

UNIVERSIDADE DO MINHO

LICENCIATURA EM ENGENHARIA INFORMÁTICA

Comunicação por Computadores

TP1: Protocolos da Camada de Transporte

Grupo 62

Rodrigo Rodrigues (A93201) David Duarte (A93253)
João Machado (A89510)

Ano Letivo 2021/2022

Conteúdo

1	Questões e Respostas	3
1.1	Parte 1	3
1.1.1	Questão 1	3
1.1.2	Questão 2	4
1.1.3	Questão 3	5
1.1.4	Questão 4	6
1.2	Parte 2	7
1.2.1	Questão 1	7
2	Conclusão	12

Capítulo 1

Questões e Respostas

1.1 Parte 1

1.1.1 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.

TCP é um protocolo de transporte fiável que garante que todos os pacotes são enviados e recebidos com sucesso. Cliente e Servidor estabelecem conexão e comunicam entre si numa ligação contínua. Tal implica que, havendo perda de pacotes, poderá haver reenvio dos mesmos (feito pela camada de transporte). Ter perdas/-duplicações de pacotes em TCP implica haver uma resposta (reenviar, ou apagar duplicados), o que demora mais tempo. Como consequência da sobrecarga imposta sobre a rede, podemos notar delay nas aplicações.

Por outro lado, UDP é um protocolo de transporte não fiável e as mensagens são enviadas mais eficientemente/rapidamente (devido a menor complexidade) mas com menos segurança/certeza de que vão chegar ao destino (é possível averiguar através do gráfico do CORE que a ligação estabelecida entre Grilo e Servidor1 tem maior probabilidade de perder/duplicar pacotes).

1.1.2 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 controle, 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.

