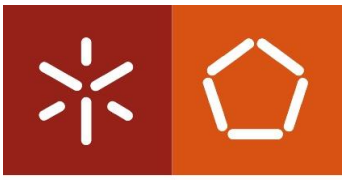

Universidade do Minho

Departamento de Informática

Comunicações por Computador
TP1:Protocolos da Camada de Transporte

Grupo Nº 3 PL6

Gonçalo Almeida (A84610)

Emanuel Rodrigues (A84776)

Lázaro Pinheiro (A86788)

4 de Março de 2020

Conteúdo

Capítulo 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	PING	-	-	-
traceroute	TRACEROUTE	UDP	33452	33.33%
telnet	TELNET	TCP	23	29.85%
ftp	FTP	TCP	21	30.30%
Tftp	TFTP	UDP	69	47.62%
browser/http	HTTP	TCP	80	10.42%
nslookup	DNS	UDP	53	28.57%
ssh	SSHv2	TCP	22	21.51%

PING

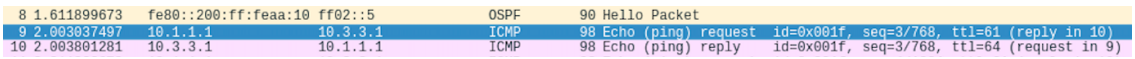

Figura 1.1: Tráfego de pacotes

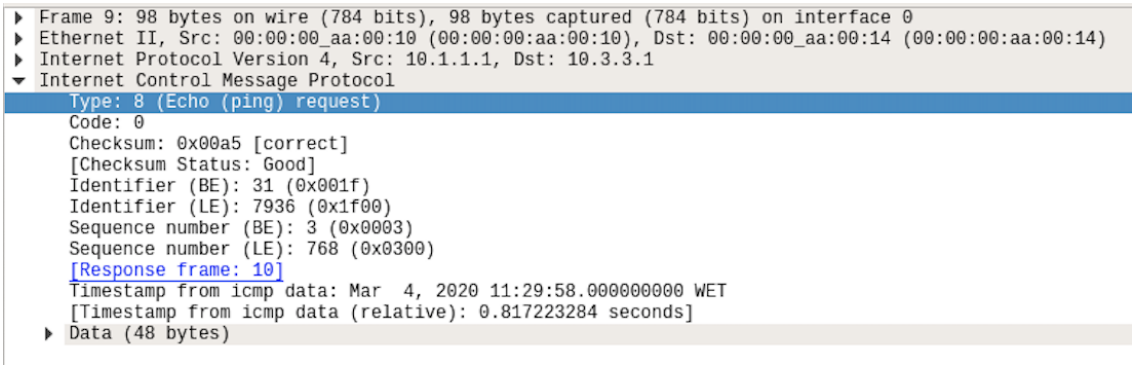

Figura 1.2: Trama ping

Este comando utiliza o protocolo de aplicação PING. Como trabalha na camada da rede, não

se aplica o protocolo de transporte, o que implica a não existência de uma porta de atendimento e de um Overhead de transporte.

