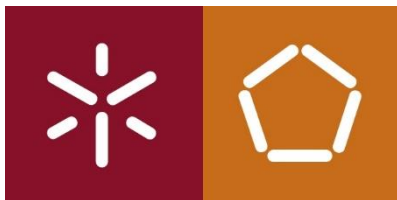


# Redes de Computadores



## Trabalho prático 3

28 de novembro de 2019

### **Grupo nº 6**

Filipa Alves dos Santos (A83631)

Hugo André Coelho Cardoso (A85006)

João da Cunha e Costa (A84775)



Mestrado Integrado em Engenharia Informática

Universidade do Minho

# Índice de conteúdos

<b>1. Questões e Repostas .....</b>	<b>3</b>
1.1. Captura e análise de tramas Ethernet.....	3
1.2. Protocolo ARP.....	5
1.3. Domínios de colisão.....	8
<b>2. Conclusões.....</b>	<b>10</b>

# 1. Questões e Respostas

## 1.1. Captura e análise de tramas Ethernet

3) A captura e análise de tramas Ethernet será efetuada usando a aplicação Wireshark. Assegure-se que utiliza a ligação com fios, i.e., a ligação à rede Ethernet da sala de aula e que a cache do seu browser está vazia e está conetado em rede através da interface Ethernet.

(...)

Obtenha o número de ordem da sequência de bytes capturada (coluna da esquerda na janela do Wireshark) correspondente à mensagem HTTP GET enviada pelo seu computador para o servidor Web, bem como o começo da respectiva mensagem HTTP Response proveniente do servidor.

(...)

Responda às perguntas seguintes com base no conteúdo da trama Ethernet que contém a mensagem HTTP GET (sempre que aplicável, deve incluir a impressão dos dados relativa ao pacote capturado (ou parte dele) necessária para fundamentar a resposta à questão colocada.

http						
No.	Time	Source	Destination	Protocol	Length	Info
217	29.820186	192.168.100.196	34.198.11.139	HTTP	374	GET /pulse?off&user=0&url_heartbeat=1,0,140,140,0 HTTP/1.1
220	29.961384	34.198.11.139	192.168.100.196	HTTP	194	HTTP/1.1 200 OK

Hypertext Transfer Protocol	
GET /pulse?off&user=0&url_heartbeat=1,0,140,140,0 HTTP/1.1\r\n	
[Expert Info (Chat/Sequence): GET /pulse?off&user=0&url_heartbeat=1,0,140,140,0 HTTP/1.1\r\n]	
Request Method: GET	
Request URI: /pulse?off&user=0&url_heartbeat=1,0,140,140,0	
Request URI Path: /pulse	
Request URI Query: off&user=0&url_heartbeat=1,0,140,140,0	
Request URI Query Parameter: off	
Request URI Query Parameter: user=0	
Request URI Query Parameter: url_heartbeat=1,0,140,140,0	
Request Version: HTTP/1.1	
User-Agent: Mozilla/5.0 EA Download Manager Origin/10.5.55.33574\r\n	
<User-Agent: Mozilla/5.0 EA Download Manager Origin/10.5.55.33574\r\n>	
X-Origin-UID: 14132179657452050087\r\n	
X-Origin-Platform: PCWIN\r\n	
localeInfo: pt_BR\r\n	
Accept-Language: pt-BR\r\n	
<Accept-Language: pt-BR\r\n>	
Connection: Keep-Alive\r\n	
<Connection: Keep-Alive\r\n>	
Accept-Encoding: gzip, deflate\r\n	

