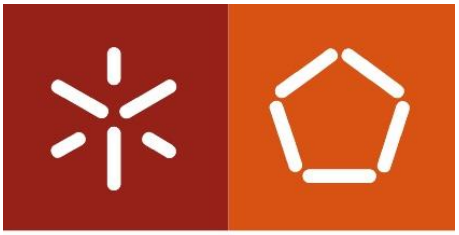

Universidade do Minho

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

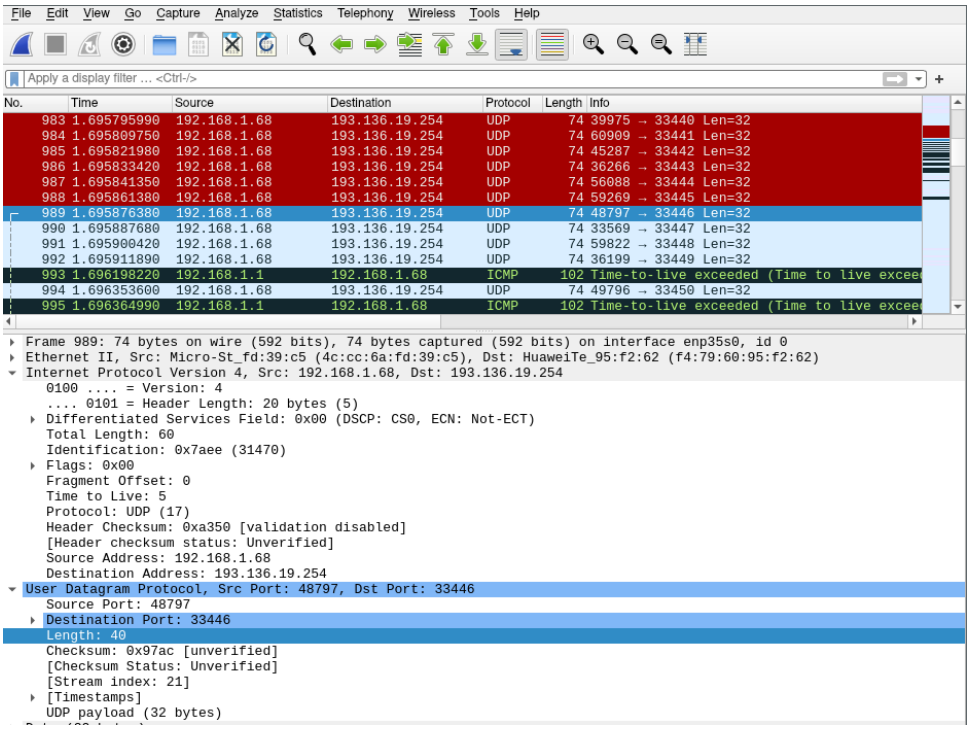

Figura 1.1: Traceroute.

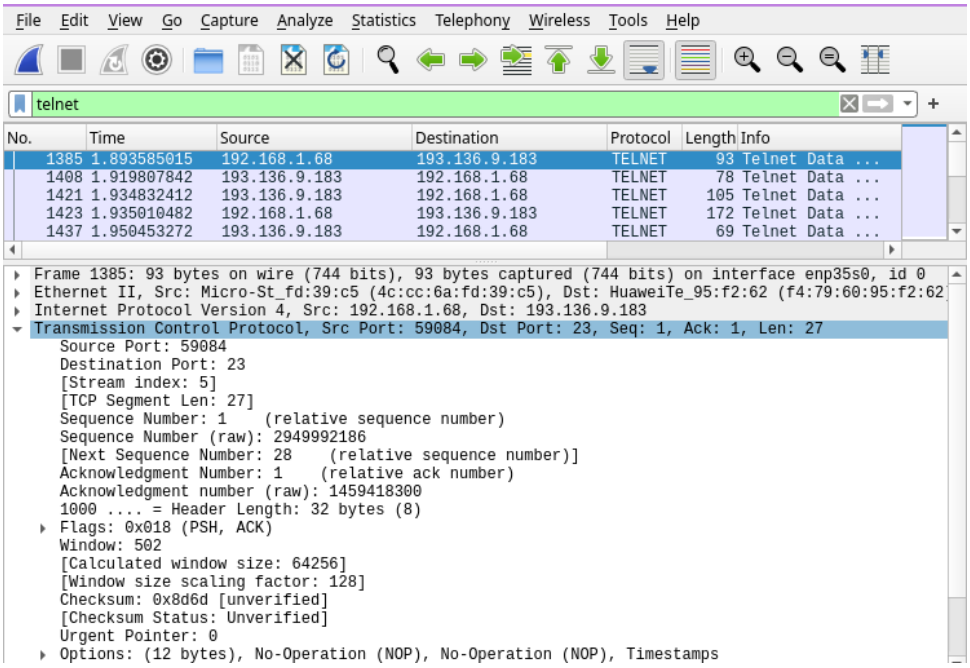

Figura 1.2: Telnet.

