

TP1: Protocolos da Camada de Transporte

Henrique José Carvalho Faria

José André Martins Pereira

University of Minho, Department of Informatics,

4710-057 Braga, Portugal e-mail: {a82200,a82880}@alunos.uminho.pt

Exercício 1:

Capturando o tráfego em determinados instantes que considere adequados, observe atentamente como as várias aplicações utilizam os serviços da camada inferior:

Resposta:

Comando usado (Aplicação)	Protocolo de Aplicação (se ap)	Protocolo de transporte (se aplicável)	Porta de atendimento (se aplicável)	Overhead de transporte em bytes (se aplicável)
Ping	Não tem aplicação	Não tem protocolo de transporte porque corre diretamente em ICMP	Não tem porta	20 bytes
tracert	Não tem aplicação	UDP	33434	8 bytes
telnet	TELNET	TCP	23	20 bytes
ftp	FTP	TCP	21 (20 para data)	20 bytes
tftp	TFTP	UDP	69	8 bytes
Browser/http	HTTP	TCP	80	32 bytes
nslookup	DNS	UDP	53	8 bytes
ssh	SSHv2	TCP	22	20 bytes
Outras:				
sftp	TLSv1.2	TCP	5223	32 bytes

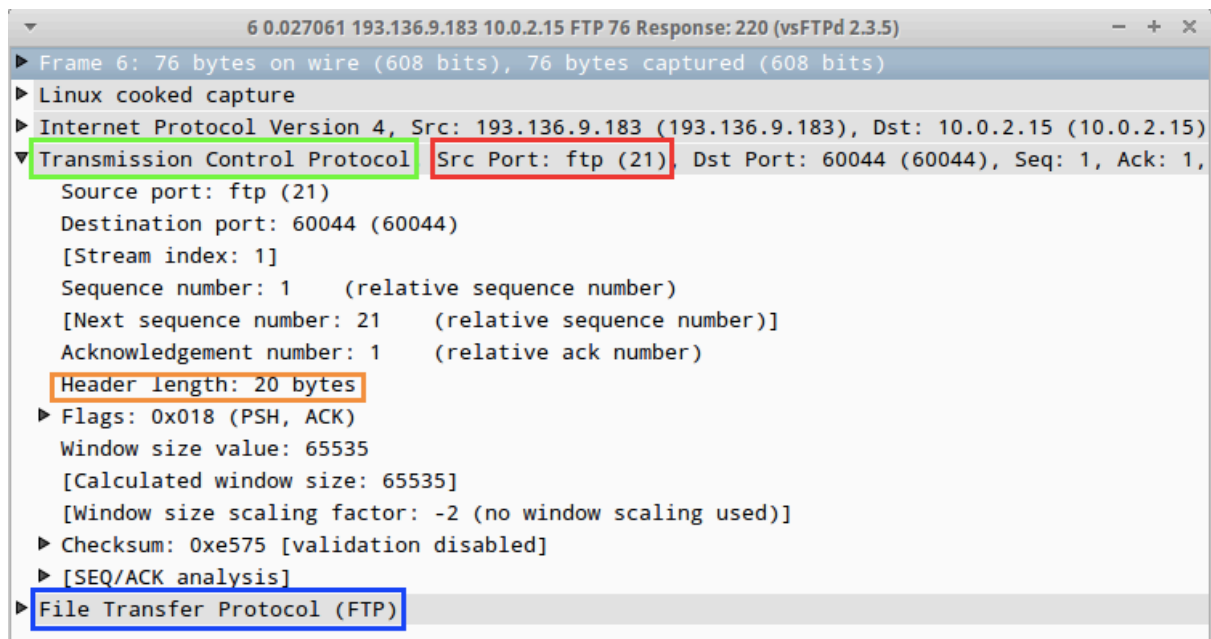


Figura 1 – Informação do FTP.

A Figura 1, representa a captura feita com o Wireshark, durante a execução do comando **ftp gr2018.ddns.net**, onde se consegue verificar o protocolo da aplicação **FTP** (retângulo azul), o protocolo de transporte **TCP** (*Transmission Control Protocol*) (retângulo verde), a porta de atendimento, **21** (retângulo vermelho) e por fim, não menos importante, o overhead de transporte, **20 bytes** (retângulo laranja).

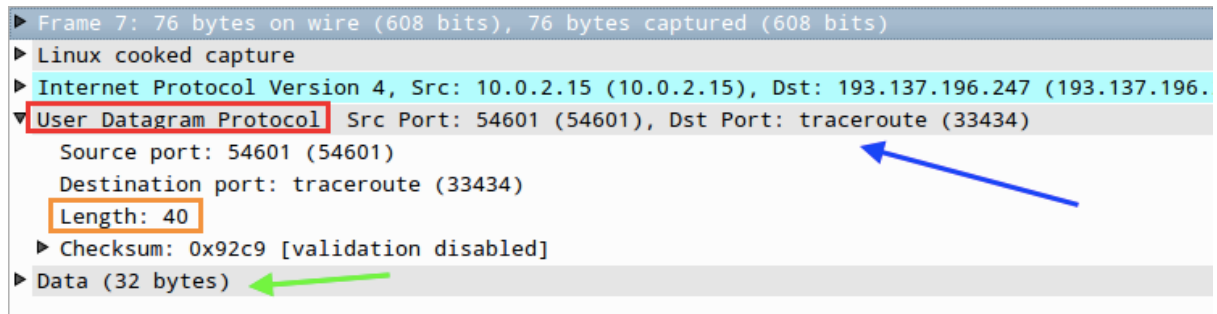


Figura 2 – Informação do traceroute.

A Figura 2, representa a captura durante a execução do comando **traceroute www.fcn.pt**, o qual não tem protocolo de aplicação, o protocolo de transporte é UDP, pelo que se pode observar no retângulo vermelho, a porta de atendimento é a 33434, como se pode verificar na seta azul, por fim, o campo **Length= 40 bytes** (retângulo laranja), ou seja, o **overhead** de transporte é 40 – **payload**, ou seja, $40 - 32 = 8$ bytes (que normalmente é sempre o mesmo quando o protocolo de transporte é UDP), pelo que tem-se $8/40 = 0.20 = 20\%$ de overhead.

Os restantes comandos, são semelhantes, pelo que não se achou necessário a sua apresentação e justificação.

Exercício 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. (Nota: a transferência por FTP envolve mais que uma conexão FTP, nomeadamente uma de controlo [ftp] e outra de dados [ftp-data]. Faça o diagrama apenas para a conexão de transferência de dados do ficheiro mais pequeno).