0030	01 fd 55 18 00 00 47 45 54 20 2f 70 75 6c 73 65	..U...GET /pulse
0040	3f 6f 66 66 26 75 73 65 72 3d 30 26 75 72 6c 5f	?off&user=0&url_
0050	68 65 61 72 74 62 65 61 74 3d 31 2c 30 2c 31 34	heartbeat=1,0,14
0060	30 2c 31 34 30 2c 30 20 48 54 54 50 2f 31 2e 31	0,140,0 HTTP/1.1
0070	0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f	..User-Agent: Mo
0080	7a 69 6c 6c 61 2f 35 2e 30 20 45 41 20 44 6f 77	zilla/5.0 EA Dow
0090	6e 6c 6f 61 64 20 4d 61 6e 61 67 65 72 20 4f 72	nload Manager Or
00a0	69 67 69 6e 2f 31 30 2e 35 2e 35 35 2e 33 33 35	igin/10. 5.55.335
00b0	37 34 0d 0a 58 2d 4f 72 69 67 69 6e 2d 55 49 44	74..X-Origin-UID
00c0	3a 20 31 34 31 33 32 31 37 39 36 35 37 34 35 32	: 141321 79657452
00d0	30 35 30 30 38 37 0d 0a 58 2d 4f 72 69 67 69 6e	050087.. X-Origin
00e0	2d 50 6c 61 74 66 6f 72 6d 3a 20 50 43 57 49 4e	-Platform: PCWIN
00f0	0d 0a 6c 6f 63 61 6c 65 49 6e 66 6f 3a 20 70 74	..locale Info: pt
0100	5f 42 52 0d 0a 41 63 63 65 70 74 2d 4c 61 6e 67	_BR..Accept-Lang

No.	Time	Source	Destination	Protocol	Length	Info
214	29.505665	fe80::1b6:1ebd:2ef1...	ff02::1:ff00:9e9	ICMPv6	86	Multicast Listener Report
215	29.569906	fe80::6df7:4915:5ae...	ff02::1:ff97:1462	ICMPv6	86	Multicast Listener Report
216	29.693115	Vmware_d2:19:f0	Broadcast	ARP	60	Who has 192.168.100.176? Tell 192.168.100.254
217	29.820186	192.168.100.196	34.198.11.139	HTTP	374	GET /pulse?off&user=0&url_heartbeat=1,0,140,140,0 HTTP/1.1
218	29.860624	Tp-LinkT_26:31:4f	Broadcast	ARP	60	Who has 192.168.25.1? Tell 192.168.25.7
219	29.944072	fe80::9417:e7f4:cb8...	ff02::fb	ICMPv6	86	Multicast Listener Report
220	29.961384	34.198.11.139	192.168.100.196	HTTP	194	HTTP/1.1 200 OK
221	30.002918	192.168.100.196	34.198.11.139	TCP	54	60419 → 80 [ACK] Seq=321 Ack=141 Win=508 Len=0
222	30.069669	192.168.100.196	35.186.224.53	TLSv1.2	145	Application Data
> Frame 217: 374 bytes on wire (2992 bits), 374 bytes captured (2992 bits) on interface 0 > Ethernet II, Src: HewlettP_93:4b:2b (f4:30:b9:93:4b:2b), Dst: Vmware_d2:19:f0 (00:0c:29:d2:19:f0) > Destination: Vmware_d2:19:f0 (00:0c:29:d2:19:f0) <[Destination (resolved): Vmware_d2:19:f0]> Address: Vmware_d2:19:f0 (00:0c:29:d2:19:f0) <[Address (resolved): Vmware_d2:19:f0]> .... 0. .... = LG bit: Globally unique address (factory default) .... 0. .... = IG bit: Individual address (unicast) > Source: HewlettP_93:4b:2b (f4:30:b9:93:4b:2b) <[Source (resolved): HewlettP_93:4b:2b]> Address: HewlettP_93:4b:2b (f4:30:b9:93:4b:2b) <[Address (resolved): HewlettP_93:4b:2b]> .... 0. .... = LG bit: Globally unique address (factory default) .... 0. .... = IG bit: Individual address (unicast) Type: IPv4 (0x0800) > Internet Protocol Version 4, Src: 192.168.100.196, Dst: 34.198.11.139 > Transmission Control Protocol, Src Port: 60419, Dst Port: 80, Seq: 1, Ack: 1, Len: 320 > Hypertext Transfer Protocol						

