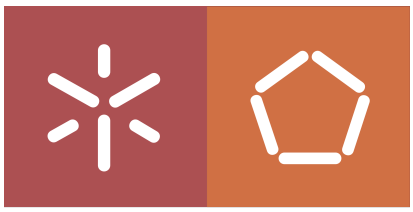

Universidade do Minho
Escola de Engenharia

Comunicações por Computador

Relatório do Trabalho Prático n.º 1
PL5 Grupo 2

LEI - 3º Ano - 1º Semestre

Realizador por:

A98695	Lucas Oliveira
A89292	Mike Pinto
A96208	Rafael Gomes

Braga,
28 de março de 2024

Conteúdo

Lista de Figuras

Capítulo 1

Parte I - Questões e Respostas

1.1 Questão 1

De que forma as perdas e duplicações de pacotes afetaram o desempenho das aplicações? Que camada lidou com as perdas e duplicações: transporte ou aplicação? Responda com base nas experiências feitas e nos resultados observados.

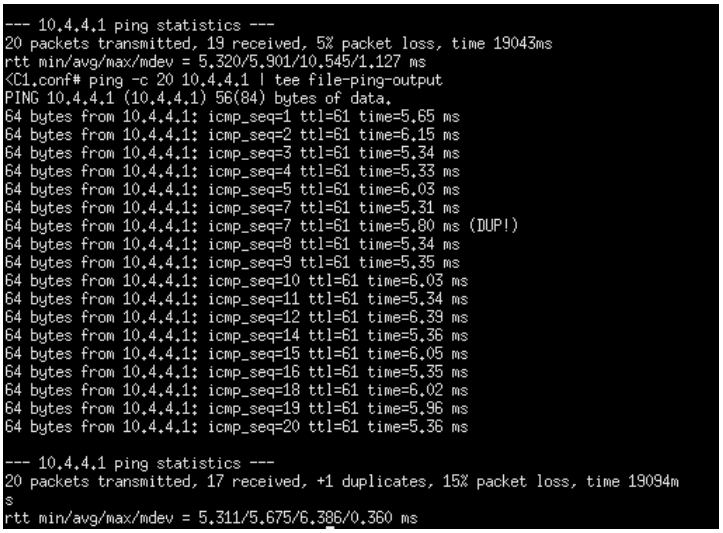

Figura 1.1.1: *Ping* realizado a partir do *PC1* para o *Servidor1*.

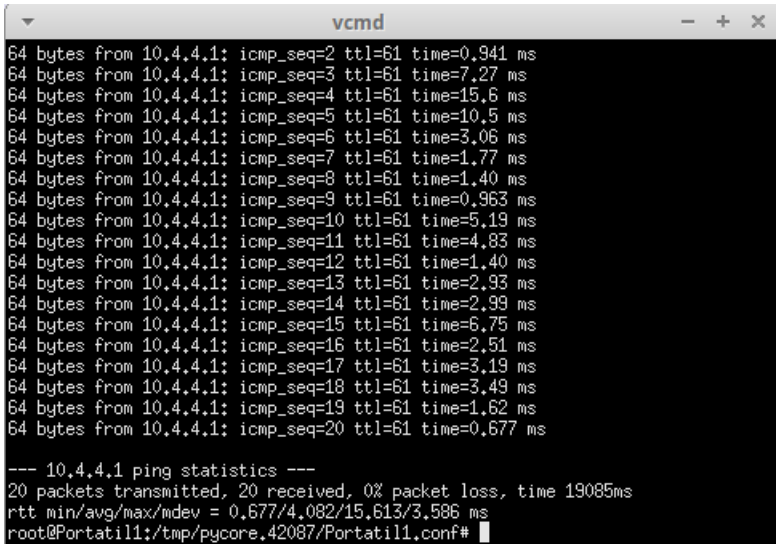

Figura 1.1.2: *Ping* realizado a partir do *Portatil1* para o *Servidor1*

Visando testar a conetividade entre os *hosts PC1 e Portatil1* foi executado o comando *Ping* com destino ao *Servidor1* como podemos observar nas figuras 1.1.1 e 1.1.2. Para ambos os casos foram enviados 20 pacotes, tendo todos os pacotes do *Portatil1* sido entregues. Contudo, apenas 17 dos 20 pacotes enviados pelo *PC1* foram entregues ao *Servidor1* havendo assim 15% de *packet loss* e um pacote duplicado.

