

# TP3: Aplicações e Camada de Transporte

Ana Silva (a91678), Paulo Freitas (a100053) e Rúben Machado (a91656)

## Questões e Respostas

### 1 Nível aplicacional

- 1) Identifique o endereço IP da estação que formulou a *query* DNS e o tipo de *query* realizada.

O endereço IP 172.26.42.81 (Fig.1 – vermelho) da estação formulou a *query* DNS, sendo do tipo A (Fig.1 – verde)

1	0.000000	172.26.42.81	193.137.16.65	DNS	82	Standard query 0x30ef A wpad.eduroam.uminho.pt
2	0.001753	193.137.16.65	172.26.42.81	DNS	136	Standard query response 0x30ef No such name A wpad.eduroam.uminho.pt SOA dns.uminho.pt
3	0.021918	172.26.42.81	130.117.190.213	TLSv...	132	Application Data
4	0.021997	172.26.42.81	130.117.190.213	TLSv...	336	Application Data
5	0.022246	172.26.42.81	142.250.185.3	TLSv...	311	Application Data
6	0.156973	142.250.185.3	172.26.42.81	TCP	60	443 → 58522 [ACK] Seq=1 Ack=258 Win=472 Len=0
7	0.186350	142.250.185.3	172.26.42.81	TLSv...	120	Application Data
8	0.186350	142.250.185.3	172.26.42.81	TLSv...	85	Application Data
9	0.186350	142.250.185.3	172.26.42.81	TLSv...	93	Application Data
10	0.186396	172.26.42.81	142.250.185.3	TCP	54	58522 → 443 [ACK] Seq=258 Ack=137 Win=512 Len=0
11	0.189170	130.117.190.213	172.26.42.81	TCP	60	443 → 58242 [ACK] Seq=1 Ack=361 Win=63292 Len=0
12	0.189170	130.117.190.213	172.26.42.81	TLSv...	109	Application Data
13	0.189170	130.117.190.213	172.26.42.81	TLSv...	165	Application Data
14	0.189170	130.117.190.213	172.26.42.81	TLSv...	152	Application Data
15	0.189204	172.26.42.81	130.117.190.213	TCP	54	58242 → 443 [ACK] Seq=361 Ack=265 Win=64214 Len=0
16	0.189939	172.26.42.81	142.250.185.3	TLSv...	93	Application Data
17	0.214241	142.250.185.3	172.26.42.81	TCP	60	443 → 58522 [ACK] Seq=137 Ack=297 Win=472 Len=0
18	0.309809	172.26.42.81	52.111.231.2	TLSv...	82	Application Data
19	0.549097	52.111.231.2	172.26.42.81	TCP	60	443 → 58368 [ACK] Seq=1 Ack=29 Win=16385 Len=0
20	0.929750	172.26.42.81	52.111.231.2	TLSv...	82	Application Data
21	1.034825	52.111.231.2	172.26.42.81	TCP	60	443 → 58371 [ACK] Seq=1 Ack=29 Win=16385 Len=0
22	1.462731	172.26.42.81	63.303.206.11	TCP	64	58522 → 443 [ACK] Seq=1 Ack=1 Win=65536 Len=0

> Frame 1: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface \Device\NPF\_{7D78C8DD-4D32-45ED-B5CC-43A2} (00:00:00:00:00:00)

> Ethernet II, Src: IntelCor\_31:25:fe (0c:dd:34:31:25:fe), Dst: ComdEnt\_ff94:00 (00:00:03:ff:94:00)

> Internet Protocol Version 4, Src: 172.26.42.81, Dst: 193.137.16.65

> User Datagram Protocol, Src Port: 63769, Dst Port: 53

▼ Domain Name System (query)

Transaction ID: 0x30ef> Flags: 0x0100 Standard queryQuestions: 1Answer RRs: 0Authority RRs: 0Additional RRs: 0▼ Queries> wpad.eduroam.uminho.pt [type A] class IN [Response in 2]

0000 00 00 03 ff 94 00 0c dd 34 31 25 fe 08 00 45 00 .....1% E  
0010 00 44 c8 0b 00 00 80 11 00 00 ac 1a 2a 51 c1 89 .....Q  
0020 10 41 f9 19 00 35 00 30 a8 77 30 ef 01 00 00 01 .....A:5 0 v0  
0030 00 00 00 00 00 00 04 77 70 61 64 07 65 64 75 72 .....w pad edur  
0040 ef 61 6d 06 75 6d 69 6e 68 6f 02 70 74 00 00 01 .....oam umin ho pt  
0050 00 01

Fig. 1. Campo Domain Name System numa query DNS

- 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.

O endereço IP 172.26.42.81 (Fig.2 -vermelho) corresponde ao servidor *web*. O servidor dns3.uminho.pt (Fig.2 – verde) forneceu a resposta, com o endereço IP 193.137.16.65.

