

UNIVERSIDADE DO MINHO

DEPARTAMENTO DE INFORMÁTICA

## Trabalho Prático 3

### Redes de Computadores

Grupo 21

Ana Filipa Pereira (A89589)      Carolina Santejo (A89500)  
Raquel Costa (A89464)

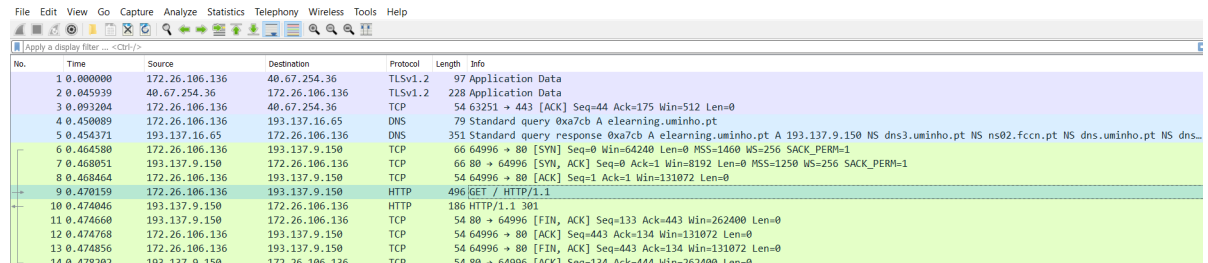
9 de dezembro de 2020

## Conteúdo

<b>1</b>	<b>Captura e análise de Tramas Ethernet</b>	<b>3</b>
1.1	.	3
1.2	1.	3
1.3	2.	3
1.4	3.	3
1.5	4.	4
1.6	5.	4
1.7	6.	4
1.8	7.	4
1.9	8.	5
<b>2</b>	<b>Protocolo ARP</b>	<b>6</b>
2.1	9.	6
2.2	10.	6
2.3	11.	6
2.4	12.	6
2.5	13.	7
2.6	14.	7
<b>3</b>	<b>ARP Gratuito</b>	<b>8</b>
3.1	15.	8
3.2	16.	8
<b>4</b>	<b>Conclusão</b>	<b>10</b>

# 1 Captura e análise de Tramas Ethernet

**1.1** 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.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.26.106.136	40.67.254.36	TLSv1.2	97	Application Data
2	0.045939	40.67.254.36	172.26.106.136	TLSv1.2	228	Application Data
3	0.093204	172.26.106.136	40.67.254.36	TCP	54	63251 → 443 [ACK] Seq=44 Ack=175 Win=512 Len=0
4	0.450089	172.26.106.136	193.137.16.65	DNS	79	Standard query 0xa7cb A elearning.uminho.pt
5	0.454371	193.137.16.65	172.26.106.136	DNS	351	Standard query response 0xa7cb A elearning.uminho.pt A 193.137.9.150 NS dns3.uminho.pt NS ns02.fccn.pt NS dns.uminho.pt NS dns...
6	0.464580	172.26.106.136	193.137.9.150	TCP	66	64996 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
7	0.468051	193.137.9.150	172.26.106.136	TCP	66	80 → 64996 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1250 WS=256 SACK_PERM=1
8	0.468464	172.26.106.136	193.137.9.150	TCP	54	64996 → 80 [ACK] Seq=1 Ack=1 Win=131072 Len=0
9	0.470159	172.26.106.136	193.137.9.150	HTTP	496	GET / HTTP/1.1
10	0.474046	193.137.9.150	172.26.106.136	HTTP	186	HTTP/1.1 301
11	0.474660	193.137.9.150	172.26.106.136	TCP	54	80 → 64996 [FIN, ACK] Seq=133 Ack=443 Win=262400 Len=0
12	0.474768	172.26.106.136	193.137.9.150	TCP	54	64996 → 80 [ACK] Seq=443 Ack=134 Win=131072 Len=0
13	0.474856	172.26.106.136	193.137.9.150	TCP	54	64996 → 80 [FIN, ACK] Seq=443 Ack=134 Win=131072 Len=0
14	0.478202	193.137.9.150	172.26.106.136	TCP	54	80 → 64996 [ACK] Seq=134 Ack=444 Win=262400 Len=0

Figura 1: Captura no *wireshark*

O número de ordem da sequência de bytes capturada é 9.