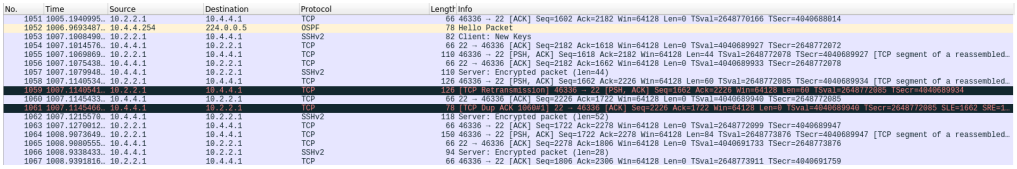

Figura 1.1.3: Captura de pacotes, através do *WireShark* na ligação *SFTP* do *PC1* para o *Servidor1*

A camada responsável pela perda e/ou duplicação de pacotes é a camada de transporte mediante o uso de protocolos fiáveis como o *TCP*, orientado à ligação, uma vez que garante que todos os pacotes são entregues ao destino e na ordem correta, pois para todos os pacotes recebidos é enviado um pacote *ACK* (*Acknowledgment*). Contudo, redes de pior qualidade, como a rede do *PC1*, são mais suscetíveis a uma perda e/ou duplicação de pacotes, o que obriga ao envido de mensagens de erro, como podemos observar nas tramas n.º 1059 e 1061 da figura 1.1.3, o que poderá levar a uma sobrecarga na rede e eventualmente atrasos na aplicação.

Recorrendo ao protocolo *UDP*, apesar deste ser mais "rápido" e de fácil utilização, poderá ser a própria aplicação responsável pela garantia de entrega dos pacotes no caso destes não serem entregues, visto o protocolo de transporte *UDP* não ser orientado à ligação, não possuindo capacidade de verificação de entrega de pacotes, o poderá levar a um eventual atraso na aplicação.

1.2 Questão 2

Obtenha a partir do wireshark, ou desenhe manualmente, um diagrama temporal para a transferência de file1 por FTP. Foque-se apenas na transferência de dados [ftp-data] e não na conexão de controlo, pois o FTP usa mais que uma conexão em simultâneo. Identifique, se aplicável, as fases de início de conexão, transferência de dados e fim de conexão. Identifique também os tipos de segmentos trocados e os números de sequência usados quer nos dados como nas confirmações.

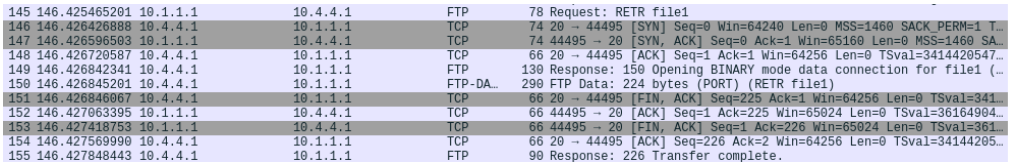

Figura 1.2.1: Captura de pacotes, através do *WireShark* na ligação *FTP* do *Portátil1* para o *Servidor1*

Através da imagem 1.2.1 verificamos que o servidor realiza um pedido de conexão com o cliente ao enviar um pacote *TCP* com a *flag [SYN]*. O cliente após receber o pacote de pedido de conexão responde com um pacote com as *flags [SYN,ACK]* estabelecendo-se assim a conexão. Foi efetuada a transferência de dados, como podemos observar pela trama n.º 150. Por fim o servidor inicia o fim da conexão enviando um pacote *TCP* com a *flag [FIN,ACK]*, seguidamente o cliente ao receber este responde com um pacote com as *flags [FIN, ACK]* como consta na trama n.º 153.

