

Relatório Trabalho Prático nº1

João Pimentel (a80874) Rodolfo Silva (a81716) Pedro Gonçalves (a82313)

Fevereiro 2019

Universidade do Minho
Mestrado Integrado em Engenharia Informática
Comunicações por Computadores
Grupo 66

Conteúdo

1	Questões e Respostas	3
2	Conclusões	7
3	Anexos	8

1 Questões e Respostas

1. Inclua no relatório uma tabela em que identifique, para cada comando executado, qual o protocolo de aplicação, o protocolo de transporte, porta de atendimento e overhead de transporte, como ilustrado no exemplo seguinte:

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	-	-	-	-
tracert	-	UDP	33446	8
telnet	TELNET	TCP	23	20
ftp	FTP	TCP	21	20
Tftp	TFTP	UDP	69	8
browser/http	HTTP	TCP	80	20
nslookup	DNS	UDP	41	8
ssh	SSHv2	TCP	22	20
Outras:				

2. Uma representação num diagrama temporal das transferências da *file1* por FTP e TFTP respetivamente. Se for caso disso, identifique as fases de estabelecimento de conexão, transferência de dados e fim de conexão. Identifica também claramente os tipos de segmentos trocados e os números de sequência usados quer nos dados como nas confirmações.

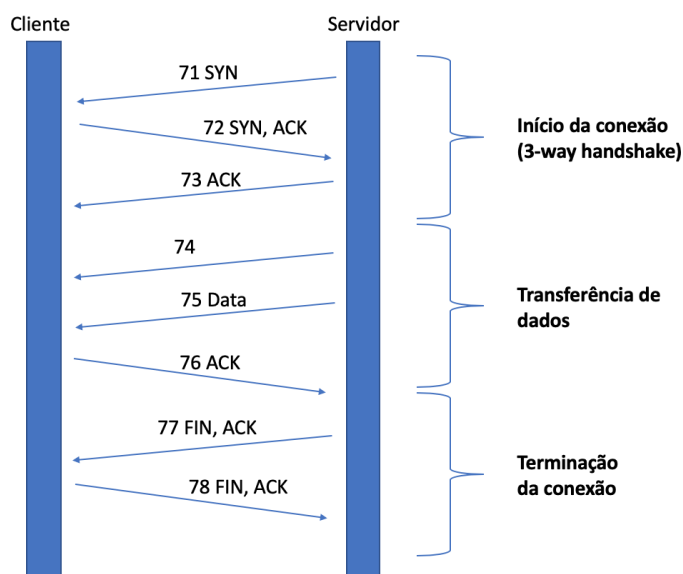


Figura 1 - Diagrama temporal da transferência por FTP.

70	129.918675	10.2.2.1	10.1.1.1	FTP	78 Request: RETR file1
71	129.918765	10.1.1.1	10.2.2.1	TCP	74 20 → 36799 [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=810992 TSecr=0 WS=16
72	129.918778	10.2.2.1	10.1.1.1	TCP	74 36799 → 20 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval=810992 TSecr=810992 WS=16
73	129.918785	10.1.1.1	10.2.2.1	TCP	66 20 → 36799 [ACK] Seq=1 Ack=1 Win=14608 Len=0 TSval=810992 TSecr=810992
74	129.918821	10.1.1.1	10.2.2.1	FTP	130 Response: 150 Opening BINARY mode data connection for file1 (193 bytes).
75	129.918838	10.1.1.1	10.2.2.1	FTP-D	259 FTP Data: 193 bytes (PORT) (RETR file1)
76	129.918851	10.2.2.1	10.1.1.1	TCP	66 36799 → 20 [ACK] Seq=1 Ack=194 Win=15552 Len=0 TSval=810992 TSecr=810992
77	129.918862	10.1.1.1	10.2.2.1	TCP	66 20 → 36799 [FIN, ACK] Seq=194 Ack=1 Win=14608 Len=0 TSval=810992 TSecr=810992
78	129.918968	10.2.2.1	10.1.1.1	TCP	66 36799 → 20 [FIN, ACK] Seq=1 Ack=195 Win=15552 Len=0 TSval=810992 TSecr=810992
79	129.918978	10.1.1.1	10.2.2.1	TCP	66 20 → 36799 [ACK] Seq=195 Ack=2 Win=14608 Len=0 TSval=810992 TSecr=810992
80	129.919039	10.1.1.1	10.2.2.1	FTP	90 Response: 226 Transfer complete.
81	129.919068	10.2.2.1	10.1.1.1	TCP	66 33756 → 21 [ACK] Seq=130 Ack=390 Win=14608 Len=0 TSval=810992 TSecr=810992

Figura 2 - Captura relativa à transferência por FTP.

