

# Trabalho III

Trabalho com base nos comandos básicos e avançados do Linux na disciplina de Redes de Computadores II do Curso de Sistemas de Informação da UFVJM.

**Docente:** Alessandro Vivas Andrade

**Discentes:**

Alisson Alessandro Nunes Ferreira

Wender de Assis dos Santos

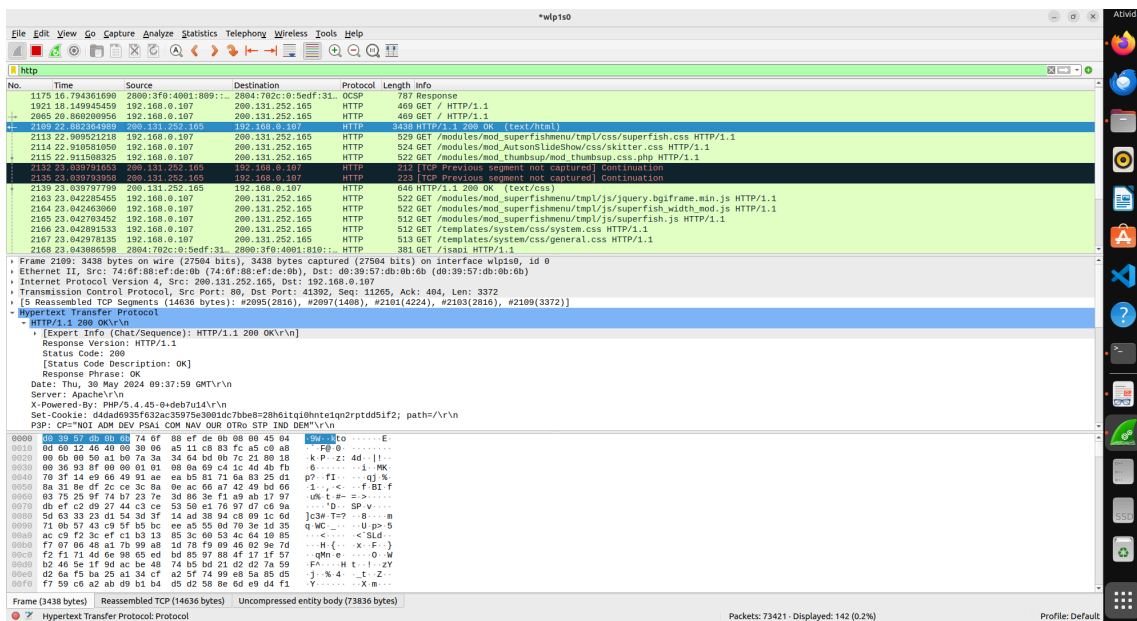
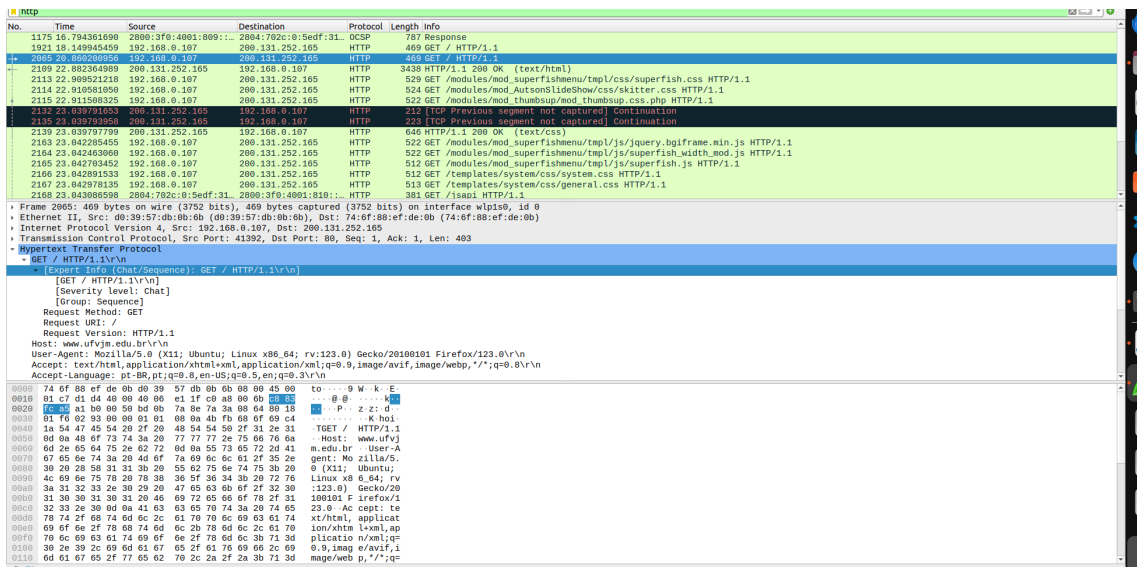
2024