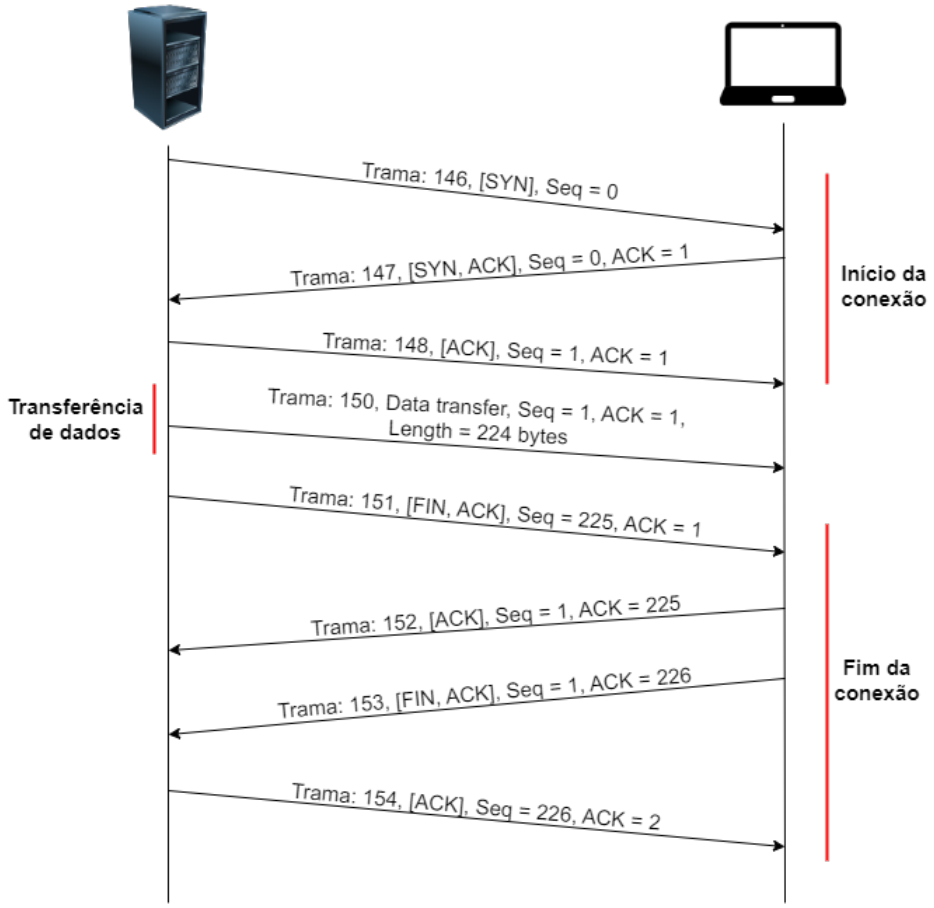

Figura 1.2.2: Diagrama temporal da aplicação *FTP*, com a transferência do *file1*

1.3 Questão 3

Obtenha a partir do wireshark, ou desenhe manualmente, um diagrama temporal para a transferência de file1 por TFTP. Identifique, se aplicável, as fases de início de conexão, transferência de dados e fim de conexão. Identifique também os tipos de segmentos trocados e os números de sequência usados quer nos dados como nas confirmações.

100	134.11747…	10.1.1.1	10.4.4.1	TFTP	56 Read Request, File: file1, Transfer type: octet
101	134.12221…	10.4.4.1	10.1.1.1	TFTP	270 Data Packet, Block: 1 (last)
102	134.12285…	10.1.1.1	10.4.4.1	TFTP	46 Acknowledgement, Block: 1

Figura 1.3.1: Captura de pacotes, através do *WireShark* na ligação *TFTP* do *Portátil1* para o *Servidor1*

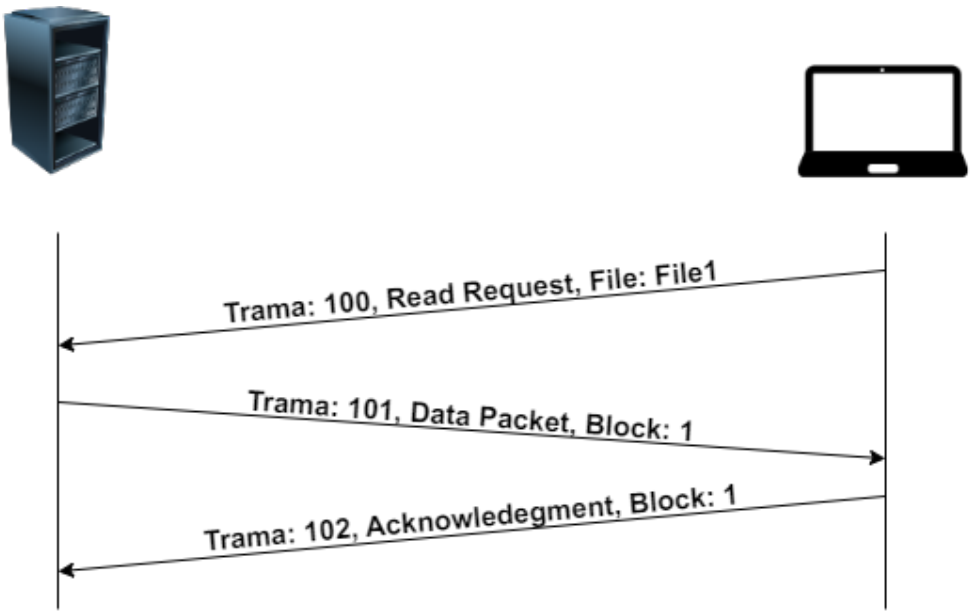

Figura 1.3.2: Diagrama temporal da aplicação *TFTP*, com a transferência do *file1*

Através da imagem 1.3.2, podemos observar que o cliente envia um *Read Request* para o servidor, que em resposta envia um *Data Packet*(pacote de dados). Por fim, o utilizador envia um pacote de *ACK*, de modo a confirmar o sucesso da operação.