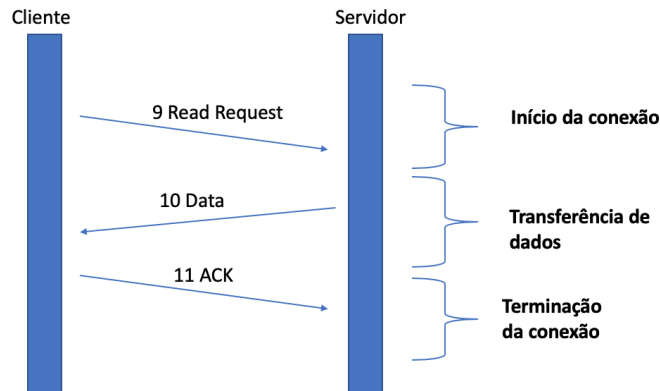


Figura 3 - Diagrama temporal da transferência por TFTP.

9	25.986935	10.2.2.1	10.1.1.1	TFTP	56 Read Request, File: file1, Transfer type: octet
10	25.987268	10.1.1.1	10.2.2.1	TFTP	239 Data Packet, Block: 1 (last)
11	25.987366	10.2.2.1	10.1.1.1	TFTP	46 Acknowledgement, Block: 1

Figura 4 - Captura relativa à transferência por TFTP.

3. Com base nas experiências realizadas, distinga e compare sucintamente as quatro aplicações de transferência de ficheiros que usou nos seguintes pontos:

- (i) uso da camada de transporte;
- (ii) eficiência na transferência;
- (iii) complexidade;
- (iv) segurança.

As quatro aplicações de transferência de ficheiros utilizados foram o FTP, SFTP, HTTP e TFTP.

O FTP é uma aplicação eficiente na transferência de ficheiros de dimensões consideráveis, usando dois canais de comunicação não encriptados, sendo, por isso, pouco segura. Um dos canais é utilizado para troca de mensagens entre o cliente e servidor e outro para troca de dados em si.

SFTP é uma aplicação que utiliza apenas um canal de comunicação encriptado por chaves de segurança, onde são feitas todas as trocas de comandos e dados entre o cliente e o servidor. Devido à existência de encriptação, é uma aplicação segura, no entanto, leva a um aumento bastante grande de complexidade e um maior uso da camada de transporte.

Já HTTP é um protocolo eficiente na transferência de ficheiros pequenos, tais como páginas web, possuindo, para essas transferências, um canal único sem grande segurança.

Por fim, TFTP é uma aplicação de baixa complexidade, não seguro, que é utilizada na transferência de ficheiros de baixa dimensão, com um uso de camada de transporte bastante baixo.

Posto isto, cada uma destas aplicações possui os seus prós e contras, sendo necessário estudar quais os aspetos que são necessário valorizar antes de escolher a aplicação a usar. Caso seja necessária uma garantia de segurança, o uso de SFTP é a escolha a fazer, mas se for preciso rapidez e pouco uso da camada de transporte seria ideal o uso de TFTP.

4. As características das ligações de rede têm uma enorme influência nos níveis de Transporte e de Aplicação. Discuta, relacionando a resposta com as experiências realizadas, as influências das situações de perda ou duplicação de pacotes IP no desempenho global de Aplicações fiáveis (se possível, relacionado com alguns dos mecanismos de transporte envolvidos).

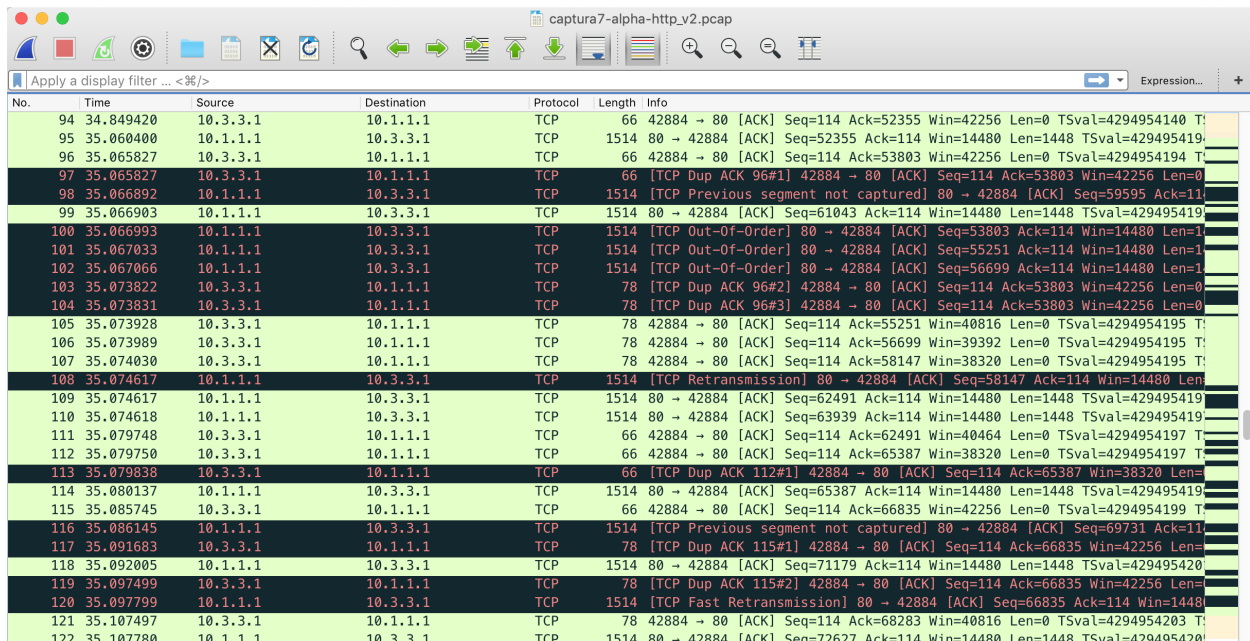
Tendo em conta a existência de uma perda de pacotes de 5% e de duplicação de 15% entre o *router*₄ com o *switch*_{Lan3}, seria expectável, em alguns casos, a ocorrência de complicações, devido à baixa fiabilidade da ligação.

