Universidade do Minho

Licenciatura em Engenharia Informática

Redes de Computadores

Grupo 135

TP3: *Link Layer*: Redes *Ethernet* e Protocolo ARP

Joana Alves (A93290)

João Machado (A89510)

Rui Armada (A90468)

Abril 2022

Questões e Respostas

1 Questão 3

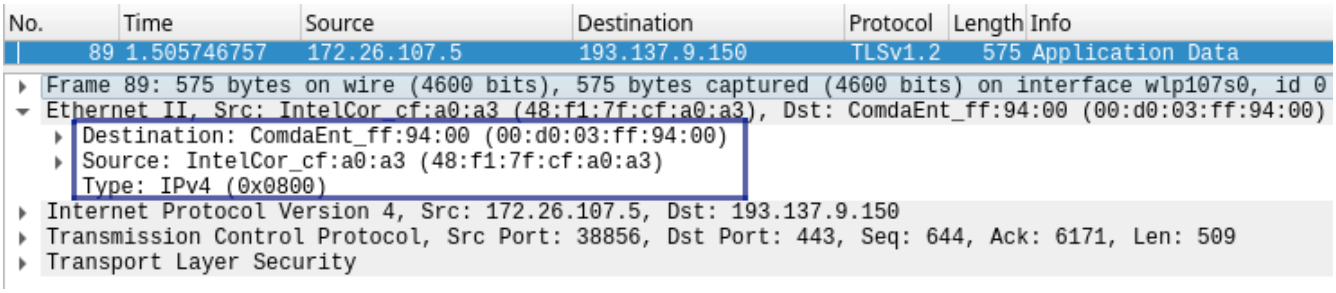


Figura 1: Captura da Trama *Ethernet* que contém mensagem de acesso.

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

Estes endereços podem ser consultados no campo *Source* e *Destination*, respetivamente, no cabeçalho da trama *Ethernet*, tal como se demonstra com a Figura 1:

- MAC Origem: **48:f1:7f:cf:a0:a3**
- MAC Destino: **00:d0:03:ff:94:00**

b. Identifique a que sistemas se referem. Justifique.

O endereço MAC de origem refere-se à máquina nativa utilizada nesta questão, tendo esta o endereço IP 172.26.107.5. No caso do endereço MAC destino, este refere-se ao sistema destino do próximo salto na rede, isto é, ao *router* de acesso ou *default gateway*, uma vez que, contrariamente ao nível de rede (protocolo IP), o nível de ligação lógica vai recalculando **salto-a-salto** o endereço MAC destino contido na trama até chegar ao endereço coincidente com o endereço IP destino, neste caso, 193.137.9.150.

c. Qual o valor hexadecimal do campo *Type* da trama *Ethernet*? O que significa?

O campo *Type* tem como valor **0x0800**, que representa o protocolo *IPv4*, ou seja, este campo indica qual o protocolo que está a ser encapsulado pela trama.

d. Quantos *bytes* são usados no encapsulamento protocolar, i.e. desde o início da trama até ao início dos dados do nível aplicacional (Application Data Protocol: http-over-tls)? Calcule e indique, em percentagem, a sobrecarga (*overhead*) introduzida pela pilha protocolar

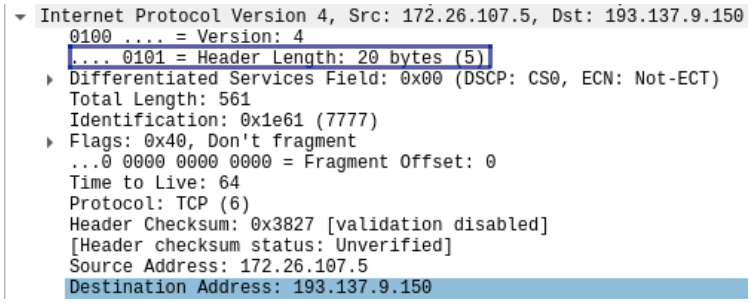


Figura 2: Protocolo IP encapsulado.

