UNIVERSIDADE DO MINHO

Trabalho Prático Nº.1 – Protocolos da Camada de Transporte

PL3 Grupo 7

Mestrado Integrado em Engenharia Informática

Comunicações por Computador

A89983	A81931	A84010
Paulo Lima	Luís Duarte	Ulisses Araújo

Braga, 3 de Março de 2020

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	Não	Não	Não	Não
traceroute	MDNS	UDP	33446	8
telnet	telnet	TCP	23	20
ftp	ftp	TCP	21	20
Tftp	Tftp	UDP	69	8
browser/http	http	TCP	80	20
nslookup	DNS	UDP	53	8
ssh	ssh	TCP	22	20

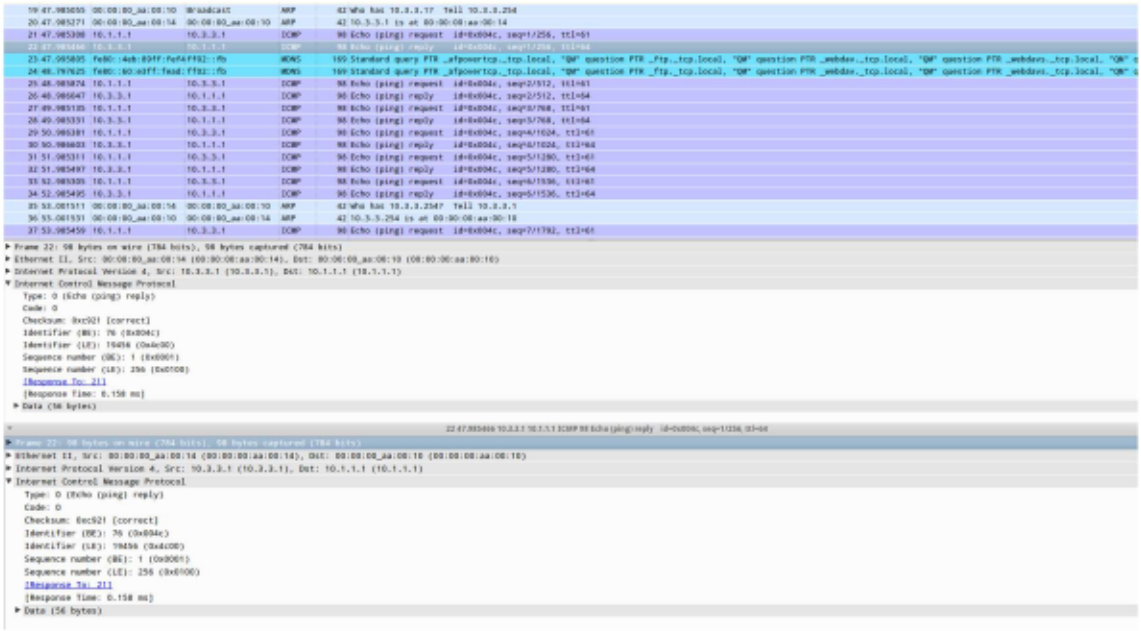

Figura 1 - Ping

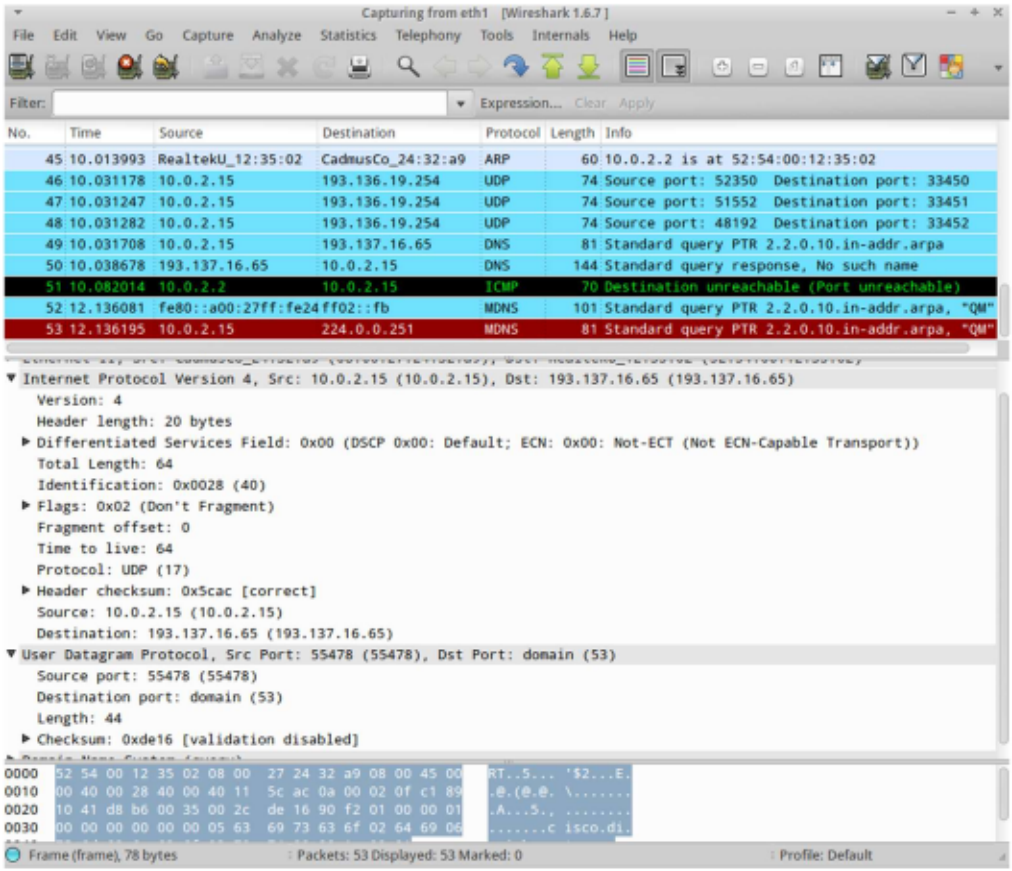

Figura 2 - Traceroute

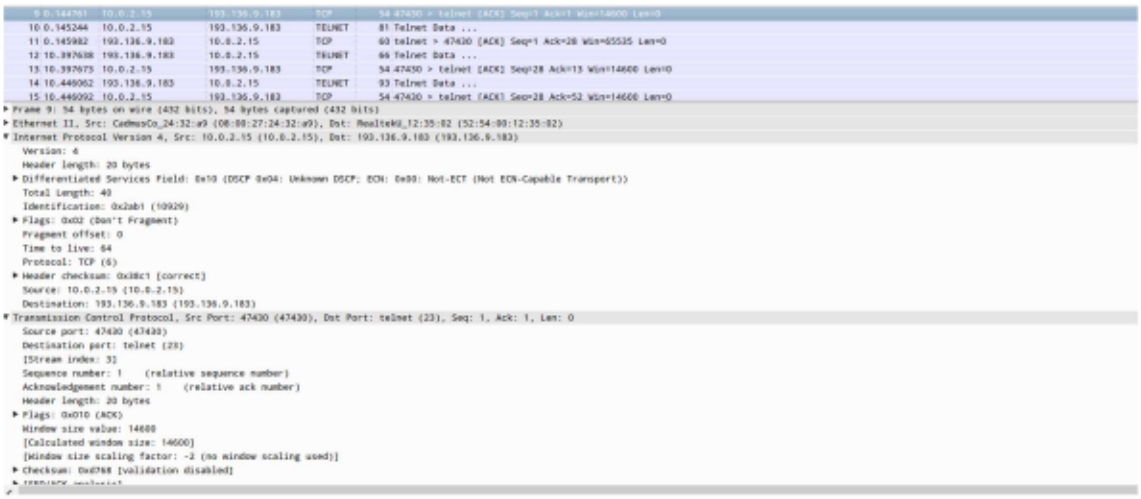

Figura 3 -Telnet

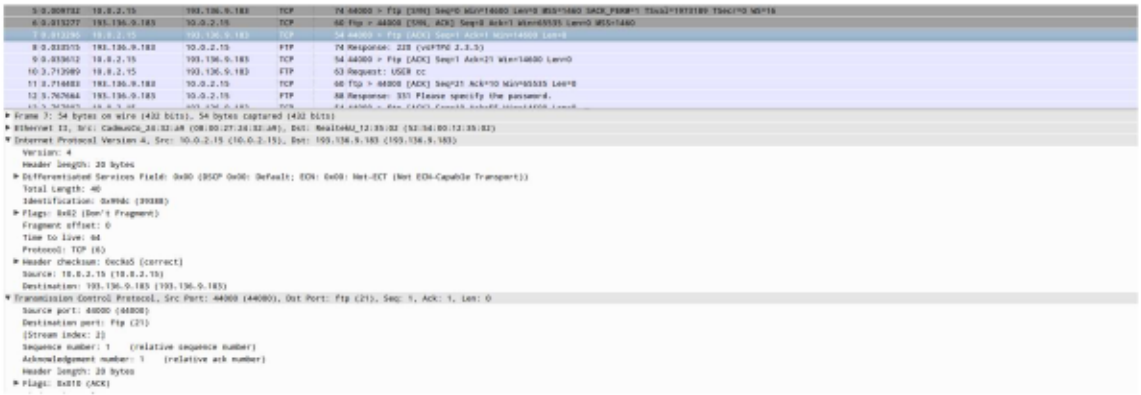

Figura 4 - FTP