No.	Time	Source	Destination	Protoc	Length	BSS Id	Info
1	0.000000	172.26.42.81	193.137.16.65	DNS	82		Standard query 0x30ef A wpad.eduroam.uminho.pt
2	0.001753	193.137.16.65	172.26.42.81	DNS	136		Standard query response 0x30ef No such name A wpad.eduroam.uminho.pt SOA dns.uminho.pt
172	0.032548	172.26.42.81	193.137.16.65	DNS	87		Standard query 0xd811 A safebrowsing.googleapis.com
174	0.033031	172.26.42.81	193.137.16.65	DNS	87		Standard query 0xe458 HTTPS safebrowsing.googleapis.com
176	0.033291	193.137.16.65	172.26.42.81	DNS	358		Standard query response 0xd811 A safebrowsing.googleapis.com A 216.58.215.138 NS ns4.google.com NS n
177	0.033291	193.137.16.65	172.26.42.81	DNS	144		Standard query response 0xe458 HTTPS safebrowsing.googleapis.com SOA ns1.google.com
999	9.737312	172.26.42.81	193.137.16.65	DNS	77		Standard query 0x8b2e A www.sas.uminho.pt
1000	9.737367	172.26.42.81	193.137.16.65	DNS	77		Standard query 0x9c51 HTTPS www.sas.uminho.pt
1012	9.765107	193.137.16.65	172.26.42.81	DNS	131		Standard query response 0x9c51 HTTPS www.sas.uminho.pt SOA dns.uminho.pt
1013	9.765107	193.137.16.65	172.26.42.81	DNS	281		Standard query response 0x8b2e A www.sas.uminho.pt A 193.137.9.178 NS dns2.uminho.pt NS dns3.uminho.pt
1083	9.836057	172.26.42.81	193.137.16.65	DNS	83		Standard query 0xd88 A safebrowsing.google.com
1084	9.836247	172.26.42.81	193.137.16.65	DNS	83		Standard query 0xb73 HTTPS safebrowsing.google.com
1085	9.838854	193.137.16.65	172.26.42.81	DNS	366		Standard query response 0xd88 A safebrowsing.google.com CNAME sb1.google.com A 142.250.200.142 NS ns1.google.com NS
1086	9.838854	193.137.16.65	172.26.42.81	DNS	152		Standard query response 0xb73 HTTPS safebrowsing.google.com CNAME sb1.google.com SOA ns1.google.com
1435	10.674406	172.26.42.81	193.137.16.65	DNS	92		Standard query 0xb9a3 HTTPS gclid.v2.scr.kaspersky-labs.com
1432	10.784125	193.137.16.65	172.26.42.81	DNS	174		Standard query response 0xb9a3 HTTPS gclid.v2.scr.kaspersky-labs.com SOA dnsmaster.kaspersky-labs.net
1455	10.794146	172.26.42.81	193.137.16.65	ICMP	202		Destination unreachable (Port unreachable)
1541	10.912395	172.26.42.81	193.137.16.65	DNS	83		Standard query 0x24a0 A www.dicas.sas.uminho.pt
1542	10.912569	172.26.42.81	193.137.16.65	DNS	83		Standard query 0xf3d0 HTTPS www.dicas.sas.uminho.pt
1597	10.994260	193.137.16.65	172.26.42.81	DNS	151		Standard query response 0xf3d0 HTTPS www.dicas.sas.uminho.pt CNAME dicas.sas.uminho.pt SOA dns.uminho.pt
1598	10.994368	193.137.16.65	172.26.42.81	DNS	301		Standard query response 0x24a0 A www.dicas.sas.uminho.pt CNAME dicas.sas.uminho.pt A 193.137.88.166 NS dns2.uminho.pt NS

Domain Name System (response)

Transaction ID: 0xb02e

Flags: 0x8580 Standard query response, No error

Questions: 1

Answer RRs: 1

Authority RRs: 3

Additional RRs: 6

Queries

Answers

- www.sas.uminho.pt type A, class IN, addr: 193.137.9.178
- Authoritative nameservers
  - sas.uminho.pt type NS, class IN, ns dns2.uminho.pt
  - sas.uminho.pt type NS, class IN, ns dns3.uminho.pt
  - sas.uminho.pt type NS, class IN, ns dns3.uminho.pt
- Additional records
  - dns.uminho.pt type A, class IN, addr: 193.137.16.75
  - dns2.uminho.pt type A, class IN, addr: 193.137.16.145
  - dns3.uminho.pt type A, class IN, addr: 193.137.16.65
  - dns.uminho.pt type AAAA, class IN, addr: 2001:690:2280:1:75
  - dns2.uminho.pt type AAAA, class IN, addr: 2001:690:2280:1:145
  - dns3.uminho.pt type AAAA, class IN, addr: 2001:690:2280:1:65

```

0000 0c d4 24 31 25 fe 00 00 03 ff 84 00 08 00 45 00 11% .....E
0010 01 00 15 78 00 00 3f 11 bd 23 c1 89 10 41 ec 1a y P 3 A
0020 2a 51 00 35 d2 84 00 f7 9f 0f 8b 2e 85 80 00 01 *Q S .....
0030 00 01 00 03 00 06 03 77 77 03 73 61 73 06 75 www sas u
0040 6d 69 6e 68 0f 02 70 74 00 00 01 00 01 c0 0c 00 minho pt
0050 01 00 01 00 00 03 84 00 04 c1 89 09 b2 c0 10 00 .....
0060 02 00 01 00 01 51 80 00 07 04 64 06 73 32 c0 14 Q dns2
0070 c0 10 00 02 00 01 00 01 51 80 00 06 03 64 6e 73 Q dns
0080 c0 14 c0 10 00 02 00 01 00 01 51 80 00 07 04 64 Q d
0090 6e 73 32 c0 14 c0 32 00 01 00 01 00 00 38 40 00 ns3 R
00a0 04 c1 89 10 40 c0 3f 00 01 00 01 00 00 38 40 00 k 7 B
00b0 04 c1 89 10 41 64 00 01 00 01 00 00 38 40 00 R .....
00c0 10 20 01 0e 02 80 00 01 00 00 00 00 00 00 00 10 20 01 0e 02 80 00 01 00 00 00 00 00 00
00d0 75 c0 3f 00 1c 00 01 00 00 38 40 00 10 20 01 06 u ? B
00e0 90 22 80 08 01 00 00 00 00 00 00 01 45 c0 64 00 E d
00f0 1c 00 01 00 00 38 40 00 10 20 01 06 00 02 80 00 .....
0100 01 00 00 00 00 00 00 65 .....e

```

Fig. 2. Nome de servidores e os seus respectivos endereços IP

## 1.1 HTTP e TCP

- 3) Aplique o filtro aos protocolos *http* // *tcp*. Identifique os endereços IP do cliente e do servidor HTTP.