1.4 Questão 4

Compare sucintamente as quatro aplicações de transferência de ficheiros que usou nos seguintes pontos (i) uso da camada de transporte; (ii) eficiência; (iii) complexidade; (iv) segurança;

Tabela 1.1: Comparação do uso da camada de transporte, eficiência, complexidade e segurança de diferentes aplicações de transferência de ficheiros

Aplicações	**SFTP**	**FTP**	**TFTP**	**HTTP**
Camada de Transporte	Protocolo TCP	Protocolo TCP	Protocolo UDP	Protocolo TCP
Eficiência	Possui uma boa eficiência	Eficiência decente	É altamente eficiente, mas não se responsabiliza pela entrega de dados	Elevada eficiência
Complexidade	Elevada complexidade	Elevada complexidade	Simples e direto com baixa complexidade	Baixa complexidade
Segurança	Muito seguro, pois recorre ao uso de autenticação e codificação dos dados	Pouco seguro, apesar de possuir autenticação	Inseguro, pois não possui autenticação e codificação de dados	Pouco seguro

Capítulo 2

Parte II - Questões e Respostas

2.1 Questão 1

Com base no trabalho realizado, tanto na parte I como na parte II, identifique para cada aplicação executada, qual o protocolo de aplicação, o protocolo de transporte, porta de atendimento e overhead de transporte.

Tabela 2.1: Identificação de protocolos de aplicação, transporte, porta de atendimento e overhead de transporte por aplicação

Comando Usado	**Protocolo de Aplicação**	**Protocolo de Transporte**	**Porta de Atendimento**	**Overhead de Transporte em Bytes**
wget, lynx ou browser	HTTP	TCP	80	$(20/406) \times 100 \approx 4,93\%$
ssh,sftp	SSH	TCP	22	$(20/67) \times 100 \approx 29,85\%$
FTP	FTP	TCP	21 (controlo conexão) 20 (transferencia de dados)	$(20/60) \times 100 \approx 33,33\%$
tftp	TFTP	UDP	69	$(8/72) \times 100 \approx 11,11\%$
telnet	TELNET	TCP	23	$(20/67) \times 100 \approx 29,85\%$
nslookup ou dig	DNS	UDP	53	$(8/70) \times 100 \approx 11,43\%$
ping	-	-	-	0
traceroute	-	UDP	33434 (em particular)	$(8/60) \times 100 \approx 13,33\%$

2.1.1 HTTP/Brower

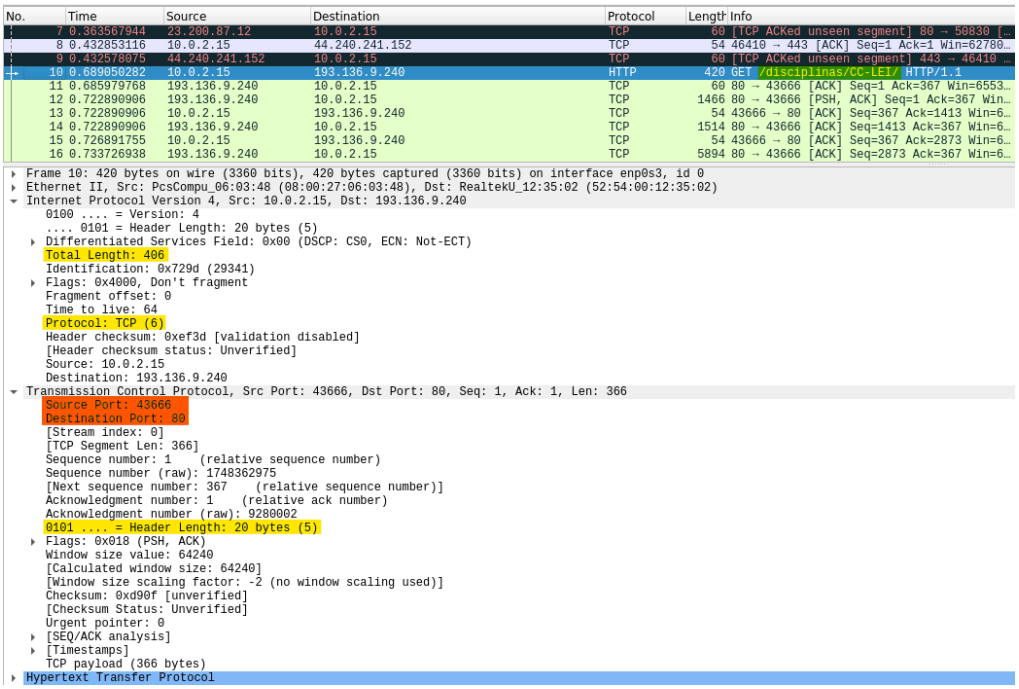

Figura 2.1.1: HTTP

Após a análise da captura de tráfego da figura 2.1.1 é possível verificar que o Protocolo de aplicação utilizado é o HTTP, o protocolo de transporte o TCP, a porta de atendimento é a 80. O tamanho total da trama é de 201 bytes com 20 bytes de header de transporte, tendo assim a trama um overhead de transporte de $(20/406) \times 100 \approx 4,92\%$