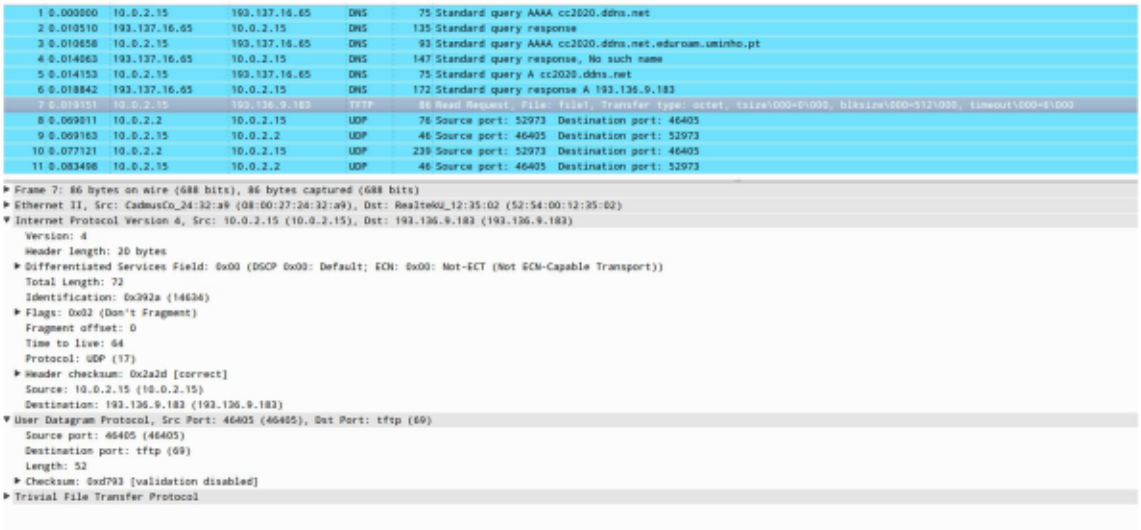

Figura 5 - TFTP

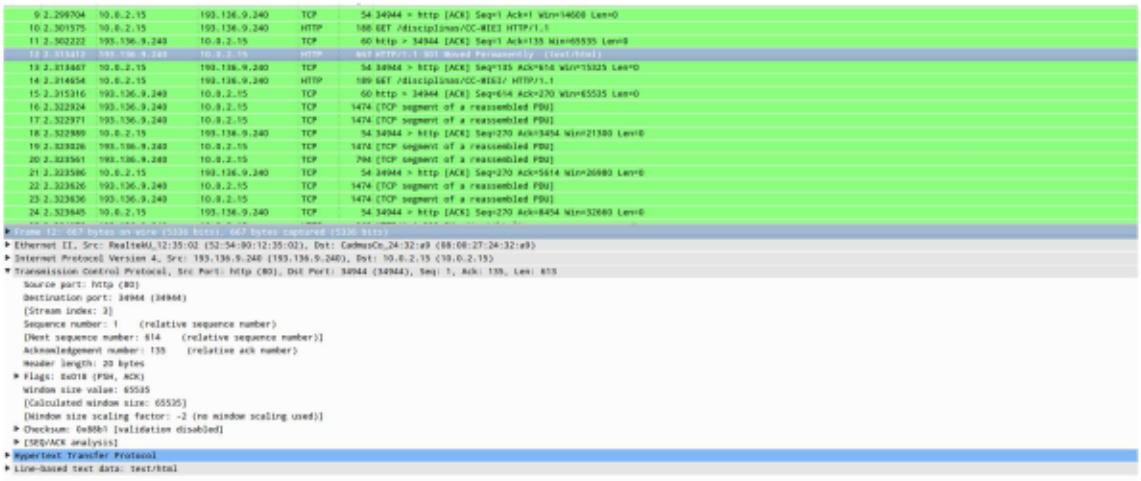

Figura 6 - Browser/HTTP

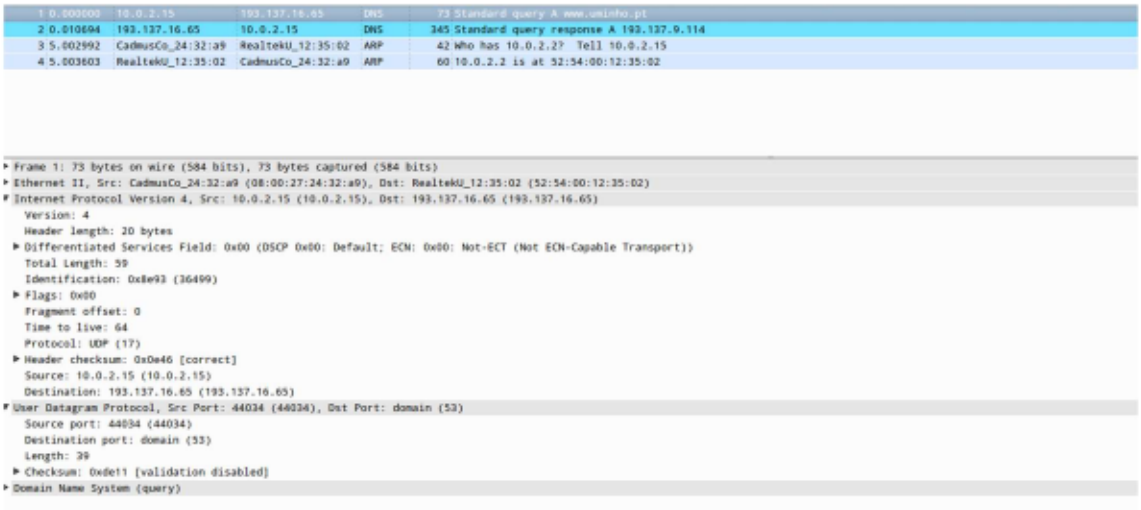

Figura 7 - Nslookup

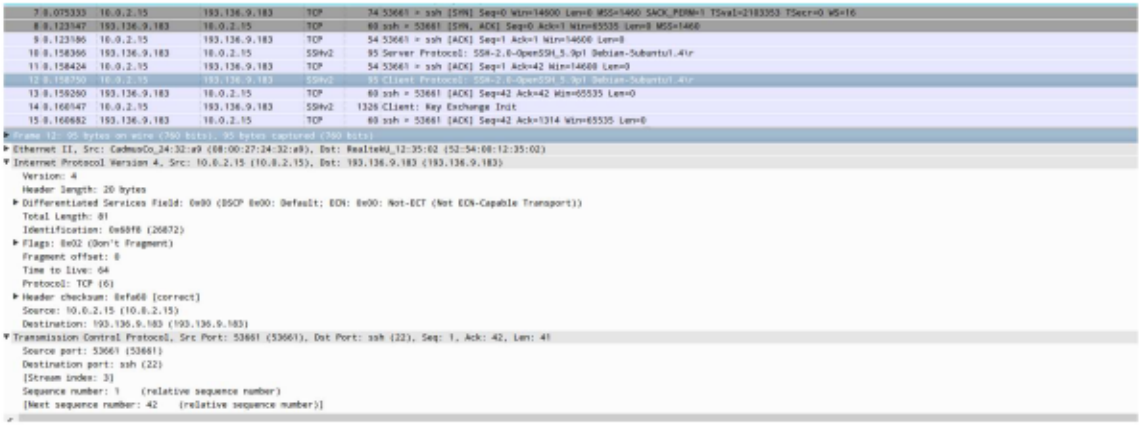

Figura 8 - SSH

2. **Uma representação num diagrama temporal das transferências da file1por 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.**