0000	00 0c 29 d2 19 f0	f4 30 b9 93 4b 2b 08 00 45 00	..).-.-0..K+...E-
0010	01 68 97 c2 40 00 80 06	00 00 c0 a8 64 c4 22 c6	-h..@... ..d."
0020	0b 8b ec 03 00 50 00 50	2b 7b 49 64 2e 33 50 18	....P.P +{Id.3P
0030	01 fd 55 18 00 00 47 45	54 20 2f 70 75 6c 73 65	--U...GE T /pulse
0040	3f 6f 66 66 26 75 73 65	72 3d 30 26 75 72 6c 5f	?off&use r=0&url_
0050	68 65 61 72 74 62 65 61	74 3d 31 2c 30 2c 31 34	heartbea t=1,0,14
0060	30 2c 31 34 30 2c 30 20	48 54 54 50 2f 31 2e 31	0,140,0 HTTP/1.1

### 3.1) Anotar os endereços MAC de origem e de destino da trama capturada.

Endereço de origem: f4:30:b9:93:4b:2b

Endereço de destino: 00:0c:29:d2:19:f0

### 3.2) Identifique a que sistemas se referem. Justifique.

O source é o nosso computador ("HewlettP") e o destino corresponde à placa de rede "virtual" da sala de aula.

### 3.3) Qual o valor hexadecimal do campo Type da trama Ethernet? O que significa?

O valor hexadecimal do campo Type é 0x0800 que significa que encapsula um pacote IPv4.

3.4) 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.

0000	00	0c	29	d2	19	f0	f4	30	b9	93	4b	2b	08	00	45	00
0010	01	68	97	c2	40	00	80	06	00	00	c0	a8	64	c4	22	c6
0020	0b	8b	ec	03	00	50	00	50	2b	7b	49	64	2e	33	50	18
0030	01	fd	55	18	00	00	47	45	54	20	2f	70	75	6c	73	65

↓  
**G**

São usados 54 bytes até ao caractere “G” . Como a quantidade total de bytes da trama é de 374 bytes, a percentagem de sobrecarga é  $54/374 = 14,44\%$ .

3.5) 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?

O campo FCS não está a ser presente nesta trama porque pacotes em que se verificam erros são descartados (mau checksums) enquanto nos que não apresentam erros, como é o caso desta trama, não existe tal campo. A nível de redes por cabo, os erros raramente ocorrem.

## 1.3. Protocolo ARP

4) Inicie a captura de tráfego com o Wireshark, e aceda a <http://miei.di.uminho.pt>. Efetue também um ping para um host da sala de aula (e.g. ping 192.168.100.xxx) que esteja a ser usado por outro grupo. Pare a captura de tráfego e tente localizar o tráfego ARP. Se necessário limite os protocolos visíveis apenas a protocolos abaixo do nível IP. Para tal, seleccione Analyze->Enabled Protocols e remova a selecção da opção IPv4 e IPv6. Responda às seguintes perguntas:

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

```
C:\Users\Zezoca>arp -a
```

Interface: 172.26.29.191 --- 0x6		
Internet Address	Physical Address	Type
1.1.1.10	00-78-88-a3-d2-89	dynamic
172.26.254.254	00-d0-03-ff-94-00	dynamic
172.26.255.255	ff-ff-ff-ff-ff-ff	static
224.0.0.22	01-00-5e-00-00-16	static
224.0.0.251	01-00-5e-00-00-fb	static
224.0.0.252	01-00-5e-00-00-fc	static
239.255.255.250	01-00-5e-7f-ff-fa	static
255.255.255.255	ff-ff-ff-ff-ff-ff	static

A primeira coluna representa o endereço IP do host e a segunda coluna representa o endereço MAC, ou seja, o endereço da ethernet associado ao IP. Já a terceira coluna representa o tipo de ligação estabelecida.