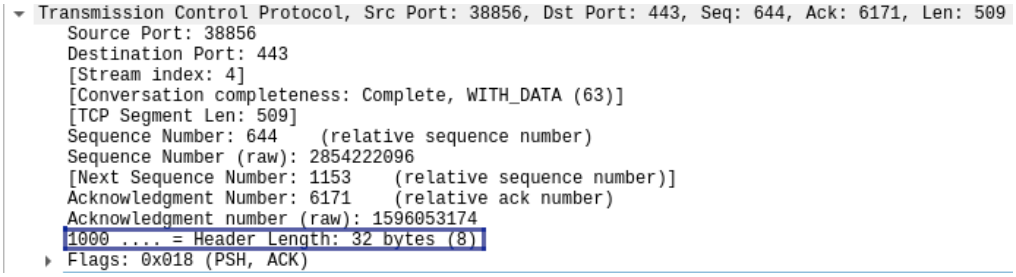


Figura 3: Protocolo TCP encapsulado.

Neste caso são vários os protocolos encapsulados pela trama *Ethernet*, como, por exemplo, o protocolo IP (nível de rede) e TCP (nível de transporte). Assim, conseguimos somar o tamanho de todos os cabeçalhos (*Ethernet*, IP, TCP) utilizados para encapsular os dados do nível aplicacional (*Application Data Protocol*), obtendo o seguinte resultado:

$$14\,(Ethernet) + 20\,(IP) + 32\,(TCP) = \mathbf{66}\ bytes$$

Pelos cálculos, obtivemos um total de 66 *bytes* de encapsulamento protocolar. Relativamente à **sobrecarga** (*overhead*), esta terá o valor de:

$$\frac{66\,(headers)}{575\,(tamanho\,total)*} = \mathbf{11{,}5}\,\%$$

*** NOTA:** O tamanho total da trama com a inclusão dos cabeçalhos foi verificada na Figura 1 na primeira linha referente à informação do *Frame*.

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

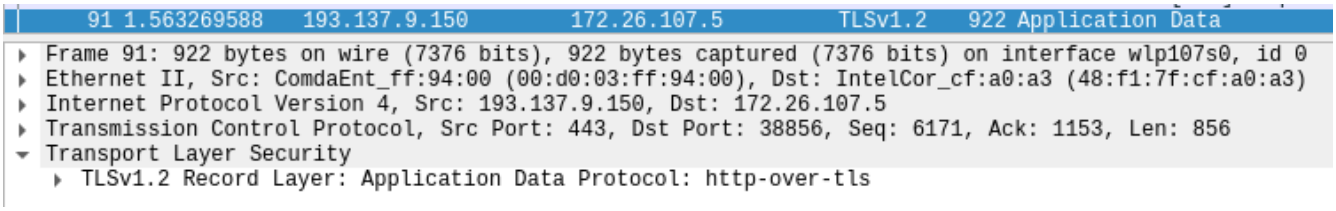


Figura 4: Captura da Trama *Ethernet* de resposta.

O endereço *Ethernet* da fonte é **00:d0:03:94:00** que corresponde ao sistema que na trama capturada nas alíneas anteriores estava no endereço MAC destino, ou seja, o *router* de acesso ou *default gateway* (último salto na rede até à máquina nativa).

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

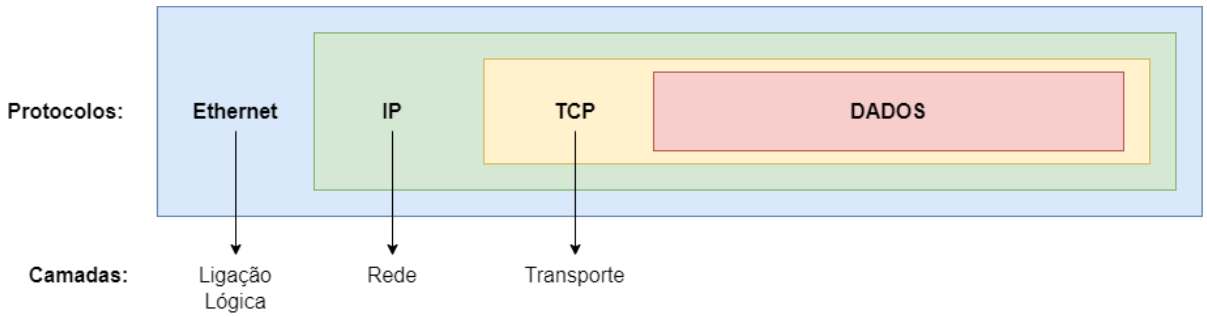
O endereço MAC destino é **48:f1:7f:cf:a0:a3** e corresponde à máquina nativa utilizada para responder a esta questão. Podemos confirmar pela análise à trama utilizada nas alíneas anteriores, pois este endereço corresponde ao endereço MAC origem da mesma.

