311/0.199 ms
gabriel@gabriel-VirtualBox:~$
```

Imagem 5.

=> Finalizada a captura do wireshark

2.6 Tráfego Ethernet

Nessa parte do trabalho foi pedido para mostrar todo o tráfego gerado pelo Ethernet. Dessa forma, o wireshark passa a exibir uma listagem com todos os pacotes de dados transmitidos na placa de rede, permitindo identificar possíveis problemas através das cores com que cada item é apresentado. No exemplo abaixo, cada linha representa um pacote(um unidade de dados - em cada camada do modelo OSI ou da arquitetura TCP/IP o PDU - Protocol Data Unit - tem seu nome mais adequado (segmento, pacote/datagrama, quadro/frame), mas é genericamente chamado de pacote) que foi enviado através da rede que o wireshark detectou.

Mesmo que você não esteja solicitando ativamente qualquer informação(como pedir ao navegador para requisitar uma página na web) o computador e os outros dispositivos da rede estão constantemente enviando mensagens entre si. Além disso, por si só, os programas no computador também podem estar enviando informações através da rede (solicitações de atualização de software seria um exemplo). A tela abaixo mostra um exemplo de pacotes de rede que foram capturados pelo wireshark.

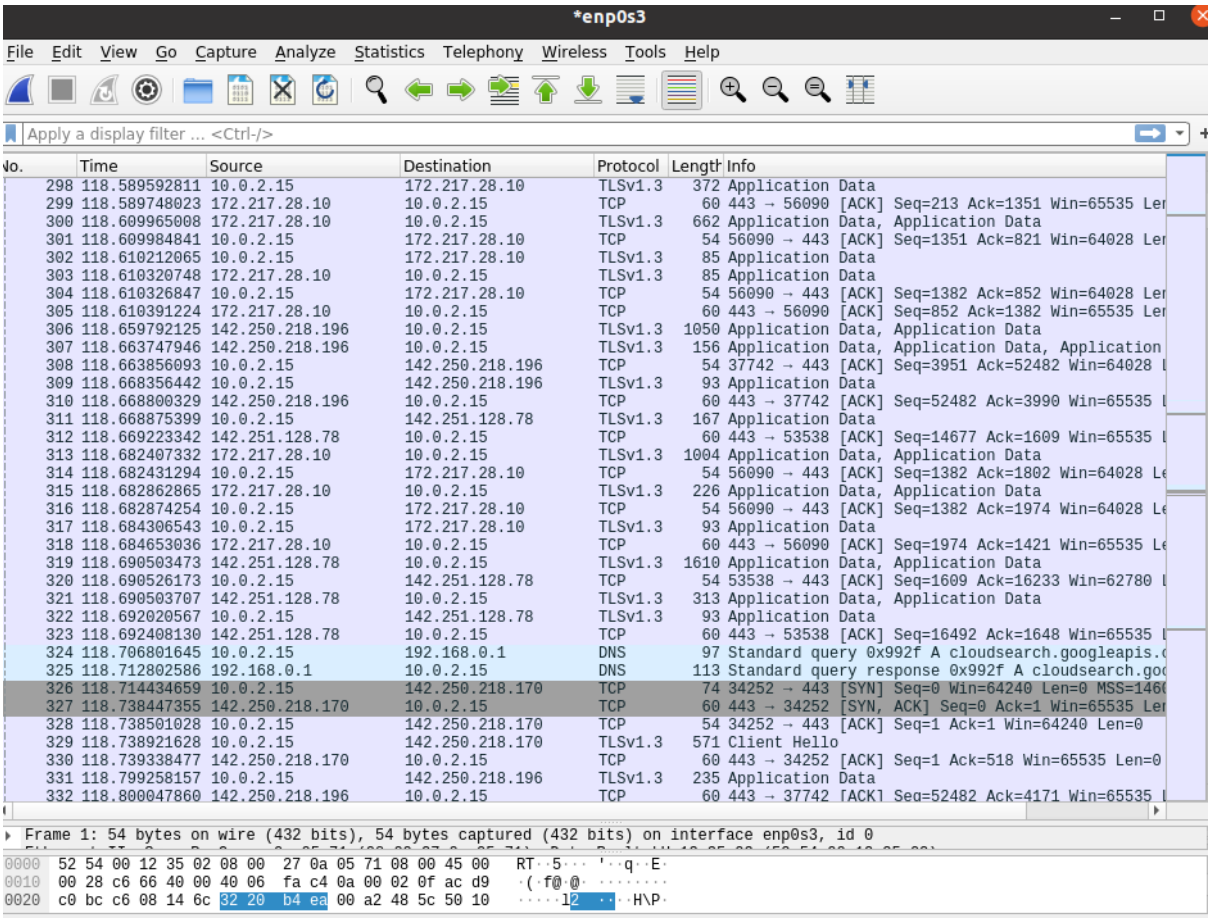


Imagem 6.

2.7 Tráfego gerado pelo ICMP

Nesta parte foi pedido para verificar o tráfego gerado pelo icmp, que é o Internet Control Message Protocol(ICMP) este é uma parte do protocolo IP, ele é usado para solucionar problemas de outros protocolos. É interessante ressaltar que a utilização do comando ping acima, é a mesma utilização do protocolo ICMP.

O padrão de ping a partir de uma linha de comando no windows pinga um host quatro vezes. Você pode ver o processo de ping no arquivo de captura na figura abaixo também.

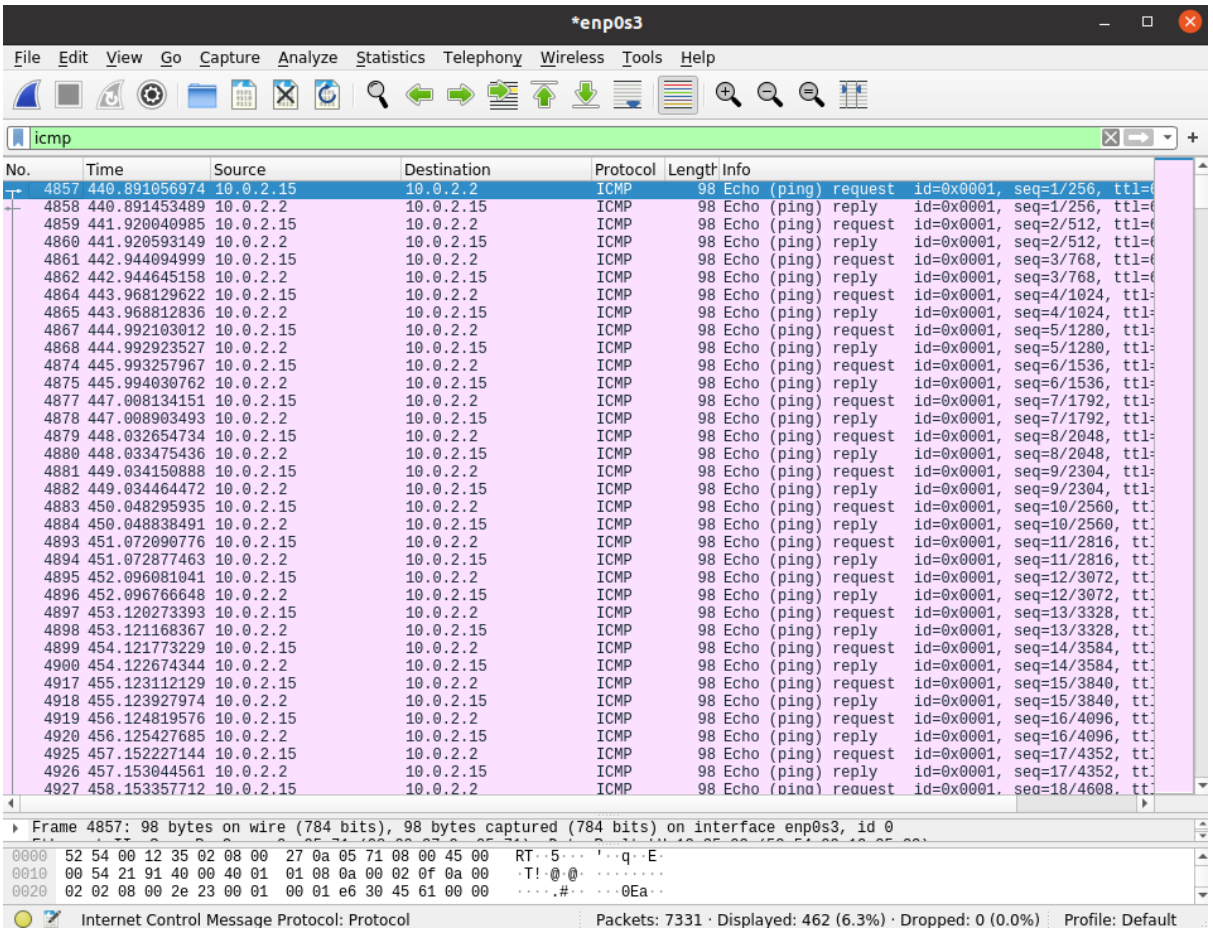


Imagem 7.

2.8 Análise do Pacote 1

Análise de pacotes, muitas vezes referida como packet sniffing (farejamento de pacotes) ou análise de protocolo, descreve o processo de capturar e interpretar dados em tempo real à medida que flui através de uma rede a fim de entender melhor o que está acontecendo na rede. Análise de pacotes é normalmente realizada por um packet sniffer (farejador de pacotes), ferramenta utilizada para capturar dados brutos atravessando os fios de uma rede. A análise de pacotes podem nos ajudar a entender características da rede, saber quem está em uma rede, ou determinar qual é a utilização da largura de banda disponível, identificar as épocas de pico de uso da rede, identificar possíveis ataques ou atividades maliciosas, e encontrar aplicações inseguras.

Existem vários tipos de programas farejadores de pacotes (Packets sniffing), incluindo tanto o software livre como o comercial. Cada programa foi concebido com objetivos diferentes em mente. Alguns dos mais populares programas de análise de pacotes são tcpdump (Um programa de linha de comando), OmniPeek e Wireshark (ambos baseados em GUI sniffers). Diante disso, foi escolhido o pacote abaixo para ser analisado.

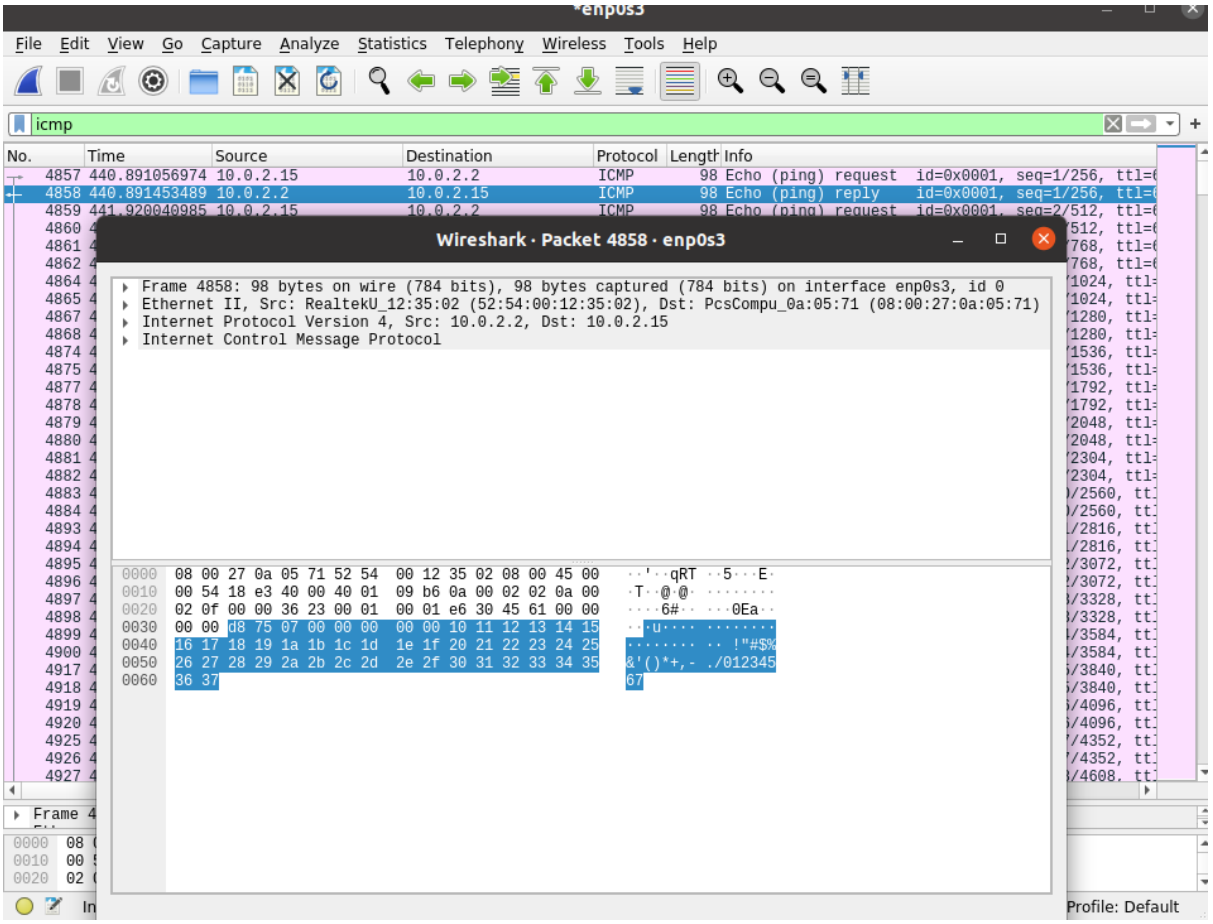


Imagem 8.

2.9 Quadro Frame

O quadro (frame) é um dos pacotes mais informativos em uma transmissão de uma rede sem fio . Um quadro frame é enviado como um pacote de difusão WAP através de um canal da rede sem fio notificando a todos os clientes sem fio que estão a escutar informando que o WAP está disponível e definir os parâmetros a serem definidos quando se conectar a ele. Portanto, esse tipo de pacote de transmissão contém uma grande quantidade de informações úteis, como mostrado na “imagem 9”.