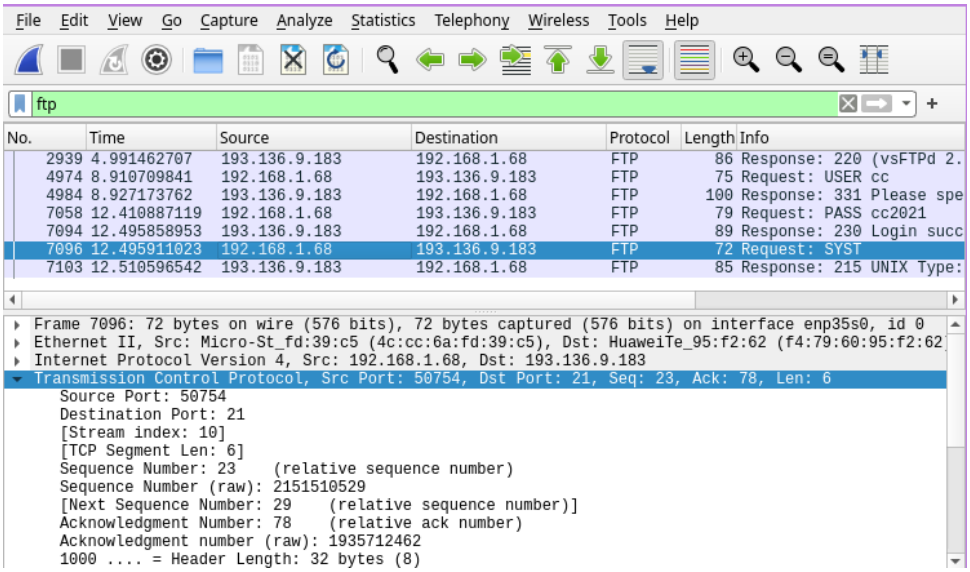

Figura 1.3: FTP.

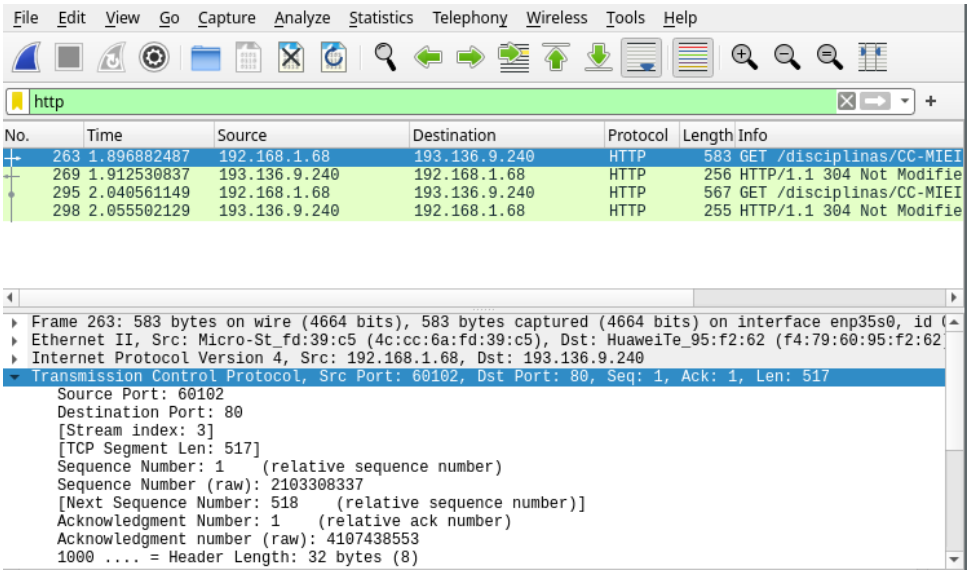

Figura 1.4: HTTP.

