



Universidade do Minho
Escola de Engenharia

Redes de Computadores TP3

Novembro de 2019



Ana Margarida Campos



Ana Catarina Gil



Tânia Rocha

1 Captura e análise de tramas Ethernet

1.1 Anote os endereços MAC de origem e de destino da trama capturada.

Origem: 3c:52:82:e5:bc:4c

Destino: 00:0c:29:d2:19:f0

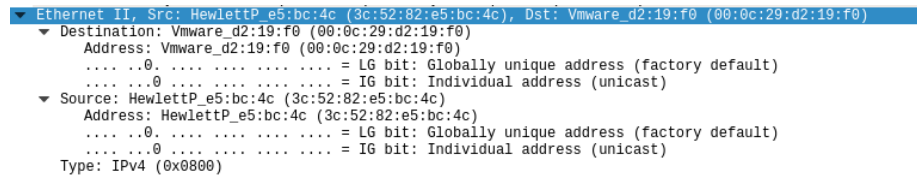


Figure 1: Endereços MAC de origem e destino

1.2 Identifique a que sistemas se referem. Justifique.

O endereço MAC de origem corresponde ao nosso computador e o de destino corresponde ao router.

1.3 Qual o valor hexadecimal do campo Type da trama Ethernet? O que significa?

O valor hexadecimal do campo Type é 0x0800 e indica o tipo de encapsulamento (IPv4 neste caso).

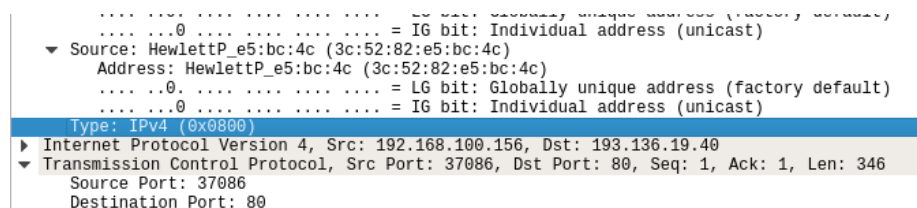


Figure 2: Valor hexadecimal do campo Type

1.4 Quantos bytes são usados desde o início da trama até ao caractere ASCII “G” do método HTTP GET? Calcule e indique, em percentagem, a sobrecarga (overhead) introduzida pela pilha protocolar no envio do HTTP GET.

Até ao carater “G” (ascii:0x47) são usados 66 bytes. No total são usados 412 bytes, logo a percentagem de sobrecarga é de $(66/412)*100 = 16,02$