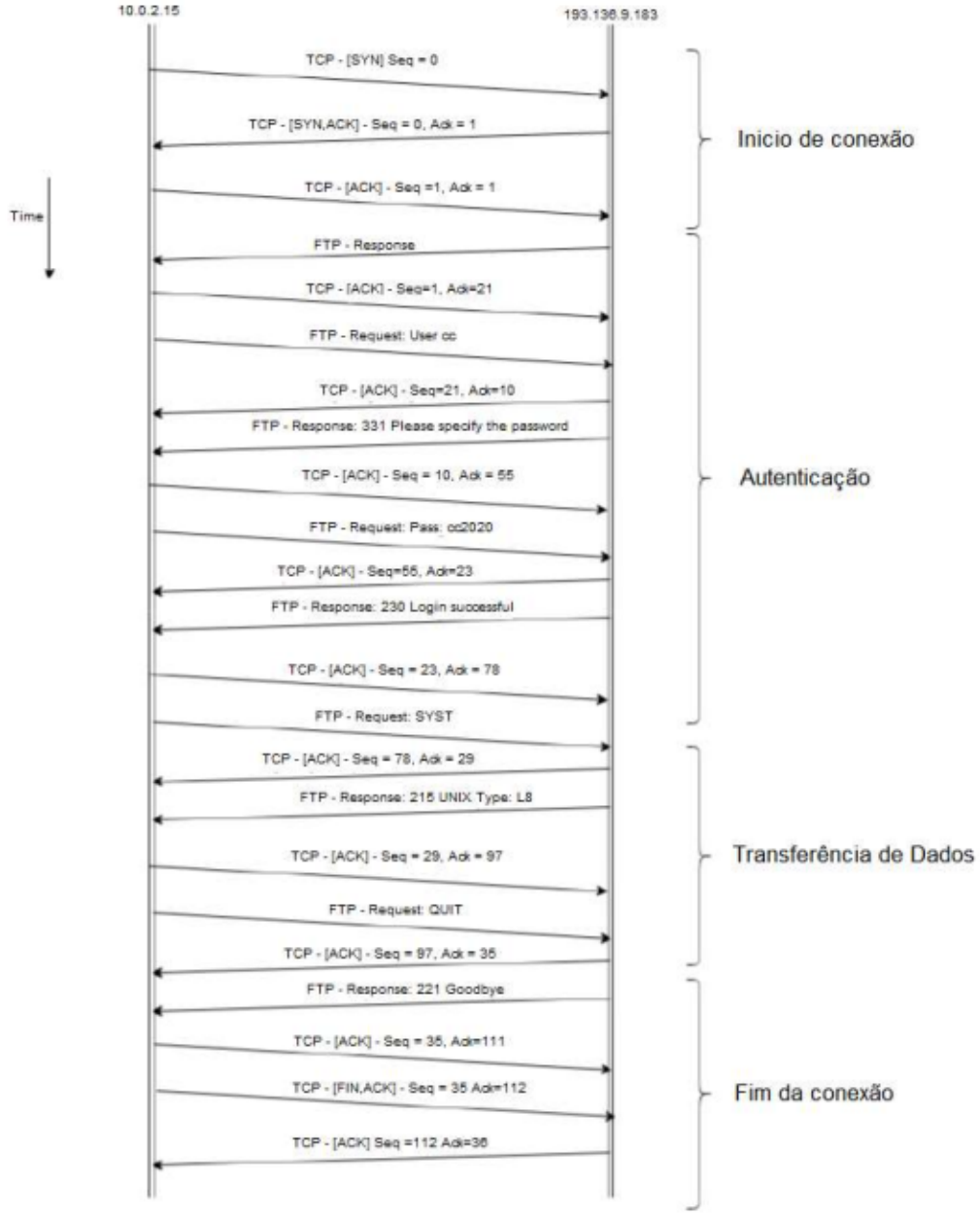

Figura 9 - Diagrama temporal de transferência por FTP

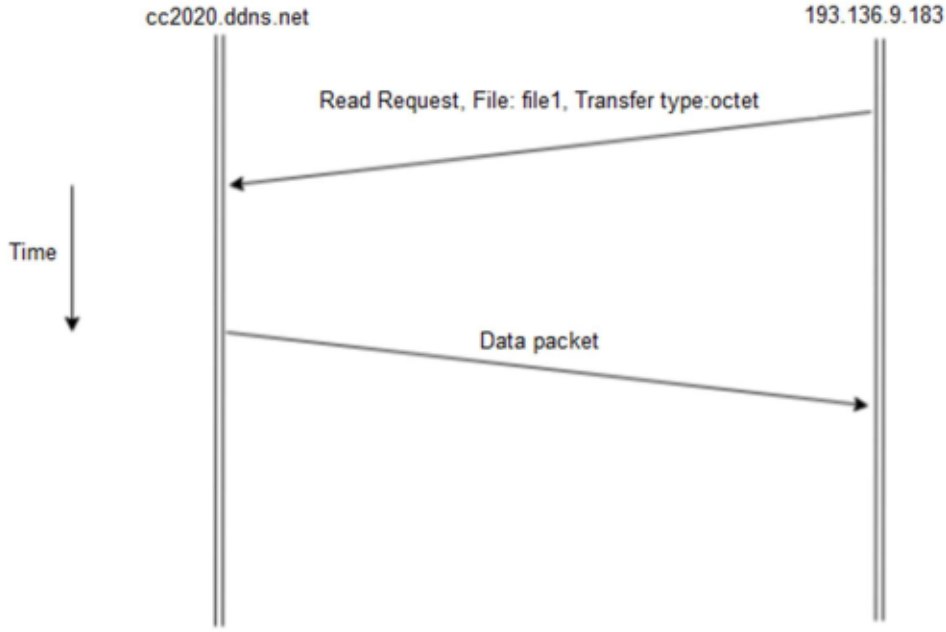

Figura 10 - Diagrama temporal de transferência por TFTP

1	0.000000	10.0.2.15	193.137.16.65	DNS	75 Standard query A cc2020.ddns.net
2	0.003962	193.137.16.65	10.0.2.15	DNS	172 Standard query response A 193.136.9.183
3	0.004229	10.0.2.15	193.137.16.65	DNS	86 Standard query PTR 183.9.136.193.in-addr.arpa
4	0.006809	193.137.16.65	10.0.2.15	DNS	410 Standard query response PTR dhcp-43.uminho.pt
5	0.006956	10.0.2.15	193.136.9.183	TCP	74 44012 > ftp [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=2771957 TSecr=0 WS=16
6	0.010544	193.136.9.183	10.0.2.15	TCP	60 ftp > 44012 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
7	0.010571	10.0.2.15	193.136.9.183	TCP	54 44012 > ftp [ACK] Seq=1 Ack=1 Win=14600 Len=0
8	0.029405	193.136.9.183	10.0.2.15	FTP	74 Response: 220 (vsFTPd 2.3.5)
9	0.030512	10.0.2.15	193.136.9.183	TCP	54 44012 > ftp [ACK] Seq=1 Ack=21 Win=14600 Len=0
10	2.208702	10.0.2.15	193.136.9.183	FTP	63 Request: USER cc
11	2.209213	193.136.9.183	10.0.2.15	TCP	60 ftp > 44012 [ACK] Seq=21 Ack=10 Win=65535 Len=0
12	2.220866	193.136.9.183	10.0.2.15	FTP	88 Response: 331 Please specify the password.
13	2.220922	10.0.2.15	193.136.9.183	TCP	54 44012 > ftp [ACK] Seq=10 Ack=55 Win=14600 Len=0
14	4.537693	10.0.2.15	193.136.9.183	FTP	67 Request: PASS cc2020
15	4.538143	193.136.9.183	10.0.2.15	TCP	60 ftp > 44012 [ACK] Seq=55 Ack=23 Win=65535 Len=0
16	4.626334	193.136.9.183	10.0.2.15	FTP	77 Response: 230 Login successful.
17	4.626491	10.0.2.15	193.136.9.183	TCP	54 44012 > ftp [ACK] Seq=23 Ack=78 Win=14600 Len=0