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

Segundo o modelo OSI, existem sete camadas protocolares, onde podemos denotar a independência e modularidade entre as mesmas. Isto é, cada camada é desenvolvida de forma independente das outras, tendo responsabilidades e funcionalidades distintas, fornecendo serviços à camada diretamente acima e utilizando serviços da camada diretamente abaixo.

Assim, cada nível protocolar utiliza uma PDU (*Protocol Data Unit*) única com informações pertinentes que apenas consegue ser interpretada e manipulada pelas camadas homólogas. Desta forma, quando uma camada utiliza os serviços da camada de baixo, a informação que é transportada para o nível inferior vai ser encapsulada pelo mesmo, ou seja, por exemplo, a camada A envia os seus dados para a camada B (nível inferior) em formato *PDU-A*, que, por sua vez, são encapsulados pela camanda B para *PDU-B*.

Em consequência, conforme os dados vão sendo enviados para as camadas inferiores, existe um encapsulamento protocolar sucessivo até ser atingido o nível mais baixo. Como tal, no caso em estudo, podemos denotar os vários protocolos encapsulados:



2 Questão 4

a. Observe o conteúdo da tabela ARP. Diga o que significa cada uma das colunas.

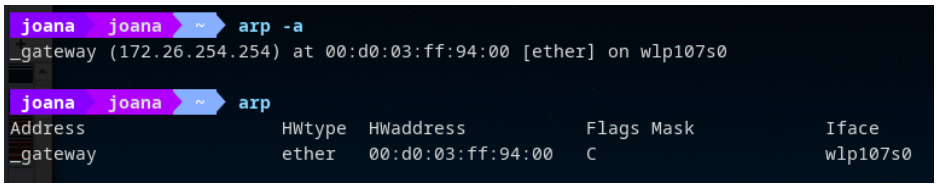


Figura 5: Tabela ARP.

Uma tabela ARP é um método de armazenamento de informações descobertas através do protocolo ARP. É usada para registar os pares correspondentes de endereços MAC e endereços IP de dispositivos conectados a uma rede. Cada dispositivo conectado possui a sua própria tabela ARP, que, tal como referido acima, é responsável por armazenar os pares de endereços com os quais um determinado dispositivo já comunicou. Assim, apresentamos uma descrição das várias colunas presentes na tabela ARP.

Coluna	Descrição
Address	endereço IP destino na rede
HWtype	tipo de *hardware*
HWaddress	endereço MAC do *hardware*
Flags*	informação sobre a entrada
Mask	máscara a aplicar ao endereço IP
Iface	interface de saída

* Neste caso, o valor da coluna é 'C'. Este tipo de entrada é visto quando as entradas são dinamicamente aprendidas pelo protocolo ARP.

b. 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?

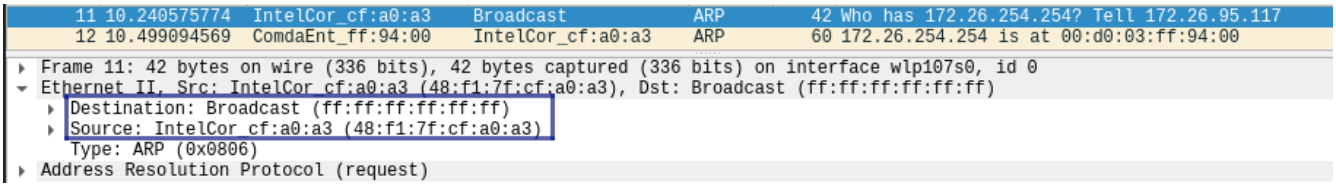


Figura 6: Captura da Trama *Ethernet* com pedido ARP.

- MAC Origem: **48:f1:7f:cf:a0:a3**
- MAC Destino: **ff:ff:ff:ff:ff:ff**

A utilização do endereço MAC destino com o valor dos *bits* todos a 1, é uma particularidade do protocolo ARP, isto é, como apagamos a *cache* da tabela ARP, esta não tem qualquer informação de tradução e correspondência entre endereços IP e endereços MAC. Assim, quando, como neste

caso, queremos enviar algum pacote para outro dispositivo, temos de primeiro descobrir o seu endereço MAC. Este problema é solucionado colocando o endereço MAC destino com todos os *bits* a 1, representando uma mensagem em ***broadcast*** (contendo o endereço IP do dispositivo alvo).