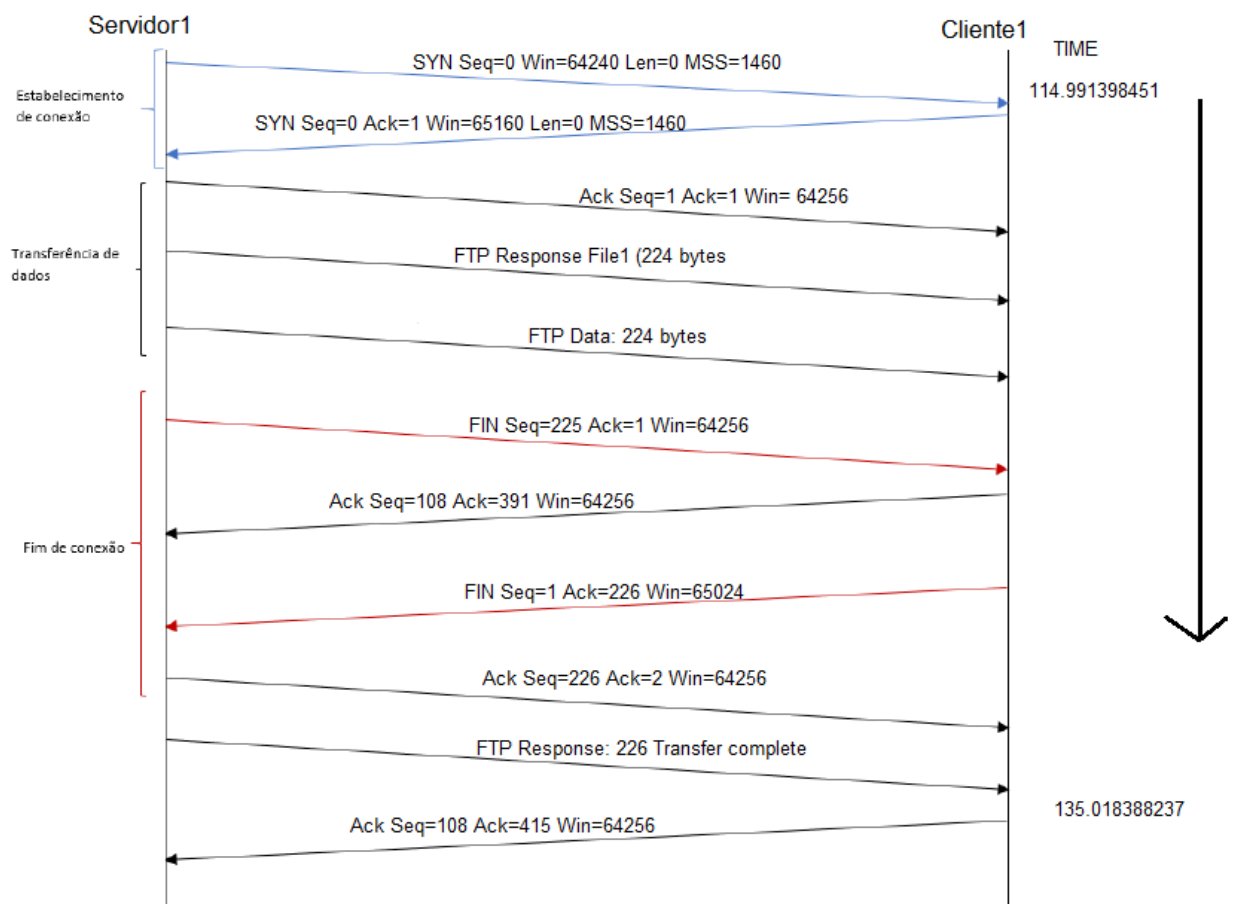


Figura 1.1: Diagrama temporal da transferência do file1 por FTP

159	135.017298946	10.2.2.1	10.1.1.1	TCP	74	20 → 60417 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
160	135.017432435	10.1.1.1	10.2.2.1	TCP	74	60417 → 20 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460
161	135.017571184	10.2.2.1	10.1.1.1	TCP	66	20 → 60417 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=241778
162	135.017609540	10.2.2.1	10.1.1.1	FTP	130	Response: 150 Opening BINARY mode data connection for file1
163	135.018022184	10.2.2.1	10.1.1.1	FTP-DA...	290	FTP Data: 224 bytes (PORT) (RETR file1)
164	135.018023832	10.2.2.1	10.1.1.1	TCP	66	20 → 60417 [FIN, ACK] Seq=225 Ack=1 Win=64256 Len=0 TSval=241778
165	135.018037002	10.1.1.1	10.2.2.1	TCP	66	59674 → 21 [ACK] Seq=108 Ack=391 Win=64256 Len=0 TSval=36288
166	135.018169881	10.1.1.1	10.2.2.1	TCP	66	60417 → 20 [ACK] Seq=1 Ack=225 Win=65024 Len=0 TSval=36288
167	135.018212834	10.1.1.1	10.2.2.1	TCP	66	60417 → 20 [FIN, ACK] Seq=1 Ack=226 Win=65024 Len=0 TSval=36288
168	135.018330347	10.2.2.1	10.1.1.1	TCP	66	20 → 60417 [ACK] Seq=226 Ack=2 Win=64256 Len=0 TSval=241778
169	135.018388237	10.2.2.1	10.1.1.1	FTP	90	Response: 226 Transfer complete.
170	135.018536724	10.1.1.1	10.2.2.1	TCP	66	59674 → 21 [ACK] Seq=108 Ack=415 Win=64256 Len=0 TSval=36288

Figura 1.2: Captura wireshark da transferência do file1 por FTP

1.1.3 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 como nas confirmações.

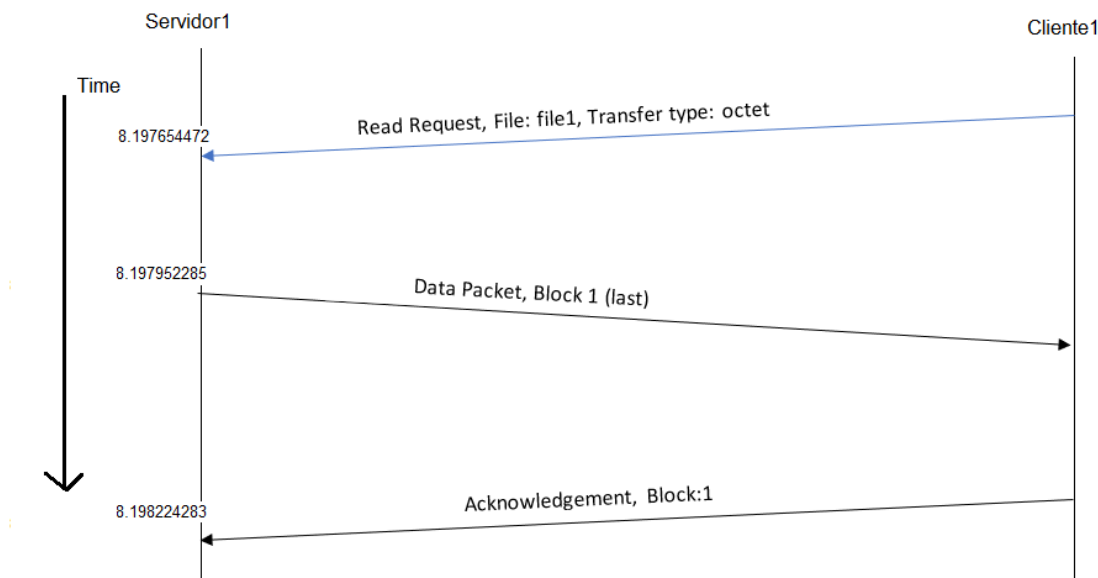


Figura 1.3: Diagrama temporal da transferência do file1 por TFTP

6	8.197654472	10.1.1.1	10.2.2.1	TFTP	56 Read Request, File: file1, Transfer type: octet
7	8.197952285	10.2.2.1	10.1.1.1	TFTP	270 Data Packet, Block: 1 (last)
8	8.198224283	10.1.1.1	10.2.2.1	TFTP	46 Acknowledgement, Block: 1