**4.10) 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?**

699	11.947110	HewlettP_93:4b:2b	Broadcast	ARP	42 Who has 192.168.100.157? Tell 192.168.100.196
700	11.947771	AsixElec_c6:45:60	HewlettP_93:4b:2b	ARP	60 192.168.100.157 is at 00:0e:c6:c6:45:60
734	12.576024	Vmware_d2:19:f0	Broadcast	ARP	60 Who has 192.168.100.191? Tell 192.168.100.254

```

> Frame 699: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
v Ethernet II, Src: HewlettP_93:4b:2b (f4:30:b9:93:4b:2b), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: HewlettP_93:4b:2b (f4:30:b9:93:4b:2b)
    Type: ARP (0x0806)
v 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_93:4b:2b (f4:30:b9:93:4b:2b)
  Sender IP address: 192.168.100.196
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.100.157
  
```

O valor hexadecimal dos endereços origem e destino são, respetivamente, f4:30:b9:93:4b:2b e ff:ff:ff:ff:ff:ff. O endereço de destino usado significa que todos os nós da rede local vão receber esta trama Ethernet.

**4.11) Qual o valor hexadecimal do campo tipo da trama Ethernet? O que indica?**

O valor hexadecimal do campo Type é 0x0806 que significa que encapsula um pacote ARP.

**4.12) Qual o valor do campo ARP opcode? O que especifica?**

O valor do campo ARP opcode é “request(1)” e especifica que é uma mensagem de pedido (“Who has 192.168.100.157?”).

**4.13) Identifique que tipo de endereços está contido na mensagem ARP? Que conclui?**

Estão contidos endereços do tipo MAC e IP. Podemos concluir através disto que host de endereço IP “192.168.100.196” (Sender IP adress) e MAC “f4:30:b9:93:4b:2b” (Sender MAC adress) é a origem da mensagem, que pretende saber o endereço MAC do host de IP “192.168.100.157” (Target IP adress). Assim, vai-se mandar a mensagem como broadcast, explicando o Target Mac Adress “00:00:00:00:00:00”.

**4.14) Explícite que tipo de pedido ou pergunta é feito pelo host de origem?**

Perguntamos a todos os hosts da rede local quem tem o IP “192.168.100.157” e pedimos para enviarem o MAC correspondente para “192.168.100.196”.

**4.15) Localize a mensagem ARP que é a resposta ao pedido ARP efectuado.**

699	11.947110	HewlettP_93:4b:2b	Broadcast	ARP	42 Who has 192.168.100.157? Tell 192.168.100.196
700	11.947771	AsixElec_c6:45:60	HewlettP_93:4b:2b	ARP	60 192.168.100.157 is at 00:0e:c6:c6:45:60
734	12.576024	Vmware_d2:19:f0	Broadcast	ARP	60 Who has 192.168.100.191? Tell 192.168.100.254

```

> Frame 700: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
> Ethernet II, Src: AsixElec_c6:45:60 (00:0e:c6:c6:45:60), Dst: HewlettP_93:4b:2b (f4:30:b9:93:4b:2b)
v Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: AsixElec_c6:45:60 (00:0e:c6:c6:45:60)
    Sender IP address: 192.168.100.157
    Target MAC address: HewlettP_93:4b:2b (f4:30:b9:93:4b:2b)
    Target IP address: 192.168.100.196

```

- a) Qual o valor do campo ARP opcode? O que especifica?

O valor do campo ARP opcode é “reply(2)” e especifica que é uma mensagem de resposta (“192.168.100.157 is at 00:0e:c6:c6:45:60”).