O modo de funcionamento pode ser caracterizado como uma difusão do pedido de *broadcast* do nosso dispositivo por todos os nodos da mesma rede local até encontrar o aparelho que contém o mesmo endereço IP que o endereço IP destino do pedido. De seguida, o aparelho destino (alvo) envia uma resposta (ARP *reply*) à nossa máquina contendo o seu endereço MAC.

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

O valor do campo *type* do cabeçalho da trama *Ethernet* é **0x0806** que indica que o protocolo encapsulado pela trama é o protocolo ARP.

d. Como pode confirmar que se trata efetivamente de um pedido ARP? Identifique que tipo de endereços estão contidos na mensagem ARP? Que conclui?

```
    11 10.240575774  IntelCor_cf:a0:a3    Broadcast           ARP         42 Who has 172.26.254.254? Tell 172.26.95.117
    12 10.499094569  ComdaEnt_ff:94:00    IntelCor_cf:a0:a3   ARP         60 172.26.254.254 is at 00:d0:03:ff:94:00

▸ Frame 11: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface wlp107s0, id 0
▾ Ethernet II, Src: IntelCor_cf:a0:a3 (48:f1:7f:cf:a0:a3), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  ▸ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  ▸ Source: IntelCor_cf:a0:a3 (48:f1:7f:cf:a0:a3)
    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: IntelCor_cf:a0:a3 (48:f1:7f:cf:a0:a3)
    Sender IP address: 172.26.95.117
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 172.26.254.254
```

Figura 7: Conteúdo ARP.

Podemos confirmar que se trata de um pedido ARP pela análise do conteúdo da mensagem ARP, nomeadamente no valor do campo ***Opcode***. Este campo indica que tipo de mensagem ARP estamos a tratar, tendo, neste caso, o valor 1 que corresponde a um ***request***.

Os endereços contidos na mensagem ARP são endereços MAC e IP dos sistemas de origem e destino, ou seja, para cada sistema está indicado o seu endereço IP e correspondente endereço MAC. No entanto, como podemos ver pela Figura 7, esta possui o valor do endereço MAC destino (*Target MAC Address*) a zeros. Isto acontece porque, como referido anteriormente, o endereço MAC destino é uma incógnita, sendo, exatamente, o que o protocolo ARP está a tentar resolver.

e. Explicite que tipo de pedido ou pergunta é feita pelo host de origem.

O *host* de origem envia um pedido (ARP *request*) a todos os nodos presentes na mesma LAN com o objetivo de obter uma resposta (ARP *reply*) do sistema cujo endereço IP corresponde ao endereço IP destino presente na mensagem enviada, contendo esta, também, o endereço MAC do sistema alvo.

f. Localize a mensagem ARP que é a resposta ao pedido ARP efetuado. (1) Qual o valor do campo ARP *opcode*? O que especifica? (2) Em que campo da mensagem ARP está a resposta ao pedido ARP?

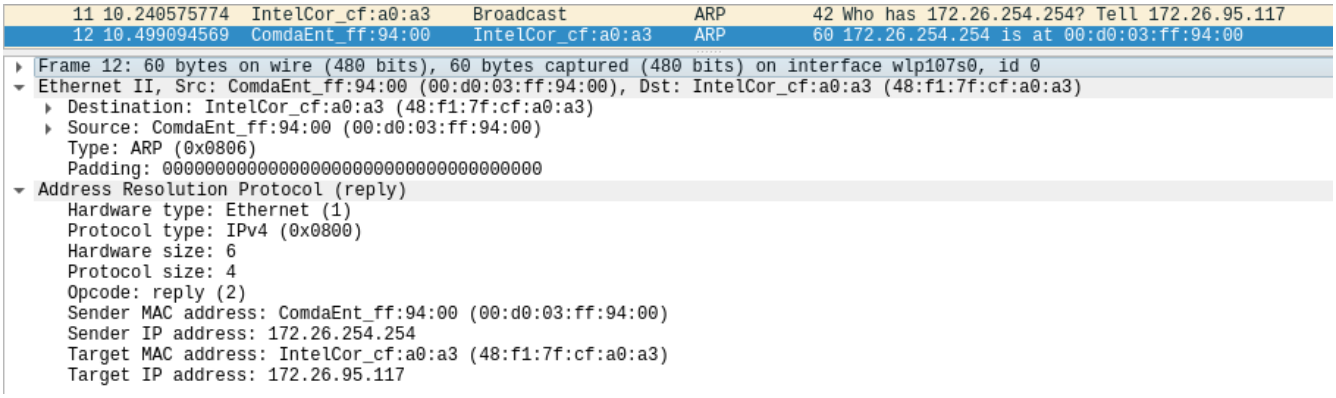
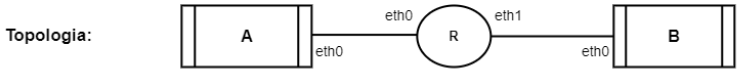


Figura 8: Captura da Trama *Ethernet* com resposta ARP.