Resposta:

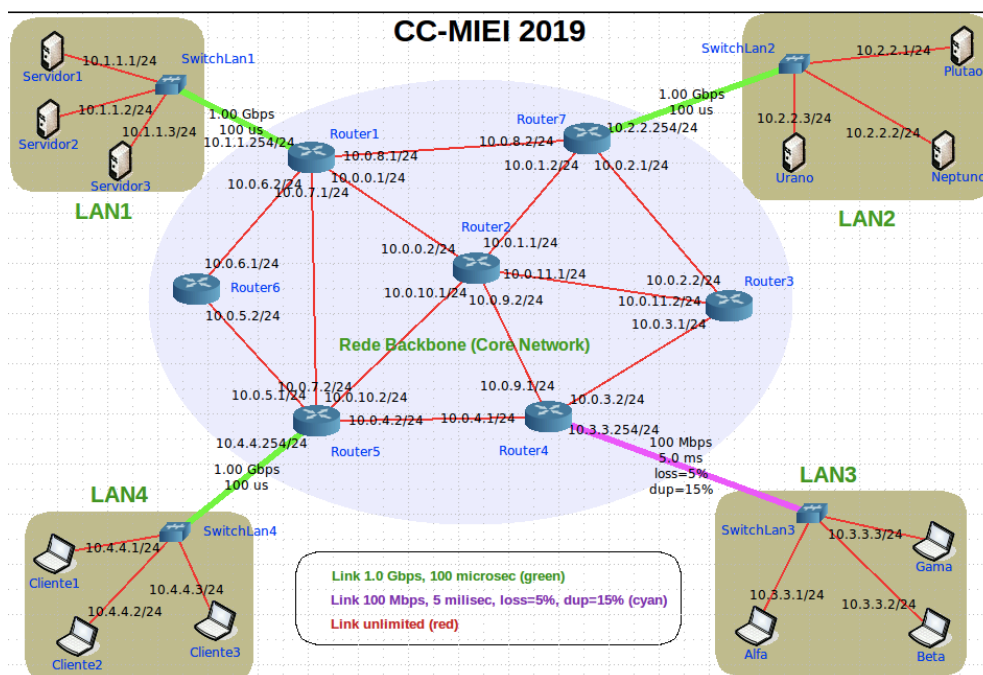


Figura 3 – Topologia utilizada.

Inicialmente verificou-se a conectividade entre o Cliente1 e Servidor1, capturando com o Wireshark no Router1, durante a execução do comando ping entre o Cliente1(10.4.4.1) e Servidor1(10.1.1.1), e obteve-se a resposta que se pode observar na Figura 4. O mesmo se verifica entre o host Alfa(10.3.3.1) eo Servidor1, como se pode ver na Figura 5.

26	108.996880	00:00:00_aa:00:16	00:00:00_aa:00:12	ARP	42	10.1.1.1	is at 00:00:00:aa:00:
27	108.996892	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request	id=0x004c
28	108.997083	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply	id=0x004c
29	109.997716	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request	id=0x004c
30	109.997894	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply	id=0x004c

Figura 4 – Captura no Router1 durante a execução do comando ping 10.1.1.1.

81	141.564052	10.3.3.1	10.1.1.1	ICMP	98	Echo (ping) request	id=0x004c
82	141.564053	10.3.3.1	10.1.1.1	ICMP	98	Echo (ping) request	id=0x004c
83	141.564637	10.1.1.1	10.3.3.1	ICMP	98	Echo (ping) reply	id=0x004c
84	141.564638	10.1.1.1	10.3.3.1	ICMP	98	Echo (ping) reply	id=0x004c

Figura 5 – Captura no Router1 durante a execução do comando ping 10.1.1.1.

7.26.379460	10.4.4.1	10.1.1.1	TFTP	56 Read Request, File: file1, Transfer type: octet
8.26.380565	10.1.1.1	10.4.4.1	TFTP	239 Data Packet, Block: 1 (last)
9.26.380622	10.4.4.1	10.1.1.1	TFTP	46 Acknowledgement, Block: 1

Figura 6 – Captura no wireshark do comando TFTP.

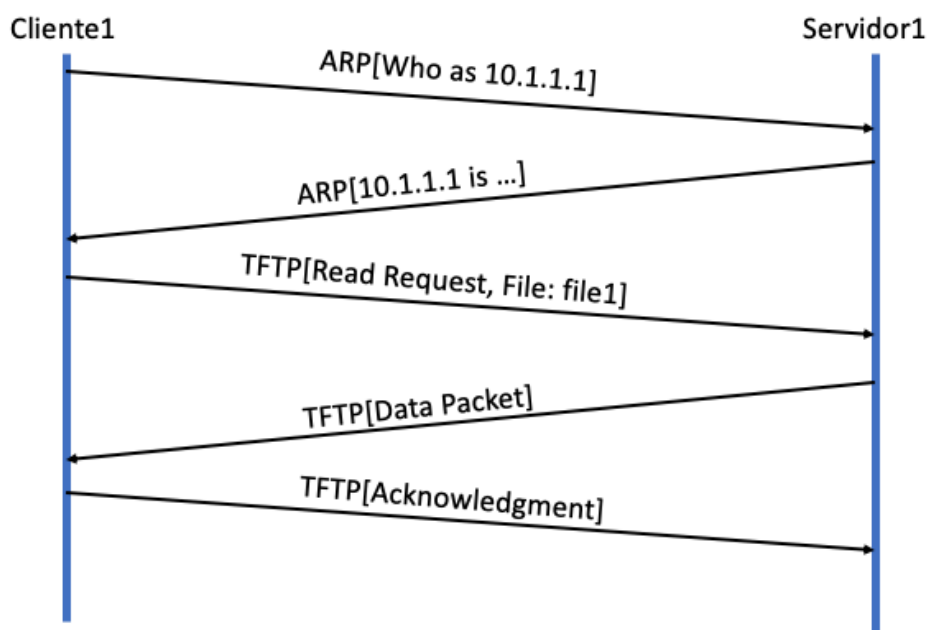


Figura 7 – Diagrama temporal de transferência do TFTP.

Observando o diagrama temporal de transferências do comando TFTP, verifica-se que não é complexo, começando pelo pedido de transferência, através do pacote TFTP Request do file1, ao qual o Servidor1 “responde” com um pacote TFTP Data Packet, com o file1, e o host Cliente1, envia um ACK a informar o servidor que recebeu o ficheiro file1.

```

root@Servidor1: /tmp/pycore.41766/Servidor1.conf
root@Servidor1:/tmp/pycore.41766/Servidor1.conf# chmod -R 777 /srv/ftp/
root@Servidor1:/tmp/pycore.41766/Servidor1.conf# touch atftpd.log
root@Servidor1:/tmp/pycore.41766/Servidor1.conf# atftpd --verbose=3 --user root.
ftp --logfile atftpd.log --bind-address 10.1.1.1 --daemon --no-fork /srv/ftp/

```

Figura 8 - Inicialização do servidor 1.

```
root@Cliente1:/tmp/pycore.41766/Cliente1.conf
root@Cliente1:/tmp/pycore.41766/Cliente1.conf# atftp 10.1.1.1
tftp> status
Connected: 10.1.1.1 port 69
Mode:      octet
Verbose:    off
Trace:      off
Options
  tsize:    disabled
  blksize:  disabled
  timeout:  disabled
  multicast: disabled
mtftp variables
  client-port: 76
  mcast-ip:    0.0.0.0
  listen-delay: 2
  timeout-delay: 2
Last command: quit
tftp> get file1
Overwrite local file [y/n]? y
tftp> quit
root@Cliente1:/tmp/pycore.41766/Cliente1.conf#
```

Figura 9 – Cliente 1 faz transferência do file1.

48	37.171820	10.4.4.1	10.1.1.1	FTP	78 Request: RETR file1
49	37.172693	10.1.1.1	10.4.4.1	TCP	74 ftp-data > 43820 [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=1011614 TSecr=0 WS=16
50	37.172711	10.4.4.1	10.1.1.1	TCP	74 43820 > ftp-data [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval=1011614 TSecr=1011614
51	37.173096	10.1.1.1	10.4.4.1	TCP	66 ftp-data > 43820 [ACK] Seq=1 Ack=1 Win=14608 Len=0 TSval=1011614 TSecr=1011614
52	37.173107	10.1.1.1	10.4.4.1	FTP	130 Response: 150 Opening BINARY mode data connection for file1 (193 bytes).
53	37.173135	10.1.1.1	10.4.4.1	FTP-DAT	259 FTP Data: 193 bytes
54	37.173142	10.4.4.1	10.1.1.1	TCP	66 43820 > ftp-data [ACK] Seq=1 Ack=194 Win=15552 Len=0 TSval=1011614 TSecr=1011614
55	37.173165	10.1.1.1	10.4.4.1	TCP	66 ftp-data > 43820 [FIN, ACK] Seq=194 Ack=1 Win=14608 Len=0 TSval=1011614 TSecr=1011614
56	37.174436	10.4.4.1	10.1.1.1	TCP	66 43820 > ftp-data [FIN, ACK] Seq=1 Ack=195 Win=15552 Len=0 TSval=1011615 TSecr=1011614
57	37.175261	10.1.1.1	10.4.4.1	TCP	66 ftp-data > 43820 [ACK] Seq=195 Ack=2 Win=14608 Len=0 TSval=1011615 TSecr=1011615
58	37.175261	10.1.1.1	10.4.4.1	FTP	90 Response: 226 Transfer complete.