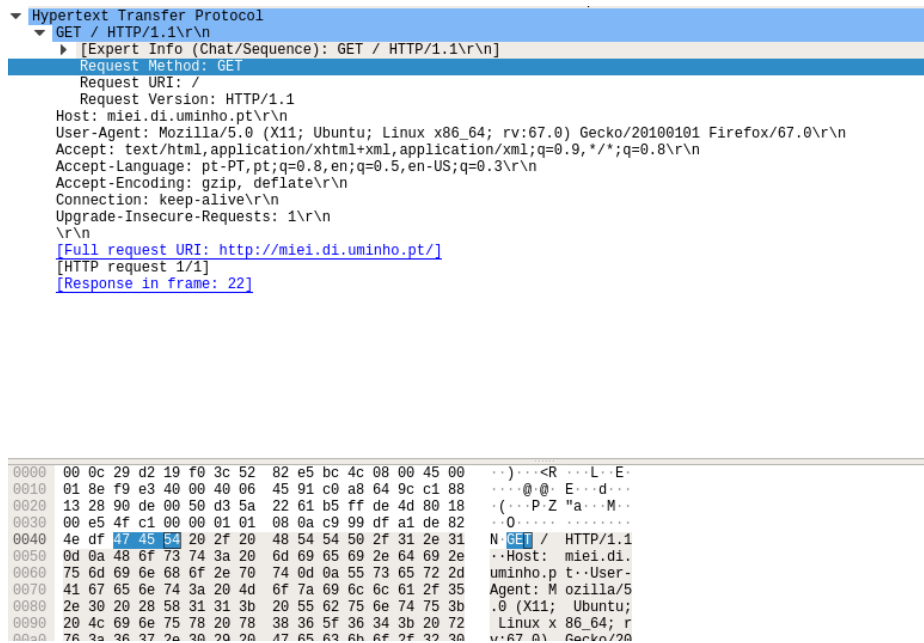


Figure 3: Pilha protocolar no envio HTTP GET

1.5 Através de visualização direta de uma trama capturada, verifique que, possivelmente, o campo FCS (Frame Check Sequence) usado para deteção de erros não está a ser usado. Em sua opinião, porque será?

Não aparece porque estamos numa ligação Ethernet um quadro danificado deve ser descartado.

1.6 Qual é o endereço Ethernet da fonte? A que sistema de rede corresponde? Justifique.

Endereço Ethernet da fonte = 00:0c:29:d2:19:f0

Corresponde ao router da rede local.

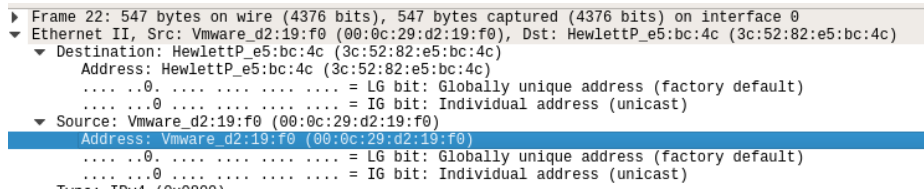


Figure 4: Endereço Ethernet

1.7 Qual é o endereço MAC do destino? A que sistema corresponde?

O endereço MAC do destino = 3c:52:82:e5:bc:4c

Este mesmo corresponde ao IP do nosso computador.

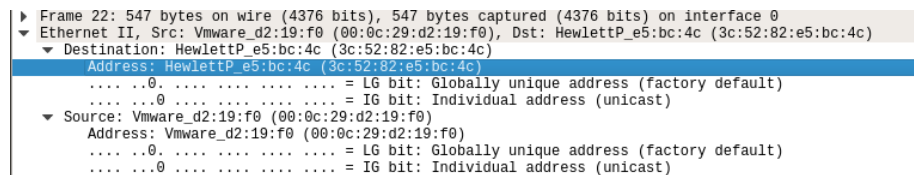


Figure 5: Endereço Mac do destino.

1.8 Atendendo ao conceito de desencapsulamento protocolar, identifique os vários protocolos contidos na trama recebida.

Os vários protocolos contidos na trama recebida são IPv4 ,TCP e HTTP.

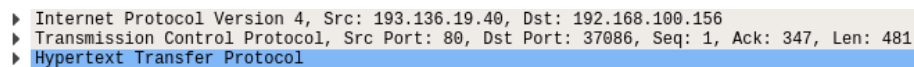


Figure 6: Protocolos contidos na trama.

2 Protocolo ARP

2.1 Observe o conteúdo da tabela ARP. Explique o significado de cada uma das colunas.

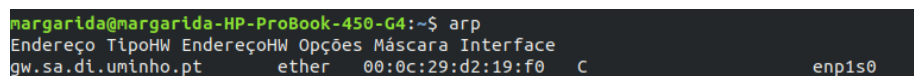


Figure 7: Comando arp.

Nota : Conforme confirmado na aula prática, o nosso computador não estava a executar o comando arp corretamente. Tal se verifica na fig.7. Mesmo assim, através de pesquisa conseguimos responder à questão.

Coluna Endereço : Representa o endereço IP do destino.

Coluna tipoHW : Representa o meio de ligação até ao destino.