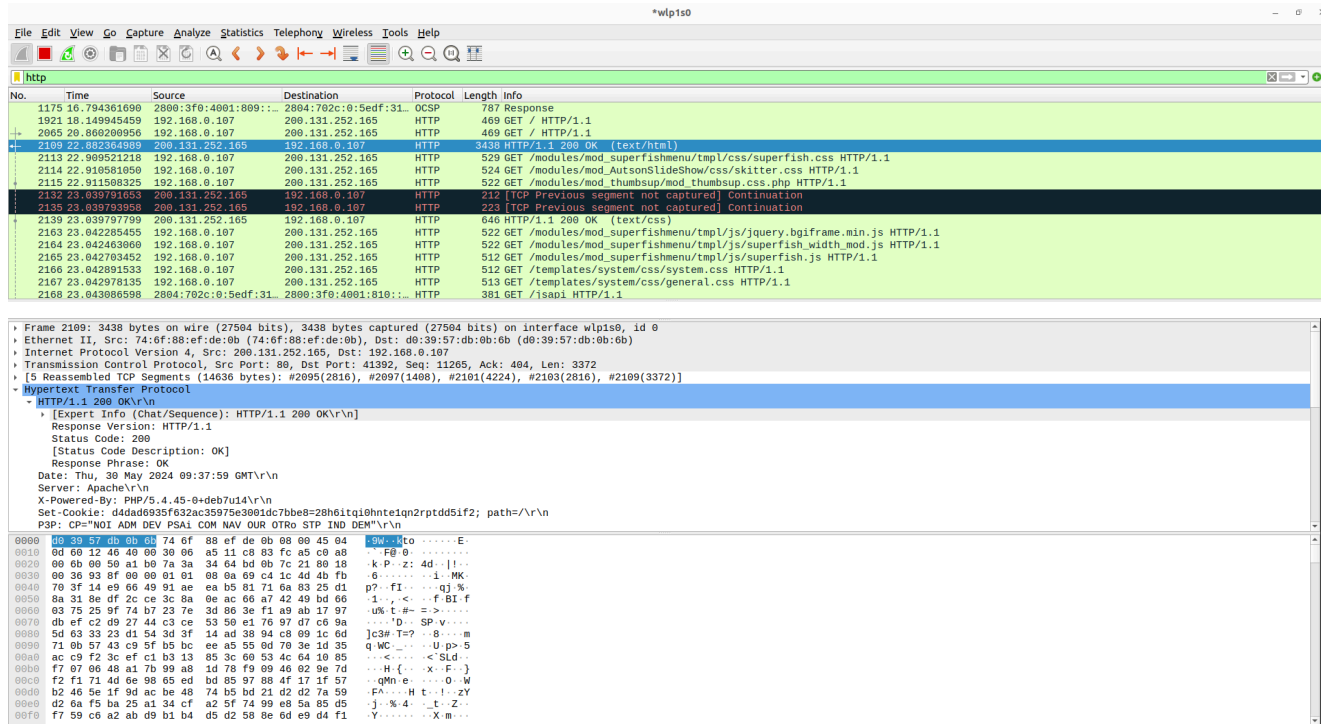
## Trabalho 3 de Redes de Computadores 2

Prof. Alessandro Vivas Andrade Utilizando o Wireshark resolva os seguintes exercícios.

### 1. Utilizando o software Wireshark capture o tráfego HTTP gerado pelo seu navegador

- 1. Abra o Wireshark
- 2. Acesse um site em seu navegador (ex: [www.google.com](http://www.google.com)).
- 3. Pare a captura e filtre os pacotes por HTTP.
- 4. Analise os cabeçalhos HTTP para identificar o método de requisição (GET, POST, etc.), o URL acessado, os cabeçalhos de resposta e o conteúdo da página.





## Capturar o tráfego HTTP:

### 1. Abrir o Wireshark:

- Abri o Wireshark e selecionei a interface de rede conectada à internet.
- Iniciei a captura clicando no ícone de "Start Capturing Packets".

### 2. Acessar um site:

- No navegador, acessei o site <https://portal.ufvjm.edu.br/>.

### 3. Parar a captura e filtrar os pacotes:

- Voltei ao Wireshark e parei a captura clicando no ícone de "Stop Capturing Packets".
- Na barra de filtros, digitei **http** e pressionei Enter para filtrar os pacotes HTTP.

### 4. Analisar os cabeçalhos HTTP:

- Cliquei em um pacote HTTP na lista.
- Na seção "Hypertext Transfer Protocol" na parte inferior do Wireshark, expandi para visualizar os detalhes.

## Resultados da Análise:

- **Método de Requisição:** O método utilizado foi **GET**.
- **URL Acessado:** O URL acessado foi <https://portal.ufvjm.edu.br/>.
- **Cabeçalhos de Requisição:** Incluíam **Host**, **User-Agent**, **Accept**, **Accept-Language** e **Connection**.

- **Cabeçalhos de Resposta:** Incluía **Server**, **Content-Type**, **Content-Length**, **Date** e **Set-Cookie**.
- **Conteúdo da Página:** O conteúdo HTML e outros recursos da página estavam presentes no pacote de resposta.

### Conclusão:

Capturando e analisando pacotes HTTP com o Wireshark, observei a comunicação entre o navegador e o servidor ao acessar a página <https://portal.ufvjm.edu.br/>, identificando métodos de requisição, cabeçalhos e conteúdo da página, o que ajuda a entender o funcionamento do protocolo HTTP.

