Universidade do Minho
Escola de Engenharia

Comunicações por Computador

TP1: Protocolos da Camada de Transporte

Guilherme da Silva Amorim Martins A89532
José Pedro Carvalho Costa A89519
SImão Paulo da Gama Castel-Branco e Brito A89482

A89532

A89519

A89482

16 de março de 2021

Questões e Respostas

1 Pergunta 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:

Comando usado	Protocolo de Aplicação	Protocolo de transporte	Porta de atendimento	Overhead de transporte
ping	DNS	TCP	443	20
traceroute	DNS	UDP	33450	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	SSHv2	TCP	22	20

Como se pode verificar pela tabela e pelas imagens a seguir que a validam, os protocolos de transporte têm sempre um overhead fixo de 8 bytes e 20 bytes para os protocolos UDP e TCP, respetivamente, no entanto, não tendo acontecido esse caso nos nossos comandos, o valor do overhead do TCP poderia aumentar mediante o uso da flag *options*.

1.1 ping

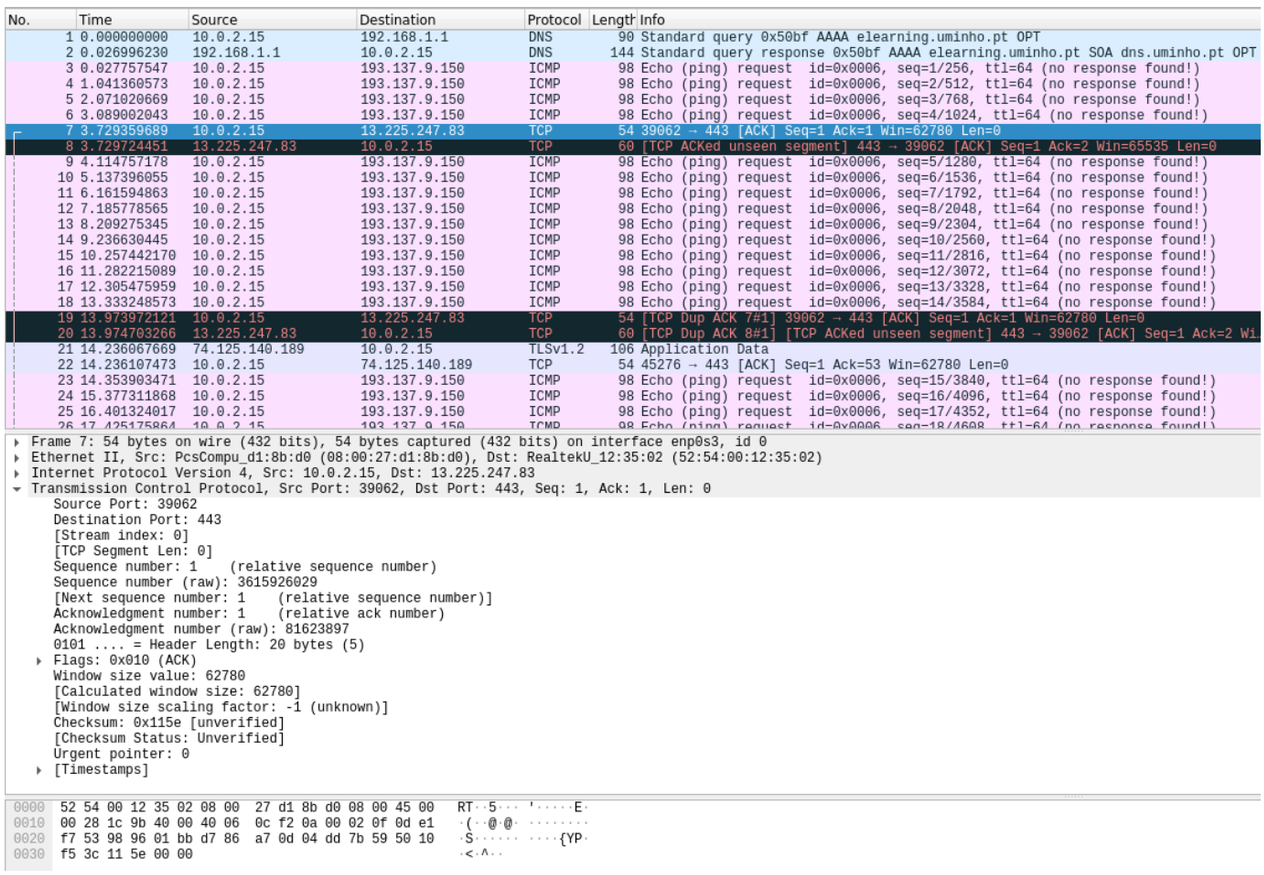

Figure 1: Captura de tráfego ao realizar o ping

1.2 traceroute

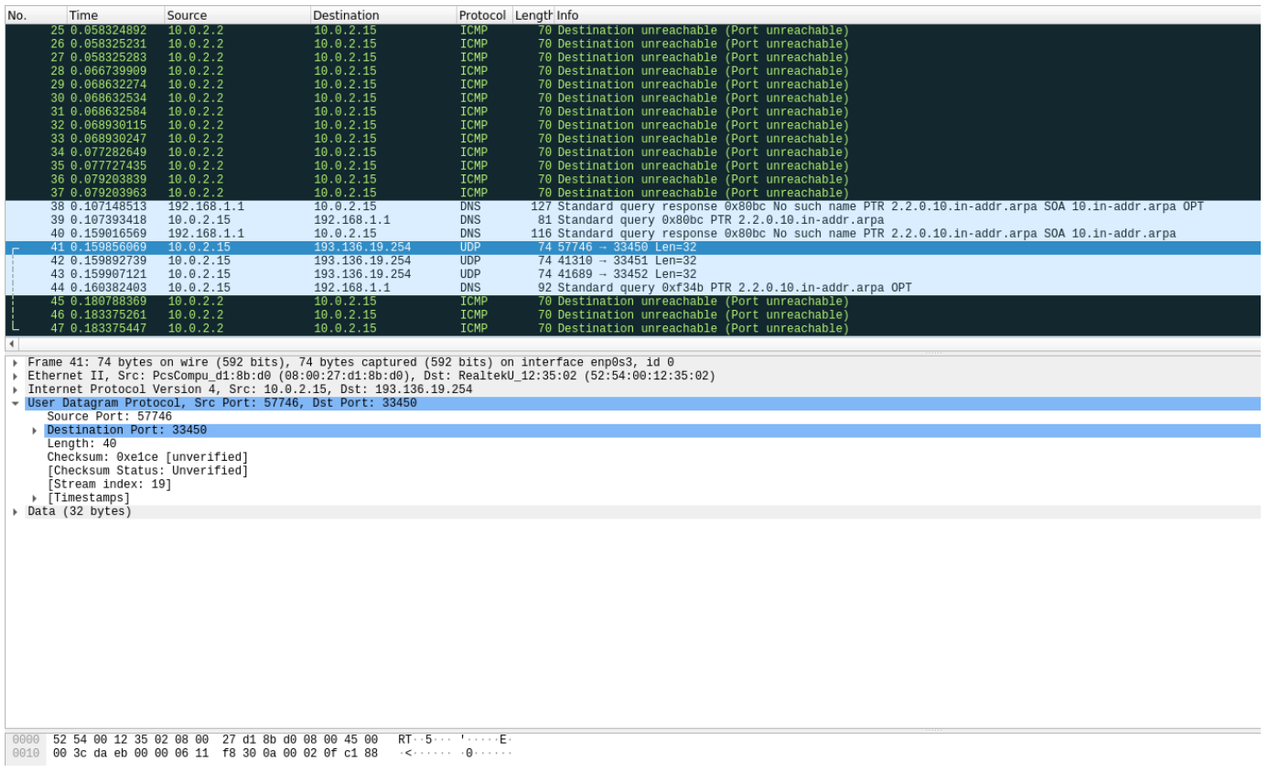

Figure 2: Captura de tráfego ao realizar o traceroute

1.3 telnet

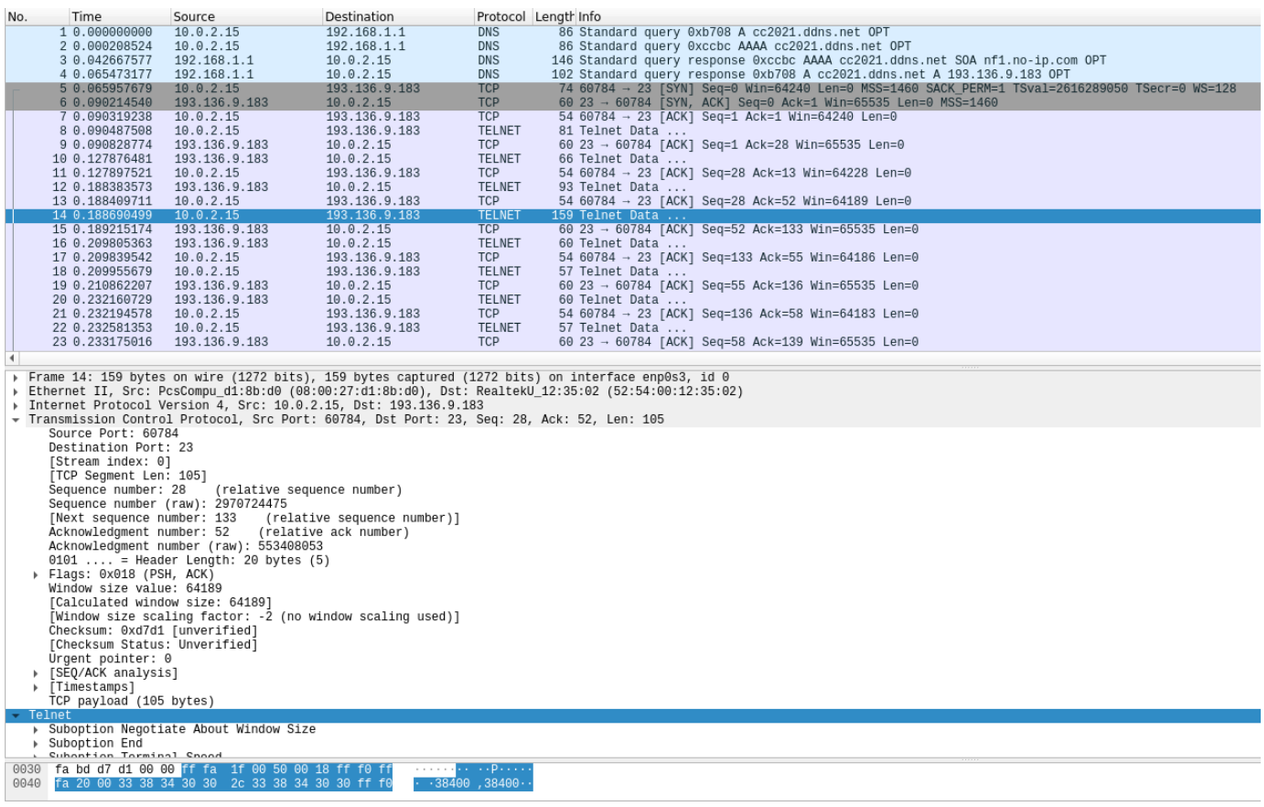

Figure 3: Captura de tráfego ao realizar o telnet

1.4 ftp

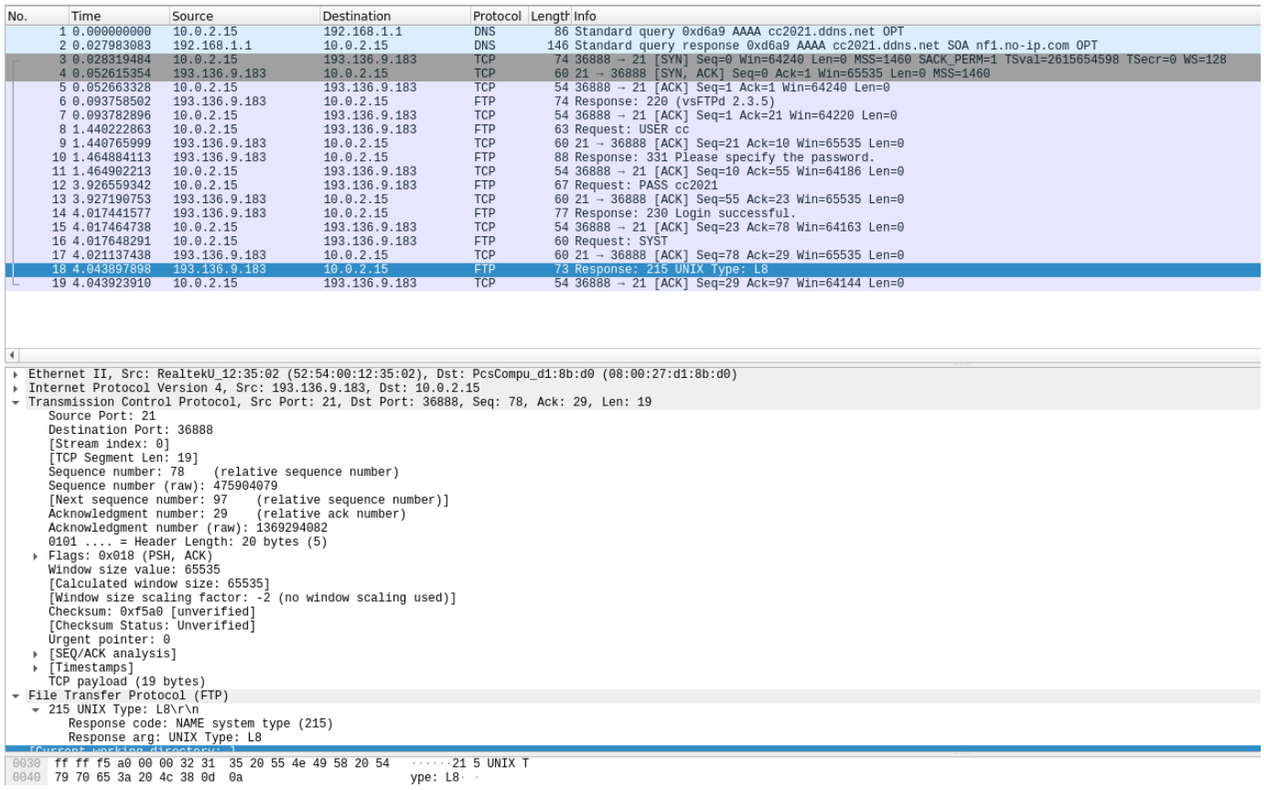

Figure 4: Captura de tráfego ao realizar o ftp

1.5 tftp

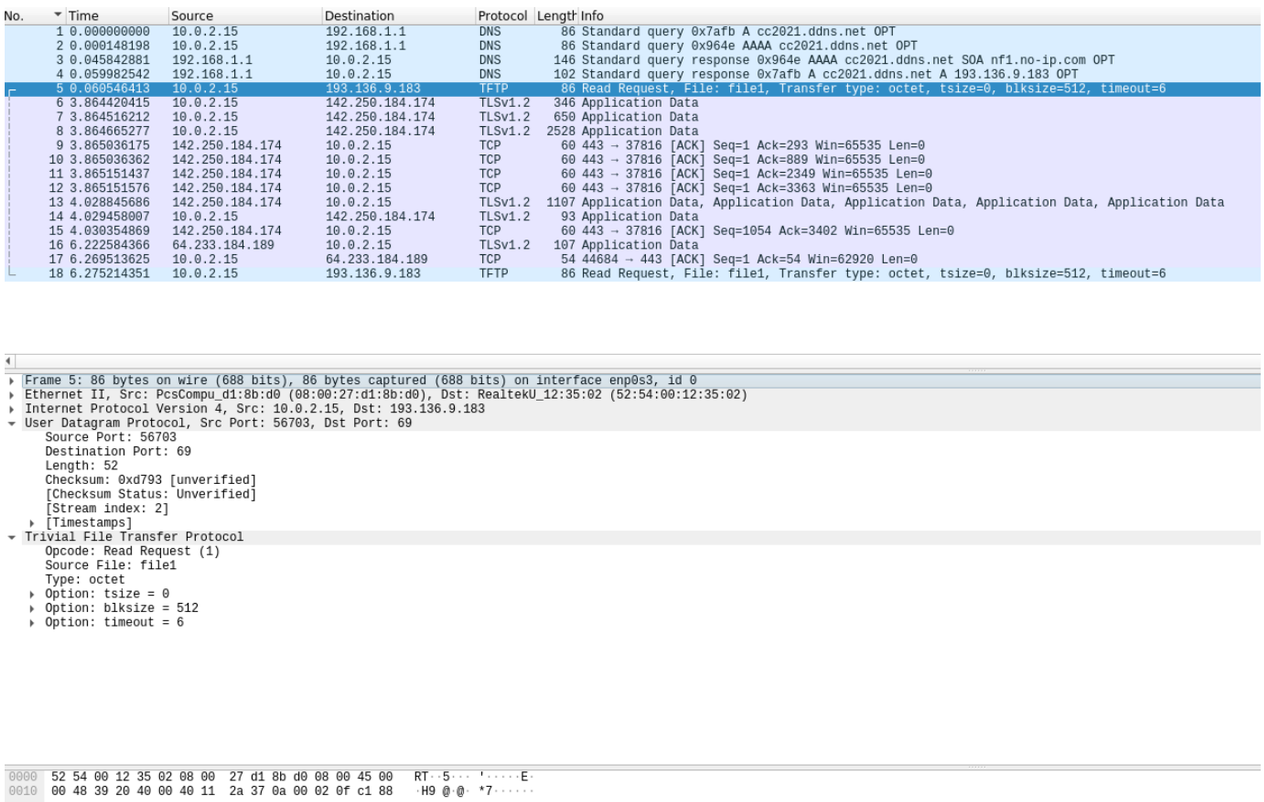