Coluna EndereçoHW : Representa o MAC adress do destino.

Coluna Máscara : Representa o tipo de entrada.

Coluna Interface : Representa o tipo de interface.

2.2 Qual é o valor hexadecimal dos endereços origem e destino na trama Ethernet que contém a mensagem com o pedido ARP (ARP Request)? Como interpreta e justifica o endereço destino usado?

Como observado na figura o valor hexadecimal do destino é 0c:9d:92:35:f2:fb e o da origem é 3c:52:82:e5:bc:4c. Estes valores representam os endereços MAC. O endereço do destino Mac é para onde estamos a testar o ping.

```
▼ Ethernet II, Src: HewlettP_e5:bc:4c (3c:52:82:e5:bc:4c), Dst: AsustekC_35:f2:fb (0c:9d:92:35:f2:fb)
  ▶ Destination: AsustekC_35:f2:fb (0c:9d:92:35:f2:fb)
  ▶ Source: HewlettP_e5:bc:4c (3c:52:82:e5:bc:4c)
  Type: ARP (0x0806)
  - Address Resolution Protocol (request)
```

Figure 8: Valores Hexadécimais dos endereços de origem e de destino.

2.3 Qual o valor hexadecimal do campo tipo da trama Ethernet? O que indica?

Como observado na figura anterior, o valor do campo tipo da trama Ethernet é ARP(0x0806). Isto indica qual o protocolo utilizado.

2.4 Qual o valor do campo ARP opcode? O que especifica? Se necessário, consulte a RFC do protocolo ARP (<http://tools.ietf.org/html/rfc826.html>)

O valor do campo ARP opcode é Request(1). Isto significa que, neste caso, a trama tem como objetivo o pedido do endereço MAC destino da trama em questão. RFC é um método de conversão de endereços IP em endereços Ethernet.

```
type: ARP (0x0806)
▼ Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: HewlettP_e5:bc:4c (3c:52:82:e5:bc:4c)
  Sender IP address: 192.168.100.156
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.100.211
```

Figure 9: Valor do campo ARP code.

2.5 Identifique que tipo de endereços está contido na mensagem ARP? Que conclui?

Os endereços contidos na mensagem ARP são o IP e o MAC da origem e do destino. Ou seja os endereços que permitem a troca de tramas entre estes.

2.6 Explícite que tipo de pedido ou pergunta é feito pelo host de origem?

O host de origem pede o endereço MAC para o qual pretende enviar a trama.

1010	28.139748147	HewlettP_e5:bc:4c	AsustekC_35:f2:fb	ARP	42 Who has 192.168.100.211? Tell 192.168.100.156
1011	28.139899925	AsustekC_35:f2:fb	HewlettP_e5:bc:4c	ARP	60 192.168.100.211 is at 0c:9d:92:35:f2:fb

Figure 10: Host de origem.

2.7 Localize a mensagem ARP que é a resposta ao pedido ARP efectuado.

2.7.1 Qual o valor do campo ARP opcode? O que especifica?

Neste caso, o valor do campo ARP opcode é Reply(2).

```

Padding: 0000000000000000000000000000000000000000000000000000000
▼ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: AsustekC_35:f2:fb (0c:9d:92:35:f2:fb)
    Sender IP address: 192.168.100.211
    Target MAC address: HewlettP_e5:bc:4c (3c:52:82:e5:bc:4c)
    Target IP address: 192.168.100.156

```

Figure 11: Mensagem ARP.

2.7.2 Em que posição da mensagem ARP está a resposta ao pedido ARP?

A posição da mensagem ARP encontra-se desde os 23 aos 28 bytes.