Figura 1.4: Captura wireshark da transferência do file1 por TFTP

1.1.4 Questão 4

Compare sucintamente as quatro aplicações de transferência de ficheiros que usou nos seguintes pontos (i) uso da camada de transporte; (ii) eficiência; (iii) complexidade; (iv) segurança;

FTP é um serviço básico de transferência fiável de ficheiros que utiliza o protocolo TCP como protocolo da camada de transporte. No entanto esta aplicação é menos segura do que o SFTP pois o conteúdo dos ficheiros pode ser interceptado por atores maliciosos antes de chegar ao recetor, uma vez que os dados não são criptografados, apesar de utilizar Login. Apresenta problemas de eficiência devido a overhead elevado.

SFTP é uma aplicação mais segura do que FTP (uma vez que usa canal seguro para transferir ficheiros, onde os dados são criptografados de forma a não serem interceptados por atores maliciosos e usa Login). A utilização de SSH faz com que SFTP tenha overhead maior logo diminui a eficiência por ser mais complexo do que as outras aplicações. Utiliza o protocolo TCP como protocolo da camada de transporte.

TFTP é um serviço básico de transferência não fiável de ficheiros, dado que usa UDP. Não apresenta segurança adicional (tal como o FTP) nem mecanismos de autenticação. O baixo overhead faz com este seja um protocolo muito eficiente na transmissão de dados.

A transferência de dados por HTTP é insegura uma vez que qualquer pessoa na rede consegue ver o conteúdo dos ficheiros antes de chegarem ao recetor. Este serviço, tal como FTP e SFTP, utiliza TCP como protocolo da camada de transporte. Digamos que é o 2º mais eficiente, depois de TFTP.

Por ordem de complexidade (do mais complexo para o menos complexo): SFTP, HTTP, FTP, TFTP

Por ordem de eficiência (do mais eficiente para o menos eficiente) TFTP, HTTP, FTP, SFTP

1.2 Parte 2

1.2.1 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 (aplicação)	Protocolo de Aplicação (se aplicável)	Protocolo de transporte (se aplicável)	Porta de atendimento (se aplicável)	<i>Overhead</i> de transporte em bytes (se aplicável)
ping	-	-	-	-
tracert	-	UDP	33436	8
telnet	TELNET	TCP	23	20
ftp	FTP	TCP	21	20
tftp	TFTP	UDP	69	8
http(browser)	HTTP	TCP	80	20
nslookup	DNS	UDP	53	8
ssh	SSH v2	TCP	22	20
Outras:				

Figura 1.5: Tabela

icmp							
No.	Time	Source	Destination	Protocol	Length	Info	
1063	13.51530737421	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=1/256, ttl=57 (request in 1062)
1065	13.605493163547	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=10/2560, ttl=57 (request in 1064)
1087	13.61539953420	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=11/2816, ttl=57 (request in 1086)
1089	13.62546356207	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=12/3072, ttl=57 (request in 1088)
1091	13.63551919735	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=13/3328, ttl=57 (request in 1090)
1093	13.64557282583	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=14/3584, ttl=57 (request in 1092)
1095	13.65542781236	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=15/3840, ttl=57 (request in 1094)
1097	13.66536960140	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=16/4096, ttl=57 (request in 1096)
1099	13.67542081956	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=17/4352, ttl=57 (request in 1098)
1101	13.68554060645	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=18/4608, ttl=57 (request in 1098)
1103	13.69542821766	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=19/4864, ttl=57 (request in 1102)
1067	13.52529093854	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=2/512, ttl=57 (request in 1066)
1105	13.70552831061	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=20/5120, ttl=57 (request in 1104)
1109	13.71553490517	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=21/5376, ttl=57 (request in 1108)
1111	13.72573225211	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=22/5632, ttl=57 (request in 1110)
1113	13.73552960754	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=23/5888, ttl=57 (request in 1112)
1115	13.74573213434	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=24/6144, ttl=57 (request in 1114)
1117	13.75582262281	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=25/6400, ttl=57 (request in 1116)
1119	13.76582222051	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=26/6656, ttl=57 (request in 1118)
1123	13.77589208000	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=27/6912, ttl=57 (request in 1122)
1125	13.78589117427	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=28/7168, ttl=57 (request in 1124)
1127	13.79590830951	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=29/7424, ttl=57 (request in 1126)
1069	13.53536015960	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=3/768, ttl=57 (request in 1068)
1129	13.8057977435	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=30/7680, ttl=57 (request in 1128)
1131	13.81580226707	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=31/7936, ttl=57 (request in 1130)
1133	13.82599126933	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=32/8192, ttl=57 (request in 1132)
1135	13.83580809080	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=33/8448, ttl=57 (request in 1134)
1137	13.84581344439	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=34/8704, ttl=57 (request in 1136)
1139	13.85591176559	142.250.178.103	10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0001, seq=35/8960, ttl=57 (request in 1138)
▶ Frame 98: 98 bytes on wire (784 bits), 98 bytes captured (784 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: 142.250.178.103, Dst: 10.0.2.15							
▶ Internet Control Message Protocol							
Type: 0 (Echo (ping) reply)							
Code: 0							
Checksum: 0x84d0 [correct]							
[Checksum Status: Good]							
Identifier (BE): 1 (0x0001)							
Identifier (LE): 256 (0x0100)							
Sequence number (BE): 1 (0x0001)							
Sequence number (LE): 256 (0x0100)							
[Request frame: 1062]							
[Response time: 51,838 ms]							
Timestamp from icmp data: Oct 24, 2021 13:15:37.000000000 WEST							
[Timestamp from icmp data (relative): 0.421722169 seconds]							
▶ Data (48 bytes)							
Data: d8a4050000000000011112131415161718191a1b1c1d1e1f...							
[Length: 48]							