Figura 10 – Captura wireshark do FTP.

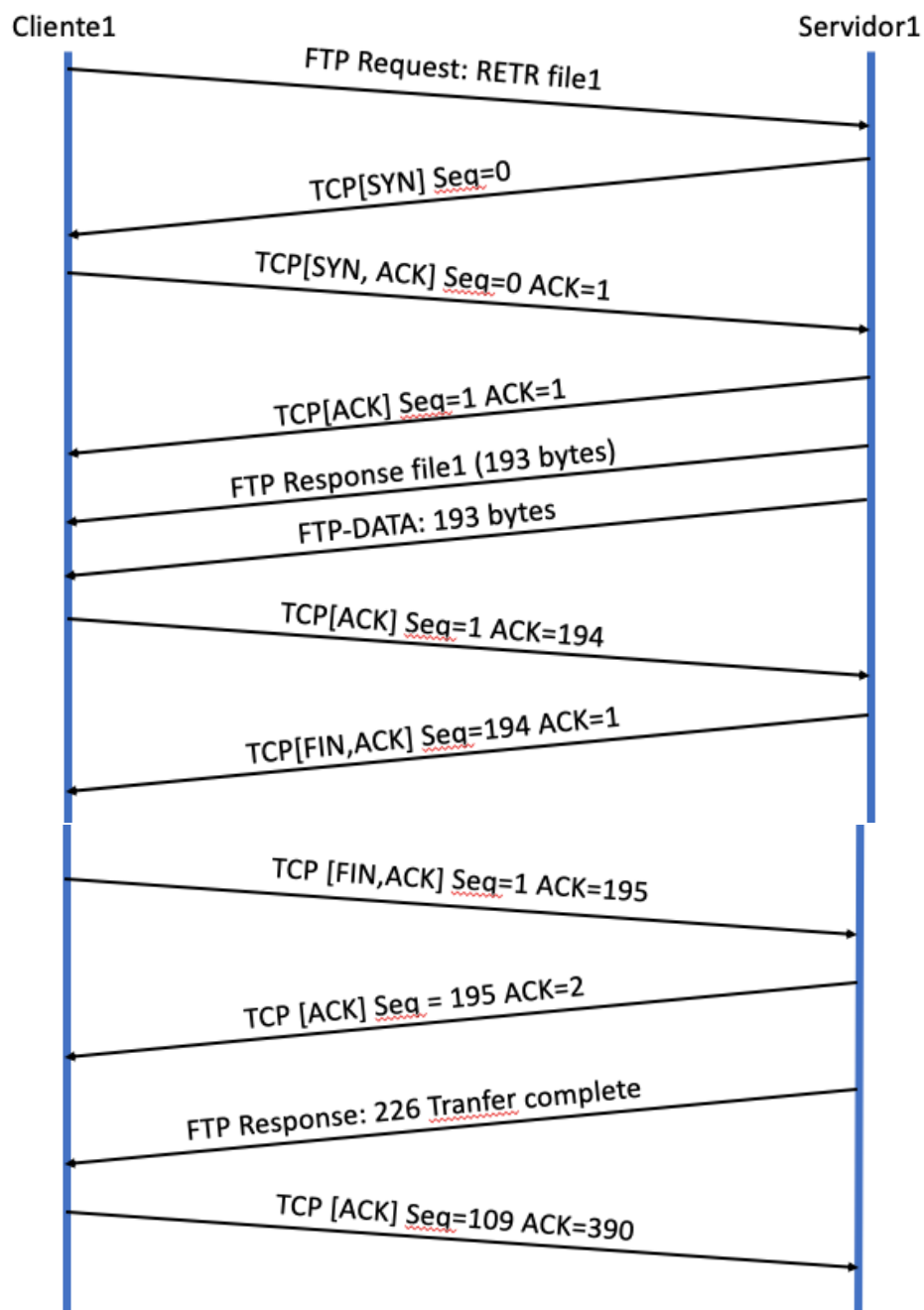


Figura 11 – Diagrama temporal de transferências do FTP.

Observando o diagrama temporal de transferências do comando FTP, verifica-se que inicialmente é enviado um pacote FTP Request, ao qual o servidor denominado Servidor1 “responde” com um pacote TCP [SYN] para se estabelecer a sincronização, ao qual o host Cliente1 responde com TCP [SYN, ACK], e de seguida, o servidor, envia TCP [ACK] com Seq=1, Ack=1, com a finalidade de informar o host que vai iniciar o envio do ficheiro.

O servidor, começa a enviar o ficheiro com o FTP Response file1, e envia o pacote com o ficheiro, FTP-DATA.

O Cliente1, informa que recebeu o file1, através do TCP [ACK] Seq=1, Ack=194, pelo que o servidor responde com TCP [FIN,ACK] Seq=194, Ack=1, a informar a finalização, ao qual o cliente, “responde” com Ack=195. De seguida o servidor, envia o pacote TCP [ACK] Seq=195, Ack=2, e para

terminar o servidor informa que a transferência está concluída com FTP Response, e o cliente envia um TCP [ACK] Seq=109, ACK=390, a informa que recebeu.

Exercício 3:

Com base nas experiências realizadas, distinga e compare sucintamente as quatro aplicações de transferência de arquivos que usou nos seguintes pontos (i) uso da camada de transporte; (ii) eficiência na transferência; (iii) complexidade; (iv) segurança;

Resposta:

No.	Time	Source	Destination	Protocol	Length	Info
18	4.761807	10.4.4.1	10.1.1.1	HTTP	179	GET /file2 HTTP/1.1
19	4.763554	10.1.1.1	10.4.4.1	TCP	66	http > 46897 [ACK] Seq=1 Ack=114 Win=14480 Len=0 TSval=1370764 TSecr=1370764
20	4.763946	10.1.1.1	10.4.4.1	TCP	292	[TCP segment of a reassembled PDU]
21	4.763966	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=227 Win=15680 Len=0 TSval=1370764 TSecr=1370764
22	4.976733	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
23	4.976754	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=1675 Win=18576 Len=0 TSval=1370818 TSecr=1370818
24	5.016557	00:00:00_aa:00:13	00:00:00_aa:00:1d	ARP	42	Who has 10.4.4.1? Tell 10.4.4.254
25	5.016582	00:00:00_aa:00:1d	00:00:00_aa:00:13	ARP	42	10.4.4.1 is at 00:00:00_aa:00:1d
26	5.185647	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
27	5.185746	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=3123 Win=21472 Len=0 TSval=1370870 TSecr=1370870
28	5.392835	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
29	5.392852	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=4571 Win=24368 Len=0 TSval=1370922 TSecr=1370922
30	5.393980	10.1.1.1	10.4.4.1	TCP	1514	[TCP Previous segment lost] [TCP segment of a reassembled PDU]
31	5.393989	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
32	5.393995	10.4.4.1	10.1.1.1	TCP	78	[TCP Dup ACK 29#1] 46897 > http [ACK] Seq=114 Ack=4571 Win=24368 Len=0 TSval=1370922 TSecr=1370922 SLE=20499 SRE=23395
33	5.393997	10.4.4.1	10.1.1.1	TCP	78	[TCP Dup ACK 29#2] 46897 > http [ACK] Seq=114 Ack=4571 Win=24368 Len=0 TSval=1370922 TSecr=1370922 SLE=20499 SRE=23395
34	5.394338	10.1.1.1	10.4.4.1	TCP	1514	[TCP Fast Retransmission] [TCP segment of a reassembled PDU]
35	5.394344	10.1.1.1	10.4.4.1	TCP	1514	[TCP Out-Of-Order] [TCP segment of a reassembled PDU]
36	5.394350	10.4.4.1	10.1.1.1	TCP	78	46897 > http [ACK] Seq=114 Ack=6019 Win=27264 Len=0 TSval=1370922 TSecr=1370922 SLE=20499 SRE=23395
37	5.394352	10.4.4.1	10.1.1.1	TCP	78	46897 > http [ACK] Seq=114 Ack=7467 Win=30160 Len=0 TSval=1370922 TSecr=1370922 SLE=20499 SRE=23395
38	5.394370	10.1.1.1	10.4.4.1	TCP	1514	[TCP Out-Of-Order] [TCP segment of a reassembled PDU]
39	5.394373	10.4.4.1	10.1.1.1	TCP	78	46897 > http [ACK] Seq=114 Ack=8915 Win=33056 Len=0 TSval=1370922 TSecr=1370922 SLE=20499 SRE=23395
40	5.394672	10.1.1.1	10.4.4.1	TCP	1514	[TCP Out-Of-Order] [TCP segment of a reassembled PDU]
41	5.394678	10.1.1.1	10.4.4.1	TCP	1514	[TCP Out-Of-Order] [TCP segment of a reassembled PDU]
42	5.394683	10.4.4.1	10.1.1.1	TCP	78	46897 > http [ACK] Seq=114 Ack=10363 Win=35952 Len=0 TSval=1370922 TSecr=1370922 SLE=20499 SRE=23395
43	5.394684	10.4.4.1	10.1.1.1	TCP	78	46897 > http [ACK] Seq=114 Ack=11811 Win=38848 Len=0 TSval=1370922 TSecr=1370922 SLE=20499 SRE=23395
44	5.394701	10.1.1.1	10.4.4.1	TCP	1514	[TCP Out-Of-Order] [TCP segment of a reassembled PDU]
45	5.394705	10.4.4.1	10.1.1.1	TCP	78	46897 > http [ACK] Seq=114 Ack=13259 Win=38320 Len=0 TSval=1370922 TSecr=1370922 SLE=20499 SRE=23395

No.	Time	Source	Destination	Protocol	Length	Info
46	5.394997	10.1.1.1	10.4.4.1	TCP	1514	[TCP Out-Of-Order] [TCP segment of a reassembled PDU]
47	5.395003	10.1.1.1	10.4.4.1	TCP	1514	[TCP Out-Of-Order] [TCP segment of a reassembled PDU]
48	5.395008	10.4.4.1	10.1.1.1	TCP	78	46897 > http [ACK] Seq=114 Ack=14707 Win=40464 Len=0 TSval=1370922 TSecr=1370922 SLE=20499 SRE=23395
49	5.395010	10.4.4.1	10.1.1.1	TCP	78	46897 > http [ACK] Seq=114 Ack=16155 Win=39392 Len=0 TSval=1370922 TSecr=1370922 SLE=20499 SRE=23395
50	5.395026	10.1.1.1	10.4.4.1	TCP	1514	[TCP Out-Of-Order] [TCP segment of a reassembled PDU]
51	5.395030	10.4.4.1	10.1.1.1	TCP	78	46897 > http [ACK] Seq=114 Ack=17603 Win=38320 Len=0 TSval=1370922 TSecr=1370922 SLE=20499 SRE=23395
52	5.395365	10.1.1.1	10.4.4.1	TCP	1514	[TCP Out-Of-Order] [TCP segment of a reassembled PDU]
53	5.395371	10.1.1.1	10.4.4.1	TCP	1514	[TCP Out-Of-Order] [TCP segment of a reassembled PDU]
54	5.395376	10.4.4.1	10.1.1.1	TCP	78	46897 > http [ACK] Seq=114 Ack=19051 Win=40464 Len=0 TSval=1370922 TSecr=1370922 SLE=20499 SRE=23395
55	5.395378	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=23395 Win=39392 Len=0 TSval=1370922 TSecr=1370922
56	5.395394	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
57	5.395398	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=24843 Win=38320 Len=0 TSval=1370922 TSecr=1370922
58	5.395766	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
59	5.395774	10.1.1.1	10.4.4.1	TCP	1514	[TCP Previous segment lost] [TCP segment of a reassembled PDU]
60	5.395779	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=26291 Win=42256 Len=0 TSval=1370922 TSecr=1370922
61	5.395782	10.4.4.1	10.1.1.1	TCP	78	[TCP Dup ACK 60#1] 46897 > http [ACK] Seq=114 Ack=26291 Win=42256 Len=0 TSval=1370922 TSecr=1370922 SLE=20499 SRE=23395
62	5.396082	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
63	5.396087	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
64	5.396092	10.4.4.1	10.1.1.1	TCP	78	[TCP Dup ACK 60#2] 46897 > http [ACK] Seq=114 Ack=26291 Win=42256 Len=0 TSval=1370922 TSecr=1370922 SLE=20499 SRE=23395
65	5.396093	10.4.4.1	10.1.1.1	TCP	78	[TCP Dup ACK 60#3] 46897 > http [ACK] Seq=114 Ack=26291 Win=42256 Len=0 TSval=1370922 TSecr=1370922 SLE=20499 SRE=23395
66	5.396459	10.1.1.1	10.4.4.1	TCP	1514	[TCP Fast Retransmission] [TCP segment of a reassembled PDU]
67	5.396468	10.1.1.1	10.4.4.1	TCP	1514	[TCP Out-Of-Order] [TCP segment of a reassembled PDU]
68	5.396476	10.4.4.1	10.1.1.1	TCP	78	46897 > http [ACK] Seq=114 Ack=27739 Win=40816 Len=0 TSval=1370923 TSecr=1370922 SLE=29187 SRE=33531
69	5.396479	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=33531 Win=38320 Len=0 TSval=1370923 TSecr=1370922
70	5.396903	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
71	5.396916	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=34979 Win=42256 Len=0 TSval=1370923 TSecr=1370923
72	5.396965	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
73	5.396973	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=36427 Win=42256 Len=0 TSval=1370923 TSecr=1370923