Assim, como se observa na Figura 5, vários pacotes foram perdido/duplicados durante a obtenção do *file2* por *alpha*, sendo que a situação é visível pelas tramas com indicação de "not captured", "dup ack" ou "out-of-order".

Os motivos que podem ter levado a que tal acontecesse são um elevado congestionamento na rede, levando a que as filas de espera dos routers perdessem a capacidade de armazenar pacotes, excluindo-os.

Assim, não só os pacotes retransmitidos têm que percorrer o percurso todo duas vezes, porém, o emissor não se apercebe de que os pacotes foram perdidos até que, ou a *ack* não seja recebido na ordem expectável, ou falha a receção de *acks* durante tanto tempo que, assume-se que o pacote foi perdido, em vez de estar, meramente, atrasado.

Em suma, quanto menos fiável for o percurso a percorrer pelos pacotes, estando estes mais sujeitos a perdas/duplicações. Assim, o uso de *acks* permite a descoberta de falhas na troca de dados, permitindo a que, caso seja desejável, se possam pedir retransmissões.



No.	Time	Source	Destination	Protocol	Length	Info
94	34.849420	10.3.3.1	10.1.1.1	TCP	66	42884 → 80 [ACK] Seq=114 Ack=52355 Win=42256 Len=0 TSval=4294954140 T...
95	35.060400	10.1.1.1	10.3.3.1	TCP	1514	80 → 42884 [ACK] Seq=52355 Ack=114 Win=14480 Len=1448 TSval=429495419...
96	35.065827	10.3.3.1	10.1.1.1	TCP	66	42884 → 80 [ACK] Seq=114 Ack=53803 Win=42256 Len=0 TSval=4294954194 T...
97	35.065827	10.3.3.1	10.1.1.1	TCP	66	[TCP Dup ACK 96#1] 42884 → 80 [ACK] Seq=114 Ack=53803 Win=42256 Len=0
98	35.066892	10.1.1.1	10.3.3.1	TCP	1514	[TCP Previous segment not captured] 80 → 42884 [ACK] Seq=59595 Ack=11...
99	35.066903	10.1.1.1	10.3.3.1	TCP	1514	80 → 42884 [ACK] Seq=61043 Ack=114 Win=14480 Len=1448 TSval=429495419...
100	35.066993	10.1.1.1	10.3.3.1	TCP	1514	[TCP Out-Of-Order] 80 → 42884 [ACK] Seq=53803 Ack=114 Win=14480 Len=1...
101	35.067033	10.1.1.1	10.3.3.1	TCP	1514	[TCP Out-Of-Order] 80 → 42884 [ACK] Seq=55251 Ack=114 Win=14480 Len=1...
102	35.067066	10.1.1.1	10.3.3.1	TCP	1514	[TCP Out-Of-Order] 80 → 42884 [ACK] Seq=56699 Ack=114 Win=14480 Len=1...
103	35.073822	10.3.3.1	10.1.1.1	TCP	78	[TCP Dup ACK 96#2] 42884 → 80 [ACK] Seq=114 Ack=53803 Win=42256 Len=0
104	35.073831	10.3.3.1	10.1.1.1	TCP	78	[TCP Dup ACK 96#3] 42884 → 80 [ACK] Seq=114 Ack=53803 Win=42256 Len=0
105	35.073928	10.3.3.1	10.1.1.1	TCP	78	42884 → 80 [ACK] Seq=114 Ack=55251 Win=40816 Len=0 TSval=4294954195 T...
106	35.073989	10.3.3.1	10.1.1.1	TCP	78	42884 → 80 [ACK] Seq=114 Ack=56699 Win=39392 Len=0 TSval=4294954195 T...
107	35.074030	10.3.3.1	10.1.1.1	TCP	78	42884 → 80 [ACK] Seq=114 Ack=58147 Win=38320 Len=0 TSval=4294954195 T...
108	35.074617	10.1.1.1	10.3.3.1	TCP	1514	[TCP Retransmission] 80 → 42884 [ACK] Seq=58147 Ack=114 Win=14480 Len...
109	35.074617	10.1.1.1	10.3.3.1	TCP	1514	80 → 42884 [ACK] Seq=62491 Ack=114 Win=14480 Len=1448 TSval=429495419...
110	35.074618	10.1.1.1	10.3.3.1	TCP	1514	80 → 42884 [ACK] Seq=63939 Ack=114 Win=14480 Len=1448 TSval=429495419...
111	35.079748	10.3.3.1	10.1.1.1	TCP	66	42884 → 80 [ACK] Seq=114 Ack=62491 Win=40464 Len=0 TSval=4294954197 T...
112	35.079750	10.3.3.1	10.1.1.1	TCP	66	42884 → 80 [ACK] Seq=114 Ack=65387 Win=38320 Len=0 TSval=4294954197 T...
113	35.079838	10.3.3.1	10.1.1.1	TCP	66	[TCP Dup ACK 112#1] 42884 → 80 [ACK] Seq=114 Ack=65387 Win=38320 Len=0
114	35.080137	10.1.1.1	10.3.3.1	TCP	1514	80 → 42884 [ACK] Seq=65387 Ack=114 Win=14480 Len=1448 TSval=429495419...
115	35.085745	10.3.3.1	10.1.1.1	TCP	66	42884 → 80 [ACK] Seq=114 Ack=66835 Win=42256 Len=0 TSval=4294954199 T...
116	35.086145	10.1.1.1	10.3.3.1	TCP	1514	[TCP Previous segment not captured] 80 → 42884 [ACK] Seq=69731 Ack=11...
117	35.091683	10.3.3.1	10.1.1.1	TCP	78	[TCP Dup ACK 115#1] 42884 → 80 [ACK] Seq=114 Ack=66835 Win=42256 Len=0
118	35.092005	10.1.1.1	10.3.3.1	TCP	1514	80 → 42884 [ACK] Seq=71179 Ack=114 Win=14480 Len=1448 TSval=429495420...
119	35.097499	10.3.3.1	10.1.1.1	TCP	78	[TCP Dup ACK 115#2] 42884 → 80 [ACK] Seq=114 Ack=66835 Win=42256 Len=0
120	35.097799	10.1.1.1	10.3.3.1	TCP	1514	[TCP Fast Retransmission] 80 → 42884 [ACK] Seq=66835 Ack=114 Win=1448...
121	35.107497	10.3.3.1	10.1.1.1	TCP	78	42884 → 80 [ACK] Seq=114 Ack=68283 Win=40816 Len=0 TSval=4294954203 T...
122	35.107780	10.1.1.1	10.3.3.1	TCP	1514	80 → 42884 [ACK] Seq=72627 Ack=114 Win=14480 Len=1448 TSval=429495420...

