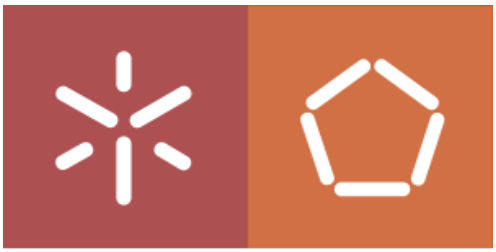


Universidade do Minho
Escola de Engenharia

Comunicações por Computador

Trabalho Prático 1

Grupo 07

2022/2023

Autores:

A91671 – João Manuel Novais da Silva

A91697 – Luís Filipe Fernandes Vilas

A91660 – Pedro António Pires Correia Leite Sequeira

Docente:

Bruno Alexandre Fernandes Dias

Braga,13 de outubro de 2022

Índice

Parte B

Questão 1

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

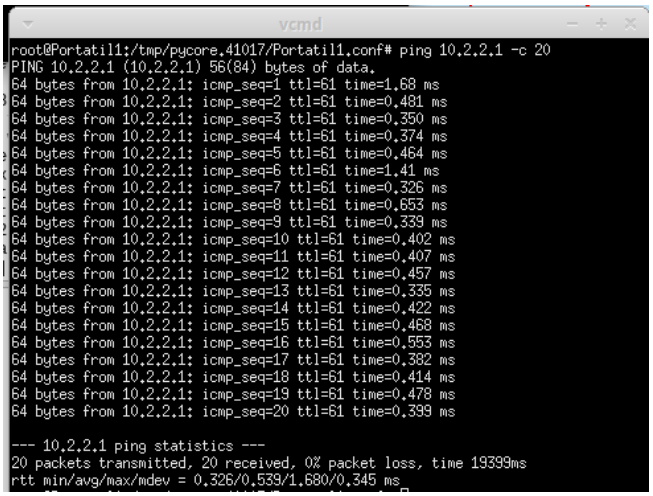


Figura 1 - Portátil

Prestando atenção às Figura 1 e Figura 2, verifica – se que, enquanto no Portatil 1 não existem pacotes duplicados e 0% de package loss, no Grilo há 1 pacote duplicado. A maior parte das vezes, isto deve – se a problemas ao nível das infraestruturas de rede, algo que não acontece com o Portátil 1, pois na rede de Backbone dispõe de 1 Gbps de banda, contrariamente à rede de Backbone do Grilo, que será possivelmente outro tipo de tecnologias e apenas dispõe de uma conexão a 100 Mbps, com 5% Loss e 10% Duplicação (Figura 4).

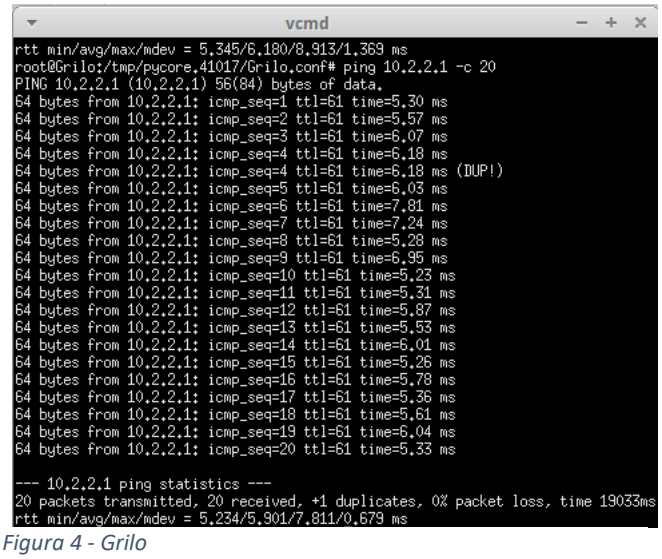


Figura 4 - Grilo

Figura 2 - Backbone Portátil

Figura 3 - Backbone Grilo

Questão 2

Obtenha a partir do Wireshark, ou desenhe manualmente, um diagrama temporal para a transferência do ficheiro file1 por FTP realizada em A.3. Foque-se apenas na transferência de dados [ftp-data] e não na conexão de controlo (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 tanto nos dados como nas confirmações.