O endereço IP do cliente corresponde ao endereço IP de origem, enquanto o endereço IP do servidor HTTP corresponde ao endereço IP de destino, ou seja, os endereços 172.26.42.81 e 193.137.9.178, respetivamente (Fig. 3).

No.	Time	Source	Destination	Protoc	Length	BSS Id	Info
1005	9.745011	172.26.42.81	142.250.201.74	TLsv...	119		Application Data
1006	9.745057	172.26.42.81	142.250.201.74	TLsv...	374		Application Data
1007	9.763988	142.250.201.78	172.26.42.81	TCP	11304		443 → 58541 [PSH, ACK] Seq=1879017 Ack=15682 Win=685 Len=11250 [TCP segment of a reassembled PDU]
1008	9.763988	142.250.201.78	172.26.42.81	TLsv...	1291		Application Data, Application Data
1009	9.763988	142.250.201.78	172.26.42.81	TCP	1304		443 → 58541 [ACK] Seq=1891504 Ack=15682 Win=685 Len=1250 [TCP segment of a reassembled PDU]
1010	9.763988	142.250.201.78	172.26.42.81	TLsv...	972		Application Data
1011	9.764191	172.26.42.81	142.250.201.78	TCP	54		58541 → 443 [ACK] Seq=24009 Ack=1893672 Win=1025 Len=0
1014	9.765816	172.26.42.81	193.137.9.178	TCP	66		58551 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
1015	9.766309	172.26.42.81	193.137.9.178	TCP	66		58552 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
1016	9.766866	142.250.201.78	172.26.42.81	TCP	3804		443 → 58541 [ACK] Seq=1893672 Ack=15682 Win=685 Len=3750 [TCP segment of a reassembled PDU]
1017	9.766866	142.250.201.78	172.26.42.81	TLsv...	463		Application Data
1018	9.766866	142.250.201.78	172.26.42.81	TLsv...	1061		Application Data
1019	9.767048	172.26.42.81	142.250.201.78	TCP	54		58541 → 443 [ACK] Seq=24009 Ack=1898838 Win=1025 Len=0
1020	9.767834	172.26.42.81	142.250.201.78	TLsv...	89		Application Data
1021	9.767896	193.137.9.178	172.26.42.81	TCP	66		80 → 58551 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1250 WS=1 SACK_PERM
1022	9.767989	172.26.42.81	193.137.9.178	TCP	54		58551 → 80 [ACK] Seq=1 Ack=1 Win=131072 Len=0
1023	9.768876	172.26.42.81	193.137.9.178	HTTP	583		GET / HTTP/1.1
1024	9.773979	142.250.201.78	172.26.42.81	TCP	1304		443 → 58541 [ACK] Seq=1898838 Ack=15682 Win=685 Len=1250 [TCP segment of a reassembled PDU]
1025	9.773979	142.250.201.78	172.26.42.81	TLsv...	79		Application Data
1026	9.773979	142.250.201.78	172.26.42.81	TLsv...	1092		Application Data
1027	9.774115	172.26.42.81	142.250.201.78	TCP	54		58541 → 443 [ACK] Seq=24044 Ack=1901151 Win=1025 Len=0
1028	9.774896	193.137.9.178	172.26.42.81	TCP	66		80 → 58551 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1250 WS=1 SACK_PERM

Fig. 3. Endereços IP do cliente e do servidor HTTP

- 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?

O tamanho máximo de segmento (MSS) corresponde a 1460 *bytes* a MTU, em que apresenta uma taxa de transmissão de 1500 *bytes* e, como o cabeçalho do TCP e IP apresentam 20 *bytes* cada. Logo, a subtração entre a MTU e os cabeçalhos corresponde a 1460 *bytes* (Fig.4).

No.	Time	Source	Destination	Protoc	Length	BSS Id	Info
1005	9.745011	172.26.42.81	142.250.201.74	TLsv...	119		Application Data
1006	9.745057	172.26.42.81	142.250.201.74	TLsv...	374		Application Data
1007	9.763988	142.250.201.78	172.26.42.81	TCP	11304		443 → 58541 [PSH, ACK] Seq=1879017 Ack=15682 Win=685 Len=11250 [TCP segment of a reassembled PDU]
1008	9.763988	142.250.201.78	172.26.42.81	TLsv...	1291		Application Data, Application Data
1009	9.763988	142.250.201.78	172.26.42.81	TCP	1304		443 → 58541 [ACK] Seq=1891504 Ack=15682 Win=685 Len=1250 [TCP segment of a reassembled PDU]
1010	9.763988	142.250.201.78	172.26.42.81	TLsv...	972		Application Data
1011	9.764191	172.26.42.81	142.250.201.78	TCP	54		58541 → 443 [ACK] Seq=24009 Ack=1893672 Win=1025 Len=0
1014	9.765816	172.26.42.81	193.137.9.178	TCP	66		58551 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
1015	9.766309	172.26.42.81	193.137.9.178	TCP	66		58552 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
1016	9.766866	142.250.201.78	172.26.42.81	TCP	3804		443 → 58541 [ACK] Seq=1893672 Ack=15682 Win=685 Len=3750 [TCP segment of a reassembled PDU]
1017	9.766866	142.250.201.78	172.26.42.81	TLsv...	463		Application Data
1018	9.766866	142.250.201.78	172.26.42.81	TLsv...	1061		Application Data
1019	9.767048	172.26.42.81	142.250.201.78	TCP	54		58541 → 443 [ACK] Seq=24009 Ack=1898838 Win=1025 Len=0
1020	9.767834	172.26.42.81	142.250.201.78	TLsv...	89		Application Data
1021	9.767896	193.137.9.178	172.26.42.81	TCP	66		80 → 58551 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1250 WS=1 SACK_PERM
1022	9.767989	172.26.42.81	193.137.9.178	TCP	54		58551 → 80 [ACK] Seq=1 Ack=1 Win=131072 Len=0
1023	9.768876	172.26.42.81	193.137.9.178	HTTP	583		GET / HTTP/1.1
1024	9.773979	142.250.201.78	172.26.42.81	TCP	1304		443 → 58541 [ACK] Seq=1898838 Ack=15682 Win=685 Len=1250 [TCP segment of a reassembled PDU]
1025	9.773979	142.250.201.78	172.26.42.81	TLsv...	79		Application Data
1026	9.773979	142.250.201.78	172.26.42.81	TLsv...	1092		Application Data
1027	9.774115	172.26.42.81	142.250.201.78	TCP	54		58541 → 443 [ACK] Seq=24044 Ack=1901151 Win=1025 Len=0
1028	9.774896	193.137.9.178	172.26.42.81	TCP	66		80 → 58551 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1250 WS=1 SACK_PERM