Figure 5: Captura de tráfego ao realizar o tftp

1.6 browser/http

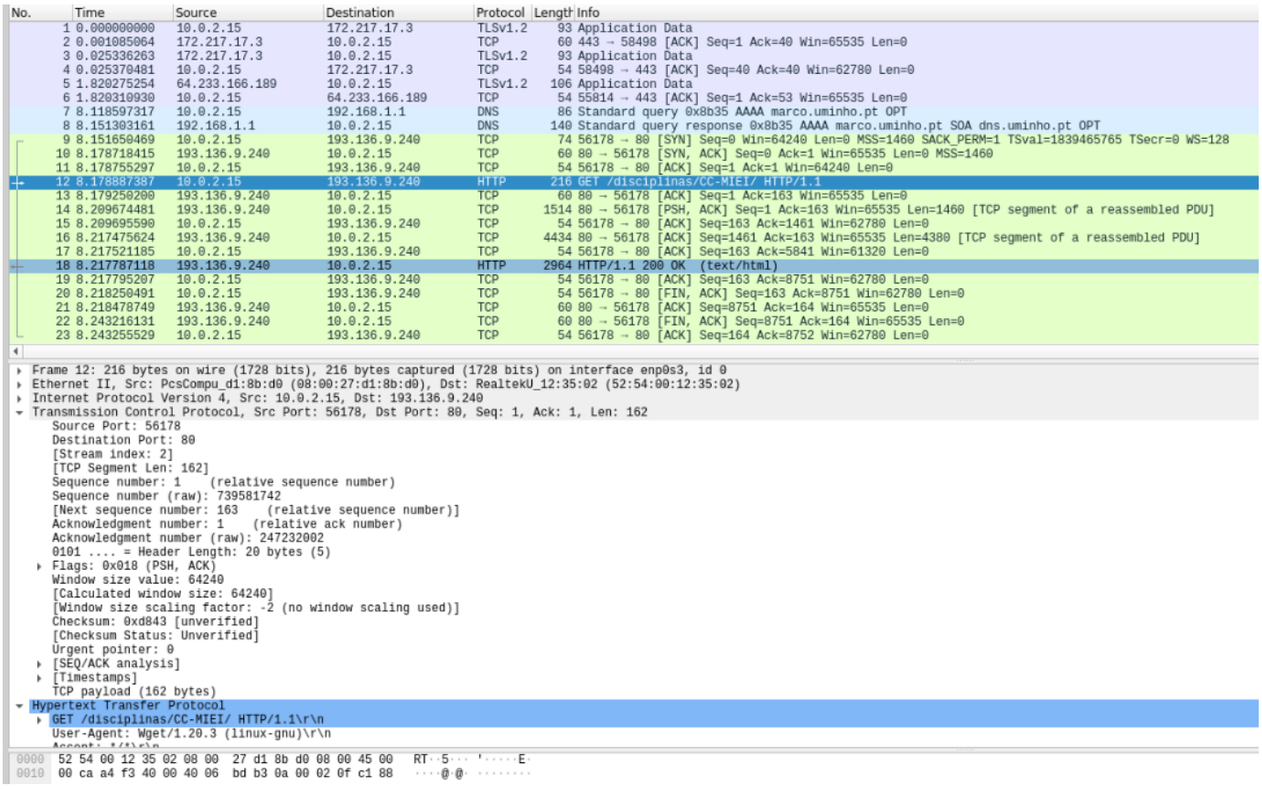

Figure 6: Captura de tráfego ao realizar o browser/http

1.7 nslookup

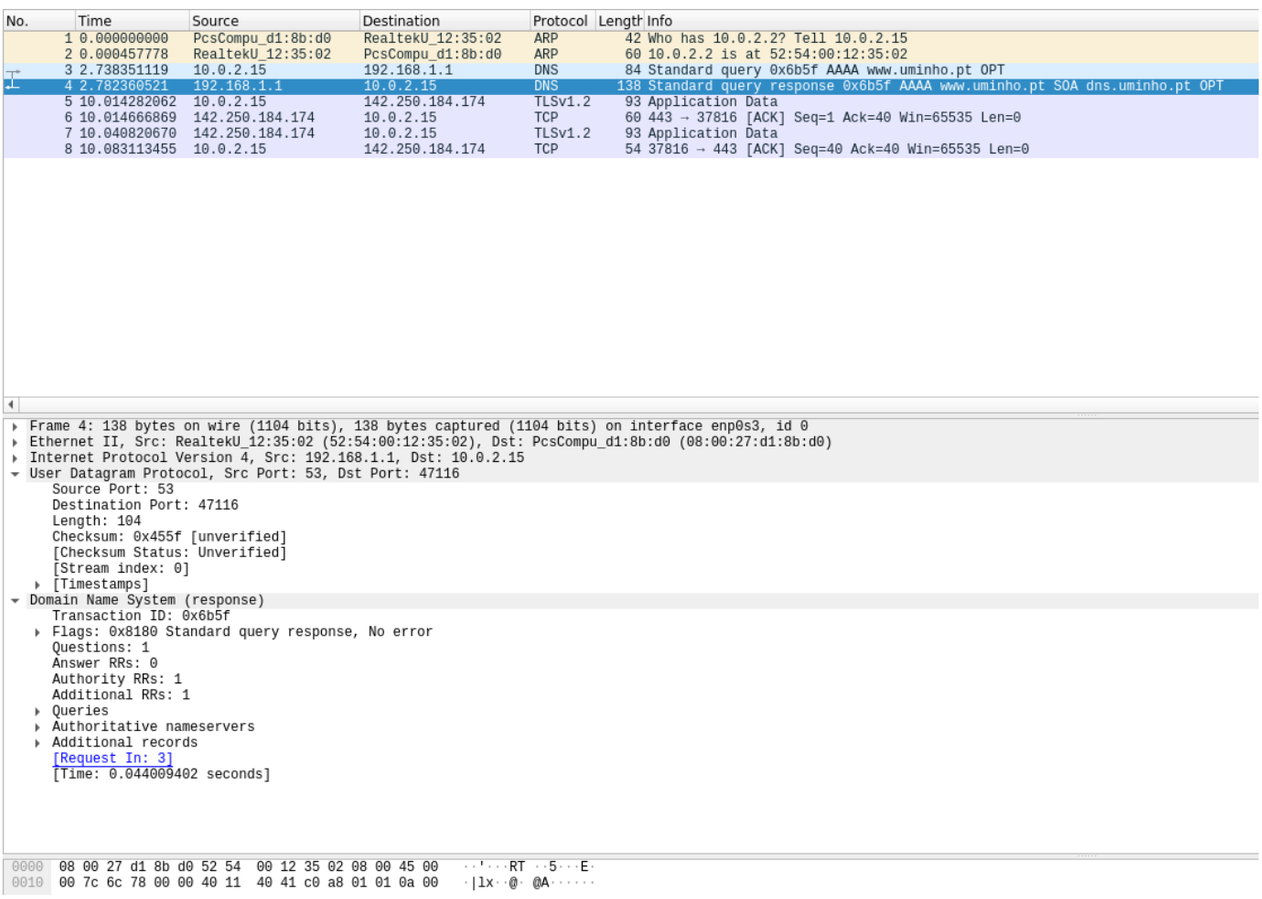

Figure 7: Captura de tráfego ao realizar o nslookup

1.8 ssh

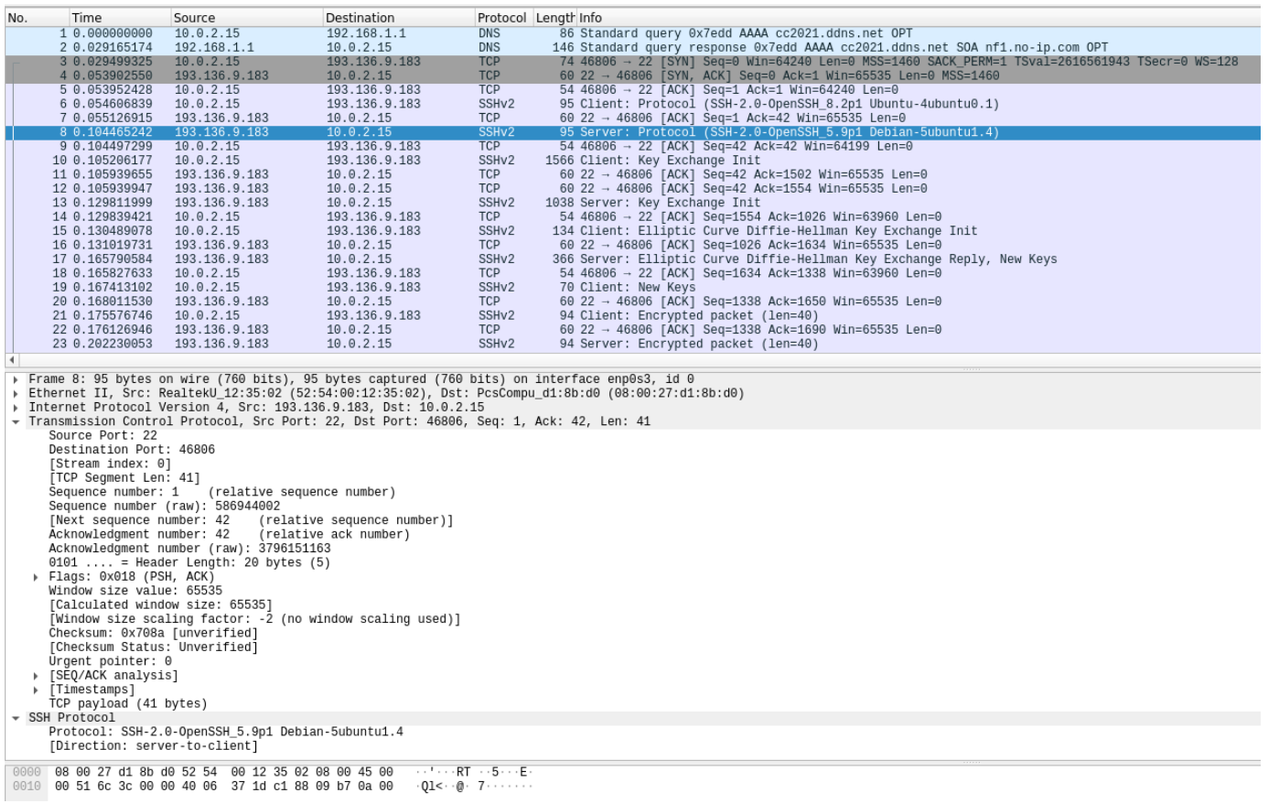

Figure 8: Captura de tráfego ao realizar o ssh

2 Pergunta 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:

2.1 Transferência por ftp

O protocolo ftp, ou *file transfer protocol*, é um protocolo de transferência de ficheiros. Utiliza como protocolo de transporte o TCP e usa um par de conexões entre o cliente e o servidor. A primeira conexão é feita com a porta 21 do servidor, sendo esta a conexão de controle, que permanece aberta até ao fim da sessão. A segunda é a conexão de dados, estabelecida na porta 20 do servidor, utilizada para a transferência de ficheiros.