## 2. Utilizando o software Wireshark capture o tráfego TCP gerado no estabelecimento de conexão

- 1. Abra o Wireshark
- 2. Acesse o servidor Web de computador local ou remoto utilizando o telnet na porta 80
- 3. Colete os dados de conexão e verifique os cabeçalhos do TCP observando o processo de estabelecimento e encerramento de conexão.

```
alisson@alisson-VirtualBox:~$ telnet 192.168.1.23 80
Trying 192.168.1.23...
telnet: Unable to connect to remote host: Connection refused
alisson@alisson-VirtualBox:~$ ** (wireshark:36148) 23:50:19.944987 [Capture MESSAGE] -- Capture Stop ...
** (wireshark:36148) 23:50:20.018500 [Capture MESSAGE] -- Capture stopped.
** (wireshark:36148) 23:50:20.018657 [Capture WARNING] ./ui/capture.c:722 -- capture_input_closed():
** (wireshark:36148) 23:54:36.939353 [Capture MESSAGE] -- Capture Start ...
** (wireshark:36148) 23:54:37.208630 [Capture MESSAGE] -- Capture started
** (wireshark:36148) 23:54:37.208698 [Capture MESSAGE] -- File: "/tmp/wireshark_enp0s3EYTH02.pcapng"
** (wireshark:36148) 23:55:14.926223 [Capture MESSAGE] -- Capture Stop ...
** (wireshark:36148) 23:55:14.972215 [Capture MESSAGE] -- Capture stopped.
** (wireshark:36148) 23:55:14.972250 [Capture WARNING] ./ui/capture.c:722 -- capture_input_closed():
```

redes 2 quest2 porta 80.pcapng						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
Interface Channel 802.11 Preferences						
No.	Time	Source	Destination	Protocol	Length	Info
198	31.689168338	192.168.1.23	91.189.91.97	TCP	74	41754 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_P...
199	31.818908459	91.189.91.97	192.168.1.23	TCP	74	80 → 41754 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=...
200	31.818952937	192.168.1.23	91.189.91.97	TCP	66	41754 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=327...
201	31.819111005	192.168.1.23	91.189.91.97	HTTP	153	GET / HTTP/1.1
202	31.946244854	91.189.91.97	192.168.1.23	TCP	66	[TCP Previous segment not captured] 80 → 41754 [FIN, A...
203	31.946298082	192.168.1.23	91.189.91.97	TCP	78	[TCP Dup ACK 200#1] 41754 → 80 [ACK] Seq=88 Ack=1 Win=...
204	31.946245294	91.189.91.97	192.168.1.23	TCP	251	[TCP Out-Of-Order] 80 → 41754 [PSH, ACK] Seq=1 Ack=88 ...
205	31.946343921	192.168.1.23	91.189.91.97	TCP	66	41754 → 80 [ACK] Seq=88 Ack=187 Win=64128 Len=0 TSval=...
206	31.946600954	192.168.1.23	91.189.91.97	TCP	66	41754 → 80 [FIN, ACK] Seq=88 Ack=187 Win=64128 Len=0 T...
207	32.353444315	192.168.1.23	91.189.91.97	TCP	66	[TCP Retransmission] 41754 → 80 [FIN, ACK] Seq=88 Ack=...
208	32.480681627	91.189.91.97	192.168.1.23	TCP	66	80 → 41754 [ACK] Seq=187 Ack=89 Win=65152 Len=0 TSval=...
211	32.877664221	2804:10dc:d11d:c700...	2620:2d:4000:1::23	TCP	94	49246 → 80 [SYN] Seq=0 Win=64800 Len=0 MSS=1440 SACK_P...
212	33.066491315	2620:2d:4000:1::23	2804:10dc:d11d:c700...	TCP	94	80 → 49246 [SYN, ACK] Seq=0 Ack=1 Win=64200 Len=0 MSS=...
213	33.066540250	2804:10dc:d11d:c700...	2620:2d:4000:1::23	TCP	86	49246 → 80 [ACK] Seq=1 Ack=1 Win=64896 Len=0 TSval=281...
214	33.066777693	2804:10dc:d11d:c700...	2620:2d:4000:1::23	HTTP	173	GET / HTTP/1.1
215	33.255835774	2620:2d:4000:1::23	2804:10dc:d11d:c700...	TCP	86	[TCP Previous segment not captured] 80 → 49246 [FIN, A...
216	33.255836096	2620:2d:4000:1::23	2804:10dc:d11d:c700...	TCP	275	[TCP Out-Of-Order] 80 → 49246 [PSH, ACK] Seq=1 Ack=88 ...
217	33.255873623	2804:10dc:d11d:c700...	2620:2d:4000:1::23	TCP	98	[TCP Dup ACK 213#1] 49246 → 80 [ACK] Seq=88 Ack=1 Win=...
218	33.255904414	2804:10dc:d11d:c700...	2620:2d:4000:1::23	TCP	86	49246 → 80 [ACK] Seq=88 Ack=191 Win=64768 Len=0 TSval=...
219	33.256157065	2804:10dc:d11d:c700...	2620:2d:4000:1::23	TCP	86	49246 → 80 [FIN, ACK] Seq=88 Ack=191 Win=64768 Len=0 T...
220	33.446316304	2620:2d:4000:1::23	2804:10dc:d11d:c700...	TCP	86	80 → 49246 [ACK] Seq=191 Ack=89 Win=64256 Len=0 TSval=...
221	33.446316790	2620:2d:4000:1::23	2804:10dc:d11d:c700...	TCP	86	[TCP Dup ACK 220#1] 80 → 49246 [ACK] Seq=191 Ack=89 Wi...
222	33.446391878	2804:10dc:d11d:c700...	2620:2d:4000:1::23	TCP	74	49246 → 80 [RST] Seq=89 Win=0 Len=0

