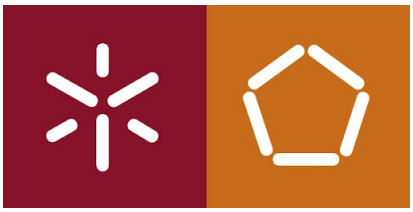

Universidade do Minho

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

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 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.

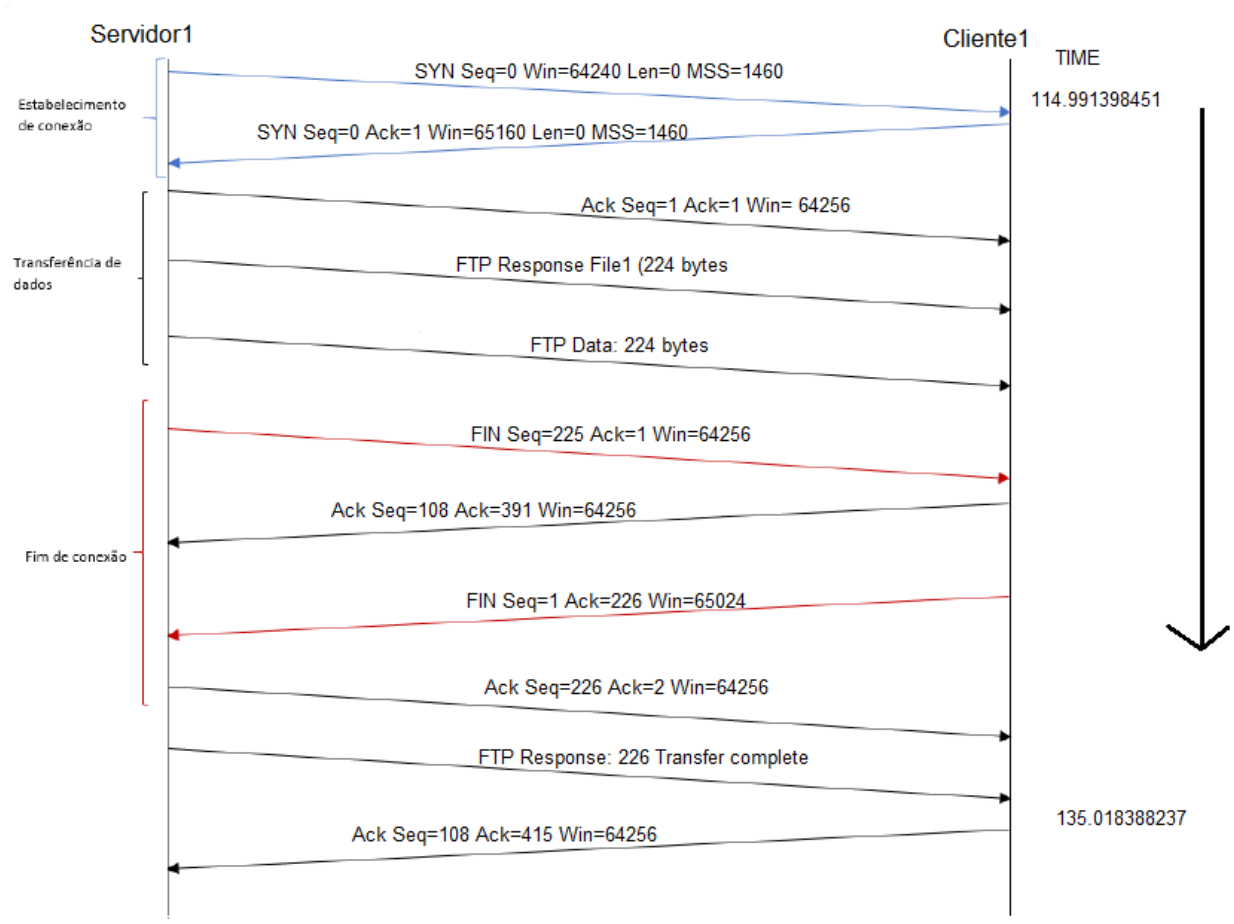

Figura 1.1: Diagrama temporal da transferência do file1 por FTP

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=
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=146
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 file
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=
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=362
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=3628
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=
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=2417
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=362

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.

