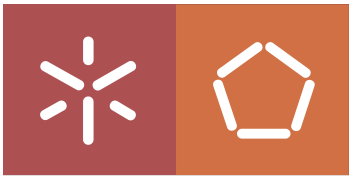


Departamento de Informática

Licenciatura em Engenharia Informática

Redes de Computadores

Universidade do Minho
Escola de Engenharia

Trabalho Prático n.º 3

Grupo n.º 18

Artur Carneiro Neto de Nóbrega Luís, n.º A95414
Hugo Ricardo Macedo Gomes, n.º A96842
Orlando José da Cunha Palmeira, n.º A97755

Braga, abril de 2022

Índice

1 - Captura e análise de Tramas Ethernet

1.1 - Alínea 1)

```
> Destination: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
> Source: IntelCor_51:8f:51 (34:cf:f6:51:8f:51)
```

Figura 1 - Endereços MAC origem e destino

1.2 - Alínea 2)

O sistema de origem é a máquina nativa e o sistema de destino é o *router* ao qual máquina nativa a máquina nativa está conectada.
Para confirmar esta afirmação, procedemos à consulta do *default gateway* da máquina nativa bem como o endereço mac associado ao *default gateway.*

```
IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0   172.26.254.254    172.26.25.248     35
```

Figura 2 - Consulta do *default gateway*

```
PS C:\Users\orlan> arp -a

Interface: 172.26.25.248 --- 0x2
  Internet Address      Physical Address      Type
  172.26.254.254        00-d0-03-ff-94-00     dynamic
```

Figura 3 - Consulta do endereço MAC associado ao *default gateway*

1.3 - Alínea 3)

Type: `IPv4 (0x0800)` (retirado do *output* do wireshark)
Este campo indica o protocolo da camada superior encapsulado no *frame Ethernet*.(fonte: https://www.ciscopress.com/articles/article.asp?p=3089352&seqNum=4)

1.4 - Alínea 4)

```
Internet Protocol Version 4, Src: 172.26.25.248, Dst: 193.137.9.150
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
```

Figura 4 - Tamanho do cabeçalho IP (dado pelo Wireshark)

```
Transmission Control Protocol, Src Port: 51405, Dst Port: 443, Seq: 644, Ack: 6171, Len: 697
    Source Port: 51405
    Destination Port: 443
    [Stream index: 0]
    [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 697]
    Sequence Number: 644    (relative sequence number)
    Sequence Number (raw): 3002136530
    [Next Sequence Number: 1341    (relative sequence number)]
    Acknowledgment Number: 6171    (relative ack number)
    Acknowledgment number (raw): 1793791068
    0101 .... = Header Length: 20 bytes (5)
```

Figura 5 - Tamanho do cabeçalho TCP (dado pelo Wireshark)

O tamanho de um cabeçalho Ethernet é de 14 octetos (fonte: https://www.sciencedirect.com/topics/computer-science/ethernet-header.

Deste modo, o n.º de bytes utilizados no encapsulamento protocolar é de 20 + 20 + 14 = 54 bytes.

```
Frame 20: 751 bytes on wire (6008 bits), 751 bytes captured (6008 bits) on inte
  > Interface id: 0 (\Device\NPF_{00C0F88F-BB2D-42FE-9ED0-57B7E7A907CD})
    Encapsulation type: Ethernet (1)
    Arrival Time: Apr 20, 2022 15:50:03.447144000 Hora de Verão de GMT
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1650466203.447144000 seconds
    [Time delta from previous captured frame: 0.000453000 seconds]
    [Time delta from previous displayed frame: 0.000453000 seconds]
    [Time since reference or first frame: 0.085141000 seconds]
    Frame Number: 20
    Frame Length: 751 bytes (6008 bits)
```

Figura 6 - Tamanho do pacote (dado pelo Wireshark)

O tamanho total do pacote é de 751 bytes (conforme destacado na figura 6). Assim, a sobrecarga imposta pela pilha protocolar é $\frac{54}{751} \times 100 \simeq 7.19\%$.