Frame 222: 74 bytes on wire (592 bits), 74 bytes captured (592 b	0000	84 93 b2 a2 75 79 08 00	27 ab 3c 76 86 dd 60 00	uy..
Ethernet II, Src: PCSSystemtec_ab:3c:76 (08:00:27:ab:3c:76), Dst	0010	c4 78 00 14 06 40 28 04	10 dc d1 1d c7 00 46 97	x...@{(
Internet Protocol Version 6, Src: 2804:10dc:d11d:c700:4697:7d49:	0020	7d 49 31 7a 12 db 26 20	00 2d 40 00 00 01 00 00	}I1z.&
Transmission Control Protocol, Src Port: 49246, Dst Port: 80, Se	0030	00 00 00 00 00 23 c0 5e	00 50 49 54 87 e9 00 00	...#.#^
	0040	00 00 50 04 00 00 3f c0	00 00	P...?.

## Acesso ao servidor web

Utilizando o Wireshark para visualização do tráfego de pacotes, utilizamos o comando “telnet” para abrir a conexão como o servidor web do computador na porta 80

## Pacotes de Estabelecimento de Conexão

1. **Pacote SYN:**
  - Sinaliza o início da conexão.
2. **Pacote SYN-ACK:**
  - Resposta do servidor ao pacote SYN.
3. **Pacote ACK:**
  - Confirmação do cliente ao pacote SYN-ACK.

## Pacotes de Encerramento de Conexão

1. **Pacote FIN:**
  - Inicia o encerramento da conexão.
2. **Pacote ACK:**
  - Acknowledgement do pacote FIN.
3. **Pacote FIN:**
  - Enviado pelo outro lado para completar o encerramento.
4. **Pacote ACK:**

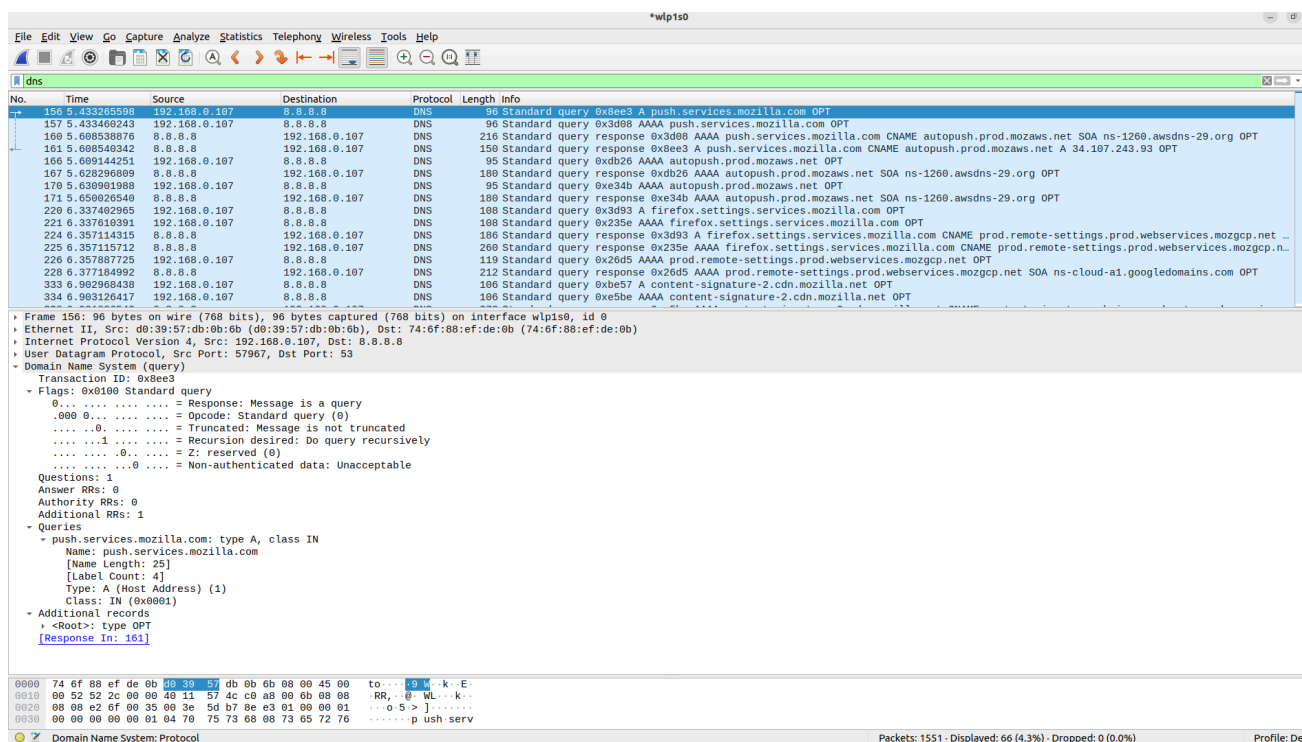
- Acknowledgement final do pacote FIN.

## Conclusão

Conseguimos desta forma ser capaz de capturar e analisar o tráfego TCP gerado durante o estabelecimento e encerramento de uma conexão utilizando o Wireshark.

### 3. Utilizando o software Wireshark capture o tráfego gerado através de uma consulta via DNS

- 1. Abra o Wireshark e inicie uma captura de pacotes.
- 2. Acesse um site em seu navegador (ex: [www.google.com](http://www.google.com)).
- 3. Pare a captura e filtre os pacotes por DNS. 4. Analise os pacotes DNS para identificar o processo de resolução de nomes, os servidores DNS utilizados e os registros DNS obtidos.