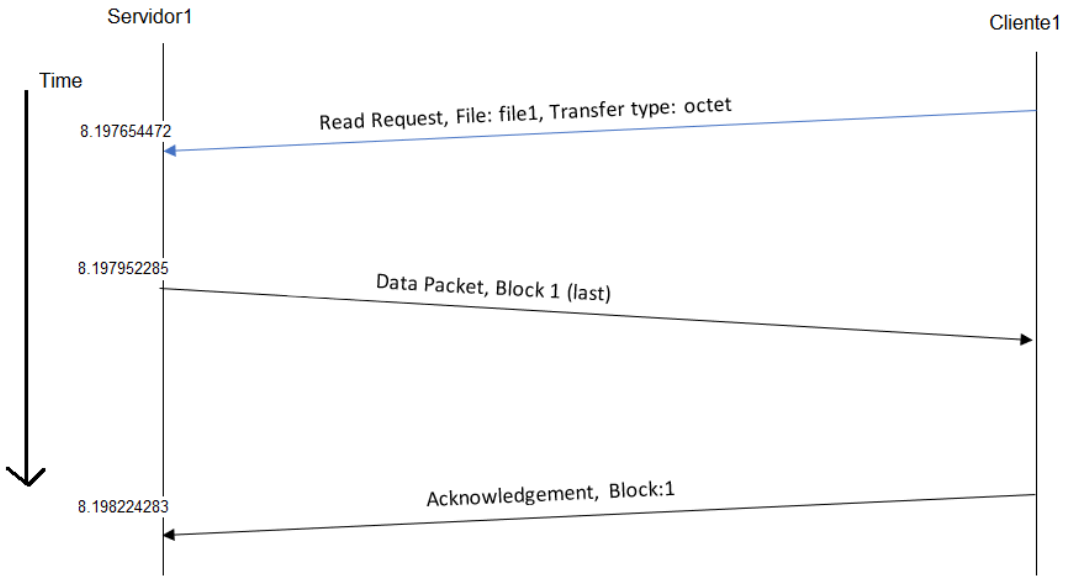

Figura 1.3: Diagrama temporal da transferência do file1 por TFTP

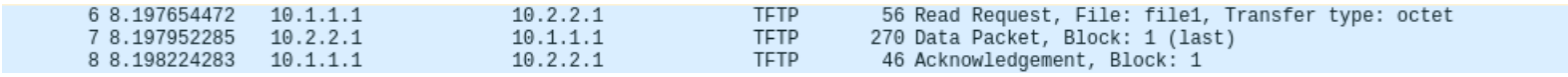

Figura 1.4: Captura wireshark da transferência do file1 por TFTP

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 intercetado 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 intercetados 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)**	***Overhead* de transporte em bytes (se aplicável)**
`ping`	-	-	-	-
`traceroute`	-	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