18	4.626592	10.0.2.15	193.136.9.183	FTP	60 Request: SYST
19	4.627117	193.136.9.183	10.0.2.15	TCP	60 ftp > 44012 [ACK] Seq=78 Ack=29 Win=65535 Len=0
20	4.629501	193.136.9.183	10.0.2.15	FTP	73 Response: 215 UNIX Type: L8
21	4.666598	10.0.2.15	193.136.9.183	TCP	54 44012 > ftp [ACK] Seq=29 Ack=97 Win=14600 Len=0
22	177.883672	10.0.2.15	193.136.9.183	FTP	60 Request: QUIT
23	177.884298	193.136.9.183	10.0.2.15	TCP	60 ftp > 44012 [ACK] Seq=97 Ack=35 Win=65535 Len=0
24	177.891117	193.136.9.183	10.0.2.15	FTP	68 Response: 221 Goodbye.
25	177.891137	193.136.9.183	10.0.2.15	TCP	60 ftp > 44012 [FIN, ACK] Seq=111 Ack=35 Win=65535 Len=0
26	177.891173	10.0.2.15	193.136.9.183	TCP	54 44012 > ftp [ACK] Seq=35 Ack=111 Win=14600 Len=0
27	177.891531	10.0.2.15	193.136.9.183	TCP	54 44012 > ftp [FIN, ACK] Seq=35 Ack=112 Win=14600 Len=0
28	177.891810	193.136.9.183	10.0.2.15	TCP	60 ftp > 44012 [ACK] Seq=112 Ack=36 Win=65535 Len=0
29	182.898571	CadmusCo_24:32:a9	RealtekU_12:35:02	ARP	42 Who has 10.0.2.2? Tell 10.0.2.15
30	182.898979	RealtekU_12:35:02	CadmusCo_24:32:a9	ARP	60 10.0.2.2 is at 52:54:00:12:35:02

Figura 11 - Captura de tráfego transferência por FTP

1	0.000000	10.0.2.15	193.137.16.65	DNS	75 Standard query AAAA cc2020.ddns.net
2	0.060793	193.137.16.65	10.0.2.15	DNS	135 Standard query response
3	0.060889	10.0.2.15	193.137.16.65	DNS	93 Standard query AAAA cc2020.ddns.net.eduroam.uminho.pt
4	0.062950	193.137.16.65	10.0.2.15	DNS	147 Standard query response, No such name
5	0.063008	10.0.2.15	193.137.16.65	DNS	75 Standard query A cc2020.ddns.net
6	0.113070	193.137.16.65	10.0.2.15	DNS	172 Standard query response A 193.136.9.183