2.1.2 SSH

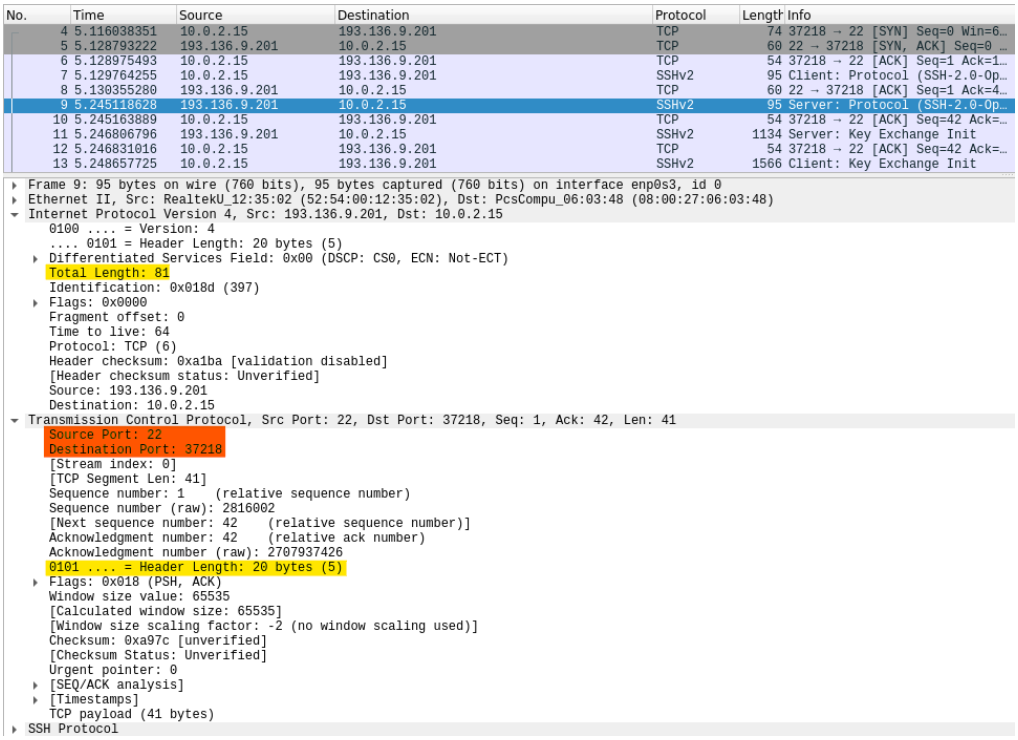

Figura 2.1.2: FTP

Pela figura 2.1.2, verificamos que o protocolo de aplicação utilizado é o SSH, o protocolo de transporte o TCP e a porta de atendimento a 22. O tamanho total da trama é de 67 bytes com 20 bytes de header de transporte, tendo assim a trama um overhead de transporte de $(20/67) \times 100 \approx 29,85\%$

