

# TP3: Aplicações e Camada de Transporte

Eduardo Cunha A98980, Gonalo Magalhães A100084, Fáblio Ribeiro A100058

Universidade do Minho, Escola de Ciências, Ciências da Computação

PL1, Grupo 1

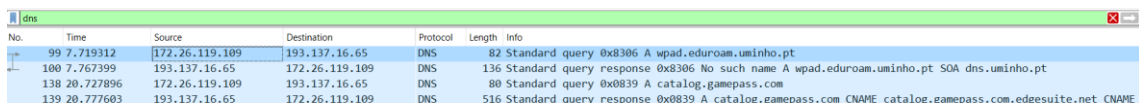
## 1. Objetivo

Este trabalho tem como objetivo a familiarização com protocolos e ferramentas do nível aplicacional, e análise dos protocolos de transporte em uso. Para tal, deve usar a sua máquina nativa (preferencialmente com o sistema operativo Linux), e não a máquina virtual.

## 2. Nível Aplicacional

Ligue-se à rede Eduroam e proceda da seguinte forma. Ative um browser na sua máquina e certifique-se que não tem outras instâncias *web* ativas. Ative o Wireshark, certificando-se que está em modo privilegiado (*root*), e proceda à captura de tráfego na interface de rede *wi-fi* em uso. Aceda à página <http://www.sas.uminho.pt> e espere que o conteúdo seja carregado. Pare a captura no Wireshark e grave-a para eventual uso posterior. Para localizar mais facilmente o tráfego correspondente ao acesso *web* realizado, comece por filtrar pelo protocolo *dns*. Para tal, introduza *dns* na caixa do *display filter* e aplique o filtro. (Também pode usar “Edit > Find Packet...” (ou CTRL+F) para encontrar os pacotes contendo *strings* relativas ao nome do servidor). Localize a resolução do nome do servidor [www.sas.uminho.pt](http://www.sas.uminho.pt).

- 2.1 Identifique o endereço IP da estação que formulou a *query* DNS e o tipo de *query* realizada. (Nota: Caso não consiga encontrar a referida *query*, limpe a cache DNS da sua máquina, executando num terminal do Ubuntu: `sudo systemd-resolve --flush-caches`; ou `sudo/etc/init.d/dns-clean restart`. No Windows deve executar o comando: `ipconfig /flushdns`).

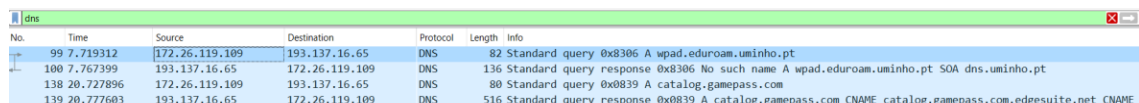


No.	Time	Source	Destination	Protocol	Length	Info
99	7.719312	172.26.119.109	193.137.16.65	DNS	82	Standard query 0x8306 A wpad.eduroam.uminho.pt
100	7.767399	193.137.16.65	172.26.119.109	DNS	136	Standard query response 0x8306 No such name A wpad.eduroam.uminho.pt SOA dns.uminho.pt
138	20.727896	172.26.119.109	193.137.16.65	DNS	80	Standard query 0x0839 A catalog.gamepass.com
139	20.777603	193.137.16.65	172.26.119.109	DNS	516	Standard query response 0x0839 A catalog.gamepass.com CNAME catalog.gamepass.com.edgesuite.net CNAME

Figura 1

Endereço IP da estação que formulou a *query* DNS é 193.137.16.65

- 2.2 Localize a trama com a resposta à *query* DNS formulada. Identifique nesta trama o endereço IP do servidor *web*. Identifique também o servidor de nomes que forneceu a resposta, através do seu IP e nome (sugestão: consulte o *Additional Records*).



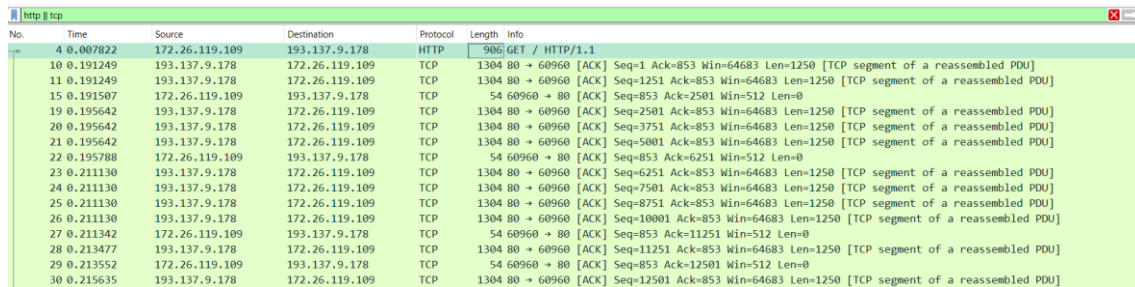
No.	Time	Source	Destination	Protocol	Length	Info
99	7.719312	172.26.119.109	193.137.16.65	DNS	82	Standard query 0x8306 A wpad.eduroam.uminho.pt
100	7.767399	193.137.16.65	172.26.119.109	DNS	136	Standard query response 0x8306 No such name A wpad.eduroam.uminho.pt SOA dns.uminho.pt
138	20.727896	172.26.119.109	193.137.16.65	DNS	80	Standard query 0x0839 A catalog.gamepass.com
139	20.777603	193.137.16.65	172.26.119.109	DNS	516	Standard query response 0x0839 A catalog.gamepass.com CNAME catalog.gamepass.com.edgesuite.net CNAME

Figura 2

O endereço IP do servidor web é 193.137.16.65. O servidor de nomes que forneceu a resposta é "dns.uminho.pt".

## HTTP e TCP

### 2.3 Aplique o filtro aos protocolos *http* // *tcp*. Identifique os endereços IP do cliente e do servidor HTTP.



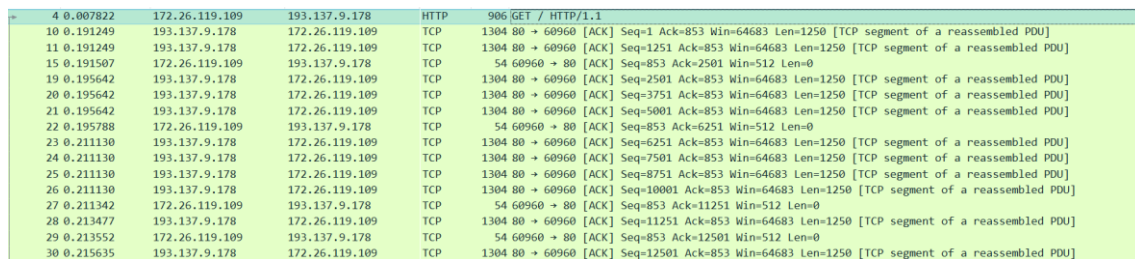
No.	Time	Source	Destination	Protocol	Length	Info
4	0.007822	172.26.119.109	193.137.9.178	HTTP	906	GET / HTTP/1.1
10	0.191249	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=1 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
11	0.191249	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=1251 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
15	0.191507	172.26.119.109	193.137.9.178	TCP	54	60960 → 80 [ACK] Seq=853 Ack=2501 Win=512 Len=0
19	0.195642	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=2501 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
20	0.195642	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=3751 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
21	0.195642	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=5001 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
22	0.195788	172.26.119.109	193.137.9.178	TCP	54	60960 → 80 [ACK] Seq=853 Ack=6251 Win=512 Len=0
23	0.211130	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=6251 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
24	0.211130	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=7501 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
25	0.211130	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=8751 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
26	0.211130	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=10001 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
27	0.211342	172.26.119.109	193.137.9.178	TCP	54	60960 → 80 [ACK] Seq=853 Ack=11251 Win=512 Len=0
28	0.213477	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=11251 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
29	0.213552	172.26.119.109	193.137.9.178	TCP	54	60960 → 80 [ACK] Seq=853 Ack=12501 Win=512 Len=0
30	0.215635	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=12501 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]

