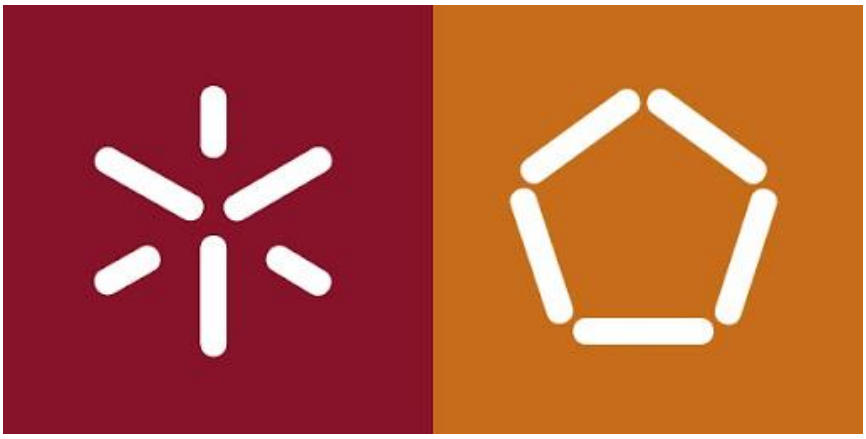

Universidade do Minho

Departamento de Informática

Comunicações por Computadores

TP1-Protocolos da Camada de Transporte

Grupo nº 25

José Pereira (89596)

Diogo Araújo (93313)

Leonardo Freitas (93281)

Questões e Respostas

(Parte I)

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.*

R: Com base nos resultados observados, podemos concluir que as perdas e duplicações de pacotes resultam num decréscimo da performance da aplicação e sobrecarga na rede.
Em termos práticos, constatamos que, com a perda e duplicação de pacotes, o seu envio/reenvio é atrasado (congestionamento, por exemplo), verificando-se uma taxa de transferência menor e uma velocidade de envio inferior à esperada. A capacidade de armazenamento também é influenciada (duplicação) A camada responsável por solucionar estes problemas é a camada de transporte, visto que esta é a responsável pela respetiva transferência de dados entre duas máquinas.

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

R: Depois de ter executado o core com a topologia virtual CC-Topo-2022.imn, criou-se uma bash Shell no nó servidor1 e no nó Portatil1 e a seguir criou-se um processo wireshark no Router1, de maneira a apanhar os pacotes que passam pela interface etho2.

Inicialmente, analisamos a transferência do file1 por FTP. Para isso, dentro da bash Shell do Servidor1 executou-se o comando “vsftpd /etc/vsftpd.conf -osecure_chroot_dir=/srv/ftp -oanonymous_enable=YES”. A seguir na bash Shell do Portatil1 executamos o comando ftp 10.2.2.1 sendo pedido o nome e a palavra-passe.

Após estes procedimentos, transferiu-se o file1 através do comando “get file1”. Uma vez terminada a transferência, termina-se a conexão com o comando quit.

O diagrama temporal associado a esta transferência é o seguinte:

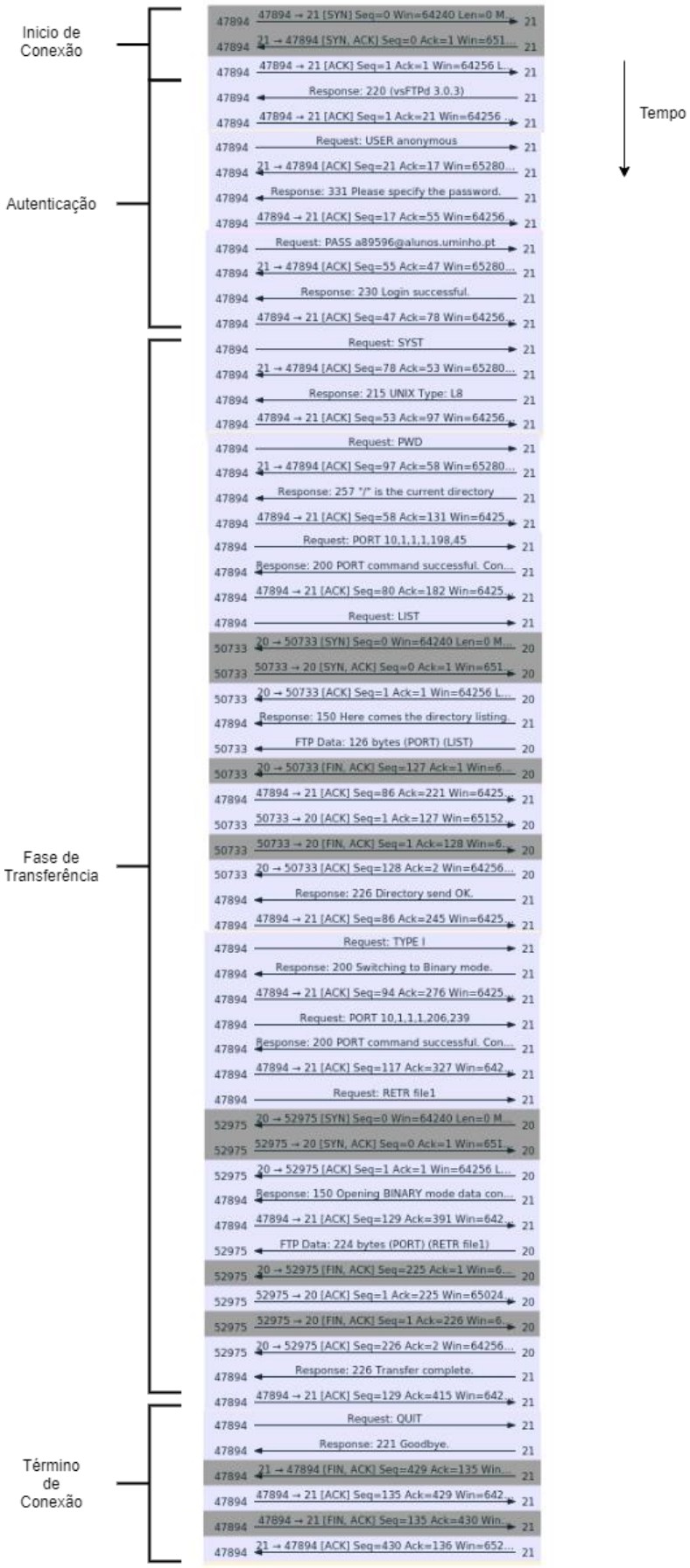

Figura 1: Diagrama temporal da transferência do file1 por FTP

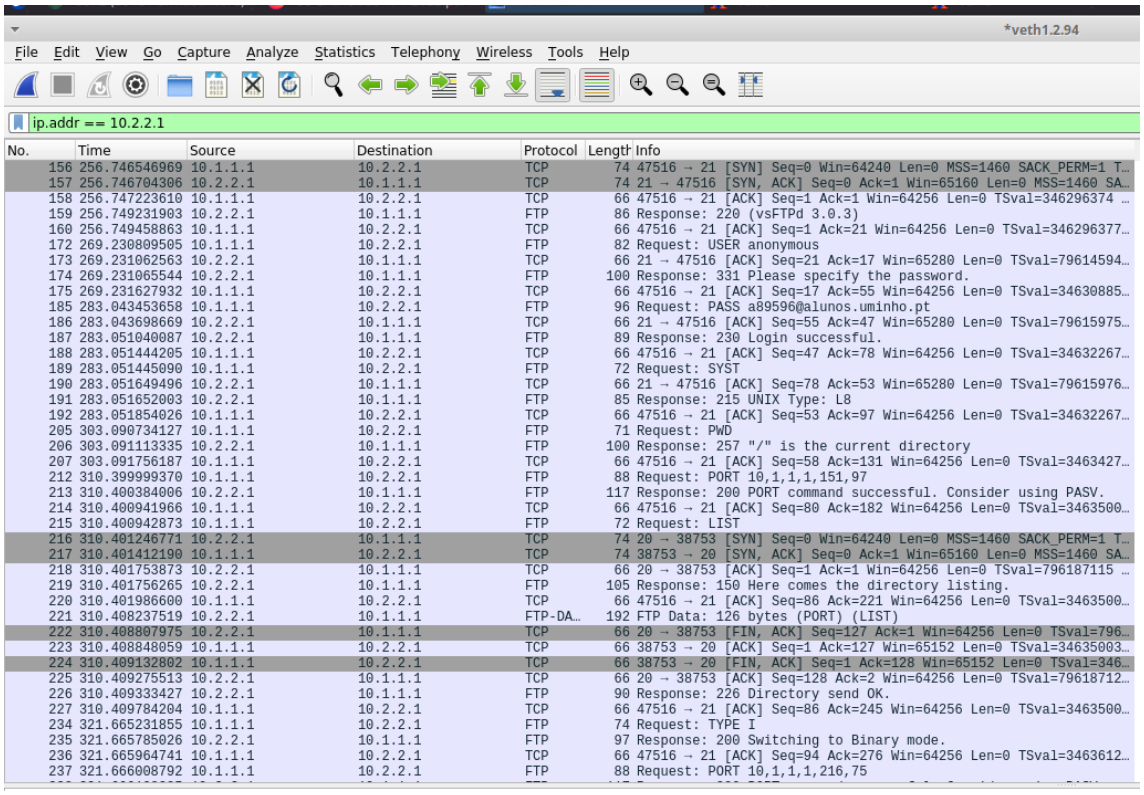

Figura 2: Transferência do file1 por FTP

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.

R: Uma conexão para transferência de dados é estabelecida aquando do uso do protocolo TCP. Porém, neste caso, faz-se uso do UDP, caracterizado por não fazer uso de tal especificidade.
Quanto aos segmentos e números de sequência usados, como é possível atentar na seguinte figura, o ficheiro foi transferido, na sua totalidade, num bloco, daí só ter havido essa interação entre a troca e não haver explicitados quaisquer números de sequência.

