



UNIVERSIDADE DO MINHO

DEPARTAMENTO DE INFORMÁTICA

Comunicações por Computador  
Trabalho Prático 1  
Grupo Nº 10

Ariana Lousada (A87998)      Rui Armada (A90468)  
Sofia Santos (A89615)

21 de outubro de 2021

# Capítulo 1

## Questões e Respostas

Para a resolução deste trabalho, foram-nos propostas várias questões, as quais vamos passar a responder neste capítulo:

### Questão 1

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	33446	8
Telnet	TELNET	TCP	23	32
FTP	FTP	TCP	21	32
browser/HTTP	HTTP	TCP	80	32
nslookup	DNS	UDP	53	31
ssh	SSHv2	TCP	22	32
TFTP	TFTP	UDP	69	44

No.	Time	Source	Destination	Protocol	Length	Info
983	1.695795990	192.168.1.68	193.136.19.254	UDP	74	39975 → 33440 Len=32
984	1.695809750	192.168.1.68	193.136.19.254	UDP	74	60909 → 33441 Len=32
985	1.695821980	192.168.1.68	193.136.19.254	UDP	74	45287 → 33442 Len=32
986	1.695833420	192.168.1.68	193.136.19.254	UDP	74	36266 → 33443 Len=32
987	1.695841350	192.168.1.68	193.136.19.254	UDP	74	56088 → 33444 Len=32
988	1.695861380	192.168.1.68	193.136.19.254	UDP	74	59269 → 33445 Len=32
989	1.695876380	192.168.1.68	193.136.19.254	UDP	74	48797 → 33446 Len=32
990	1.695887680	192.168.1.68	193.136.19.254	UDP	74	33569 → 33447 Len=32
991	1.695900420	192.168.1.68	193.136.19.254	UDP	74	59822 → 33448 Len=32
992	1.695911890	192.168.1.68	193.136.19.254	UDP	74	36199 → 33449 Len=32
993	1.696198220	192.168.1.1	192.168.1.68	ICMP	102	Time-to-live exceeded (Time to live exceeded)
994	1.696353600	192.168.1.68	193.136.19.254	UDP	74	49796 → 33450 Len=32
995	1.696364990	192.168.1.1	192.168.1.68	ICMP	102	Time-to-live exceeded (Time to live exceeded)

Frame 989: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface enp3s0, id 0

Ethernet II, Src: Micro-St\_fd:39:c5 (4c:cc:6a:fd:39:c5), Dst: HuaweiTe\_95:f2:62 (f4:79:60:95:f2:62)

Internet Protocol Version 4, Src: 192.168.1.68, Dst: 193.136.19.254

0100 .... = Version: 4

.... 0101 = Header Length: 20 bytes (5)

Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 60

Identification: 0x7aee (31470)

Flags: 0x00

Fragment Offset: 0

Time to Live: 5

Protocol: UDP (17)

Header Checksum: 0xa350 [validation disabled]

[Header checksum status: Unverified]

Source Address: 192.168.1.68

Destination Address: 193.136.19.254

User Datagram Protocol, Src Port: 48797, Dst Port: 33446

Source Port: 48797

Destination Port: 33446

Length: 40

Checksum: 0x97ac [unverified]

[Checksum Status: Unverified]

[Stream index: 21]

[Timestamps]

UDP payload (32 bytes)

Figura 1.1: Traceroute.

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
telnet						
No.	Time	Source	Destination	Protocol	Length Info	
1385	1.893585015	192.168.1.68	193.136.9.183	TELNET	93	Telnet Data ...
1408	1.919807842	193.136.9.183	192.168.1.68	TELNET	78	Telnet Data ...
1421	1.934832412	193.136.9.183	192.168.1.68	TELNET	105	Telnet Data ...
1423	1.935010482	192.168.1.68	193.136.9.183	TELNET	172	Telnet Data ...
1437	1.950453272	193.136.9.183	192.168.1.68	TELNET	69	Telnet Data ...

