

# Comunicações por Computador

a87983 Pedro Pinto  
a100659 Rui Pinto  
a100066 Ricardo Jesus  
Grupo 3 - PL7

Setembro 2023

## 1 Questões (Parte I)

### 1.1 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.

As perdas e duplicações de pacotes podem afetar significativamente o desempenho das aplicações numa rede. Quando ocorre a perda de pacotes, o receptor não consegue receber a mensagem corretamente, o que leva o emissor a retransmitir a mensagem após um período de espera. Isso resulta em atrasos na comunicação entre as aplicações, uma vez que a entrega da informação é adiada devido à necessidade de retransmissão.

No caso das duplicações, quando uma mensagem é duplicada na rede, o emissor recebe múltiplas cópias da mesma mensagem. Isso não apenas causa atrasos na comunicação, semelhantes à perda de pacotes, mas também exige que o emissor processe e responda a essas duplicações, dependendo quantas delas ocorreram. Isso pode sobrecarregar a aplicação receptora dos pacotes e piorar ainda mais o atraso na comunicação.

Em relação à camada que lida com esses problemas, é a camada de transporte que desempenha este papel. O protocolo de transporte é responsável por detectar e lidar com perdas de pacotes, garantindo a entrega confiável e a ordenação correta das mensagens. Isso ajuda a minimizar os impactos das perdas e duplicações de pacotes no desempenho das aplicações, tornando a comunicação mais confiável e eficiente.


A horizontal bar representing a network packet capture entry. It contains the following text: 401.432.04855022 10.4.4.1 10.2.2.1 TCP 66 [TCP Dup ACK 398#1] 20 -> 56223 [ACK] Seq=226 Ack=2 Win=64256 Len=0

Figura 1: Reconhecimento de um pacote duplicado na utilização do FTP, a ser lido pelo TCP.

- 1.2 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 configurações.