Figura 3

Endereço IP do cliente: 172.26.119.109

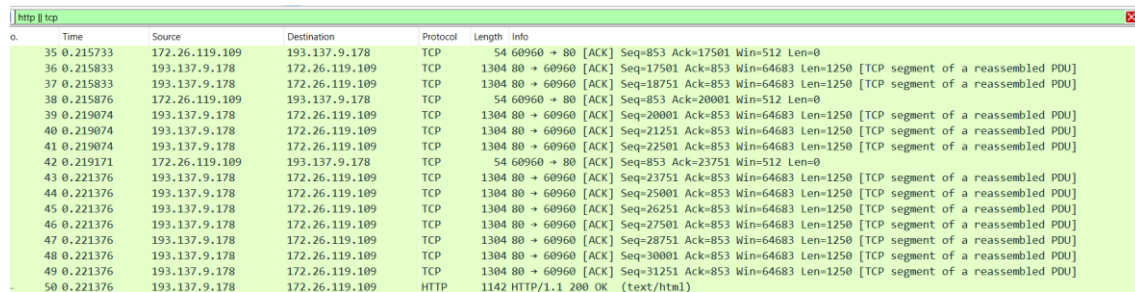
Endereço IP do servidor HTTP: 193.137.9.178

### 2.4 Identifique os segmentos TCP correspondentes ao estabelecimento da ligação entre o cliente e o servidor HTTP. Qual o tamanho máximo de segmento (MSS) que o servidor aceita receber?



No.	Time	Source	Destination	Protocol	Length	Info
4	0.007822	172.26.119.109	193.137.9.178	HTTP	906	GET / HTTP/1.1
10	0.191249	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=1 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
11	0.191249	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=1251 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
15	0.191507	172.26.119.109	193.137.9.178	TCP	54	60960 → 80 [ACK] Seq=853 Ack=2501 Win=512 Len=0
19	0.195642	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=2501 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
20	0.195642	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=3751 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
21	0.195642	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=5001 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
22	0.195788	172.26.119.109	193.137.9.178	TCP	54	60960 → 80 [ACK] Seq=853 Ack=6251 Win=512 Len=0
23	0.211130	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=6251 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
24	0.211130	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=7501 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
25	0.211130	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=8751 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
26	0.211130	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=10001 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
27	0.211342	172.26.119.109	193.137.9.178	TCP	54	60960 → 80 [ACK] Seq=853 Ack=11251 Win=512 Len=0
28	0.213477	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=11251 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
29	0.213552	172.26.119.109	193.137.9.178	TCP	54	60960 → 80 [ACK] Seq=853 Ack=12501 Win=512 Len=0
30	0.215635	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=12501 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]

Figura 4



No.	Time	Source	Destination	Protocol	Length	Info
35	0.215733	172.26.119.109	193.137.9.178	TCP	54	60960 → 80 [ACK] Seq=853 Ack=17501 Win=512 Len=0
36	0.215833	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=17501 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
37	0.215833	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=18751 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
38	0.215876	172.26.119.109	193.137.9.178	TCP	54	60960 → 80 [ACK] Seq=853 Ack=20001 Win=512 Len=0
39	0.219074	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=20001 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
40	0.219074	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=21251 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
41	0.219074	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=22501 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
42	0.219171	172.26.119.109	193.137.9.178	TCP	54	60960 → 80 [ACK] Seq=853 Ack=23751 Win=512 Len=0
43	0.221376	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=23751 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
44	0.221376	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=25001 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
45	0.221376	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=26251 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
46	0.221376	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=27501 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
47	0.221376	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=28751 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
48	0.221376	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=30001 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
49	0.221376	193.137.9.178	172.26.119.109	TCP	1304	80 → 60960 [ACK] Seq=31251 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
50	0.221376	193.137.9.178	172.26.119.109	HTTP	1142	HTTP/1.1 200 OK (text/html)

Figura 5

O tamanho máximo que o servidor aceita receber, como se pode observar pelo "Len" é 1250.

**2.5 Identifique a resposta HTTP do servidor respeitante ao primeiro pedido GET efetuado pelo cliente. Quantos bytes de dados aplicacionais contém essa resposta HTTP?**

No.	Time	Source	Destination	Protocol	Length	Info
4	0.007822	172.26.119.109	193.137.9.178	HTTP	906	GET / HTTP/1.1
50	0.221376	193.137.9.178	172.26.119.109	HTTP	1142	HTTP/1.1 200 OK (text/html)
53	0.287231	172.26.119.109	193.137.9.178	HTTP	906	GET / HTTP/1.1
86	0.403100	193.137.9.178	172.26.119.109	HTTP	1142	HTTP/1.1 200 OK (text/html)
88	0.750836	172.26.119.109	193.137.9.178	HTTP	852	GET /favicon.ico HTTP/1.1
90	0.908454	193.137.9.178	172.26.119.109	HTTP	599	HTTP/1.1 404 Not Found (text/html)
93	1.030087	172.26.119.109	193.137.9.178	HTTP	871	GET /images/tab/Alojamento_Over.gif HTTP/1.1
94	1.037062	193.137.9.178	172.26.119.109	HTTP	745	HTTP/1.1 200 OK (GIF89a)

Figura 6

```
> Internet Protocol Version 4, Src: 193.137.9.178, Dst: 172.26.119.109
> Transmission Control Protocol, Src Port: 80, Dst Port: 60960, Seq: 32501, Ack: 853, Len: 1088
> [27 Reassembled TCP Segments (33588 bytes): #10(1250), #11(1250), #19(1250), #20(1250), #21(1250), #23(1250), #2
```

Figura 7

A resposta HTTP contém 1142 bytes dos quais 1088 são aplicáveis.

**2.6 A resposta HTTP identificada na alínea anterior foi transmitida em quantos segmentos TCP? Apresente também uma estimativa teórica para essa quantidade.**

```
> Ethernet II, Src: Cisco_ab:ac:cf (90:77:ee:ab:ac:cf), Dst: Chongqin_47:1e:69 (c8:94:02:47:1e:69)
> Internet Protocol Version 4, Src: 193.137.9.178, Dst: 172.26.119.109
> Transmission Control Protocol, Src Port: 80, Dst Port: 60960, Seq: 32501, Ack: 853, Len: 1088
> [27 Reassembled TCP Segments (33588 bytes): #10(1250), #11(1250), #19(1250), #20(1250), #21(1250), #23(1250), #24(1250)]
✓ Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n
    Date: Fri, 09 Dec 2022 16:04:48 GMT\r\n
    Server: Microsoft-IIS/6.0\r\n
    X-Powered-By: ASP.NET\r\n
    X-AspNet-Version: 1.1.4322\r\n
    Cache-Control: private\r\n
    Content-Type: text/html; charset=iso-8859-15\r\n
  ✓ Content-Length: 33361\r\n
    [Content length: 33361]
    \r\n
    [HTTP response 1/4]
    [Time since request: 0.213554000 seconds]
```

Figura 8

Como podemos ver pela imagem a resposta HTTP foi enviada em 27 segmentos. O que seria de esperar pois, como o total do comprimento da resposta é de 333361 e a capacidade máxima de cada segmento TCP é 1250, então  $333361/1250 = 26,6(8)$ , logo são necessários 27 segmentos.