(1) O valor do campo *Opcode* é 2 que corresponde a uma mensagem do tipo ***ARP reply***, ou seja, é uma mensagem de resposta a um pedido (*request*) ARP.

(2) O conteúdo da resposta ao pedido ARP está presente no campo ***Sender MAC Address***, onde podemos denotar a introdução de um endereço MAC válido, contrariamente ao presente no campo *Target MAC Address* do ARP *request* analisado anteriormente.

g. Na situação em que efetua um ping a outro host, assuma que este está diretamente ligado ao mesmo router, mas noutra subrede, e que todas as tabelas ARP se encontram inicialmente vazias. Esboce um diagrama em que indique claramente, e de forma cronológica, todas as mensagens ARP e ICMP trocadas, até à recepção da resposta ICMP do host destino.



Tabelas A , B e R

Instante 0

Address	HWtype	HWaddress	Flags	Mask	Iface
-	-	-	-	-	-

Instante 1

Tabela A

Address	HWtype	HWaddress	Flags	Mask	Iface
IP R	ether	MAC R	C	-	eth0

Tabela R

Address	HWtype	HWaddress	Flags	Mask	Iface
IP A	ether	MAC A	C	-	eth0

Tabela B

Address	HWtype	HWaddress	Flags	Mask	Iface
-	-	-	-	-	-

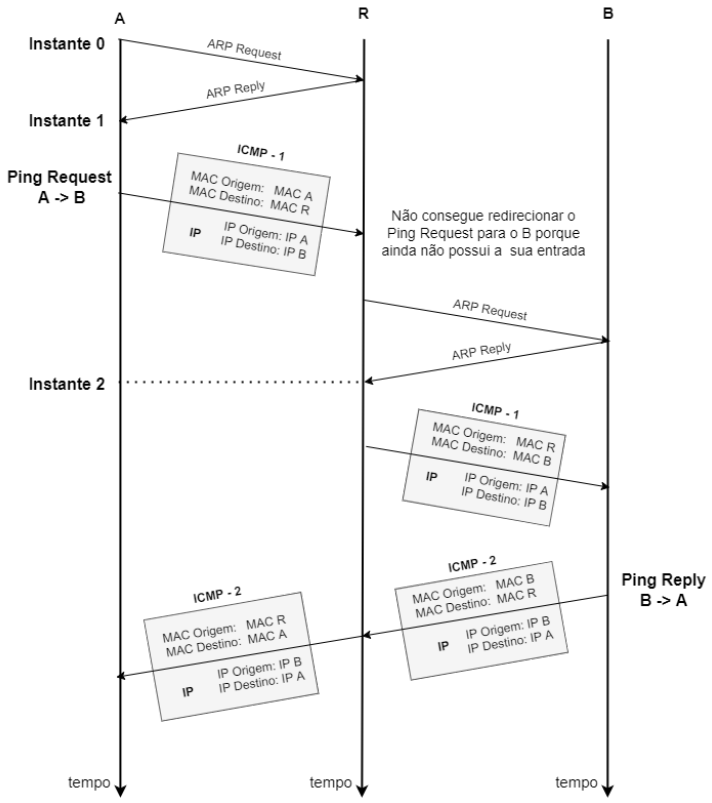
Instante 2

Tabelas A e B

Address	HWtype	HWaddress	Flags	Mask	Iface
IP R	ether	MAC R	C	-	eth0

Tabela R

Address	HWtype	HWaddress	Flags	Mask	Iface
IP B	ether	MAC B	C	-	eth1
IP A	ether	MAC A	C	-	eth0



3 Questão 5

a. Através da opção tcpdump verifique e compare como flui o tráfego nas diversas interfaces do dispositivo de interligação no departamento A (LAN partilhada) e no departamento B (LAN comutada) quando se gera tráfego intra-departamento (por exemplo, fazendo ping IPaddr da Bela para Monstro, da Jasmine para o Alladin, etc.) Que conclui?