Fig. 4. MSS do TCP

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

A resposta HTTP do servidor respeitante ao primeiro pedido GET apresenta com um tamanho de 924 *bytes* (Fig.5).

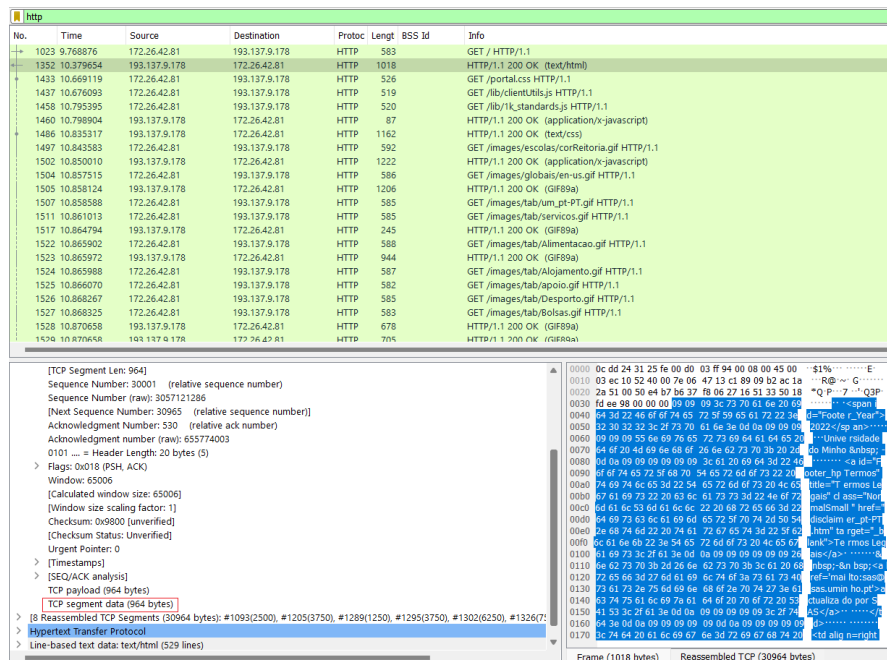


Fig. 5. Campo TCP da resposta HTTP ao pedido GET

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

A resposta HTTP foi transmitida em 8 segmentos TCP (Fig.6). Teoricamente, sabendo que o MSS corresponde a 1460 *bytes* e o tamanho da resposta é de 30964 *bytes*. Logo, a divisão entre o tamanho total e o MSS corresponde a 22 segmentos TCP possíveis.

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.

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.

Contudo, quando se acede novamente à página, a quantidade de mensagens HTTP é significativamente menor, isto deve-se ao facto, da *cache* do *browser* que armazena essas mensagens, permitindo assim que as mensagens HTTP sejam num número reduzido.

No.	Time	Source	Destination	Protocol	Length	SSS Id	Info
1023	9.768876	172.26.42.81	193.137.9.178	HTTP	583		GET / HTTP/1.1
1352	10.379654	193.137.9.178	172.26.42.81	HTTP	1018		HTTP/1.1 200 OK (text/html)
1433	10.669119	172.26.42.81	193.137.9.178	HTTP	526		GET /portal.css HTTP/1.1
1437	10.676093	172.26.42.81	193.137.9.178	HTTP	519		GET /lib/clientUtils.js HTTP/1.1
1458	10.795395	172.26.42.81	193.137.9.178	HTTP	520		GET /lib/ik_standards.js HTTP/1.1
1460	10.798904	193.137.9.178	172.26.42.81	HTTP	87		HTTP/1.1 200 OK (application/x-javascript)
1486	10.835317	193.137.9.178	172.26.42.81	HTTP	1162		HTTP/1.1 200 OK (text/css)
1497	10.843583	172.26.42.81	193.137.9.178	HTTP	592		GET /images/escolas/corReitoria.gif HTTP/1.1
1502	10.850010	193.137.9.178	172.26.42.81	HTTP	1222		HTTP/1.1 200 OK (application/x-javascript)
1504	10.857515	172.26.42.81	193.137.9.178	HTTP	586		GET /images/globais/en-us.gif HTTP/1.1
1505	10.858124	193.137.9.178	172.26.42.81	HTTP	1206		HTTP/1.1 200 OK (GIF89a)
1507	10.858588	172.26.42.81	193.137.9.178	HTTP	585		GET /images/tab_um_pt-PT.gif HTTP/1.1
1511	10.861013	172.26.42.81	193.137.9.178	HTTP	585		GET /images/tab/servicos.gif HTTP/1.1
1517	10.864794	193.137.9.178	172.26.42.81	HTTP	245		HTTP/1.1 200 OK (GIF89a)
1522	10.865902	172.26.42.81	193.137.9.178	HTTP	588		GET /images/tab/Alimentacao.gif HTTP/1.1
1523	10.865972	193.137.9.178	172.26.42.81	HTTP	944		HTTP/1.1 200 OK (GIF89a)
1524	10.865988	172.26.42.81	193.137.9.178	HTTP	587		GET /images/tab/Alojamento.gif HTTP/1.1
1525	10.866070	172.26.42.81	193.137.9.178	HTTP	582		GET /images/tab/apoio.gif HTTP/1.1
1526	10.868267	172.26.42.81	193.137.9.178	HTTP	585		GET /images/tab/Resumo.gif HTTP/1.1

