

Universidade do Minho

# REDES DE COMPUTADORES

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA

## CAMADA DE LIGAÇÃO LÓGICA

## ETHERNET E PROTOCOLO ARP

[TP3]

GRUPO 9

A85227 João Pedro Rodrigues Azevedo

A85729 Paulo Jorge da Silva Araújo

A83719 Pedro Filipe Costa Machado

Braga  
Novembro 2019

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Análise das questões propostas</b>	<b>3</b>
2.1	Captura e análise de tramas Ethernet . . . . .	3
2.2	Protocolo ARP . . . . .	6
2.2.1	Domínios de colisão . . . . .	10
<b>3</b>	<b>Conclusões</b>	<b>13</b>

# Capítulo 1

## Introdução

Este trabalho foi realizado no âmbito da Unidade Curricular de Redes de computadores, trata-se do terceiro trabalho prático e teve como objetivo principal o estudo da camada de ligação lógica, focando no uso da tecnologia Ethernet e o protocolo ARP (*Address Resolution Protocol*).

De forma resumida, este protocolo é usado pelos diferentes equipamentos de rede para mapear endereços de rede em endereços de uma tecnologia de ligação de dados, como por exemplo, mapear um endereço MAC Ethernet, Wi-fi, num endereço IP particular.

Assim, ao longo deste relatório vão ser apresentadas as várias questões enunciadas e respostas estruturadas às mesmas com diversas demonstrações práticas da sua validade levando a uma secção final de conclusões onde fazemos um balanço de todo o trabalho realizado e a sua importância nesta Unidade Curricular.

## Capítulo 2

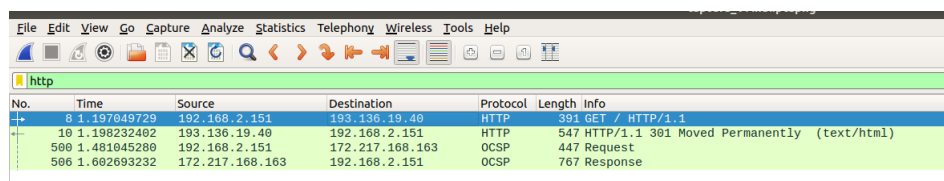
# Análise das questões propostas

### 2.1 Captura e análise de tramas Ethernet

*”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.”*

- **Ative o Wireshark na sua máquina nativa.**
- **No seu browser, aceda ao URL `http://miei.di.uminho.pt`.**
- **Pare a captura do Wireshark.**
- **Obtenha o número de ordem da sequência de bytes capturada (...) correspondente à mensagem HTTP GET (...) bem como o começo da respectiva mensagem HTTP Response proveniente do servidor.**

**R:** Após parar a captura no Wireshark aplicamos o filtro ”http”. Obtivemos o que se pode observar na imagem seguinte:



No.	Time	Source	Destination	Protocol	Length	Info
8	1.197049729	192.168.2.151	193.136.19.40	HTTP	391	GET / HTTP/1.1
10	1.198232482	193.136.19.40	192.168.2.151	HTTP	547	HTTP/1.1 301 Moved Permanently (text/html)
500	1.481045280	192.168.2.151	172.217.168.163	OCSP	447	Request
506	1.682693232	172.217.168.163	192.168.2.151	OCSP	767	Response

Figura 2.1: Captura feita pelo Wireshark, com filtro ”http”.

Verificamos então que o número de ordem correspondente à mensagem ”HTTP GET” é No. 8 e o número de ordem correspondente à respetiva ”HTTP Response” é o No. 10.

- **No sentido de proceder à análise do tráfego, selecione a trama Ethernet que contém a mensagem HTTP GET (...). Expand a informação do nível da ligação de dados (Ethernet II) e observe o conteúdo (...).**