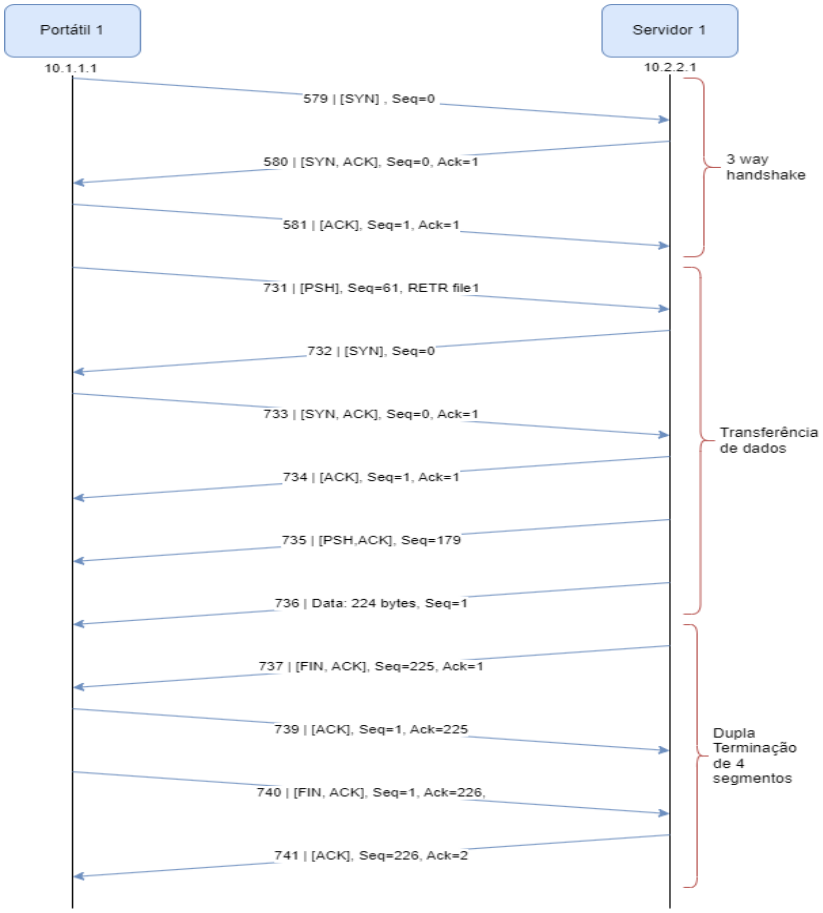


Figura 5 - TCP na transferência do ficheiro 1

Na transferência do file1 por FTP realizada na A.3 houveram pedidos de início de conexão (figura 6) que consistiram num envio inicial de um segmento do tipo [SYN] com [Seq=0] do portátil 1 (10.1.1.1) para o servidor 1 (10.2.2.1), este respondeu com um segmento de tipo [ACK,SYN] e [Seq=0], e por fim um segmento de [ACK] e [Seq=1] do portátil 1. Sendo assim, óbvio que houve um *3 way handshake* para dar início á conexão entre o servidor 1 e o portátil 1 a partir do FTP uma vez que este recorre ao TPC como protocolo de transporte.

No que diz respeito ao início da transferência de dados, é enviado, pelo portátil 1, um segmento de tipo [PSH] com [Seq=61] nomeadamente **Request: RETR file1** para o servidor 1 (figura 7). Este é respondido, pelo servidor, com um segmento do tipo [SYN], assim como, a indicação do número de sequência inicial [Seq=0] e do tamanho de janela para a seguinte transferência. Com isto, o portátil 1 aloca o espaço de armazenamento

necessário e especifica o número de sequência [Seq=0] enviando um segmento do tipo [SYN,ACK]. Por fim, o servidor 1 responde com um segmento [ACK] com o número de sequência igual a 1 ([Seq=1]) e agora já é possível haver transferência de dados.

Analisando os segmentos capturados conseguimos identificar o processo de transferência do file1 entre o servidor e o portátil, visível no segmentos 735 com número de sequência [Seq=179] do tipo [PSH,ACK] e no segmento 736 com número de sequência [Seq=1] e do tipo [PSH,ACK] (figura 8).