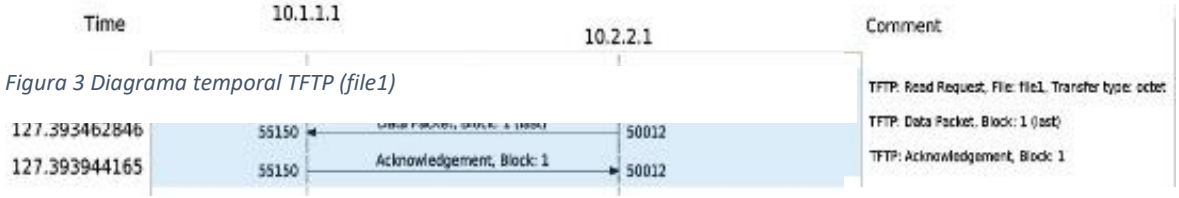

Figura 3 Diagrama temporal TFTP (file1)

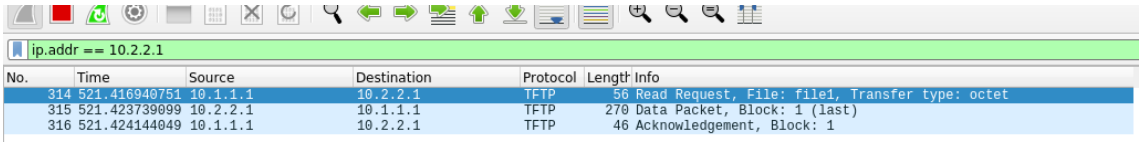

Figura 4:Transferência do file1 por TFTP

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;*

R:

(I) Uso da camada de transporte

Aplicação	*Ficheiro*	*Camada transporte*	*Porta atendimento*
SFTP	File1	TCP	22
FTP	File1	TCP	21
TFTP	File1	UDP	69
HTTP	File1	TCP	80

Relativamente às camadas de transporte utilizadas, depreende-se que apenas a aplicação TFTP utiliza o protocolo UDP (user datagram protocol).