**R:** A trama Ethernet que estava presente no frame da mensagem ”HTTP GET” obtida foi a seguinte:

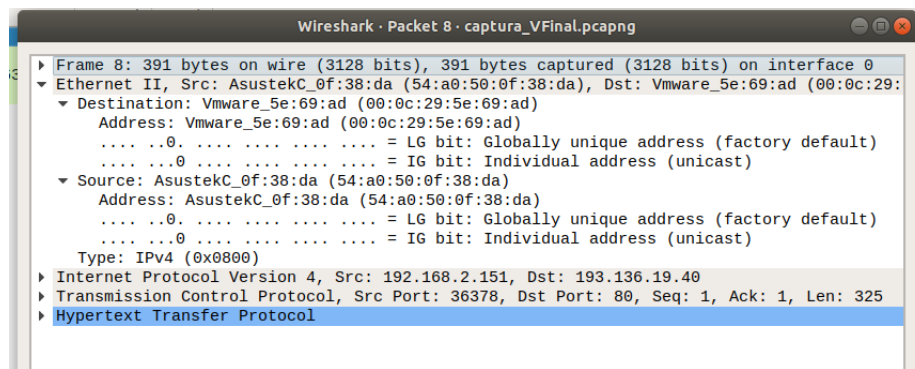


Figura 2.2: Trama Ethernet correspondente à mensagem "HTTP GET".

- Responda às perguntas seguintes com base no conteúdo da trama Ethernet que contém a mensagem HTTP GET (...). Selecione o mínimo detalhe necessário):

1) "Anotar os endereços MAC de origem e de destino da trama capturada."

R: Endereço MAC (Ethernet) origem: [54:a0:50:0f:38:da] (nossa máquina) e destino: [00:0c:29:5e:69:ad] (servidor do website).

2) "Identifique a que sistemas se referem. Justifique."

R: O endereço MAC, de um total de 6 bytes, dedica os 3 primeiros bytes, segunda a norma IEEE, à identificação do "fabricante do endereço". Assim, o endereço origem identifica o sistema **AsustekC** e o endereço destino identifica o sistema **Vmware**.

3) "Qual o valor hexadecimal do campo Type da trama Ethernet? O que significa?"

R: O campo **Type** da trama pode ser observado no cabeçalho da secção Ethernet II e tem o valor em hexadecimal: **0x0800** que identifica o tipo de dados (payload) da trama Ethernet que, neste caso, se refere ao protocolo **IPv4**. Por outro lado, podemos interpretar o payload da trama Ethernet como sendo a camada superior (*Layer 3 - Network*).

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."

R: Para melhor visualização dos bytes usados desde o início da trama até ao caractere ASCII "G", vejamos a seguinte imagem:

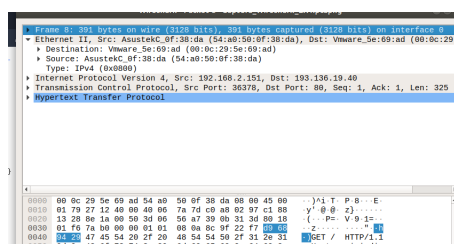


Figura 2.3: Sequência de bytes que compõem o frame capturado.

Podemos visualizar que até ao caractere "G" (sublinhado a azul) temos 66 bytes (byte 0-65). Por outro lado, temos um total de 391 bytes no frame capturado pelo Wireshark, o que nos indica, em percentagem  $\frac{66}{391} * 100 = 18.28\%$  de *overhead* introduzido pela pilha protocolar no envio do HTTP GET.

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?"

**R:** Efetivamente o campo FCS (*Frame Check Sequence*) não está a ser usado, visto que, se fosse usado apareceria no fim do frame capturado, como se pode ver na seguinte representação:

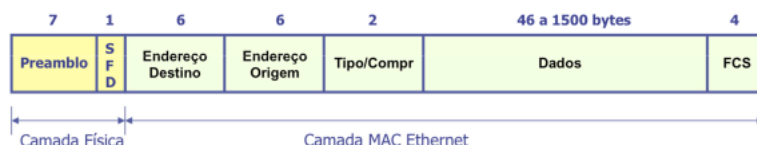


Figura 2.4: Divisão de uma trama Ethernet

Na nossa opinião, uma possível explicação para abdicar deste campo de verificação de erros deve-se ao tipo de ligação estabelecida para o envio de pacotes que, neste caso, trata-se de uma conexão física através de um cabo ethernet com uma probabilidade de erro muito baixa (quase nula), assumindo-se que toda a estrutura de rede está bem "montada". Tal facto, possivelmente, não aconteceria se estivessemos a falar de conexões Wi-Fi que estão associadas a uma probabilidade de erro muito maior o que é consequência de um maior *delay* de ligação.