## 1.2 1. Anote os endereços MAC de origem e de destino da trama capturada.

```
> Frame 1: 97 bytes on wire (776 bits), 97 bytes captured (776 bits) on interface \Device\NPF_{02A1AA6F-9117-4F04-8ACD-98A41E7D99A1}, id 0
  Ethernet II, Src: Microsof_83:33:98 (b8:31:b5:83:33:98), Dst: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
    Destination: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
      Address: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
        ... .. = LG bit: Globally unique address (factory default)
        ... .. = IG bit: Individual address (unicast)
    Source: Microsof_83:33:98 (b8:31:b5:83:33:98)
      Address: Microsof_83:33:98 (b8:31:b5:83:33:98)
        ... .. = LG bit: Globally unique address (factory default)
        ... .. = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
  > Data (83 bytes)
```

Figura 2: Trama capturada no *wireshark*

Endereço de origem: b8:31:b5:83:33:98

Endereço de destino: 00:d0:03:ff:94:00

## 1.3 2. Identifique a que sistemas se referem. Justifique.

O endereço de origem refere-se à interface Ethernet da nossa máquina uma vez que representa o local de onde é enviada a trama. O destino é a interface do router da rede local porque a nossa máquina só reconhece endereços da rede local, logo a trama será encaminhada para o router que tem acesso à rede exterior.

## 1.4 3. Qual o valor hexadecimal do campo Type da trama Ethernet? O que significa?

O campo type possui valor 0x0800, o que significa que a trama encapsula um pacote IPv4.

**1.5 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.**

```
> Frame 9: 496 bytes on wire (3968 bits), 496 bytes captured (3968 bits) on interface \Device\NPF_{02A1...}
> Ethernet II, Src: Microsof_83:33:98 (b8:31:b5:83:33:98), Dst: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
> Internet Protocol Version 4, Src: 172.26.106.136, Dst: 193.137.9.150
> Transmission Control Protocol, Src Port: 64996, Dst Port: 80, Seq: 1, Ack: 1, Len: 442
> Hypertext Transfer Protocol

0000 00 d0 03 ff 94 00 b8 31 b5 83 33 98 08 00 45 00 .....1..3...E-
0010 01 e2 27 10 40 00 80 06 f0 43 ac 1a 6a 88 c1 89 ...'@...C...j...
0020 09 96 fd e4 00 50 cb 39 61 a5 be 85 8b 79 50 18 .....P.9 a...yP-
0030 02 00 ad 49 00 00 47 45 54 20 2f 20 48 54 54 50 ...I..GE / HTTP
0040 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 65 6c 65 61 /1.1..Host: elea
0050 72 6e 69 6e 67 2e 75 6d 69 6e 68 6f 2e 70 74 0d rning.um inho.pt-
0060 0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 .Connect ion: kee
0070 70 2d 61 6c 69 76 65 0d 0a 55 70 67 72 61 64 65 p-alive-Upgrade
0080 2d 49 6e 73 65 63 75 72 65 2d 52 65 71 75 65 73 ~Insecur e-Reques
0090 74 73 3a 20 31 0d 0a 55 73 65 72 2d 41 67 65 6e ts: 1..U ser-Agen
00a0 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 t: Mozil la/5.0 (
00b0 57 69 6e 64 6f 77 73 20 4e 54 20 31 30 2e 30 3b Windows NT 10.0;
00c0 20 57 69 6e 36 34 3b 20 78 36 34 29 20 41 70 70 Win64; x64) App
00d0 6c 65 57 65 62 4b 69 74 2f 35 33 37 2e 33 36 20 leWebKit /537.36
00e0 28 4b 48 54 4d 4c 2c 20 6c 69 6b 65 20 47 65 63 (KHTML, like Gec
00f0 6b 6f 29 20 43 68 72 6f 6d 65 2f 38 37 2e 30 2e ko) Chro me/87.0.
0100 34 32 38 30 2e 36 37 20 53 61 66 61 72 69 2f 35 4280.67 Safari/5
0110 33 37 2e 33 36 20 45 64 67 2f 38 37 2e 30 2e 36 37.36 Ed g/87.0.6
0120 36 34 2e 34 37 0d 0a 41 63 63 65 70 74 3a 20 74 64.47..Accept: t
```