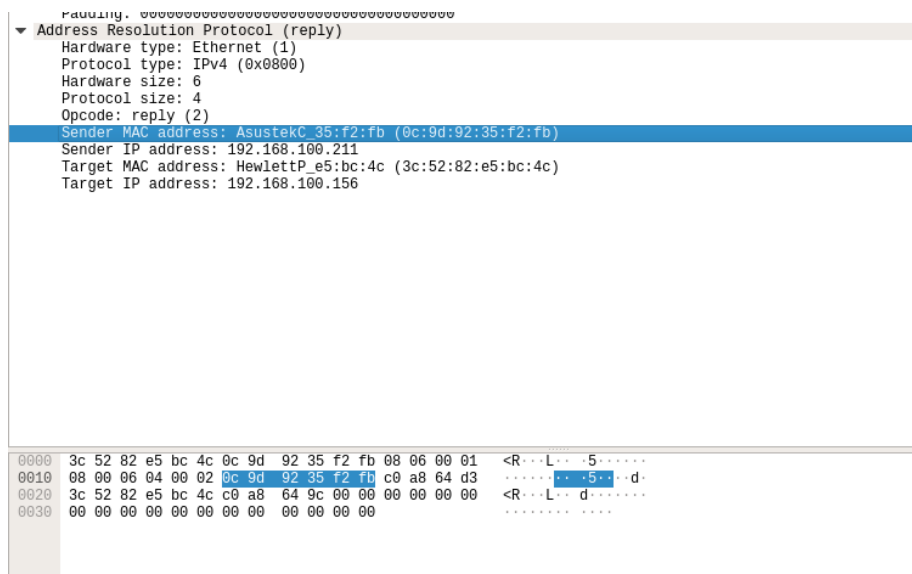


Figure 12: Mensagem ARP.

2.8 Identifique um pacote de pedido ARP gratuito originado pelo seu sistema. Analise o conteúdo de um pedido ARP gratuito e identifique em que se distingue dos restantes pedidos ARP. Registe a trama Ethernet correspondente. Qual o resultado esperado face ao pedido ARP gratuito enviado?

Com a utilização do comando `arping -U 192.168.100.156` (o IP da nossa máquina), obtivemos ARP's gratuitos. O que diferencia o pedido ARP gratuito dos restantes pedidos é que no pedido ARP gratuito existe uma flag `Is gratuitous: True` e apresenta endereço de destino igual ao de origem. Prevê-se que não exista resposta por parte deste ARP gratuito, pois isso significaria que existe algo na rede com um IP igual ao da nossa máquina.

No.	Time	Source	Destination	Protocol	Length	Info
42	9.965619533	Vmware_d2:19:f0	HewlettP_e5:bc:4c	ARP	60	Who has 192.168.100.156? Tell 192.168.100.254
43	9.965635900	HewlettP_e5:bc:4c	Vmware_d2:19:f0	ARP	42	192.168.100.156 is at 3c:52:82:e5:bc:4c
760	41.473699527	HewlettP_e5:bc:4c	Broadcast	ARP	42	Who has 192.168.100.254? Tell 192.168.100.156
761	41.473982766	Vmware_d2:19:f0	HewlettP_e5:bc:4c	ARP	60	192.168.100.254 is at 00:0c:29:d2:19:f0
949	50.434249642	Vmware_d2:19:f0	HewlettP_e5:bc:4c	ARP	60	Who has 192.168.100.156? Tell 192.168.100.254
950	50.434287861	HewlettP_e5:bc:4c	Vmware_d2:19:f0	ARP	42	192.168.100.156 is at 3c:52:82:e5:bc:4c
1127	93.974371514	Vmware_d2:19:f0	HewlettP_e5:bc:4c	ARP	60	Who has 192.168.100.156? Tell 192.168.100.254
1128	93.974394210	HewlettP_e5:bc:4c	Vmware_d2:19:f0	ARP	42	192.168.100.156 is at 3c:52:82:e5:bc:4c
1168	107.001965764	HewlettP_e5:bc:4c	Broadcast	ARP	42	Gratuitous ARP for 192.168.100.156 (Request)
1170	108.092227558	HewlettP_e5:bc:4c	Broadcast	ARP	42	Gratuitous ARP for 192.168.100.156 (Request)
1172	109.092436023	HewlettP_e5:bc:4c	Broadcast	ARP	42	Gratuitous ARP for 192.168.100.156 (Request)
▶ Frame 1168: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0						
▶ Ethernet II, Src: HewlettP_e5:bc:4c (3c:52:82:e5:bc:4c), Dst: Broadcast (ff:ff:ff:ff:ff:ff)						
▼ Address Resolution Protocol (request/gratuitous ARP)						
Hardware type: Ethernet (1)						
Protocol type: IPv4 (0x0800)						
Hardware size: 6						
Protocol size: 4						
Opcode: request (1)						
[is gratuitous: true]						
Sender MAC address: HewlettP_e5:bc:4c (3c:52:82:e5:bc:4c)						
Sender IP address: 192.168.100.156						
Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)						
Target IP address: 192.168.100.156						