2.1.3 FTP

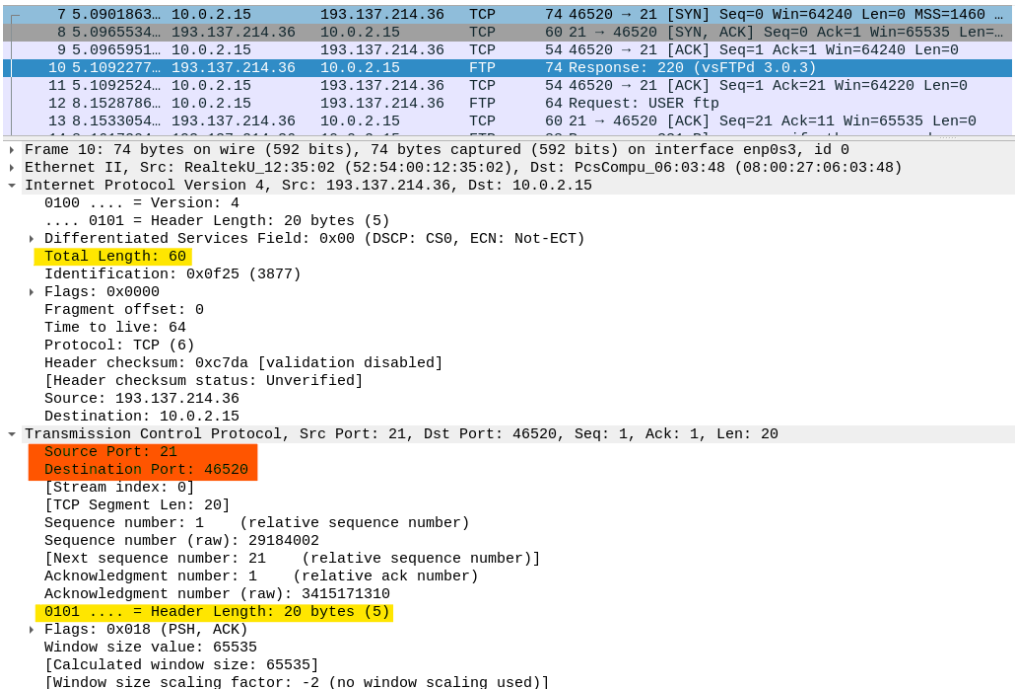

Figura 2.1.3: FTP

Como podemos observar pela figura 2.1.3 o protocolo de aplicação utilizado é o FTP, o de transporte o TCP e a porta de atendimento a porta 21(utilizada para controlo de conexão) e a porta 20(utilizada para transferência de dados). O tamanho do pacote é de 60 bytes com 20 bytes de header, tendo a trama um overhead de transporte de $(20/60) \times 100 \approx 33,33\%$.

2.1.4 TFTP

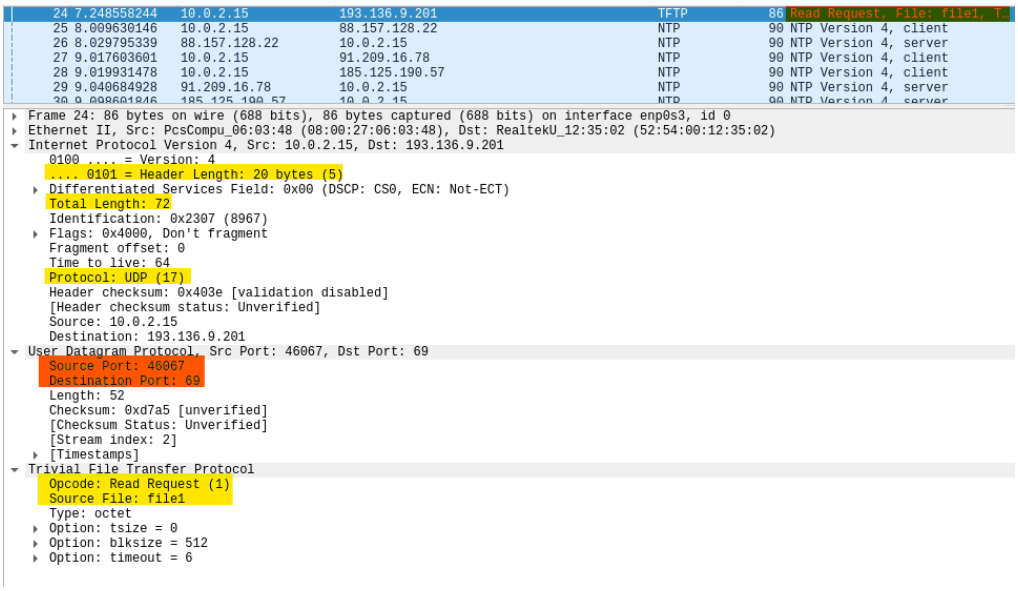

Figura 2.1.4: TFTP

Pela figura 2.1.4 verificamos que o protocolo de aplicação utilizado é o TFTP, o protoloco de transporte o UDP e a porta de atendimento a 69. O tamanho do pacote é de 72 bytes e um header UDP possui o tamanho de 8 bytes, tendo a trama um overhead de transporte de $(8/72) \times 100 \approx 11,11\%$.