Figura 1.6: Captura wireshark: ping

Ping: Não tem aplicação, não tem protocolo de transporte porque corre diretamente em ICMP, não tem porta de atendimento

icmp							
No.	Time	Source	Destination	Protocol	Length	Info	
1118	1376.520657224	10.0.2.15	142.250.178.103	ICMP	98	Echo (ping) request id=0x0001, seq=26/6556, ttl=64 (reply in 1119)	
1122	1377.522323056	10.0.2.15	142.250.178.103	ICMP	98	Echo (ping) request id=0x0001, seq=27/6912, ttl=64 (reply in 1123)	
1124	1378.524074253	10.0.2.15	142.250.178.103	ICMP	98	Echo (ping) request id=0x0001, seq=28/7168, ttl=64 (reply in 1125)	
1126	1379.527060574	10.0.2.15	142.250.178.103	ICMP	98	Echo (ping) request id=0x0001, seq=29/7424, ttl=64 (reply in 1127)	
1068	1353.482392642	10.0.2.15	142.250.178.103	ICMP	98	Echo (ping) request id=0x0001, seq=3/768, ttl=64 (reply in 1069)	
1128	1380.528866350	10.0.2.15	142.250.178.103	ICMP	98	Echo (ping) request id=0x0001, seq=30/7680, ttl=64 (reply in 1129)	
1130	1381.529370920	10.0.2.15	142.250.178.103	ICMP	98	Echo (ping) request id=0x0001, seq=31/7936, ttl=64 (reply in 1131)	
1132	1382.530771247	10.0.2.15	142.250.178.103	ICMP	98	Echo (ping) request id=0x0001, seq=32/8192, ttl=64 (reply in 1133)	
1134	1383.532170600	10.0.2.15	142.250.178.103	ICMP	98	Echo (ping) request id=0x0001, seq=33/8448, ttl=64 (reply in 1135)	
1136	1384.534100922	10.0.2.15	142.250.178.103	ICMP	98	Echo (ping) request id=0x0001, seq=34/8704, ttl=64 (reply in 1137)	
1138	1385.535580497	10.0.2.15	142.250.178.103	ICMP	98	Echo (ping) request id=0x0001, seq=35/8960, ttl=64 (reply in 1139)	
1140	1386.537400101	10.0.2.15	142.250.178.103	ICMP	98	Echo (ping) request id=0x0001, seq=36/9216, ttl=64 (reply in 1141)	
1142	1387.538597791	10.0.2.15	142.250.178.103	ICMP	98	Echo (ping) request id=0x0001, seq=37/9472, ttl=64 (reply in 1143)	
1144	1388.539074281	10.0.2.15	142.250.178.103	ICMP	98	Echo (ping) request id=0x0001, seq=38/9728, ttl=64 (reply in 1145)	
1146	1389.541543215	10.0.2.15	142.250.178.103	ICMP	98	Echo (ping) request id=0x0001, seq=39/9984, ttl=64 (reply in 1147)	
1070	1354.484719905	10.0.2.15	142.250.178.103	ICMP	98	Echo (ping) request id=0x0001, seq=4/1024, ttl=64 (reply in 1071)	
1148	1390.543198361	10.0.2.15	142.250.178.103	ICMP	98	Echo (ping) request id=0x0001, seq=40/10240, ttl=64 (reply in 1149)	
1150	1391.545060838	10.0.2.15	142.250.178.103	ICMP	98	Echo (ping) request id=0x0001, seq=41/10496, ttl=64 (reply in 1151)	
1152	1392.545502171	10.0.2.15	142.250.178.103	ICMP	98	Echo (ping) request id=0x0001, seq=42/10752, ttl=64 (reply in 1153)	
1154	1393.546606478	10.0.2.15	142.250.178.103	ICMP	98	Echo (ping) request id=0x0001, seq=43/11008, ttl=64 (reply in 1155)	
1072	1355.486222520	10.0.2.15	142.250.178.103	ICMP	98	Echo (ping) request id=0x0001, seq=5/1280, ttl=64 (reply in 1073)	
1074	1356.487517198	10.0.2.15	142.250.178.103	ICMP	98	Echo (ping) request id=0x0001, seq=6/1536, ttl=64 (reply in 1075)	
1076	1357.488597844	10.0.2.15	142.250.178.103	ICMP	98	Echo (ping) request id=0x0001, seq=7/1792, ttl=64 (reply in 1079)	
1080	1358.490493988	10.0.2.15	142.250.178.103	ICMP	98	Echo (ping) request id=0x0001, seq=8/2048, ttl=64 (reply in 1081)	
1082	1359.493253903	10.0.2.15	142.250.178.103	ICMP	98	Echo (ping) request id=0x0001, seq=9/2304, ttl=64 (reply in 1083)	
1200	1556.563736768	10.0.2.2	10.0.2.15	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)	
1208	1556.563785661	10.0.2.2	10.0.2.15	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)	
1301	1556.563874224	10.0.2.2	10.0.2.15	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)	