Figura 3: Trama capturada no *wireshark*

São utilizados 52 bytes.  
 $\text{overhead} = 52/496 * 100 = 10,5\%$

**1.6 5. Através de visualização direta ou construindo um filtro específico, verifique se foram detetadas tramas com erros (por verificação do campo FCS (Frame Check Sequence))**

O campo FCS (Frame Check Sequence) não aparece na trama capturada porque as redes com fios (como a ethernet) são muito pouco suscetíveis a erros.

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

Endereço da fonte: 00:d0:03:ff:94:00

Neste caso o endereço fonte corresponde à interface do gateway da rede local, uma vez que estamos perante a mensagem de resposta e só é possível saber endereços da rede local.

**1.8 7. Qual é o endereço MAC do destino? A que sistema corresponde?**

Endereço destino: b8:31:b5:83:33:98

Neste caso o endereço destino passa a ser referente à nossa máquina, porque é esta que vai receber a trama.

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

Ethernet, TCP e IPv4

## 2 Protocolo ARP

**2.1 9. Observe o conteúdo da tabela ARP. Diga o que significa cada uma das colunas.**

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

Interface: 192.168.1.83 --- 0xd
Internet Address      Physical Address      Type
192.168.1.254         10-13-31-72-f2-ec    dynamic
192.168.1.255         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

Interface: 192.168.56.1 --- 0x10
Internet Address      Physical Address      Type
192.168.56.255        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
```

Figura 4: Output do comando arp -a

Primeira coluna: IP address do Host

Segunda coluna: MAC address do Host

Terceira coluna: tipo de encaminhamento

Neste caso cada tabela corresponde a uma interface de rede diferente.

**2.2 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?**

Origem: fc:01:7c:68:a2:65

Destino: ff:ff:ff:ff:ff:ff

O destino é o endereço do broadcast e significa que a mensagem será enviada para todos os hosts da rede local.

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

Tipo da trama: 0x0806

Este campo significa que a mensagem encapsula um frame do tipo ARP.

**2.4 12. Como pode confirmar que se trata efetivamente de um pedido ARP? Identifique que tipo de endereços estão contidos na mensagem ARP? Que conclui? (Se necessário, consulte a RFC do protocolo ARP <http://tools.ietf.org/html/rfc826.html>).**

Pode-se confirma que se trata de um pedido ARP pois o campo opcode é 1 (valor correspondente a *request*). A mensagem ARP contém endereços MAC e IP (*sender mac address*, *sender ip address*, *target mac address* e *target ip address*). Assim, pode-se concluir que o host com ip 172.26.16.43 e mac fc:01:7c:68:a2:65 precisa de saber o *mac address* do host com ip 172.26.254.254 (*target ip address*).

**2.5 13. Explícite que tipo de pedido ou pergunta é feita pelo host de origem?**

É feito o pedido "Who has 172.26.254.254? Tell 172.26.16.43".

Este tipo de pedido significa que foi solicitado o *mac address* correspondente ao host de ip 172.26.254.254 e a resposta terá de ser enviada para a máquina de endereço ip 172.26.16.43.

**2.6 14. Localize a mensagem ARP que é a resposta ao pedido ARP efetuado.**

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

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

Neste caso o valor do opcode é 2 (reply), logo estamos perante uma mensagem de resposta.

A resposta ao pedido ARP está situada no campo *sender mac address* e *sender ip address*.

### 3 ARP Gratuito

**3.1 15. 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?**