```
▾ Frame 4858: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface enp0s3, id 0
  ▸ Interface id: 0 (enp0s3)
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 17, 2021 21:20:54.489337493 -03
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1631924454.489337493 seconds
    [Time delta from previous captured frame: 0.000396515 seconds]
    [Time delta from previous displayed frame: 0.000396515 seconds]
    [Time since reference or first frame: 440.891453489 seconds]
    Frame Number: 4858
    Frame Length: 98 bytes (784 bits)
    Capture Length: 98 bytes (784 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:icmp:data]
    [Coloring Rule Name: ICMP]
    [Coloring Rule String: icmp || icmpv6]
▸ Ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_0a:05:71 (08:00:27:0a:05:71)
▸ Internet Protocol Version 4, Src: 10.0.2.2, Dst: 10.0.2.15
▸ Internet Control Message Protocol

0000  08 00 27 0a 05 71 52 54  00 12 35 02 08 00 45 00   ··'··qRT ··5···E·
0010  00 54 18 e3 40 00 40 01  09 b6 0a 00 02 02 0a 00   ·T··@·@· ........
0020  02 0f 00 00 36 23 00 01  00 01 e6 30 45 61 00 00   ····6#·· ···0Ea··
0030  00 00 d8 75 07 00 00 00  00 00 10 11 12 13 14 15   ···u···· ........
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ........ ·· !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                              67
```

Imagem 9.

2.10 Quadro Ethernet II

```
▸ Frame 4858: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface enp0s3, id 0
▾ Ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_0a:05:71 (08:00:27:0a:05:71)
  ▾ Destination: PcsCompu_0a:05:71 (08:00:27:0a:05:71)
      Address: PcsCompu_0a:05:71 (08:00:27:0a:05:71)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ▾ Source: RealtekU_12:35:02 (52:54:00:12:35:02)
      Address: RealtekU_12:35:02 (52:54:00:12:35:02)
      .... ..1. .... .... .... .... = LG bit: Locally administered address (this is NOT the factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 10.0.2.2, Dst: 10.0.2.15
▸ Internet Control Message Protocol

0000  08 00 27 0a 05 71 52 54  00 12 35 02 08 00 45 00   ··'··qRT ··5···E·
0010  00 54 18 e3 40 00 40 01  09 b6 0a 00 02 02 0a 00   ·T··@·@· ........
0020  02 0f 00 00 36 23 00 01  00 01 e6 30 45 61 00 00   ····6#·· ···0Ea··
0030  00 00 d8 75 07 00 00 00  00 00 10 11 12 13 14 15   ···u···· ........
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ........ ·· !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                              67
```

Imagem 10.

2.11 Destination, source e type - Pacote 1

Destination:

=> Campo Destination selecionado.

Destination: endereço IP do destino do pacote.

Como já dito anteriormente, este campo é o campo endereço MAC destino: Como mostrado na parte inferior da imagem, onde está mostra 6 valores sublinhados em azul. Ou seja, o endereço no quadro é comparado ao endereço MAC no dispositivo. Se houver correspondência, o dispositivo aceitará o quadro.

```
▸ Frame 4858: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface enp0s3, id 0
▾ Ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_0a:05:71 (08:00:27:0a:05:71)
  ▾ Destination: PcsCompu_0a:05:71 (08:00:27:0a:05:71)
      Address: PcsCompu_0a:05:71 (08:00:27:0a:05:71)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ▾ Source: RealtekU_12:35:02 (52:54:00:12:35:02)
      Address: RealtekU_12:35:02 (52:54:00:12:35:02)
      .... ..1. .... .... .... .... = LG bit: Locally administered address (this is NOT the factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 10.0.2.2, Dst: 10.0.2.15
▸ Internet Control Message Protocol

0000  08 00 27 0a 05 71 52 54  00 12 35 02 08 00 45 00   ··'··qRT ··5···E·
0010  00 54 18 e3 40 00 40 01  09 b6 0a 00 02 02 0a 00   ·T··@·@· ........
0020  02 0f 00 00 36 23 00 01  00 01 e6 30 45 61 00 00   ····6#·· ···0Ea··
0030  00 00 d8 75 07 00 00 00  00 00 10 11 12 13 14 15   ···u···· ........
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ........ ·· !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                              67
```

Imagem 11.

Source:

=> Campo Source selecionado.

Source: endereço IP da origem do pacote.

Esse campo de 6 bytes identifica a placa de rede ou a interface de origem do quadro, os 6 bytes estão mostrados no quadro abaixo.

```
▸ Frame 4858: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface enp0s3, id 0
▾ Ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_0a:05:71 (08:00:27:0a:05:71)
  ▾ Destination: PcsCompu_0a:05:71 (08:00:27:0a:05:71)
      Address: PcsCompu_0a:05:71 (08:00:27:0a:05:71)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ▸ Source: RealtekU_12:35:02 (52:54:00:12:35:02)
    Type: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 10.0.2.2, Dst: 10.0.2.15
▸ Internet Control Message Protocol

0000  08 00 27 0a 05 71 52 54  00 12 35 02 08 00 45 00   ··'··qRT ··5···E·
0010  00 54 18 e3 40 00 40 01  09 b6 0a 00 02 02 0a 00   ·T··@·@· ........
0020  02 0f 00 00 36 23 00 01  00 01 e6 30 45 61 00 00   ····6#·· ···0Ea··
0030  00 00 d8 75 07 00 00 00  00 00 10 11 12 13 14 15   ···u···· ........
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ........ ·· !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                              67
```

Imagem 12.

Type:

=> Campo type selecionado.

Este é o campo comprimento, o campo comprimento define o comprimento exato do campo do campo de dados do quadro. Este campo está representado abaixo com seus 2 bytes sublinhados em azul.