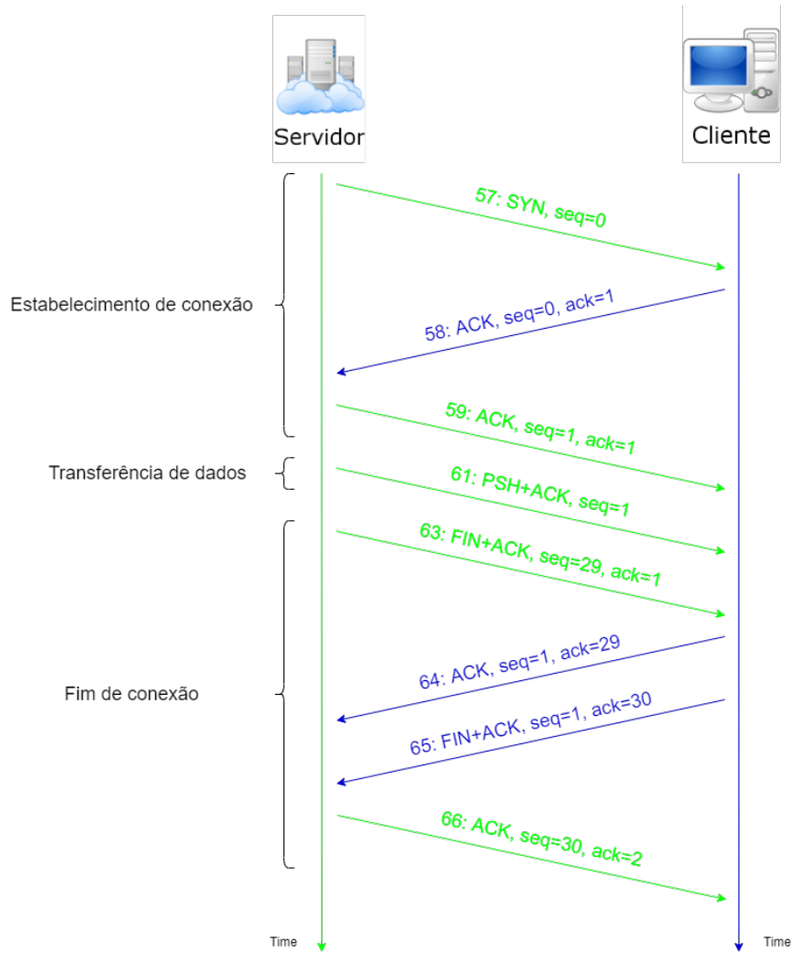

Figure 9: Diagrama de transferência por ftp

2.2 Transferência por tftp

O protocolo tftp, ou *trivial file transfer protocol*, é, como o nome indica, um protocolo de transferência de ficheiros mais simples. Utiliza como protocolo de transporte o UDP. A utilização deste protocolo começa com um pedido do cliente para ler ou escrever num servidor. Este pedido serve também como autenticação. De seguida são enviados os blocos de 512 bytes, sendo que um pacote com menos de 512 bytes significa o fim da transferência. Para finalizar,o cliente envia um ACK ao servidor, para informar que recebeu os pacotes.

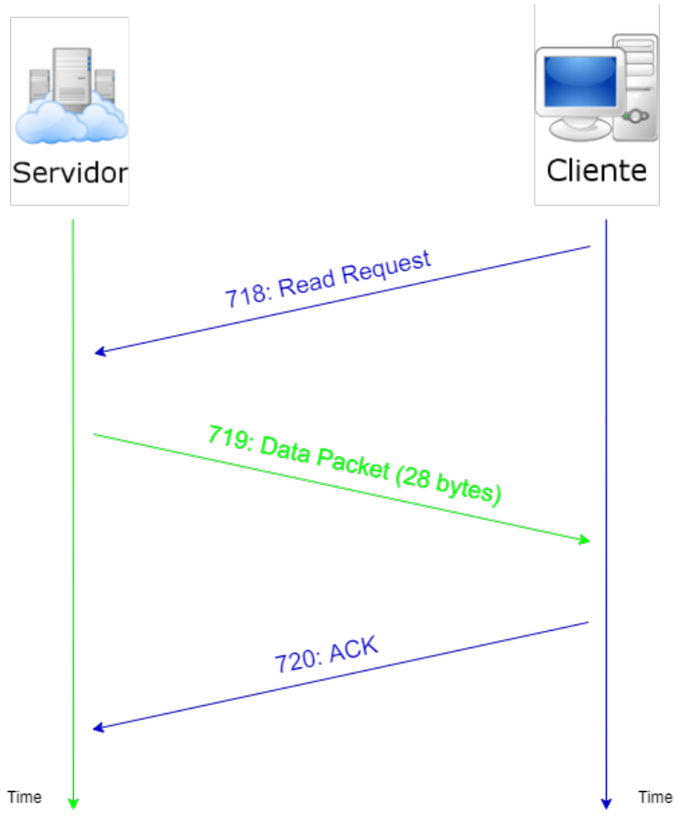

Figure 10: Diagrama de transferência por tftp

2.3 Comparação entre ftp e tftp

Estes protocolos, apesar de terem funções parecidas, devem ser usados com objetivos diferentes. É possível perceber que, devido a necessidade de autenticação e ao uso de TCP, o protocolo ftp é um pouco mais seguro e muito mais confiável. Com o uso de TCP, o protocolo torna-se também mais lento, devido ao *overhead* e ao reenvio de pacotes. Pelo contrário, o protocolo tftp é muito inseguro e não confiavel, mas bem mais rápido, pelo que é geralmente utilizado para transferir pequenos ficheiros entre hosts numa rede.