```
> Frame 309: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{02A1AA6F-9117-4F04-8ACD-98A41E7D99A1}, id 0
> Ethernet II, Src: Microsof_83:33:98 (b8:31:b5:83:33:98), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  Address Resolution Protocol (ARP Announcement)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    [Is gratuitous: True]
    [Is announcement: True]
    Sender MAC address: Microsof_83:33:98 (b8:31:b5:83:33:98)
    Sender IP address: 172.26.106.136
    Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)
    Target IP address: 172.26.106.136
```

Figura 5: Trama capturada no *wireshark*

O que distingue é a flag *Is gratuitous* que neste caso é true.

**3.2 16. Através da opção tcpdump verifique e compare como flui o tráfego nas diversas interfaces dos vários dispositivos no departamento A (LAN comutada) e no departamento B (LAN partilhada) quando gera tráfego intra-departamento (por exemplo, através do comando ping). Que conclui?**

**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.?**

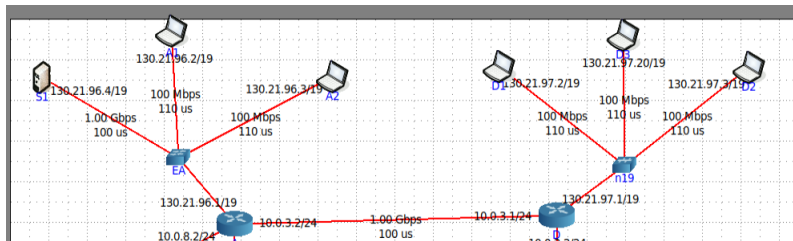


Figura 6: Topologia Core (Departamento A e B)



```

root@B1:/tmp/pycore-42455/1.conf# tcpdump -l
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:45:01.307770 IP 130.21.36.3 > 130.21.36.4: ICMP echo request, id 28, seq 1, len 64
11:45:01.307770 IP 130.21.36.4 > 130.21.36.3: ICMP echo reply, id 28, seq 1, len 64
11:45:02.314881 IP 130.21.36.3 > 130.21.36.4: ICMP echo request, id 28, seq 2, len 64
11:45:02.315004 IP 130.21.36.4 > 130.21.36.3: ICMP echo reply, id 28, seq 2, len 64
11:45:03.151962 IP fe80::e8b1:13ff:fe5a:3e37 > ff02::fb:adns: 0 [2q] PIR (DHCP) _ipps_top.local, PIR (DHCP) _ipps_top.local, (45)
11:45:04.208914 IP fe80::e8b1:13ff:fe5a:3e37 > ff02::fb:adns: 0 [2q] PIR (DHCP) _ipps_top.local, PIR (DHCP) _ipps_top.local, (45)
11:45:04.510842 IP fe80::e8b1:13ff:fe5a:3e37 > ip6-allrouters: ICMP6, router solicitation, length 16
11:45:04.829702 IP fe80::e8b1:13ff:fe5a:3e37 > ip6-allrouters: ICMP6, router solicitation, length 16
11:45:06.410721 HRP Request who-has 130.21.36.3 tell 130.21.36.4, length 28
11:45:06.411322 HRP Request who-has 130.21.36.4 tell 130.21.36.3, length 28
11:45:06.411335 HRP Reply 130.21.36.4 is-at 00:00:00:00:00:00 (oui Ethernet), length 28
11:45:06.411774 HRP Reply 130.21.36.3 is-at 00:00:00:00:00:00 (oui Ethernet), length 28
11:45:08.867012 IP fe80::200:ff:feaa:16 > ip6-allrouters: ICMP6, router solicitation, length 16
11:45:07.428172 IP 130.21.36.1 > 224.0.0.5: OSPFv2, Hello, length 44
11:45:07.428702 IP fe80::200:ff:feaa:16 > ff02::5: OSPFv3, Hello, length 36
16 packets captured
15 packets received by filter
0 packets dropped by kernel
root@B1:/tmp/pycore-42455/1.conf#