```
▸ Frame 4858: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface enp0s3, id 0
▾ Ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_0a:05:71 (08:00:27:0a:05:71)
  ▾ Destination: PcsCompu_0a:05:71 (08:00:27:0a:05:71)
      Address: PcsCompu_0a:05:71 (08:00:27:0a:05:71)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ▸ Source: RealtekU_12:35:02 (52:54:00:12:35:02)
    Type: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 10.0.2.2, Dst: 10.0.2.15
▸ Internet Control Message Protocol

0000  08 00 27 0a 05 71 52 54  00 12 35 02 08 00 45 00   ··'··qRT ··5···E·
0010  00 54 18 e3 40 00 40 01  09 b6 0a 00 02 02 0a 00   ·T··@·@· ........
0020  02 0f 00 00 36 23 00 01  00 01 e6 30 45 61 00 00   ····6#·· ···0Ea··
0030  00 00 d8 75 07 00 00 00  00 00 10 11 12 13 14 15   ···u···· ........
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ........ ·· !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                              67
```

Imagem 13.

2.12 Análise do Pacote 2

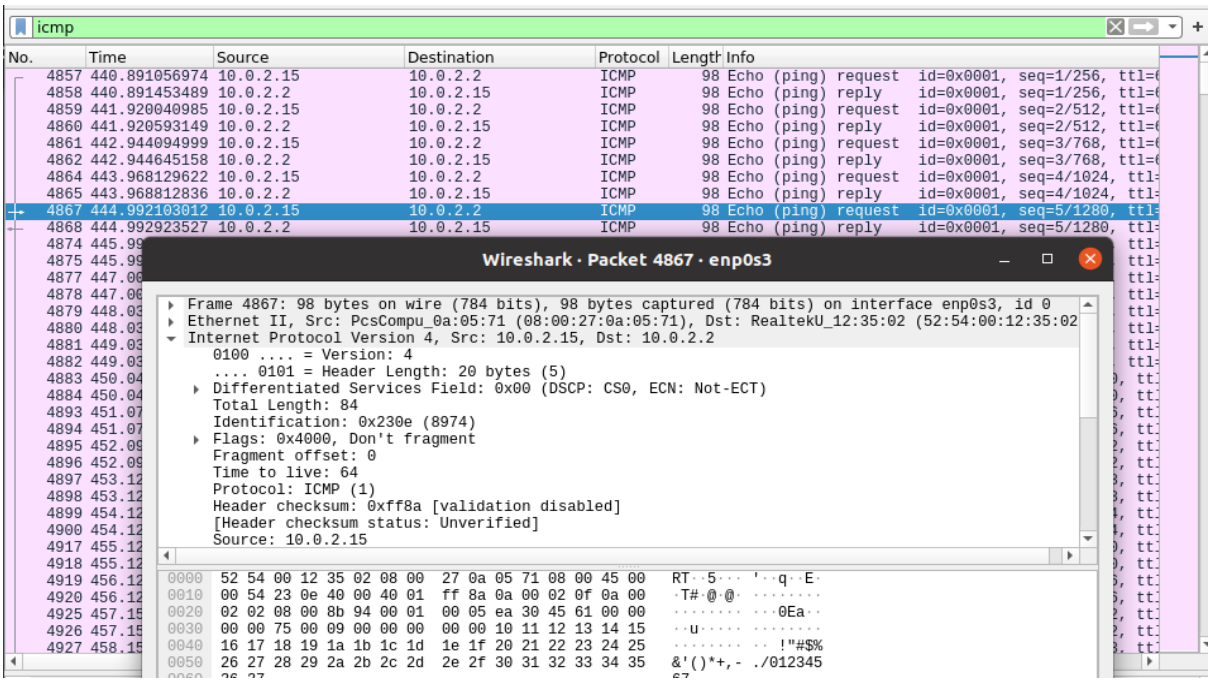


Imagem 14.

2.13 Frame, Destination, source e type - pacote 2

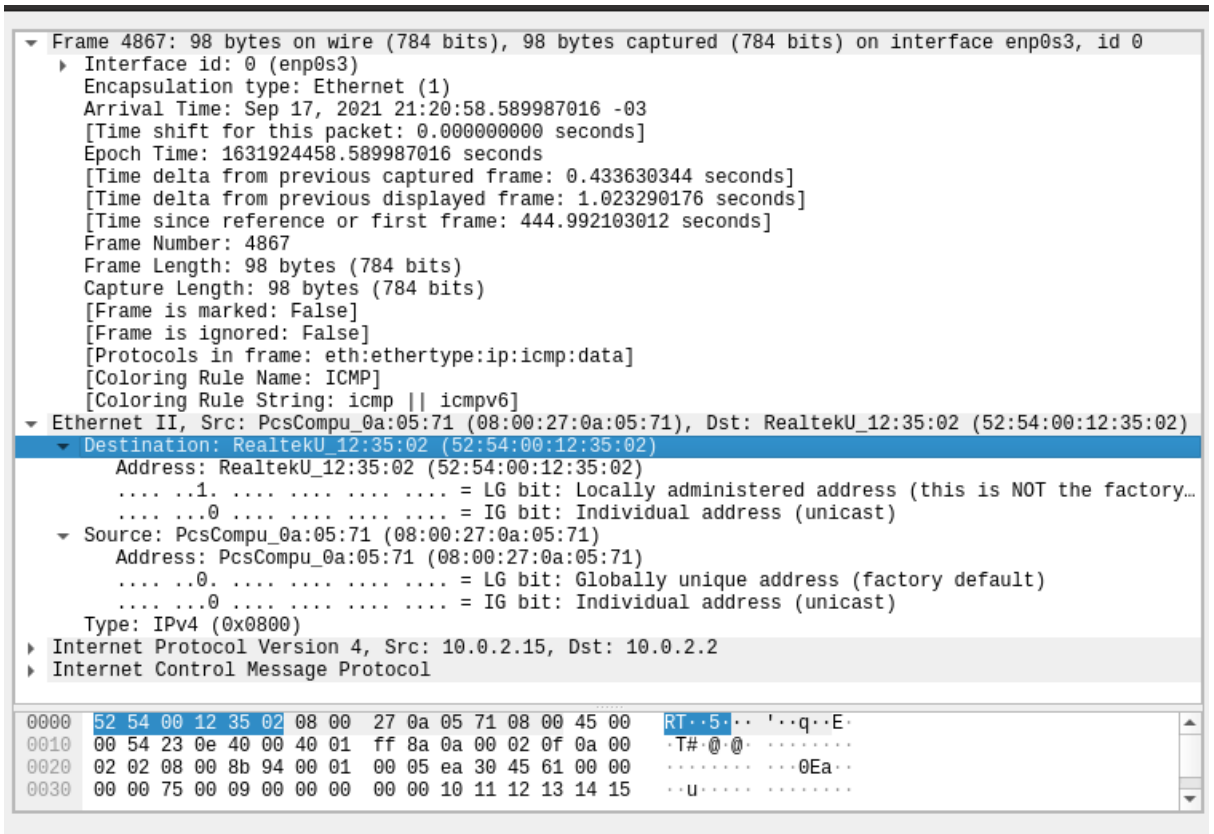


Imagem 15.

Source:

```
▾ Frame 4867: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface enp0s3, id 0
  ▸ Interface id: 0 (enp0s3)
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 17, 2021 21:20:58.589987016 -03
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1631924458.589987016 seconds
    [Time delta from previous captured frame: 0.433630344 seconds]
    [Time delta from previous displayed frame: 1.023290176 seconds]
    [Time since reference or first frame: 444.992103012 seconds]
    Frame Number: 4867
    Frame Length: 98 bytes (784 bits)
    Capture Length: 98 bytes (784 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:icmp:data]
    [Coloring Rule Name: ICMP]
    [Coloring Rule String: icmp || icmpv6]
▾ Ethernet II, Src: PcsCompu_0a:05:71 (08:00:27:0a:05:71), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
  ▾ Destination: RealtekU_12:35:02 (52:54:00:12:35:02)
      Address: RealtekU_12:35:02 (52:54:00:12:35:02)
      .... ..1. .... .... .... .... = LG bit: Locally administered address (this is NOT the factory…
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ▾ Source: PcsCompu_0a:05:71 (08:00:27:0a:05:71)
      Address: PcsCompu_0a:05:71 (08:00:27:0a:05:71)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.2.2
▸ Internet Control Message Protocol

0000  52 54 00 12 35 02 08 00  27 0a 05 71 08 00 45 00   RT··5··· '··q··E·
0010  00 54 23 0e 40 00 40 01  ff 8a 0a 00 02 0f 0a 00   ·T#·@·@· ........
0020  02 02 08 00 8b 94 00 01  00 05 ea 30 45 61 00 00   ........ ···0Ea··
0030  00 00 75 00 09 00 00 00  00 00 10 11 12 13 14 15   ··u····· ........
```

Imagem 16.

Type:

```
▾ Frame 4867: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface enp0s3, id 0
  ▸ Interface id: 0 (enp0s3)
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 17, 2021 21:20:58.589987016 -03
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1631924458.589987016 seconds
    [Time delta from previous captured frame: 0.433630344 seconds]
    [Time delta from previous displayed frame: 1.023290176 seconds]
    [Time since reference or first frame: 444.992103012 seconds]
    Frame Number: 4867
    Frame Length: 98 bytes (784 bits)
    Capture Length: 98 bytes (784 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:icmp:data]
    [Coloring Rule Name: ICMP]
    [Coloring Rule String: icmp || icmpv6]
▾ Ethernet II, Src: PcsCompu_0a:05:71 (08:00:27:0a:05:71), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
  ▾ Destination: RealtekU_12:35:02 (52:54:00:12:35:02)
      Address: RealtekU_12:35:02 (52:54:00:12:35:02)
      .... ..1. .... .... .... .... = LG bit: Locally administered address (this is NOT the factory…
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ▾ Source: PcsCompu_0a:05:71 (08:00:27:0a:05:71)
      Address: PcsCompu_0a:05:71 (08:00:27:0a:05:71)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.2.2
▸ Internet Control Message Protocol

0000  52 54 00 12 35 02 08 00  27 0a 05 71 08 00 45 00   RT··5··· '··q··E·
0010  00 54 23 0e 40 00 40 01  ff 8a 0a 00 02 0f 0a 00   ·T#·@·@· ........
0020  02 02 08 00 8b 94 00 01  00 05 ea 30 45 61 00 00   ........ ···0Ea··
0030  00 00 75 00 09 00 00 00  00 00 10 11 12 13 14 15   ··u····· ........
```

Imagem 17.

2.14 Endereço físico do Gateway

Utilizando o ifconfig novamente

O endereço físico aparece na frente da palavra ether: 08:00:27:0a:05:71

```
gabriel@gabriel-VirtualBox: ~
gabriel@gabriel-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::9bcb:a327:4672:85a8  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:0a:05:71  txqueuelen 1000  (Ethernet)
        RX packets 24850  bytes 20271214 (20.2 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 17772  bytes 2486146 (2.4 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  (Loopback Local)
        RX packets 1035  bytes 99506 (99.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1035  bytes 99506 (99.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Imagem 18.

Compare o endereço físico do computador com os Campos Source e destination:

Comparando com o 1º pacote:

- No destination do pacote 1 foi enviado um quadro com o seguinte endereço "PcsCompu_0a:05:71". Este foi enviado para o endereço físico da máquina utilizada:(08:00:27:0a:05:71).
- No Source do pacote 1 temos o pacote com o seguinte endereço "RealteKU_12:35:02" que se originou do seguinte endereço: (52:54:00:12:35:02).

Comparando com o 2º pacote:

- No destination foi enviado um quadro com o seguinte endereço "RealteKU_12:35:02". Este foi enviado para o endereço (52:54:00:12:35:02).
- No source tem-se o pacote "PcsCompu_0a:05:71". que se originou do seguinte endereço:(08:00:27:0a:05:71), no caso o endereço físico da máquina usada.

2.15 arp -a

Cache ARP:

- Para manter o número de broadcasts a um nível mínimo, os hosts que usam o ARP mantêm um cache de mapeamentos Internet-Ethernet já resolvidos pois, assim, não precisam usar o ARP toda hora que se quiser transmitir um pacote.
- Antes de transmitir um pacote o host sempre examina o seu cache ARP, buscando verificar se já existe mapeamento anterior para o endereço destino.

Resumindo:

- O cache ARP (ou Tabela ARP) é uma estrutura que mantém os mais recentes mapeamentos de endereços IP em endereços físicos.
- Quando o Quando o host origem A recebe a resposta do recebe a resposta do host destino B, ele guarda no seu cache o endereço IP (IB) e o endereço físico (FB) de B.
- Quando B recebe o broadcast de A pedindo seu endereço

- físico, B guarda no seu cache os valores de IA e FA.
- O mapeamento endereço IP x endereço físico residirá no cache por um certo período. Esse tempo é denominado de TTL (Time To Live).

```
gabriel@gabriel-VirtualBox: ~
gabriel@gabriel-VirtualBox:~$ arp -a
_gateway (10.0.2.2) em 52:54:00:12:35:02 [ether] em enp0s3
gabriel@gabriel-VirtualBox:~$
```

Imagem 19.

=> verificando o endereço físico do gateway

```
gabriel@gabriel-VirtualBox: ~
gabriel@gabriel-VirtualBox:~$ route
Tabela de Roteamento IP do Kernel
Destino         Roteador        MáscaraGen.     Opções Métrica Ref    Uso Iface
default         _gateway        0.0.0.0         UG     100    0        0 enp0s3
10.0.2.0        0.0.0.0         255.255.255.0   U      100    0        0 enp0s3
link-local      0.0.0.0         255.255.0.0     U      1000   0        0 enp0s3
gabriel@gabriel-VirtualBox:~$
```

Imagem 20.

2.16 Comparação Endereço físico com os campos

O endereço físico do gateway padrão: 52:54:00:12:35:02

Comparando com o 1º pacote

- No campo destination tem-se o pacote com o seguinte endereço que foi: ”PcsCompu_0a:05:71”, como já dito antes este foi enviado para o endereço da máquina.
- No source o endereço “RealteKU_12:35:02” foi recebido do endereço físico do gateway.

Comparando com o 2º pacote:

- No destination foi enviado um quadro com o seguinte endereço “RealteKU_12:35:02”. Este foi enviado para o endereço (52:54:00:12:35:02). no caso o endereço físico do gateway.
- No source tem-se o pacote “PcsCompu_0a:05:71”. que se originou do seguinte endereço:(08:00:27:0a:05:71), no caso o endereço físico da máquina usada.

3. Parte 2: Tecnologia Wifi

3.1 Formatos dos quadros Wifi

Os campos apresentados no formato do quadro Wifi são, onde o padrão IEEE 802.11 define o formato dos quadros trocados, cada quadro contém um cabeçalho que tem o comprimento de 30 bytes, um corpo e um FCS (4 bytes que permite a correção de erros), o tamanho mínimo do quadro é 34 bytes e o máximo é 2346 bytes.

MAC Header

Frame control	Duration / ID	Address 1	Address 2	Address 3	Sequence control	Address 4	Frame body	FCS
2B	2B	6B	6B	6B	6B	2B	0–2312B	4B

Imagem 21.

Na imagem o MAC Header é o cabeçalho, o FCS é o Frame check sequency de 4 bytes, na imagem também estão sendo apresentadas a quantidade de bytes para cada quadro. Agora, serão descritos o funcionamento de cada um dos quadros apresentados.

Controle do quadro: é um campo com 2 bytes que é constituído pelas informações do quadro que está sendo apresentado abaixo:

Protocol Version	Type	Subtype	To DS	From DS	More Fragm.	Retry	Power Manag.	More data	WEP	Order
2b	2b	4b	1b	1b	1b	1b	1b	1b	1b	1b

Imagem 22.

No controle de quadro que é apresentado acima, temos:

Campo Versão do protocolo: que indica a versão do protocolo wifi 802.11 que está sendo utilizado, assim as entidades de rede receptoras irão usar esse valor para determinar se a versão do protocolo do quadro que está sendo recebida é suportada.

Campos Tipo e Subtipo: Identificam uma das três funções e subfunções do quadro, sendo elas, controle, dados e gerenciamento. São respectivamente 2 e 4 bits que determinam a função do quadro, onde são múltiplos subtipos para cada tipo de quadro associado, o gerenciamento corresponde aos pedidos de associação e também as mensagens de anúncio do ponto de acesso, o tipo de controle é utilizado para o acesso aos meios de comunicação para pedir autorização para emitir e os dados são referente ao envio de dados, que é a maior parte do tráfego da rede.

Campo para DS (Sistema de Distribuição): definido como 1Bit nos quadros de dados destinados ao sistema de distribuição, qualquer quadro enviado por uma estação com destino a um ponto de acesso possui assim um campo DS posicionado em 1, se o quadro decorre do sistema de distribuição esse bit vale 1, então quando tem-se uma comunicação direta entre duas entidades de rede os campos são definidos em 0.

Campo Mais Fragmentos: definido como 1 em quadros que tenham outro fragmento. Um bit que permite indicar que ainda são fragmentos do quadro, sendo dados ou gerenciamento, a transmitir.

Campo Repetir: um bit que indica se a informação corrente, sendo dado ou gerenciamento está ou não retransmitindo devida a uma perda, definido como 1 se o quadro for uma retransmissão do quadro anterior.

Campo Gerenciamento de energia: um bit que indica se a gestão que enviou este fragmento está em modo de economia de energia ou modo ativo, ou seja, definido como 1 para indicar que um nó estará em modo de economia de energia.

Campo mais dados: um bit que indica para uma estação operando em modo de economia de energia que o ponto de acesso tem mais quadros a transmitir, então definido como 1 para indicar a um nó no modo de economia de energia que mais quadros estão em um buffer para aquele nó.

Campo Protocolo WEP: é um bit que indica ou não se está sendo usado no quadro o processo de criptografia e autenticação, podendo ser configurado para todos os quadros de dados e gerenciamento que tem o subtipo configurado para autenticação, onde o processo de criptografia usado é o algoritmo de codificação WEP. definido como 1 se o quadro contiver informações WEP criptografadas para segurança.

Campo Ordem: um bit que indica que o quadro enviado está utilizando a classe de serviço estritamente ordenada, onde os quadros recebidos devem ser processados em ordem. Definido como 1 em um quadro de tipo de dado que usa a classe de serviço Estritamente Ordenado (não precisa reordenar).

Campo de Duração: dão 2 bytes usados para todos os campos de controle exceto com o subtipo PS, Poll para indicar o tempo restante necessário para receber a próxima transmissão, ou seja, dependendo do tipo de quadro, representa o tempo, em microssegundos, exigido para transmitir o quadro ou uma identidade de associação (AID) para a estação que transmitiu o quadro.