TRACEROUTE

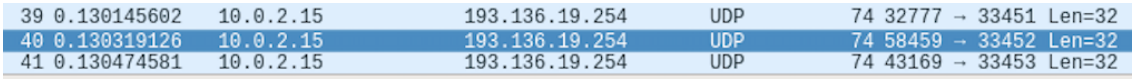

Figura 1.3: Tráfego de pacotes

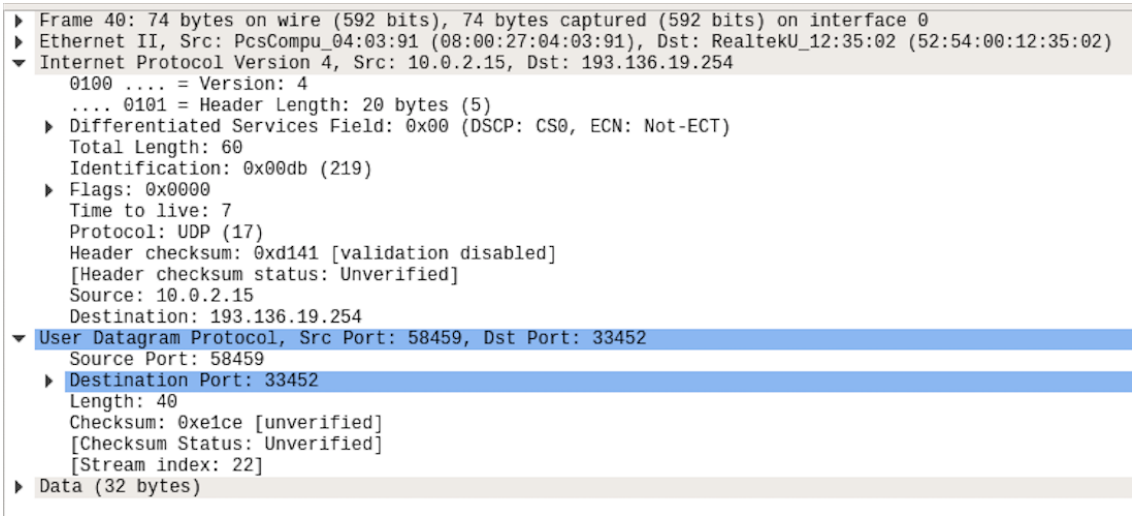

Figura 1.4: Trama traceroute

Este comando utiliza o protocolo de aplicação TRACEROUTE. Podemos verificar que o protocolo de transporte é o UDP (olhando para o campo Protocol), a porta de atendimento é a 33452 (olhando para o campo DST Port) e o Overhead de transporte = Header Length / Total Length = 20 / 60 = 0.3333.

TELNET

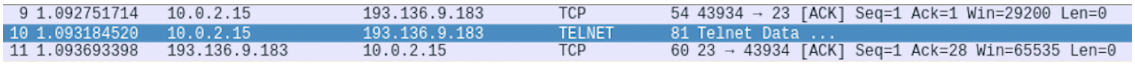

Figura 1.5: Tráfego de pacotes

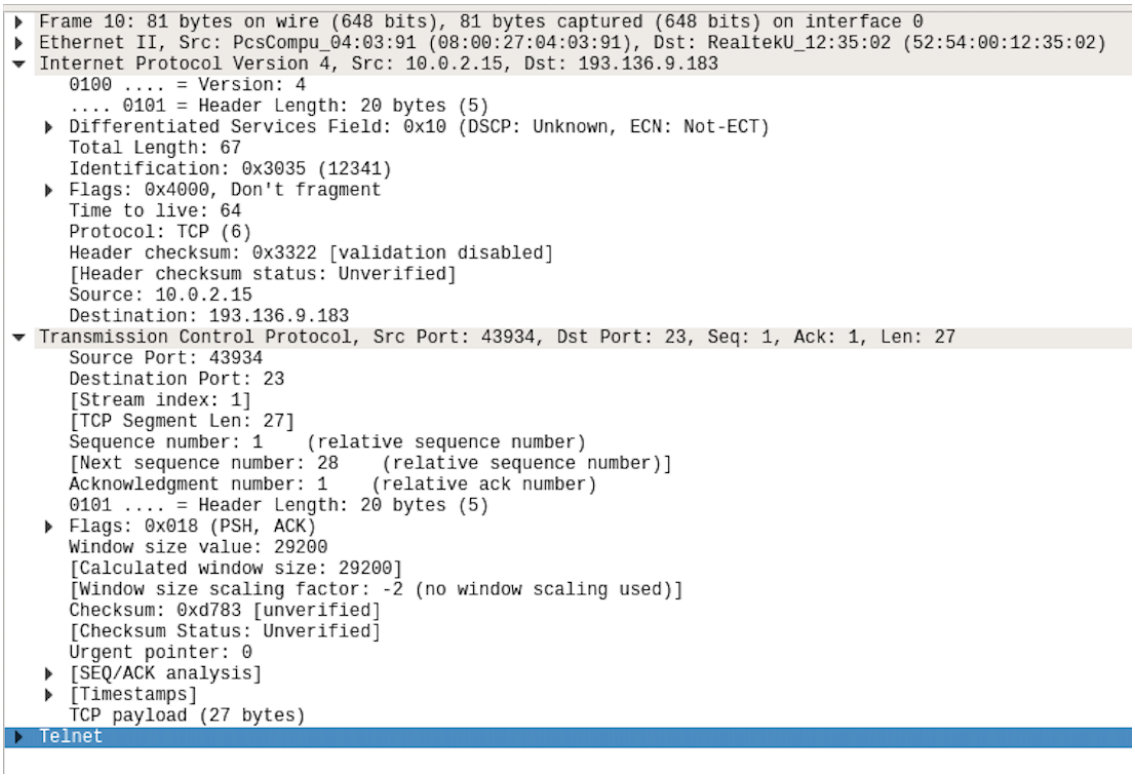

Figura 1.6: Trama telnet

Este comando utiliza o protocolo de aplicação TELNET. Podemos verificar que o protocolo de transporte é o TCP (olhando para o campo Protocol), a porta de atendimento é a 23 (olhando para o campo DST Port) e o Overhead de transporte = Header Length / Total Length = 20/67 = 0.2985.