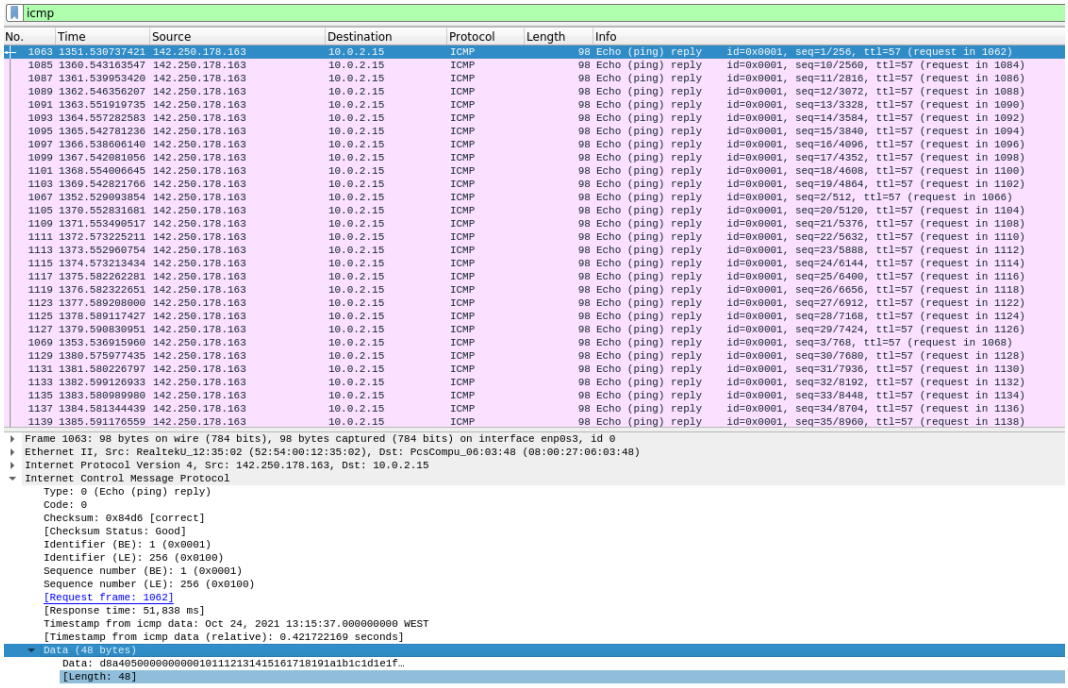

Figura 1.6: Captura wireshark: ping

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

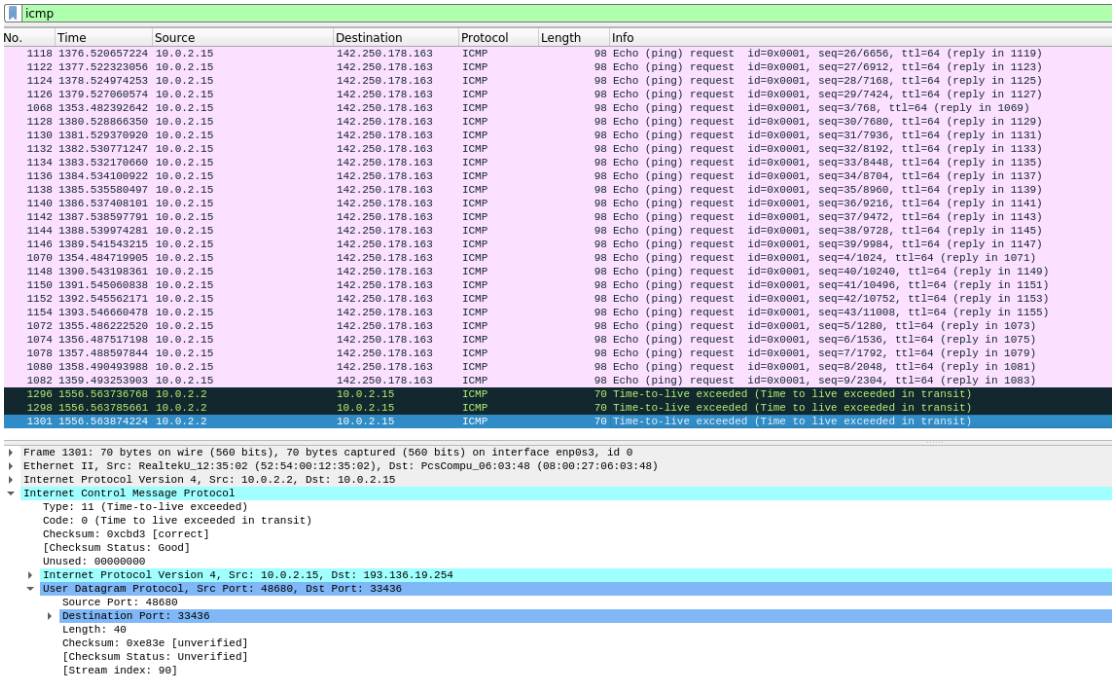

Figura 1.7: Captura wireshark: traceroute

Traceroute: O overhead de transporte é 40 – payload, ou seja, 40 – 32 = 8 bytes pelo que se tem 8/40 = 0.20 ou seja 20% de overhead.