> Frame 1023: 583 bytes on wire (4664 bits), 583 bytes captured (4664 bits) on in	0000	2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 77 77 77 2e	/1.1 Ho st: ww.
> Ethernet II, Src: IntelCor_31:25:fe (0c:dd:24:31:25:fe), Dst: Comdant_ff:94:00	0000	73 61 73 2e 75 6d 69 6e 68 6f 2e 70 74 0d 0a 43	sas.unin ho.pt C
> Internet Protocol Version 4, Src: 172.26.42.81, Dst: 193.137.9.178	0000	6f 6e 6e 65 63 74 69 6f 6e 3a 20 0b 65 65 70 2d	connectio n: keep-
> Transmission Control Protocol, Src Port: 58551, Dst Port: 80, Seq: 1, Ack: 1, Len: 0	0070	61 6c 69 76 65 0d 0a 55 70 67 72 61 64 65 2d 49	alive -U pgrade-I
> Hypertext Transfer Protocol	0080	6e 73 65 63 75 72 65 2d 52 65 71 75 65 73 74 73	nsecure- Requests
> GET / HTTP/1.1\r\n	0090	3a 20 31 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a	: 1- Use r-Agent:
> Host: www.sas.uninho.pt\r\n	00a0	20 4d 6f 7a 69 6c 61 2f 35 2e 30 20 28 57 69	Mozilla /5.0 (Wi
> Connection: keep-alive\r\n	00b0	6e 64 6f 77 73 20 4e 54 20 31 30 2e 30 3b 20 57	ndows NT 10.0; W
> Upgrade-Insecure-Requests: 1\r\n	00c0	69 6e 36 34 3b 20 78 36 34 29 20 41 70 70 6c 65	in64; x6 4) Apple
> User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36	00d0	57 65 62 4b 69 74 2f 35 33 37 2e 33 36 20 28 4b	Webkit/5 37.36 (K
> Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8	00e0	48 54 4d 4c 2c 20 6c 69 68 65 20 47 65 63 68 6f	HTML, li ke Gecko
> Accept-Encoding: gzip, deflate\r\n	00f0	29 20 43 68 72 6f 6d 65 2f 31 30 38 2e 30 2e 30	) Chrome /108.0.0
> Accept-Language: pt-PT,pt;q=0.9,en-US;q=0.8,en;q=0.7\r\n	0100	2e 30 20 53 61 66 61 72 69 2f 35 33 37 2e 33 36	.0 Safari i/537.36
> Cookie: _ga=G41.2.819498540.1665347397; _fbp=fb.1.1669911176505.1467464822\r\n	0110	0d 0a 41 63 63 65 70 74 3a 20 74 65 78 74 2f 68	Accept : text/h
> [Full request URI: http://www.sas.uninho.pt/]	0120	74 6d 6c 2c 61 70 70 6c 69 63 61 74 69 6f 6e 2f	tml,appl ication/
> [HTTP request 1/7]	0130	73 69 74 6d 6c 2b 78 6d 6c 2c 61 70 70 6c 69 63	html,xm l,appl ic
> [Response in frame 1352]	0140	61 74 69 6f 6e 2f 78 6d 6c 3b 71 3d 30 2c 39 2c	ation/xm l;q=0.9,
	0150	69 6d 61 67 65 2f 61 76 69 66 2c 69 6d 61 67 65	image/av if,image
	0160	2f 77 65 62 70 2c 69 6d 61 67 65 2f 61 70 6e 67	/webp,im age/apn
	0170	2c 2a 2f 2a 3b 71 3d 30 2e 38 2c 61 70 70 6c 65	/*;q=0 .8,appl

Fig. 7. Host do pedido HTTP

- 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.

O servidor está a funcionar em modo de conexão persistente, visto que o HTTP utiliza por definição a conexão persistente, mas em paralelo (vários envios ao mesmo tempo). Isto é, a conexão TCP fica aberta e disponível para o envio de mensagens do servidor ou do cliente. As conexões são encerradas quando ao fim de um certo tempo não existe envio ou pedidos.

- 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. 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 é a principal vantagem e desvantagem inerente ao *hard refresh*.

O *hard refresh* vai limpar a *cache* do *browser*, e por isso, terá de realizar novamente todas as trocas de mensagens como fosse a primeira vez a aceder a página. (Fig.8)

De seguida, ao procurar a mesma página, a informação estará guardada na *cache* o que torna a troca de mensagens mínima (Fig.9)