**2.7 A partir da informação contida nos cabeçalhos dos protocolos IP e TCP, determine o número de bytes de dados enviados no primeiro e no último segmento TCP respeitantes à resposta HTTP.**

4 0.007822	172.26.119.109	193.137.9.178	HTTP	906 GET / HTTP/1.1
10 0.191249	193.137.9.178	172.26.119.109	TCP	1304 80 → 60960 [ACK] Seq=1 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]

Figura 9

49 0.221376	193.137.9.178	172.26.119.109	TCP	1304 80 → 60960 [ACK] Seq=31251 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
50 0.221376	193.137.9.178	172.26.119.109	HTTP	1142 HTTP/1.1 200 OK (text/html)
51 0.231550	172.26.119.109	193.137.9.178	TCP	54 60960 → 80 [ACK] Seq=853 Ack=32680 Win=512 Len=0

*Figura 10*

Como podemos observar pelas imagens o número de bytes enviados no primeiro e no último segmento TCP é 1250.

**2.8** Observe a informação apresentada no campo *host* do cabeçalho do pedido HTTP e diga qual o seu interesse? Experimente aceder à mesma página *web* através de *http://endereço\_IP*, em que *endereço\_IP* é o respeitante a *www.sas.uminho.pt* (identificado na alínea 2). Justifique o comportamento observado.

```

v Hypertext Transfer Protocol
  > GET / HTTP/1.1\r\n
    Host: www.sas.uminho.pt\r\n
    Connection: keep-alive\r\n

```

Figura 11

A informação guardada no host do pedido http é "www.sas.uminho.pt", que é o site para o qual estamos a tentar aceder.

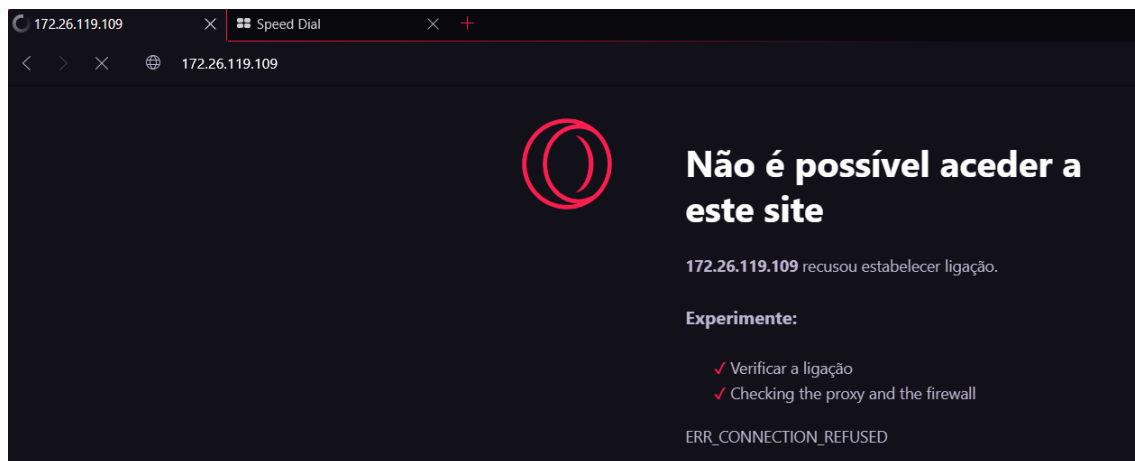


Figura 12

O acesso dá erro pois a mesma máquina pode estar a alojar vários servidores de nomes diferentes.

**2.9** Com base na sequência de dados trocados entre o cliente e o servidor diga, justificando, se o servidor HTTP está a funcionar em modo de conexão persistente ou não persistente.

19	5.873404	172.26.119.109	193.137.9.178	TCP	54 60396 → 80 [ACK] Seq=1 Ack=1 Win=131072 Len=0
20	5.873490	172.26.119.109	193.137.9.178	TCP	54 60397 → 80 [ACK] Seq=1 Ack=1 Win=131072 Len=0
21	5.873972	172.26.119.109	193.137.9.178	HTTP	906 GET / HTTP/1.1
23	6.071497	193.137.9.178	172.26.119.109	TCP	1304 80 → 60396 [ACK] Seq=1 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
24	6.071497	193.137.9.178	172.26.119.109	TCP	1304 80 → 60396 [ACK] Seq=1251 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
26	6.071669	172.26.119.109	193.137.9.178	TCP	54 60396 → 80 [ACK] Seq=853 Ack=2501 Win=131072 Len=0
27	6.073045	172.26.119.109	185.26.182.93	TCP	66 60398 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
28	6.075387	193.137.9.178	172.26.119.109	TCP	1304 80 → 60396 [ACK] Seq=2501 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
29	6.075387	193.137.9.178	172.26.119.109	TCP	1304 80 → 60396 [ACK] Seq=3751 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
30	6.075469	172.26.119.109	193.137.9.178	TCP	54 60396 → 80 [ACK] Seq=853 Ack=5001 Win=131072 Len=0
31	6.077198	193.137.9.178	172.26.119.109	TCP	1304 80 → 60396 [ACK] Seq=5001 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
32	6.077198	193.137.9.178	172.26.119.109	TCP	1304 80 → 60396 [ACK] Seq=6251 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
33	6.077198	193.137.9.178	172.26.119.109	TCP	1304 80 → 60396 [ACK] Seq=7501 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
34	6.077332	172.26.119.109	193.137.9.178	TCP	54 60396 → 80 [ACK] Seq=853 Ack=8751 Win=131072 Len=0
35	6.082569	193.137.9.178	172.26.119.109	TCP	1304 80 → 60396 [ACK] Seq=8751 Ack=853 Win=64683 Len=1250 [TCP segment of a reassembled PDU]
36	6.082709	172.26.119.109	193.137.9.178	TCP	54 60396 → 80 [ACK] Seq=853 Ack=10001 Win=131072 Len=0

Figura 13

A conexão realizada com o servidor está a funcionar de modo persistente porque há o envio de múltiplas tramas TCP sobre a mesma conexão.

**2.10** Aplique o filtro apenas ao protocolo *http*. O *hard refresh* permite limpar a cache do browser para uma determinada página, forçando o browser a carregar a última versão da página existente no servidor. Normalmente o *hard refresh* numa página faz-se com CTRL+F5 ou SHIFT+page reload (caso não funcione, procure na Internet a forma de fazer *hard refresh* no seu browser). Coloque o Wireshark a capturar tráfego e faça *hard refresh* da página indicada anteriormente. Depois volte a aceder à mesma página mas sem fazer *hard refresh*. Pare a captura de tráfego. Identifique a principal diferença observada no tráfego HTTP entre carregar a referida página com e sem *hard refresh*. Qual a principal vantagem e desvantagem inerente ao *hard refresh*?