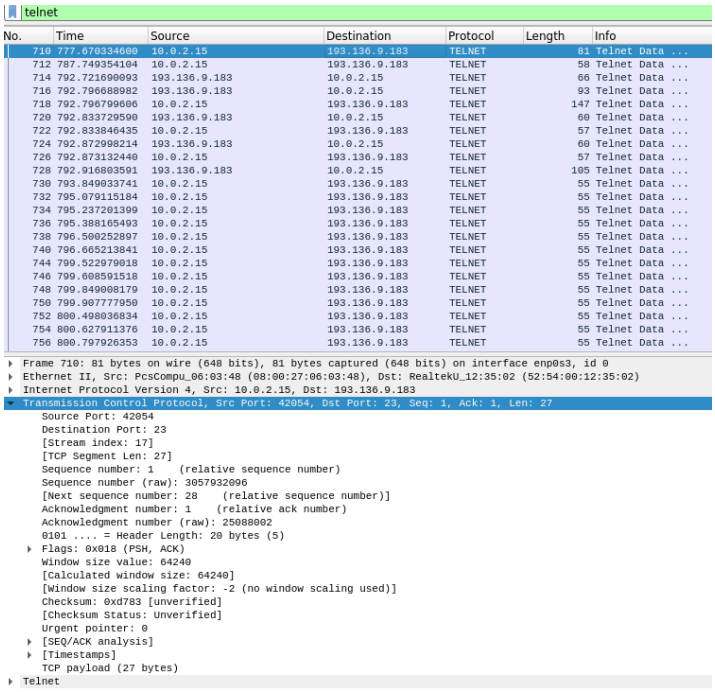

Figura 1.8: Captura wireshark: telnet

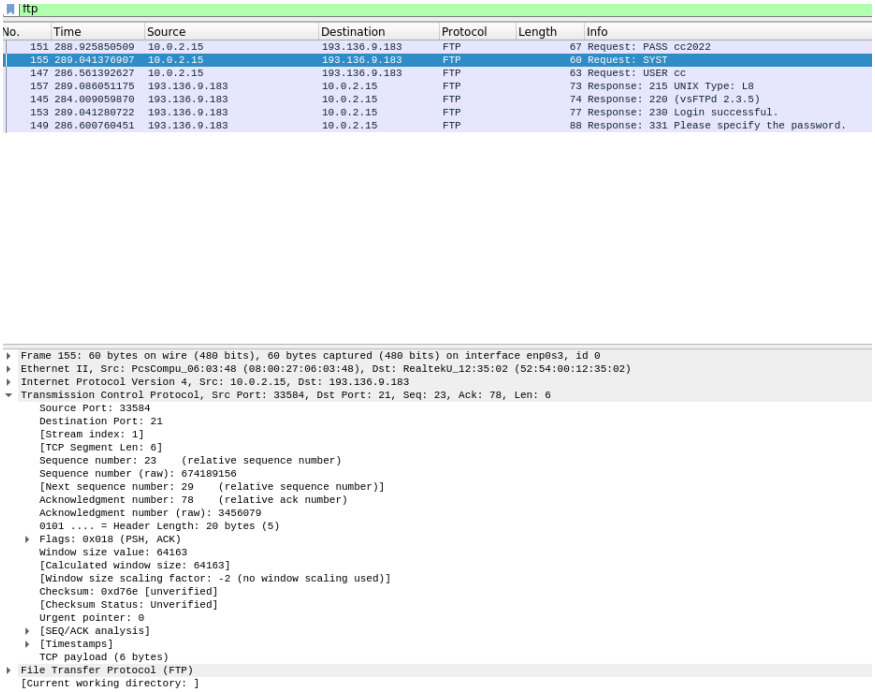

Figura 1.9: Captura wireshark: FTP

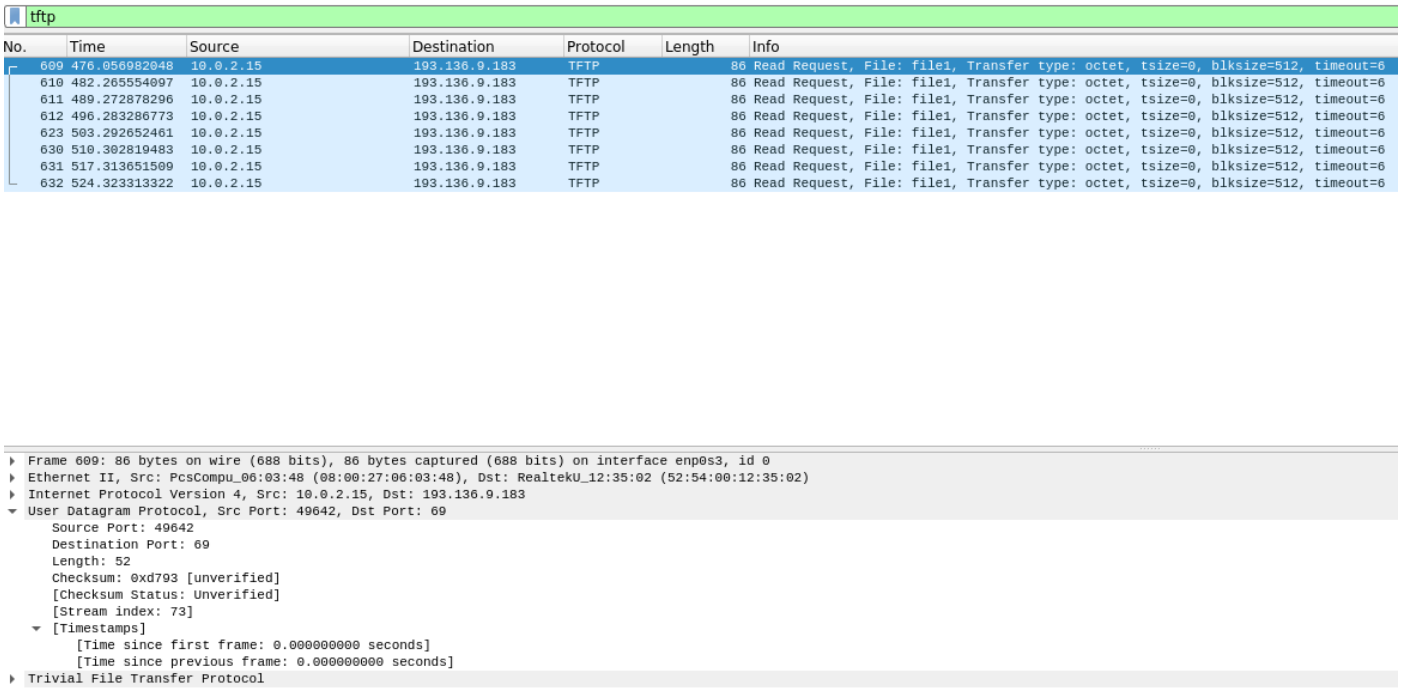

Figura 1.10: Captura wireshark: TFTP