Figura 5 - Captura com perda e duplicação de pacotes durante o percurso.

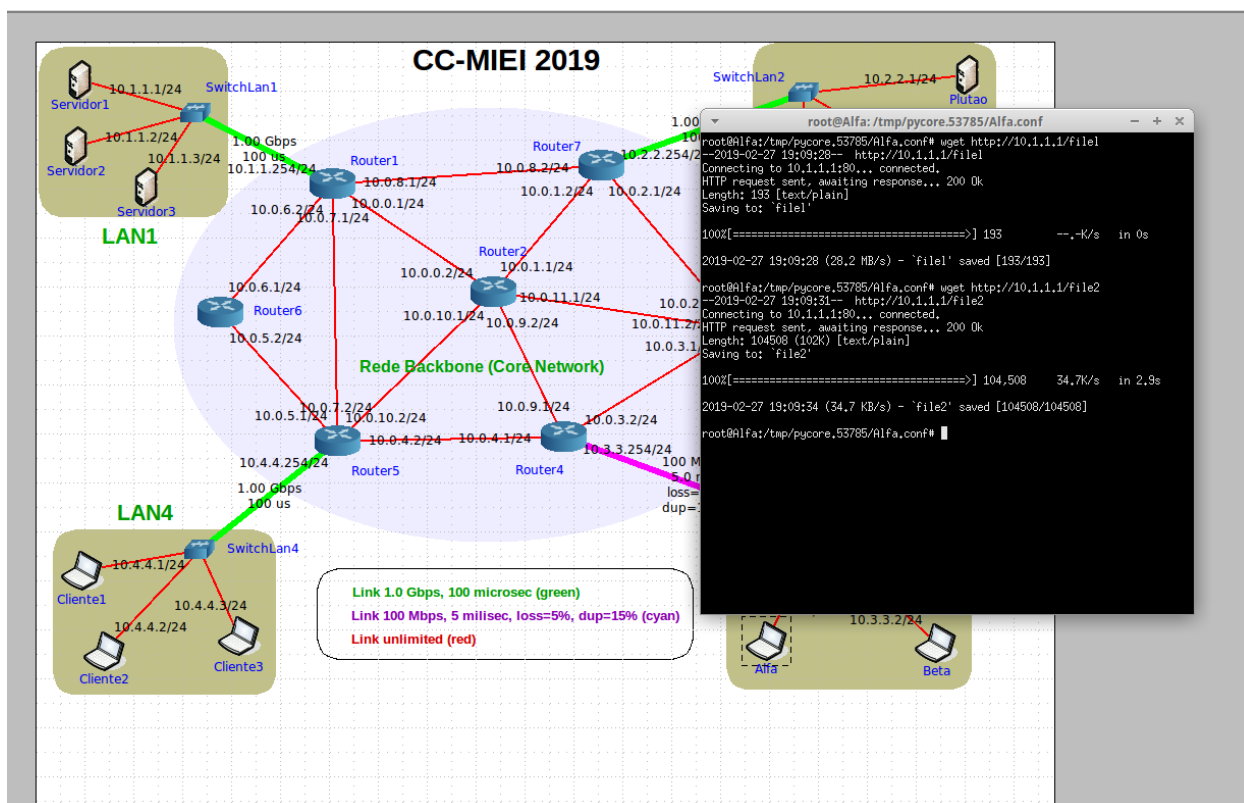


Figura 6 - Captura relativa à transferência do *file2* por HTTP.

2 Conclusões

O desenvolvimento deste projeto permitiu um aumento no conhecimento relativamente ao protocolo da camada de transporte e aplicação.

A utilização de *acknowledgements* permite garantir a receção de pacotes, bem como saber quando estes não foram recebidos, dando a possibilidade de pedir um reenvio do mesmo. Além disso, permite a moderação de uma ligação entre cliente servidor, tornando possível a troca de dados.

Consoante a finalidade de uma dada aplicação, é necessário pensar no protocolo aplicacional a utilizar, sendo que estas diferenciam em termos de segurança, complexidade e outros fatores. Além disso, em termos da camada de transporte, o uso de um protocolo UDP permite ligações rápidas, sem qualquer controlo, ao contrário de TCP, que permite o controlo de erros e com segurança.

Em suma, este projeto permitiu solidificar o conhecimento adquirido nas aulas teóricas, tendo sido bastante proveitoso para a possível realização de qualquer trabalho futuro relativo aos protocolos aplicacionais e de transporte.

3 Anexos

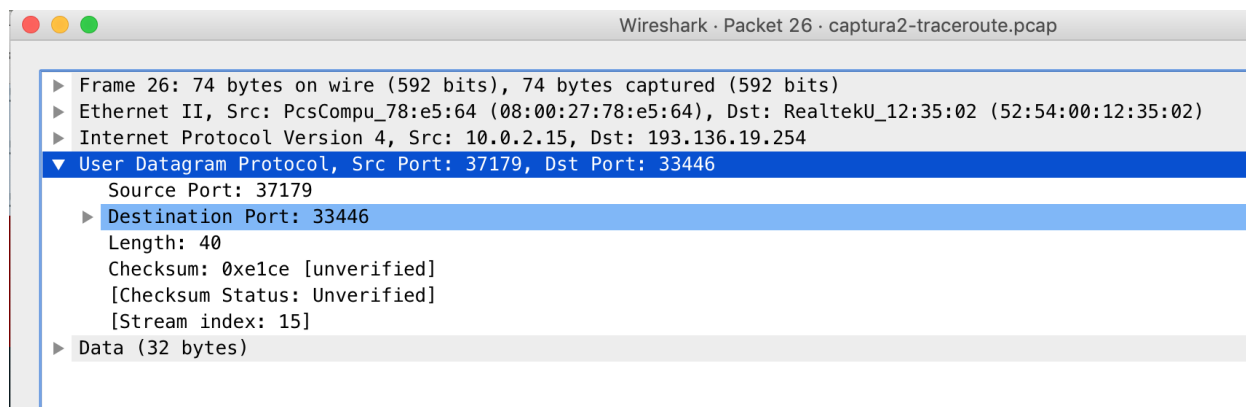


Figura 7 - Trama relativa ao uso de um traceroute.