▶ Frame 1301: 70 bytes on wire (560 bits), 70 bytes captured (560 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: 10.0.2.2, Dst: 10.0.2.15

▶ Internet Control Message Protocol

Type: 11 (Time-to-live exceeded)
Code: 0 (Time to live exceeded in transit)
Checksum: 0xcdb3 [correct]
[Checksum Status: Good]
Unused: 00000000

▶ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 193.136.19.254

▶ User Datagram Protocol, Src Port: 48698, Dst Port: 33436

Source Port: 48698
Destination Port: 33436
Length: 40
Checksum: 0xe83e [unverified]
[Checksum Status: Unverified]
[Stream index: 90]

Figura 1.7: Captura wireshark: traceroute

Traceroute: O overhead de transporte é $40 - \text{payload}$, ou seja, $40 - 32 = 8$ bytes pelo que se tem $8/40 = 0.20$ ou seja 20% de overhead.

telnet						
No.	Time	Source	Destination	Protocol	Length	Info
710	777.076334688	10.0.2.15	193.136.9.183	TELNET	81	Telnet Data ...
712	787.749354184	10.0.2.15	193.136.9.183	TELNET	58	Telnet Data ...
714	792.721690893	193.136.9.183	10.0.2.15	TELNET	66	Telnet Data ...
716	792.796688982	193.136.9.183	10.0.2.15	TELNET	93	Telnet Data ...
718	792.796799686	10.0.2.15	193.136.9.183	TELNET	147	Telnet Data ...
720	792.833729598	193.136.9.183	10.0.2.15	TELNET	60	Telnet Data ...
722	792.833846435	10.0.2.15	193.136.9.183	TELNET	57	Telnet Data ...
724	792.872998214	193.136.9.183	10.0.2.15	TELNET	60	Telnet Data ...
726	792.873132448	10.0.2.15	193.136.9.183	TELNET	57	Telnet Data ...
728	792.916803591	193.136.9.183	10.0.2.15	TELNET	105	Telnet Data ...
730	793.849033741	10.0.2.15	193.136.9.183	TELNET	55	Telnet Data ...
732	795.079115184	10.0.2.15	193.136.9.183	TELNET	55	Telnet Data ...
734	795.237201390	10.0.2.15	193.136.9.183	TELNET	55	Telnet Data ...
736	795.388165493	10.0.2.15	193.136.9.183	TELNET	55	Telnet Data ...
738	796.508252897	10.0.2.15	193.136.9.183	TELNET	55	Telnet Data ...
740	796.605213841	10.0.2.15	193.136.9.183	TELNET	55	Telnet Data ...
744	799.522979818	10.0.2.15	193.136.9.183	TELNET	55	Telnet Data ...
746	799.608591518	10.0.2.15	193.136.9.183	TELNET	55	Telnet Data ...
748	799.849088179	10.0.2.15	193.136.9.183	TELNET	55	Telnet Data ...
750	799.907777958	10.0.2.15	193.136.9.183	TELNET	55	Telnet Data ...
752	800.498036834	10.0.2.15	193.136.9.183	TELNET	55	Telnet Data ...
754	800.627911376	10.0.2.15	193.136.9.183	TELNET	55	Telnet Data ...
756	800.797926353	10.0.2.15	193.136.9.183	TELNET	55	Telnet Data ...
▶ Frame 710: 81 bytes on wire (648 bits), 81 bytes captured (648 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: 42054, Dst Port: 23, Seq: 1, Ack: 1, Len: 27 Source Port: 42054 Destination Port: 23 [Stream index: 17] [TCP Segment Len: 27] Sequence number: 1 (relative sequence number) Sequence number (raw): 3957932896 [Next sequence number: 28 (relative sequence number)] Acknowledgment number: 1 (relative ack number) Acknowledgment number (raw): 25088002 0101 = Header Length: 20 bytes (5) ▶ Flags: 0x018 (PSH, ACK) Window size value: 64240 [Calculated window size: 64240] [Window size scaling factor: -2 (no window scaling used)] Checksum: 0xd783 [unverified] [Checksum Status: Unverified] Urgent pointer: 0 ▶ [SEQ/ACK analysis] ▶ [Timestamps] TCP payload (27 bytes) ▶ Telnet						