The image shows a Wireshark capture of DNS traffic. The top pane displays a list of captured packets, with the first packet (No. 156) selected. The middle pane shows the details of the selected packet, which is a Standard query (0x8ee3) for push.services.mozilla.com. The bottom pane shows the raw packet data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
156	5.43326598	192.168.0.107	8.8.8.8	DNS	96	Standard query 0x8ee3 A push.services.mozilla.com OPT
157	5.433466243	192.168.0.107	8.8.8.8	DNS	96	Standard query 0x3d08 AAAA push.services.mozilla.com OPT
160	5.608538876	8.8.8.8	192.168.0.107	DNS	216	Standard query response 0x3d08 AAAA push.services.mozilla.com CNAME autopush.prod.mozaws.net SOA ns-1260.awsdns-29.org OPT
161	5.608548342	8.8.8.8	192.168.0.107	DNS	150	Standard query response 0x8ee3 A push.services.mozilla.com CNAME autopush.prod.mozaws.net A 34.197.243.93 OPT
166	5.609144251	192.168.0.107	8.8.8.8	DNS	95	Standard query 0xdb26 AAAA autopush.prod.mozaws.net OPT
167	5.628296889	8.8.8.8	192.168.0.107	DNS	180	Standard query response 0xdb26 AAAA autopush.prod.mozaws.net SOA ns-1260.awsdns-29.org OPT
170	5.630901988	192.168.0.107	8.8.8.8	DNS	95	Standard query 0xe34b AAAA autopush.prod.mozaws.net OPT
171	5.650020540	8.8.8.8	192.168.0.107	DNS	180	Standard query response 0xe34b AAAA autopush.prod.mozaws.net SOA ns-1260.awsdns-29.org OPT
220	6.337402965	192.168.0.107	8.8.8.8	DNS	180	Standard query 0x3d93 A firefox.settings.services.mozilla.com OPT
221	6.337616391	192.168.0.107	8.8.8.8	DNS	180	Standard query 0x235e AAAA firefox.settings.services.mozilla.com OPT
224	6.357114315	8.8.8.8	192.168.0.107	DNS	186	Standard query response 0x3d93 A firefox.settings.services.mozilla.com CNAME prod.remote-settings.prod.webservices.mozgcp.net
225	6.357115712	8.8.8.8	192.168.0.107	DNS	260	Standard query response 0x235e AAAA firefox.settings.services.mozilla.com CNAME prod.remote-settings.prod.webservices.mozgcp.net
226	6.357887725	192.168.0.107	8.8.8.8	DNS	119	Standard query 0x20d5 AAAA prod.remote-settings.prod.webservices.mozgcp.net OPT
228	6.377184992	8.8.8.8	192.168.0.107	DNS	212	Standard query response 0x20d5 AAAA prod.remote-settings.prod.webservices.mozgcp.net SOA ns-cloud-a1.googledomains.com OPT
333	6.992968438	192.168.0.107	8.8.8.8	DNS	106	Standard query 0xe5e7 A content-signature-2.cdn.mozilla.net OPT
334	6.993126417	192.168.0.107	8.8.8.8	DNS	106	Standard query 0xe5be AAAA content-signature-2.cdn.mozilla.net OPT

Frame 156: 96 bytes on wire (768 bits), 96 bytes captured (768 bits) on interface wlp1s0, id 0  
 Ethernet II, Src: d0:39:57:db:0b:0b (d0:39:57:db:0b:0b), Dst: 74:6f:88:ef:de:0b (74:6f:88:ef:de:0b)  
 Internet Protocol Version 4, Src: 192.168.0.107, Dst: 8.8.8.8  
 User Datagram Protocol, Src Port: 57967, Dst Port: 53  
 Domain Name System (query)  
 Transaction ID: 0x8ee3  
 Flags: 0x0100 Standard query  
 0... .. = Response: Message is a query  
 0000 0... .. = Opcode: Standard query (0)  
 ... .. = Truncated: Message is not truncated  
 ... .. = Recursion desired: Do query recursively  
 ... .. = Z: reserved (0)  
 ... .. = Non-authenticated data: Unacceptable  
 Questions: 1  
 Answer RRs: 0  
 Authority RRs: 0  
 Additional RRs: 1  
 Queries  
 push.services.mozilla.com: type A, class IN  
 Name: push.services.mozilla.com  
 [Name Length: 25]  
 [Label Count: 4]  
 Type: A (Host Address) (1)  
 Class: IN (0x0001)  
 Additional records  
 <Root>: type OPT  
 [Response In: 161]

0000 74 6f 88 ef de 0b 0b 39 57 db 0b 0b 08 00 45 00 to...:9 W...k...E:  
 0010 00 52 52 2c 00 00 40 11 57 4c c9 a8 00 0b 08 08 RR, @ WL...k...  
 0020 00 00 c2 6f 00 35 00 3e 5d b7 8e e3 01 00 00 01 ...o 5-> ] .....  
 0030 00 00 00 00 00 01 04 70 75 73 68 08 73 65 72 76 .....p ush serv