- A seguir responda às seguintes perguntas, baseado no conteúdo da trama Ethernet que contém o primeiro byte da resposta HTTP:

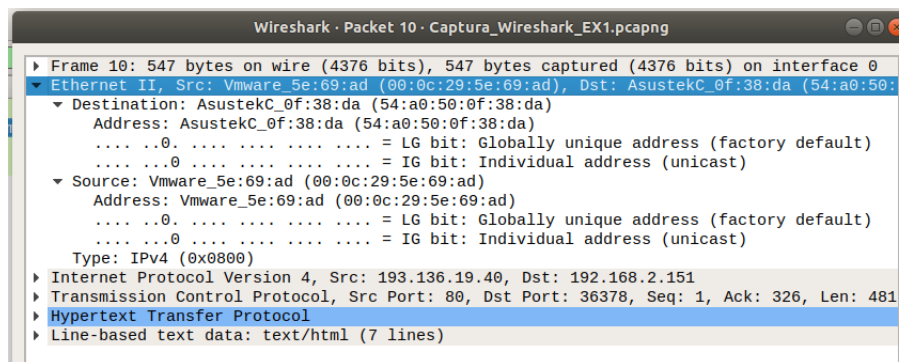


Figura 2.5: Trama Ethernet correspondente à mensagem "HTTP Response".

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

**R:** O endereço MAC ethernet da fonte (*Source*) é: [00:0c:29:5e:69:ad]. Corresponde ao sistema **Vmware** identificado pelos primeiros 3 bytes do endereço.

7) "Qual é o endereço MAC do destino? A que sistema corresponde?"

**R:** O endereço MAC ethernet do destino (*Destination*) é: [54:a0:50:0f:38:da]. Corresponde ao sistema **AsustekC** identificado pelos primeiros 3 bytes do endereço.

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

**R:** Na trama recebida conseguimos identificar, com a ajuda do Wireshark, 4 protocolos: Ethernet II (802.3), IPv4, TCP e o HTTP. Ao selecionar os diferentes protocolos no programa vemos que o protocolo Ethernet II utiliza os bytes 0-13 (14 bytes), o IPv4 utiliza os bytes 13-33 (20 bytes), o TCP utiliza os bytes 33-65 (32 bytes) e o HTTP utiliza os bytes 65-312 (247 bytes). O frame capturado Ethernet II (*Layer 2 - Link*) é usado pelo protocolo da camada superior IPv4 (*Layer 3 - Network*) que por sua vez é utilizado pelo TCP (*Layer 4 - Transport*) e por fim o TCP encapsula o protocolo HTTP (*Layer 7 - Application*).

## 2.2 Protocolo ARP

**Nota:** O exercício anterior foi realizado com um IP diferente do IP usado neste exercício devido às diferentes portas associadas a diferentes redes na mesma sala, algo que foi explicitado pelo docente.

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

**R:** A tabela ARP (que contém o conteúdo da cache ARP), num sistema Linux/Unix como o nosso, pode ser encontrada executando o comando **arp -a** ou simplesmente **arp** (este último apresenta a tabela com colunas fixas). Após executar esses comandos obtivemos o seguinte resultado:

```
joao@azevedo-n550jk:~$ arp
Address                  HWtype  HWaddress    Flags Mask    Iface
gw.sa.di.uminho.pt      ether    00:0c:29:d2:19:f0    C              enp5s0
joao@azevedo-n550jk:~$ arp -a
gw.sa.di.uminho.pt (192.168.100.254) at 00:0c:29:d2:19:f0 [ether] on enp5s0
joao@azevedo-n550jk:~$
```

Figura 2.6: Tabelas ARP.

A tabela ARP, com a execução do segundo comando, apresenta um mapeamento do host com o nome "gw.sa.di.uminho.pt" e endereço de rede **192.168.100.254** para o seu respetivo endereço MAC **00:0c:29:d2:19:f0**, por sua vez, este endereço pode ser acedida por uma ligação ethernet (no *output link* enp5s0).

De forma mais geral, a partir da execução de **arp** apenas, temos que, **Address** indica-nos o endereço de rede; **HWtype** o tipo de ligação; **HWaddress** o endereço MAC respetivo; as **Flags** o estado da entrada (linha), que pode ser **C** - *Complete*, **P** - *Published* ou **M** - *Permanent*; **Iface** a interface a utilizar para aceder ao endereço anterior.