Campos de Endereço: um quadro pode ter até 3 endereços além do endereço de 48 bits.

- **Endereço de Destino (DA):** indica o endereço MAC do destino final para a recepção do quadro, sendo assim, o endereço MAC do nó destino final na rede.
- **Endereço de Origem (SA):** endereço fonte, indica o endereço MAC da fonte que originou, ou seja, criou e transmitiu inicialmente o quadro, então é o endereço MAC do nó que iniciou o quadro.
- **Endereço do Receptor (RA):** indica o endereço MAC da próxima estação que irá receber o quadro, ele identifica o dispositivo de rede sem fio que é o destinatário imediato do quadro.
- **Endereço do transmissor (TA):** mostra o endereço MAC da estação que transmitiu o quadro na rede sem fio.

Campo Controle de Sequência: nesse campo são encontrados os campos de Número do Fragmento, que contém 12 bits que indica o número da sequência de cada quadro, onde, esse número é sempre o mesmo para cada quadro enviado para o caso de um quadro fragmentado, já no quadro não fragmentado, o número é incrementado até atingir o valor de 4095 e então retornar o valor zero novamente e fragmento é uma sequência de 4 bits que vai indicar o número para cada fragmento do quadro enviado, começando pelo valor inicial de zero é ir

incrementando para cada fragmento.

Campo corpo do quadro: vai conter a informação específica de dados ou gerenciamento, então contém a informação que está sendo transportada para quadros de dados que são normalmente um pacote IP.

Campo FCS: contém uma verificação de redundância cíclica (CRC) de 32 bits do quadro

Campo Sequência de verificação de quadro: são 4 bytes que o transmissor do quadro aplica um CRC-32, que é a verificação de redundância cíclica sobre todos os campos do cabeçalho MAC e sobre o corpo do quadro para gerar o FCS, então o receptor do quadro utiliza mesmo CRC para determinar o seu próprio valor de FCS e assim verificar se ocorreu ou não erro durante a transmissão, sendo assim, uma soma de controle que serve para verificar a integridade do quadro.

3.2 Controle de acesso

A subcamada de acesso ao meios (MAC) é uma parte da camada enlace de dados responsável por estabelecer uma lógica quanto ao uso do meio de transmissão em topologias de difusão, quanto a rede tem a topologia de difusão, significa que vários nós usam exatamente o mesmo meio para poderem enviar mensagens, nesse tipo de topologia se mais de um nó tentar enviar dados ao mesmo tempo, ocorre o que se chama de colisão, sendo que toda vez que essa colisão ocorre todos os dados enviados são perdidos e precisam ser retransmitidos, então o objetivo do MAC é justamente tentar evitar ao máximo as colisões, pois elas fazem com que a rede se torne mais lenta, para conseguir realizar isso existem vários protocolos que foram desenvolvidos.

O protocolo CSMA com prevenção de colisão, é para redes sem fio, temos o funcionamento dele da seguinte forma, quando um nó deseja se comunicar com outro, pede autorização para ele enviando um sinal RTS (Request To Send), onde se um nó receber o RTS e estiver livre para se comunicar ele vai enviar um sinal chamado CTS (Clear To Send), onde somente depois de receber um CTS um nó vai poderá começar a transmitir dados para outro, então toda vez que um nó não está envolvido na troca de dados percebe um RTS ou CTS na rede, ele fica sem enviar dados por algum tempo, então a taxa de sucesso para esse envio chega a 100%, assim todas as colisões são evitadas pois o os nós só podem enviar dados quando receber a confirmação do receptor que eles podem transmitir sem nenhum problema. Assim, temos que as colisões para esse protocolo só serão possíveis se cada nó possuir um alcance de transmissão diferente ou caso existam nós móveis capazes de se deslocar pela área de transmissão.

3.3 Atividade considerando a interface Wifi

Para escolher a interface wifi basta verificar se tem sinal de rede nela, caso de afirmativo basta clicar duas vezes e a interface é escolhida. Na imagem apresentada abaixo é mostrada a tela inicial do wireshark e a marcação de azul é a escolha da interface wifi, além disso pode-se observar que tem uma alteração na linha ao lado indicando o tráfego na linha.

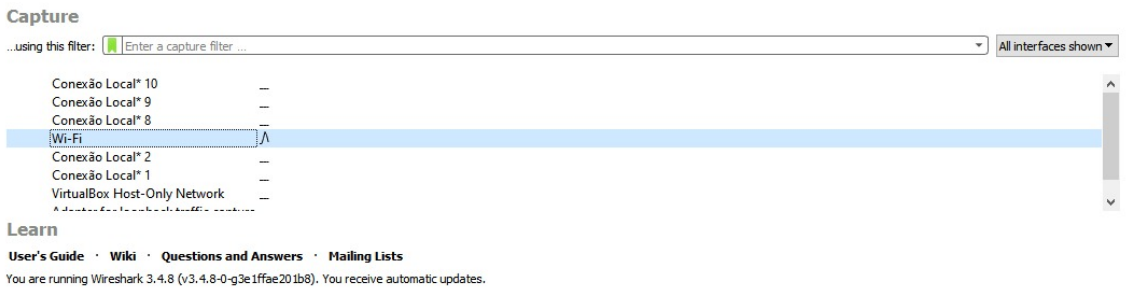
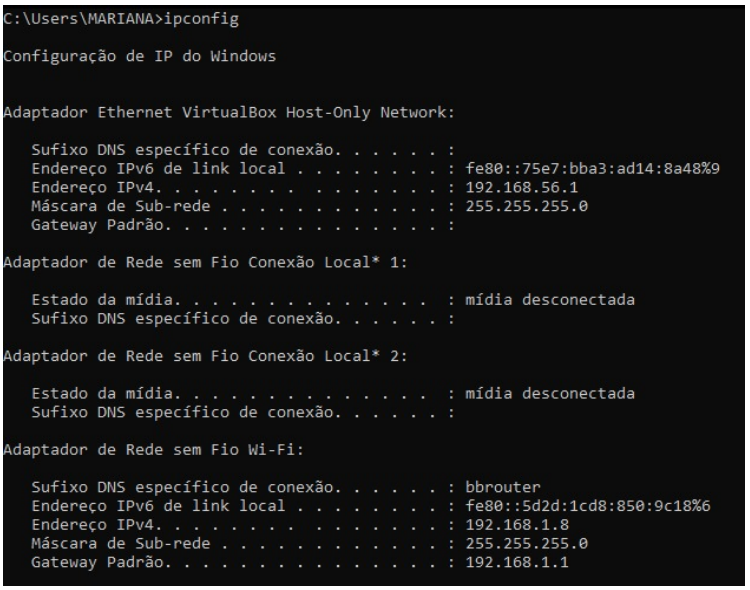


Imagem 23. Escolhendo interface wifi.

ipconfig - endereço padrão

O comando utilizado no windows "ipconfig" indica o endereço padrão gateway que está configurado na captura do wifi, sendo assim a imagem abaixo mostra o comando sendo executado no terminal do windows e como saída temos o endereço de IP do wifi que está em tráfego no momento e o gateway padrão que é "192.168.1.1" que será utilizado para o ping.



```
C:\Users\MARIANA>ipconfig

Configuração de IP do Windows


Adaptador Ethernet VirtualBox Host-Only Network:

   Sufixo DNS específico de conexão. . . . . :
   Endereço IPv6 de link local . . . . . . . : fe80::75e7:bba3:ad14:8a48%9
   Endereço IPv4. . . . . . . . . . . . . . : 192.168.56.1
   Máscara de Sub-rede . . . . . . . . . . . : 255.255.255.0
   Gateway Padrão. . . . . . . . . . . . . . :

Adaptador de Rede sem Fio Conexão Local* 1:

   Estado da mídia. . . . . . . . . . . . . . : mídia desconectada
   Sufixo DNS específico de conexão. . . . . . :

Adaptador de Rede sem Fio Conexão Local* 2:

   Estado da mídia. . . . . . . . . . . . . . : mídia desconectada
   Sufixo DNS específico de conexão. . . . . . :

Adaptador de Rede sem Fio Wi-Fi:

   Sufixo DNS específico de conexão. . . . . . : bbrouter
   Endereço IPv6 de link local . . . . . . . : fe80::5d2d:1cd8:850:9c18%6
   Endereço IPv4. . . . . . . . . . . . . . : 192.168.1.8
   Máscara de Sub-rede . . . . . . . . . . . : 255.255.255.0
   Gateway Padrão. . . . . . . . . . . . . . : 192.168.1.1
```

Imagem 24. ipconfig

Ip Ping

Nesse tópico foi utilizado o comando ping no prompt de comando do Windows para testar a conectividade entre equipamentos da rede, onde o ping foi seguido do IP gateway encontrado anteriormente, verificando se o gateway está ativo, como pode ser observado na imagem abaixo, além disso ele apresenta a latência e as estatísticas de pacotes que foram enviados, recebidos e perdidos.

```
C:\Users\MARIANA>Ping 192.168.1.1

Disparando 192.168.1.1 com 32 bytes de dados:
Resposta de 192.168.1.1: bytes=32 tempo=4ms TTL=64
Resposta de 192.168.1.1: bytes=32 tempo=4ms TTL=64
Resposta de 192.168.1.1: bytes=32 tempo=4ms TTL=64
Resposta de 192.168.1.1: bytes=32 tempo=4ms TTL=64

Estatísticas do Ping para 192.168.1.1:
    Pacotes: Enviados = 4, Recebidos = 4, Perdidos = 0 (0% de
             perda),
Aproximar um número redondo de vezes em milissegundos:
    Mínimo = 4ms, Máximo = 4ms, Média = 4ms
```

Imagem 25. Ping comIP.

Executando o ping é executada também simultaneamente a captura com o wireshark, assim quando o ping é finalizado a captura que estava sendo realizada também foi parada.