2.1.5 Telnet

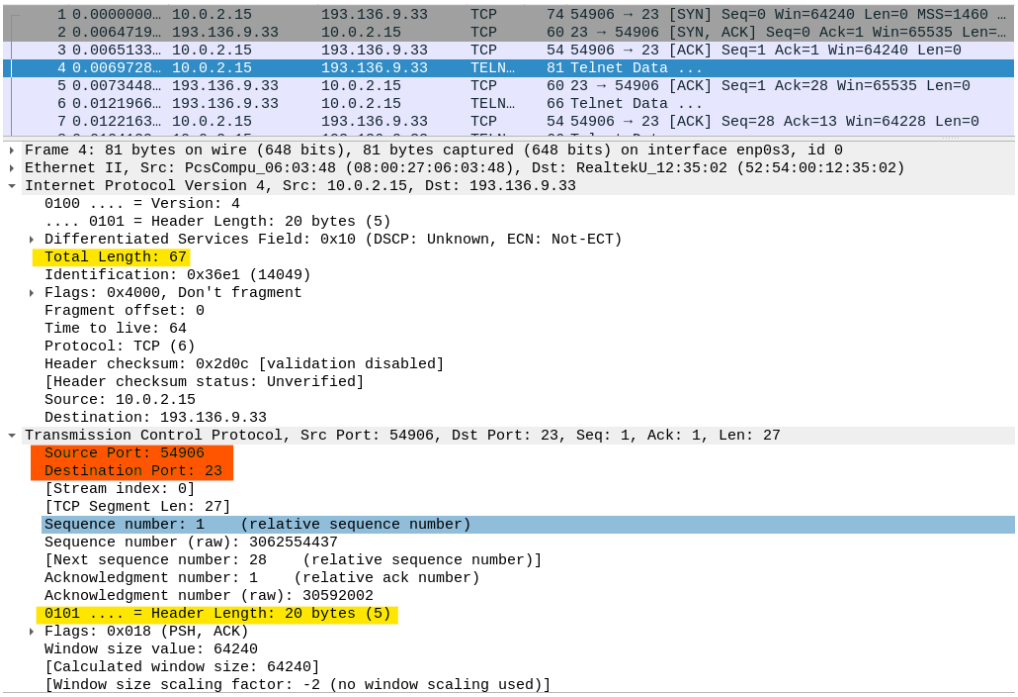

Figura 2.1.5: Telnet

Após a analise da captura da figura 2.1.5, verificamos que o protocolo de aplicação utilizado é o TELNET, o protocolo de transporte o TCP e a porta de atendimento a 23. O tamanho desta trama é de 67 bytes com um header de 20 bytes, tendo então um overhead de transporte de $(20/67) \times 100 \approx 29,85\%$

2.1.6 nslookup

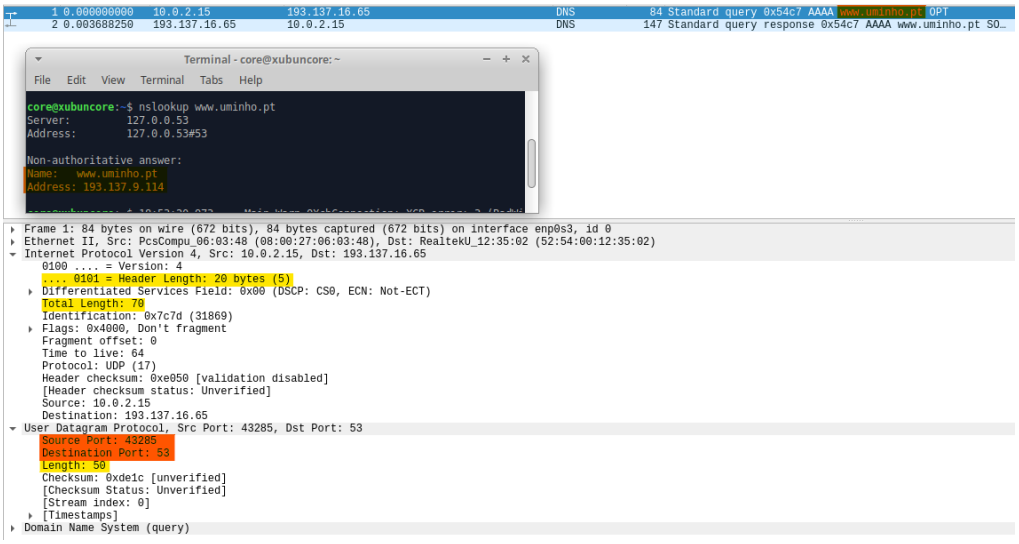

Figura 2.1.6: Nslookup

Pela figura 2.1.6 temos que o protocolo de aplicação utilizado é o DNS, o de transporte o UDP e a porta de atendimento a 53. Esta trama possui um tamanho de 70 bytes eo protoloco UDP um header de 8 bytes, tendo então um overhead de transporte de $(8/70) \times 100 \approx 11,42\%$.