---

No sentido de observar o envio e recepção de mensagens ARP, é conveniente apagar o conteúdo da cache ARP. Caso contrário, é provável que a associação entre endereços IP e MAC já exista em cache.

---

**R:** Deste modo executamos o comando (**arp -d \* ... arp -d -a**) para apagar a cache ARP e também removemos toda a cache do nosso *browser*.

Inicie a captura de tráfego com o Wireshark, e acesse 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:

```
joao@azevedo-n550jk:~$ arp
```

Address	HWtype	HWaddress	Flags	Mask	Iface
gw.sa.di.uminho.pt	ether	00:0c:29:d2:19:f0	C		enp5s0
192.168.100.206	ether	00:24:32:17:35:8a	C		enp5s0

```
joao@azevedo-n550jk:~$
```

Figura 2.7: Tabela **arp** após limpeza de *cache* e execução dos passos acima descritos.

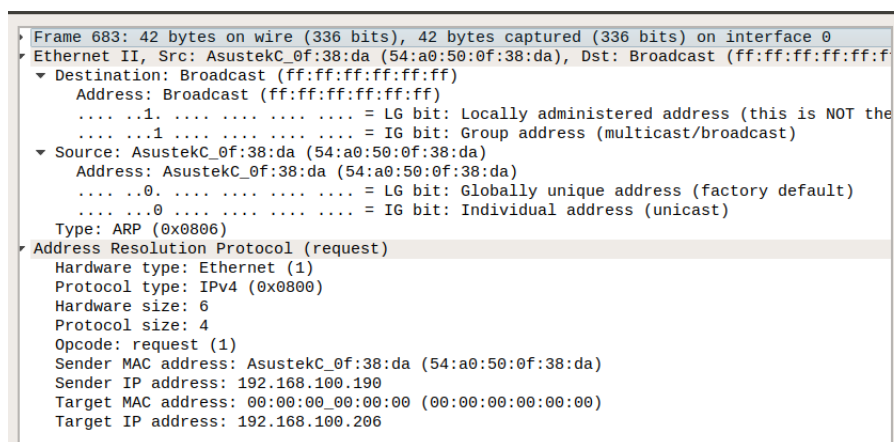


Figura 2.8: Trama ethernet correspondente à mensagem ARP Request.

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?"

**R:** O endereço origem da trama Ethernet é **54:a0:50:0f:38:da** e o endereço destino **ff:ff:ff:ff:ff:ff**. O endereço origem corresponde ao IP da interface do nosso computador; O endereço destino corresponde a um destino *broadcast*, no qual, a mesma mensagem é enviada a todos os *hosts* da rede de modo a obter uma resposta do destino correspondente.

11) "Qual o valor hexadecimal do campo tipo da trama Ethernet? O que indica?"

**R:** O campo **Type** da trama pode ser observado no cabeçalho da secção Ethernet II e tem o valor em hexadecimal: **0x0806** que identifica o tipo de dados (payload) da trama Ethernet que, neste caso, se refere ao protocolo **ARP** (também pertencente ao *Layer 2 - Link*).

12) "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>)."

**R:** O campo **opcode**, no Wireshark, apresenta o valor "request (1)". Este campo, segundo a documentação consultada, especifica o tipo de mensagem ARP que foi enviada e, pode tomar vários valores possíveis, desde (1) para *request* e (2) para *reply*.





Um ARP Gratuito envolve o envio de um ARP request ou ARP reply gratuito, i.e. um host faz um pedido ou uma resposta ARP sem que, segundo a especificação ARP, haja necessidade de o fazer. (...). Arranque o Wireshark na sua máquina nativa e inicie a captura de dados. Desligue e volte a ligar a sua ligação à rede local Ethernet, (...) Utilize o filtro de visualização ARP para facilitar a identificação dos pacotes respectivos.