3 Pergunta 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:

1. **Uso da camada de transporte**

   As 4 aplicações de transferência de ficheiros que usamos não seguem todas a mesma camada de transporte:

   - **STFP:** usa o TCP.
   - **FTP:** usa o TCP.
   - **TFTP:** usa o UDP.
   - **HTTP:** usa o TCP.

2. **Eficiência na transferência**

   - **STFP:** o facto de usar TCP (orientado à conexão), resulta numa transferência de dados mais fiável e ordenado, ou seja muito complexo, logo será menos eficiente, visto que para continuar é sempre preciso ficar à espera de um *acknowledge* e porque é muito pesado devido ao grande overhead.
   - **FTP:** como também usa TCP, a eficiência será parecida ao protocolo STFP. Este protocolo acaba por ser mais eficiente para ficheiros grandes.
   - **TFTP:** ao usar UDP, demonstra ser uma conexão não fiável e desordenado, mas que será mais eficiente devido à rápida transferência de dados (não perde tempo com controlos). Se não conseguir realizar a entrega dos dados, terá que retransmitir as vezes que forem necessárias, o que diminuirá a eficiência.
   - **HTTP:** usa TCP, no entanto revela ser bastante eficiente devido ao uso de *pipelining*. É mais eficiente para ficheiros mais pequenos.

3. **Complexidade**

   - **STFP:** em detrimento de possuir uma forte segurança e de ser um protocolo bastante fiável, este terá um nível de complexidade muito elevado, já que as funções implementadas serão muito custosas.
   - **FTP:** é um protocolo que revela ser fiável na transferência de dados, realizando uma nova conexão a cada transferência, logo será muito complexo.
   - **TFTP:** é a única das 4 que utiliza o UDP como camada de transporte, ou seja podemos desde já concluir que será a menos complexa. É um protocolo simples de fácil implementação e apenas lê e grava ficheiros de ou num servidor remoto.
   - **HTTP:** será um protocolo razoavelmente complexo, pois as suas implementações não são nem muito nem pouco custosas e transmite uma segurança moderada.

4. **Segurança**

   - **STFP:** fornece um canal seguro sobre uma rede insegura numa arquitetura cliente-servidor, conectando uma aplicação cliente SSH com um servidor SSH. Das 4 aplicações de transferência de ficheiros, deu para concluir que esta será a mais fiável, pois é auxiliada pelo protocolo TCP/IP e possui encriptação, autenticação e proteção da integridade dos dados.
   - **FTP:** apesar de suportar autenticação, não é seguro, uma vez que a transferência de ficheiros são feitos sem encriptação, ou seja, a transmissão dos dados é feita pela rede em formato de texto plano o que significa que se alguém intercetar o pacote TCP durante a transmissão, método também conhecido como *Sniffing*, dará ao hacker acesso a todas informações de seus arquivos e credenciais colocando em risco o servidor e o computador local.
   - **TFTP:** é um protocolo simples que não implementa autenticação e não possui a maioria dos recursos de segurança do FTP, logo é fácil prever que não será fiável.
   - **HTTP:** utiliza autenticação, no entanto não é seguro ao nível da transferência de dados, visto que a informação é transferida em texto e não é encriptada, o que pode levar a uma interceção dos dados a meio do caminho. Com o uso de HTTPS, que é um pouco diferente, é possível uma ligação muito mais segura, graças à encriptação da informação entre o servidor e o cliente, à autenticação do servidor e proteje a informação de ser alterada.

4 Pergunta 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).

As situações de perda ou duplicação de pacotes IP nos níveis de transporte e aplicação são bastante normais, mas são também bastante custosos. É por isso um foco de estudo o combate a estas situações. Podemos olhar para este tema de duas maneiras na camada de transporte: o uso de TCP e o uso de UDP.

O TCP é um protocolo que se foca na fiabilidade da comunicação, fazendo a deteção e correção de erros. Apesar disto, utiliza pacotes maiores e é necessário, por vezes, reenviar pacotes, devido ao pacote se perder pelo caminho, ou por estar corrompido fora de alcance de correção. Isto leva a uma diminuição de largura de banda da rede.

Contrariamente, o UDP é um protocolo bem mais leve e menos complexo, com deteção mas sem correção de erros. O facto de serem mais pequenos leva ao facto de serem menos custosos de retransmitir. O grande problema da retransmissão, neste caso, é saber quando e o que retransmitir, sendo que isso não é suportado pelo protocolo.

Podemos observar pela topologia que foi fornecida no enunciado, que numa ligação cem por cento fiável, é melhor usar UDP, pois não haverá erros nos pacotes nem perda de dados que seja preciso retransmissão. Se olharmos para uma situação mais realista, é necessário pensar na prioridade dos objetivos. Se esse objetivo for a velociade de transmissão e não ocupar muita largura de banda, deve-se escolher UDP, tendo em mente que alguns pacotes se irão perder, e a retransmissão será dificil e custosa. Se se der preferência à fiabilidade, deve-se escolher TCP, sabendo que parte da largura de banda estará ocupada com retransmissões, às vezes até desnecessárias.

As diferenças entre estes dois protocolos são demonstradas, na prática, nas figuras 11 e 12.

