

Comunicações por Computador



TP1: Protocolos da Camada de Transporte

2021/2022

PL7 - Grupo 77

Jorge Vieira A84240

Joana Alves A93290

PARTE I

Questão 1: *De que forma as perdas e duplicações de pacotes afetaram o desempenho das aplicações? Que camada lidou com as perdas e duplicações: transporte ou aplicação? Responda com base nas experiências feitas e nos resultados observados.*

Resposta:

O **protocolo TCP**, sendo um protocolo de transporte fiável, garante que os pacotes sejam enviados de forma correta e sem erros, sendo que com cada pacote é sempre enviado pelo menos um *acknowledgment* em como esse pacote foi recebido com sucesso. A consequência mais óbvia deste acréscimo de segurança é o processo tornar-se muito mais complexo devido ao número de pacotes de controlo que são trocados por ambas as partes, logo, é de notar a importância das características da rede neste protocolo.

No caso de uma rede de menor qualidade, são perdidos e corrompidos imensos pacotes o que obriga à transmissão de mensagens de erros e que os pacotes que foram mal enviados sejam reenviados, levando assim a uma sobrecarga na rede e atraso na aplicação.

O **protocolo UDP**, contrariamente ao protocolo TCP, é uma alternativa não fiável pois não temos garantias da entrega de mensagens. O custo de usar este protocolo é que a aplicação em si terá de ficar responsável pelos dados que são recebidos. Daí surge uma das maiores vantagens do UDP, um protocolo de latência de ligação reduzida que não sobrecarrega a rede com *acknowledgments* nem com reenvios em condições de rede menos favoráveis. É o protocolo mais indicado, por exemplo, para serviços de *streaming*.

A camada que lidou com as perdas e duplicações foi a camada de transporte, ou seja, os protocolos de transporte UDP e TCP.

Questão 2: *Obtenha a partir do wireshark, ou desenhe manualmente, um diagrama temporal para a transferência de file1 por FTP. Foque-se apenas na transferência de dados [ftp-data] e não na conexão de controlo, pois o FTP usa mais que uma conexão em simultâneo. Identifique, se aplicável, as fases de início de conexão, transferência de dados e fim de conexão. Identifique também os tipos de segmentos trocados e os números de sequência usados quer nos dados como nas confirmações.*