No.	Time	Source	Destination	Protocol	Length	Info
73	5.396973	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=36427 Win=42256 Len=0 TSval=1370923 TSecr=1370923
74	5.397270	10.1.1.1	10.4.4.1	TCP	1514	[TCP Previous segment lost] [TCP segment of a reassembled PDU]
75	5.397279	10.4.4.1	10.1.1.1	TCP	78	[TCP Dup ACK 73#1] 46897 > http [ACK] Seq=114 Ack=36427 Win=42256 Len=0 TSval=1370923 TSecr=1370923 SLE=39
76	5.397668	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
77	5.397680	10.4.4.1	10.1.1.1	TCP	78	[TCP Dup ACK 73#2] 46897 > http [ACK] Seq=114 Ack=36427 Win=42256 Len=0 TSval=1370923 TSecr=1370923 SLE=39
78	5.398050	10.1.1.1	10.4.4.1	TCP	1514	[TCP Fast Retransmission] [TCP segment of a reassembled PDU]
79	5.398063	10.4.4.1	10.1.1.1	TCP	78	46897 > http [ACK] Seq=114 Ack=37875 Win=40816 Len=0 TSval=1370923 TSecr=1370923 SLE=39323 SRE=42219
80	5.398440	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
81	5.398453	10.4.4.1	10.1.1.1	TCP	78	[TCP Dup ACK 79#1] 46897 > http [ACK] Seq=114 Ack=37875 Win=40816 Len=0 TSval=1370923 TSecr=1370923 SLE=39
82	5.398815	10.1.1.1	10.4.4.1	TCP	1514	[TCP Out-Of-Order] [TCP segment of a reassembled PDU]
83	5.398826	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
84	5.398876	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=43667 Win=39392 Len=0 TSval=1370923 TSecr=1370923
85	5.398878	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=45115 Win=38320 Len=0 TSval=1370923 TSecr=1370923
86	5.399209	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
87	5.399217	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=46563 Win=42256 Len=0 TSval=1370923 TSecr=1370923
88	5.399527	10.1.1.1	10.4.4.1	TCP	1514	[TCP Previous segment lost] [TCP segment of a reassembled PDU]
89	5.399535	10.4.4.1	10.1.1.1	TCP	78	[TCP Dup ACK 87#1] 46897 > http [ACK] Seq=114 Ack=46563 Win=42256 Len=0 TSval=1370923 TSecr=1370923 SLE=49
90	5.399866	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
91	5.399877	10.4.4.1	10.1.1.1	TCP	78	[TCP Dup ACK 87#2] 46897 > http [ACK] Seq=114 Ack=46563 Win=42256 Len=0 TSval=1370923 TSecr=1370923 SLE=49
92	5.400191	10.1.1.1	10.4.4.1	TCP	1514	[TCP Fast Retransmission] [TCP segment of a reassembled PDU]
93	5.400200	10.4.4.1	10.1.1.1	TCP	78	46897 > http [ACK] Seq=114 Ack=48011 Win=40816 Len=0 TSval=1370923 TSecr=1370923 SLE=49459 SRE=52355
94	5.400512	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
95	5.400520	10.4.4.1	10.1.1.1	TCP	78	[TCP Dup ACK 93#1] 46897 > http [ACK] Seq=114 Ack=48011 Win=40816 Len=0 TSval=1370924 TSecr=1370923 SLE=49
96	5.400827	10.1.1.1	10.4.4.1	TCP	1514	[TCP Out-Of-Order] [TCP segment of a reassembled PDU]
97	5.400832	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
98	5.400837	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=53803 Win=39392 Len=0 TSval=1370924 TSecr=1370924
99	5.400839	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=55251 Win=38320 Len=0 TSval=1370924 TSecr=1370924
100	5.401270	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]

No.	Time	Source	Destination	Protocol	Length	Info
100	5.401270	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
101	5.401271	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=56699 Win=45152 Len=0 TSval=1370924 TSecr=1370924
102	5.401564	10.1.1.1	10.4.4.1	TCP	1514	[TCP Previous segment lost] [TCP segment of a reassembled PDU]
103	5.401570	10.4.4.1	10.1.1.1	TCP	78	[TCP Dup ACK 101#1] 46897 > http [ACK] Seq=114 Ack=56699 Win=45152 Len=0 TSval=1370924 TSecr=1370924 SLE=59
104	5.401895	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
105	5.401903	10.4.4.1	10.1.1.1	TCP	78	[TCP Dup ACK 101#2] 46897 > http [ACK] Seq=114 Ack=56699 Win=45152 Len=0 TSval=1370924 TSecr=1370924 SLE=59
106	5.402229	10.1.1.1	10.4.4.1	TCP	1514	[TCP Fast Retransmission] [TCP segment of a reassembled PDU]
107	5.402237	10.4.4.1	10.1.1.1	TCP	78	46897 > http [ACK] Seq=114 Ack=58147 Win=48048 Len=0 TSval=1370924 TSecr=1370924 SLE=59595 SRE=62491
108	5.402544	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
109	5.402551	10.4.4.1	10.1.1.1	TCP	78	[TCP Dup ACK 107#1] 46897 > http [ACK] Seq=114 Ack=58147 Win=48048 Len=0 TSval=1370924 TSecr=1370924 SLE=59
110	5.402976	10.1.1.1	10.4.4.1	TCP	1514	[TCP Out-Of-Order] [TCP segment of a reassembled PDU]
111	5.402985	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
112	5.402995	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=63939 Win=50944 Len=0 TSval=1370924 TSecr=1370924
113	5.402998	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=65387 Win=53840 Len=0 TSval=1370924 TSecr=1370924
114	5.403491	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
115	5.403491	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=66835 Win=56736 Len=0 TSval=1370924 TSecr=1370924
116	5.403771	10.1.1.1	10.4.4.1	TCP	1514	[TCP Previous segment lost] [TCP segment of a reassembled PDU]
117	5.403777	10.4.4.1	10.1.1.1	TCP	78	[TCP Dup ACK 115#1] 46897 > http [ACK] Seq=114 Ack=66835 Win=56736 Len=0 TSval=1370924 TSecr=1370924 SLE=69
118	5.404057	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
119	5.404063	10.4.4.1	10.1.1.1	TCP	78	[TCP Dup ACK 115#2] 46897 > http [ACK] Seq=114 Ack=66835 Win=56736 Len=0 TSval=1370924 TSecr=1370924 SLE=69
120	5.404402	10.1.1.1	10.4.4.1	TCP	1514	[TCP Fast Retransmission] [TCP segment of a reassembled PDU]
121	5.404411	10.4.4.1	10.1.1.1	TCP	78	46897 > http [ACK] Seq=114 Ack=68283 Win=59632 Len=0 TSval=1370924 TSecr=1370924 SLE=69731 SRE=72627
122	5.404717	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
123	5.404724	10.4.4.1	10.1.1.1	TCP	78	[TCP Dup ACK 121#1] 46897 > http [ACK] Seq=114 Ack=68283 Win=59632 Len=0 TSval=1370925 TSecr=1370924 SLE=69
124	5.405040	10.1.1.1	10.4.4.1	TCP	1514	[TCP Out-Of-Order] [TCP segment of a reassembled PDU]
125	5.405048	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
126	5.405055	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=74075 Win=62528 Len=0 TSval=1370925 TSecr=1370925
127	5.405056	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=75523 Win=65424 Len=0 TSval=1370925 TSecr=1370925
128	5.405431	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]