▶ Frame 1385: 93 bytes on wire (744 bits), 93 bytes captured (744 bits) on interface enp35s0, id 0  
 ▶ Ethernet II, Src: Micro-St\_fd:39:c5 (4c:cc:6a:fd:39:c5), Dst: HuaweiTe\_95:f2:62 (f4:79:60:95:f2:62)  
 ▶ Internet Protocol Version 4, Src: 192.168.1.68, Dst: 193.136.9.183  
 ▼ Transmission Control Protocol, Src Port: 59084, Dst Port: 23, Seq: 1, Ack: 1, Len: 27  
   Source Port: 59084  
   Destination Port: 23  
   [Stream index: 5]  
   [TCP Segment Len: 27]  
   Sequence Number: 1 (relative sequence number)  
   Sequence Number (raw): 2949992186  
   [Next Sequence Number: 28 (relative sequence number)]  
   Acknowledgment Number: 1 (relative ack number)  
   Acknowledgment number (raw): 1459418300  
   1000 .... = Header Length: 32 bytes (8)  
   Flags: 0x018 (PSH, ACK)  
   Window: 502  
   [Calculated window size: 64256]  
   [Window size scaling factor: 128]  
   Checksum: 0x8d6d [unverified]  
   [Checksum Status: Unverified]  
   Urgent Pointer: 0  
   Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps

Figura 1.2: Telnet.

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
ftp						
No.	Time	Source	Destination	Protocol	Length Info	
2939	4.991462707	193.136.9.183	192.168.1.68	FTP	86	Response: 220 (vsFTPd 2.
4974	8.910709841	192.168.1.68	193.136.9.183	FTP	75	Request: USER cc
4984	8.927173762	193.136.9.183	192.168.1.68	FTP	100	Response: 331 Please spe
7058	12.410887119	192.168.1.68	193.136.9.183	FTP	79	Request: PASS cc2021
7094	12.495858953	193.136.9.183	192.168.1.68	FTP	89	Response: 230 Login succ
7096	12.495911023	192.168.1.68	193.136.9.183	FTP	72	Request: SYST
7103	12.510596542	193.136.9.183	192.168.1.68	FTP	85	Response: 215 UNIX Type:

▶ Frame 7096: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface enp35s0, id 0  
 ▶ Ethernet II, Src: Micro-St\_fd:39:c5 (4c:cc:6a:fd:39:c5), Dst: HuaweiTe\_95:f2:62 (f4:79:60:95:f2:62)  
 ▶ Internet Protocol Version 4, Src: 192.168.1.68, Dst: 193.136.9.183  
 ▼ Transmission Control Protocol, Src Port: 50754, Dst Port: 21, Seq: 23, Ack: 78, Len: 6  
   Source Port: 50754  
   Destination Port: 21  
   [Stream index: 10]  
   [TCP Segment Len: 6]  
   Sequence Number: 23 (relative sequence number)  
   Sequence Number (raw): 2151510529  
   [Next Sequence Number: 29 (relative sequence number)]  
   Acknowledgment Number: 78 (relative ack number)  
   Acknowledgment number (raw): 1935712462  
   1000 .... = Header Length: 32 bytes (8)

Figura 1.3: FTP.

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
http						
No.	Time	Source	Destination	Protocol	Length Info	
263	1.896882487	192.168.1.68	193.136.9.240	HTTP	583	GET /disciplinas/CC-MIEI
269	1.912530837	193.136.9.240	192.168.1.68	HTTP	256	HTTP/1.1 304 Not Modifie
295	2.040561149	192.168.1.68	193.136.9.240	HTTP	567	GET /disciplinas/CC-MIEI
298	2.055502129	193.136.9.240	192.168.1.68	HTTP	255	HTTP/1.1 304 Not Modifie