1	0.000000	172.26.119.109	193.137.9.178	HTTP	949 GET / HTTP/1.1
32	0.170477	193.137.9.178	172.26.119.109	HTTP	1142 HTTP/1.1 200 OK (text/html)
34	0.181049	172.26.119.109	193.137.9.178	HTTP	848 GET /portal.css HTTP/1.1
35	0.184302	172.26.119.109	193.137.9.178	HTTP	841 GET /lib/clientUtils.js HTTP/1.1
36	0.184656	172.26.119.109	193.137.9.178	HTTP	842 GET /lib/1k_standards.js HTTP/1.1
39	0.195159	193.137.9.178	172.26.119.109	HTTP	1222 HTTP/1.1 200 OK (application/x-javascript)
41	0.195159	193.137.9.178	172.26.119.109	HTTP	87 HTTP/1.1 200 OK (application/x-javascript)
44	0.201652	172.26.119.109	193.137.9.178	HTTP	914 GET /images/escolas/corReitoria.gif HTTP/1.1
45	0.209802	193.137.9.178	172.26.119.109	HTTP	1206 HTTP/1.1 200 OK (GIF89a)
46	0.214890	172.26.119.109	193.137.9.178	HTTP	908 GET /images/globais/en-us.gif HTTP/1.1
48	0.219528	193.137.9.178	172.26.119.109	HTTP	245 HTTP/1.1 200 OK (GIF89a)
50	0.225334	172.26.119.109	193.137.9.178	HTTP	907 GET /images/tab/um_pt-PT.gif HTTP/1.1
52	0.237282	193.137.9.178	172.26.119.109	HTTP	1261 HTTP/1.1 200 OK (GIF89a)
68	0.252004	172.26.119.109	193.137.9.178	HTTP	907 GET /images/tab/servicos.gif HTTP/1.1
72	0.254039	193.137.9.178	172.26.119.109	HTTP	1162 HTTP/1.1 200 OK (text/css)
74	0.258193	193.137.9.178	172.26.119.109	HTTP	944 HTTP/1.1 200 OK (GIF89a)

Figura 14 - acesso com *hard refresh*

	Time	Source	Destination	Protocol	Length	Info
21	1.928426	172.26.119.109	193.137.9.178	HTTP	906	GET / HTTP/1.1
57	2.245194	193.137.9.178	172.26.119.109	HTTP	1142	HTTP/1.1 200 OK (text/html)

Figura 15 – acesso normal à página

Com a utilização do *hard refresh* é possível aceder ao servidor de forma mais rápida porque limpa a cache, porém caso seja necessário aceder a informação contida na cache pode originar erros e existe o risco de aceder a páginas não atualizadas.

## HTTPS

**2.11** Aceda a <https://elearning.uminho.pt>, ao mesmo tempo que captura o tráfego desse acesso com o Wireshark.

- a. De que forma o seu *browser* assinala que o utilizador está perante, ou não, uma ligação HTTP ao servidor segura? Apresente uma captura de écran com essa indicação.

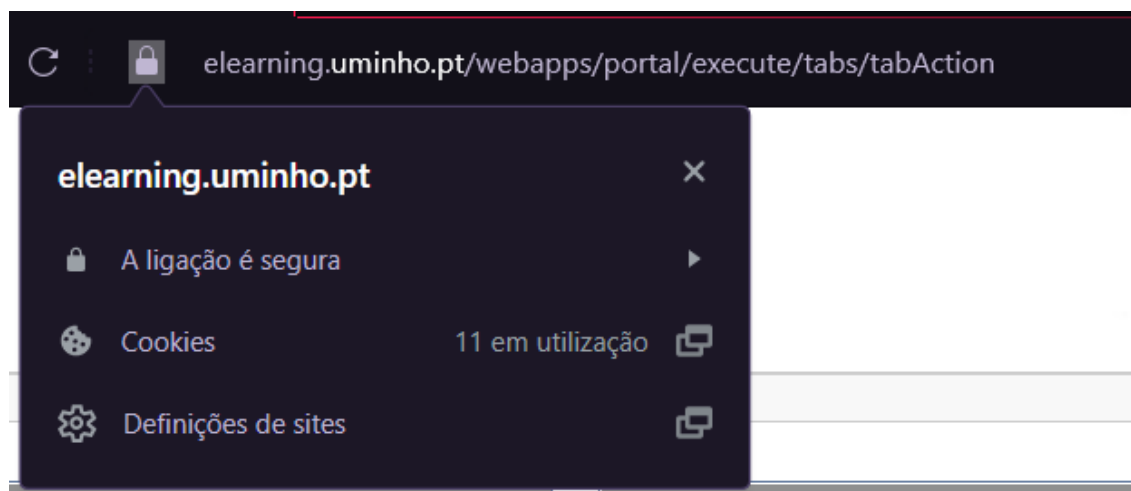


Figura 16

- b. Porque razão o tráfego HTTP não é identificado como tal no Wireshark? Apesar disso, pode detetar-se qual o protocolo aplicacional. Como é que o Wireshark sabe que se trata duma ligação *http-over-tls*?

6 0.001761	172.26.119.109	193.137.9.150	TLSv1.2	571 Client Hello
7 0.003369	193.137.9.150	172.26.119.109	TCP	54 [TCP Window Update] 443 → 56300 [ACK] Seq=1 Ack=1 Win=262144 Len=0
8 0.016153	172.26.119.109	193.137.9.150	TCP	66 56301 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
9 0.018323	193.137.9.150	172.26.119.109	TCP	66 443 → 56301 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1250 SACK_PERM WS=64
10 0.018428	172.26.119.109	193.137.9.150	TCP	54 56301 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
11 0.018580	172.26.119.109	193.137.9.150	TLSv1.2	571 Client Hello
12 0.021359	193.137.9.150	172.26.119.109	TCP	54 [TCP Window Update] 443 → 56301 [ACK] Seq=1 Ack=1 Win=262144 Len=0
13 0.021359	193.137.9.150	172.26.119.109	TCP	54 443 → 56300 [ACK] Seq=1 Ack=518 Win=262144 Len=0
14 0.038563	193.137.9.150	172.26.119.109	TCP	54 443 → 56301 [ACK] Seq=1 Ack=518 Win=262144 Len=0
15 0.075401	193.137.9.150	172.26.119.109	TLSv1.2	140 Server Hello
16 0.075401	193.137.9.150	172.26.119.109	TLSv1.2	60 Change Cipher Spec
17 0.075401	193.137.9.150	172.26.119.109	TLSv1.2	99 Encrypted Handshake Message
18 0.075461	172.26.119.109	193.137.9.150	TCP	54 56301 → 443 [ACK] Seq=518 Ack=138 Win=131072 Len=0
19 0.078506	172.26.119.109	193.137.9.150	TLSv1.2	105 Change Cipher Spec, Encrypted Handshake Message
20 0.078708	172.26.119.109	193.137.9.150	TCP	1304 56301 → 443 [ACK] Seq=569 Ack=138 Win=131072 Len=1250 [TCP segment of a reassembled PDU]
21 0.078708	172.26.119.109	193.137.9.150	TLSv1.2	125 Application Data

Figura 17

O tráfego HTTP não é identificado no Wireshark, porque é usado um protocolo adicional que não o http mas sim o protocolo TLSv1.2.

O wireshark sabe que se trata de uma ligação *http-over-tls* porque é feito um acesso ao site, através de um URL do tipo "https://..." em vez de "http://..."

- 2.12 Diga, justificando, quais dos seguintes elementos uma comunicação HTTPS permite manter ocultos dum atacante: *i)* o endereço IP do cliente, *ii)* o endereço IP do servidor *web*, *iii)* o nome do servidor *web*, *iv)* o tamanho da mensagem trocada entre o cliente o servidor, *v)* a identificação da página acedida no servidor *web*, *vi)* a frequência das conexões estabelecidas entre o cliente e o servidor, *vii)* os dados da aplicação trocados entre o servidor e o cliente.