FTP

Figura 1.7: Tráfego de pacotes

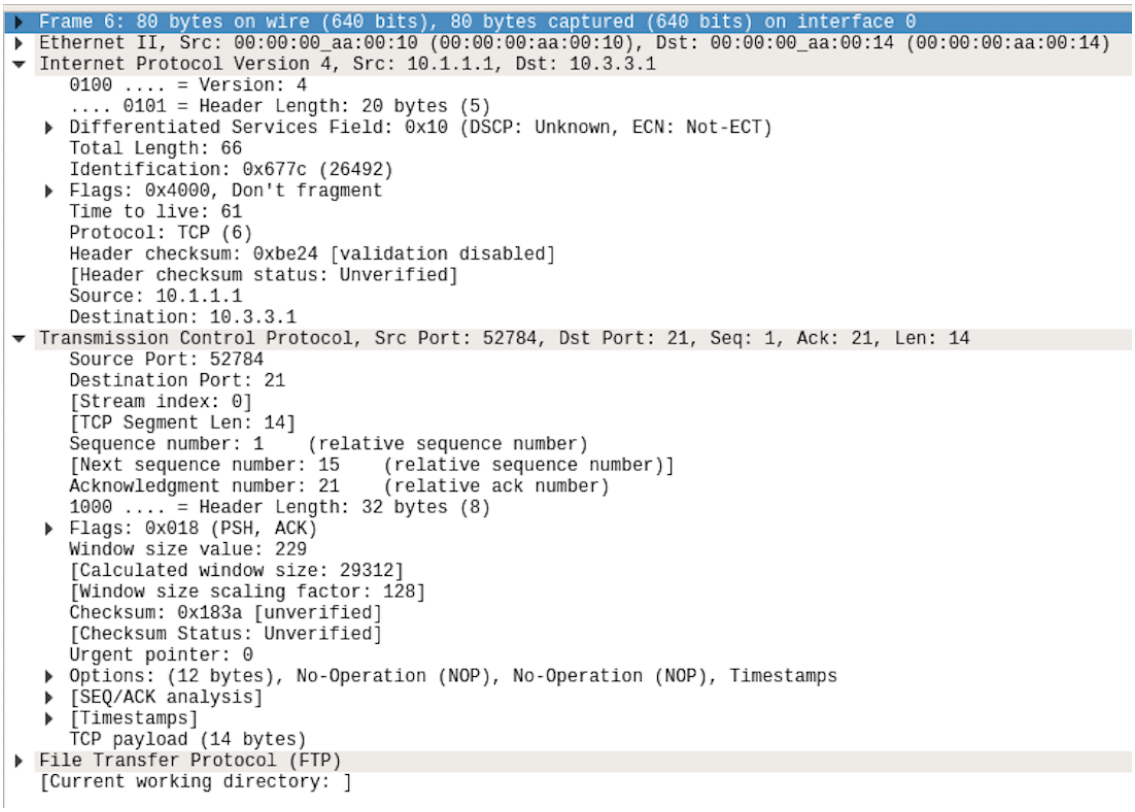

Figura 1.8: Trama ftp

Este comando utiliza o protocolo de aplicação FTP. Podemos verificar que o protocolo de transporte é o TCP (olhando para o campo Protocol), a porta de atendimento é a 21 (olhando para o campo DST Port) e o Overhead de transporte = Header Length / Total Length = 20 / 66 = 0.3030.