7	0.113308	10.0.2.15	193.136.9.183	TFTP	86 Read Request, File: file1, Transfer type: octet, tsize\000=0\000, blksize\000=512\000, timeout\000=6\000
8	0.201683	10.0.2.2	10.0.2.15	UDP	76 Source port: 52601 Destination port: 54120
9	0.205581	10.0.2.15	10.0.2.2	UDP	46 Source port: 54120 Destination port: 52601
10	0.208559	10.0.2.2	10.0.2.15	UDP	239 Source port: 52601 Destination port: 54120
11	0.208607	10.0.2.15	10.0.2.2	UDP	46 Source port: 54120 Destination port: 52601
12	5.001934	CadmusCo_24:32:a9	RealtekU_12:35:02	ARP	42 Who has 10.0.2.2? Tell 10.0.2.15
13	5.002545	RealtekU_12:35:02	CadmusCo_24:32:a9	ARP	60 10.0.2.2 is at 52:54:00:12:35:02

Figura 12 - Captura de tráfego transferência por FTP

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;**

O HTTP, "Hipertext Transfer Protocol", é um protocolo de comunicação inerente à camada de aplicação, sendo a base de comunicação de dados da "World Wide Web". Através da análise de pacotes, usando o wireshark no router 1 da topologia do CORE fornecida, verificou-se que utiliza o protocolo TCP como protocolo da camada de transporte. Em termos de eficiência, o HTTP sofre devido ao "slow start" implementado pelo TCP. Isto graças ao tamanho dos "HTTP headers", que são maiores que o MSS. Por exemplo, o cliente TCP logo na fase de conexão necessita de usar dois pacotes, e como o controlo de congestão implementado apenas permite uma janela de tamanho 1 no início, o cliente não consegue comunicar simultaneamente os dois pacotes, o que causa uma queda na eficiência. Além disto, ainda se verificou que possui vários esquemas de autenticação, no entanto, é inseguro, uma que vez que qualquer pessoa na rede consegue ver o conteúdo dos ficheiros antes destes chegarem ao recetor.