Figura 1.8: Captura wireshark: telnet

ftp						
No.	Time	Source	Destination	Protocol	Length	Info
151	288.925859859	10.0.2.15	193.136.9.183	FTP	67	Request: PASS cc2022
153	288.941200177	10.0.2.15	193.136.9.183	FTP	69	Request: USER cc
147	286.961392627	10.0.2.15	193.136.9.183	FTP	63	Request: USER cc
157	289.086051175	193.136.9.183	10.0.2.15	FTP	73	Response: 215 UNIX Type: L8
145	284.009059870	193.136.9.183	10.0.2.15	FTP	74	Response: 220 (vsFTPd 2.3.5)
153	289.041280722	193.136.9.183	10.0.2.15	FTP	77	Response: 230 Login successful.
140	286.060760451	193.136.9.183	10.0.2.15	FTP	88	Response: 331 Please specify the password.
▶ Frame 155: 60 bytes on wire (480 bits), 60 bytes captured (480 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: 33584, Dst Port: 21, Seq: 23, Ack: 78, Len: 6 Source Port: 33584 Destination Port: 21 [Stream index: 1] [TCP Segment Len: 6] Sequence number: 23 (relative sequence number) Sequence number (raw): 674189156 [Next sequence number: 29 (relative sequence number)] Acknowledgment number: 78 (relative ack number) Acknowledgment number (raw): 3456079 0101 = Header Length: 20 bytes (5) ▶ Flags: 0x018 (PSH, ACK) Window size value: 64163 [Calculated window size: 64163] [Window size scaling factor: -2 (no window scaling used)] Checksum: 0xd76e [unverified] [Checksum Status: Unverified] Urgent pointer: 0 ▶ [SEQ/ACK analysis] ▶ [Timestamps] TCP payload (6 bytes) ▶ File Transfer Protocol (FTP) [Current working directory:]						

Figura 1.9: Captura wireshark: FTP

tftp						
No.	Time	Source	Destination	Protocol	Length	Info
609	476.056082048	10.0.2.15	193.136.9.183	TFTP	86	Read Request, File: file1, Transfer type: octet, tsize=0, blksize=512, timeout=6
610	482.265554097	10.0.2.15	193.136.9.183	TFTP	86	Read Request, File: file1, Transfer type: octet, tsize=0, blksize=512, timeout=6
611	489.272878296	10.0.2.15	193.136.9.183	TFTP	86	Read Request, File: file1, Transfer type: octet, tsize=0, blksize=512, timeout=6
612	496.283286773	10.0.2.15	193.136.9.183	TFTP	86	Read Request, File: file1, Transfer type: octet, tsize=0, blksize=512, timeout=6
623	503.292652461	10.0.2.15	193.136.9.183	TFTP	86	Read Request, File: file1, Transfer type: octet, tsize=0, blksize=512, timeout=6
630	510.302819483	10.0.2.15	193.136.9.183	TFTP	86	Read Request, File: file1, Transfer type: octet, tsize=0, blksize=512, timeout=6
631	517.313651509	10.0.2.15	193.136.9.183	TFTP	86	Read Request, File: file1, Transfer type: octet, tsize=0, blksize=512, timeout=6
632	524.323313322	10.0.2.15	193.136.9.183	TFTP	86	Read Request, File: file1, Transfer type: octet, tsize=0, blksize=512, timeout=6

```

▶ Frame 609: 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: 49642, Dst Port: 69
  Source Port: 49642
  Destination Port: 69
  Length: 52
  Checksum: 0xd793 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 73]
  ▼ [Timestamps]
    [Time since first frame: 0.000000000 seconds]
    [Time since previous frame: 0.000000000 seconds]
▶ Trivial File Transfer Protocol