TFTP

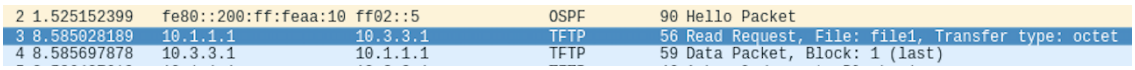

Figura 1.9: Tráfego de pacotes

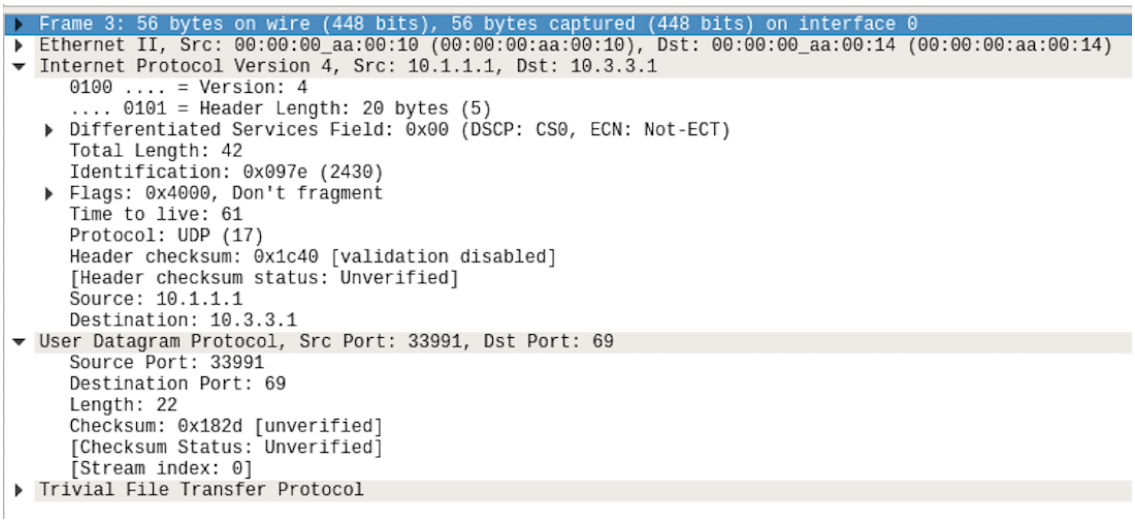

Figura 1.10: Trama tftp

Este comando utiliza o protocolo de aplicação TFTP. Podemos verificar que o protocolo de transporte é o UDP (olhando para o campo Protocol), a porta de atendimento é a 69 (olhando para o campo DST Port) e o Overhead de transporte = Header Length / Total Length = 20 / 42 = 0.4762.

BROWSER/HTTP

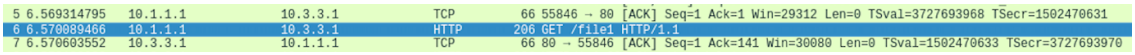

Figura 1.11: Tráfego de pacotes

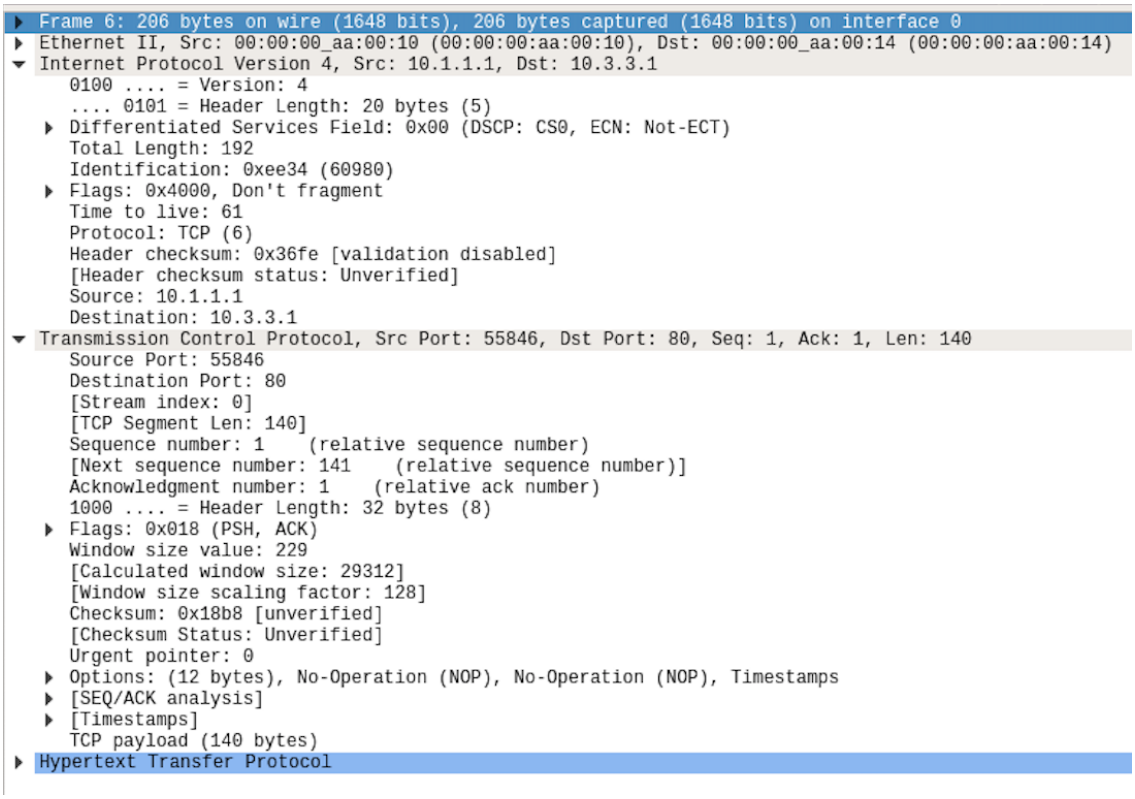

Figura 1.12: Trama browser/http

Este comando utiliza o protocolo de aplicação HTTP. Podemos verificar que o protocolo de transporte é o TCP (olhando para o campo Protocol), a porta de atendimento é a 80 (olhando para o campo DST Port) e o Overhead de transporte = Header Length / Total Length = 20 / 192 = 0.1042.