Figure 13: ARP Gratuito.

3 Domínios de colisão

3.1 Faça ping de n1 para n2. Verifique com a opção tcpdump como flui o tráfego nas diversas interfaces dos vários dispositivos. Que conclui?

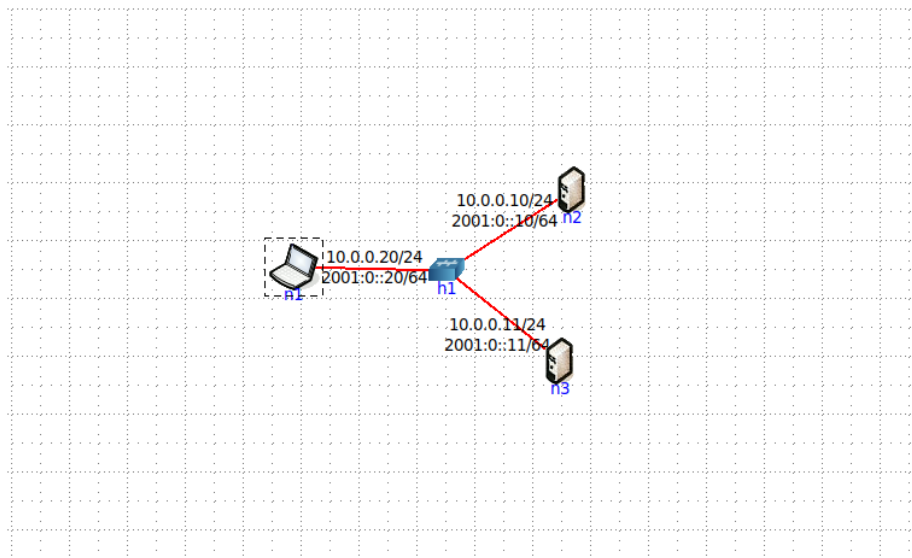


Figure 14: Mensagem ARP.


```

root@n2:/tmp/pycore.45533/n2.conf# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
<16:34:13.213922 IP6 fe80::fc66:97ff:fe61:c388 > ip6-allrouters: ICMP6, router solicitation, length 16
16:34:14.237975 IP6 fe80::fc66:97ff:fe61:c388 > ip6-allrouters: ICMP6, router solicitation, length 16
16:34:16.429949 IP6 fe80::200:ff:feaa:0 > ip6-allrouters: ICMP6, router solicitation, length 16
16:34:18.126371 IP6 fe80::3c35:18ff:fe8b:2368.mnns > ff02::fb.mnns: 0 [9q] PTR (QM)? nfs_tcp.local PTR (QM)? ip6_tcp.local PTR (QM)? ip6_tcp.local PTR (QM)? ftp_tcp.local PTR (QM)? webdav_tcp.local PTR (QM)? webdav_tcp.local PTR (QM)? smb_tcp.local PTR (QM)? afpovertcp_tcp.local (141)
16:34:19.403971 IP6 n2 > ip6-allrouters: ICMP6, router solicitation, length 16
16:34:19.403971 IP6 n2 > ip6-allrouters: ICMP6, router solicitation, length 16
16:34:19.102807 IP6 fe80::200:ff:feaa:0 > ip6-allrouters: ICMP6, router solicitation, length 16
16:34:43.478800 IP6 fe80::fc66:97ff:fe61:c388 > ip6-allrouters: ICMP6, router solicitation, length 16
16:34:43.478893 IP6 fe80::3c35:18ff:fe8b:2368 > ip6-allrouters: ICMP6, router solicitation, length 16
16:34:49.413855 ARP, Request who-has n2 tell 10.0.0.20, length 28