```

Figura 1.10: Captura wireshark: TFTP

http						
No.	Time	Source	Destination	Protocol	Length	Info
11	23.844639299	10.0.2.15	193.136.9.240	HTTP	214	GET /disciplinas/CC-LEI HTTP/1.1
13	23.897283614	193.136.9.240	10.0.2.15	HTTP	616	HTTP/1.1 301 Moved Permanently (text/html)
15	23.897578785	10.0.2.15	193.136.9.240	HTTP	215	GET /disciplinas/CC-LEI/ HTTP/1.1
27	23.938188466	193.136.9.240	10.0.2.15	HTTP	552	HTTP/1.1 200 OK (text/html)


```

▶ Frame 15: 215 bytes on wire (1720 bits), 215 bytes captured (1720 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: 193.136.9.240
▼ Transmission Control Protocol, Src Port: 38186, Dst Port: 80, Seq: 161, Ack: 563, Len: 161
  Source Port: 38186
  Destination Port: 80
  [Stream index: 0]
  [TCP Segment Len: 161]
  Sequence number: 161 (relative sequence number)
  Sequence number (raw): 1956449598
  [Next sequence number: 322 (relative sequence number)]
  Acknowledgment number: 563 (relative ack number)
  Acknowledgment number (raw): 3328564
  0101 .... = Header Length: 20 bytes (5)
  ▶ Flags: 0x018 (PSH, ACK)
    Window size value: 64060
    [Calculated window size: 64060]
    [Window size scaling factor: -2 (no window scaling used)]
    Checksum: 0xd842 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▶ [SEQ/ACK analysis]
  ▶ [Timestamps]
    TCP payload (161 bytes)
▶ Hypertext Transfer Protocol

```

Figura 1.11: Captura wireshark: HTTP

dns							
Io.	Time	Source	Destination	Protocol	Length	Info	
282	382.802565884	10.0.2.15	192.168.1.1	DNS	79	Standard query 0x1aab AAAA elearning.uminho.pt	
278	382.799829883	10.0.2.15	192.168.1.1	DNS	77	Standard query 0x1dd2 AAAA ww5.di.uminho.pt	
481	386.016110628	10.0.2.15	192.168.1.1	DNS	74	Standard query 0x21b6 A www.reddit.com	
322	382.984466915	10.0.2.15	192.168.1.1	DNS	84	Standard query 0x31f5 AAAA autopush.prod.mozaws.net	
393	382.980741381	10.0.2.15	192.168.1.1	DNS	76	Standard query 0x32bb AAAA books.google.com	
270	382.732371227	10.0.2.15	192.168.1.1	DNS	74	Standard query 0x33de AAAA r3.o.lencr.org	
284	382.803390391	10.0.2.15	192.168.1.1	DNS	83	Standard query 0x3683 A e11138.x.akamaiedge.net	
654	636.018218588	10.0.2.15	192.168.1.1	DNS	74	Standard query 0x39c6 AAAA cc2022.dns.net	
376	383.317996637	10.0.2.15	192.168.1.1	DNS	78	Standard query 0x3dd3 AAAA cs9.wac.phicdn.net	
392	382.980461725	10.0.2.15	192.168.1.1	DNS	76	Standard query 0x51f5 A books.google.com	
130	283.832966798	10.0.2.15	192.168.1.1	DNS	75	Standard query 0x5652 AAAA cc2022.ddns.net	
224	382.401972696	10.0.2.15	192.168.1.1	DNS	70	Standard query 0x5ac1 AAAA www.w3.org	
6	23.780508514	10.0.2.15	192.168.1.1	DNS	75	Standard query 0x5b73 AAAA marco.uminho.pt	
793	777.548164856	10.0.2.15	192.168.1.1	DNS	75	Standard query 0x6130 A cc2022.dns.net	
343	383.179291269	10.0.2.15	192.168.1.1	DNS	103	Standard query 0x631c AAAA prod.ingestion-edge.prod.dataops.mozgcp.net	
+	997.1084.484661705	10.0.2.15	192.168.1.1	DNS	73	Standard query 0x6430 AAAA www.uminho.pt	
	222.382.409577074	10.0.2.15	192.168.1.1	DNS	75	Standard query 0x65f0 AAAA marco.uminho.pt	
	295.382.648701793	10.0.2.15	192.168.1.1	DNS	81	Standard query 0x69ae AAAA e290.x.akamaiedge.net	
	689.758.606195407	10.0.2.15	192.168.1.1	DNS	74	Standard query 0x6a41 A cc2022.dns.net	
	209.382.206939859	10.0.2.15	192.168.1.1	DNS	84	Standard query 0x713e AAAA detectportal.firefox.com	
	474.386.850285359	10.0.2.15	192.168.1.1	DNS	87	Standard query 0x7295 AAAA star-mini.c10r.facebook.com	
	242.382.527195279	10.0.2.15	192.168.1.1	DNS	88	Standard query 0x7c02 A cc2022.dns.net	
	243.382.527326798	10.0.2.15	192.168.1.1	DNS	88	Standard query 0x80c2 AAAA contile.services.mozilla.com	
▶ Frame 997: 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: 192.168.1.1							
▶ User Datagram Protocol, Src Port: 38070, Dst Port: 53							
Source Port: 38070							
Destination Port: 53							
Length: 39							
Checksum: 0xcdfe [unverified]							
[Checksum Status: Unverified]							
[Stream index: 82]							
▼ [Timestamps]							
[Time since first frame: 0.000000000 seconds]							
[Time since previous frame: 0.000000000 seconds]							
Domain Name System (query)							
Transaction ID: 0x0430							
▶ Flags: 0x0100 Standard query							
Questions: 1							
Answer RRs: 0							
Authority RRs: 0							
Additional RRs: 0							
▶ Queries							
[Response in: 998]							