Depois de concluída esta transferência o servidor dá sinal que quer terminar a conexão, sendo que já não tem mais informação para enviar, a partir do segmento 737 do tipo [FIN,ACK] e com [Seq=225]. O portátil responde com a verificação da receção do ficheiro através do segmento 738 do tipo [ACK] e [Seq=73], e também, seguidamente, reponde com o segmento 739 que é uma resposta ao pedido de término da conexão do tipo [ACK] e [Seq=1]. Além disto o portátil envia um segmento a indicar o fim da conexão do tipo [FIN,ACK] e [Seq=1]. Por fim o servidor respondo com um [ACK] [Seq=226] na forma do segmento 741.

Com este processo de final de conexão podemos identificar uma dupla terminação de 4 segmentos característico do protocolo TCP.

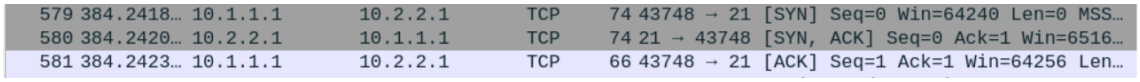


Figura 6 - 3 way handshake no FTP.

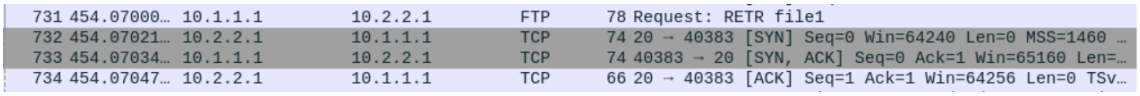


Figura 7 - Início de transferência de dados.

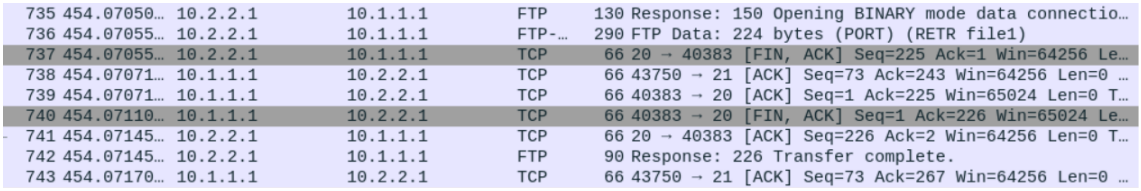


Figura 8 - Processo de transferência de dados e fim da mesma conexão.

Questão 3

Obtenha a partir do Wireshark, ou desenhe manualmente, um diagrama temporal para a transferência do ficheiro file1 por TFTP realizada em A.4. 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 tanto nos dados como nas confirmações.

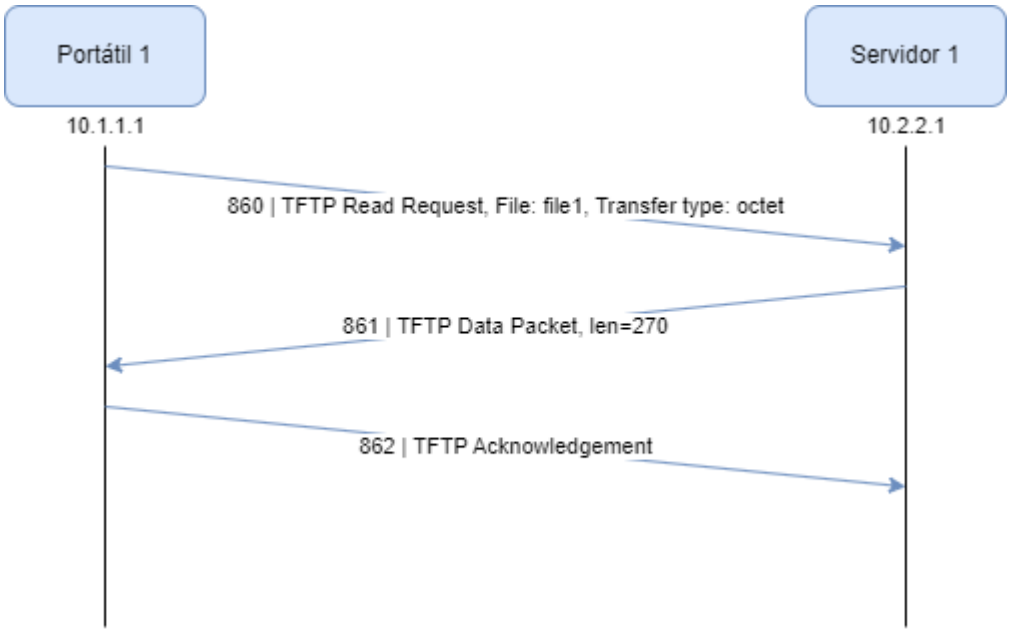
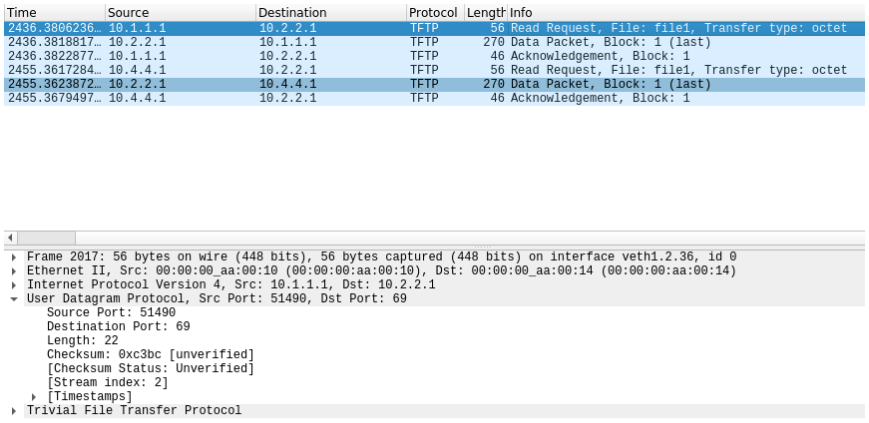