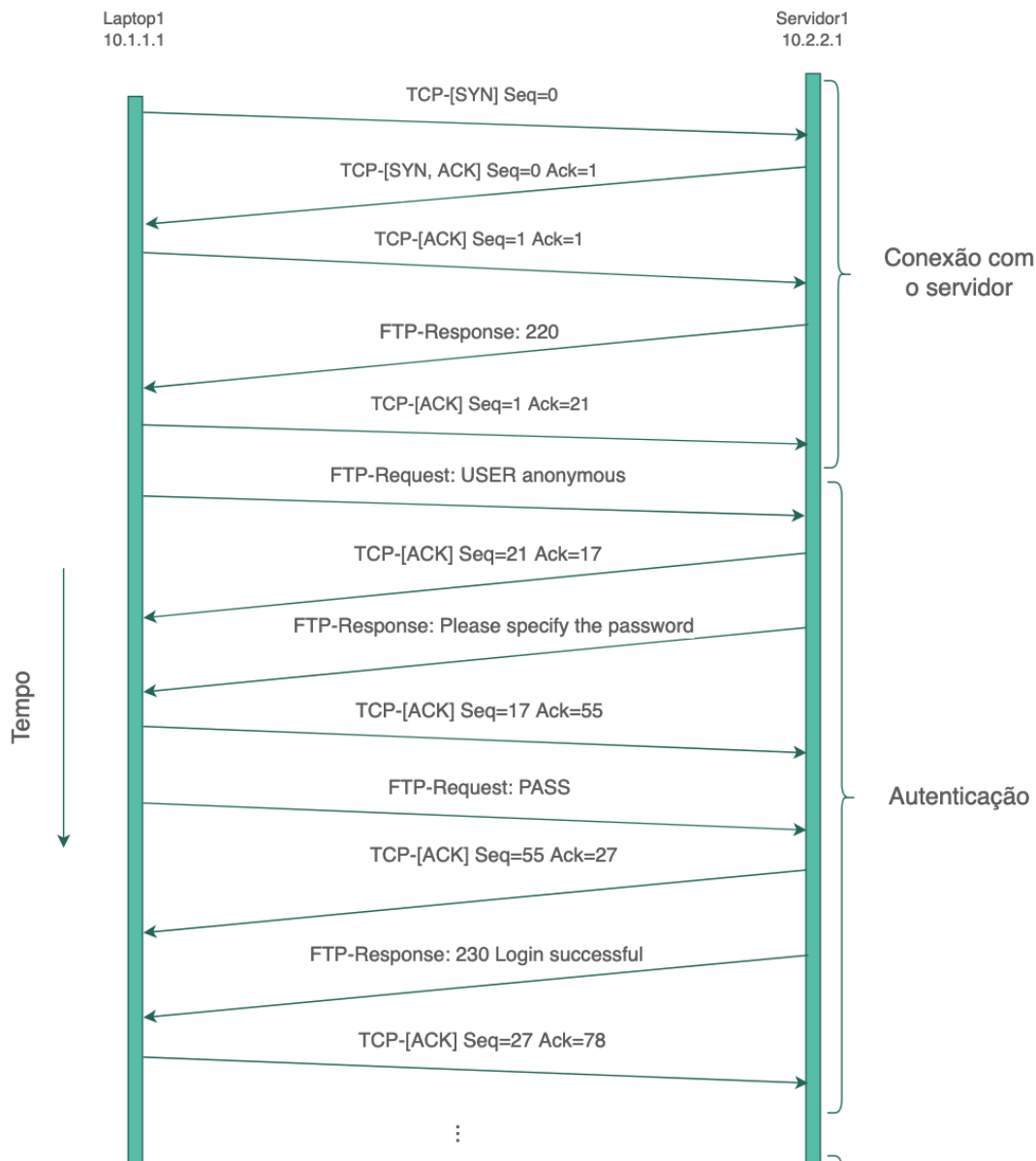


Diagrama temporal da transferência do ficheiro file1 por FTP

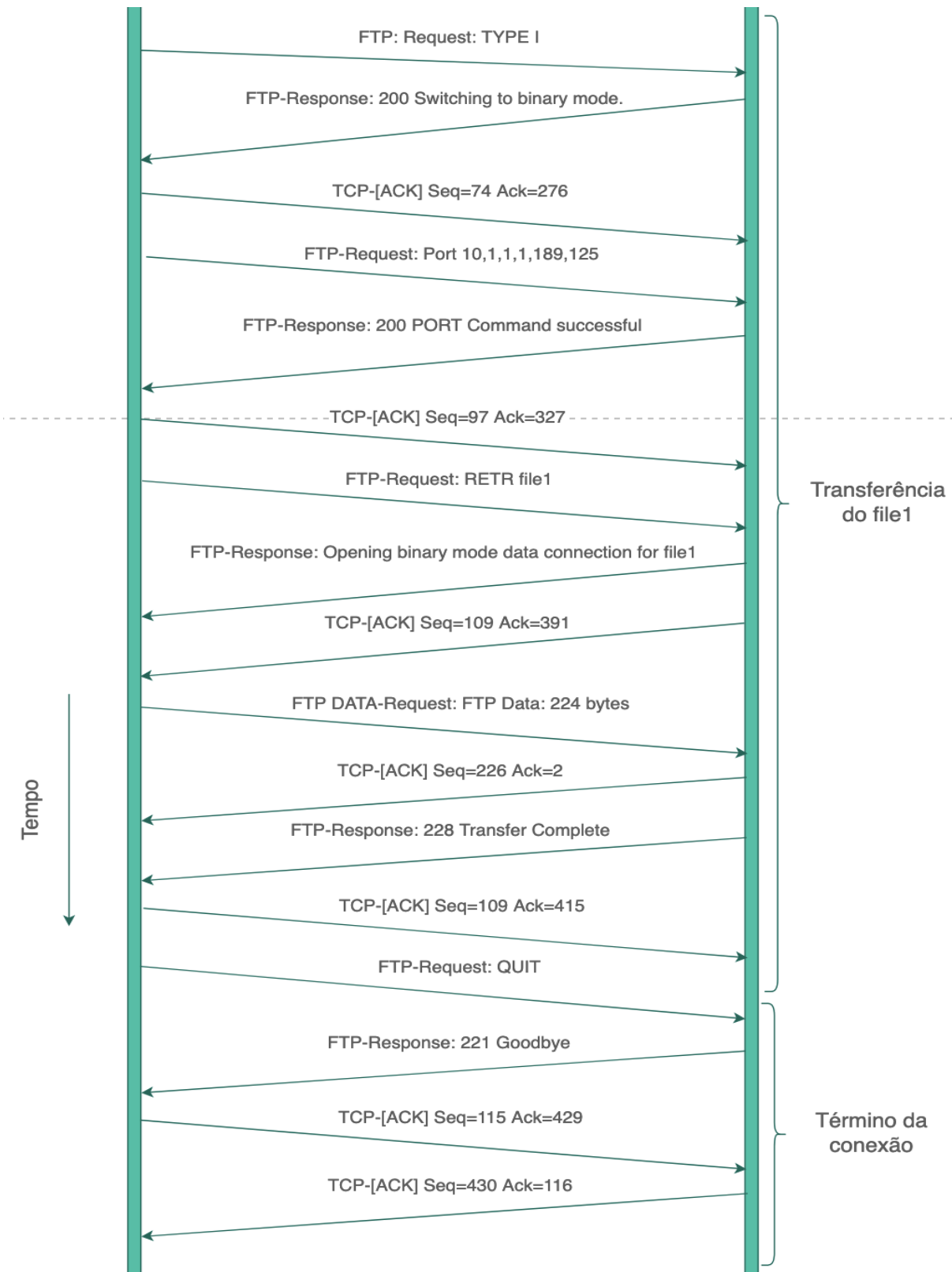


Diagrama temporal da transferência do ficheiro file1 por FTP (continuação)

Questão 3: *Obtenha a partir do wireshark, ou desenhe manualmente, um diagrama temporal para a transferência de file1 por TFTP. Identifique, se aplicável, as fases de início de conexão, transferência de dados e fim de conexão. Identifique também os tipos de segmentos trocados e os números de sequência usados quer nos dados quer nos dados como nas confirmações*