NSLOOKUP

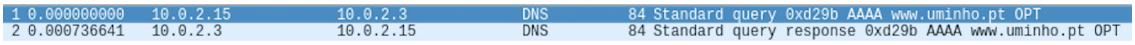

Figura 1.13: Tráfego de pacotes

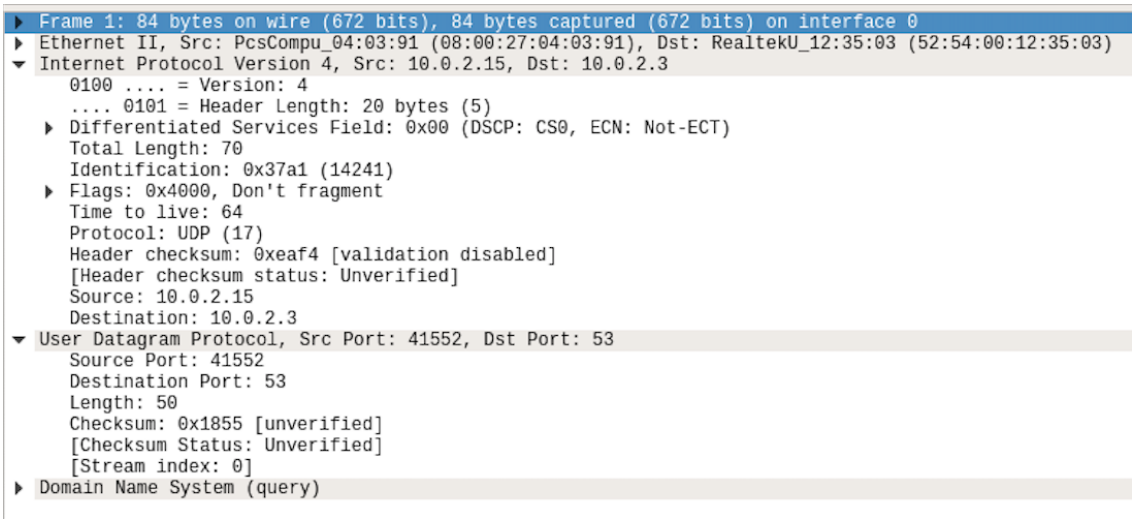

Figura 1.14: Trama nslookup

Este comando utiliza o protocolo de aplicação DNS. Podemos verificar que o protocolo de transporte é o UDP (olhando para o campo Protocol), a porta de atendimento é a 53 (olhando para o campo DST Port) e o Overhead de transporte = Header Length / Total Length = 20 / 70 = 0.2857.