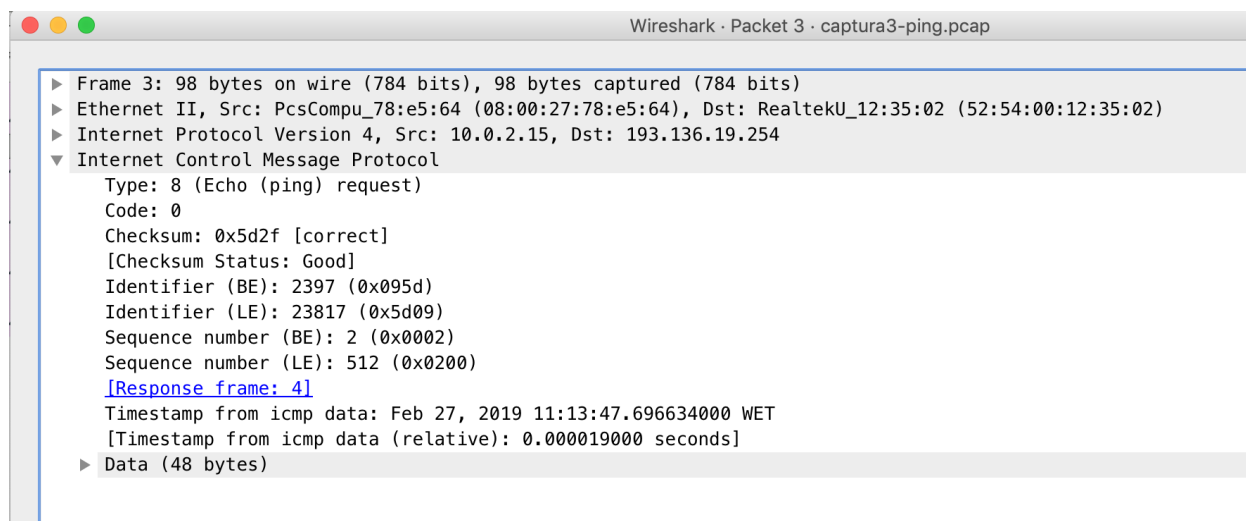


Figura 8 - Trama relativa ao uso de um ping.

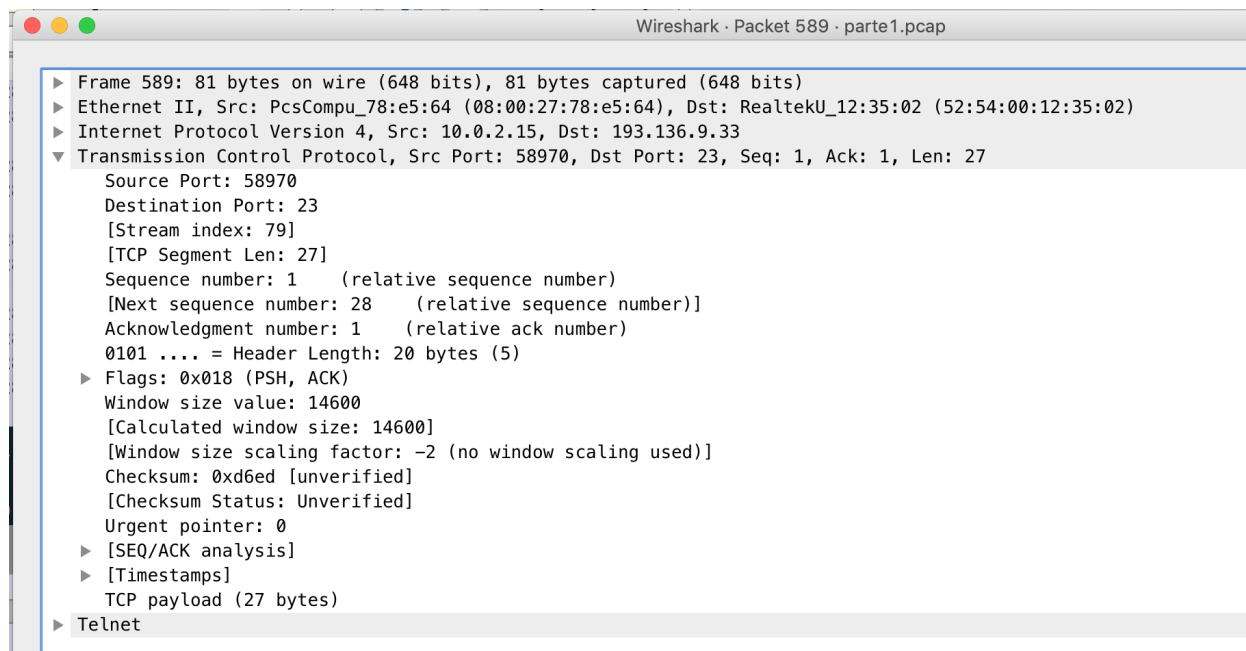


Figura 9 - Trama relativa ao uso de um telnet.