129	5.405439	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=76971 Win=68320 Len=0 TSval=1370925 TSecr=1370925
130	5.405803	10.1.1.1	10.4.4.1	TCP	1514	[TCP Previous segment lost] [TCP segment of a reassembled PDU]
131	5.405822	10.4.4.1	10.1.1.1	TCP	78	[TCP Dup ACK 129#1] 46897 > http [ACK] Seq=114 Ack=76971 Win=68320 Len=0 TSval=1370925 TSecr=1370925 SLE=79
132	5.406153	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
133	5.406162	10.4.4.1	10.1.1.1	TCP	78	[TCP Dup ACK 129#2] 46897 > http [ACK] Seq=114 Ack=76971 Win=68320 Len=0 TSval=1370925 TSecr=1370925 SLE=79
134	5.406636	10.1.1.1	10.4.4.1	TCP	1514	[TCP Fast Retransmission] [TCP segment of a reassembled PDU]
135	5.406650	10.4.4.1	10.1.1.1	TCP	78	46897 > http [ACK] Seq=114 Ack=78419 Win=71216 Len=0 TSval=1370925 TSecr=1370925 SLE=79867 SRE=82763
136	5.407153	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
137	5.407171	10.4.4.1	10.1.1.1	TCP	78	[TCP Dup ACK 135#1] 46897 > http [ACK] Seq=114 Ack=78419 Win=71216 Len=0 TSval=1370925 TSecr=1370925 SLE=79
138	5.407566	10.1.1.1	10.4.4.1	TCP	1514	[TCP Out-Of-Order] [TCP segment of a reassembled PDU]
139	5.407577	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
140	5.407587	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=84211 Win=74112 Len=0 TSval=1370925 TSecr=1370925
141	5.407590	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=85659 Win=77008 Len=0 TSval=1370925 TSecr=1370925
142	5.407980	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
143	5.407987	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=87107 Win=79904 Len=0 TSval=1370925 TSecr=1370925
144	5.408492	10.1.1.1	10.4.4.1	TCP	1514	[TCP Previous segment lost] [TCP segment of a reassembled PDU]
145	5.408501	10.4.4.1	10.1.1.1	TCP	78	[TCP Dup ACK 143#1] 46897 > http [ACK] Seq=114 Ack=87107 Win=79904 Len=0 TSval=1370926 TSecr=1370925 SLE=9
146	5.408787	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
147	5.408794	10.4.4.1	10.1.1.1	TCP	78	[TCP Dup ACK 143#2] 46897 > http [ACK] Seq=114 Ack=87107 Win=79904 Len=0 TSval=1370926 TSecr=1370925 SLE=9
148	5.409076	10.1.1.1	10.4.4.1	TCP	1514	[TCP Fast Retransmission] [TCP segment of a reassembled PDU]
149	5.409084	10.4.4.1	10.1.1.1	TCP	78	46897 > http [ACK] Seq=114 Ack=88555 Win=82800 Len=0 TSval=1370926 TSecr=1370926 SLE=90003 SRE=92899
150	5.409411	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
151	5.409423	10.4.4.1	10.1.1.1	TCP	78	[TCP Dup ACK 149#1] 46897 > http [ACK] Seq=114 Ack=88555 Win=82800 Len=0 TSval=1370926 TSecr=1370926 SLE=9
152	5.409745	10.1.1.1	10.4.4.1	TCP	1514	[TCP Out-Of-Order] [TCP segment of a reassembled PDU]
153	5.409752	10.1.1.1	10.4.4.1	TCP	1514	[TCP segment of a reassembled PDU]
154	5.409758	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=94347 Win=85696 Len=0 TSval=1370926 TSecr=1370926
155	5.409760	10.4.4.1	10.1.1.1	TCP	66	46897 > http [ACK] Seq=114 Ack=95795 Win=84656 Len=0 TSval=1370926 TSecr=1370926

156	5.410077	10.1.1.1	10.4.4.1	TCP	1514 [TCP segment of a reassembled PDU]
157	5.410084	10.4.4.1	10.1.1.1	TCP	66 46897 > http [ACK] Seq=114 Ack=97243 Win=88592 Len=0 TSval=1370926 TSecr=1370926
158	5.410391	10.1.1.1	10.4.4.1	TCP	1514 [TCP Previous segment lost] [TCP segment of a reassembled PDU]
159	5.410399	10.4.4.1	10.1.1.1	TCP	78 [TCP Dup ACK 157#1] 46897 > http [ACK] Seq=114 Ack=97243 Win=88592 Len=0 TSval=1370926 TSecr=1370926 SLE
160	5.410705	10.1.1.1	10.4.4.1	TCP	1514 [TCP segment of a reassembled PDU]
161	5.410712	10.4.4.1	10.1.1.1	TCP	78 [TCP Dup ACK 157#2] 46897 > http [ACK] Seq=114 Ack=97243 Win=88592 Len=0 TSval=1370926 TSecr=1370926 SLE
162	5.411023	10.1.1.1	10.4.4.1	TCP	1514 [TCP Fast Retransmission] [TCP segment of a reassembled PDU]
163	5.411031	10.4.4.1	10.1.1.1	TCP	78 46897 > http [ACK] Seq=114 Ack=98691 Win=87152 Len=0 TSval=1370926 TSecr=1370926 SLE=100139 SRE=103035
164	5.411358	10.1.1.1	10.4.4.1	TCP	1514 [TCP segment of a reassembled PDU]
165	5.411365	10.1.1.1	10.4.4.1	TCP	318 [TCP segment of a reassembled PDU]
166	5.411369	10.4.4.1	10.1.1.1	TCP	78 [TCP Dup ACK 163#1] 46897 > http [ACK] Seq=114 Ack=98691 Win=87152 Len=0 TSval=1370926 TSecr=1370926 SLE
167	5.411370	10.4.4.1	10.1.1.1	TCP	78 [TCP Dup ACK 163#2] 46897 > http [ACK] Seq=114 Ack=98691 Win=87152 Len=0 TSval=1370926 TSecr=1370926 SLE
168	5.411754	10.1.1.1	10.4.4.1	HTTP	1514 [TCP Fast Retransmission] HTTP/1.1 200 Ok (text/plain)
169	5.411765	10.4.4.1	10.1.1.1	TCP	66 46897 > http [ACK] Seq=114 Ack=104736 Win=85248 Len=0 TSval=1370926 TSecr=1370926
170	5.411863	10.4.4.1	10.1.1.1	TCP	66 46897 > http [FIN, ACK] Seq=114 Ack=104736 Win=88592 Len=0 TSval=1370926 TSecr=1370926
171	5.412485	10.1.1.1	10.4.4.1	TCP	66 http > 46897 [ACK] Seq=104736 Ack=115 Win=14480 Len=0 TSval=1370926 TSecr=1370926

Figura 12 – Troca de mensagens HTTP.

Protocolos	Uso da camada de transporte	Eficiência na transferência	complexidade e	segurança
FTP	TCP	3º mais eficiente	3º mais complexo	Login
TFTP	UDP	1º mais eficiente	4º mais complexo	
SFTP	TCP	4º mais eficiente	1º mais complexo	Login + encriptação
HTTP	TCP	2º mais eficiente	2º mais complexo	

Eficiência na transferência:

Devido às características de cada protocolo, o TFTP é o mais eficiente por ser mais rápido a transferir os dados, quer por não implementar segurança quer por usar UDP não fazendo verificação na receção de erros.

Em segundo vem o HTTP que não implementa segurança mas utiliza TCP, verificando os pacotes mal os recebe.

Em 3º lugar temos o FTP que utiliza login e TCP e em último temos o SFTP que para além de TCP e login utiliza também encriptação.

Complexidade:

7	26.379460	10.4.4.1	10.1.1.1	TFTP	56 Read Request, File: file1, Transfer type: octet
8	26.380565	10.1.1.1	10.4.4.1	TFTP	239 Data Packet, Block: 1 (last)
9	26.380622	10.4.4.1	10.1.1.1	TFTP	46 Acknowledgement, Block: 1

Figura 13 – Complexidade TFTP.

Em matéria de complexidade o menos complexo trata-se do TFTP que faz uso apenas de 3 frames para além dos ARPs trocados visto que não faz uso de medidas de segurança ou autenticação, análogamente o HTTP não usa medidas de segurança ou encriptação, mas usa o protocolo TCP na camada de transporte acabando por fazer tratamento de erros na receção dos frames ao contrário do TFTP

Por fazer uso do login e de encriptação o SFTP é o mais complexo e segue-se o FTP em complexidade visto que só usa login e não encriptação.