(II) Eficiência

De modo a poder analisar a eficiência de cada uma das 4 aplicações, para a posterior comparação, efetuou-se a transferência do “file1”, com 230 bytes, e, para cada uma, averiguou-se o número de bytes capturados, assim como o tamanho do header.

Como era previsível, pelo protocolo utilizado, a transferência por TFTP foi a mais rápida (o header de controlo em UPD é mais reduzido).

Relativamente às outras 3, todas se comportam de maneira semelhante. Contudo, de acordo com os resultados obtidos, pôde organizar-se a seguinte lista, do mais eficiente para o menos: TFTP, FTP, HTTP, SFTP.

(III) Complexidade

Devido à preparação necessária para se fazer uso, as aplicações TFTP e FTP são as mais complexas entre as estudadas, impõem um maior cuidado e trabalho por parte do utilizador (para efetuar transferências é necessário

configurar e ativar os servidores). Seguindo a mesma lógica e contexto, segue-se o HTTP, de mais simples utilização e, por fim, o SFTP.

Por sua vez, há que indicar que aquando do uso de SFTP, ligações SSH foram estabelecidas de modo a estabelecer uma ligação segura, sendo, portanto, a aplicação mais completa entre as consideradas. Ao contrário desta, pudemos também reparar que utilizando HTTP o conteúdo do ficheiro transferido era facilmente legível, concluindo que o protocolo utilizado por este será mais simples.

(IV) Segurança

De modo análogo com a análise ao ponto anterior, a complexidade da aplicação está diretamente relacionada também com a segurança da mesma. Como já foi indicado, as ligações SSH utilizadas por SFTP estabelecem uma ligação segura e encriptada, sendo a aplicação mais segura a apontar, entre as 4. A partir do momento que as restantes não procedem de forma a encriptar a informação, são menos seguras.
Por sua vez, o protocolo FTP requer um login e password para ser utilizado, mas aquando da captura pelo *wireshark* foi possível notar a password utilizada, isto é, oferece uma ideia de proteção, mas essa é inutilizável na prática.

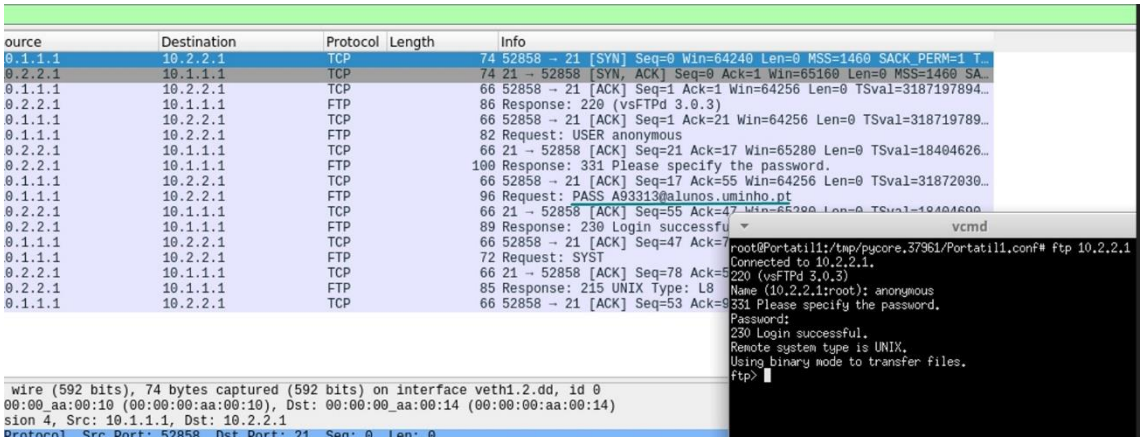

Figura 5: Ftp security vulnerabilities

(Parte II)

1. *Com base na captura de pacotes feita, preencha a seguinte tabela, identificando para cada aplicação executada, qual o protocolo de aplicação, o protocolo de transporte, porta de atendimento e overhead de transporte.*