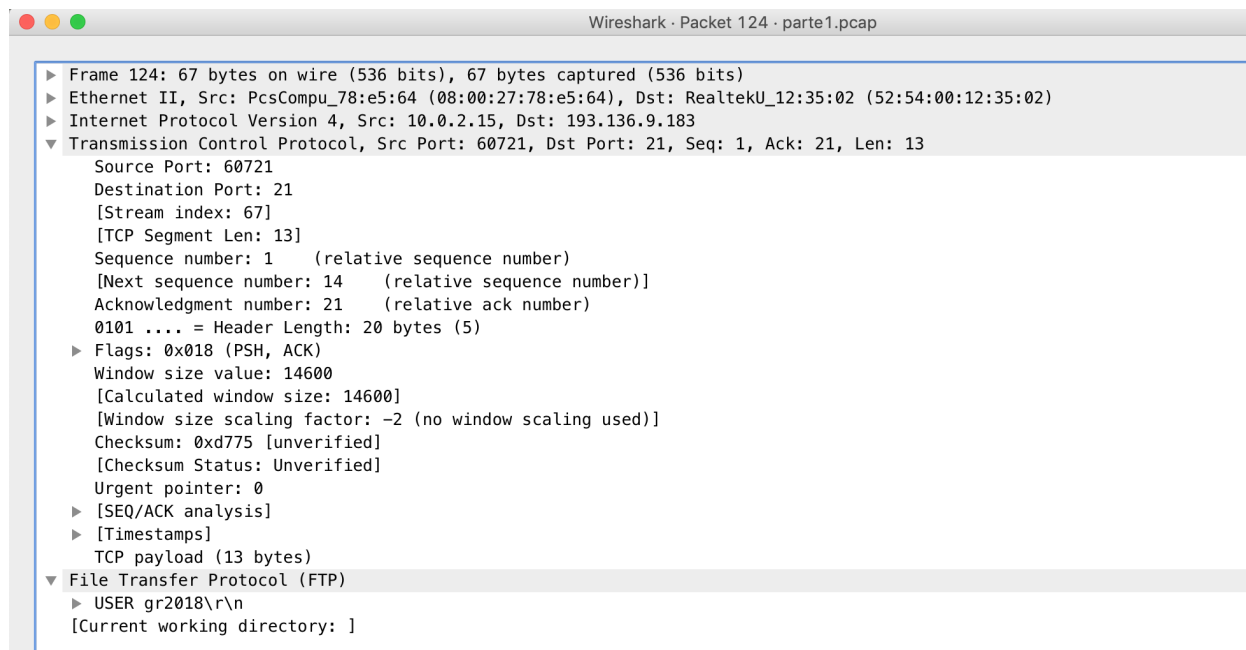


Figura 10 - Trama relativa ao uso de um ftp.

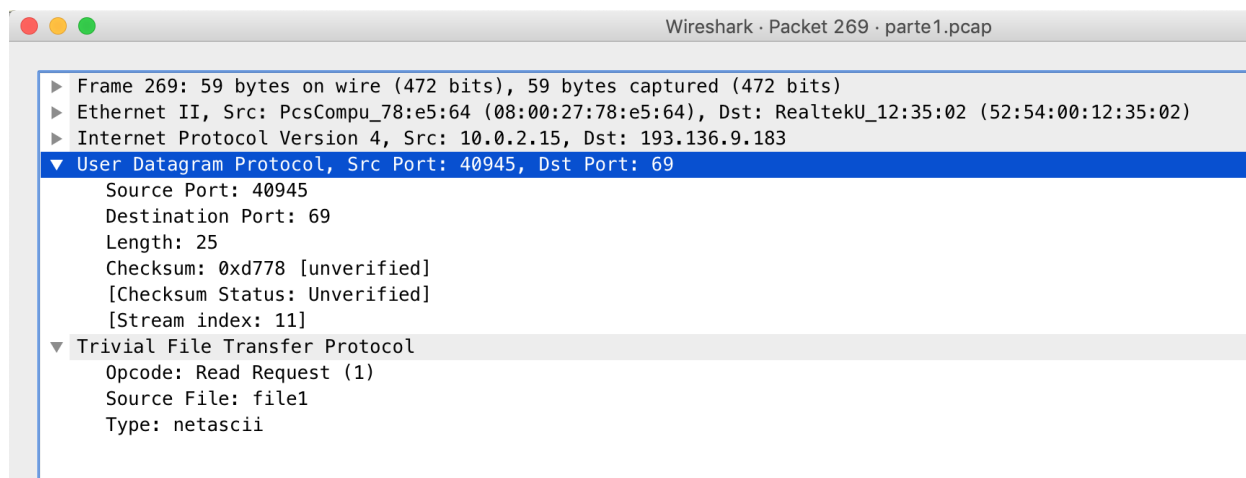