Tráfego com o Wifi

Na imagem abaixo está sendo mostrado todo o tráfego que foi realizado na captura do wifi, onde temos o painel de listas de pacote, este painel exibe os pacotes capturados. Cada linha representa um pacote individual que você pode clicar e analisar em detalhes usando os outros dois painéis que serão exemplificados nos tópicos abaixo.

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.8	31.13.74.52	TLSv1.2	85	Application Data
2	0.014993	31.13.74.52	192.168.1.8	TCP	54	443 → 59776 [ACK] Seq=1 Ack=32 Win=404 Len=0
3	0.124504	192.168.1.8	142.251.128.46	UDP	76	51271 → 443 Len=34
4	0.143952	31.13.74.52	192.168.1.8	TLSv1.2	92	Application Data
5	0.159774	142.251.128.46	192.168.1.8	UDP	68	443 → 51271 Len=26
6	0.186890	192.168.1.8	31.13.74.52	TCP	54	59776 → 443 [ACK] Seq=32 Ack=39 Win=514 Len=0
7	0.477214	20.80.218.34	192.168.1.8	TLSv1.2	584	Application Data
8	0.482473	192.168.1.8	20.80.218.34	TCP	1494	49821 → 443 [ACK] Seq=1 Ack=531 Win=517 Len=1440 [TCP segment of a reassembled PDU]
9	0.482473	192.168.1.8	20.80.218.34	TLSv1.2	441	Application Data
10	0.605441	20.80.218.34	192.168.1.8	TCP	54	443 → 49821 [ACK] Seq=531 Ack=1828 Win=716 Len=0
11	0.868821	52.86.88.22	192.168.1.8	TCP	54	443 → 50944 [ACK] Seq=1 Ack=1 Win=15 Len=0
12	0.868850	192.168.1.8	52.86.88.22	TCP	54	[TCP ACKed unseen segment] 50944 → 443 [ACK] Seq=1 Ack=2 Win=516 Len=0
13	5.774534	192.168.1.8	192.168.1.1	ICMP	74	Echo (ping) request id=0x0001, seq=185/47360, ttl=128 (reply in 14)
14	5.778814	192.168.1.1	192.168.1.8	ICMP	74	Echo (ping) reply id=0x0001, seq=185/47360, ttl=64 (request in 13)
15	6.786248	192.168.1.8	192.168.1.1	ICMP	74	Echo (ping) request id=0x0001, seq=186/47616, ttl=128 (reply in 16)
16	6.790958	192.168.1.1	192.168.1.8	ICMP	74	Echo (ping) reply id=0x0001, seq=186/47616, ttl=64 (request in 15)
17	7.801480	192.168.1.8	192.168.1.1	ICMP	74	Echo (ping) request id=0x0001, seq=187/47872, ttl=128 (reply in 18)
18	7.806212	192.168.1.1	192.168.1.8	ICMP	74	Echo (ping) reply id=0x0001, seq=187/47872, ttl=64 (request in 17)
19	8.816963	192.168.1.8	192.168.1.1	ICMP	74	Echo (ping) request id=0x0001, seq=188/48128, ttl=128 (reply in 20)
20	8.821798	192.168.1.1	192.168.1.8	ICMP	74	Echo (ping) reply id=0x0001, seq=188/48128, ttl=64 (request in 19)
21	10.191636	192.168.1.8	13.68.233.9	TCP	54	59777 → 443 [FIN, ACK] Seq=1 Ack=1 Win=516 Len=0
22	10.319530	13.68.233.9	192.168.1.8	TCP	54	443 → 59777 [FIN, ACK] Seq=1 Ack=2 Win=2052 Len=0
23	10.319567	192.168.1.8	13.68.233.9	TCP	54	59777 → 443 [ACK] Seq=2 Ack=2 Win=516 Len=0

```
> Frame 1: 85 bytes on wire (680 bits), 85 bytes captured (680 bits) on interface \Device\NPF_{603F82BF-420F-4AF8-B0F0-3511F2F73057}, id 0
> Ethernet II, Src: LiteonTe_a6:e8:29 (00:f4:8d:a6:e8:29), Dst: TendaTec_04:5a:98 (c8:3a:35:04:5a:98)
> Internet Protocol Version 4, Src: 192.168.1.8, Dst: 31.13.74.52
> Transmission Control Protocol, Src Port: 59776, Dst Port: 443, Seq: 1, Ack: 1, Len: 31
> Transport Layer Security
```

Imagem 26.

Filtro ICMP

Ao aplicar o filtro ICMP foram exibidos os seguintes pacotes que estão na figura

icmp

No.	Source	Destination	Protocol	Length	Info
13 5.774534	192.168.1.8	192.168.1.1	ICMP	74	Echo (ping) request id=0x0001, seq=185/47360, ttl=128 (reply in 14)
14 5.778814	192.168.1.1	192.168.1.8	ICMP	74	Echo (ping) reply id=0x0001, seq=185/47360, ttl=64 (request in 13)
15 6.786248	192.168.1.8	192.168.1.1	ICMP	74	Echo (ping) request id=0x0001, seq=186/47616, ttl=128 (reply in 16)
16 6.790958	192.168.1.1	192.168.1.8	ICMP	74	Echo (ping) reply id=0x0001, seq=186/47616, ttl=64 (request in 15)
17 7.801480	192.168.1.8	192.168.1.1	ICMP	74	Echo (ping) request id=0x0001, seq=187/47872, ttl=128 (reply in 18)
18 7.806212	192.168.1.1	192.168.1.8	ICMP	74	Echo (ping) reply id=0x0001, seq=187/47872, ttl=64 (request in 17)
19 8.816963	192.168.1.8	192.168.1.1	ICMP	74	Echo (ping) request id=0x0001, seq=188/48128, ttl=128 (reply in 20)
20 8.821798	192.168.1.1	192.168.1.8	ICMP	74	Echo (ping) reply id=0x0001, seq=188/48128, ttl=64 (request in 19)

Imagem 27.

Pacote 1

Analisando os pacotes, para esse tópico foi escolhido um pacote onde será analisado mais detalhadamente, ele irá exibir informações como o endereço do IP, as portas e outras informações que tiverem contidas no pacote que foi selecionado, para selecionar um pacote utilizando o wireshark, basta ir no painel de lista de pacotes e clicar em qual pacote deseja, onde cada linha do painel irá representar um pacote.

icmp

No.	Time	Source	Destination	Protocol	Length	Info
14	5.778814	192.168.1.1	192.168.1.8	ICMP	74	Echo (ping) reply id=0
16	6.790958	192.168.1.1	192.168.1.8	ICMP	74	Echo (ping) reply id=0
18	7.806212	192.168.1.1	192.168.1.8	ICMP	74	Echo (ping) reply id=0
20	8.821798	192.168.1.1	192.168.1.8	ICMP	74	Echo (ping) reply id=0
13	5.774534	192.168.1.8	192.168.1.1	ICMP	74	Echo (ping) request id=0
15	6.786248	192.168.1.8	192.168.1.1	ICMP	74	Echo (ping) request id=0
17	7.801480	192.168.1.8	192.168.1.1	ICMP	74	Echo (ping) request id=0

Wireshark · Packet 13 · Wi-Fi

```
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x4ca2 [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence Number (BE): 185 (0x00b9)
    Sequence Number (LE): 47360 (0xb900)
    [Response frame: 14]
    Data (32 bytes)
        Data: 6162636465666768696a6b6c6d6e6f70717273747576776162636465666768 69
        [Length: 32]

0000  c8 3a 35 04 5a 98 00 f4  8d a6 e8 29 08 00 45 00   ·:5·Z··· ···)··E·
0010  00 3c 66 d0 00 00 80 01  50 97 c0 a8 01 08 c0 a8   ·<f····· P·····
0020  01 01 08 00 4c a2 00 01  00 b9 61 62 63 64 65 66   ····L··· ··abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                     wabcdefg hi
```

Imagem 28.

Quadro Frame

Para o quadro frame do pacote 1, como pode ser observado na imagem o quadro frame possui todas os bytes do painel de bytes do pacote, pois esse tipo de pacote contém uma grande quantidade de informações para serem transmitidas.

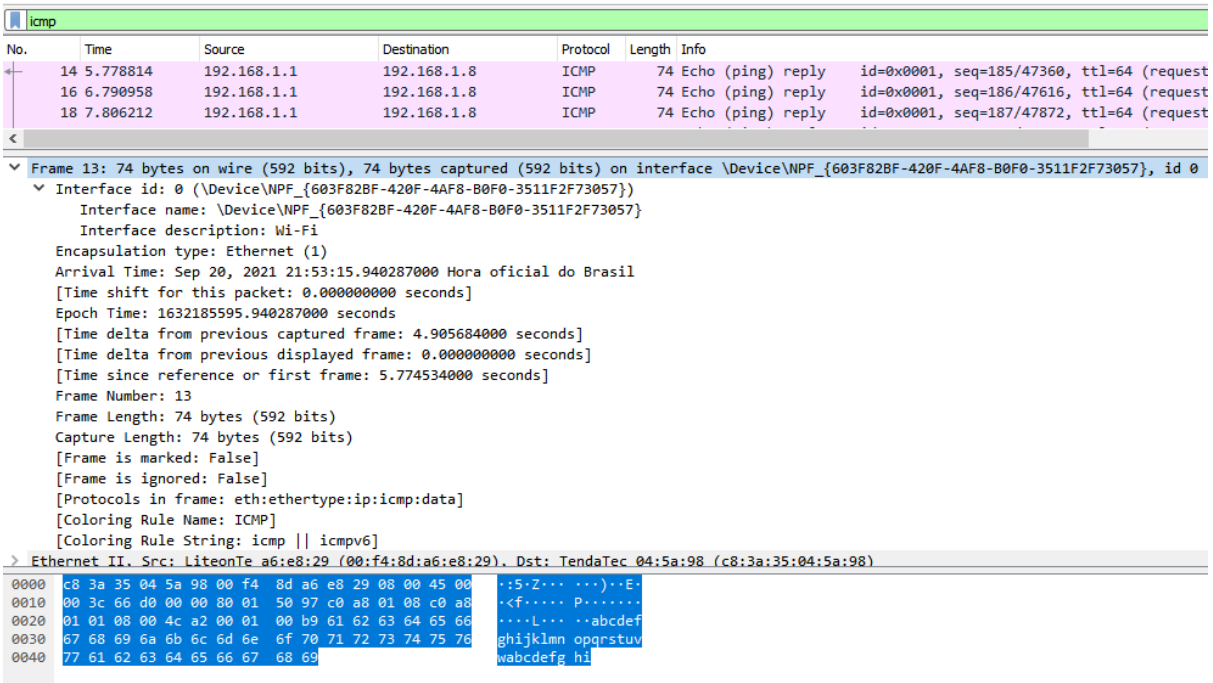


Imagem 29.