1	0.000000000	Vmware_d2:19:f0	Broadcast	ARP	60 Who has 192.168.100.163? Tell 192.168.100.254
2	0.285937874	BizlinkK_07:8b:e5	Broadcast	ARP	60 Gratuitous ARP for 192.168.100.203 (Request)
18	1.000559366	Vmware_d2:19:f0	Broadcast	ARP	60 Who has 192.168.100.163? Tell 192.168.100.254
21	1.286077376	BizlinkK_07:8b:e5	Broadcast	ARP	60 Gratuitous ARP for 192.168.100.203 (Request)
23	2.001160770	Vmware_d2:19:f0	Broadcast	ARP	60 Who has 192.168.100.163? Tell 192.168.100.254
24	2.121418304	AsustekC_0f:38:da	Broadcast	ARP	42 Gratuitous ARP for 192.168.100.190 (Request)
25	2.286188994	BizlinkK_07:8b:e5	Broadcast	ARP	60 Gratuitous ARP for 192.168.100.203 (Request)
26	3.121463913	AsustekC_0f:38:da	Broadcast	ARP	42 Gratuitous ARP for 192.168.100.190 (Request)

Figura 2.10: Captura Wireshark após execução do comando `arping -U 192.168.100.190`

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?"

**R:** O primeiro ARP gratuito enviado pelo nosso sistema foi o seguinte:

```

▶ Frame 26: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▼ Ethernet II, Src: AsustekC_0f:38:da (54:a0:50:0f:38:da), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  ▶ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  ▶ Source: AsustekC_0f:38:da (54:a0:50:0f:38:da)
    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: AsustekC_0f:38:da (54:a0:50:0f:38:da)
  Sender IP address: 192.168.100.190
  Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)
  Target IP address: 192.168.100.190

```

Figura 2.11: ARP Gratuito originado pelo nosso sistema (AsustekC).

O pacote ARP Gratuito originado pelo nosso sistema difere dos outros pacotes ARP Request na medida em que o pedido é enviado do nosso sistema para o nosso sistema o que é transcrito através da documentação consultada associada a estes pacotes. Por outro lado, ao contrário dos normais ARP Request, estes não obtêm um ARP Reply comum, o que faz sentido.

Estes pedidos gratuitos permitem assim que o nosso sistema informe a *hosts* e *switches* da rede local sobre o nosso endereço MAC de modo tornar mais eficiente e rápida a criação e atualização das suas tabelas ARP e, por outro lado, a garantia de que o IP atribuído (por DHCP ou manualmente) é efetivamente único na rede, ou seja, não existem conflitos.

### 2.2.1 Domínios de colisão

Uma rede local onde existam vários equipamentos ligados através de um meio partilhado comum constitui o que é denominado um domínio de colisão. (...) As normas Ethernet implementam um método de controlo de acesso ao meio denominado CSMA/CD (estudado nas aulas teóricas) que tenta prever e resolver estas colisões. (...) Construa uma topologia no emulador CORE com um host (n1) e dois servidores (n2, n3) interligados através de um hub.

---

A topologia de rede com o **hub** definida no CORE foi a seguinte:

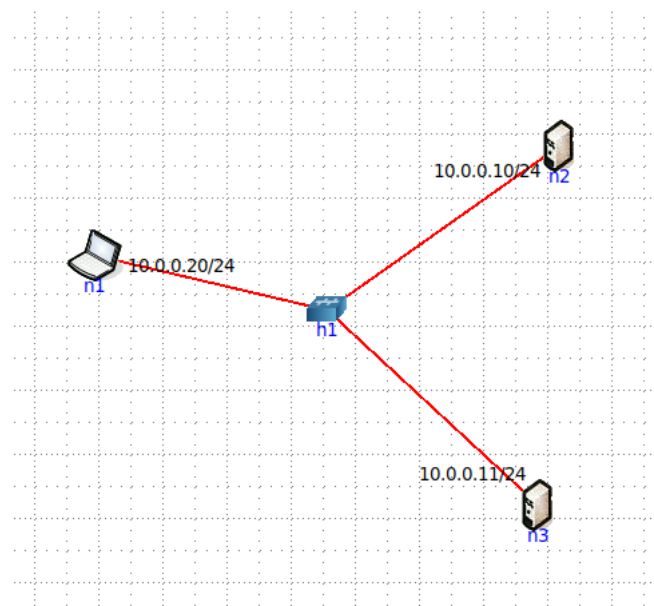


Figura 2.12: Topologia de rede.