No.	Time	Source	Destination	Protocol	Length	Info
11	10.391996993	10.4.4.1	10.1.1.1	TCP	74	58512 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=…
12	10.393954896	10.1.1.1	10.4.4.1	TCP	74	21 → 58512 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PE…
13	10.394205986	10.4.4.1	10.1.1.1	TCP	66	58512 → 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3500275906 TSec…
14	10.447022516	10.1.1.1	10.4.4.1	FTP	86	Response: 220 (vsFTPd 3.0.3)
15	10.447421087	10.4.4.1	10.1.1.1	TCP	66	58512 → 21 [ACK] Seq=1 Ack=21 Win=64256 Len=0 TSval=3500275959 TSe…
16	12.008263226	fe80::d459:f8ff:fe5…	ff02::fb	MDNS	107	Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR _ipp…
17	15.627258329	00:00:00_aa:00:14	00:00:00_aa:00:10	ARP	42	Who has 10.1.1.254? Tell 10.1.1.1
18	15.627292555	00:00:00_aa:00:10	00:00:00_aa:00:14	ARP	42	10.1.1.254 is at 00:00:00:aa:00:10
19	18.184503806	fe80::200:ff:feaa:15	ff02::2	ICMPv6	70	Router Solicitation from 00:00:00:aa:00:15
20	18.900914437	10.4.4.1	10.1.1.1	FTP	77	Request: USER core
21	18.901187733	10.1.1.1	10.4.4.1	TCP	66	21 → 58512 [ACK] Seq=21 Ack=12 Win=65280 Len=0 TSval=2158266363 TS…
22	18.901191289	10.1.1.1	10.4.4.1	FTP	100	Response: 331 Please specify the password.
23	18.901373326	10.4.4.1	10.1.1.1	TCP	66	58512 → 21 [ACK] Seq=12 Ack=55 Win=64256 Len=0 TSval=3500284409 TS…
24	20.001438548	10.1.1.254	224.0.0.5	OSPF	78	Hello Packet
25	20.007004402	fe80::200:ff:feaa:10	ff02::5	OSPF	90	Hello Packet
26	20.235620720	fe80::200:ff:feaa:20	ff02::2	ICMPv6	70	Router Solicitation from 00:00:00:aa:00:20
27	21.828489560	10.4.4.1	10.1.1.1	FTP	77	Request: PASS core
28	21.876628079	10.1.1.1	10.4.4.1	TCP	66	21 → 58512 [ACK] Seq=55 Ack=23 Win=65280 Len=0 TSval=2158269333 TS…
29	22.032203116	10.1.1.1	10.4.4.1	FTP	89	Response: 230 Login successful.
30	22.032723583	10.4.4.1	10.1.1.1	TCP	66	58512 → 21 [ACK] Seq=23 Ack=78 Win=64256 Len=0 TSval=3500287539 TS…
31	22.032771976	10.4.4.1	10.1.1.1	FTP	72	Request: SYST
32	22.032968647	10.1.1.1	10.4.4.1	TCP	66	21 → 58512 [ACK] Seq=78 Ack=29 Win=65280 Len=0 TSval=2158269493 TS…

Figure 11: Laptop1 a transferir o file1 através do protocolo FTP

207	208.836693799	10.4.4.1	10.1.1.1	TFTP	56	Read Request, File: file1, Transfer type: octet
208	208.840286022	10.1.1.1	10.4.4.1	TFTP	50	Data Packet, Block: 1 (last)
209	208.840713691	10.4.4.1	10.1.1.1	TFTP	46	Acknowledgement, Block: 1
210	210.016564651	10.1.1.254	224.0.0.5	OSPF	78	Hello Packet
211	210.124310729	fe80::200:ff:feaa:10	ff02::5	OSPF	90	Hello Packet
212	214.025254768	00:00:00_aa:00:10	00:00:00_aa:00:14	ARP	42	Who has 10.1.1.1? Tell 10.1.1.254
213	214.025616187	00:00:00_aa:00:14	00:00:00_aa:00:10	ARP	42	Who has 10.1.1.254? Tell 10.1.1.1
214	214.025618170	00:00:00_aa:00:14	00:00:00_aa:00:10	ARP	42	10.1.1.1 is at 00:00:00:aa:00:14
215	214.025622178	00:00:00_aa:00:10	00:00:00_aa:00:14	ARP	42	10.1.1.254 is at 00:00:00:aa:00:10
216	216.435286692	10.4.4.1	10.1.1.1	TFTP	56	Read Request, File: file2, Transfer type: octet
217	216.435765996	10.1.1.1	10.4.4.1	TFTP	52	Data Packet, Block: 1 (last)
218	216.436135761	10.4.4.1	10.1.1.1	TFTP	46	Acknowledgement, Block: 1
219	220.016650402	10.1.1.254	224.0.0.5	OSPF	78	Hello Packet
220	220.139905662	fe80::200:ff:feaa:10	ff02::5	OSPF	90	Hello Packet

Figure 12: Laptop1 a transferir o file1 através do protocolo TFTP

Deste modo, podemos observar que as caracteristicas das ligações de rede são informações vitais à comunicação, e irão fazer variar a escolha dos protocolos que devemos utilizar.

Conclusão

Com a realização deste trabalho conseguimos consolidar, numa vertente mais prática, diversos conceitos relativos aos diversos protocolos das camadas de Transporte e de Aplicação.

Um dos pontos fulcrais a retirar é que, apesar de o protocolo TCP apresentar um nível de fiabilidade bastante superior ao do UDP, garantindo sempre que todos os pacotes enviados chegam ao destino, este último pode ser uma alternativa viável em situações nas quais a perda de pacotes não é extremamente grave sendo a máxima velocidade de transmissão o objetivo fulcral. O protocolo UDP é mais do que suficiente para lidar com situações em tempo real, como por exemplo, o streaming.

Explorámos também as diferenças ao nível da camada de Aplicação, nomeadamente, os protocolos SFTP, FTP, TFTP e HTTP.