- b) Em que posição da mensagem ARP está a resposta ao pedido ARP?**

```
Sender MAC address: AsixElec_c6:45:60 (00:0e:c6:c6:45:60)
Sender IP address: 192.168.100.157
Target MAC address: HewlettP_93:4b:2b (f4:30:b9:93:4b:2b)
Target IP address: 192.168.100.196
```

0000	f4 30 b9 93 4b 2b 00 0e c6 c6 45 60 08 06 00 01	.0..K+...E'...
0010	08 00 06 04 00 02 00 0e c6 c6 45 60 c0 a8 64 9d	.....E'd.
0020	f4 30 b9 93 4b 2b c0 a8 64 c4 00 00 00 00 00 00	.0..K+...d.....
0030	00 00 00 00 00 00 00 00 00 00 00 00	.....

A mensagem está contida entre os bytes 23 e 28.

**4.16)** 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?

89 10.026689	HewlettP_93:4b:2b	Broadcast	ARP	42 Who has 192.168.100.196? Tell 0.0.0.0
174 11.026605	HewlettP_93:4b:2b	Broadcast	ARP	42 Who has 192.168.100.196? Tell 0.0.0.0
388 12.026727	HewlettP_93:4b:2b	Broadcast	ARP	42 Gratuitous ARP for 192.168.100.196 (Request)

```

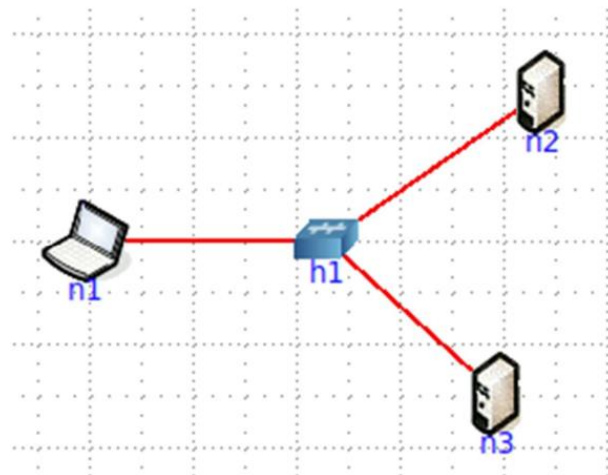
> Frame 388: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▼ Ethernet II, Src: HewlettP_93:4b:2b (f4:30:b9:93:4b:2b), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: HewlettP_93:4b:2b (f4:30:b9:93:4b:2b)
  Type: ARP (0x0806)
▼ Address 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_93:4b:2b (f4:30:b9:93:4b:2b)
  Sender IP address: 192.168.100.196
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.100.196

```

O que distingue o ARP gratuito dos restantes pedidos é a flag “[Is gratuitos: True]”, que indica que o pedido é de facto, gratuito. O host envia um ARP gratuito ao ligar-se de novo à rede, quando lhe é atribuído um endereço IP. Este envio permite informar os dispositivos da rede local do seu novo endereço MAC, para poderem atualizar as suas tabelas ARP.

### 1.3. Domínios de colisão

5) Construa uma topologia no emulador CORE com um host (n1) e dois servidores (n2, n3) interligados através de um hub.