<p>Frame 263: 583 bytes on wire (4664 bits), 583 bytes captured (4664 bits) on interface enp35s0, id 0</p> <p>Ethernet II, Src: Micro-St_fd:39:c5 (4c:cc:6a:fd:39:c5), Dst: HuaweiTe_95:f2:62 (f4:79:60:95:f2:62)</p> <p>Internet Protocol Version 4, Src: 192.168.1.68, Dst: 193.136.9.240</p> <p>Transmission Control Protocol, Src Port: 60102, Dst Port: 80, Seq: 1, Ack: 1, Len: 517</p> <p>Source Port: 60102</p> <p>Destination Port: 80</p> <p>[Stream index: 3]</p> <p>[TCP Segment Len: 517]</p> <p>Sequence Number: 1 (relative sequence number)</p> <p>Sequence Number (raw): 2103308337</p> <p>[Next Sequence Number: 518 (relative sequence number)]</p> <p>Acknowledgment Number: 1 (relative ack number)</p> <p>Acknowledgment number (raw): 4107438553</p> <p>1000 .... = Header Length: 32 bytes (8)</p>
--

Figura 1.4: HTTP.

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
dns						
No.	Time	Source	Destination	Protocol	Length Info	
878	1.659484387	192.168.1.68	192.168.1.1	DNS	73	Standard query 0x7fc7 A
884	1.662679029	192.168.1.1	192.168.1.68	DNS	89	Standard query response
885	1.662828769	192.168.1.68	192.168.1.1	DNS	73	Standard query 0x4491 AA
893	1.669363333	192.168.1.1	192.168.1.68	DNS	127	Standard query response

<p>Frame 878: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface enp35s0, id 0</p> <p>Ethernet II, Src: Micro-St_fd:39:c5 (4c:cc:6a:fd:39:c5), Dst: HuaweiTe_95:f2:62 (f4:79:60:95:f2:62)</p> <p>Internet Protocol Version 4, Src: 192.168.1.68, Dst: 192.168.1.1</p> <p>User Datagram Protocol, Src Port: 34280, Dst Port: 53</p> <p>Source Port: 34280</p> <p>Destination Port: 53</p> <p>Length: 39</p> <p>Checksum: 0x83ce [unverified]</p> <p>[Checksum Status: Unverified]</p> <p>[Stream index: 6]</p> <p>[Timestamps]</p> <p>UDP payload (31 bytes)</p>
---

Figura 1.5: Nslookup.

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
ssh						
No.	Time	Source	Destination	Protocol	Length Info	
2650	4.897077507	192.168.1.68	193.136.9.183	SSHv2	87	Client: Protocol
2690	4.945572228	193.136.9.183	192.168.1.68	SSHv2	107	Server: Protocol
2693	4.945869338	192.168.1.68	193.136.9.183	SSHv2	178	Client: Key Excha
2698	4.960324427	193.136.9.183	192.168.1.68	SSHv2	1050	Server: Key Excha
2701	4.961074538	192.168.1.68	193.136.9.183	SSHv2	146	Client: Elliptic
2707	4.985400144	193.136.9.183	192.168.1.68	SSHv2	378	Server: Elliptic
2709	4.985991684	192.168.1.68	193.136.9.183	SSHv2	82	Client: New Keys
2728	5.038392237	192.168.1.68	193.136.9.183	SSHv2	106	Client: Encrvoted

▶ Frame 2650: 87 bytes on wire (696 bits), 87 bytes captured (696 bits) on interface enp35s0, id 0  
 ▶ Ethernet II, Src: Micro-St\_fd:39:c5 (4c:cc:6a:fd:39:c5), Dst: HuaweiTe\_95:f2:62 (f4:79:60:95:f2:62)  
 ▶ Internet Protocol Version 4, Src: 192.168.1.68, Dst: 193.136.9.183  
 ▶ Transmission Control Protocol, Src Port: 54582, Dst Port: 22, Seq: 1, Ack: 1, Len: 21  
   Source Port: 54582  
   Destination Port: 22  
   [Stream index: 10]  
   [TCP Segment Len: 21]  
   Sequence Number: 1 (relative sequence number)  
   Sequence Number (raw): 198444619  
   [Next Sequence Number: 22 (relative sequence number)]  
   Acknowledgment Number: 1 (relative ack number)  
   Acknowledgment number (raw): 3790323867  
   1000 .... = Header Length: 32 bytes (8)

Figura 1.6: Ssh.