2.1.7 Ping

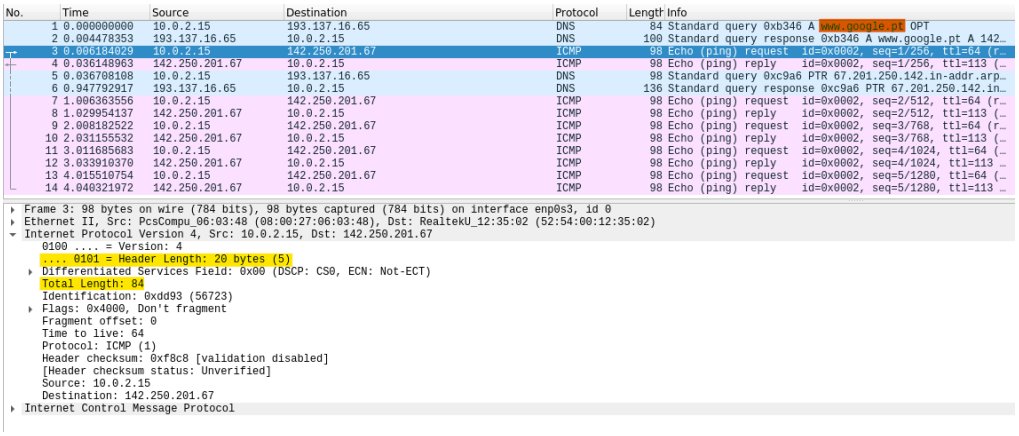

Figura 2.1.7: Ping

Pela figura 2.1.7, é possivel observar que nenhum protocolo de aplicação ou transporte é utilizado devido ao comando ping trabalhar diretamente sobre a camada de rede, logo não possui nenhuma porta de atendimento e um overhead de transporte de 0 bytes.

2.1.8 Traceroute

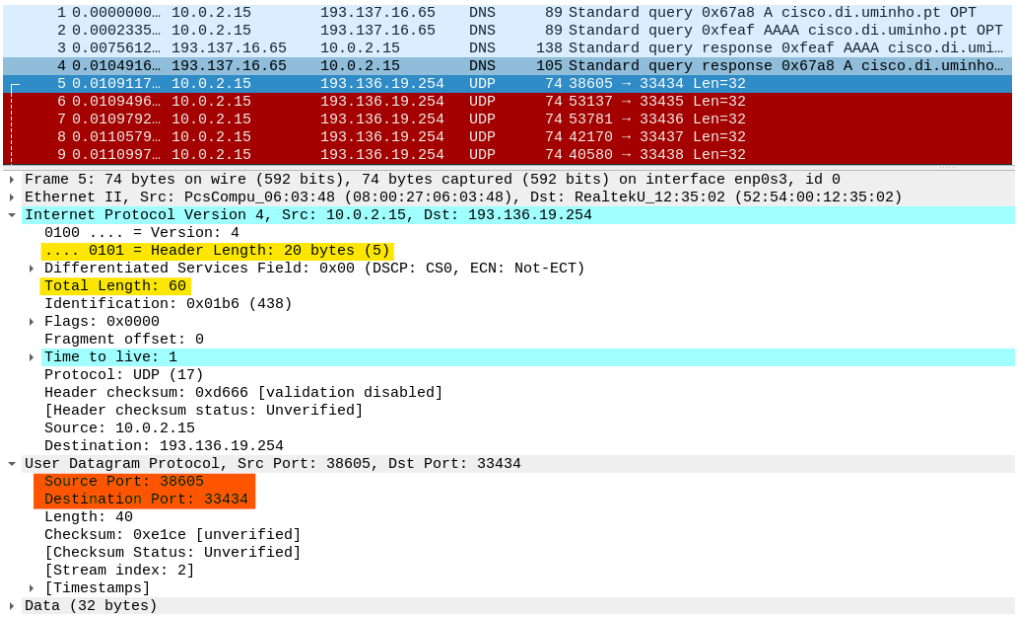

Figura 2.1.8: Traceroute

Como podemos observar pela figura 2.1.8 nenhum protocolo de aplicação é utilizado na execução do comando traceroute. O protocolo de transporte é o UDP e a porta de atendimento a 33434 neste caso particular(podendo outras implementações do traceroute utilizar outras portas). Esta trama possui um overhead de transporte de $(8/60) \times 100 \approx 13,33\%$

Capítulo 3

Conclusão

Ao realizarmos este trabalho prático, conseguimos visualizar o funcionamento dos vários tipos de serviço sobre a transferência de ficheiros numa rede, mais precisamente os seus protocolos de transporte, os desempenhos, o modo como estabelecem conexão com o servidor, a sua complexidade e também a sua segurança.

Com o conhecimento adquirido durante as aulas teóricas, consideramos ter alcançado todos os objetivos propostos neste trabalho prático.