Figura 1.12: Captura wireshark: nslookup

ssh						
No.	Time	Source	Destination	Protocol	Length	Info
883	944.924839984	193.136.9.183	10.0.2.15	SSHv2	95	Server: Protocol (SSH-2.0-openssh_5.9p1 Debian-Subuntu1.4)
888	944.964997405	193.136.9.183	10.0.2.15	SSHv2	1038	Server: Key Exchange Init
903	946.145139629	193.136.9.183	10.0.2.15	SSHv2	118	Server: Encrypted packet (len=56)
898	945.084752390	193.136.9.183	10.0.2.15	SSHv2	94	Server: Encrypted packet (len=40)
892	945.033882195	193.136.9.183	10.0.2.15	SSHv2	366	Server: Elliptic Curve Diffie-Hellman Key Exchange Reply, New Keys
881	944.848192183	10.0.2.15	193.136.9.183	SSHv2	95	Client: Protocol (SSH-2.0-openssh_8.2p1 Ubuntu-4ubuntu0.3)
894	945.034498040	10.0.2.15	193.136.9.183	SSHv2	70	Client: New Keys
885	944.927223707	10.0.2.15	193.136.9.183	SSHv2	1566	Client: Key Exchange Init
899	945.084829925	10.0.2.15	193.136.9.183	SSHv2	110	Client: Encrypted packet (len=56)
896	945.041727732	10.0.2.15	193.136.9.183	SSHv2	94	Client: Encrypted packet (len=40)
890	944.956694683	10.0.2.15	193.136.9.183	SSHv2	134	Client: Elliptic Curve Diffie-Hellman Key Exchange Init

```

▶ Frame 881: 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: 32816, Dst Port: 22, Seq: 1, Ack: 1, Len: 41
  Source Port: 32816
  Destination Port: 22
  [Stream index: 18]
  [TCP Segment Len: 41]
  Sequence number: 1 (relative sequence number)
  Sequence number (raw): 1100937076
  [Next sequence number: 42 (relative sequence number)]
  Acknowledgment number: 1 (relative ack number)
  Acknowledgment number (raw): 31552002
  0101 .... = Header Length: 20 bytes (5)
  ▶ Flags: 0x018 (PSH, ACK)
  Window size value: 64240
  [Calculated window size: 64240]
  [Window size scaling factor: -2 (no window scaling used)]
  Checksum: 0xd791 [unverified]
  [Checksum Status: Unverified]
  urgent pointer: 0
  ▶ [SEQ/ACK analysis]
  ▶ [Timestamps]
  TCP payload (41 bytes)
  ▶ SSH Protocol

```

Figura 1.13: Captura wireshark: SSH

Capítulo 2

Conclusão

Ao longo deste trabalho pudemos pôr em prática os conhecimentos adquiridos nas aulas teóricas acerca de Protocolos da Camada de Transporte bem como consolidá-los.

TCP - Transmission Control Protocol, é caracterizado por ser um protocolo da camada de transporte muito completo, pois tem controlo de fluxo, erros e congestão.

Por outro lado, o protocolo UDP - User Datagram Protocol é caracterizado por ser um protocolo menos complexo da camada de transporte, pois não apresenta controlos tal como o TCP. No entanto apresenta vantagens para aplicações (devido a maior velocidade de transmissão de dados) como, por exemplo streaming, uma vez que não perde tempo com controlo de erros, congestão, etc.

A escolha do protocolo de transporte a utilizar deve ter em conta o tipo de aplicação que se pretende e as vantagens/desvantagens oferecidas por cada protocolo(que conseguimos averiguar ao longo deste trabalho).

Utilizando ferramentas como o *CORE* e o *Wireshark* conseguimos explorar de forma concreta a grande maioria das propriedades que caracterizam cada uma das aplicações de transferência de dados que consideramos.

Tiramos ainda conclusões acerca da segurança das aplicações, sendo SFTP a mais segura, seguida de FTP, HHTTP e, por fim, TFTP que é a menos segura pelo que usa o protocolo UDP.