16:34:49.413116 ARP, Reply n2 is-at 00:00:00:aa:00:01 (oui Ethernet), length 28
16:34:49.413142 IP 10.0.0.20 > n2: ICMP echo request, id 27, seq 1, length 64
16:34:49.413163 IP n2 > 10.0.0.20: ICMP echo reply, id 27, seq 1, length 64
16:34:49.422411 IP6 fe80::200:ff:feaa:2 > ip6-allrouters: ICMP6, router solicitation, length 16
16:34:50.42911 IP6 fe80::3c35:18ff:fe8b:2368.mnns > ff02::fb.mnns: 0 [9q] PTR (QM)? nfs_tcp.local PTR (QM)? ip6_tcp.local PTR (QM)? ip6_tcp.local PTR (QM)? ftp_tcp.local PTR (QM)? webdav_tcp.local PTR (QM)? webdav_tcp.local PTR (QM)? smb_tcp.local PTR (QM)? afpovertcp_tcp.local (141)
16:34:50.429254 IP6 fe80::fc66:97ff:fe61:c388.mnns > ff02::fb.mnns: 0 [9q] PTR (QM)? nfs_tcp.local PTR (QM)? ip6_tcp.local PTR (QM)? ip6_tcp.local PTR (QM)? ftp_tcp.local PTR (QM)? webdav_tcp.local PTR (QM)? webdav_tcp.local PTR (QM)? smb_tcp.local PTR (QM)? afpovertcp_tcp.local (141)
16:34:50.41411 IP 10.0.0.20 > n2: ICMP echo request, id 27, seq 2, length 64
16:34:50.414052 IP n2 > 10.0.0.20: ICMP echo reply, id 27, seq 2, length 64
16:34:54.941970 ARP, Request who-has 10.0.0.20 tell n2, length 28
16:34:54.942153 ARP, Reply 10.0.0.20 is-at 00:00:00:aa:00:00 (oui Ethernet), length 28
16:34:55.965974 IP6 n2 > ip6-allrouters: ICMP6, router solicitation, length 16
16:34:55.966004 IP6 fe80::200:ff:feaa:0 > ip6-allrouters: ICMP6, router solicitation, length 16

22 packets captured
22 packets received by filter
0 packets dropped by kernel
root@n2:/tmp/pycore.45533/n2.conf#