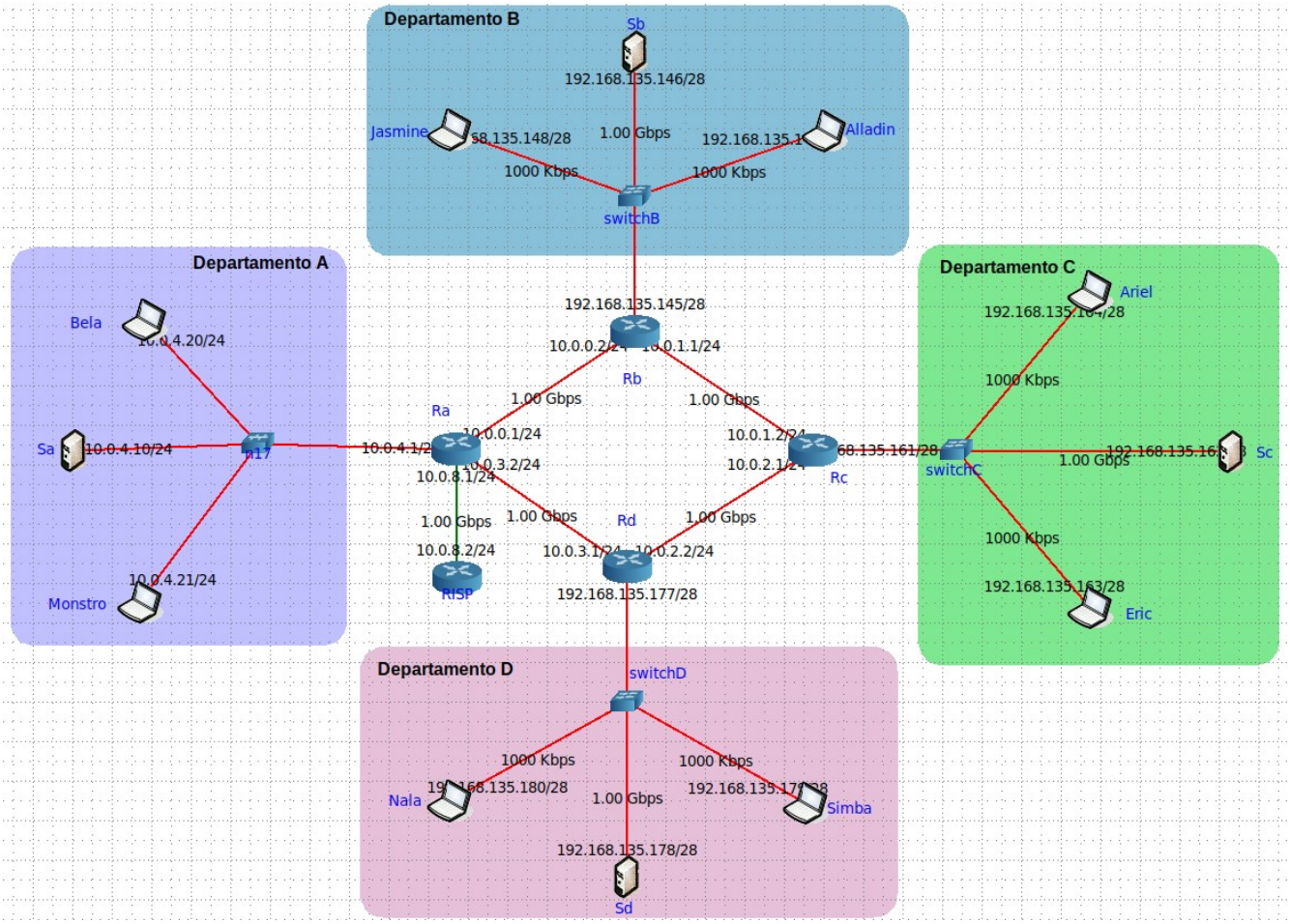


Figura 9: Topologia com o *hub* adicionado.