Figura 18

Em sites com endereço HTTPS, a comunicação é criptografada ponto a ponto, aumentando significativamente a segurança dos dados. Devido então à criptografia o tamanho da mensagem trocada e os dados trocados entre o servidor e o cliente são protegidos pela ligação HTTPS.



### 3. Consultas ao serviço de resolução de nomes DNS

A maioria dos sistemas operativos (Windows, Linux, etc) inclui um cliente DNS genérico designado por *nslookup*. No entanto este cliente tem vindo a ser preterido a favor de outros como o *dig* e o *host*. O package *dnsutils* inclui todos. Se no Linux não conseguir usar nenhum deles tente reinstalar o package com o comando: *sudo apt-get install dnsutils*.

A base de dados dum servidor DNS é constituída por registos de diversos tipos, como por exemplo: *A*, *AAAA*, *NS*, *SOA*, *MX*, *PTR*. Usando o *nslookup* ou o *dig* e com base nos seus manuais (*man nslookup* ou *man dig*) procure responder às seguintes questões, devendo incluir os resultados que sustentam as suas respostas:

- 3.1 Se estiver a usar o Linux, observe o conteúdo do ficheiro */etc/resolv.conf*. Se estiver a usar o Windows, abra uma janela de comandos e execute *nslookup*. Indique qual o servidor de nomes que a sua máquina está a usar?

```
C:\Users\eduar>nslookup
Default Server:  dns3.uminho.pt
Address:  193.137.16.65
```

Figura 19

- 3.2 Usando o registo do tipo *A*, identifique os endereços IPv4 dos servidores *www.sas.uminho.pt*, *marco.uminho.pt* e *www.google.com*? Usando o registo *AAAA*, identifique o endereço IPv6 do servidor *www.fccn.pt*.

```
C:\Users\eduar>ping www.google.com -4

Pinging www.google.com [172.217.168.164] with 32 bytes of data:
Reply from 172.217.168.164: bytes=32 time=48ms TTL=114
Reply from 172.217.168.164: bytes=32 time=20ms TTL=114
Reply from 172.217.168.164: bytes=32 time=37ms TTL=114
Reply from 172.217.168.164: bytes=32 time=37ms TTL=114

Ping statistics for 172.217.168.164:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 20ms, Maximum = 48ms, Average = 35ms
```

Figura 20 - Ipv4 do google - 172.217.168.164

```
C:\Users\Guga>ping marco.uminho.pt -4

Pinging marco.uminho.pt [193.136.9.240] with 32 bytes of data:
Reply from 193.136.9.240: bytes=32 time=3ms TTL=61
Reply from 193.136.9.240: bytes=32 time=3ms TTL=61
Reply from 193.136.9.240: bytes=32 time=8ms TTL=61
Reply from 193.136.9.240: bytes=32 time=8ms TTL=61

Ping statistics for 193.136.9.240:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 3ms, Maximum = 8ms, Average = 5ms
```

Figura 21 - Ipv4 do marco.uminho.pt - 193.136.9.240

```
C:\Users\Guga>ping www.sas.uminho.pt -4

Pinging www.sas.uminho.pt [193.137.9.178] with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 193.137.9.178:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Figura 22 - Ipv4 do sas.uminho.pt - 193.137.9.178

```
C:\Users\eduar>nslookup
Default Server:  dns3.uminho.pt
Address:  193.137.16.65

> set type=AAAA
> www.fccn.pt
Server:  dns3.uminho.pt
Address:  193.137.16.65

Non-authoritative answer:
Name:      www.fccn.pt
Address:  2001:690:a00:1036:1113::247
```

Figura 23 – Ipv6 do www.fccn.pt



- 3.3 Experimente fazer uma *query* aos registos *PTR* para os nomes 240.9.136.193.in-addr.arpa. e 7.4.2.0.0.0.0.0.0.0.3.1.1.1.6.3.0.1.0.0.a.0.0.9.6.0.1.0.0.2.ip6.arpa. Comente os resultados face aos obtidos na alínea anterior.

```
9.136.193.in-addr.arpa
    primary name server = marco.uminho.pt
    responsible mail addr = dnsop.marco.uminho.pt
    serial = 2021040601
    refresh = 28800 (8 hours)
    retry = 7200 (2 hours)
    expire = 604800 (7 days)
    default TTL = 86400 (1 day)
>
```

Figura 24

```
> 7.4.2.0.0.0.0.0.0.0.3.1.1.1.6.3.0.1.0.0.a.0.0.9.6.0.1.0.0.2.ip6.arpa.
Server: dns3.uminho.pt
Address: 193.137.16.65

6.3.0.1.0.0.a.0.0.9.6.0.1.0.0.2.ip6.arpa
    primary name server = ns01.fccn.pt
    responsible mail addr = hostmaster.fccn.pt
    serial = 2022071501
    refresh = 21600 (6 hours)
    retry = 7200 (2 hours)
    expire = 604800 (7 days)
    default TTL = 14400 (4 hours)
```

Figura 25

Com base nos resultados obtidos na alínea anterior concluímos que foi possível aceder às informações do servidor através de uma pesquisa inversa.

- 3.4 Usando o registo *NS*, identifique os servidores de nomes definidos para os domínios: “uminho.com.”, “sas.uminho.pt.”, “pt.” e “.” (*root*). *i)* Perante a informação obtida, diga, justificando, se os servidores de nomes de diferentes domínios podem coexistir numa mesma máquina física. *ii)* Encontra domínios geridos por servidores de nomes localizados em redes IP distintas? Se sim, apresente esses domínios e diga qual a vantagem resultante desse procedimento?

```

C:\Users\eduar>nslookup
Default Server:  dns3.uminho.pt
Address:  193.137.16.65

> set type=NS
> uminho.com
Server:  dns3.uminho.pt
Address:  193.137.16.65

*** dns3.uminho.pt can't find uminho.com: Non-existent domain
> sas.uminho.pt
Server:  dns3.uminho.pt
Address:  193.137.16.65

sas.uminho.pt    nameserver = dns2.uminho.pt
sas.uminho.pt    nameserver = dns3.uminho.pt
sas.uminho.pt    nameserver = dns.uminho.pt
dns.uminho.pt    internet address = 193.137.16.75
dns2.uminho.pt   internet address = 193.137.16.145
dns3.uminho.pt   internet address = 193.137.16.65
dns.uminho.pt    AAAA IPv6 address = 2001:690:2280:1::75
dns2.uminho.pt   AAAA IPv6 address = 2001:690:2280:801::145
dns3.uminho.pt   AAAA IPv6 address = 2001:690:2280:1::65

```

Figura 26