root@B2:/tmp/pycore-42455/2.conf# ping -c 2 130.21.36.4
PING 130.21.36.4 (130.21.36.4): 56(84) bytes of data:
64 bytes from 130.21.36.4: icmp_seq=1 ttl=64 time=1.26 ms
--- 130.21.36.4 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1010ms
rtt min/avg/max/ndev = 0.880/0.883/0.907/0.057 ms
root@B2:/tmp/pycore-42455/2.conf#

root@B1:/tmp/pycore-42455/1.conf# tcpdump -l
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:42:47.433177 IP 130.21.36.1 > 224.0.0.5: OSPFv2, Hello, length 44
11:42:55.463770 IP fe80::200:ff:feaa:16 > ff02::5: OSPFv3, Hello, length 36
11:42:47.433889 IP 130.21.36.1 > 224.0.0.5: OSPFv2, Hello, length 44
11:42:47.471437 IP fe80::200:ff:feaa:16 > ff02::5: OSPFv3, Hello, length 36
11:42:55.520162 IP fe80::200:ff:feaa:16 > ip6-allrouters: ICMP6, router solicitation, length 16
11:42:57.458031 IP 130.21.36.1 > 224.0.0.5: OSPFv2, Hello, length 44
11:42:57.472008 IP fe80::200:ff:feaa:16 > ff02::5: OSPFv3, Hello, length 36
11:42:58.476223 IP fe80::200:ff:feaa:16 > ip6-allrouters: ICMP6, router solicitation, length 16
11:43:05.151563 IP fe80::e8b1:13ff:fe5a:3e37 > ff02::fb:adns: 0 [2q] PIR (DHCP) _ipps_top.local, PIR (DHCP) _ipps_top.local, (45)
11:43:04.438620 IP fe80::e8b1:13ff:fe5a:3e37 > ff02::fb:adns: 0 [2q] PIR (DHCP) _ipps_top.local, PIR (DHCP) _ipps_top.local, (45)
11:43:04.510842 IP fe80::e8b1:13ff:fe5a:3e37 > ip6-allrouters: ICMP6, router solicitation, length 16
11:43:06.854545 IP fe80::200:ff:feaa:16 > ip6-allrouters: ICMP6, router solicitation, length 16
11:43:07.428173 IP 130.21.36.1 > 224.0.0.5: OSPFv2, Hello, length 44
11:43:07.428703 IP fe80::200:ff:feaa:16 > ff02::5: OSPFv3, Hello, length 36
14 packets captured
14 packets received by filter
0 packets dropped by kernel
root@B1:/tmp/pycore-42455/1.conf#

```

Figura 7: Departamento A - envio e análise do tráfego

```

root@B1:/tmp/pycore-42455/1.conf# tcpdump -l
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:47:07.930153 IP fe80::200:ff:feaa:16 > ff02::5: OSPFv3, Hello, length 36
11:47:07.930153 IP 130.21.97.1 > 224.0.0.5: OSPFv2, Hello, length 44
11:47:08.330720 IP fe80::200:ff:feaa:17 > ip6-allrouters: ICMP6, router solicitation, length 16
11:47:12.072393 IP 130.21.97.20 > 130.21.97.2: ICMP echo request, id 26, seq 1, length 64
11:47:12.072790 IP 130.21.97.2 > 130.21.97.20: ICMP echo reply, id 26, seq 1, length 64
11:47:13.103066 IP 130.21.97.20 > 130.21.97.2: ICMP echo request, id 26, seq 2, length 64
11:47:13.103023 IP 130.21.97.2 > 130.21.97.20: ICMP echo reply, id 26, seq 2, length 64
11:47:17.994867 IP 130.21.97.1 > 224.0.0.5: OSPFv2, Hello, length 44
11:47:17.994840 IP fe80::200:ff:feaa:16 > ff02::5: OSPFv3, Hello, length 36
9 packets captured
9 packets received by filter
0 packets dropped by kernel
root@B1:/tmp/pycore-42455/1.conf#