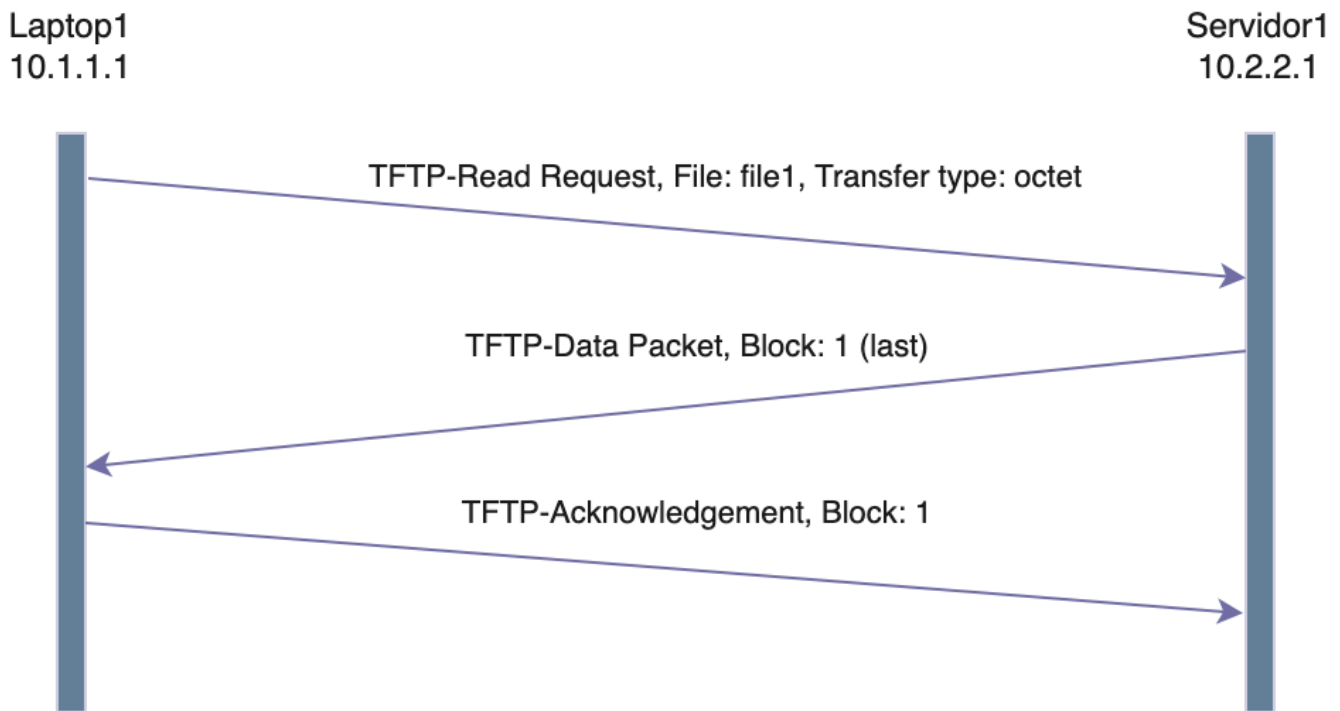


Diagrama temporal da transferência do ficheiro file1 por TFTP

Questão 4: *Compare sucintamente as quatro aplicações de transferência de arquivos que usou nos seguintes pontos (i) uso da camada de transporte; (ii) eficiência; (iii) complexidade; (iv) segurança.*

Resposta:

Relativamente à **camada de transporte**, três das quatro aplicações de transferência utilizam o protocolo de transporte TCP, sendo elas HTTP, SFTP e FTP. Uma vez que o protocolo TCP é orientado à conexão e por isso consegue lidar com perdas e duplicações de pacotes, estas aplicações possuem um nível ‘extra’ de controle no tráfego de pacotes. Contrariamente, a aplicação TFTP utiliza o protocolo de transporte UDP que não é orientado à conexão, e consequentemente pode não ser tão competente na transferência bem-sucedida de pacotes.

Relativamente à **eficiência**, no que toca à velocidade de download do file1, a aplicação HTTP superou todas as outras obtendo em média uma velocidade de 40MB/s. Ao comparar apenas as aplicações SFTP e FTP, vemos que esta última apresenta uma velocidade (média) de download superior à primeira, 400KB/s em contraste com 17.5KB/s.

Num cenário de perda de pacotes, as aplicações que utilizam TCP terão um desempenho mais eficiente, visto que este protocolo é orientado à conexão. No entanto, num cenário ideal de não perda de pacotes, a aplicação TFTP, que utiliza o protocolo UDP, seria a mais eficiente.