Figura 9 - Transferência de ficheiro por UDP



Time	Source	Destination	Protocol	Length	Info
2436.3806236…	10.1.1.1	10.2.2.1	TFTP	56	Read Request, File: file1, Transfer type: octet
2436.3818817…	10.2.2.1	10.1.1.1	TFTP	270	Data Packet, Block: 1 (last)
2436.3822877…	10.1.1.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 1
2455.3617284…	10.4.4.1	10.2.2.1	TFTP	56	Read Request, File: file1, Transfer type: octet
2455.3623872…	10.2.2.1	10.4.4.1	TFTP	270	Data Packet, Block: 1 (last)
2455.3679497…	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 1

Figura 10 - Pedido do ficheiro através do TFTP

No caso da obtenção do file1 através do método TFTP (figura 9 e 10), como este utiliza o UDP para transporte dos dados, não existe nem fase de início de conexão nem de término da conexão. De igual forma, não existe número de sequência.

Existe apenas um pedido do cliente para transferência do file1, e posteriormente o envio do mesmo por parte do servidor. Quando o ficheiro é recebido, a aplicação "TFTP" envia um *acknowledgement*.

Questão 4

Compare sucintamente as quatro aplicações de transferência de ficheiros que usou, tendo em consideração os seguintes aspetos:

(i) *identificação da camada de transporte;*
(ii) *eficiência;*
(iii) *complexidade;*
(iv) *segurança.*

Tabela 1 - Os dados de eficiência, complexidade e segurança são cotados de Baixa, Médio e Alta, explicando esse raciocínio em baixo.

	Camada de Transporte	Eficiência	Complexidade	Segurança
SFTP	TCP	baixa	alta	alta
FTP	TCP	médio	médio	médio
TFTP	UDP	alta	baixa	baixa
HTTP	TCP	média	média	baixa

O SFTP utiliza o TCP para transporte que permite correção de erros e prevenção de falhas de pacotes. De igual modo, os dados são encriptados o que revela maior complexidade subjacente a uma menor eficiência (tabela 1).

O FTP, também, o TCP para transporte dos dados, mas contrariamente ao SFTP estes não são encriptados, o que revela menor segurança e complexidade em relação ao SFTP. Em troca disto o FTP possuí maior eficiência (tabela 1).

O TFTP é o único que utiliza UDP para transporte, apesar de a aplicação controlar o envio de *acknowledgement* para o destino. Desta forma, não existe nem início de conexão (*3 way handshake* do TCP) nem término da mesma (dupla transmissão de 4 segmentos do TCP). Esta aplicação também não utiliza qualquer encriptação dos dados o que, por sua vez, leva a uma baixa complexidade e baixa segurança, mas uma elevada eficiência (tabela 1).