O FTP, "File Tranfer Protocol", é um serviço básico de transferência fiável de ficheiros. Usa o protocolo TCP e também tem certos problemas de eficiência devido ao elevado "overhead". Além disto, não aparenta implementar nenhuma segurança adicional, podendo se verificar no momento de autenticação, que o processo não possui nenhuma encriptação, sendo as mensagens trocadas em formato texto.

O SFTP usa novamente, como protocolo da camada de transporte, o TCP. É o protocolo mais complexo dos quatro analisados e também o mais seguro, uma vez que usa o SSH, "Secure Shell", na encriptação dos dados comunicados. É ainda o menos eficiente devido à complexidade que possui (por exemplo, o uso de SSH causa elevado "overhead").

O TFTP é um serviço básico de transferência não fiável de ficheiros, uma vez que usa, como protocolo da camada de transporte, o UDP. É o protocolo mais simples analisado, e não possui nenhum mecanismo de segurança adicional, nem mecanismos de autenticação. Tem ainda, elevada eficiência na transmissão, devido à simplicidade e baixo "overhead". É útil de usar para a transferência de pequenos ficheiros entre hosts na mesma rede.

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).**

No desenvolvimento de uma aplicação é de extrema importância ter cuidado na escolha do protocolo de transporte de forma a garantir fiabilidade.

Assim, uma das principais razões para a escolha do TCP é porque este se trata de um protocolo da camada de transporte fiável, pois garante que todos os pacotes são transmitidos na ordem correta e sem erros. Isto acontece porque para cada pacote recebido é sempre transmitido um "Acknowledgment", em como esse pacote foi recebido com sucesso, ou caso não tenha sido, com um pedido de retransmissão, quer este tenha sido causado por perda ou duplicação de pacotes. Ora este nível de controlo causa também elevada complexidade o que faz baixar a eficiência da transmissão, sendo que quantos mais erros ocorrerem, mais a eficiência baixa. Desta forma, uma rede de menor qualidade leva a pacotes perdidos ou corrompidos com mais frequência, o que obriga à maior transmissão de mensagens de erro discutidas anteriormente e que sejam reenviados os vários pacotes, causando uma maior sobrecarga na rede e atraso na aplicação.

Isto pode ser verificado, pela comparação dos tempos de transferência do "file1" e do "file2", uma vez que na transferência do "file2" ocorreram erros, apresentados de seguida, que aliado ao tamanho do "file2" também ser maior que o do "file1", levou a uma queda na eficiência.

Por outro lado, as alternativas não são fiáveis, como o UDP. Este protocolo não implementa uma comunicação fiável e é bem menos complexo, o que por outro lado aumenta a eficiência. Embora não seja fiável, a própria aplicação poderá sempre garantir que os dados são corretamente recebidos. Desta forma a vantagem do UDP reside na redução da latência de ligação e como não faz verificações não sobrecarrega a rede com mensagens de erro, nem retransmissões.