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
tftp						
No.	Time	Source	Destination	Protocol	Length Info	
1127	1.745205591	192.168.1.68	193.136.9.183	TFTP	86	Read Request, File: file
6225	8.952234888	192.168.1.68	193.136.9.183	TFTP	86	Read Request, File: file
11111	15.957953166	192.168.1.68	193.136.9.183	TFTP	86	Read Request, File: file
16304	22.965567656	192.168.1.68	193.136.9.183	TFTP	86	Read Request, File: file
21046	29.971175325	192.168.1.68	193.136.9.183	TFTP	86	Read Request, File: file
28275	40.022165852	192.168.1.68	193.136.9.183	TFTP	95	Read Request, File: text
34220	47.230033729	192.168.1.68	193.136.9.183	TFTP	95	Read Request, File: text

▶ Frame 1127: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface enp35s0, id 0  
 ▶ Ethernet II, Src: Micro-St\_fd:39:c5 (4c:cc:6a:fd:39:c5), Dst: HuaweiTe\_95:f2:62 (f4:79:60:95:f2:62)  
 ▶ Internet Protocol Version 4, Src: 192.168.1.68, Dst: 193.136.9.183  
 ▶ User Datagram Protocol, Src Port: 48598, Dst Port: 69  
   Source Port: 48598  
   Destination Port: 69  
   Length: 52  
   Checksum: 0x8d71 [unverified]  
   [Checksum Status: Unverified]  
   [Stream index: 6]  
   ▶ [Timestamps]  
   UDP payload (44 bytes)

Figura 1.7: TFTP.