---

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

**R:** Após executar o comando **ping 10.0.0.10** (de n1 para n2) podemos verificar o seguinte fluxo de tráfego nos diferentes dispositivos da rede: (ver próx. página)

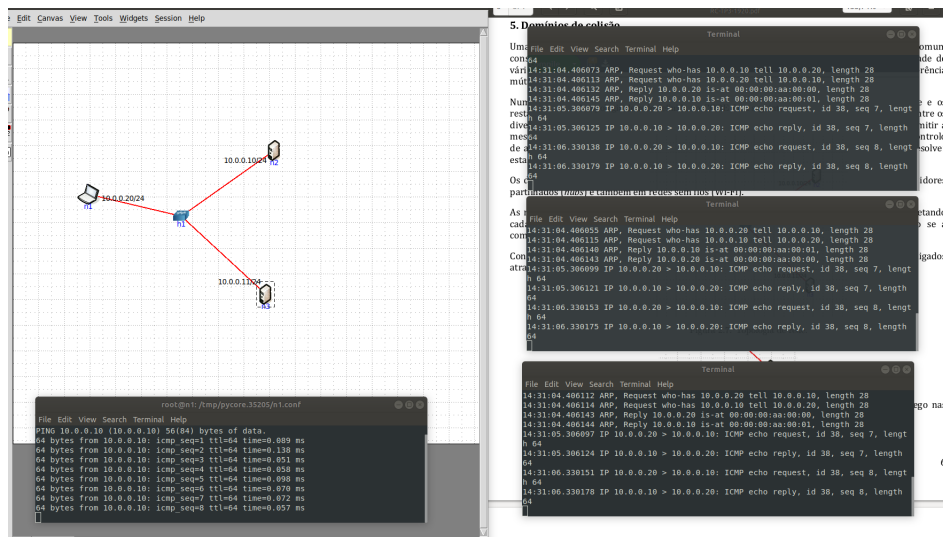


Figura 2.13: Fluxo obtido com o comando **tcdump**.

Deste modo, observando o resultado do comando **tcdump** nas diversas interfaces dos vários dispositivos conseguimos ver os pacotes que estão a chegar a cada dispositivo na rede em que os mesmos estão inseridos. Devido ao facto de todos os dispositivos estarem interligados num **hub**<sup>1</sup>, todos vão receber o mesmo *output* do comando enviado pelo n1 de *request* e enviado pelo n2 de *reply*, ou seja, se n1 envia pacotes, seja para qual destino for, o hub transmite esses pacotes para n2 e n3; depois a resposta dada por n2 vai ser transmitida para n1 e n3 (neste tempo, n3 rejeita a informação recebida visto que não é ele o destino do pacote); note-se que para transmitir estes pacotes utilizam-se os endereços físicos dos dispositivos (mais propriamente, os Ethernet MAC).

18) *”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.”*

**R:** A topologia de rede com o **switch** definida no CORE foi a seguinte:

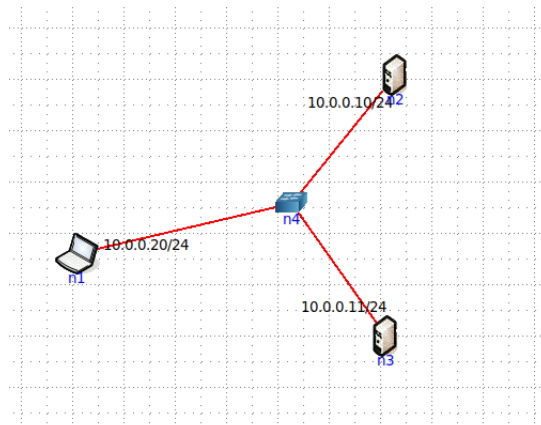


Figura 2.14: Topologia de rede usando um **switch**.

<sup>1</sup>Quando recebe informação numa determinada porta, o HUB transmite essa informação por todas as outras portas, excepto por aquela que recebeu essa informação, criando assim um único domínio de colisão e diminuindo a performance.