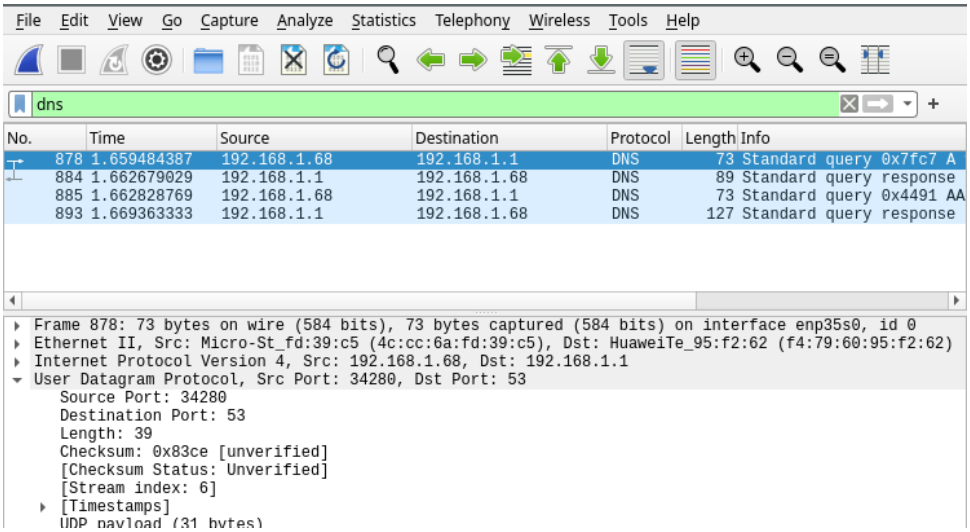

Figura 1.5: Nslookup.

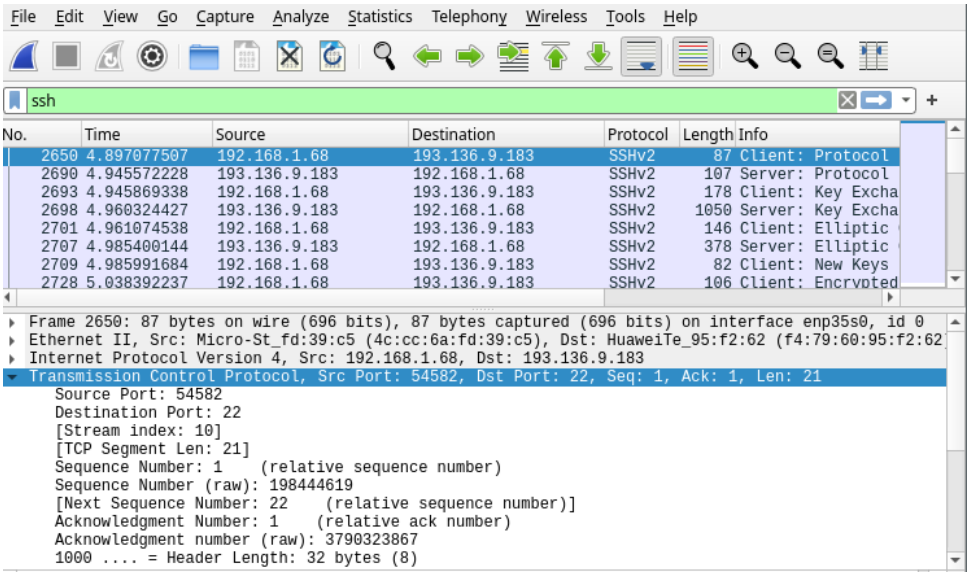

Figura 1.6: Ssh.