No.	Time	Source	Destination	Protocol	Length	BSS Id	Info
15	0.638522	172.26.42.81	193.137.9.178	HTTP	538		GET / HTTP/1.1
59	1.063610	193.137.9.178	172.26.42.81	HTTP	1020		HTTP/1.1 200 OK (text/html)
61	1.173310	172.26.42.81	193.137.9.178	HTTP	500		GET /portal.css HTTP/1.1
63	1.174453	172.26.42.81	193.137.9.178	HTTP	493		GET /lib/clientUtils.js HTTP/1.1
67	1.181653	172.26.42.81	193.137.9.178	HTTP	494		GET /lib/lk_standards.js HTTP/1.1
69	1.185790	193.137.9.178	172.26.42.81	HTTP	87		HTTP/1.1 200 OK (application/x-javascript)
74	1.236395	193.137.9.178	172.26.42.81	HTTP	1222		HTTP/1.1 200 OK (application/x-javascript)
75	1.244326	172.26.42.81	193.137.9.178	HTTP	555		GET /images/escolas/correitoria.gif HTTP/1.1
76	1.253634	193.137.9.178	172.26.42.81	HTTP	1206		HTTP/1.1 200 OK (GIF89a)
77	1.256814	172.26.42.81	193.137.9.178	HTTP	549		GET /images/globals/en-us.gif HTTP/1.1
79	1.262964	193.137.9.178	172.26.42.81	HTTP	245		HTTP/1.1 200 OK (GIF89a)
81	1.266434	172.26.42.81	193.137.9.178	HTTP	548		GET /images/tab_um_pt-PT.gif HTTP/1.1
83	1.272874	193.137.9.178	172.26.42.81	HTTP	1261		HTTP/1.1 200 OK (GIF89a)
85	1.275703	172.26.42.81	193.137.9.178	HTTP	548		GET /images/tab/servicos.gif HTTP/1.1
88	1.282655	193.137.9.178	172.26.42.81	HTTP	1162		HTTP/1.1 200 OK (text/css)
90	1.283671	193.137.9.178	172.26.42.81	HTTP	944		HTTP/1.1 200 OK (GIF89a)
91	1.286764	172.26.42.81	193.137.9.178	HTTP	551		GET /images/tab/Alimentacao.gif HTTP/1.1
92	1.289980	172.26.42.81	193.137.9.178	HTTP	550		GET /images/tab/Alojamento.gif HTTP/1.1
93	1.291601	193.137.9.178	172.26.42.81	HTTP	733		HTTP/1.1 200 OK (GIF89a)

> Frame 15: 538 bytes on wire (4304 bits), 538 bytes captured (4304 bits) on interface  
 > Ethernet II, Src: IntelCor\_31:25:fe (0c:dd:24:31:25:fe), Dst: ComdaEnt\_ff:94:00 (08:00:00:00:00:00)  
 > Internet Protocol Version 4, Src: 172.26.42.81, Dst: 193.137.9.178  
 > Transmission Control Protocol, Src Port: 59055, Dst Port: 80, Seq: 1, Ack: 1, Len: 538  
 > Hypertext Transfer Protocol

0000 00 d0 03 ff 94 00 0c dd 24 31 25 fe 08 00 45 00 ..... \$1%--E  
 0010 02 0c 0d ae 40 00 80 06 00 00 ac 1a 2a 51 c1 89 ...@...-\*Q-  
 0020 09 b2 e8 71 00 50 2c cb 5b 0a ac eb 2c 66 50 18 ...q P, [-, ., P  
 0030 02 00 a3 d9 00 00 47 45 54 20 2f 20 48 54 54 50 .....GE T / HTTP  
 0040 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 77 77 77 2e /1.1--Ho st: www.  
 0050 73 61 73 2e 75 6d 69 6e 68 6f 2e 70 74 0d 0a 43 sas.umin ho.pt--C  
 0060 6f 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 70 2d connectio n: keep-  
 0070 61 6c 69 76 65 0d 0a 55 70 67 72 61 64 65 2d 49 alive- U pgrade-I  
 0080 6e 73 65 63 75 72 65 2d 52 65 71 75 65 73 74 73 nsecure- Requests  
 0090 3a 20 31 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a : 1- Use r-Agent:  
 00a0 20 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 57 69 Mozilla /5.0 (Wi  
 00b0 6e 64 6f 77 73 20 4e 54 20 31 30 2e 30 3b 20 57 ndows NT 10.0; W  
 00c0 69 6e 36 34 3b 20 78 36 34 29 20 41 70 70 6c 65 in64; x6 4) Apple  
 00d0 57 65 62 4b 69 74 2f 35 33 37 2e 33 36 20 28 4b WebKit/5 37.36 (K  
 00e0 48 54 4d 4c 2c 20 6c 69 6b 65 20 47 65 63 0b 6f HTML, 11 ke Gecko  
 00f0 29 20 43 68 72 6f 6d 65 2f 31 30 38 2e 30 2e 30 ) Chrome /108.0.0  
 0100 2e 30 20 53 61 66 61 72 69 2f 35 33 37 2e 33 36 0 Safari /537.36  
 0110 20 45 64 67 2f 31 30 38 2e 30 2e 31 34 36 32 2e Edg/108 .0.1462.  
 0120 34 32 0d 0a 41 63 63 65 70 74 3a 20 74 65 78 74 42- Acce pt: text  
 0130 2f 68 74 6d 6c 2c 61 70 70 6c 69 63 61 74 69 6f /html,ap plicatio

Fig. 8. Tráfego HTTP com *hard refresh*

No.	Time	Source	Destination	Protocol	Length	BSS Id	Info
135	1.874792	172.26.42.81	193.137.9.178	HTTP	590		GET / HTTP/1.1
200	2.623125	193.137.9.178	172.26.42.81	HTTP	956		HTTP/1.1 200 OK (text/html)

> Frame 135: 590 bytes on wire (4720 bits), 590 bytes captured (4720 bits) on interface  
 > Ethernet II, Src: IntelCor\_31:25:fe (0c:dd:24:31:25:fe), Dst: ComdaEnt\_ff:94:00 (08:00:00:00:00:00)  
 > Internet Protocol Version 4, Src: 172.26.42.81, Dst: 193.137.9.178  
 > Transmission Control Protocol, Src Port: 59512, Dst Port: 80, Seq: 1, Ack: 1, Len: 590  
 > Hypertext Transfer Protocol