Para terminar temos o HTTP que, não dispõe de medidas de segurança, mas utiliza o TCP para uma transmissão fiável dos dados ponto a ponto realizando início e término de conexão. Esta aplicação é menos eficiente do que o TFTP e mais complexa que a mesma (tabela 1).

Questão 5

Com base no trabalho realizado, construa uma tabela informativa identificando, para cada aplicação executada (ping, traceroute, telnet, ftp, tftp, wget/lynx, nslookup, ssh, etc.), qual o protocolo de aplicação, o protocolo de transporte, a porta de atendimento e o overhead de transporte.

Tabela 2 - Tabela Informativa

Aplicação	Protocolo de Aplicação	Protocolo Transporte	Porta de Atendimento	Overhead de Transporte
ping	-	-	-	-
traceroute	-	UDP/TCP	33445-33545	8 bytes
telnet	Telnet	TCP	23	20 bytes *
ftp	FTP	TCP	21	44 bytes
tftp	TFTP	UDP	69	8 bytes
wget	HTTP	TCP	80	44 bytes ***
nslookup	DNS	UDP	53	8 bytes *
ssh	SSH	TCP	22	44 bytes ***

* - não foi possível, devido a problemas no servidor, estabelecer a conexão, logo não nos foi possível obter fotos do wireshark.