1.5 - Alínea 5)

```
Ethernet II, Src: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00),
  > Destination: IntelCor_51:8f:51 (34:cf:f6:51:8f:51)
  > Source: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00)
    Type: IPv4 (0x0800)
```

Figura 7 - Endereços MAC origem e destino do pacote vindo do servidor

O endereço MAC da fonte é: `00:d0:03:ff:94:00` e corresponde ao *router* ao qual a máquina nativa está conectada.

1.6 - Alínea 6)

O endereço MAC de destino é: `34:cf:f6:51:8f:51` e corresponde à máquina nativa.

1.7 - Alínea 7)

```
Ethernet II, Src: ComdaEnt_ff:94:00 (00:d0:03:ff:94:00), Dst: IntelCor_51:8f:51 (34:cf:f6:51:8f:51)
Internet Protocol Version 4, Src: 193.137.9.150, Dst: 172.26.25.248
Transmission Control Protocol, Src Port: 443, Dst Port: 51405, Seq: 6171, Ack: 1341, Len: 851
Transport Layer Security
```

Figura 8 - Protocolos contidos na trama (dado pelo Wireshark)

Como podemos ver na figura 8, os protocolos contidos na trama recebida são: Ethernet, IPv4, TCP e TLS.

2 - Protocolo ARP

2.1 - Alínea 8)

```
PS C:\Users\orlan> arp -a

Interface: 192.168.1.6 --- 0x2
  Internet Address      Physical Address      Type
  192.168.1.1           08-b0-55-0c-64-ae     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 --- 0x11
  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 9 - Tabela ARP

A primeira coluna da tabela apresenta os endereços IP das interfaces, a segunda apresenta os endereços físicos (MAC) associados a cada uma das interfaces e, por fim, a terceira coluna indica o tipo de encaminhamento (estático ou dinâmico).

2.2 - Alínea 9)

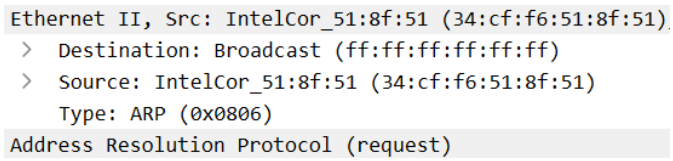


Figura 10 - Endereços origem e destino do *ARP Request*

Endereço origem: `34:cf:f6:51:8f:51` (máquina nativa)
Endereço destino: `ff:ff:ff:ff:ff:ff` (*broadcast*)
Neste *ARP Request*, o endereço destino é de broadcast uma vez que o objetivo do envio deste pacote é o *host* de origem saber qual é o endereço físico (MAC) de um certo *host*. Ora, como a cache ARP havia sido limpa antes e após o início da captura no Wireshark, o *host* de origem tem de enviar o *ARP Request* a todos os *hosts* (*broadcast*) da subrede para obter essa informação.

2.3 - Alínea 10)

Como se pode ver na figura 10, o valor do campo 'tipo' é `0x0806` e indica que o protocolo da camada superior encapsulado neste *frame* Ethernet é ARP.

2.4 - Alínea 11)

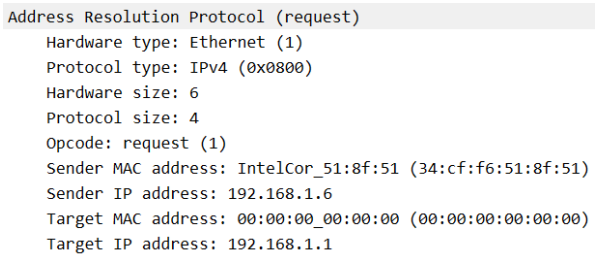


Figura 11 - Conteúdo do cabeçalho ARP

Podemos verificar que se trata de um pedido ARP uma vez que o valor do *opcode* é 1.
Os tipos de endereços presentes no pedido ARP são IP e MAC.