SSH

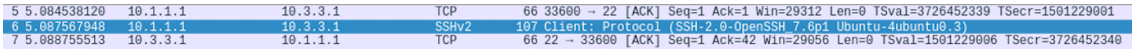

Figura 1.15: Tráfego de pacotes

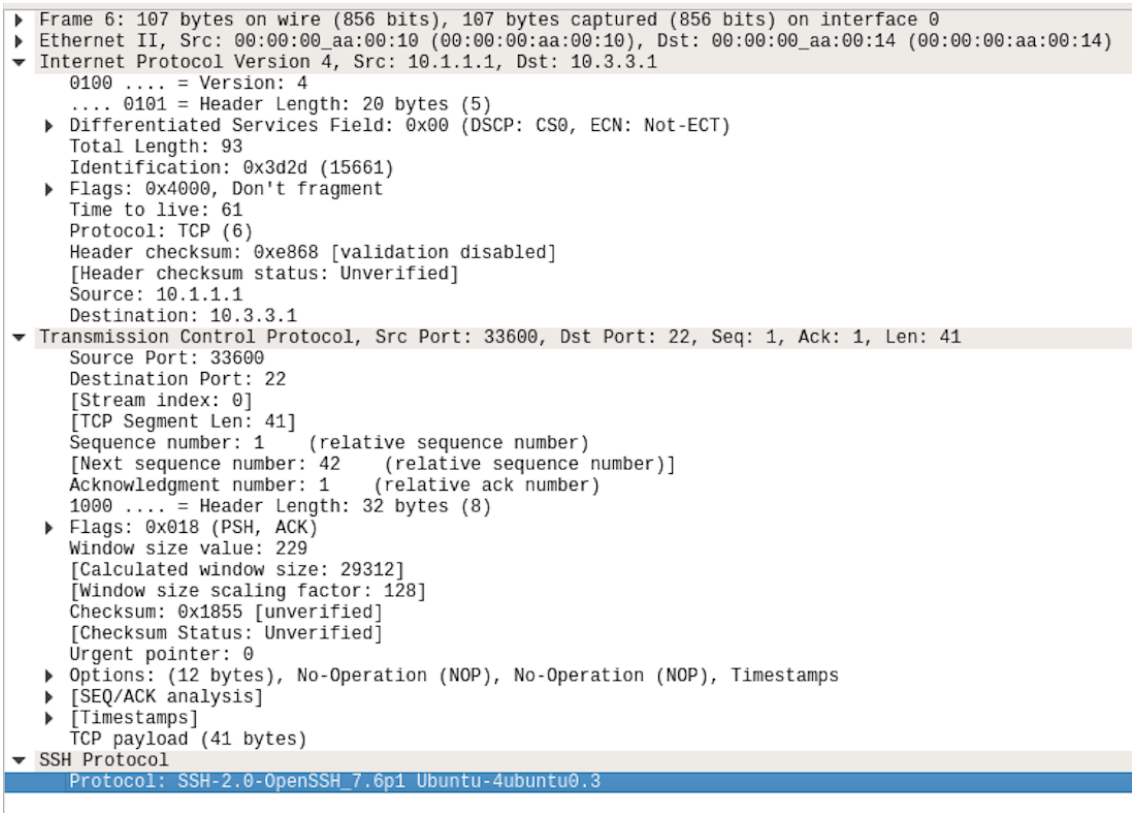

Figura 1.16: Trama ssh

Este comando utiliza o protocolo de aplicação SSHv2. Podemos verificar que o protocolo de transporte é o TCP (olhando para o campo Protocol), a porta de atendimento é a 22 (olhando para o campo DST Port) e o Overhead de transporte = Header Length / Total Length = 20 / 93 = 21.51.

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.

FTP

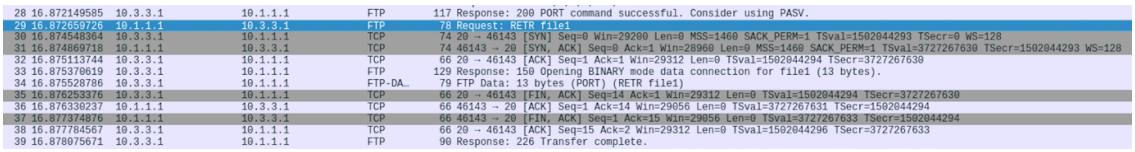

Figura 1.17: Tráfego de pacotes

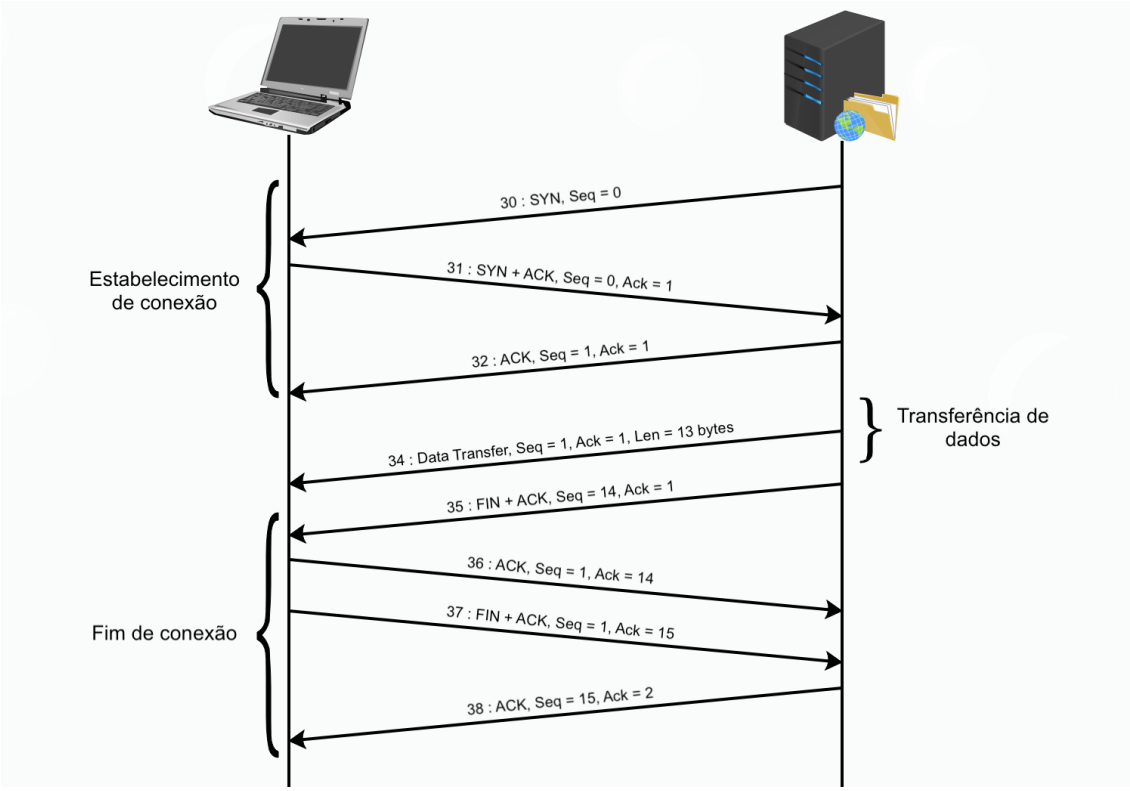

Figura 1.18: Protocolo de controlo de transmissão FTP

Na fase de estabelecimento de conexão, a conexão começa com o envio de um pacote SYN (indica que os números de sequência devem ser sincronizados para se iniciar uma conexão) por parte do servidor com número de sequência igual a 0, ao qual o utilizador responde com um SYN+ACK. Posteriormente, o servidor envia um ACK (indica se o número de sequência de confirmação é válido) de volta ao utilizador.