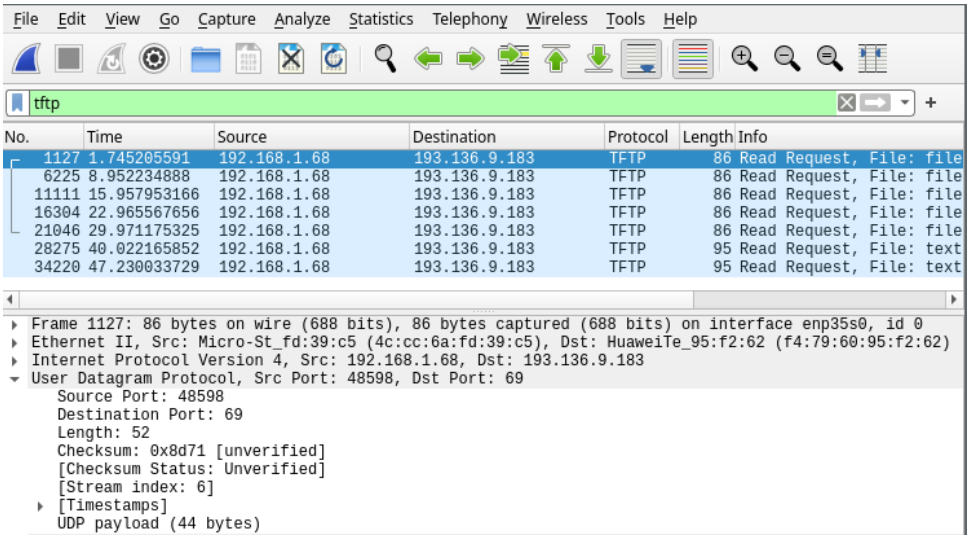

Figura 1.7: TFTP.

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 quer nos dados como nas confirmações.