Segurança:

```
RSA key fingerprint is c1:0a:8c:2a:4e:a0:e6:88:a5:bc:b2:37:b5:7d:df:91.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.1.1.1' (RSA) to the list of known hosts.
core@10.1.1.1's password:
Connected to 10.1.1.1.
sftp> 
```

Figura 14 - SFTP – pedido de password

10	15.600239	10.1.1.1	10.4.4.1	SSHv2	107 Server Protocol: SSH-2.0-OpenSSH_5.9p1 Debian-Subuntu1.4r
11	15.600288	10.4.4.1	10.1.1.1	TCP	66 48618 > ssh [ACK] Seq=1 Ack=42 Win=14608 Len=0 TSval=1316861 TSecr=1316861
12	15.600363	10.4.4.1	10.1.1.1	SSHv2	107 Client Protocol: SSH-2.0-OpenSSH_5.9p1 Debian-Subuntu1.4r
13	15.600742	10.1.1.1	10.4.4.1	TCP	66 ssh > 48618 [ACK] Seq=12 Ack=42 Win=14480 Len=0 TSval=1316861 TSecr=1316861
14	15.603105	10.1.1.1	10.4.4.1	SSHv2	101 Server: Key Exchange Init
15	15.612180	10.4.4.1	10.1.1.1	SSHv2	133 Client: Key Exchange Init
16	15.651169	10.1.1.1	10.4.4.1	TCP	66 ssh > 48618 [ACK] Seq=994 Ack=1314 Win=17376 Len=0 TSval=1316874 TSecr=1316864
17	15.651189	10.4.4.1	10.1.1.1	SSHv2	146 Client: Diffie-Hellman Key Exchange Init
18	15.652083	10.1.1.1	10.4.4.1	TCP	66 ssh > 48618 [ACK] Seq=994 Ack=1394 Win=17376 Len=0 TSval=1316874 TSecr=1316874
19	15.664393	10.1.1.1	10.4.4.1	SSHv2	722 Server: New Keys
20	15.666859	10.4.4.1	10.1.1.1	SSHv2	82 Client: New Keys
21	15.706008	10.1.1.1	10.4.4.1	TCP	66 ssh > 48618 [ACK] Seq=1650 Ack=1410 Win=17376 Len=0 TSval=1316888 TSecr=1316877
22	15.706026	10.4.4.1	10.1.1.1	TCP	114 [TCP segment of a reassembled PDU]
23	15.706334	10.1.1.1	10.4.4.1	TCP	66 ssh > 48618 [ACK] Seq=1650 Ack=1458 Win=17376 Len=0 TSval=1316888 TSecr=1316888
24	15.706396	10.1.1.1	10.4.4.1	TCP	114 [TCP segment of a reassembled PDU]
25	15.706455	10.4.4.1	10.1.1.1	TCP	130 [TCP segment of a reassembled PDU]
26	15.707725	10.1.1.1	10.4.4.1	TCP	130 [TCP segment of a reassembled PDU]
27	15.746031	10.4.4.1	10.1.1.1	TCP	66 48618 > ssh [ACK] Seq=1522 Ack=1762 Win=19408 Len=0 TSval=1316898 TSecr=1316888
28	18.545545	10.4.4.1	10.1.1.1	TCP	210 [TCP segment of a reassembled PDU]
29	18.581457	10.1.1.1	10.4.4.1	TCP	98 [TCP segment of a reassembled PDU]
30	18.581671	10.4.4.1	10.1.1.1	TCP	66 48618 > ssh [ACK] Seq=1666 Ack=1794 Win=19408 Len=0 TSval=1317606 TSecr=1317606
31	18.581791	10.4.4.1	10.1.1.1	TCP	194 [TCP segment of a reassembled PDU]
32	18.623015	10.1.1.1	10.4.4.1	TCP	66 ssh > 48618 [ACK] Seq=1794 Ack=1794 Win=22464 Len=0 TSval=1317617 TSecr=1317606
33	18.867924	10.1.1.1	10.4.4.1	TCP	114 [TCP segment of a reassembled PDU]
34	18.868018	10.4.4.1	10.1.1.1	TCP	194 [TCP segment of a reassembled PDU]

Figura 15 - SFTP – uso de encriptação

8	3.069123	10.4.4.1	10.1.1.1	FTP	82 Request: USER anonymous
9	3.069733	10.1.1.1	10.4.4.1	TCP	66 ftp > 59690 [ACK] Seq=21 Ack=17 Win=14480 Len=0 TSval=1003088 TSecr=1003088
10	3.069743	10.1.1.1	10.4.4.1	FTP	100 Response: 331 Please specify the password.
11	3.069762	10.4.4.1	10.1.1.1	TCP	66 59690 > ftp [ACK] Seq=17 Ack=55 Win=14608 Len=0 TSval=1003089 TSecr=1003088
12	5.003257	00:00:00_aa:00:13	00:00:00_aa:00:1d	ARP	42 Who has 10.4.4.1? Tell 10.4.4.254
13	5.003281	00:00:00_aa:00:1d	00:00:00_aa:00:13	ARP	42 10.4.4.1 is at 00:00:00_aa:00:1d
14	5.373334	10.4.4.1	10.1.1.1	FTP	77 Request: PASS core
15	5.392141	10.1.1.1	10.4.4.1	FTP	89 Response: 230 Login successful.
16	5.392307	10.4.4.1	10.1.1.1	TCP	66 59690 > ftp [ACK] Seq=28 Ack=78 Win=14608 Len=0 TSval=1003669 TSecr=1003669
17	5.392464	10.4.4.1	10.1.1.1	FTP	72 Request: SYST
18	5.394596	10.1.1.1	10.4.4.1	FTP	85 Response: 215 UNIX Type: L8
19	5.434624	10.4.4.1	10.1.1.1	TCP	66 59690 > ftp [ACK] Seq=34 Ack=97 Win=14608 Len=0 TSval=1003680 TSecr=1003670

Figura 16 - FTP – uso de autenticação

No caso da segurança o mais seguro é o SFTP pois para além de utilizar autenticação, usa também encriptação na troca de dados. Em segundo lugar fica o FTP que utiliza apenas login para garantir a autenticidade do utilizador.

Tanto o TFTP como o HTTP são menos seguros visto que não fazem uso nem de autenticação nem de encriptação.

Exercício 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, relacionando com alguns dos mecanismos de transporte envolvidos).

Resposta:

66	82.472971	10.3.3.1	10.1.1.1	FTP	78 Request: RETR file1
67	82.473737	10.1.1.1	10.3.3.1	TCP	74 ftp-data > 48668 [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=1542665 TSecr=0 WS=16
68	82.479039	10.3.3.1	10.1.1.1	TCP	74 48668 > ftp-data [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval=1542665 TSecr=1542665
69	82.479084	10.3.3.1	10.1.1.1	TCP	74 48668 > ftp-data [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval=1542665 TSecr=1542665
70	82.479803	10.1.1.1	10.3.3.1	TCP	66 ftp-data > 48668 [ACK] Seq=1 Ack=1 Win=14608 Len=0 TSval=1542667 TSecr=1542665
71	82.480028	10.1.1.1	10.3.3.1	TCP	66 [TCP Dup ACK 70#1] ftp-data > 48668 [ACK] Seq=1 Ack=1 Win=14608 Len=0 TSval=1542667 TSecr=1542665
72	82.480072	10.1.1.1	10.3.3.1	FTP	130 Response: 150 Opening BINARY mode data connection for file1 (437 bytes).
73	82.480105	10.1.1.1	10.3.3.1	FTP-DATA	503 FTP Data: 437 bytes
74	82.480129	10.1.1.1	10.3.3.1	TCP	66 ftp-data > 48668 [FIN, ACK] Seq=438 Ack=1 Win=14608 Len=0 TSval=1542667 TSecr=1542665
75	82.485621	10.3.3.1	10.1.1.1	TCP	66 48668 > ftp-data [ACK] Seq=1 Ack=438 Win=15552 Len=0 TSval=1542667 TSecr=1542667
76	82.485622	10.3.3.1	10.1.1.1	TCP	66 48668 > ftp-data [FIN, ACK] Seq=1 Ack=439 Win=15552 Len=0 TSval=1542667 TSecr=1542667
77	82.485623	10.3.3.1	10.1.1.1	TCP	66 48668 > ftp-data [FIN, ACK] Seq=1 Ack=439 Win=15552 Len=0 TSval=1542667 TSecr=1542667
78	82.486483	10.1.1.1	10.3.3.1	TCP	66 ftp-data > 48668 [ACK] Seq=439 Ack=2 Win=14608 Len=0 TSval=1542668 TSecr=1542667
79	82.486644	10.1.1.1	10.3.3.1	TCP	66 [TCP Dup ACK 78#1] ftp-data > 48668 [ACK] Seq=439 Ack=2 Win=14608 Len=0 TSval=1542668 TSecr=1542667
80	82.486710	10.1.1.1	10.3.3.1	FTP	90 Response: 226 Transfer complete.