Comando Usado	Protocolo de Aplicação (se aplicável)	Protocolo de transporte (se aplicável)	Porta de atendimento (se aplicável)	Overhead de Transporte em Bytes (se aplicável)
Ping	Não aplicável	Não aplicável	Não aplicável	Não aplicável
Traceroute	Não aplicável	UDP	55867-33434	8
Telnet	TELNET	TCP	23	20
Ftp	FTP	TCP	21	20
Tftp	TFTP	UDP	69	8
http(browser)	HTTP	TCP	80	20
Nslookup	DNS	UDP	53	8
Ssh	SSHV2	TCP	22	20

Ping

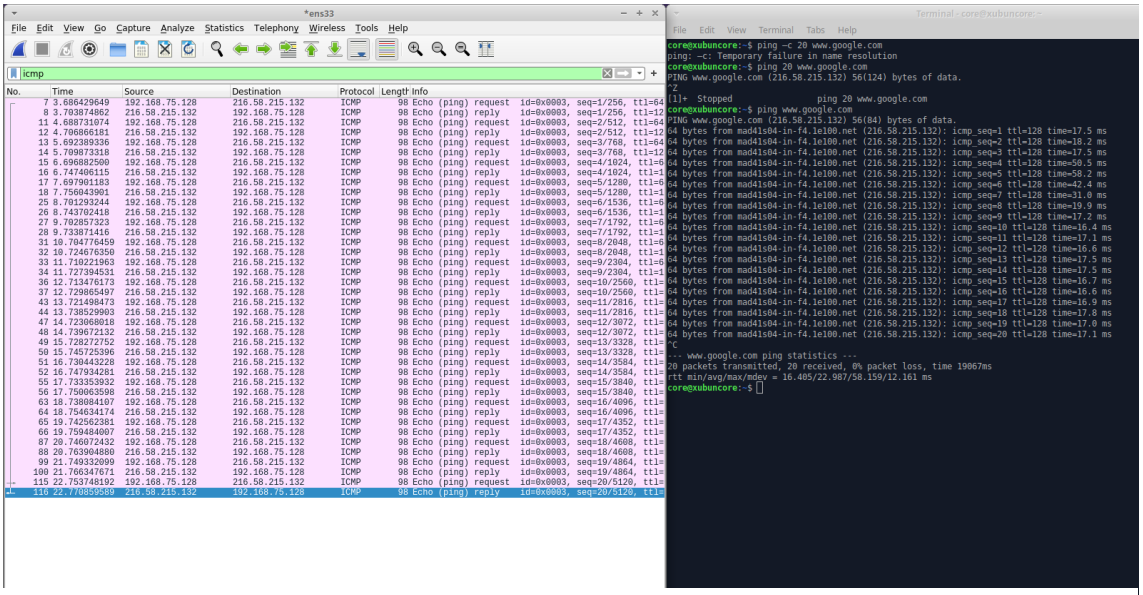

Figura 6: Ping www.google.com

Traceroute

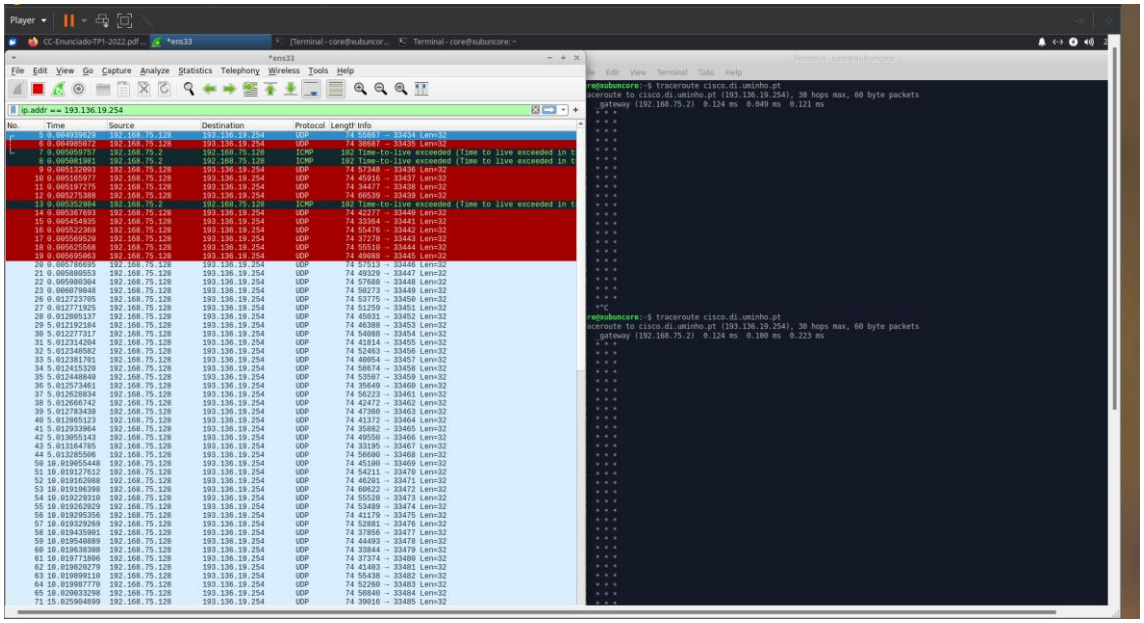

Figura 7: Traceroute ao cisco.di.uminho.pt

Telnet

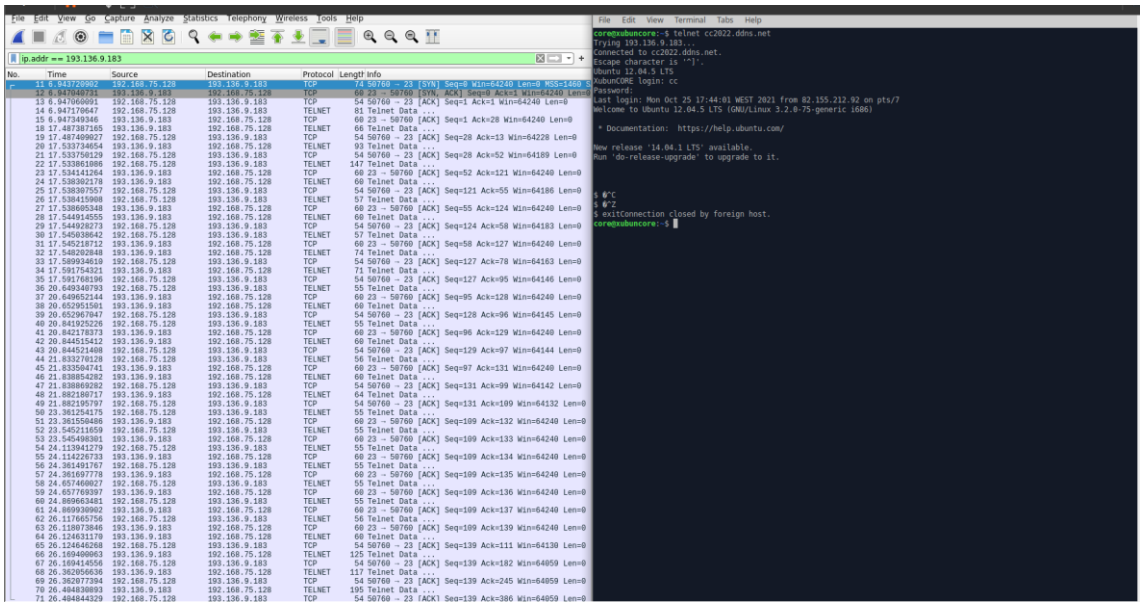