Ordenando por ordem crescente de **complexidade** ficamos com o seguinte segmento: FTP > SFTP > HTTP > TFTP. Construímos este segmento comparando os protocolos de transporte e os passos de autenticação, caso se aplicasse. Assim, a aplicação FTP fica em primeiro lugar de complexidade pois utiliza TCP e um par de autenticação (utilizador, palavra-passe). Em segundo temos a aplicação SFTP pois apesar de também ter TCP como protocolo de transporte, possui apenas um passo de autenticação (palavra-passe). De seguida, colocamos a aplicação HTTP uma vez que possui protocolo de transporte TCP, ao contrário da aplicação em último lugar, TFTP, que utiliza o protocolo UDP.

Relativamente à **segurança**, tal como já foi dito em cima, as aplicações que utilizam o protocolo de transporte TCP (HTTP, SFTP e FTP) têm uma vantagem relativamente à que usa UDP (TFTP) na medida em que têm um maior controlo sobre perdas de pacotes.

Para além disto, algumas das aplicações implementam um nível extra de segurança, como é o caso de FTP (autenticação de utilizador e sua password) e SFTP (autenticação por password).

PARTE II

Questão 1: *Com base na captura de pacotes feita, preencha a seguinte tabela, identificando para cada aplicação executada, qual o protocolo de aplicação, o protocolo de transporte, porta de atendimento e overhead de transporte.*

Comando usado	Protocolo de Aplicação	Protocolo de transporte	Porta de atendimento	Overhead de transporte em bytes
ping	NA	NA	NA	NA
tracert	NA	NA	NA	NA
telnet	TELNET	TCP	23	20
ftp	FTP	TCP	50930	20
tftp	TFTP	UDP	69	8
http (browser)	HTTP	TCP	80	20
nslookup	DNS	UDP	53	8
ssh	SSH	TCP	22	20

NA - Não aplicável

Justificações:

Ping: Como o ping trabalha ao nível da camada de rede (camada 3) usando como protocolo o ICMP (integrante do protocolo IP), não tem protocolos de aplicação e transporte nem porta de atendimento e, como tal, não possui overhead de transporte.

Traceroute: Como o traceroute trabalha ao nível da camada de rede (camada 3), tal como o ping, usando como protocolo o ICMP (integrante do protocolo IP), não tem protocolos de aplicação e transporte nem porta de atendimento e, como tal, não possui overhead de transporte.

Telnet:

16	15.159203781	10.0.2.15	193.136.9.183	TELNET	57 Telnet Data ...
17	15.159470919	193.136.9.183	10.0.2.15	TCP	60 23 → 41110 [ACK]
18	15.200792160	193.136.9.183	10.0.2.15	TELNET	60 Telnet Data ...
19	15.200814974	10.0.2.15	193.136.9.183	TCP	54 41110 → 23 [ACK]

▶ Frame 16: 57 bytes on wire (456 bits), 57 bytes captured (456 bits) on interface enp0s3, id 0
 ▶ Ethernet II, Src: PcsCompu_06:03:48 (08:00:27:06:03:48), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
 ▶ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 193.136.9.183
 ▶ Transmission Control Protocol, Src Port: 41110, Dst Port: 23, Seq: 123, Ack: 52, Len: 3
 ▶ Telnet

0000	52 54 00 12 35 02 08 00	27 06 03 48 08 00 45 10	RT..5... '...H..E..
0010	00 2b 91 3d 40 00 40 06	d2 31 0a 00 02 0f c1 88	..+.=@.@..1.....
0020	09 b7 a0 96 00 17 65 9d	88 cf 0c ab 2a 35 50 18	...e...*5P..
0030	fa bd d7 6b 00 00 1b 5b	41	...k...[A

FTP:

24	19.992785770	193.136.9.183	10.0.2.15	FTP	88 Response: 331 Please specify the password.
25	19.992804109	10.0.2.15	193.136.9.183	TCP	54 39274 → 21 [ACK] Seq=10 Ack=55 Win=64186 L

▶ Frame 24: 88 bytes on wire (704 bits), 88 bytes captured (704 bits) on interface enp0s3, id 0
 ▶ Ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_06:03:48 (08:00:27:06:03:48)
 ▶ Internet Protocol Version 4, Src: 193.136.9.183, Dst: 10.0.2.15
 ▶ Transmission Control Protocol, Src Port: 21, Dst Port: 39274, Seq: 21, Ack: 10, Len: 34
 ▶ File Transfer Protocol (FTP)