De seguida, ocorre a fase de transferência de Dados, onde o servidor envia os dados solicitados ao utilizador.

No final do envio, o servidor inicia a fase de fim de conexão (dupla terminação com 4 segmentos). Note-se que, o servidor após emitir o pacote FIN (indica que terminou o envio de dados), fica à espera de um pacote ACK, para poder terminar a conexão com sucesso.

TFTP

```
2 1.525152399   fe80::200:ff:feaa:10 ff02::5              OSPF        90 Hello Packet
3 8.585028189   10.1.1.1              10.3.3.1            TFTP        56 Read Request, File: file1, Transfer type: octet
4 8.585697878   10.3.3.1              10.1.1.1            TFTP        59 Data Packet, Block: 1 (last)
5 8.586427619   10.1.1.1              10.3.3.1            TFTP        46 Acknowledgement, Block: 1
6 10.001038434  10.3.3.254            224.0.0.5           OSPF        78 Hello Packet
```

Figura 1.19: Tráfego de pacotes

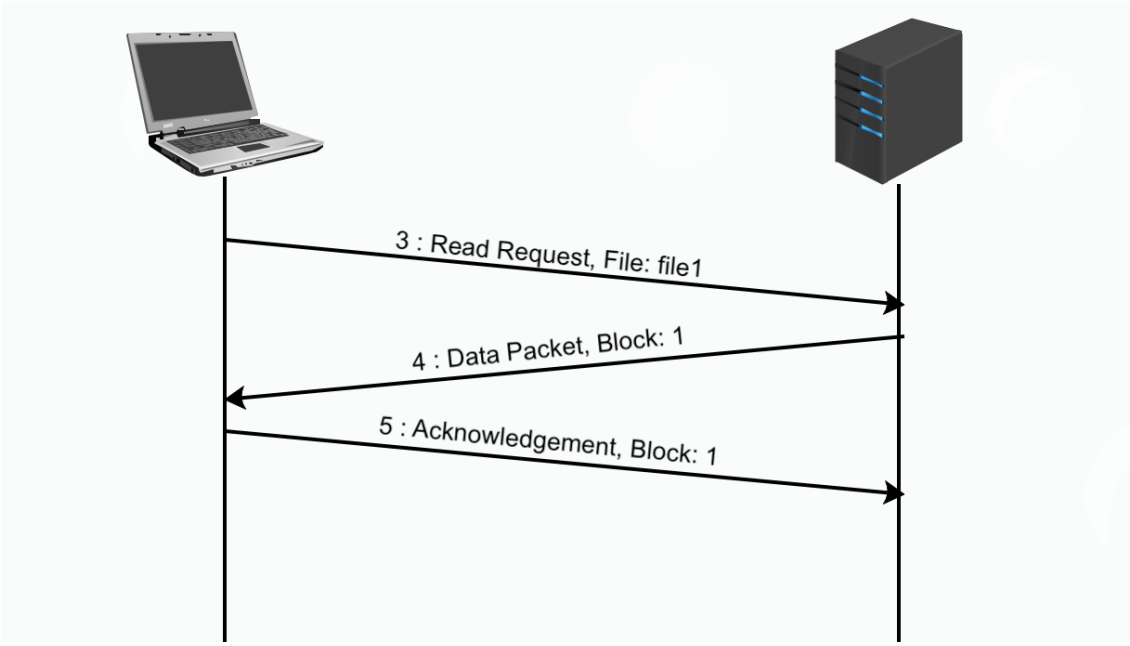

Figura 1.20: Protocolo de controlo de transmissão TFTP

O utilizador envia um Read Request para o servidor, o qual contém o nome do ficheiro, ao qual o servidor responde com um pacote de dados. O utilizador após receber os dados envia um pacote ACK.

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;

	SFTP	FTP	TFTP	HTTP
Uso da camada de transporte	Protocolo TCP	Protocolo TCP	Protocolo UDP	Protocolo TCP
Eficiência na transferência	A eficiência depende da fiabilidade da transferência de dados	Eficiente, visto que é fiável(maior overhead)	Pouco eficiente, pois não se responsabiliza pela entregua dos dados(menor overhead)	Muito eficiente
Complexidade	Muito complexo, pois disponibiliza inúmeras funcionalidades	Complexo, pois existe segurança na transferência de dados	Versão simplificada do FTP, uma vez que o protocolo de transporte é o UDP, disponivel menos funcionalidades do que o FTP	Pouco complexo
Segurança	Seguro, pois recorre à autenticação e existe encriptação dos dados	Pouco Seguro, apesar de utilizar autenticação	Pouco seguro, uma vez que não existe autenticação nem encriptação de dados	Pouco seguro, apesar de recorrerà autenticação, não utiliza encriptação dos dados

Figura 1.21: Distinção das aplicações SFTP, FTP, TFTP e HTTP

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).