root@B2:/tmp/pycore-42455/2.conf# tcpdump -l
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:47:12.072721 IP 130.21.97.20 > 130.21.97.2: ICMP echo request, id 26, seq 1, length 64
11:47:12.072799 IP 130.21.97.2 > 130.21.97.20: ICMP echo reply, id 26, seq 1, length 64
11:47:13.103007 IP 130.21.97.20 > 130.21.97.2: ICMP echo request, id 26, seq 2, length 64
11:47:13.103031 IP 130.21.97.2 > 130.21.97.20: ICMP echo reply, id 26, seq 2, length 64
11:47:17.994888 IP 130.21.97.1 > 224.0.0.5: OSPFv2, Hello, length 44
11:47:17.994842 IP fe80::200:ff:feaa:16 > ff02::5: OSPFv3, Hello, length 36
6 packets captured
6 packets received by filter
0 packets dropped by kernel
root@B2:/tmp/pycore-42455/2.conf#

root@B3:/tmp/pycore-42455/3.conf# ping 130.21.97.2
PING 130.21.97.2 (130.21.97.2): 56(84) bytes of data:
64 bytes from 130.21.97.2: icmp_seq=1 ttl=64 time=0.086 ms
64 bytes from 130.21.97.2: icmp_seq=2 ttl=64 time=0.084 ms
64 bytes from 130.21.97.2: icmp_seq=3 ttl=64 time=0.043 ms
64 bytes from 130.21.97.2: icmp_seq=4 ttl=64 time=0.038 ms
64 bytes from 130.21.97.2: icmp_seq=5 ttl=64 time=0.040 ms
--- 130.21.97.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 408ms
rtt min/avg/max/ndev = 0.038/0.063/0.086/0.020 ms
root@B3:/tmp/pycore-42455/3.conf#

```

Figura 8: Departamento B - envio e análise do tráfego

Quando geramos tráfego intra-departamento através do comando ping no departamento A, vemos que apenas o servidor de destino recebeu os pacotes (Figura 7 e Figura 8). No caso do departamento B, reparamos o contrário, isto é, todas as interfaces recebem os pacotes do tráfego enviado. Isto acontece uma vez que um switch envia diretamente para um host, ao contrário de um hub.

Portanto, com base nos testes realizados, podemos concluir que os switches são uma escolha mais viável pois, conectam cada dispositivo a uma porta do comutador, e cada uma das portas é um domínio de colisão, eliminando assim a ocorrência de colisões.

## **Conclusão**

A realização deste trabalho prático não só nos permitiu consolidar o conhecimento obtido ao longo das aulas teóricas como também clarificou e aprofundou alguns conceitos, tais como : Endereços MAC, Address Resolution Protocol, Tramas Ethernet, Protocolo ARP, e Domínios de colisão.

Ao longo do trabalho tivemos o auxílio de ferramentas como o wireshark e o CORE de forma a capturar e a analisar Tramas Ethernet. Além disso, ainda analisamos a operação do protocolo ARP. Finalmente, usando o CORE tivemos a oportunidade de simular e analisar o envio de pacotes de forma a gerar tráfego intra-departamento numa topologia já criada anteriormente , e comparar a viabilidade de um switch e de um hub, observando o funcionamento dos domínios de colisão.

Em suma, recapitulamos o Link Layer, não só consolidando os seus principais conceitos como também fazendo vários testes , capturas e análises com as ferramentas que se encontram ao nosso dispor, de forma a cumprir os objetivos traçados ao longo do enunciado.