Como podemos ver pela Figura 9, os endereços IP dos dispositivos Bela, Monstro e servidor Sa foram alterados em consequência da inserção do *hub* e remoção do *switch*. No entanto, o grupo optou por não voltar a alterar os endereços para os obtidos com o exercício de *subnetting* do trabalho prático anterior, uma vez que não não iria alterar o comportamento esperado neste exercício em particular. Os testes foram realizados da seguinte forma: executou-se um *ping* de um *host* A para outro *host* B e executou-se o comando *tcpdump* num *host* C não envolvido na comunicação do *ping*. Assim, no **departamento A** executamos o comando *ping* do *host* Monstro para Bela e o comando *tcpdump* no servidor Sa. No caso do **departamento B**, executamos o comando *ping* do *host* Alladin para Jasmine e o comando *tcpdump* no servidor Sb. Por fim, obtivemos os seguintes resultados com os vários comandos:

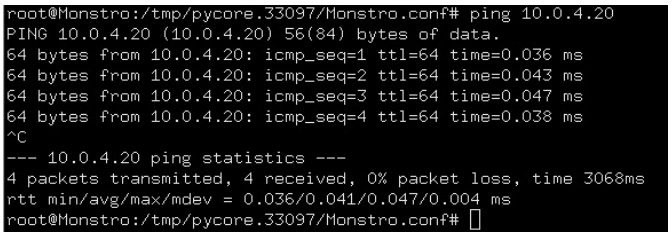


Figura 10: Ping do *host* Monstro para Bela.

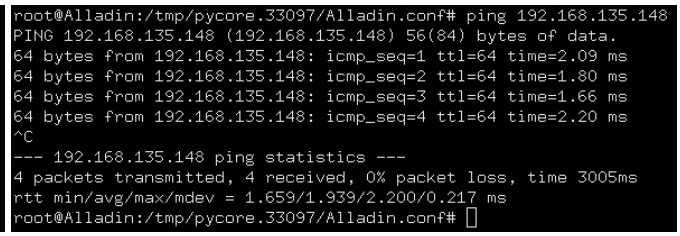


Figura 11: Ping do *host* Alladin para Jasmine.

Figura 12: Comando *tcpdump* no servidor Sa (LAN partilhada).

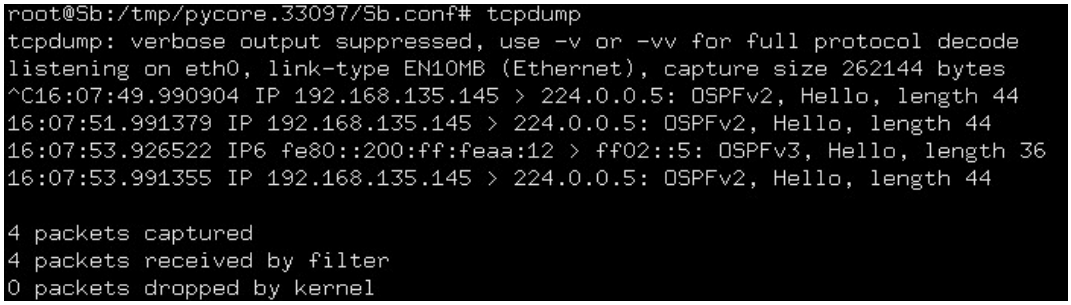


Figura 13: Comando *tcpdump* no servidor Sb (LAN comutada).

Existem várias diferenças entre *hubs* e *switches*, sendo a mais distinta a camada protocolar onde operam, isto é, o *hub* opera ao nível físico enquanto que os *switches* operam na camada de ligação lógica. Para além disto, os ***hubs*** são dispositivos que repetem o sinal que chega através de uma porta de entrada para todas as outras portas, isto é, difundem o sinal por todas as interfaces que possuem. No que toca aos ***switches***, estes, com a ajuda de uma tabela de *switching* (comutação), comutam as tramas para a interface de saída apropriada. No entanto, caso se depare com uma trama que não tem um endereço que conste na tabela, este difunde a trama por todas as suas interfaces, obtendo, neste caso em específico, um comportamento semelhante ao *hub*.

Assim, e tendo estas distinções em mente, conseguimos de imediato notar diferenças no resultado do comando *tcpdump* nas duas situações. Na **LAN partilhada**, o terceiro *host* não envolvido diretamente na comunicação também recebeu os pacotes, provando assim a particularidade dos *hubs* no que toca à difusão das comunicações por todas as portas dos mesmos. No caso da **LAN comutada**, denotamos que o resultado da captura é vazio no que toca a pacotes relacionados com a comunicação derivada do comando *ping*, provando assim a diferença de comportamento do *switch* com o *hub*, pois o tráfego é comutado para as interfaces devidas, não sendo difundido pela rede.

b. Construa manualmente a tabela de comutação do switch do Departamento B, atribuindo números de porta à sua escolha.