Packets: 1551 · Displayed: 66 (4.3%) · Dropped: 0 (0.0%) Profile: Def

### Capturar o tráfego DNS:

1. **Abrir o Wireshark e iniciar a captura de pacotes:**
  - Abri o Wireshark e selecionei a interface de rede ativa (Wi-Fi ou Ethernet).
  - Iniciei a captura clicando no ícone de "Start Capturing Packets" (ícone de aleta de tubarão verde).
2. **Acessar um site:**
  - No navegador, acessei o site [www.google.com](http://www.google.com).
3. **Parar a captura e filtrar os pacotes:**
  - Voltei ao Wireshark e cliquei no ícone de "Stop Capturing Packets" (ícone quadrado vermelho) para parar a captura.
  - Na barra de filtros, digitei **dns** e pressionei Enter para filtrar os pacotes DNS.
4. **Analisar os pacotes DNS:**
  - Cliquei em um pacote DNS na lista.
  - Na parte inferior do Wireshark, expandi a seção "Domain Name System (DNS)" para visualizar os detalhes.

### Resultados da Análise:

1. **Processo de Resolução de Nomes:**
  - Identifiquei pacotes DNS com requisições (queries) e respostas (responses).
  - As requisições (queries) eram do tipo **A** (Address), solicitando o endereço IP para o domínio [www.google.com](http://www.google.com).
2. **Servidores DNS Utilizados:**
  - Vi os endereços IP dos servidores DNS que responderam às requisições.
  - Por exemplo, o servidor DNS configurado no meu dispositivo ou fornecido pelo meu provedor de internet.
3. **Registros DNS Obtidos:**
  - Nas respostas DNS (responses), vi os registros **A** retornando os endereços IP associados ao domínio [www.google.com](http://www.google.com).
  - Por exemplo, a resposta pode ter incluído vários endereços IP para balanceamento de carga.

### Conclusão:

Ao capturar e analisar pacotes DNS com o Wireshark, observei como meu dispositivo realiza a resolução de nomes de domínio ao acessar um site. Identifiquei os pacotes de requisição e resposta, os servidores DNS envolvidos e os registros DNS obtidos. Isso é essencial para entender o funcionamento do DNS e diagnosticar problemas de resolução de nomes na rede.

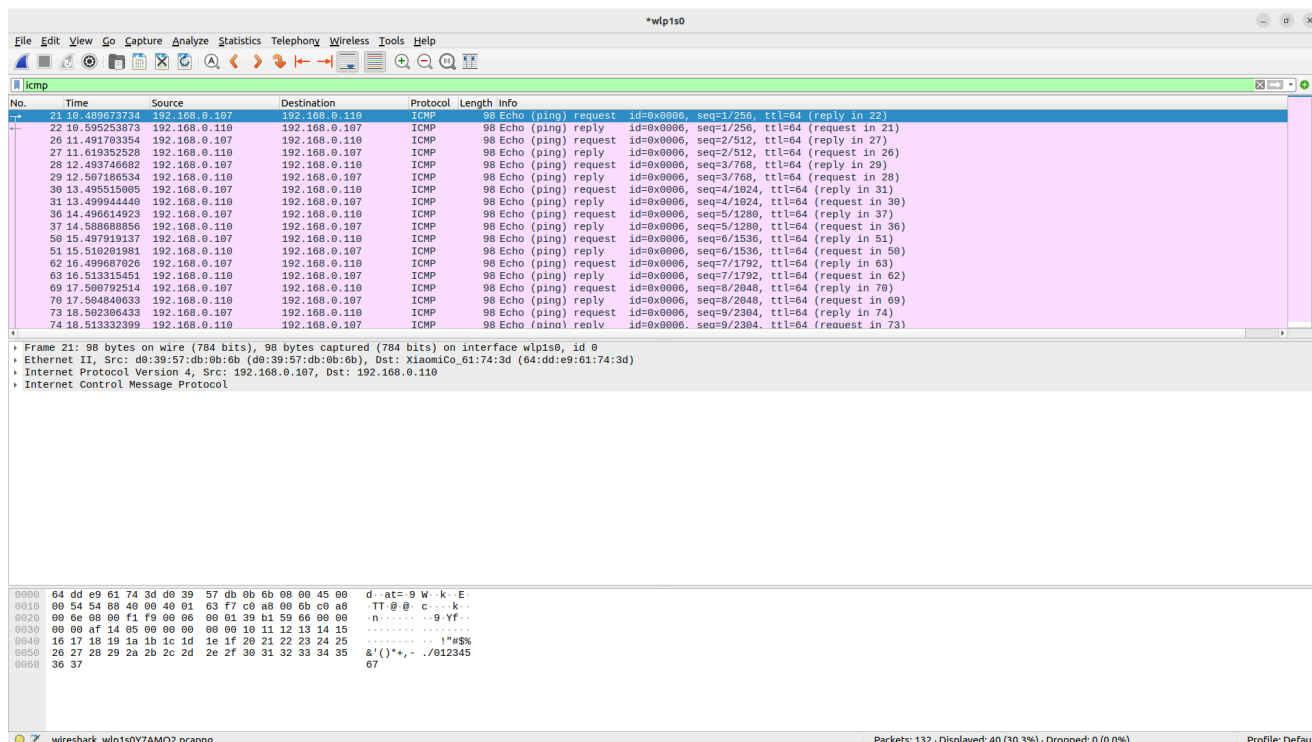


#### 4. Utilizando o software Wireshark capture o tráfego gerado pelo comando ping.

- 1. Abra o Wireshark e inicie a capture de pacotes
- 2. Utilize o comando ping teste algum computador em sua rede
- 3. Pare a captura e filtre os pacotes por ICMP

[illegible]





## Capturar o tráfego gerado pelo comando ping:

### 1. Abrir o Wireshark e iniciar a captura de pacotes:

- Abri o Wireshark no meu computador.
- Selecionei a interface de rede ativa (Wi-Fi ou Ethernet).
- Iniciei a captura clicando no ícone de "Start Capturing Packets" (ícone de aleta de tubarão verde).