2.5 - Alínea 12)

Neste *ARP Request*, o *host* de origem "pergunta" a todos os *hosts* (*broadcast*) da subrede qual deles é que possui um certo endereço IP mencionado no pedido e recebe como resposta (do dispositivo com o IP pedido) um pacote ARP em que tem mencionado o endereço MAC pretendido.

2.6 - Alínea 13)

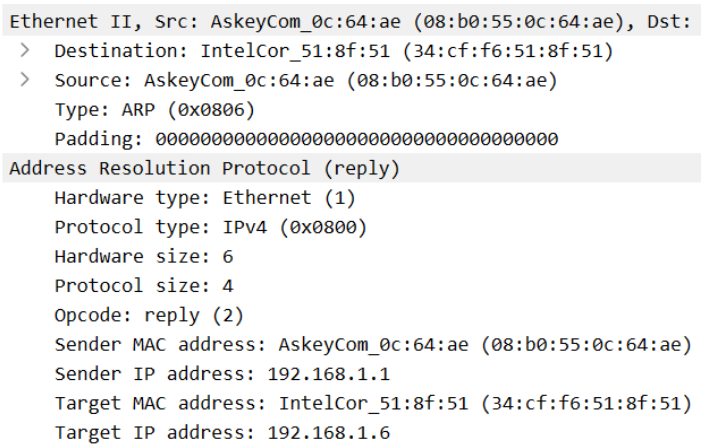


Figura 12 - Alguns dados da mensagem de resposta ao *ARP Request*

2.6.1 - Alínea 13.a)

Como podemos ver na figura 12, o valor do *opcode* é 2. Este campo especifica a natureza da mensagem, neste caso é uma resposta. (fonte: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipaddr_arp/configuration/15-mt/arp-15-mt-book/arp-config-arp.html)

2.6.2 - Alínea 13.b)

A resposta ao pedido ARP (endereço MAC) está no campo *Sender MAC address* (figura 12).

2.7 - Alínea 14)

Considerando que existem duas subredes (sendo 1 a subrede do *host* de origem (máquina nativa) e 2 a subrede do *host* de destino), podemos construir a seguinte tabela:

Ordem	Subrede	Origem	Destino	Protocolo	Conteúdo relevante
1	1	MAC_MáquinaNativa	*Broadcast*	ARP	**MAC_SRC**: MAC_MáquinaNativa **IP_SRC**: IP_MáquinaNativa **MAC_DEST**: ??:??:??:??:??:?? **IP_DEST**: IP_InterfaceRouterSubrede1
2	1	MAC_InterfaceRouter Subrede1	MAC_Máquina Nativa	ARP	**MAC_SRC**: MAC_InterfaceRouterSubrede1 **IP_SRC**: IP_InterfaceRouterSubrede1 **MAC_DEST**: MAC_MáquinaNativa **IP_DEST**: IP_MáquinaNativa
3	1	IP_MáquinaNativa	IP_HostDestino	ICMP	**MAC_SRC**: MAC_MáquinaNativa **IP_SRC**: IP_MáquinaNativa **MAC_DEST**: MAC_InterfaceRouterSubrede1 **IP_DEST**: IP_HostDestino
4	2	MAC_InterfaceRouter Subrede2	*Broadcast*	ARP	**MAC_SRC**: MAC_InterfaceRouterSubrede2 **IP_SRC**: IP_InterfaceRouterSubrede2 **MAC_DEST**: ??:??:??:??:??:?? **IP_DEST**: IP_HostDestino
5	2	MAC_HostDestino	MAC_Interface RouterSubrede2	ARP	**MAC_SRC**: MAC_HostDestino **IP_SRC**: IP_HostDestino **MAC_DEST**: MAC_InterfaceRouterSubrede2 **IP_DEST**: IP_InterfaceRouterSubrede2
6	2	IP_HostDestino	IP_Máquina Nativa	ICMP	**MAC_SRC**: MAC_HostDestino **IP_SRC**: IP_HostDestino **MAC_DEST**: MAC_InterfaceRouterSubrede2 **IP_DEST**: IP_MáquinaNativa

3 - Domínios de Colisão

3.1 - Alínea 15)

Substituindo o *switch* da subrede do departamento A por um *hub*, obtém-se a seguinte topologia:

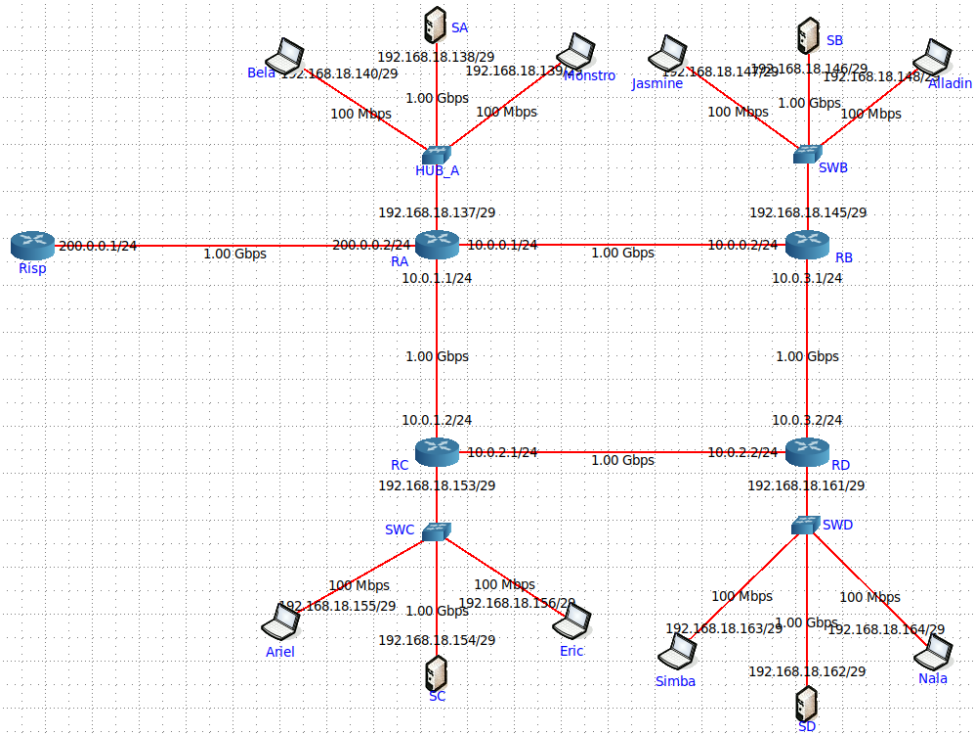


Figura 13 - Nova topologia do TP2 com o *hub* no departamento A

Ao executar os pings do pc Bela para o pc Monstro e do pc Jasmine para o pc Alladin simultaneamente com o tcpdump ativado nos servidores SA e SB, obtemos os seguintes resultados:

```
root@SA:/tmp/pycore.42423/SA.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
^C17:57:41.359921 IP 192.168.18.137 > 224.0.0.5: OSPFv2, Hello, length 44
17:57:42.681174 IP 192.168.18.140 > 192.168.18.139: ICMP echo request, id 31, se
q 1, length 64
17:57:42.681487 IP 192.168.18.139 > 192.168.18.140: ICMP echo reply, id 31, seq
1, length 64
17:57:43.360552 IP 192.168.18.137 > 224.0.0.5: OSPFv2, Hello, length 44
17:57:43.699108 IP 192.168.18.140 > 192.168.18.139: ICMP echo request, id 31, se
q 2, length 64
17:57:43.699243 IP 192.168.18.139 > 192.168.18.140: ICMP echo reply, id 31, seq
2, length 64
17:57:44.723173 IP 192.168.18.140 > 192.168.18.139: ICMP echo request, id 31, se
q 3, length 64
17:57:44.723303 IP 192.168.18.139 > 192.168.18.140: ICMP echo reply, id 31, seq
3, length 64
17:57:45.360872 IP 192.168.18.137 > 224.0.0.5: OSPFv2, Hello, length 44
17:57:45.420383 IP6 fe80::200:ff:feaa:16 > ff02::5: OSPFv3, Hello, length 36
17:57:45.747032 IP 192.168.18.140 > 192.168.18.139: ICMP echo request, id 31, se
q 4, length 64
17:57:45.747136 IP 192.168.18.139 > 192.168.18.140: ICMP echo reply, id 31, seq
4, length 64
17:57:46.772033 IP 192.168.18.140 > 192.168.18.139: ICMP echo request, id 31, se
q 5, length 64
17:57:46.772163 IP 192.168.18.139 > 192.168.18.140: ICMP echo reply, id 31, seq
5, length 64