O fluxo que obtivemos nos diferentes dispositivos da rede foi o seguinte:

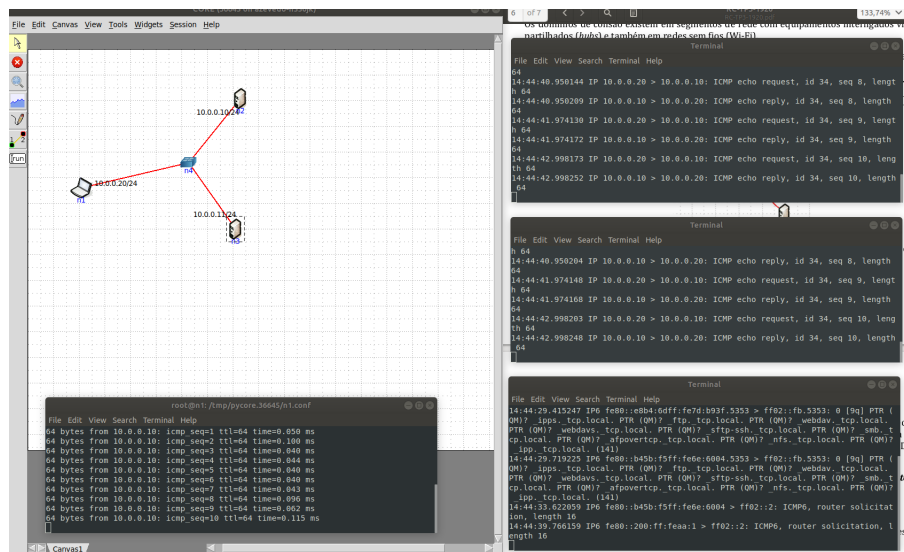


Figura 2.15: Fluxo obtido com o comando `tcddump`.

Concluimos então, pela observação dos *outputs* dos diferentes terminais que o *output* do dispositivo n3 não regista qualquer tipo de tráfego ICMP produzido pelo comando `ping 10.0.0.10`. O que faz sentido visto que os dispositivos estão interligados com um **switch**<sup>2</sup> e não um **hub**, ou seja, o sinal recebido no hub com o pacote enviado por n1 é apenas repetido para n2 e não para n3, e a resposta de n2 é repetida para n1. Assim, dividimos o domínio de colisão, i.e., o meio partilhado não é o mesmo que num hub.

<sup>2</sup>Envia os dados directamente para o destino, ou seja, os dados não são repetidos desnecessariamente por todas as portas.

## Capítulo 3

# Conclusões

A elaboração deste trabalho permitiu aprimorar a vertente prática associada a esta Unidade Curricular, no qual se inclui o estudo do encapsulamento protocolar estruturado onde cada camada fornece serviços às camadas superiores e usa serviços disponibilizados pelas camadas inferiores. Vimos como principal exemplo aceder a um website e registar a pilha protocolar associada a uma trama Ethernet (em duas fases *HTTP Request* e *HTTP Response*). A análise do tráfego deu-nos a conhecer a estrutura da trama Ethernet explorando os seus campos: Desde a posição dos endereços MAC, o tipo de dados do payload e até o campo opcional para deteção de erros (*FCS - Frame Check Sequence*).

Numa segunda fase, exploramos a necessidade de existência de um protocolo de mapeamento entre endereços de nível de rede (IP) e endereços de ligação lógica (MAC), chamado ARP (*Address Resolution Protocol*). Este protocolo permitia criar e atualizar essas tabelas de mapeamento de modo a conhecer os endereços MAC associados a um dado endereço IP. Por outro lado, ficamos a conhecer também o conceito de ARP Gratuito que se traduz num sistema importante para aumentar a eficiência e rapidez de criação dessas mesmas tabelas antes de sequer haver necessidade de conhecer certos endereços. Aprendemos também que este último conceito server para testar a unicidade de endereços IP existentes numa rede, i.e., após ser atribuído um endereço IP à máquina (por DHCP ou manualmente) deve ser testada a unicidade do mesmo.

Por fim, analisámos o último tema proposto que incidia sobre os domínios de colisão em redes locais, i.e., que podem interligar os seus dispositivos através de repetidores *hubs* ou comutadores *switches* e chegámos à conclusão que a utilização de um ou outro depende do conceito de rede que pretendemos implementar. Por um lado, temos que o repetidor repete o sinal que recebe para todos os nós ligados ao mesmo, ou seja, existe um domínio partilhado entre os diferentes dispositivos (chamado de domínio de colisão) que pode levar a que exista um momento onde estes coincidam no envio de tramas o que causaria uma interferência (colisão) e, por outro lado, temos um switch que repete o sinal recebido apenas para o nó destino, dividindo assim, o domínio de colisão em vários domínios onde um dispositivo comunica diretamente com outro.