```

Figure 15: tcpdump no servidor n2.

```

root@n3:/tmp/pycore.45533/n3.conf# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
<16:34:13.102804 IP6 fe80::200:ff:feaa:0 > ip6-allrouters: ICMP6, router solicitation, length 16
16:34:13.102202 IP6 fe80::200:ff:feaa:1 > ip6-allrouters: ICMP6, router solicitation, length 16
16:34:13.109359 IP6 fe80::ac67:c5ff:fe17:283f.mnns > ff02::fb.mnns: 0 [9q] PTR (QM)? nfs_tcp.local PTR (QM)? ip6_tcp.local PTR (QM)? ip6_tcp.local PTR (QM)? ftp_tcp.local PTR (QM)? webdav_tcp.local PTR (QM)? webdav_tcp.local PTR (QM)? smb_tcp.local PTR (QM)? afpovertcp_tcp.local (141)
16:34:43.478800 IP6 fe80::3c35:18ff:fe8b:2368 > ip6-allrouters: ICMP6, router solicitation, length 16
16:34:45.725843 IP6 fe80::ac67:c5ff:fe17:283f > ip6-allrouters: ICMP6, router solicitation, length 16
16:34:49.413855 ARP, Request who-has 10.0.0.10 tell 10.0.0.20, length 28
16:34:49.413124 ARP, Reply 10.0.0.10 is-at 00:00:00:aa:00:01 (oui Ethernet), length 28
16:34:49.413241 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 27, seq 1, length 64
16:34:49.413168 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 27, seq 1, length 64
16:34:50.422801 IP6 n3 > ip6-allrouters: ICMP6, router solicitation, length 16
16:34:50.157907 IP6 fe80::3c35:18ff:fe8b:2368.mnns > ff02::fb.mnns: 0 [9q] PTR (QM)? nfs_tcp.local PTR (QM)? ip6_tcp.local PTR (QM)? ip6_tcp.local PTR (QM)? ftp_tcp.local PTR (QM)? webdav_tcp.local PTR (QM)? webdav_tcp.local PTR (QM)? smb_tcp.local PTR (QM)? afpovertcp_tcp.local (141)
16:34:50.414008 IP 10.0.0.10 > 10.0.0.10: ICMP echo request, id 27, seq 2, length 64
16:34:50.414058 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 27, seq 2, length 64
16:34:51.200476 IP6 fe80::ac67:c5ff:fe17:283f.mnns > ff02::fb.mnns: 0 [9q] PTR (QM)? nfs_tcp.local PTR (QM)? ip6_tcp.local PTR (QM)? ip6_tcp.local PTR (QM)? ftp_tcp.local PTR (QM)? webdav_tcp.local PTR (QM)? webdav_tcp.local PTR (QM)? smb_tcp.local PTR (QM)? afpovertcp_tcp.local (141)
16:34:54.942149 ARP, Request who-has 10.0.0.20 tell 10.0.0.10, length 28
16:34:54.942149 ARP, Reply 10.0.0.20 is-at 00:00:00:aa:00:00 (oui Ethernet), length 28
16:34:55.966004 IP6 fe80::200:ff:feaa:0 > ip6-allrouters: ICMP6, router solicitation, length 16
16:34:55.966004 IP6 fe80::200:ff:feaa:1 > ip6-allrouters: ICMP6, router solicitation, length 16

18 packets captured
18 packets received by filter
0 packets dropped by kernel
root@n3:/tmp/pycore.45533/n3.conf#

```

Figure 16: tcpdump no servidor n3.

Como se pode observar nas figuras, notamos que todos os servidores recebem pacotes quando é efetuado ping do laptop n1 para o servidor n2 com recorrencia ao comando tcpdump.

Isto deve-se ao facto de que quando o hub recebe um pacote de dados, este reencaminha os mesmos para todos os dispositivos que se encontram na mesma rede.

3.2 Na topologia de rede substitua o hub por um switch.

Repita os procedimentos que realizou na pergunta anterior. Comente os resultados obtidos quanto à utilização de hubs e switches no contexto de controlar ou dividir domínios de colisão. Documente as suas observações e conclusões com base no tráfego observado/capturado.

4 Conclusão

Com a realização deste trabalho conseguimos aprofundar conhecimentos acerca dos endereços MAC, ARP, Ethernet e interligações de redes locais. Isto deve-se, ao facto de para a resposta às questões do enunciado termos de efetuar capturas e as fazer as respetivas análises de tramas Ethernet com auxílio do software wireshark.

Utilizamos também a ferramenta CORE para podermos comparara eficácia da utilização dos switchs e hubs na diminuição de colisões de tramas Ethernet.