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

```

root@n1:/tmp/pycore.33851/n1.conf# ping 10.0.0.10
PING 10.0.0.10 (10.0.0.10) 56(84) bytes of data.
64 bytes from 10.0.0.10: icmp_seq=1 ttl=64 time=0.110 ms
64 bytes from 10.0.0.10: icmp_seq=2 ttl=64 time=0.193 ms
64 bytes from 10.0.0.10: icmp_seq=3 ttl=64 time=0.258 ms
64 bytes from 10.0.0.10: icmp_seq=4 ttl=64 time=0.191 ms
--- 10.0.0.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3078ms
rtt min/avg/max/mdev = 0.110/0.189/0.258/0.054 ms
root@n1:/tmp/pycore.33851/n1.conf#

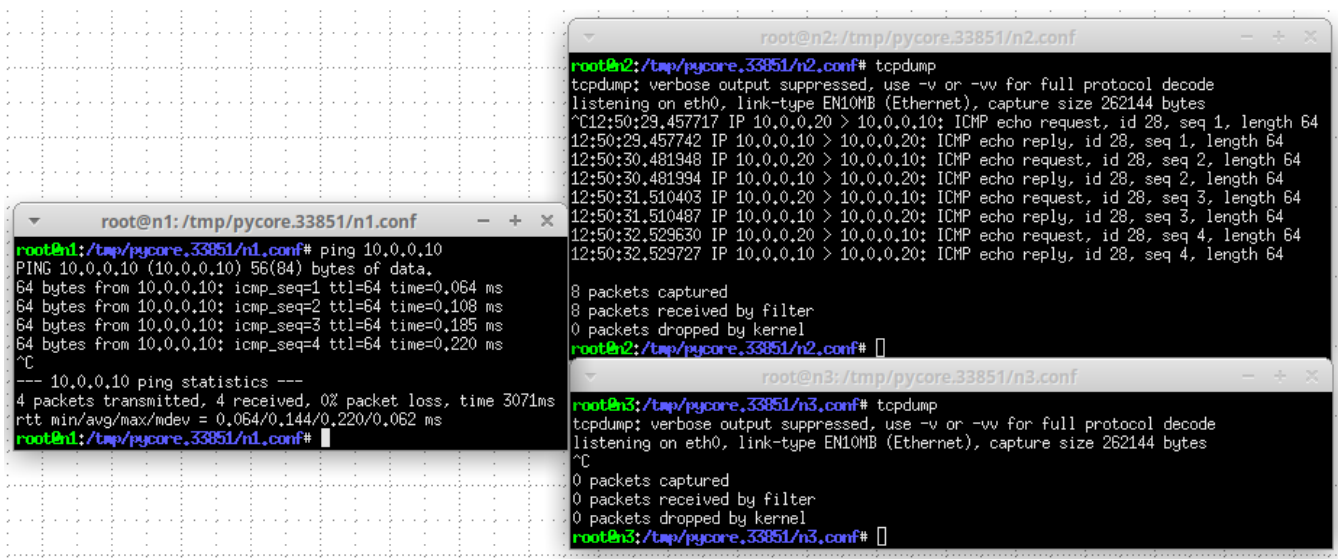
root@n2:/tmp/pycore.33851/n2.conf# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C12:47:02.447281 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 32, seq 1, length 64
12:47:02.447322 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 32, seq 1, length 64
12:47:03.473657 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 32, seq 2, length 64
12:47:03.473750 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 32, seq 2, length 64
12:47:04.498232 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 32, seq 3, length 64
12:47:04.498347 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 32, seq 3, length 64
12:47:05.525579 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 32, seq 4, length 64
12:47:05.525665 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 32, seq 4, length 64
8 packets captured
8 packets received by filter
0 packets dropped by kernel
root@n2:/tmp/pycore.33851/n2.conf#

root@n3:/tmp/pycore.33851/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
^C12:46:57.329145 IP6 fe80::a808:96ff:fe53:1d14 > ip6-allrouters: ICMP6, router solicitation, length 16
12:47:02.447278 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 32, seq 1, length 64
12:47:02.447330 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 32, seq 1, length 64
12:47:03.473653 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 32, seq 2, length 64
12:47:03.473765 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 32, seq 2, length 64
12:47:04.498226 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 32, seq 3, length 64
12:47:04.498376 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 32, seq 3, length 64
12:47:05.525575 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 32, seq 4, length 64
12:47:05.525681 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 32, seq 4, length 64
9 packets captured
9 packets received by filter
0 packets dropped by kernel
root@n3:/tmp/pycore.33851/n3.conf#
  
```



Ao correr a opção tcpdump nos dois servidores (n2 e n3), podemos concluir que ambos capturam o mesmo tráfego, embora a comunicação seja feita apenas entre n1 e n2. O servidor n3 recebe na mesma os pacotes enviados pelo host dado que os dispositivos da rede estão ligados por um hub, que funciona como um repetidor de múltiplas portas, redistribuindo qualquer sinal enviado na porta de input por todas as outras portas.