Figura 17 – Captura do FTP entre o host Alfa e o Servidor1.

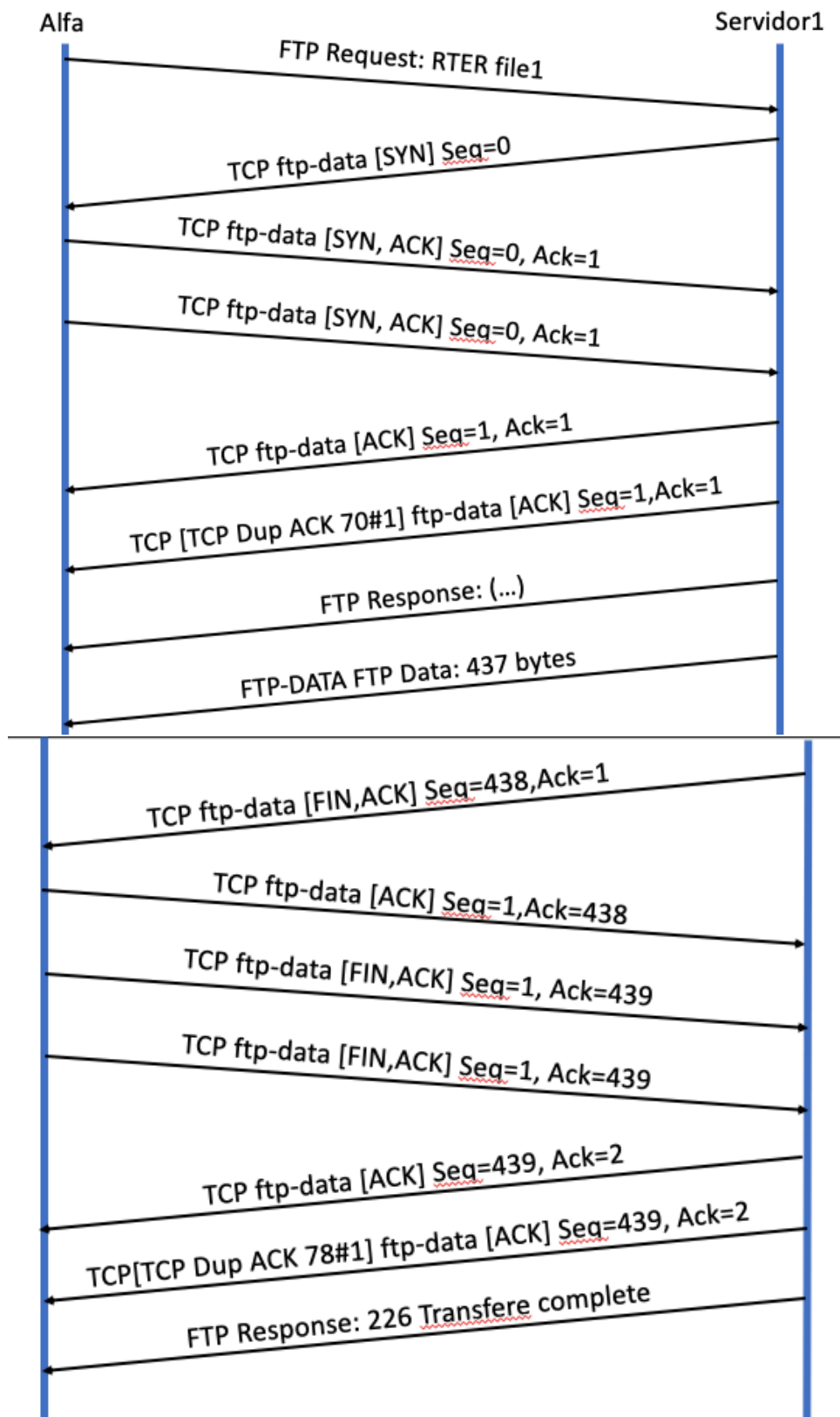


Figura 18 – Diagrama de transferências do FTP entre o host Alfa e o Servidor1.

Na figura 17, está representado o diagrama de transferências entre o host Alfa1 e o Servidor1, através da aplicação FTP. Inicialmente é feito um REQUEST [RTER file1], ao qual o Servidor1 envia um TCP[SYN] com Seq = 0.

Ao receber este pacote o host Alfa, envia dois pacotes TCP [SYN, ACK], com a mesma informação, pelo que o Servidor1 ao receber os mesmos, envia um pacote TCP [Dup ACK 70#1] ftp-data [ACK] com **Ack = 1**, com o objetivo de “avisar” o Alfa de algo não está bem, pois houve **duplicação** de dois pacotes. Depois é enviado o file1 através de **FTP-DATA (437 bytes)**.

O servidor informa o host Alfa da finalização da transferência, pelo que o cliente, envia inicialmente um TCP ftp-data [ACK] e depois dois TCP ftp-data [FIN,ACK] Seq=1, Ack=439. O servidor, ao receber, vai identificar com uma duplicação, e informar o cliente do mesmo com TCP [TCP Dup ACK 78#1] ftp-data [ACK] Seq=439, Ack=2.

Por fim, o servidor, envia um FTP Response, a informar o cliente de que a transferência foi completada.

Conclusões:

Em suma, os protocolos de transportes TCP e UDP tem bastantes diferenças. O protocolo TCP - *Transmission Control Protocol*, é caracterizado por ser um protocolo da camada de transporte muito completo, pois tem controlo de fluxo, erros e congestão e não está dependente de um circuito pré-determinado.

Por outro lado, o protocolo UDP - *User Datagram Protocol* é caracterizado por ser um protocolo simples da camada de transporte, pois não tem os controlos tal como o TCP, pelo que o circuito é pré-determinado.

No entanto, o protocolo UDP tem imensas vantagens para aplicações, onde a velocidade de transmissão é importante, como por exemplo streaming, pois não “perde” tempo em controlo de erros, etc..., com UDP tenho controlo do que mando e quando mando.

A utilização de cada um dos protocolos de transporte, depende do tipo de aplicação e qual a sua finalidade, e assim, com estas informações, consegue-se determinar o protocolo mais adequado.

Por fim, a nível de segurança, descobriu-se que a aplicação mais segura é o SFTP, de seguida o FTP, que usa o protocolo de transporte TCP, depois o HTTP, e por fim o TFTP que não tem segurança nenhum, pelo que usa o protocolo UDP.