*** - SYN (40 bytes) + SYN,ACK (40 bytes) + 3*ACK (32 bytes) + FIN (32 bytes) + TCP Segments (32 bytes)

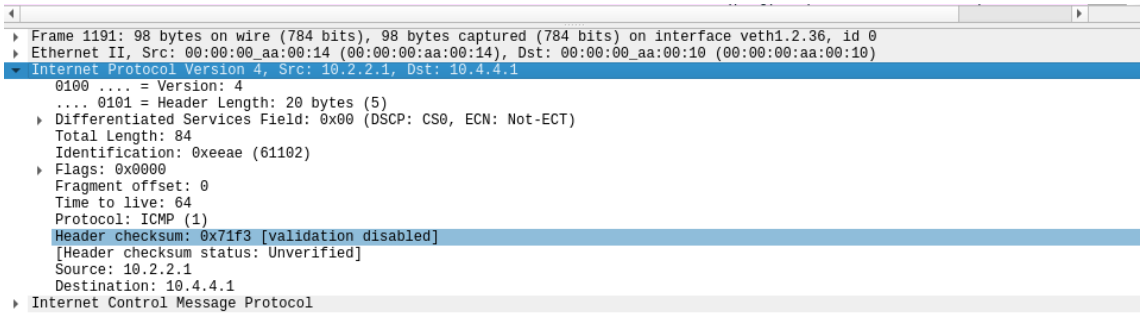


Figura 11 - Ping Wireshark

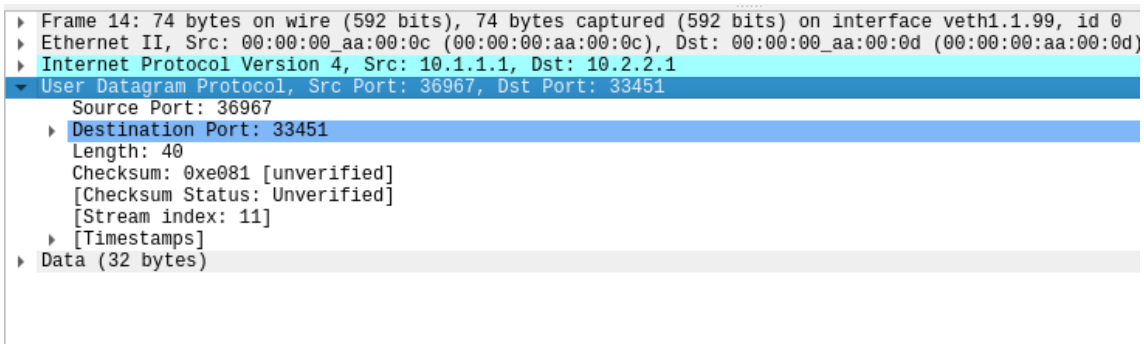


Figura 12 - UDP Wireshark no traceroute

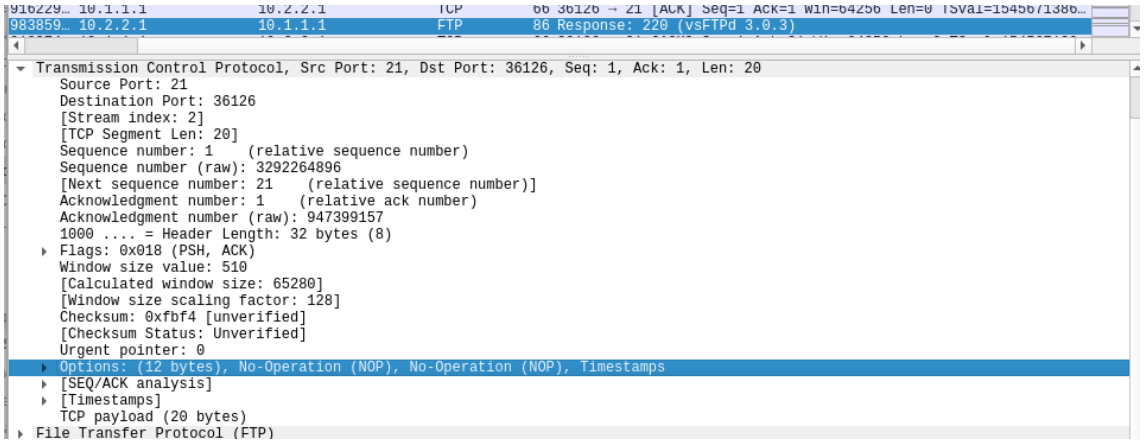


Figura 13 - FTP (12 bytes Opções + 32 Header)