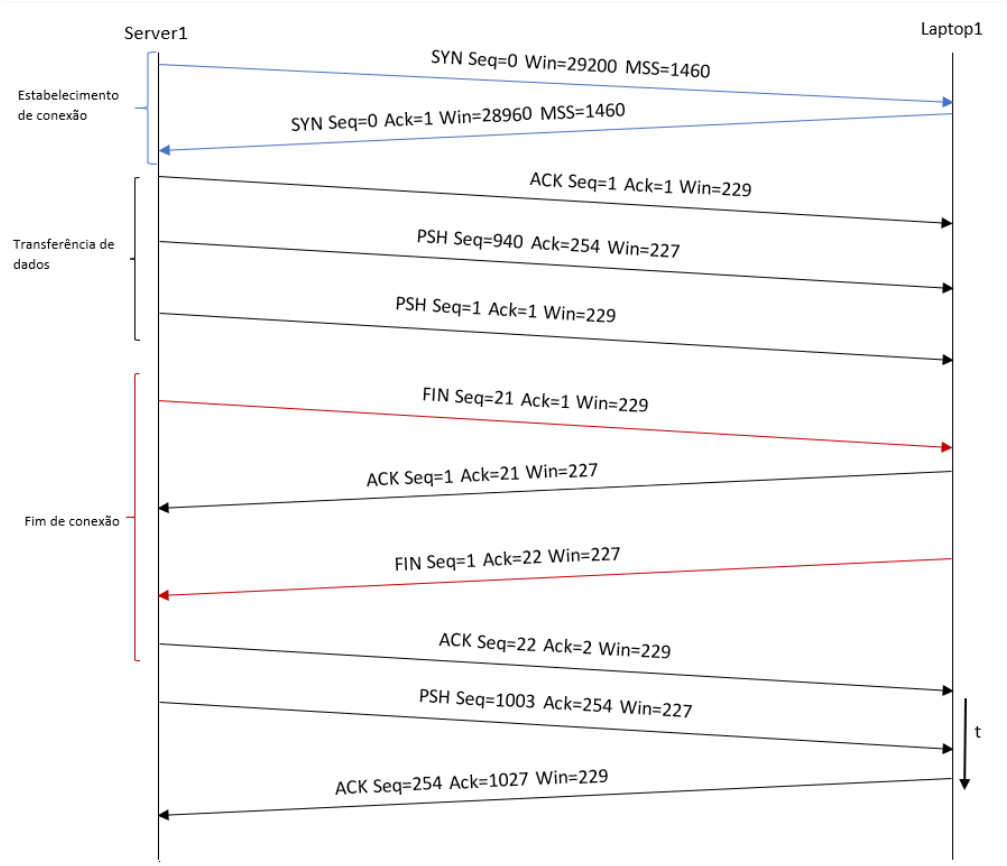

Figura 1.8: Esquema da transferência do file 1 por FTP

```
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
202 261.902080405 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

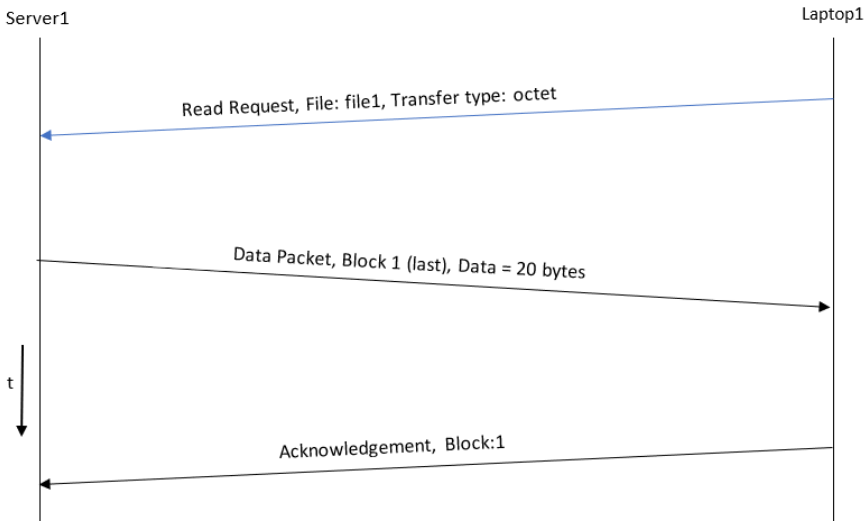

Figura 1.10: Esquema da transferência do file 1 por TFTP

```
    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 potocolo 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.