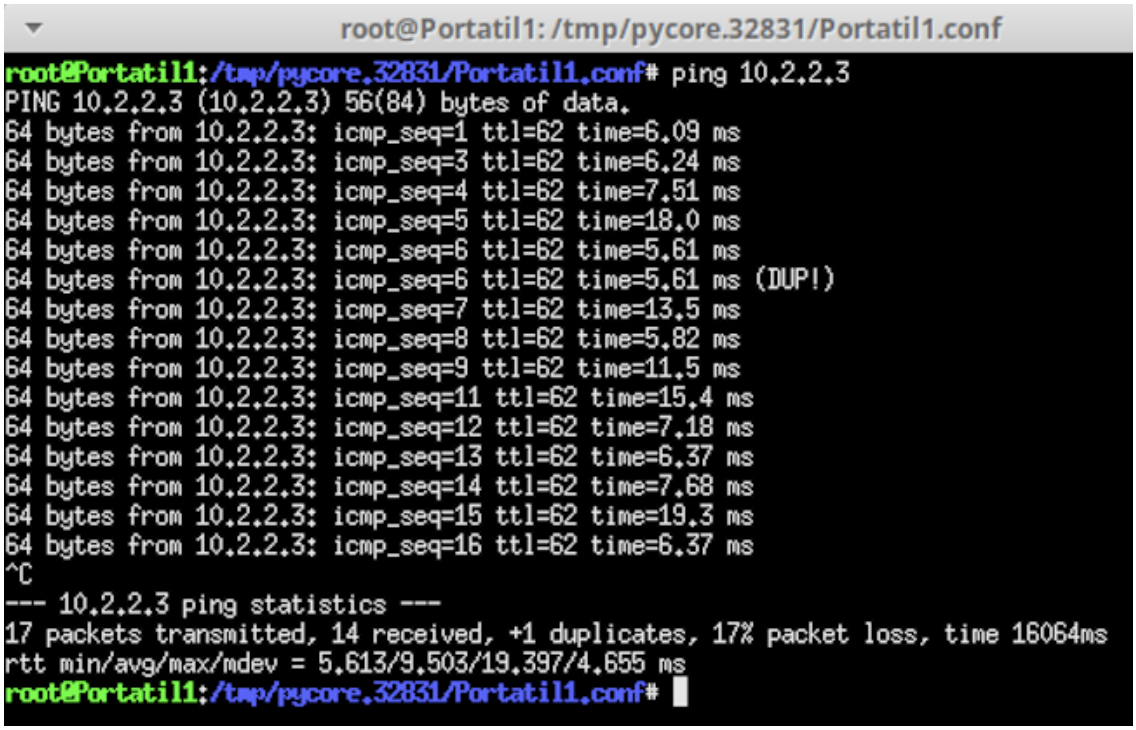

Figura 1.22: Comando ping

Observando a figura 23, na execução do comando ping ao servidor 1 no portátil 1 vemos que, dos 17 pacotes transmitidos, houve uma perda de 17% dos pacotes e 1 foi duplicado.

Os protocolos de transporte TCP e UDP resolvem os problemas no nível da ligação de rede de maneiras diferentes. O protocolo TCP é o mais usado dos dois, pois garante a entrega de todos os pacotes, e caso haja perda de um pacote, este é retransmitido. Esta garantia é obtida através do método Three Way Handshake (SYN, SYN-ACK, ACK).

Quanto ao protocolo UDP, é mais simples que o anterior, sendo que não é capaz de detetar e recuperar perdas de pacotes. A garantia da entrega dos pacotes fica a cargo das camadas superiores como a aplicação em si. Relativamente ao portátil 1 da figura, a recuperação dos pacotes perdidos pode ser feita através do reenvio destes no caso do TCP.

Capítulo 2

Conclusão

Com a realização deste primeiro trabalho prático aprofundamos o nosso conhecimento sobre as camadas inferiores na utilização de diversas aplicações. Começamos por analisar protocolos de aplicação e de transporte, as portas que estes utilizam e os overheads associados através da análise de tráfego de pacotes e tramas. De seguida comparamos quatro serviços distintos (SFTP, FTP, TFTP e HTTP) na transferência de ficheiros. Por fim analisamos as diferenças entre os protocolos TCP e UDP relativamente à entrega de pacotes. A realização deste trabalho foi de extrema importância para a melhor compreensão da camada de transporte.