0000	08 00 27 06 03 48 52 54	00 12 35 02 08 00 45 00	..'.HRT..5...E..
0010	00 4a 08 ef 00 00 40 06	9a 71 c1 88 09 b7 0a 00	..J...@..q.....
0020	02 0f 00 15 99 6a 0e 7c	fc 16 20 1f f8 90 50 18	...j... ...P..
0030	ff ff d4 78 00 00 33 33	31 20 50 6c 65 61 73 65	...x...33 1 Please
0040	20 73 70 65 63 69 66 79	20 74 68 65 20 70 61 73	specify the pas
0050	73 77 6f 72 64 2e 0d 0a		sword...

TFTP:

9	36.051515182	10.0.2.15	193.136.9.183	TFTP	86 Read Request, File: file1, Transfer type: octet, tsize=0, blk
10	38.000811791	10.0.2.15	91.209.16.78	NTP	90 NTP Version 4, client

▶ Frame 9: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface enp0s3, id 0
 ▶ Ethernet II, Src: PcsCompu_06:03:48 (08:00:27:06:03:48), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
 ▶ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 193.136.9.183
 ▶ User Datagram Protocol, Src Port: 40723, Dst Port: 69
 ▶ Trivial File Transfer Protocol

0000	52 54 00 12 35 02 08 00	27 06 03 48 08 00 45 00	RT..5... '...H..E..
0010	00 48 49 e0 40 00 40 11	19 77 0a 00 02 0f c1 88	..HI @.@..w.....
0020	09 b7 9f 13 00 45 00 34	d7 93 00 01 66 69 6c 65	...E..4...file
0030	31 00 6f 63 74 65 74 00	74 73 69 7a 65 00 30 00	1.octet: tsize=0
0040	62 6c 6b 73 69 7a 65 00	35 31 32 00 74 69 6d 65	blksize: 512.time
0050	6f 75 74 00 00 36 00		out:6..

HTTP:

+	10	8.299197062	10.0.2.15	216.58.215.131	HTTP	194	GET / HTTP/1.1
	11	8.299404421	216.58.215.131	10.0.2.15	TCP	60 80 → 37716	[ACK]
	12	8.496783479	216.58.215.131	10.0.2.15	TCP	1462 80 → 37716	[PSH]
▶ Frame 10: 194 bytes on wire (1552 bits), 194 bytes captured (1552 bits) on interface enp0s3,							
▶ Ethernet II, Src: PcsCompu_06:03:48 (08:00:27:06:03:48), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)							
▶ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 216.58.215.131							
▶ Transmission Control Protocol, Src Port: 37716, Dst Port: 80, Seq: 1, Ack: 1, Len: 140							
▶ Hypertext Transfer Protocol							
0000	52	54	00	12	35	02	08 00 27 06 03 48 08 00 45 00 RT..5... '...H..E.
0010	00	b4	b5	07	40	00	40 06 c9 6f 0a 00 02 0f d8 3a ...@.@. .o.....
0020	d7	83	93	54	00	50	52 4b cf a1 0f 7c d8 02 50 18 ..T.PRKP.
0030	fa	f0	bc	73	00	00	47 45 54 20 2f 20 48 54 54 50 ...s...GE T / HTTP
0040	2f	31	2e	31	0d	0a	55 73 65 72 2d 41 67 65 6e 74 /1.1..Us er-Agent
0050	3a	20	57	67	65	74	2f 31 2e 32 30 2e 33 20 28 6c : Wget/1 .20.3 (1
0060	69	6e	75	78	2d	67	6e 75 29 0d 0a 41 63 63 65 70 inux-gnu)..Accep
0070	74	3a	20	2a	2f	2a	0d 0a 41 63 63 65 70 74 2d 45 t: /*... Accept-E
0080	6e	63	6f	64	69	6e	67 3a 20 69 64 65 6e 74 69 74 ncoding: identit
0090	79	0d	0a	48	6f	73	74 3a 20 77 77 77 2e 67 6f 6f y..Host: www.goo
00a0	67	6c	65	2e	70	74	0d 0a 43 6f 6e 6e 65 63 74 69 gle.pt.. Connecti
00b0	6f	6e	3a	20	4b	65	65 70 2d 41 6c 69 76 65 0d 0a on: Keep -Alive..
00c0	0d	0a					..