Figura 8: Telnet cc2022.ddns.net

,

FTP

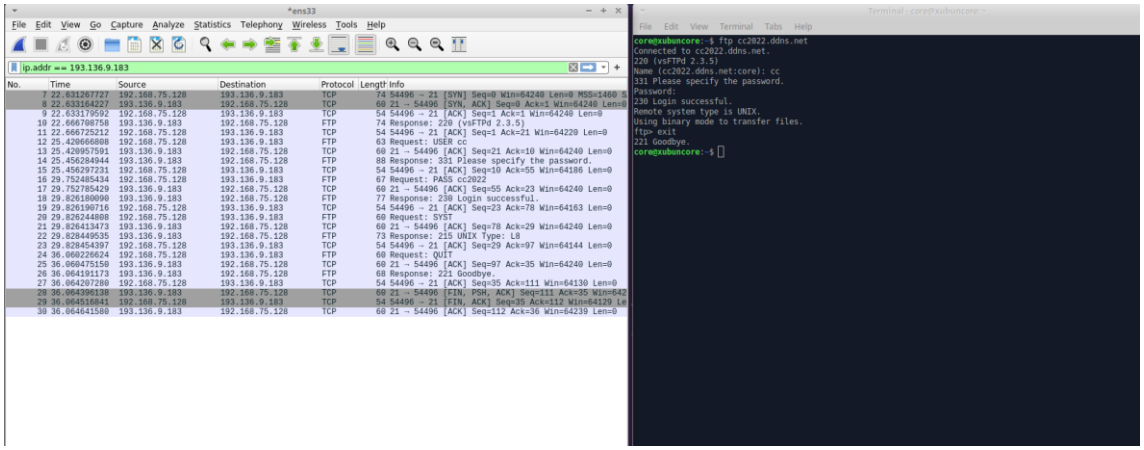

Figura 9: Ftp cc2022.ddns.net

NSLOOKUP

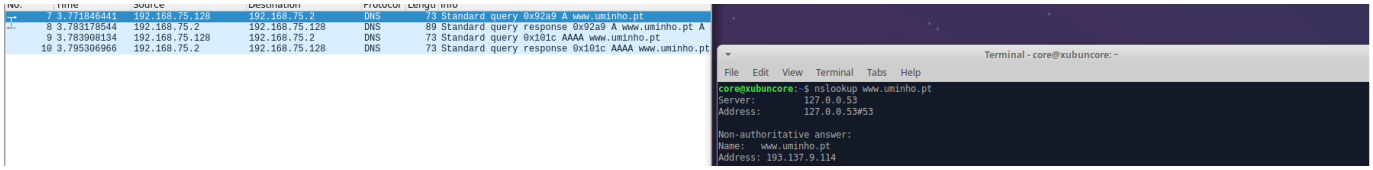

Figura 10: Nslookup www.uminho.pt

TFTP

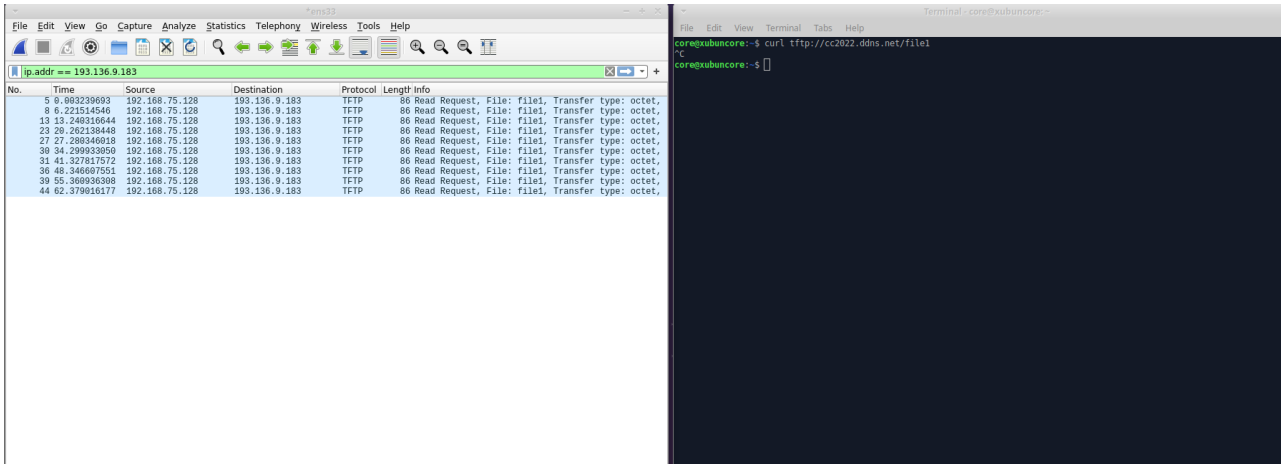

Figura 11: TFTP curl tftp://cc2022.ddns.net/file1

HTTP.