0000 00 d0 03 ff 94 00 0c dd 24 31 25 fe 08 00 45 00 ..... \$1%--E  
 0010 02 0c 0d ae 40 00 80 06 00 00 ac 1a 2a 51 c1 89 ...@...-\*Q-  
 0020 09 b2 e8 71 00 50 2c cb 5b 0a ac eb 2c 66 50 18 ...q P, [-, ., P  
 0030 02 00 a3 d9 00 00 47 45 54 20 2f 20 48 54 54 50 .....GE T / HTTP  
 0040 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 77 77 77 2e /1.1--Ho st: www.  
 0050 73 61 73 2e 75 6d 69 6e 68 6f 2e 70 74 0d 0a 43 sas.umin ho.pt--C  
 0060 6f 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 70 2d connectio n: keep-  
 0070 61 6c 69 76 65 0d 0a 55 70 67 72 61 64 65 2d 49 alive- U pgrade-I  
 0080 6e 73 65 63 75 72 65 2d 52 65 71 75 65 73 74 73 nsecure- Requests  
 0090 3a 20 31 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a : 1- Use r-Agent:  
 00a0 20 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 57 69 Mozilla /5.0 (Wi  
 00b0 6e 64 6f 77 73 20 4e 54 20 31 30 2e 30 3b 20 57 ndows NT 10.0; W  
 00c0 69 6e 36 34 3b 20 78 36 34 29 20 41 70 70 6c 65 in64; x6 4) Apple  
 00d0 57 65 62 4b 69 74 2f 35 33 37 2e 33 36 20 28 4b WebKit/5 37.36 (K  
 00e0 48 54 4d 4c 2c 20 6c 69 6b 65 20 47 65 63 0b 6f HTML, 11 ke Gecko  
 00f0 29 20 43 68 72 6f 6d 65 2f 31 30 38 2e 30 2e 30 ) Chrome /108.0.0  
 0100 2e 30 20 53 61 66 61 72 69 2f 35 33 37 2e 33 36 0 Safari /537.36  
 0110 20 45 64 67 2f 31 30 38 2e 30 2e 31 34 36 32 2e Edg/108 .0.1462.  
 0120 34 32 0d 0a 41 63 63 65 70 74 3a 20 74 65 78 74 42- Acce pt: text  
 0130 2f 68 74 6d 6c 2c 61 70 70 6c 69 63 61 74 69 6f /html,ap plicatio

ip3\_ext0sem.pcapng Packets: 256 - Displayed: 2 (0.8%)

Fig. 9. Tráfego HTTP sem *hard refresh*



## 1.2 HTTPS

- 11) Aceda a <https://elearning.uminho.pt> , ao mesmo tempo que captura 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.

O *browser* assinala que o utilizador está perante uma ligação HTTP segura ao servidor através da imagem de um “cadeado fechado”(Fig.10).

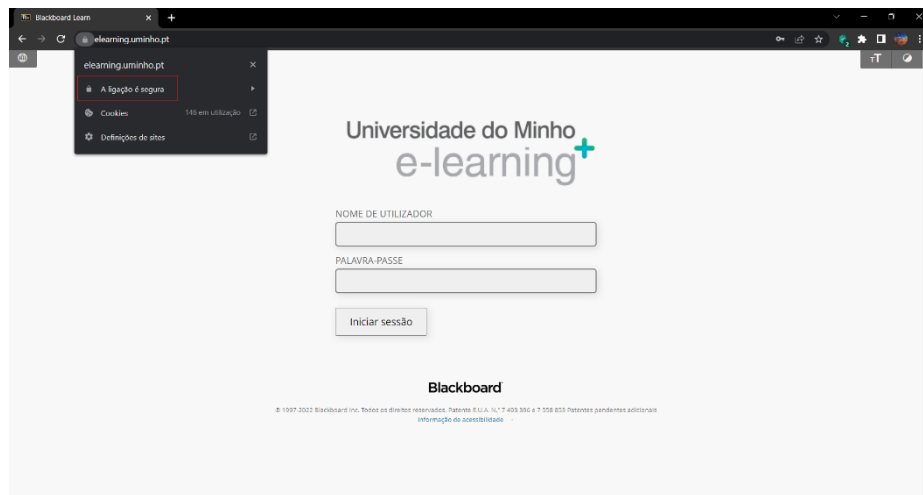


Fig. 10. Ligação segura

- 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*?

O tráfego HTTP não é identificado como tal no *Wireshark*, porque a informação encontra-se cifrada, tratando assim de uma ligação *http-over-tls*. Porém, através das portas de atendimento, consegue identificar se se trata de um canal seguro (443) ou não (80).



- 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 e o servidor, *v)* a identificação da página acessada 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.

Uma comunicação HTTPS para se manter oculta dum atacante necessita de conter a informação cifrada quando se utiliza um *sniffer*, ou seja, a informação oculta corresponde ao nome de servidor *web* e aos dados da aplicação trocados entre o servidor e o cliente.

## 2 Consultas ao serviço de resolução de nome DNS

- 1) Indique qual o servidor de nomes que a sua máquina está a usar?

O servidor de nomes que a máquina está a usar é *dns3.uminho.pt*.

- 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.fcn.pt*.

Os servidores *www.sas.uminho.pt*, *marco.uminho.pt* e *www.google.com* têm como endereços IPv4: *193.137.9.178*, *193.136.9.240* e *172.217.168.164*, respetivamente. O servidor *www.fcn.pt* têm como endereço IPv6: *2001:690:a00:1036:1113:247*.

- 3) Experimente fazer uma *query* aos registos PTR para os nomes **240.9.136.193.in-addr.arpa.** e **7.4.2.0.0.0.0.0.0.0.0.0.3.1.1.6.3.0.1.0.0.2.ip6.arpa.** Comente os resultados face aos obtidos na alínea anterior.

Os nomes no enunciado correspondem a: *marco.uminho.pt* e *www.fcnn.pt*, respetivamente.

Estes nomes correspondem aos servidores de nomes, na alínea anterior.

- 4) Usando os registos NS, identifique os servidores de nomes definidos para os domínios: “*uminho.com*”, “*sas.uminho.pt*”, “*pt.*” e “*.*” (*root*).