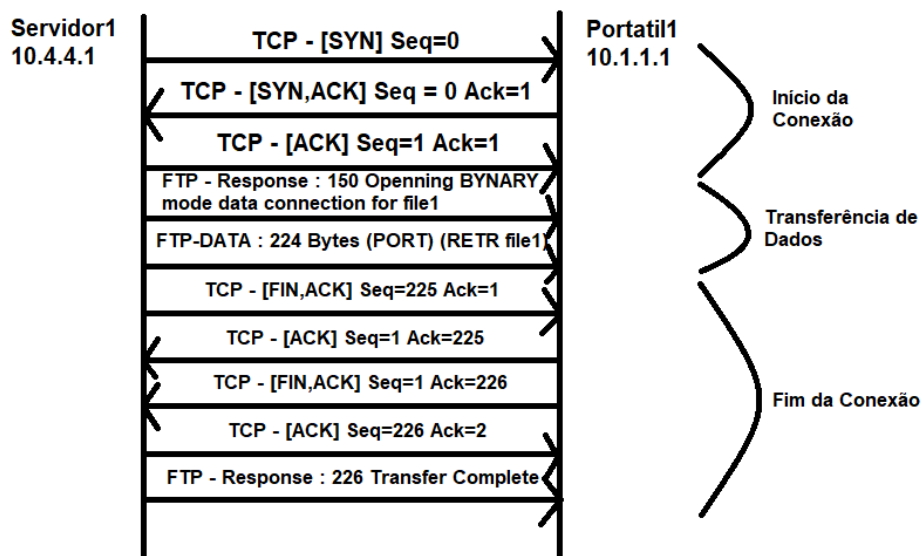


Figura 2: Diagrama temporal para a transferência do file1 por FTP

255	387.758129548	10.4.4.1	10.1.1.1	TCP	74 20 - 43679 [SYN] Seq=0 Win=64248 Len=0 MSS=1460 SACK_PERM=1 TSval=...
256	387.752241946	10.1.1.1	10.4.4.1	TCP	74 43679 - 20 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=...
257	387.752467166	10.4.4.1	10.1.1.1	TCP	66 20 - 43679 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2885029749 TSecr=...
258	387.752681150	10.4.4.1	10.1.1.1	FTP	130 Response: 150 Opening BINARY mode data connection for file1 (224 b...
259	387.753132748	10.4.4.1	10.1.1.1	FTP-DATA	290 FTP Data: 224 bytes (PORT) (RETR file1)
260	387.753138739	10.4.4.1	10.1.1.1	TCP	66 20 - 43679 [FIN, ACK] Seq=225 Ack=1 Win=64256 Len=0 TSval=2885029749 TSecr=...
261	387.753607391	10.1.1.1	10.4.4.1	TCP	66 43679 - 20 [ACK] Seq=1 Ack=225 Win=65024 Len=0 TSval=3638432804 TSecr=...
262	387.754671199	10.1.1.1	10.4.4.1	TCP	66 43679 - 20 [FIN, ACK] Seq=1 Ack=226 Win=65024 Len=0 TSval=3638432804 TSecr=...
263	387.754851337	10.4.4.1	10.1.1.1	TCP	66 20 - 43679 [ACK] Seq=226 Ack=2 Win=64256 Len=0 TSval=2885029751 TSecr=...
264	387.755081947	10.4.4.1	10.1.1.1	FTP	90 Response: 226 Transfer complete.

Figura 3: Exemplo de Tramas na transferência do file1 por FTP no Wireshark

**1.3** 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

Na generalidade, uma conexão para transferência de dados é estabelecida através do uso de protocolos TCP. No entanto, neste caso, utiliza-se o UDP . Nesse sentido, o ficheiro foi transferido, na totalidade, num bloco, e, por esse motivo, só houve intereção entre a troca e não são explicitados quaisquer números de sequência.

846	1161.06622657	10.1.1.1	10.4.4.1	TFTP	56 Read Request, File: file1, Transfer type: octet
847	1161.068265061	10.4.4.1	10.1.1.1	TFTP	270 Data Packet, Block: 1 (last)
848	1161.070104655	10.1.1.1	10.4.4.1	TFTP	46 Acknowledgement, Block: 1

Figura 4: Transferência do file1 por TFTP

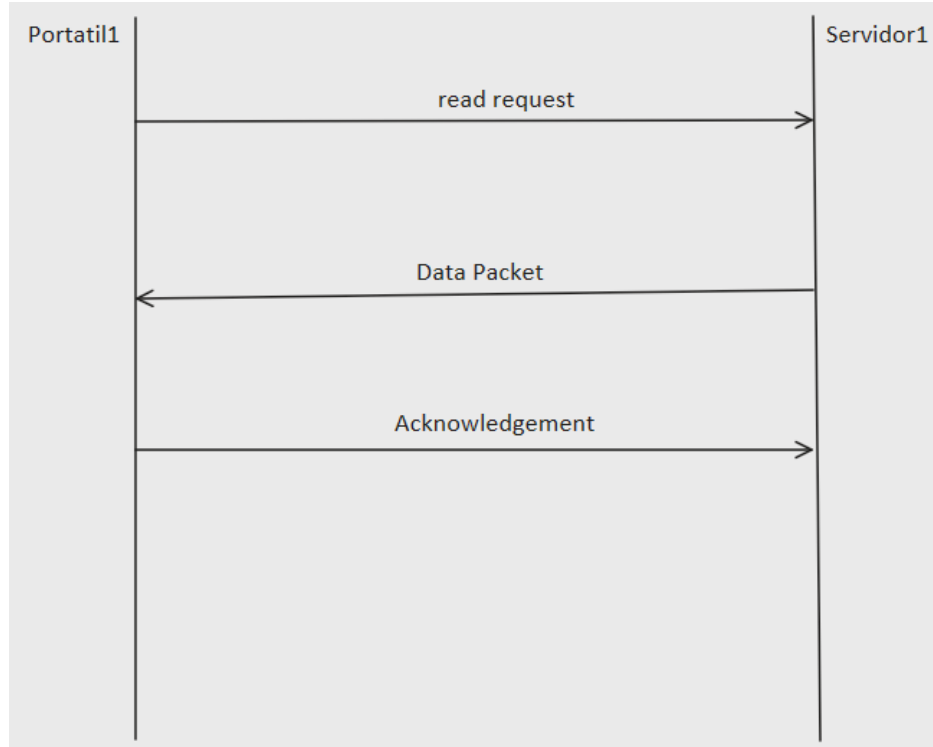


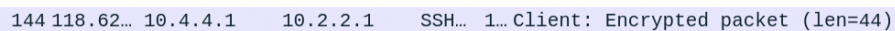
Figura 5: Diagrama da transferência do file1 por TFTP

#### 1.4 Compare sucintamente as quatro aplicações de transferência de ficheiros que usou, tendo em consideração os seguintes aspetos: (i) uso da camada de transporte; (ii) eficiência; (iii) complexidade; (iv) segurança.