**5.18)** Fa 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.



The image displays three terminal windows from a network simulation. The left window, titled 'root@n1:/tmp/pycore.33851/n1.conf', shows a successful ping of 10.0.0.10 with four packets, 0% loss, and an average RTT of 0.144ms. The top-right window, titled 'root@n2:/tmp/pycore.33851/n2.conf', shows a tcpdump capture on eth0 with 8 packets: four ICMP echo requests from 10.0.0.10 to 10.0.0.20 and four corresponding replies. The bottom-right window, titled 'root@n3:/tmp/pycore.33851/n3.conf', shows a tcpdump capture on eth0 with 0 packets, indicating no traffic was captured on this host.

```
root@n1:/tmp/pycore.33851/n1.conf# ping 10.0.0.10
PING 10.0.0.10 (10.0.0.10) 56(84) bytes of data:
64 bytes from 10.0.0.10: icmp_seq=1 ttl=64 time=0.064 ms
64 bytes from 10.0.0.10: icmp_seq=2 ttl=64 time=0.108 ms
64 bytes from 10.0.0.10: icmp_seq=3 ttl=64 time=0.185 ms
64 bytes from 10.0.0.10: icmp_seq=4 ttl=64 time=0.220 ms
^C
--- 10.0.0.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3071ms
rtt min/avg/max/mdev = 0.064/0.144/0.220/0.062 ms
root@n1:/tmp/pycore.33851/n1.conf#
```

```
root@n2:/tmp/pycore.33851/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
^C12:50:29.457717 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 28, seq 1, length 64
12:50:29.457742 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 28, seq 1, length 64
12:50:30.481948 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 28, seq 2, length 64
12:50:30.481994 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 28, seq 2, length 64
12:50:31.510403 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 28, seq 3, length 64
12:50:31.510487 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 28, seq 3, length 64
12:50:32.529630 IP 10.0.0.20 > 10.0.0.10: ICMP echo request, id 28, seq 4, length 64
12:50:32.529727 IP 10.0.0.10 > 10.0.0.20: ICMP echo reply, id 28, seq 4, length 64
8 packets captured
8 packets received by filter
0 packets dropped by kernel
root@n2:/tmp/pycore.33851/n2.conf#
```

```
root@n3:/tmp/pycore.33851/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
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
root@n3:/tmp/pycore.33851/n3.conf#
```

Um switch mantém para cada endereço MAC a indicação da interface de saída, com a ajuda de uma tabela de comutação, direcionando qualquer trama Ethernet que chegue para a interface apropriada.

Ao substituir o hub da arquitetura por um switch, paramos de ter a redistribuição de dados observada na pergunta anterior. Assim, neste caso, n3 não recebe tráfego nenhum, sendo a comunicação exclusivamente entre n1 e n2.

## 2. Conclusões

Este trabalho prático permitiu aprofundar e consolidar a matéria dada nas aulas teóricas, em particular a Camada de Ligação Lógica: Ethernet e Protocolo ARP.

Estudamos a partilha de endereços MAC em redes de computadores, através da análise de tráfego capturado com recurso ao Wireshark, e as vantagens e implicações do uso do protocolo ARP, que serve para efetuar o mapeamento de endereços de redes e endereços de uma tecnologia de ligação de dados, bem como o funcionamento dos domínios de colisão nestas últimas. Para tal efeito, recorreremos também ao uso da ferramenta CORE fornecida pelos docentes, tal como já tínhamos feito no trabalho prático anterior.

Concluindo, este trabalho serviu como um bom meio de estudar a camada de ligação de redes de computadores e avaliar os conhecimentos da mesma.