## Questão 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 nos dados como nas confirmações.

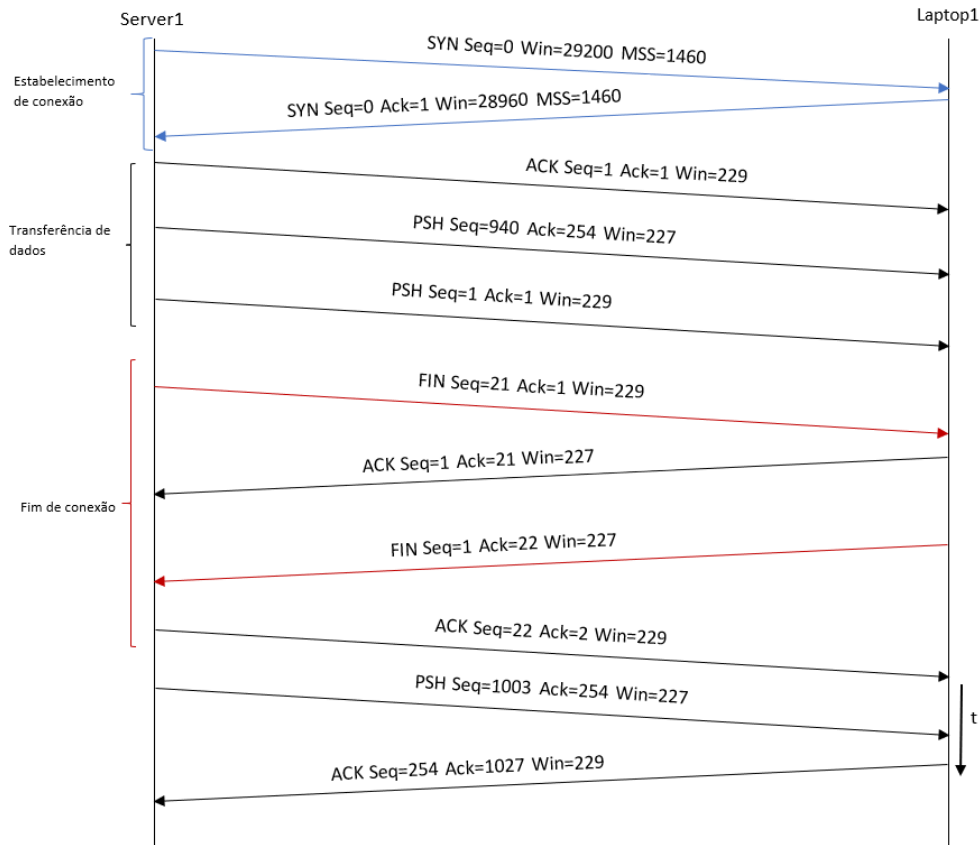


Figura 1.8: Esquema da transferência do file 1 por FTP

193	261.900912491	10.1.1.1	10.4.4.1	TCP	74 20 → 52327 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1383494257 TSecr=0 WS=128
194	261.901430644	10.4.4.1	10.1.1.1	TCP	74 52327 → 20 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=282276294 TSecr=1383494257 WS=128
195	261.901563350	10.1.1.1	10.4.4.1	TCP	66 20 → 52327 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=1383494258 TSecr=282276294
196	261.901610313	10.1.1.1	10.4.4.1	FTP	129 Response: 150 Opening BINARY mode data connection for file1 (20 bytes).
197	261.901710111	10.1.1.1	10.4.4.1	FTP-DA	86 FTP Data: 20 bytes (PORT) (RETR file1)
198	261.901711549	10.1.1.1	10.4.4.1	TCP	66 20 → 52327 [FIN, ACK] Seq=21 Ack=1 Win=29312 Len=0 TSval=1383494258 TSecr=282276294
199	261.901874447	10.4.4.1	10.1.1.1	TCP	66 52327 → 20 [ACK] Seq=1 Ack=21 Win=29056 Len=0 TSval=282276294 TSecr=1383494258
200	261.901884389	10.4.4.1	10.1.1.1	TCP	66 52327 → 20 [FIN, ACK] Seq=1 Ack=22 Win=29056 Len=0 TSval=282276294 TSecr=1383494258
201	261.902006362	10.1.1.1	10.4.4.1	TCP	66 20 → 52327 [ACK] Seq=22 Ack=2 Win=29312 Len=0 TSval=1383494259 TSecr=282276294
202	261.902000405	10.1.1.1	10.4.4.1	FTP	90 Response: 226 Transfer complete.
203	261.902212381	10.4.4.1	10.1.1.1	TCP	66 48454 → 21 [ACK] Seq=254 Ack=1027 Win=29312 Len=0 TSval=282276295 TSecr=1383494258