### 2. Utilizar o comando ping para testar um computador na rede:

- Abri o terminal (Prompt de Comando no Windows ou Terminal no macOS/Linux).
- Utilizei o comando `ping` para testar a conectividade com um computador na minha rede. Por exemplo:

**ping 192.168.0.110**

### 1. Parar a captura e filtrar os pacotes por ICMP:

- Voltei ao Wireshark e cliquei no ícone de "Stop Capturing Packets" (ícone quadrado vermelho) para parar a captura.
- Na barra de filtros do Wireshark, digitei `icmp` e pressionei Enter para filtrar os pacotes ICMP.

## Resultados da Análise:

### 1. Pacotes ICMP Echo Request:

- Identifiquei pacotes ICMP Echo Request enviados pelo meu dispositivo para o endereço IP 192.168.0.110. Estes pacotes são solicitações de ping enviadas para verificar a conectividade.

### 2. Pacotes ICMP Echo Reply:

- Identifiquei pacotes ICMP Echo Reply recebidos do endereço IP 192.168.0.110 em resposta às solicitações de ping. Estes pacotes confirmam que o dispositivo destino está respondendo e a comunicação está estabelecida.

## Conclusão:

Ao capturar e analisar pacotes ICMP com o Wireshark, consegui observar a troca de mensagens de ping entre meu dispositivo e outro computador na rede. Identifiquei pacotes ICMP Echo Request e Echo Reply, o que confirma a conectividade e o funcionamento correto do comando ping. Isso é útil para diagnosticar problemas de rede e verificar a disponibilidade de dispositivos na rede.

## 5. Utilizando o software Wireshark capture o tráfego gerado pelo DHCP

- 1. Abra o Wireshark e inicie a capture de pacotes
- 2. Conecte e desconecte da sua rede
- 3. Pare a captura e filtre os pacotes por DHCP

[illegible]

\*wlp1s0
⌵ ⌶ ⌷

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

dhcpcd
⌵ ⌶ ⌷

No.	Time	Source	Destination	Protocol	Length	Info
8365	51.422256657	192.168.0.8	255.255.255.255	DHCP	323	DHCP Request - Transaction ID 0x29ed9b5
8367	51.465754108	192.168.0.254	255.255.255.255	DHCP	343	DHCP ACK - Transaction ID 0x29ed9b5

```

Seconds elapsed: 1
+ Bootp flags: 0x0000 (Unicast)
  Client IP address: 0.0.0.0
  Your (client) IP address: 0.0.0.0
  Next server IP address: 0.0.0.0
  Relay agent IP address: 0.0.0.0
  Client MAC address: d0:39:57:db:0b:6b (d0:39:57:db:0b:6b)
  Client hardware address padding: 00000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
+ Option: (53) DHCP Message Type (Request)
+ Option: (61) Client Identifier
+ Option: (55) Parameter Request List
+ Option: (57) Maximum DHCP Message Size
+ Option: (50) Requested IP Address (192.168.0.107)
+ Option: (12) Host Name
+ Option: (255) End

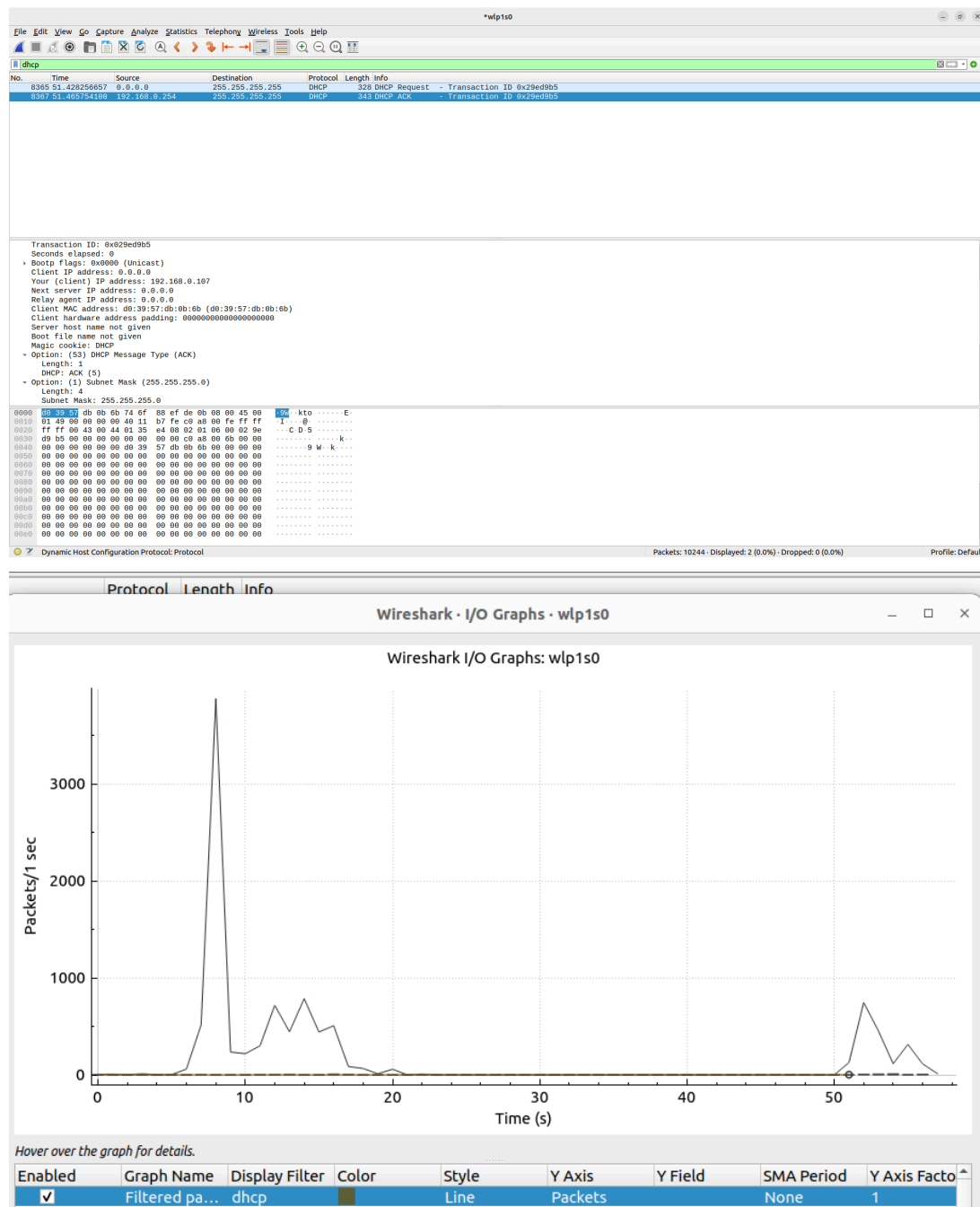
```

```

0000 ff ff ff ff ff ff d0 39 57 db 0b 6b 00 00 45 c0 .....9 W - k - E -
0010 01 3a 00 00 40 00 40 11 38 f4 00 00 00 00 ff ff ...- @ - @ 8 .....
0020 ff ff 00 44 00 43 01 20 26 82 01 01 00 00 52 9e ...- D - C & .....
0030 d9 b5 00 01 00 00 00 00 00 00 00 00 00 00 00 00 .....
0040 00 00 00 00 00 00 d0 39 57 db 0b 6b 00 00 00 00 .....9 W - k - .....
0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