Nslookup:

+	3	0.328596140	10.0.2.15	192.168.43.1	DNS	73	Standard query 0x06a5 AAAA www.uminho.pt
	4	0.662996871	192.168.43.1	10.0.2.15	DNS	127	Standard query response 0x06a5 AAAA www.uminho.pt SOA dns.umi...
▶ Frame 3: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface enp0s3, id 0							
▶ Ethernet II, Src: PcsCompu_06:03:48 (08:00:27:06:03:48), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)							
▶ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 192.168.43.1							
▶ User Datagram Protocol, Src Port: 54404, Dst Port: 53							
▶ Domain Name System (query)							
0000	52	54	00	12	35	02	08 00 27 06 03 48 08 00 45 00 RT..5... '...H..E.
0010	00	3b	fc	06	40	00	40 11 46 f3 0a 00 02 0f c0 a8 ;..@. F.....
0020	2b	01	d4	84	00	35	00 27 f7 f0 06 a5 01 00 00 01 +...5.'
0030	00	00	00	00	00	00	03 77 77 77 06 75 6d 69 6e 68w ww uminh
0040	6f	02	70	74	00	00	1c 00 01 o pt.....

SSH:

	8	0.109852169	10.0.2.15	193.136.9.183	SSHv2	95	Client: Protocol (SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.3)
	9	0.110139382	193.136.9.183	10.0.2.15	TCP	60 22 → 40962	[ACK] Seq=1 Ack=42 Win=65535 Len=0
▶ Frame 8: 95 bytes on wire (760 bits), 95 bytes captured (760 bits) on interface enp0s3, id 0							
▶ Ethernet II, Src: PcsCompu_06:03:48 (08:00:27:06:03:48), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)							
▶ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 193.136.9.183							
▶ Transmission Control Protocol, Src Port: 40962, Dst Port: 22, Seq: 1, Ack: 1, Len: 41							
▶ SSH Protocol							
0000	52	54	00	12	35	02	08 00 27 06 03 48 08 00 45 00 RT..5... '...H..E.
0010	00	51	9d	85	40	00	40 06 c5 d3 0a 00 02 0f c1 88 .Q..@.
0020	09	b7	a0	02	00	16	43 39 7e dd 06 8e b6 02 50 18C9 ~.....P.
0030	fa	f0	d7	91	00	00	53 53 48 2d 32 2e 30 2d 4f 70SS H-2.0-Op
0040	65	6e	53	53	48	5f	38 2e 32 70 31 20 55 62 75 6e enSSH_8. 2p1 Ubin
0050	74	75	2d	34	75	62	75 6e 74 75 30 2e 33 0d 0a tu-4ubun tu0.3..

Conclusão

Quanto à parte pedagógica, concluímos que este trabalho foi bastante enriquecedor para todo o grupo, pelo que permitiu que aprofundássemos os nossos conhecimentos em relação aos protocolos de transporte (TCP, UDP) e também sobre o funcionamento da camada de transporte. Ajudou também a interligar os conhecimentos aprendidos na aula teórica, como protocolos de aplicação, com exemplos práticos do dia-a-dia.

Em jeito de conclusão, achamos o trabalho prático interessante e desafiador, pois teve que haver esforço de ambos os membros para garantir uma boa solução às questões propostas.