Para obtermos os endereços MAC dos vários sistemas diretamente ligados ao *switch*, executamos o comando ***ifconfig -a***. Assim, obtivemos os seguintes endereços, notando que no *router* o endereço da interface *eth2* corresponde à interface de ligação ao *switch*.

```
root@Jasmine:/tmp/pycore.42783/Jasmine.conf# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.135.148  netmask 255.255.255.240  broadcast 0.0.0.0
        inet6 fe80::200:ff:feaa:9  prefixlen 64  scopeid 0x20<link>
        inet6 2001:5::20  prefixlen 64  scopeid 0x0<global>
        ether 00:00:00:aa:00:09  txqueuelen 1000  (Ethernet)
        RX packets 188  bytes 16875 (16.8 KB)
```

Figura 14: Endereço MAC Jasmine.

```
root@Alladin:/tmp/pycore.42783/Alladin.conf# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.135.147  netmask 255.255.255.240  broadcast 0.0.0.0
        inet6 2001:5::21  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::200:ff:feaa:a  prefixlen 64  scopeid 0x20<link>
        ether 00:00:00:aa:00:0a  txqueuelen 1000  (Ethernet)
        RX packets 192  bytes 17167 (17.1 KB)
```

Figura 15: Endereço MAC Alladin.

```
root@Sb:/tmp/pycore.42783/Sb.conf# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.135.146  netmask 255.255.255.240  broadcast 0.0.0.0
        inet6 2001:5::10  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::200:ff:feaa:8  prefixlen 64  scopeid 0x20<link>
        ether 00:00:00:aa:00:08  txqueuelen 1000  (Ethernet)
        RX packets 195  bytes 17622 (17.6 KB)
```

Figura 16: Endereço MAC Servidor.

```
eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.135.145  netmask 255.255.255.240  broadcast 0.0.0.0
        inet6 2001:5::1  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::200:ff:feaa:12  prefixlen 64  scopeid 0x20<link>
        ether 00:00:00:aa:00:12  txqueuelen 1000  (Ethernet)
        RX packets 66  bytes 6535 (6.5 KB)
```

Figura 17: Endereço MAC Alladin.

Após obtermos os endereços MAC, desenvolvemos um esboço da topologia da subrede do Departamento B (incluindo o *router*) para assim termos uma visão simplificada dos sistemas integrantes. Como tal, obtivemos a seguinte tabela de comutação:

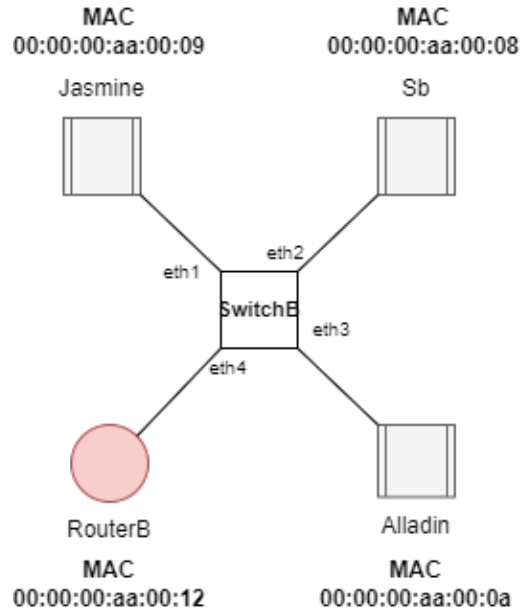


Tabela de Comutação do Switch B:

MAC Address	Interface	TTL
00:00:00:aa:00:09	eth1	60
00:00:00:aa:00:08	eth2	60
00:00:00:aa:00:0a	eth3	60
00:00:00:aa:00:12	eth4	60

4 Conclusão

Com a conclusão deste guião prático, encontramo-nos, em geral, satisfeitos com o trabalho desenvolvido, tendo alcançado todos os objetivos propostos pelos docentes no enunciado.

Em particular, este trabalho prático incidiu sobre a resolução de vários exercícios e problemas da área de redes, nomeadamente a camada de ligação lógica. Assim, o grupo teve a oportunidade de aprofundar os seus conhecimentos em estudo de endereços MAC, funcionamento do protocolo *Ethernet* e interpretação do protocolo ARP. Para além disso, mais uma vez, foi necessário o manuseamento da ferramenta *wireshark*, provando, novamente, a sua utilidade prática no contexto de análise de tráfego.

Em suma, a construção e desenvolvimento deste trabalho prático permitiu a todo o grupo aprofundar os seus conhecimentos no que toca à camada de ligação lógica, atingindo uma ampla percepção de alguns assuntos englobados pela mesma.