```

Dynamic Host Configuration Protocol: Protocol
Packets: 10244 - Displayed: 2 (0.0%) - Dropped: 0 (0.0%)
Profile: Default



## O que é DHCP

### Definição e Função:

O DHCP (Dynamic Host Configuration Protocol) é um protocolo de rede usado para atribuir endereços IP e outras configurações de rede de forma automática a dispositivos em uma rede. Isso facilita a administração da rede, já que elimina a necessidade de configuração manual de cada dispositivo.

### Como o DHCP Funciona:

1. Descoberta (DHCP Discover): Quando um dispositivo (cliente) se conecta à rede, ele envia um broadcast DHCP Discover para localizar servidores DHCP disponíveis.
2. Oferta (DHCP Offer): O servidor DHCP responde com um pacote DHCP Offer, oferecendo um endereço IP ao cliente.
3. Requisição (DHCP Request): O cliente responde com um pacote DHCP Request, solicitando o endereço IP oferecido pelo servidor.
4. Confirmação (DHCP Acknowledge): O servidor DHCP responde com um pacote DHCP Acknowledge, confirmando a concessão do endereço IP ao cliente.

Além do endereço IP, o DHCP também pode fornecer outras configurações, como a máscara de sub-rede, gateway padrão e servidores DNS.

O que foi feito para demonstrar o funcionamento do DHCP

### Passos realizados:

1. **Abrir o Wireshark e iniciar a captura de pacotes:**
  - Abri o Wireshark, selecionei a interface de rede ativa (Wi-Fi ou Ethernet) e iniciei a captura de pacotes clicando no ícone de "Start Capturing Packets".
2. **Desconectar e reconectar da rede:**
  - Desconectei meu dispositivo da rede (desligando a conexão Wi-Fi ou desconectando o cabo Ethernet).
  - Reconectei meu dispositivo à rede (ligando a conexão Wi-Fi novamente ou reconectando o cabo Ethernet).
3. **Parar a captura e filtrar os pacotes por DHCP:**
  - Após reconectar à rede, voltei ao Wireshark e parei a captura de pacotes clicando no ícone de "Stop Capturing Packets".
  - Na barra de filtros do Wireshark, digitei `bootp` e pressionei Enter para filtrar os pacotes DHCP. No Wireshark, o protocolo DHCP é identificado como BOOTP (Bootstrap Protocol).

### Conclusão

Ao capturar e analisar pacotes DHCP com o Wireshark, consegui observar todo o processo de alocação de endereços IP dinâmicos entre meu dispositivo e um servidor DHCP. Esse processo é essencial para a configuração automática de dispositivos em uma rede, garantindo que todos obtenham endereços IP e outras configurações de rede necessárias para comunicação adequada.