Figura 1.9: Captura wireshark da transferência do file 1 por FTP

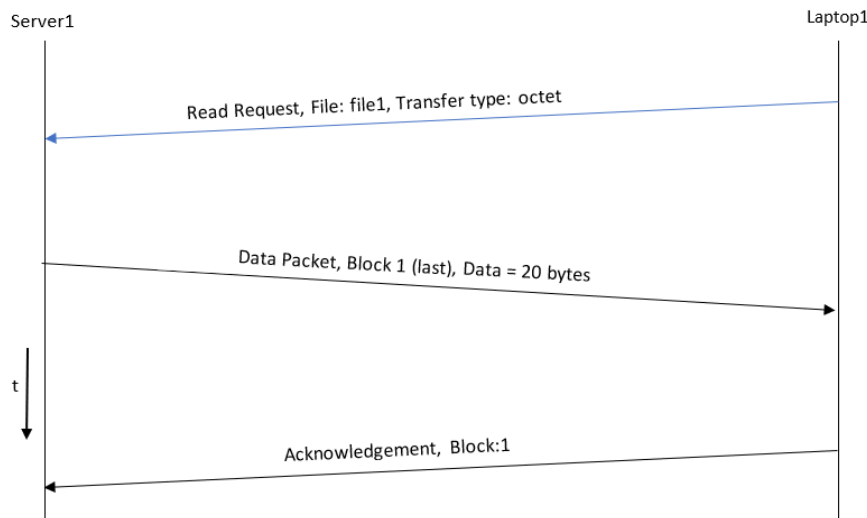


Figura 1.10: Esquema da transferência do file 1 por TFTP

36	166.770809379	10.4.4.1	10.1.1.1	TFTP	56 Read Request, File: file1, Transfer type: octet
37	166.771192801	10.1.1.1	10.4.4.1	TFTP	66 Data Packet, Block: 1 (last)
38	166.771558622	10.4.4.1	10.1.1.1	TFTP	46 Acknowledgement, Block: 1

Figura 1.11: Captura wireshark da transferência do file 1 por TFTP

### Questão 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;

#### Resposta:

Pela análise da transferência dos ficheiros por SFTP, podemos dizer que é uma aplicação segura, uma vez que requer sempre a autenticação por parte do cliente. Como utiliza SSH acaba por ter um *overhead* maior, o que diminui a sua eficiência por ser mais complexo que o resto das aplicações. Utiliza o protocolo TCP como protocolo da camada de transporte.

Na transferência dos ficheiros por FTP observamos que, em comparação com as restantes aplicações, apresenta um elevado *overhead*, o que afeta significativamente a sua eficiência. Assim como o SFTP utiliza o TCP como protocolo da camada de transporte, não apresentando medidas adicionais de segurança. Em termos de complexidade, é uma aplicação mais básica de transferência fiável de ficheiros.

Na transferência dos ficheiros por TFTP observámos que este é um serviço de transferência não fiável de ficheiros, uma vez que utiliza o UDP como protocolo da camada de transporte. Tal como o FTP, não implementa medidas de segurança adicionais e não oferece autenticação por parte do cliente. Isto tudo leva a que este serviço tenha um baixo *overhead*, o que o torna mais eficiente que os dois serviços anteriormente mencionados.

Por fim, na transferência por HTTP observámos que é bastante inseguro, uma vez que qualquer pessoa pode aceder ao conteúdo durante a transferência dos ficheiros. Este serviço, tal como o SFTP e o FTP, utiliza o TCP como protocolo da camada de transporte.

Através da análise dos timestamps nas tramas de transferência dos ficheiros, podemos comparar facilmente a eficiência de cada um dos protocolos.