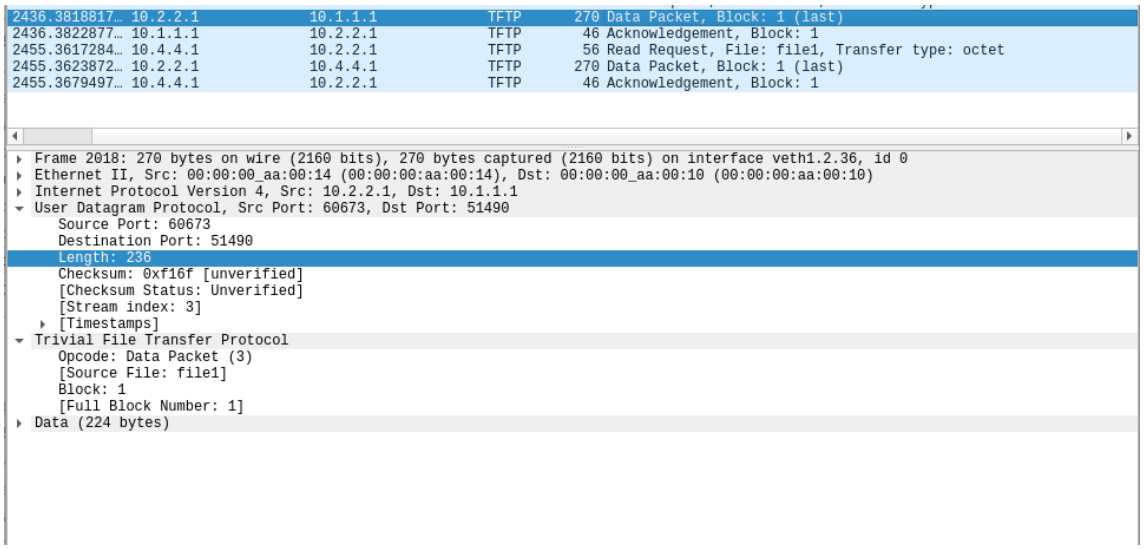


Figura 14 - TFTP UDP

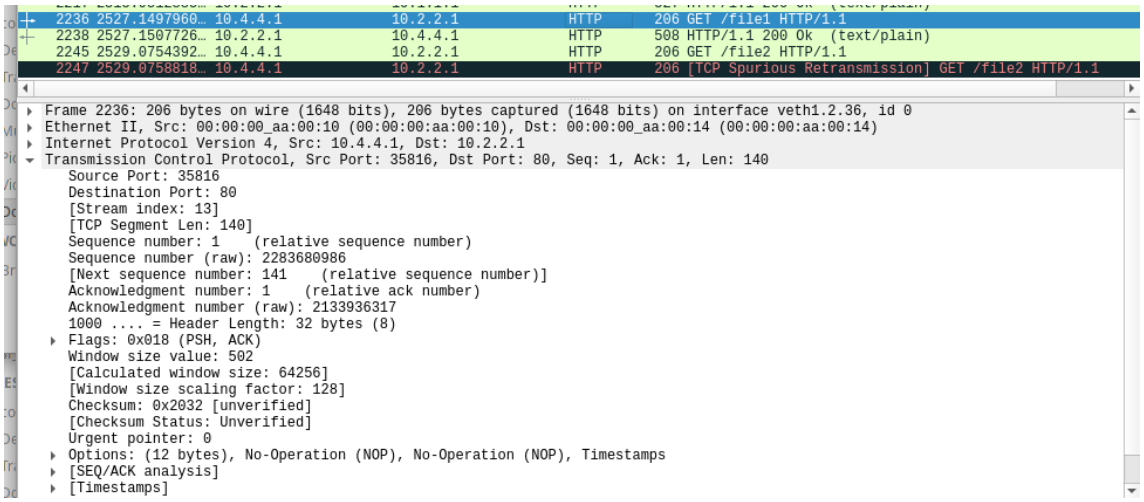


Figura 15 - TCP no HTTP

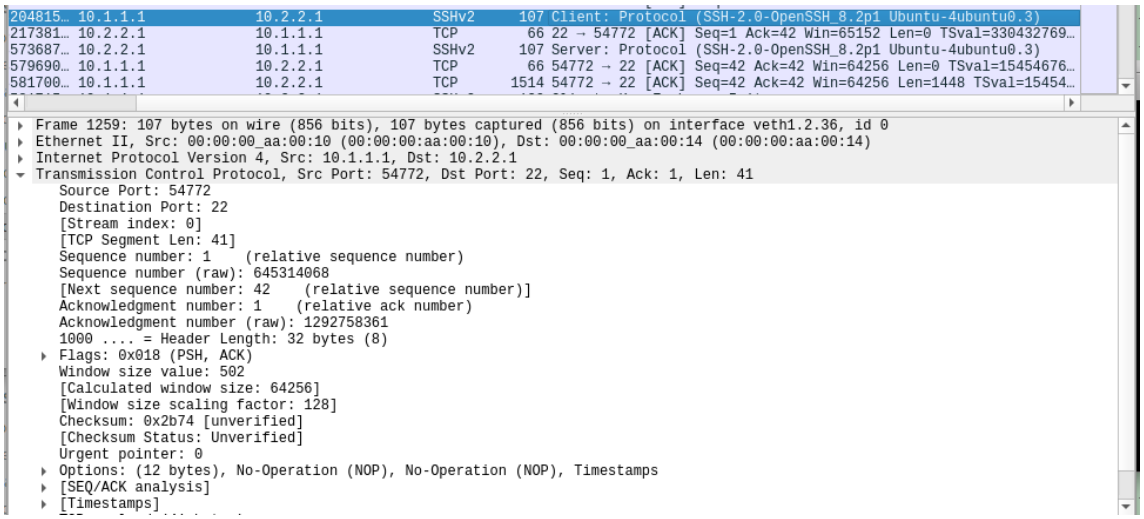


Figura 16 - SSH com base em TCP

Conclusão

A realização deste trabalho ajudou-nos a consolidar vários conceitos alusivos ás camadas de transporte e aplicação vistas nas aulas teóricas.

Estudamos a diferença entre os protocolos TPC (Transmission Control Protocol) e UDP (User Datagram Protocol) e as situações em que os mesmos são aplicados. A complexidade do TCP permite um melhor controlo de erros e de fluxo e permite a segurança na transmissão de dados. O UDP possui uma maior eficiência devido a ser menos complexo permitindo as aplicações que usam o mesmo ter um maior through put de dados apesar de ocorrerem mais erros e perda de informação.

Assim, ficamos a compreender melhor os casos em que estes protocolos são usados e as situações em que os devemos implementar.