```

> pt.
Server:  dns3.uminho.pt
Address:  193.137.16.65

Non-authoritative answer:
pt        nameserver = e.dns.pt
pt        nameserver = h.dns.pt
pt        nameserver = ns.dns.br
pt        nameserver = d.dns.pt
pt        nameserver = g.dns.pt
pt        nameserver = a.dns.pt
pt        nameserver = ns2.nic.fr
pt        nameserver = c.dns.pt
pt        nameserver = b.dns.pt

ns2.nic.fr    internet address = 192.93.0.4
b.dns.pt      internet address = 194.0.25.23
g.dns.pt      internet address = 193.136.2.226
c.dns.pt      internet address = 204.61.216.105
e.dns.pt      internet address = 193.136.192.64
d.dns.pt      internet address = 185.39.210.1
a.dns.pt      internet address = 185.39.208.1
ns.dns.br     internet address = 200.160.0.5
h.dns.pt      internet address = 194.146.106.138
ns2.nic.fr    AAAA IPv6 address = 2001:660:3005:1::1:2
b.dns.pt      AAAA IPv6 address = 2001:678:20::23
g.dns.pt      AAAA IPv6 address = 2001:690:a80:4001::100
c.dns.pt      AAAA IPv6 address = 2001:500:14:6105:ad::1
e.dns.pt      AAAA IPv6 address = 2001:690:a00:4001::64

```

Figura 27

```

> .
Server: dns3.uminho.pt
Address: 193.137.16.65

Non-authoritative answer:
(root) nameserver = b.root-servers.net
(root) nameserver = g.root-servers.net
(root) nameserver = c.root-servers.net
(root) nameserver = d.root-servers.net
(root) nameserver = m.root-servers.net
(root) nameserver = j.root-servers.net
(root) nameserver = l.root-servers.net
(root) nameserver = e.root-servers.net
(root) nameserver = k.root-servers.net
(root) nameserver = a.root-servers.net
(root) nameserver = f.root-servers.net
(root) nameserver = i.root-servers.net
(root) nameserver = h.root-servers.net

i.root-servers.net      internet address = 192.36.148.17
j.root-servers.net      internet address = 192.58.128.30
g.root-servers.net      internet address = 192.112.36.4
d.root-servers.net      internet address = 199.7.91.13
c.root-servers.net      internet address = 192.33.4.12
f.root-servers.net      internet address = 192.5.5.241
m.root-servers.net      internet address = 202.12.27.33
b.root-servers.net      internet address = 199.9.14.201
a.root-servers.net      internet address = 198.41.0.4
h.root-servers.net      internet address = 198.97.190.53
e.root-servers.net      internet address = 192.203.230.10
l.root-servers.net      internet address = 199.7.83.42
k.root-servers.net      internet address = 193.0.14.129
i.root-servers.net      AAAA IPv6 address = 2001:7fe::53
j.root-servers.net      AAAA IPv6 address = 2001:503:c27::2:30

```

Figura 28

Nome dos servidores - uminho.com: erro

Nome dos servidores - sas.uminho.pt: dns2.uminho.pt; dns3.uminho.pt; dns.uminho.pt

Nome dos servidores - pt: e.dns.pt; h.dns.pt; ns.dns.br;...

Nome dos servidores - ".":b.root-servers.net; g.root-servers.net; c.root-servers.net;...

i) A mesma máquina pode estar a alojar vários servidores de nomes diferentes.

ii) Sim, por exemplo no caso do "pt" está alocado em domínios ".pt", ".br" e ".fr", por exemplo. Este processo é vantajoso porque aumenta o alcance e a sua visibilidade na internet.

**3.5 Usando o registo SOA, identifique o servidor DNS primário definido para os domínios “uminho.pt.”, “pt.” e “.”? Em que difere o servidor primário de um servidor secundário e qual o significado dos parâmetros temporais associados ao servidor primário?**

```
C:\Users\eduar>nslookup
Default Server:  dns3.uminho.pt
Address:  193.137.16.65

> set type=SOA
> uminho.pt
Server:  dns3.uminho.pt
Address:  193.137.16.65

uminho.pt
    primary name server = dns.uminho.pt
    responsible mail addr = servicos.scom.uminho.pt
    serial = 2022121501
    refresh = 14400 (4 hours)
    retry = 7200 (2 hours)
    expire = 1209600 (14 days)
    default TTL = 300 (5 mins)
uminho.pt      nameserver = dns.uminho.pt
uminho.pt      nameserver = dns2.uminho.pt
uminho.pt      nameserver = ns02.fccn.pt
uminho.pt      nameserver = dns3.uminho.pt
dns.uminho.pt  internet address = 193.137.16.75
dns2.uminho.pt internet address = 193.137.16.145
dns3.uminho.pt internet address = 193.137.16.65
ns02.fccn.pt   internet address = 193.136.2.228
dns.uminho.pt  AAAA IPv6 address = 2001:690:2280:1::75
dns2.uminho.pt AAAA IPv6 address = 2001:690:2280:801::145
dns3.uminho.pt AAAA IPv6 address = 2001:690:2280:1::65
ns02.fccn.pt   AAAA IPv6 address = 2001:690:a80:4001::200
>
```

Figura 4 – servidor primário – dns.uminho.pt

```
> pt.
Server:  dns3.uminho.pt
Address:  193.137.16.65

Non-authoritative answer:
pt
    primary name server = curiosity.dns.pt
    responsible mail addr = request.dns.pt
    serial = 2022121630
    refresh = 21600 (6 hours)
    retry = 7200 (2 hours)
    expire = 2592000 (30 days)
    default TTL = 300 (5 mins)

pt      nameserver = d.dns.pt
pt      nameserver = ns2.nic.fr
pt      nameserver = a.dns.pt
pt      nameserver = g.dns.pt
pt      nameserver = c.dns.pt
pt      nameserver = b.dns.pt
pt      nameserver = ns.dns.br
pt      nameserver = e.dns.pt
pt      nameserver = h.dns.pt
a.dns.pt      internet address = 185.39.208.1
b.dns.pt      internet address = 194.0.25.23
c.dns.pt      internet address = 204.61.216.105
d.dns.pt      internet address = 185.39.210.1
e.dns.pt      internet address = 193.136.192.64
g.dns.pt      internet address = 193.136.2.226
h.dns.pt      internet address = 194.146.106.138
```

Figura 5 – servidor primário – curiosity.dns.pt

```
Non-authoritative answer:
(root)      primary name server = a.root-servers.net
            responsible mail addr = nstld.verisign-grs.com
            serial    = 2022121600
            refresh   = 1800 (30 mins)
            retry     = 900 (15 mins)
            expire    = 604800 (7 days)
            default TTL = 86400 (1 day)

(root) nameserver = l.root-servers.net
(root) nameserver = b.root-servers.net
(root) nameserver = d.root-servers.net
(root) nameserver = g.root-servers.net
(root) nameserver = a.root-servers.net
(root) nameserver = f.root-servers.net
(root) nameserver = j.root-servers.net
(root) nameserver = e.root-servers.net
(root) nameserver = i.root-servers.net
(root) nameserver = k.root-servers.net
(root) nameserver = m.root-servers.net
(root) nameserver = c.root-servers.net
(root) nameserver = h.root-servers.net
i.root-servers.net      internet address = 192.36.148.17
j.root-servers.net      internet address = 192.58.128.30
g.root-servers.net      internet address = 192.112.36.4
d.root-servers.net      internet address = 199.7.91.13
c.root-servers.net      internet address = 192.33.4.12
f.root-servers.net      internet address = 192.5.5.241
```

Figura 6 – servidor primário – a.root-servers.net

A diferença entre DNS primário e secundário é que o primário hospeda o arquivo onde estão contidas todas as informações relevantes para um domínio, como o número de IP. Já o servidor secundário conta com uma cópia somente de leitura desse arquivo, que é fornecida pelo servidor primário por meio de uma comunicação chamada de transferência de zona. O servidor secundário serve para estabelecer uma conexão caso haja uma falha do servidor primário. Os parâmetros associados ao servidor primário indicam o tempo que falta para o próximo refresh, retry, expire e default TTL.

**3.6 Usando o registo MX, diga qual(uais) o(s) servidor(s) de email do domínio *edu.ulisboa.pt*?  
A que sistema são entregues preferencialmente as mensagens dirigidas a *geral@edu.ulisboa.pt*?**