Figura 11 - Trama relativa ao uso de um tftp.

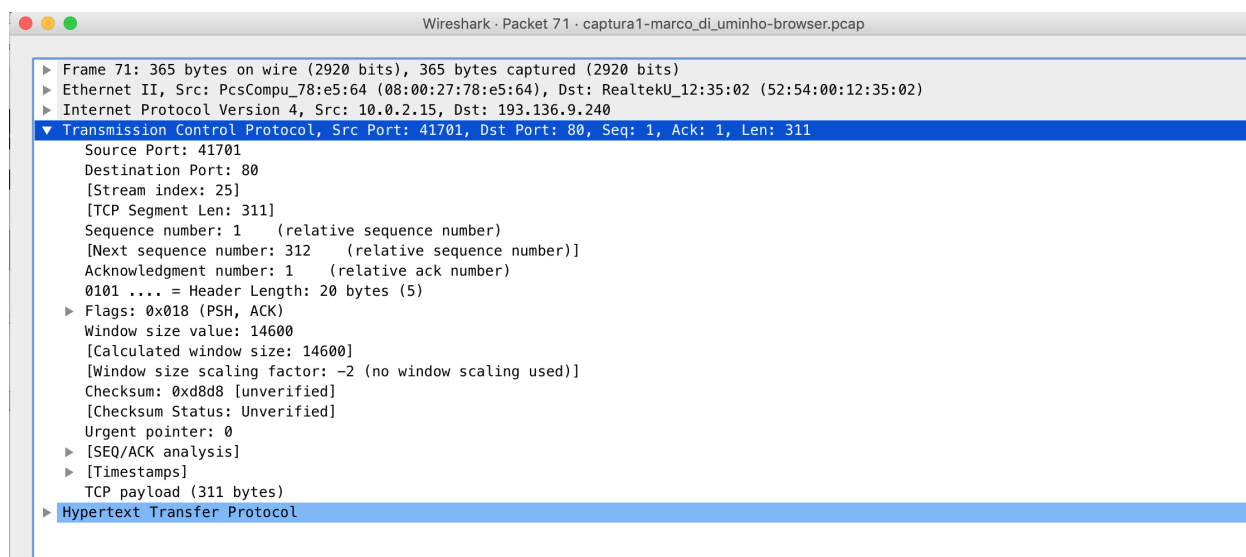


Figura 12 - Trama relativa ao uso de um http.