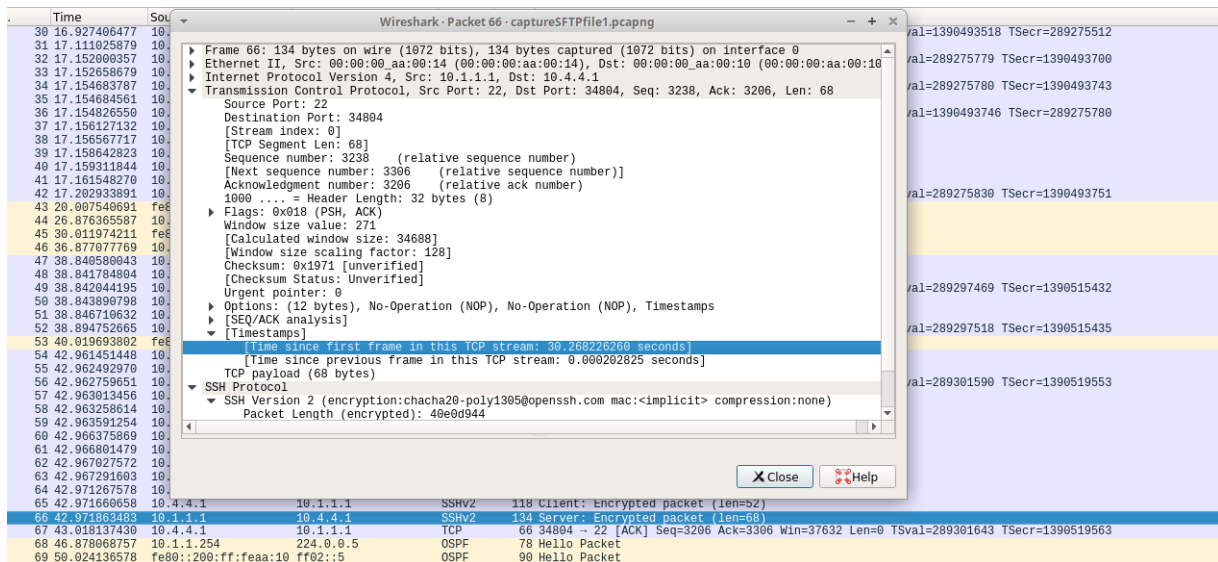


Figura 1.12: Tempo de transferência do ficheiro 1 por SFTP

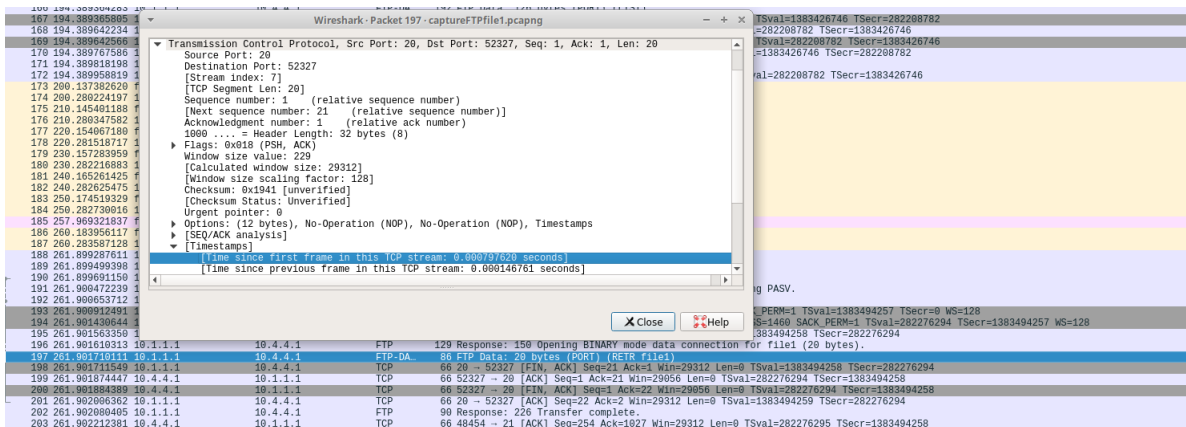


Figura 1.13: Tempo de transferência do ficheiro 1 por FTP

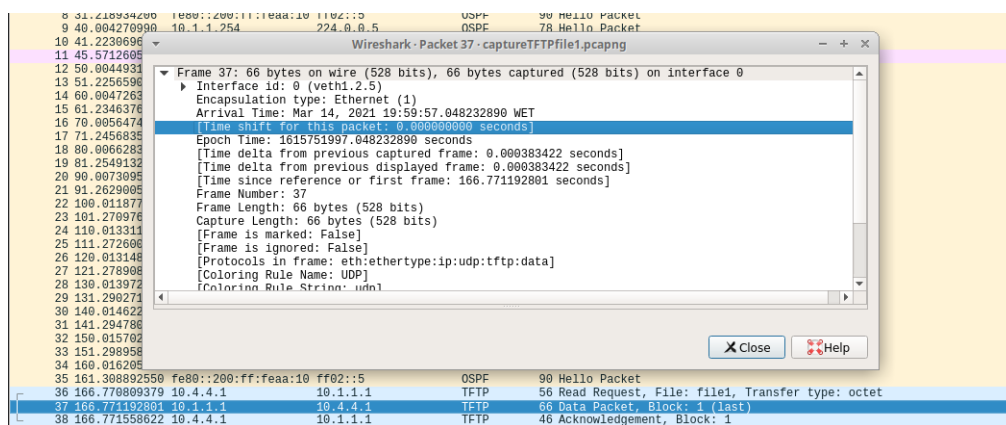


Figura 1.14: Tempo de transferência do ficheiro 1 por TFTP



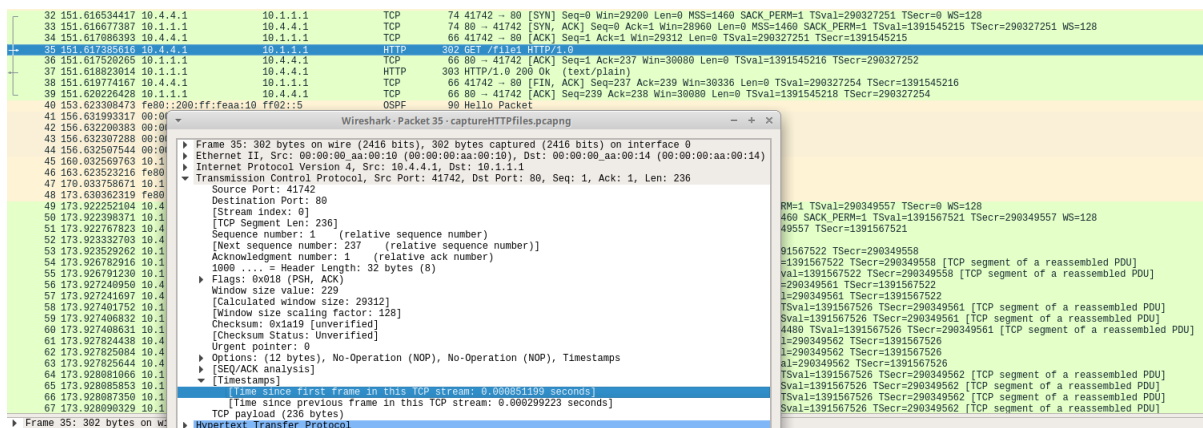


Figura 1.15: Tempo de transferência do ficheiro 1 por HTTP

## Questão 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:

Um dos aspetos mais importantes a considerar durante o desenvolvimento de aplicações é a escolha do protocolo de transporte.

Hoje em dia, existem dois protocolos mais utilizados que são praticamente o oposto um do outro: o UDP e o TCP.

O TCP é um protocolo de transporte fiável que garante que todos os pacotes são enviados e recebidos com sucesso pelo recetor. Para cada pacote recebido, é enviado de volta ao servidor um *acknowledgment* de sucesso de envio. Apesar de tudo, este processo é bastante complexo, exigindo a troca de várias tramas durante a transferência de dados, o que acaba por exigir mais da rede. Quando estamos perante uma rede mais fraca, é provável que se percam pacotes. Também pode ser necessária a retransmissão dos mesmos. Isto pode causar mais sobrecarga na rede, assim como atrasos na transmissão, que é notável com o *delay* das aplicações. Com as experiências realizadas conseguimos perceber o quão mais complexo o TCP realmente é, pela quantidade de campos e informações presentes em cada trama.

Ao contrário do TCP, o UDP é um protocolo de transporte não fiável, o que o torna de certa forma mais eficiente por ser de uma complexidade menor. Neste caso, é a aplicação que tem de assegurar que os dados são enviados/recebidos com sucesso, o que pode complicar o lado da programação da própria aplicação. Por ser um protocolo mais simples, não sobrecarrega a rede, o que o torna preferível para algumas aplicações, nomeadamente aplicações relacionadas com *streaming*. Conseguimos observar isto nas experiências realizadas através dos *timestamps*, que nos mostraram uma melhor *performance* em termos de tempo de transferência.