```
C:\Users\eduar>nslookup
Default Server:  dns3.uminho.pt
Address:  193.137.16.65

> set type=MX
> edu.ulisboa.pt
Server:  dns3.uminho.pt
Address:  193.137.16.65

Non-authoritative answer:
edu.ulisboa.pt  MX preference = 1, mail exchanger = ASPMX.L.GOOGLE.COM
edu.ulisboa.pt  MX preference = 5, mail exchanger = ALT2.ASPMX.L.GOOGLE.COM
edu.ulisboa.pt  MX preference = 10, mail exchanger = ASPMX2.GOOGLEMAIL.COM
edu.ulisboa.pt  MX preference = 5, mail exchanger = ALT1.ASPMX.L.GOOGLE.COM
edu.ulisboa.pt  MX preference = 10, mail exchanger = ASPMX3.GOOGLEMAIL.COM

ulisboa.pt      nameserver = ns2.tecnico.ulisboa.pt
ulisboa.pt      nameserver = b.ul.pt
ulisboa.pt      nameserver = a.ul.pt
ulisboa.pt      nameserver = ns1.tecnico.ulisboa.pt
alt1.aspmx.l.google.com internet address = 142.250.153.27
alt2.aspmx.l.google.com internet address = 142.250.147.26
aspmx.l.google.com   internet address = 74.125.71.27
ns2.tecnico.ulisboa.pt internet address = 193.136.128.2
a.ul.pt internet address = 194.117.0.150
b.ul.pt internet address = 194.117.1.150
ns1.tecnico.ulisboa.pt internet address = 193.136.128.1
alt1.aspmx.l.google.com AAAA IPv6 address = 2a00:1450:4013:c16::1b
alt2.aspmx.l.google.com AAAA IPv6 address = 2a00:1450:4025:c01::1a
aspmx.l.google.com      AAAA IPv6 address = 2a00:1450:400c:c08::1a
>
```

Figura 27

Pela imagem podemos concluir que há 5 mails exchanger:

ASPMX.L.GOOGLE.COM

ALT2.ASPMX.L.GOOGLE.COM

ASPMX2.GOOGLEMAIL.COM

ALT1.ASPMX.L.GOOGLE.COM

ASPMX3.GOOGLEMAIL.COM

**3.7 Usando o registo CNAME, diga qual(uais) o(s) *aliases* do nome *www.ebay.com*? O que é que isso significa?**

```
C:\Users\eduar>nslookup www.ebay.com
Server:  dns3.uminho.pt
Address:  193.137.16.65

Non-authoritative answer:
Name:     e9428.a.akamaiedge.net
Address:  23.49.245.22
Aliases:  www.ebay.com
          slot9428.ebay.com.edgekey.net
```

Figura 28

Aliases: *www.ebay.com* e *slot9428.ebay.com.edgekey.net*

Um registo CNAME é um tipo de registo DNS que mapeia um nome de aliases para um nome de domínio verdadeiro.

**3.8 Qual a diferença entre uma resposta adjetivada como *non-authoritative answer* (“não-autoritativa”) e uma resposta “autoritativa” para uma determinada *query*?**

Uma resposta autoritativa vem de um servidor de DNS oficial do domínio que procuramos.

Uma resposta não autoritativa vem de um servidor não oficial de DNS para o domínio em questão.

**4. Uso da camada de transporte por parte das aplicações**

Verifique se na sua máquina de trabalho tem disponíveis as seguintes aplicações ou ferramentas: clientes *ftp*, *ssh*, *traceroute* (*tracert* em Windows), *ping* e *telnet*, senão instale. Corra novamente o Wireshark. Capturando o tráfego nos momentos que considere adequados, observe atentamente como as várias aplicações utilizam o serviço de transporte, quando é efetuado:

1 0.000000	172.26.100.232	172.217.168.164	TLSv1.2	277 Application Data
2 0.000040	172.26.100.232	172.217.168.164	TLSv1.2	93 Application Data
3 0.000064	172.26.100.232	172.217.168.164	TLSv1.2	371 Application Data
4 0.014637	172.217.168.164	172.26.100.232	TCP	60 443 → 56386 [ACK] Seq=1 Ack=224 Win=329 Len=0
5 0.014637	172.217.168.164	172.26.100.232	TCP	60 443 → 56386 [ACK] Seq=1 Ack=263 Win=329 Len=0
6 0.014637	172.217.168.164	172.26.100.232	TCP	60 443 → 56386 [ACK] Seq=1 Ack=580 Win=339 Len=0
7 0.016039	172.217.168.164	172.26.100.232	TLSv1.2	93 Application Data
8 0.062541	172.26.100.232	172.217.168.164	TCP	54 56386 → 443 [ACK] Seq=580 Ack=40 Win=512 Len=0
9 0.071962	172.217.168.164	172.26.100.232	TLSv1.2	1192 Application Data
10 0.071962	172.217.168.164	172.26.100.232	TLSv1.2	759 Application Data
11 0.071962	172.217.168.164	172.26.100.232	TLSv1.2	86 Application Data
12 0.071962	172.217.168.164	172.26.100.232	TLSv1.2	759 Application Data
13 0.071962	172.217.168.164	172.26.100.232	TLSv1.2	86 Application Data
14 0.071962	172.217.168.164	172.26.100.232	TLSv1.2	85 Application Data
15 0.071962	172.217.168.164	172.26.100.232	TLSv1.2	93 Application Data

Figura 29

13 2.413050	172.26.100.232	172.217.168.164	TLSv1.2	89 Application Data
14 2.415573	172.217.168.164	172.26.100.232	TLSv1.2	86 Application Data
15 2.415573	172.217.168.164	172.26.100.232	TLSv1.2	85 Application Data
16 2.415573	172.217.168.164	172.26.100.232	TLSv1.2	93 Application Data
17 2.415601	172.26.100.232	172.217.168.164	TCP	54 56386 → 443 [ACK] Seq=524 Ack=809 Win=509 Len=0
18 2.415711	172.26.100.232	172.217.168.164	TLSv1.2	93 Application Data
19 2.438775	172.217.168.164	172.26.100.232	TCP	60 443 → 56386 [ACK] Seq=809 Ack=563 Win=417 Len=0
20 2.737444	172.26.100.232	193.137.16.65	DNS	77 Standard query 0xc9c0 A fonts.gstatic.com
21 2.737581	172.26.100.232	193.137.16.65	DNS	77 Standard query 0xc9c0 HTTPS fonts.gstatic.com
22 2.739151	193.137.16.65	172.26.100.232	DNS	348 Standard query response 0xc9c0 A fonts.gstatic.com A 142.250.200.67 NS ns1.google.com NS ns4.google.com
23 2.739151	193.137.16.65	172.26.100.232	DNS	134 Standard query response 0xc9c0 HTTPS fonts.gstatic.com SOA ns1.google.com
24 2.739738	172.26.100.232	142.250.200.67	TCP	66 56409 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
25 2.741392	172.26.100.232	172.217.168.164	TCP	1304 56386 → 443 [ACK] Seq=563 Ack=809 Win=509 Len=1250 [TCP segment of a reassembled PDU]
26 2.741392	172.26.100.232	172.217.168.164	TLSv1.2	642 Application Data
27 2.756006	172.217.168.164	172.26.100.232	TCP	60 443 → 56386 [ACK] Seq=809 Ack=2401 Win=436 Len=0

Figura 30





```
C:\Users\fabio>tracert router-di.uminho.pt

Tracing route to router-di.uminho.pt [193.136.9.254]
over a maximum of 30 hops:

  1    25 ms    6 ms    3 ms  172.26.254.254
  2     1 ms    1 ms    1 ms  172.16.2.1
  3     1 ms    2 ms    3 ms  router-di.uminho.pt [193.136.9.254]

Trace complete.