14 packets captured
14 packets received by filter
0 packets dropped by kernel
```

Figura 14 - Resultado do tcpdump em SA

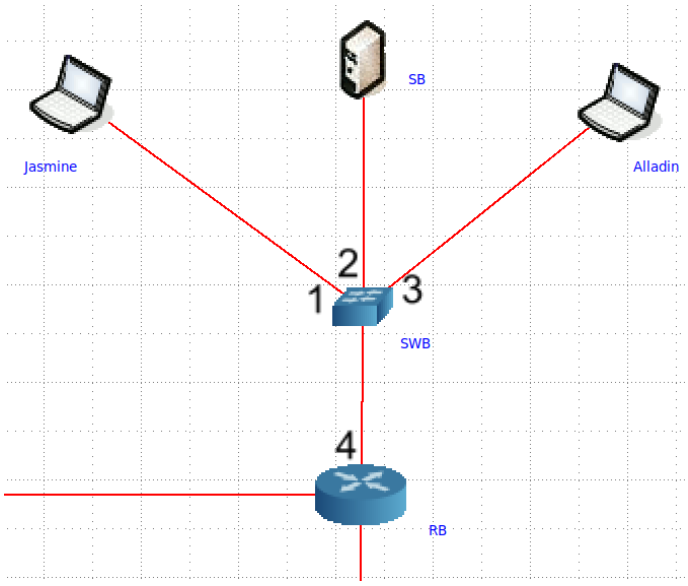
```
root@SB:/tmp/pycore.42423/SB.conf# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C17:59:29.655060 IP 192.168.18.145 > 224.0.0.5: OSPFv2, Hello, length 44
17:59:31.655521 IP 192.168.18.145 > 224.0.0.5: OSPFv2, Hello, length 44
17:59:33.655833 IP 192.168.18.145 > 224.0.0.5: OSPFv2, Hello, length 44
17:59:35.655771 IP 192.168.18.145 > 224.0.0.5: OSPFv2, Hello, length 44
17:59:35.700628 IP6 fe80::200:ff:feaa:8 > ff02::5: OSPFv3, Hello, length 36
17:59:37.656925 IP 192.168.18.145 > 224.0.0.5: OSPFv2, Hello, length 44

6 packets captured
6 packets received by filter
0 packets dropped by kernel
```

Figura 15 - Resultado do tcpdump em SB

Pelos resultados obtidos (figuras 14 e 15), podemos verificar que um *hub* envia todos os pacotes que recebe para todas as interfaces da subrede enquanto que o *switch* faz isso apenas uma vez quando a sua tabela não está preenchida, ou seja, o *switch* tem capacidade de "aprendizagem" enquanto que o *hub* não. É por este motivo que o servidor SA consegue captar pacotes que estão a ser enviados do pc Bela para o pc Monstro.
Atualmente, como todas as redes são *full-duplex, não* haverá grande problema na comunicação em geral quando se usa um *hub* (tirando o facto de haver mais tráfego na rede). Porém, se as redes fossem *half-duplex*, haveria colisões que gerariam atrasos na comunicação uma vez que os dispositivos teriam de esperar um certo tempo para reenviar os pacotes na ocorrência de uma colisão.

3.2 - Alínea 16)



MAC Address	Interface	TTL
MAC_Jasmine	1	----------
MAC_SB	2	----------
MAC_Alladin	3	----------
MAC_RB	4	----------

Figura 16 - Subrede do departamento B com a respetiva tabela do *switch*

4 - Conclusões

Com a realização deste trabalho, ficamos a perceber melhor como analisar tramas *Ethernet* e como ter noção da sobrecarga que o encapsulamento protocolar impõe no tamanho de um pacote.
Também aplicámos na prática maneiras de alterar as tabelas ARP dos nossos dispositivos para podermos visualizar o funcionamento do protocolo ARP e, desse modo, podermos analisar esses pacotes.
Finalmente, pudemos ver as diferenças na utilização de *switches* e *hubs* nas subredes e quais as implicações da utilização desses dispositivos.