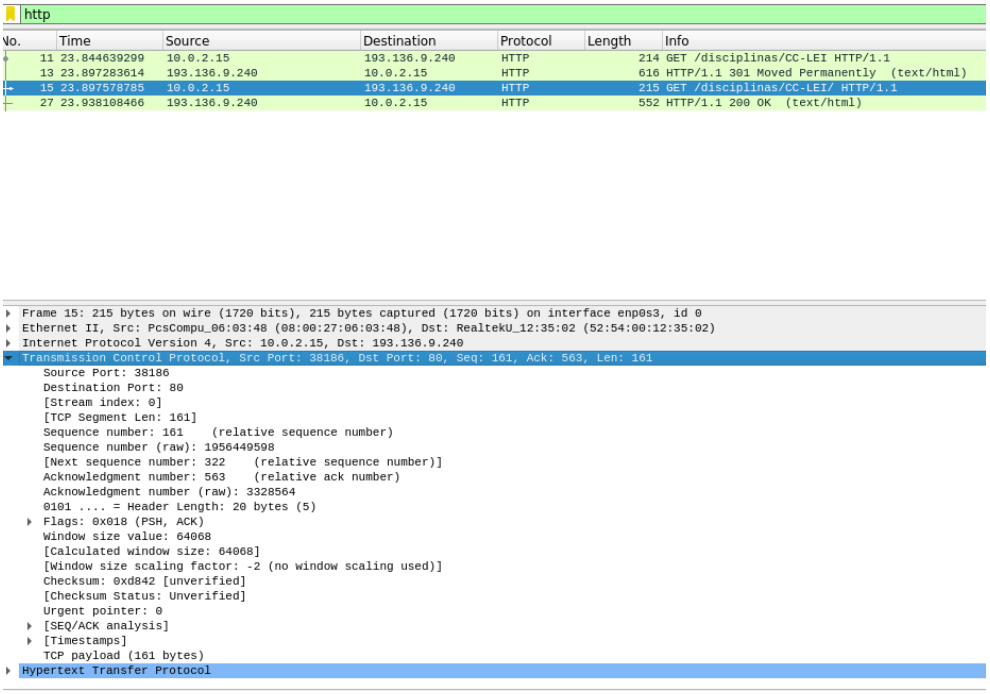

Figura 1.11: Captura wireshark: HTTP

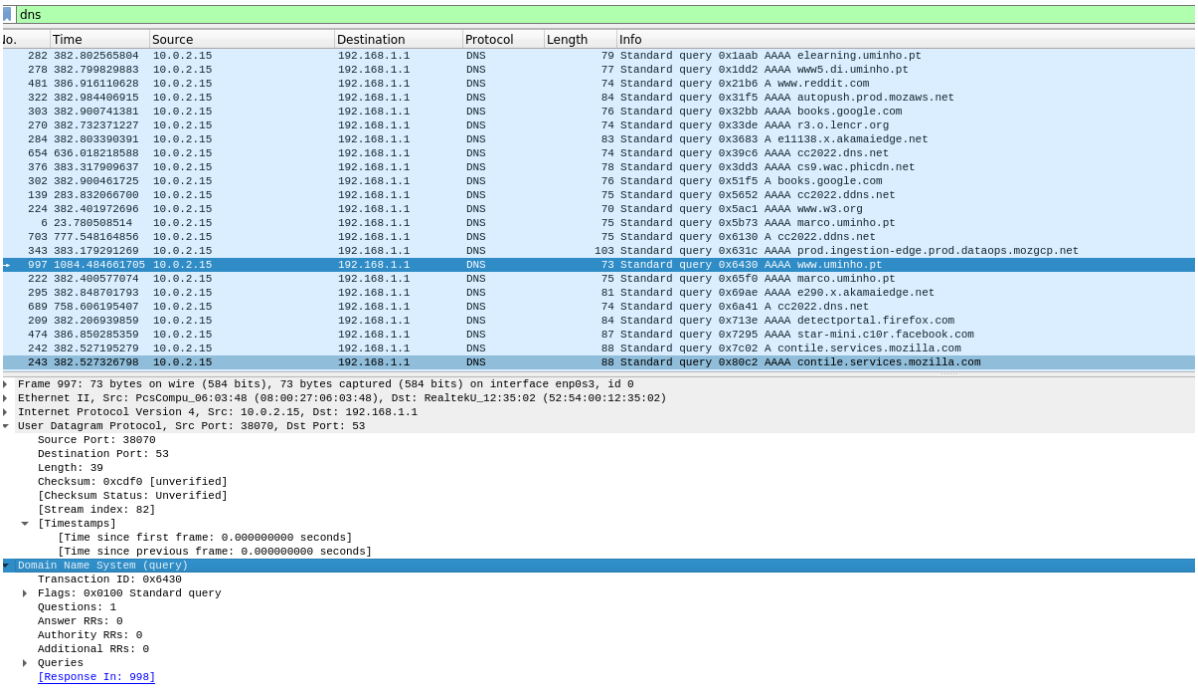

Figura 1.12: Captura wireshark: nslookup

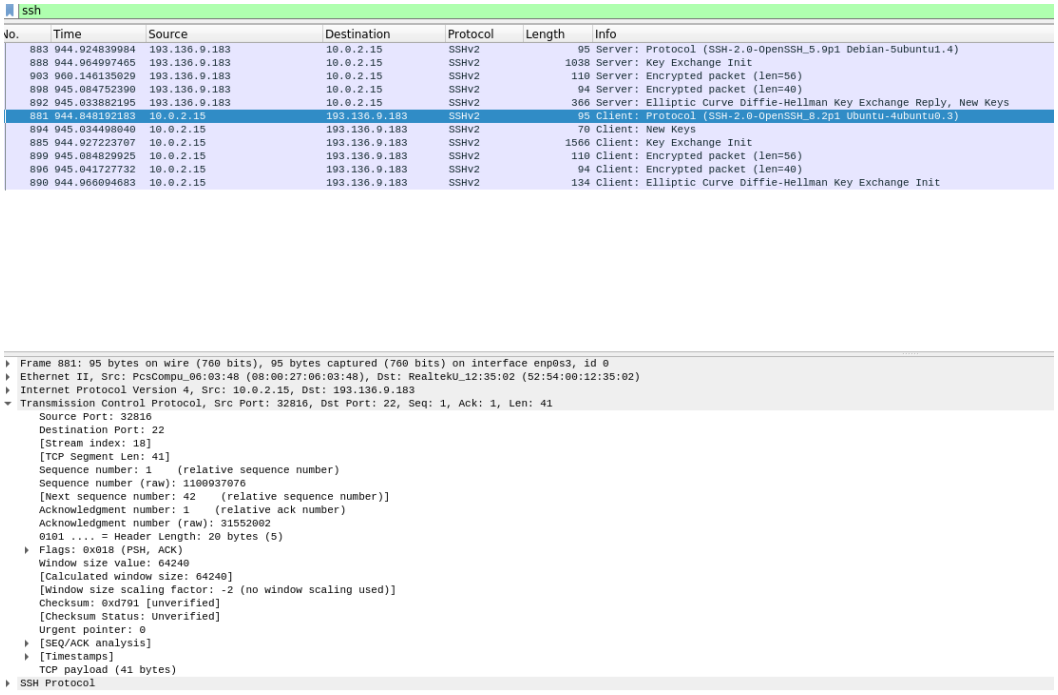

Figura 1.13: Captura wireshark: SSH

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 destre 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, HHTP e, por fim, TFTP que é a menos segura pelo que usa o protocolo UDP.