C:\Users\fabio>
```

[illegible]

**4.1 Preencha a seguinte tabela com base nos resultados que obteve:**

- Acesso via browser a *http://www.sas.uminho.pt*
- Acesso via browser a *https://elearning.uminho.pt*
- Acesso em ftp para *ftp.di.uminho.pt* (login: *anonymous*)
- ping cisco.di.uminho.pt*
- Acesso ssh para *marco.uminho.pt*
- nslookup www.fccn.pt*
- traceroute router-di.uminho.pt*
- telnet freechess.org*

Comando/aplicação	Canal seguro?	Protocolo de transporte (se aplicável)	Porta de atendimento (se aplicável)
browser http://	NÃO SEGURO	TCP	80
browser https://	SEGURO	TCP	443
ftp	SEGURO	TCP	21
ping	SEGURO	NÃO APLICÁVEL	NÃO APLICÁVEL
ssh	SEGURO	TCP	22
nslookup / dig	SEGURO	UDP	53
tracert	NÃO SEGURO	UDP	53
telnet	NÃO SEGURO	TCP	23

Figura 37 – Tabela questão 4.1 (Esta tabela foi feita no paint e colocado aqui um print recortado da mesma)

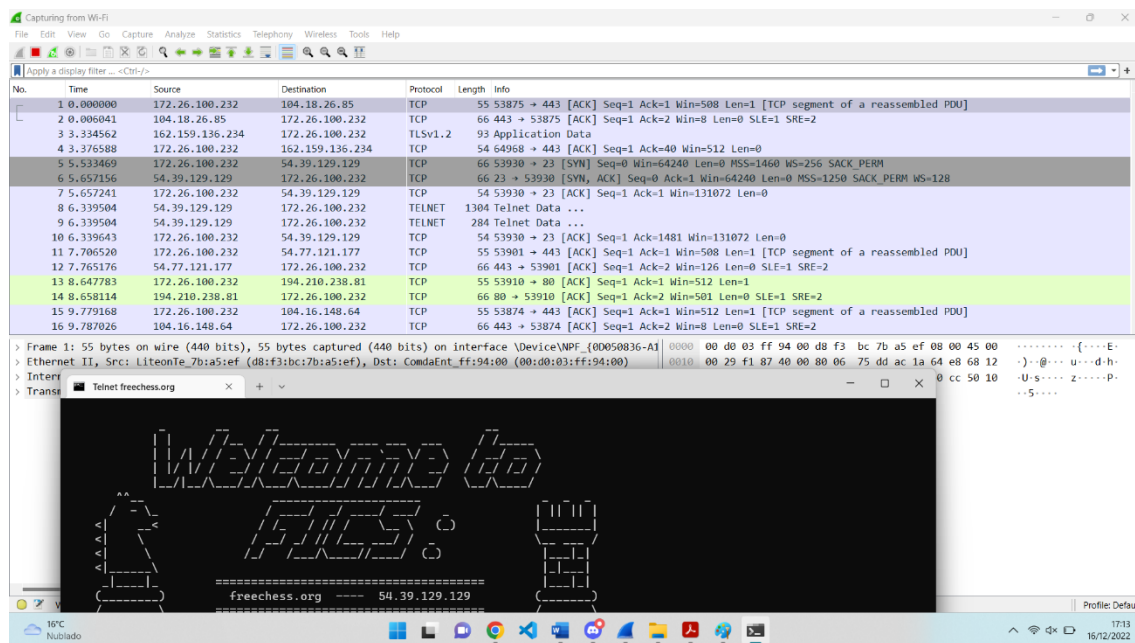


Figura 38

▼ User Datagram Protocol, Src Port: 53, Dst Port: 53

Figura 39

110	5.006411	193.137.9.178	172.26.100.232	TCP	170	HTTP/1.1 200 OK (text/html)
111	5.006504	172.26.100.232	193.137.9.178	TCP	54	56396 → 80 [ACK] Seq=495 A
112	5.014368	172.26.100.232	193.137.16.65	DNS	87	Standard query 0xa5b3 A sa
113	5.014538	172.26.100.232	193.137.16.65	DNS	87	Standard query 0xa577 HTTP
114	5.014709	172.26.100.232	193.137.9.178	HTTP	491	GET /portal.css HTTP/1.1
115	5.015760	193.137.16.65	172.26.100.232	DNS	358	Standard query response 0x

Frame 110: 170 bytes on wire (1360 bits), 170 bytes captured (1360 bits) on interface \Device\NPF\_{0D0... Ethernet II, Src: Cisco\_ab:ac:cf (90:77:ee:ab:ac:cf), Dst: LiteonTe\_7b:a5:ef (d8:f3:bc:7b:a5:ef) Internet Protocol Version 4, Src: 193.137.9.178, Dst: 172.26.100.232 Transmission Control Protocol, Src Port: 80, Dst Port: 56396, Seq: 33751, Ack: 495, Len: 116

Figura 40

✓ Transmission Control Protocol, Src Port: 443, Dst Port: 53778, Seq: 482, Ack: 283, Len: 443  
Source Port: 443

Figura 41

#### 4.2 Comente as principais diferenças entre os protocolos TCP e UDP. Relacione-as com as experiências realizadas onde observou os campos dos cabeçalhos respectivos e o *overhead* protocolar. Em particular, identifique os campos do TCP responsáveis pelo controlo de fluxo, ordenação e fiabilidade do protocolo.

TCP é o protocolo mais usado porque fornece garantia na entrega de todos os pacotes.

No estabelecimento de ligação dos remetentes existe um "pré-acordo", tornando-o mais seguro e mais fiável nas comunicações, contudo é mais lento pois necessita verificações. Se um pacote se perder existe solicitação de reenvio. Se um pacote apresentar erros existe correção dos mesmos.

Os pacotes têm de ser enviados em sequência neste protocolo.

UDP é um protocolo mais simples, não fornece garantia na entrega dos pacotes.

Sendo mais rápido entre os dois, porém o menos seguro. Se um pacote se perder não existe solicitação de reenvio. Se um pacote apresentar erros ao contrário do TCP não existe correção dos mesmos. Os pacotes são enviados em fluxo neste protocolo.

## 5. Conclusões

Com a realização deste trabalho prático, foi-nos permitido aprofundar e consolidar assuntos relacionados com as diversas camadas ou níveis, tendo esta proposta de trabalho o objetivo de nos familiarizar com protocolos e ferramentas do nível aplicacional e análise dos protocolos de transporte.

Ao longo da realização do trabalho deparámo-nos com diversas dificuldades, que tentámos esclarecer, fazendo diversas pesquisas, tanto em páginas web, como nos documentos fornecidos pelo docente e até por vezes recorrendo a tentativa/erro. Achamos também que o trabalho pedido era um pouco extenso demais. Porém apesar disso, acreditámos que conseguimos alcançar com sucesso a maioria dos objetivos propostos pelo trabalho.