Os nomes definidos para os domínios correspondem a “*uminho.com*” os servidores de nome: *dns2.uminho.pt*, *dns3.uminho.pt*, *dns.uminho.pt*, *ns02.fcnn.pt*; “*sas.uminho.pt*” os servidores de nome: *dns.uminho.pt*, *dns2.uminho.pt* e *dns.uminho.pt*; “*pt.*” Os servidores de nome: *ns2.nic.fr*, *b.dns.pt*, *ns.dns.br*, *d.dns.pt*, *g.dns.pt*, *c.dns.pt*, *h.dns.pt*, *e.dns.pt* e *a.dns.pt*; “*.*” (*root*) os servidores de nomes: *d.root-servers.net*, *h.root-servers.net*, *k.root-servers.net*, *b.root-servers.net*, *g.root-servers.net*, *e.root-servers.net*, *l.root-servers.net*, *a.root-servers.net*, *m.root-servers.net*, *f.root-servers.net*, *c.root-servers.net*, *j.root-servers.net* e *i.root-servers.net*.

- a) **Perante a informação obtida, diga, justificando, se os servidores de nomes de diferentes domínios podem coexistir numa mesma máquina física.**

Os servidores de nomes de diferentes domínios podem coexistir numa mesma máquina física, caso ambos possuem os mesmos servidores de nomes, neste caso, temos o *sas.uminho.pt* e o *uminho.pt*.

- b) **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?**

Os domínios *uminho.pt* e *sas.uminho.pt* apresentam IP iguais, ou seja, a mesma máquina pode gerir vários domínios, como também ser possível máquinas distintas gerir o mesmo domínio.

- 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?**

O servidor DNS primário para os domínios “*uminho.pt*”, “*pt.*” e “*.*” Correspondem a: *dns.uminho.pt*, *curiosity.dns.pt* e *a.root-servers.net*.

Os servidores primários contêm todos os registos de recursos relevantes e administram as consultas DNS para um domínio, no qual os tempos indicados correspondem ao tempo de execução diferentes tarefas, ou, seja, (recarregar) *refresh*, (tentar) *retry* e (expirar) *expire* ao fim de um determinado tempo. Por outro lado, os servidores secundários de DNS contêm cópias de arquivos de zona somente para leitura, o que significa que não podem ser modificados.

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

Os servidores de email do domínio *edu.ulisboa.pt* são: *ASPMX.L.GOOGLE.COM*, *ASPM3.GOOGLEMAIL.COM*, *ALT2.ASPMX.L.GOOGLE.COM*, *ASPMX2.GOOGLEMAIL.COM* e *ALT1.ASPMX.L.GOOGLE.COM*.

As mensagens dirigidas a *geral@edu.ulisboa.pt* são entregues preferencialmente a *sistemas.reitoria.ulisboa.pt*.

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

O nome dos *aliases* de *www.ebay.com* corresponde a *slot9428.ebay.com.edgekey.net* (*canonical name*).

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

Uma resposta é autoritativa quando esta é gerada por um servidor de nomes em que recebe um pedido de resolução de endereços para um domínio numa zona em que este tem autoridade. Pelo mesmo raciocínio, uma resposta não autoritativa é obtida quando um servidor de nomes recebe um pedido referente a um domínio numa zona sobre a qual este não tem autoridade, ou seja, a resposta não pertence ao servidor primário ou secundário.

### 3 Uso da camada de transporte por parte das aplicações

- 1) Preencha a seguinte tabela com base nos resultados que obteve:

Comando/aplicação	Canal seguro?	Protocolo de transporte (se aplicável)	Porta de atendimento (se aplicável)
<i>browser http://</i>	Sim	TCP	80
<i>browser https://</i>	Sim	TCP	443
<i>ftp</i>	Sim	TCP	21
<i>Ping</i>	Não	Não aplicável	Não aplicável
<i>Ssh</i>	Sim	TCP	22
<i>nslookup/dig</i>	Não	UDP	53
<i>Traceroute</i>	Não	UDP	33434
<i>telnet</i>	Sim	TCP	23

- 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.**

O TCP (*Transmission Control Protocol*) é um protocolo de transporte fiável de dados ponto-a-ponto orientado por conexão, enquanto o UDP (*User Datagram Protocol*) é um protocolo de transporte ponto-a-ponto não fiável que não usa conexões. Contudo, o UDP apresenta uma maior velocidade do que TCP, sendo este último responsável pelo controlo de fluxo e congestionamento de erros.

Os diferentes campos do TCP são a porta Origem/Destino, número de sequência, número de ACK (*Acknowledgements*), o comprimento do cabeçalho, as *Flags* (que são indicações específicas que podem ser: ACK, PDH, RST, SYN, FIN, URG), a janela, a soma de controlo, e ainda o apontador de emergência, podendo ainda haver outras opcionais.

Destes campos, a janela é a responsável pelo controlo de fluxo, e é através deste campo que o software TCP indica a quantidade de dados que tem capacidade de receber.

A soma de controlo é usada para verificar a integridade/fiabilidade do cabeçalho e dos dados do segmento TCP.

O número de sequência e o número de ACK são usados para a ordenação dos octetos, visto que o número de ACK vai identificar a posição do último *byte* recebido e vai especificar o número do próximo *byte* que o recetor espera receber; enquanto o número de sequência vai identificar a posição, no fluxo de *bytes*, do segmento que foi enviado pelo transmissor, ou seja, refere-se ao fluxo de dados que vai na mesma direção do segmento.

## 4 Conclusão

Os temas abordados neste trabalho prático têm como finalidade a exposição de conhecimentos dos autores ao docente da Unidade Curricular. Apesar dos percalços durante o trabalho, não impediu a realização do mesmo. Os autores com recurso ao *Wireshark* proporcionou uma ajuda indispensável no estudo do *http*, assim como, nos protocolos de transporte, TCP e UDP, não esquecendo do serviço de resolução de nome DNS. No seu geral, o estudo contribui de forma pedagógica nos temas abordados, proporcionando grande interessante aos autores e elevando o grau de conhecimento dos mesmos.