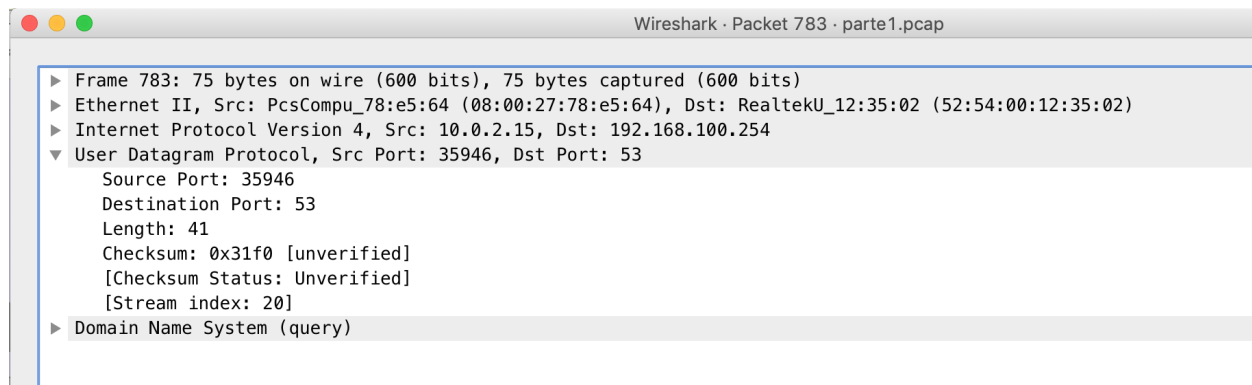


Figura 14 - Trama relativa ao uso de um nslookup.

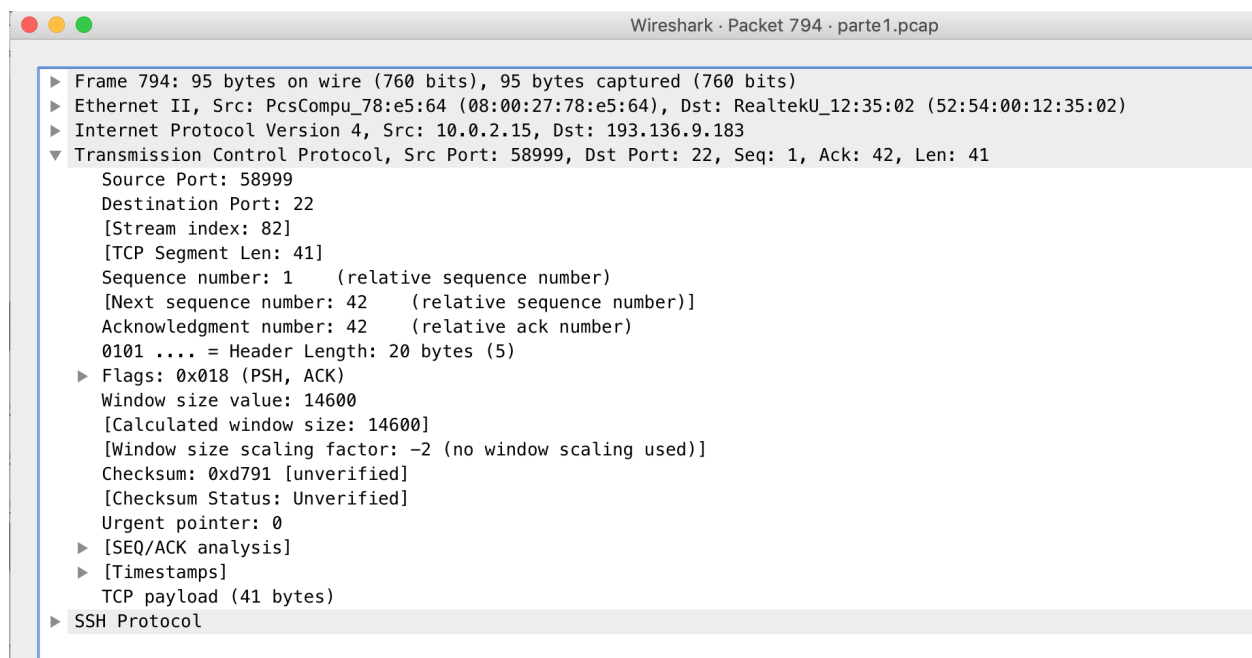


Figura 15 - Trama relativa ao uso de um ssh.