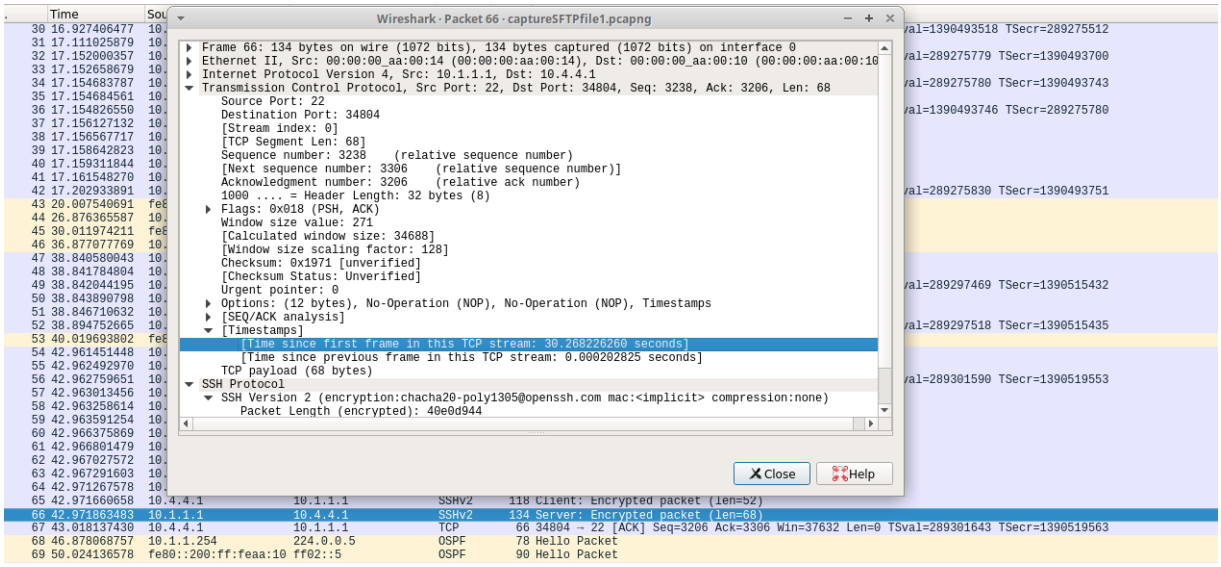

Figura 1.12: Tempo de transferência do ficheiro 1 por SFTP

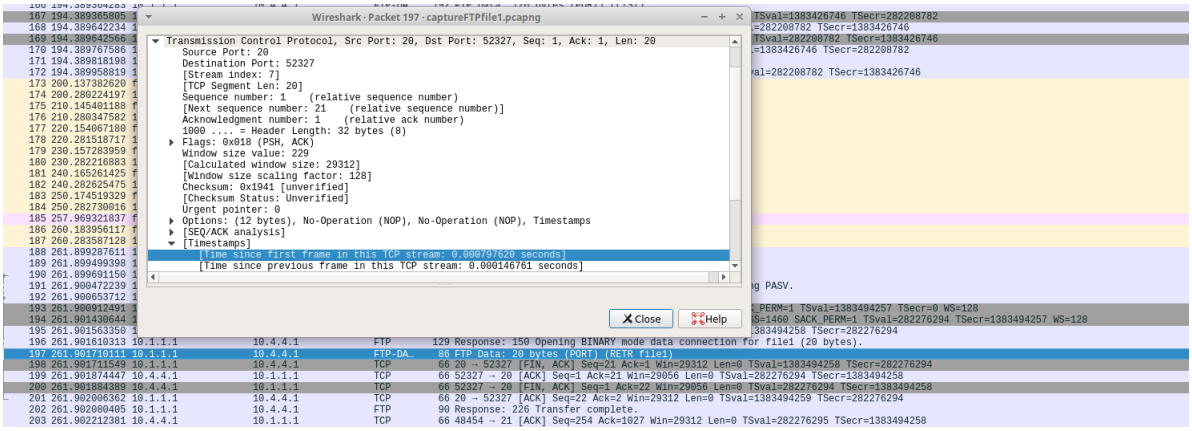

Figura 1.13: Tempo de transferência do ficheiro 1 por FTP

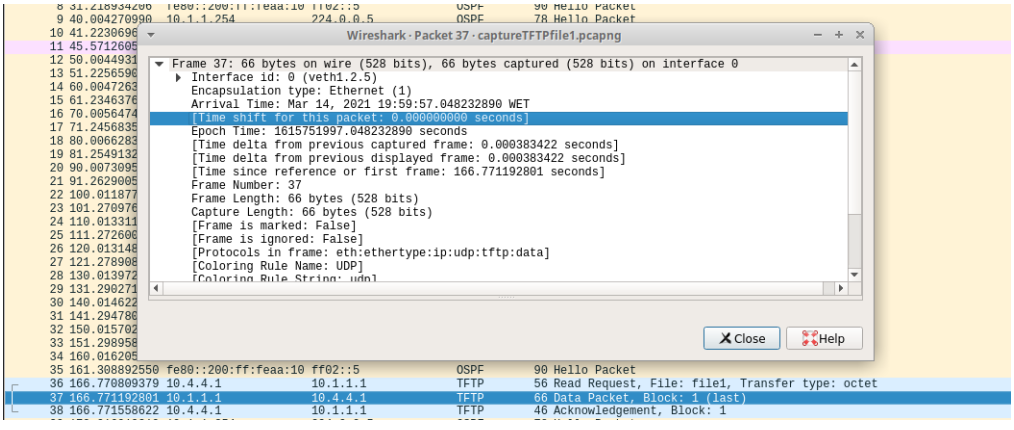

Figura 1.14: Tempo de transferência do ficheiro 1 por TFTP

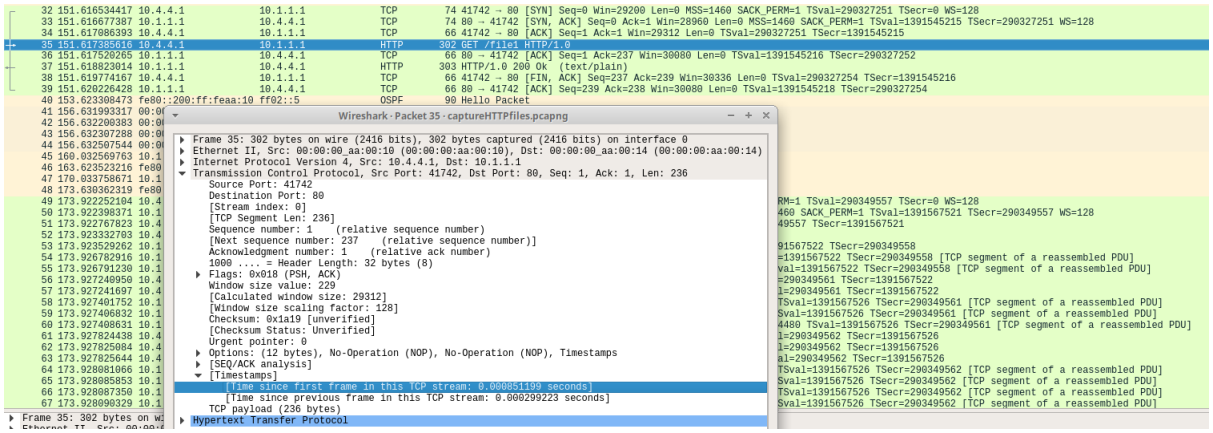

Figura 1.15: Tempo de transferência do ficheiro 1 por HTTP

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.