Destination, source e type - Pacote 1

Para os campos de destination, source e type do pacote 1 utilizando o wifi temos o destination como campo que contém o IP do destino desse pacote que está sendo analisado, já o campo source tem o endereço de IP da origem do pacote 1 e o campo type terá o comprimento. Sendo assim abaixo são apresentadas as respectivas imagens para cada um dos campos que foram citados

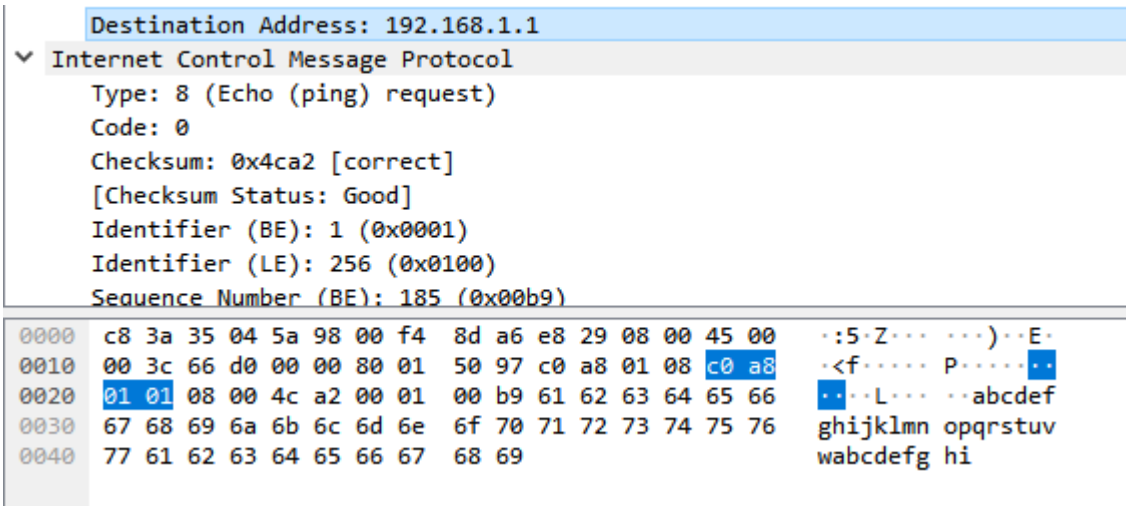


Imagem 30. Destination pacote 1

```
[Header checksum status: Unverified]
Source Address: 192.168.1.8
Destination Address: 192.168.1.1
Internet Control Message Protocol

0000  c8 3a 35 04 5a 98 00 f4  8d a6 e8 29 08 00 45 00   ·:5·Z··· ···)··E·
0010  00 3c 66 d0 00 00 80 01  50 97 c0 a8 01 08 c0 a8   ·<f····· P·······
0020  01 01 08 00 4c a2 00 01  00 b9 61 62 63 64 65 66   ····L··· ··abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                     wabcdefg hi
```

Imagem 31. Source pacote 1.

```
Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0x4ca2 [correct]
[Checksum Status: Good]
Identifier (BE): 1 (0x0001)
Identifier (LE): 256 (0x0100)

0000  c8 3a 35 04 5a 98 00 f4  8d a6 e8 29 08 00 45 00   ·:5·Z··· ···)··E·
0010  00 3c 66 d0 00 00 80 01  50 97 c0 a8 01 08 c0 a8   ·<f····· P·····
0020  01 01 08 00 4c a2 00 01  00 b9 61 62 63 64 65 66   ···L··· ··abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                     wabcdefg hi
```

Imagem 32. Type pacote 1

Análise Pacote 2

Pacote 2 selecionado:

```
icmp
No.  Time        Source        Destination   Protocol  Length  Info
14  5.778814    192.168.1.1   192.168.1.8   ICMP      74  Echo (ping) reply   id=0x0001, seq=185/47360, ttl=64 (request in 13)
16  6.790958    192.168.1.1   192.168.1.8   ICMP      74  Echo (ping) reply   id=0x0001, seq=186/47616, ttl=64 (request in 15)
18  7.806212    192.168.1.1   192.168.1.8   ICMP      74  Echo (ping) reply   id=0x0001, seq=187/47872, ttl=64 (request in 17)
20  8.821798    192.168.1.1   192.168.1.8   ICMP      74  Echo (ping) reply   id=0x0001, seq=188/48128, ttl=64 (request in 19)
13  5.774534    192.168.1.8   192.168.1.1   ICMP      74  Echo (ping) request id=0x0001, seq=185/47360, ttl=128 (reply in 14)
15  6.786248    192.168.1.8   192.168.1.1   ICMP      74  Echo (ping) request id=0x0001, seq=186/47616, ttl=128 (reply in 16)
17  7.801480    192.168.1.8   192.168.1.1   ICMP      74  Echo (ping) request id=0x0001, seq=187/47872, ttl=128 (reply in 18)

Wireshark · Packet 18 · Wi-Fi

Frame 18: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{603F82BF-420F-4AF8-B0F0-3511F2F73057}, id 0
  Interface id: 0 (\Device\NPF_{603F82BF-420F-4AF8-B0F0-3511F2F73057})
     Interface name: \Device\NPF_{603F82BF-420F-4AF8-B0F0-3511F2F73057}
     Interface description: Wi-Fi
  Encapsulation type: Ethernet (1)
  Arrival Time: Sep 20, 2021 21:53:17.971965000 Hora oficial do Brasil
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1632185597.971965000 seconds
  [Time delta from previous captured frame: 0.004732000 seconds]
  [Time delta from previous displayed frame: 0.004732000 seconds]
  [Time since reference or first frame: 7.806212000 seconds]
  Frame Number: 18
  Frame Length: 74 bytes (592 bits)

0000  00 f4 8d a6 e8 29 c8 3a  35 04 5a 98 08 00 45 00   ·····)·: 5·Z···E·
0010  00 3c b3 55 00 00 40 01  44 12 c0 a8 01 01 c0 a8   ·<·U··@· D·····
0020  01 08 00 00 54 a0 00 01  00 bb 61 62 63 64 65 66   ····T··· ··abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                     wabcdefg hi
```

Imagem 33.

Frame, Destination, source e type - Pacote 2

Frame:

```
Frame 18: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{603F82BF-420F-4AF8-B0F0-3511F2F73057}, id 0
  Interface id: 0 (\Device\NPF_{603F82BF-420F-4AF8-B0F0-3511F2F73057})
      Interface name: \Device\NPF_{603F82BF-420F-4AF8-B0F0-3511F2F73057}
      Interface description: Wi-Fi
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 20, 2021 21:53:17.971965000 Hora oficial do Brasil
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1632185597.971965000 seconds
    [Time delta from previous captured frame: 0.004732000 seconds]
    [Time delta from previous displayed frame: 0.004732000 seconds]
    [Time since reference or first frame: 7.806212000 seconds]
    Frame Number: 18
    Frame Length: 74 bytes (592 bits)
    Capture Length: 74 bytes (592 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:icmp:data]
    [Coloring Rule Name: ICMP]
    [Coloring Rule String: icmp || icmpv6]

0000  00 f4 8d a6 e8 29 c8 3a  35 04 5a 98 08 00 45 00   ·····)·: 5·Z···E·
0010  00 3c b3 55 00 00 40 01  44 12 c0 a8 01 01 c0 a8   ·<·U··@· D·····
0020  01 08 00 00 54 a0 00 01  00 bb 61 62 63 64 65 66   ····T··· ··abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                     wabcdefg hi
```

Imagem 34.

Destination:

```
        .0.. .... = Don't fragment: Not set
        ..0. .... = More fragments: Not set
    Fragment Offset: 0
    Time to Live: 64
    Protocol: ICMP (1)
    Header Checksum: 0x4412 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.1.1
    Destination Address: 192.168.1.8

0000  00 f4 8d a6 e8 29 c8 3a  35 04 5a 98 08 00 45 00   ·····)·: 5·Z···E·
0010  00 3c b3 55 00 00 40 01  44 12 c0 a8 01 01 c0 a8   ·<·U··@· D·······
0020  01 08 00 00 54 a0 00 01  00 bb 61 62 63 64 65 66   ····T··· ··abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                     wabcdefg hi
```

Imagem 35.