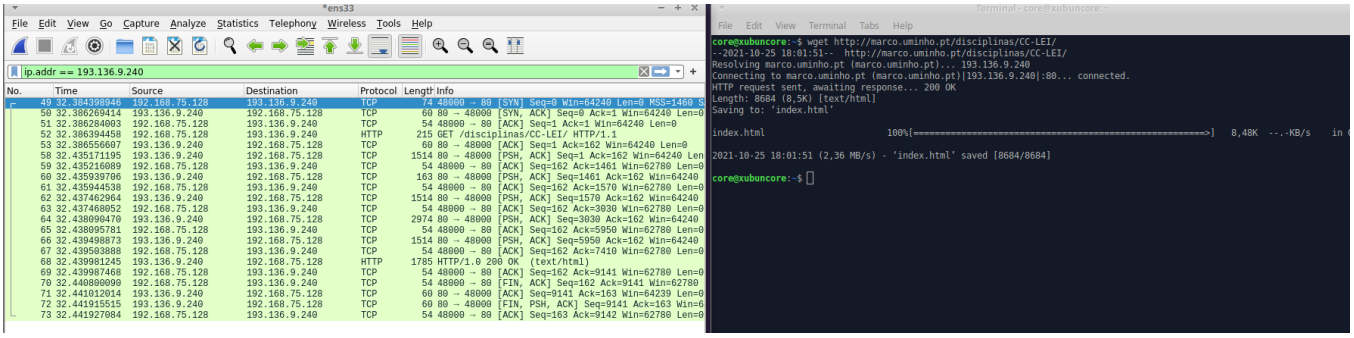

Figura 12: HTTP http://marco.uminho.pt/disciplinas/CC-LEI/

SSH

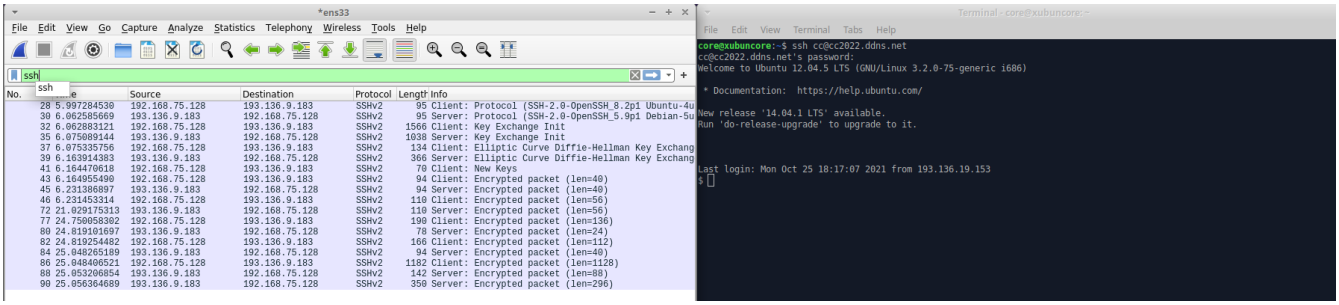

Figura 13: SSH cc@cc2022.ddns.net

Conclusão:

A realização deste trabalho serviu para conciliar as experiências práticas com os conhecimentos adquiridos nas aulas teóricas, facilitando a consolidação da matéria lecionada.

De um modo geral, este primeiro trabalho serviu para reconhecer e analisar ao pormenor os conceitos dos protocolos TCP, UPD e ainda o funcionamento de outros protocolos de aplicação como FTP, HTTP, etc.

Inicialmente, a dificuldade surgiu no controlo e filtragem de toda a informação presente no Wireshark, aquando da captura da rede, mas com a prática foi-se tornando mais evidente o método a seguir. Como aspetos positivos, ficamos a entender como se caracterizam os protocolos em termos de complexidade e segurança, em cada uma das transferências realizadas, permitindo obter informação pormenorizada e importante, não evidente à primeira vista.