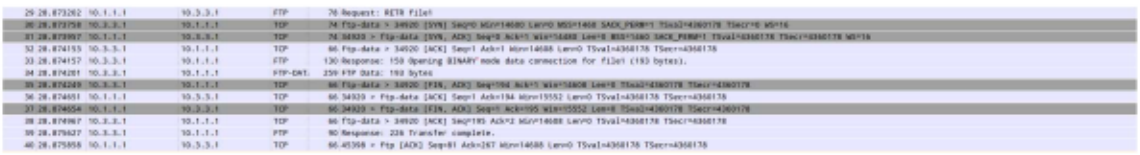

Figura 13 - Captura de tráfego transferência file 1

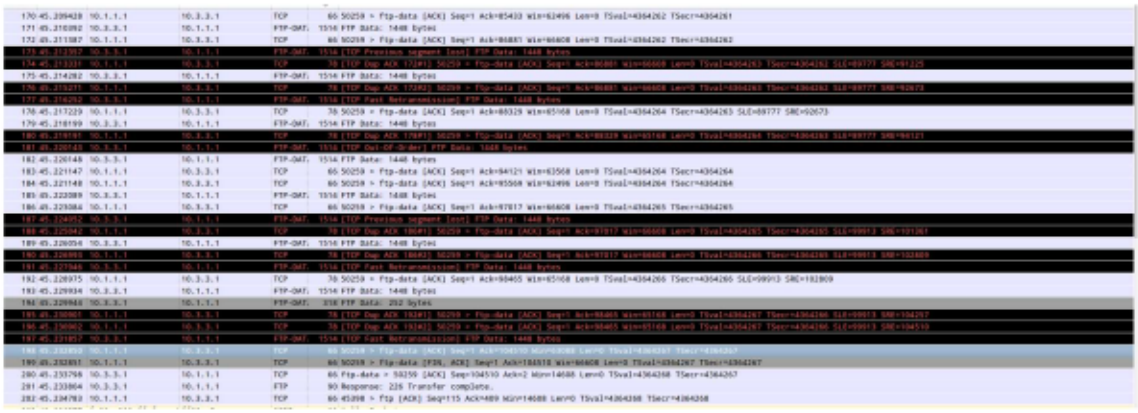

Figura 14 - Ocorrência de pacotes duplicados

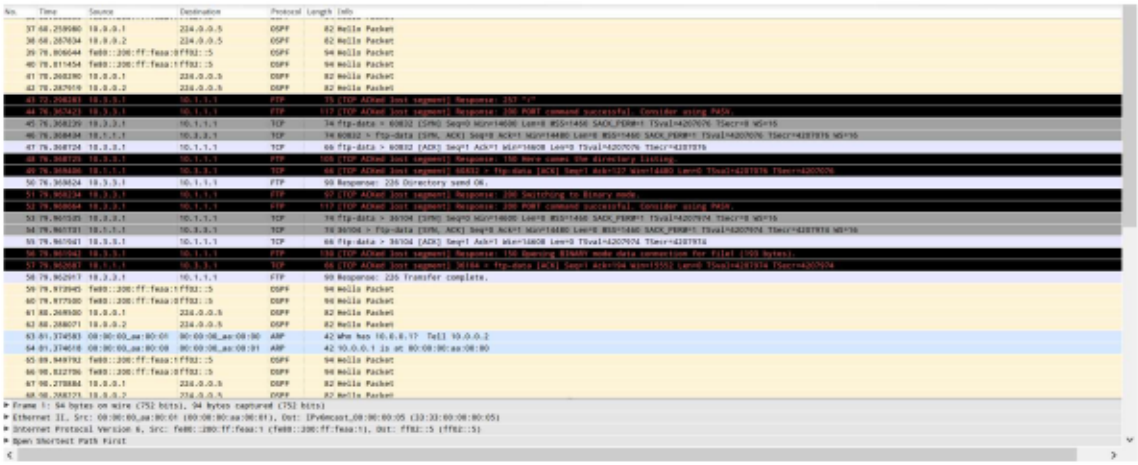

Figura 15 - Ocorrência de perda pacotes

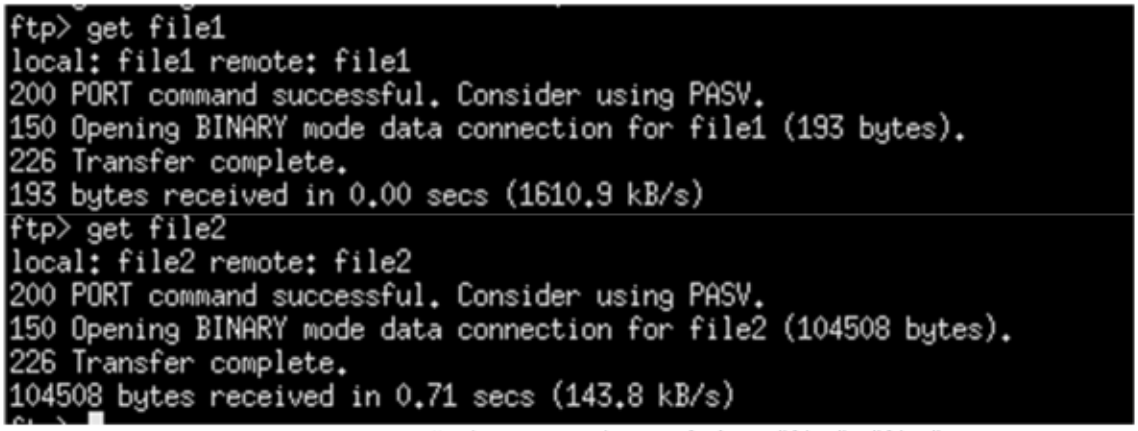

Figura 16 - Comparação dos tempos de transferência "file1" e "file2"

Conclusões

O trabalho realizado serviu de complemento à matéria lecionada nas aulas teóricas desta unidade curricular, nomeadamente o estudo de dois protocolos distintos, o TCP e o UDP.

Conseguiu-se visualizar a diferença na implementação dos dois protocolos referidos anteriormente, desde a eficiência até à fiabilidade de cada um, o que permitiu observar de forma prática aquilo que era esperado. Além disto foram observados ainda alguns protocolos da camada de aplicação e como estes usam a camada de transporte, isto é, se implementam TCP ou UDP, e as consequências disso.

Também foi interessante perceber como a qualidade da construção de uma rede, principalmente ao usar o protocolo TCP, pode afetar a sua própria eficiência.

O mais trabalhoso foi a análise de tráfego gerado pelos diferentes protocolos recorrendo ao "wireshark".

Em conclusão, este trabalho foi um excelente complemento ao capítulo da camada de transporte lecionado na primeira parte da unidade curricular.