Source:

```
Header Checksum: 0x4412 [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.1.1
Destination Address: 192.168.1.8

0000  00 f4 8d a6 e8 29 c8 3a  35 04 5a 98 08 00 45 00   ·····)·: 5·Z···E·
0010  00 3c b3 55 00 00 40 01  44 12 c0 a8 01 01 c0 a8   ·<·U··@· D·····
0020  01 08 00 00 54 a0 00 01  00 bb 61 62 63 64 65 66   ····T··· ··abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                     wabcdefg hi
```

Type:

```
Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0

0000  00 f4 8d a6 e8 29 c8 3a  35 04 5a 98 08 00 45 00   ·····)·: 5·Z···E·
0010  00 3c b3 55 00 00 40 01  44 12 c0 a8 01 01 c0 a8   ·<·U··@· D·····
0020  01 08 00 00 54 a0 00 01  00 bb 61 62 63 64 65 66   ····T··· ··abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                     wabcdefg hi
```

Imagem 36.

Endereço Físico wifi

```
C:\Users\MARIANA>getmac

Endereço físico     Nome de transporte
=================== ==========================================================
00-F4-8D-A6-E8-29   \Device\Tcpip_{603F82BF-420F-4AF8-B0F0-3511F2F73057}
0A-00-27-00-00-09   \Device\Tcpip_{8871CB49-FC00-4067-A998-11E3C973DED0}
```

Imagem 37.

3.4 Execução do Wireshark em modo monitor

Foi realizada a execução do Wireshark em modo monitor como é mostrado nas figuras abaixo, foi deixado o programa executando por um bom tempo e mesmo assim não foi encontrado nenhum frame em que o índice MCS esteja presente, então resolvi usar o frame que condiz com a minha própria rede wifi e como pode ser observado a variante dela é a 802.11a, aliás foram encontradas apenas variantes 802.11a e 802.11b, a frequência do frame é de 5745 MHz, e a taxa de transmissão de dados é de 6Mb/s.

Como não foi encontrado nenhum índice MCS e nenhuma variante condiz com a tabela de especificação, não é possível concluir o restante do que foi pedido.

```
symphony@symphony-pc:~$ sudo su
root@symphony-pc:/home/symphony# airmon-ng

PHY     Interface       Driver          Chipset

phy0    wlp1s0          iwlwifi         Intel Corporation Dual Band Wireless-AC
3165 Plus Bluetooth (rev 99)
```

Imagem 38. Nome da Interface usada

```
▾ Frame 796: 365 bytes on wire (2920 bits), 365 bytes captured (2920 bits) on interface wlp1s0mon, id 0
  ▾ Interface id: 0 (wlp1s0mon)
      Interface name: wlp1s0mon
    Encapsulation type: IEEE 802.11 plus radiotap radio header (23)
    Arrival Time: Sep 20, 2021 22:33:16.149236194 -03
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1632187996.149236194 seconds
    [Time delta from previous captured frame: 0.102233486 seconds]
    [Time delta from previous displayed frame: 0.102233486 seconds]
    [Time since reference or first frame: 658.933283086 seconds]
    Frame Number: 796
    Frame Length: 365 bytes (2920 bits)
    Capture Length: 365 bytes (2920 bits)
    [Frame MD5 Hash: 389209151f1fb2a0dbbf8d0654ef742b]
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: radiotap:wlan_radio:wlan]
▸ Radiotap Header v0, Length 56
▾ 802.11 radio information
    PHY type: 802.11a (OFDM) (5)

0000  00 00 38 00 2f 40 40 a0  20 08 00 a0 20 08 00 00   ··8 /@@· ····
0010  7c c1 c7 c6 00 00 00 00  10 0c 71 16 40 01 c3 00   |·******
0020  00 00 00 00 00 00 00 00  67 c1 c7 c6 00 00 00 00   ********
0030  16 00 11 03 c3 00 c1 01  80 00 00 00 ff ff ff ff   ********
0040  ff ff 90 55 de a8 14 2c  90 55 de a8 14 2c b0 de   ···U···,
0050  44 30 73 61 53 02 00 00  64 00 11 01 00 09 4c 49   D0saS···
0060  4c 49 2d 50 4c 55 53 01  08 8c 12 98 24 b0 48 60   LI-PLUS·
0070  6c 05 04 00 01 00 00 07  4e 23 61 20 24 01 1e 28   l······ N#a
0080  01 1e 2c 01 1e 30 01 1e  34 01 1e 38 01 1e 3c 01   ··,··0··
0090  1e 40 01 1e 64 01 1e 68  01 1e 6c 01 1e 70 01 1e   ·@··d··h
00a0  74 01 1e 78 01 1e 7c 01  1e 80 01 1e 84 01 1e 88   t··x··|·
00b0  01 1e 8c 01 1e 90 01 1e  95 01 1e 99 01 1e 9d 01   ********
00c0  1e a1 01 1e a5 01 1e 20  01 00 23 02 14 00 30 14   *******
00d0  01 00 00 0f ac 04 01 00  00 0f ac 04 01 00 00 0f   ********
00e0  ac 02 0c 00 0b 05 00 00  04 00 00 2d 1a ef 08 17   ******** ···-
00f0  ff ff 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ********
0100  00 00 00 00 00 00 00 3d  16 95 0d 00 00 00 00 00   *******=
0110  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 7f   ********
0120  08 04 00 08 00 00 00 00  40 bf 0c b2 00 80 03 fa   ********
0130  ff 00 00 fa ff 00 00 c0  05 01 9b 00 00 00 c3 04   ********
0140  02 02 02 02 dd 09 00 10  18 02 00 00 1c 00 00 dd   ********
0150  18 00 50 f2 02 01 01 84  00 03 a4 00 00 27 a4 00   ··P·····
0160  00 42 43 5e 00 62 32 2f  00 e3 b4 0a b5            ·BC^·b2/ ·····
```

```
▾ 802.11 radio information
    PHY type: 802.11a (OFDM) (5)
    Turbo type: Non-turbo (0)
    Data rate: 6,0 Mb/s
    Channel: 149
    Frequency: 5745MHz
    Signal strength (dBm): -63 dBm
    TSF timestamp: 3334979964
  ▸ [Duration: 436µs]

  ▸ Capabilities Information: 0x0111
▾ Tagged parameters (269 bytes)
  ▸ Tag: SSID parameter set: LILI-PLUS
  ▸ Tag: Supported Rates 6(B), 9, 12(B), 18, 24(B), 36, 48, 54, [Mbit/sec]
  ▸ Tag: Traffic Indication Map (TIM): DTIM 1 of 1 bitmap
  ▸ Tag: Country Information: Country Code #a, Environment Any
  ▸ Tag: Power Constraint: 0
  ▸ Tag: TPC Report Transmit Power: 20, Link Margin: 0
  ▸ Tag: RSN Information
  ▸ Tag: QBSS Load Element 802.11e CCA Version
  ▸ Tag: HT Capabilities (802.11n D1.10)
  ▸ Tag: HT Information (802.11n D1.10)
  ▸ Tag: Extended Capabilities (8 octets)
  ▸ Tag: VHT Capabilities
  ▸ Tag: VHT Operation
  ▸ Tag: VHT Tx Power Envelope
  ▸ Tag: Vendor Specific: Broadcom
  ▸ Tag: Vendor Specific: Microsoft Corp.: WMM/WME: Parameter Element
```

Imagens 39, 40 e 41 . Execução no Wireshark pelo Modo Monitor

Conclusão

Neste trabalho prático foram discutidos dois temas, o primeiro sobre a tecnologia Ethernet, e o segundo Wifi, estes dois tópicos foram analisados com a ferramenta wireshark, cujo objetivo principal foi verificar alguns campos que são de suma importância para o seu funcionamento. Ademais, o trabalho teve em um contexto geral não só a análise e verificação de alguns quadros da Ethernet, como também quadros da tecnologia wife. Que foram identificados e analisados com a ferramenta wireshark.

Na análise da tecnologia Ethernet, foram pesquisados como são os formatos de seus quadros, como é feito o envio de quadros, e quantos bytes cada quadro tem. Três desses quadros apareceram de forma corriqueira neste trabalho, foram eles: Campo endereço MAC origem, que é o quadro "destination" que é o quadro apresentado no wireshark, quadro este que pode ser analisado escolhendo-se um pacote na ferramenta. Este quadro é o endereço IP do destino do pacote. Foi analisado também o campo Source que é endereço IP da origem de um pacote, este campo de 6 bytes identifica a placa de rede ou a interface de origem do quadro. Já o campo type analisado é o campo comprimento que define o comprimento exato do campo de dados do quadro. Diante disso, também foram analisados os pacotes e os endereços de pacotes enviados e recebidos.

Outrossim, analisando o tráfego wifi, temos que o sistema operacional utilizado foi o Windows, e analisando o destination, frame, source e type, pode-se perceber que o tráfego não altera muito, ele se mantém em alguns casos mesmo quando os pacotes são alterados, então na análise de pacote temos que os endereços IP de source e destination, ficam variando entre si.

O trabalho foi de grande importância para aplicarmos o conteúdo visto na matéria de redes de computadores e entender melhor o funcionamento da ferramenta Wireshark, onde encontramos diversas funcionalidades.

Referências:

Download wireshark:https://www.edivaldobrito.com.br/wireshark-2-0-no-ubuntu-15-10/

Ethernet:https://www.networkworld.com/article/3225865/ethernet-frames-and-packets-whats-the-difference.html

deptal(http://deptal.estgp.pt:9090/cisco/ccna1/course/module5/5.1.2.3/5.1.2.3.html

Livro:http://187.52.54.51/nataniel/REDES_DE_COMPUTADORES/Laboratorio_Redes1/Aula13_Wireshark/Apoio/wireshark_practical_packet_analysis.pdf-

Quadros Wifi:http://deptal.estgp.pt:9090/cisco/ccna1/course/module4/4.4.4.8/4.4.4.8.html

Aircrack: http://www.insecure.net.br/aircrack-ng.txt