Os protocolos de transferência de ficheiros utilizados neste trabalho foram: *sftp*, *ftp*, *tftp* e *http*. Quanto ao uso da camada de transport, eficiência, complexidade e segurança, temos:

##### **SFTP(SSH File Transfer Protocol):**

- **Uso da camada de transporte:** Utiliza o protocolo TCP. Os pacotes de dados são enviados no mesmo canal do que os pacotes de sincronização e controlo.
- **Eficiência:** Menos eficiente do que o FTP e do TFTP, dado a não ter múltiplos canais e não usar UDP respetivamente.
- **Complexidade:** É um protocolo poderoso com a capacidade de gerir os ficheiros e diretórios numa máquina remota.
- **Segurança:** Bastante seguro com autenticação de *username/password* e SSH Key. Para além disso encripta os dados enviados.



144 118.62... 10.4.4.1 10.2.2.1 SSH... 1... Client: Encrypted packet (len=44)

Figura 6: Pacote SSH encriptado.

### FTP(File Transfer Protocol):

- **Uso da camada de transporte:** Utiliza o protocolo TCP. Os pacotes de dados são enviados para a porta 20 e os pacotes de sincronização e controle para a porta 21.

```
Transmission Control Protocol, Src Port: 20, Dst Port: 47913, Seq: 1, Ack: 1, Len: 1134
FTP Data (1134 bytes data)
```

Figura 7: FTP: Dados enviados para a porta 20.

```
Transmission Control Protocol, Src Port: 35394, Dst Port: 21, Seq: 62, Ack: 230, Len: 0
```

Figura 8: FTP: Dados enviados para a porta 21.

- **Eficiência:** mais eficiente dos três protocolos que usam tcp(sftp,ftp e http), pois utiliza dois canais(um para dados e o outro para controle).
- **Complexidade:** Complexidade ao nível do sftp, pois também permite manipulação de ficheiros e diretorias *file system*.
- **Segurança:** Usa autenticação *username/password*, mas não encripta os dados.

### TFTP(Trivial File Transfer Protocol):

- **Uso da camada de transporte:** Utiliza o protocolo UDP.

```
User Datagram Protocol, Src Port: 49291, Dst Port: 69
Trivial File Transfer Protocol
```

Figura 9: TFTP using UDP

- **Eficiência:** A utilização de UDP torna o TFTP bastante eficiente, pois é um protocolo menos complexo e há um potencial *output* constante de dados, dado que não há esperas de respostas como no TCP.
- **Complexidade:** O UDP é um protocolo simples, cuja complexidade é movida para a camada aplicacional para gestão de pacotes, o que não permite ao TFTP manipular ficheiros/diretorias e o que resulta na necessidade de fornecer o *path* para o ficheiro em questão.
- **Segurança:** Não possui qualquer mecanismo de segurança.

### HTTP(Hyper Text Transfer Protocol):

- **Uso da camada de transporte:** Utiliza o protocolo TCP.
- **Eficiência:** Protocolo menos eficiente (na maioria das situações) dado ao seu grande overhead e consequente processamento.
- **Complexidade:** Com o sistema de requests consegue manipular ficheiros/diretorias (ex:PUT,REMOVE,etc.). Para além disso suporta também várias sessões ao mesmo tempo.

```
84 133.48... 10.4.4.1 10.2.2.1 HTTP 206 GET /file1 HTTP/1.1
```

Figura 10: Pedido HTTP GET.

- **Segurança:** Tem possibilidade de autenticação, mas não é obrigatório.

## 2 Questões (Parte II)

- 2.1 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.

Aplicação	Protocolo de Aplicação	Protocolo de transporte	Porta de Atendimento	Overhead de transporte em bytes
wget, linx ou via browser	DNS/http	TCP	80	20
ssh, sftp	ssh	TCP	22	20
ftp	ftp	TCP	21	20
Tftp	tftp	UDP	69	8
telnet	telnet	TCP	23	20
nslookup ou dig	DNS	UDP	53	8
Ping	----	----	----	----
Traceroute	DNS	UDP	53	8

### 2.1.1 Wget ou via browser

659	1061.9262891..	10.0.2.15	193.136.152.72	NTP	90 NTP Version 4, client
660	1061.9351419..	193.136.152.72	10.0.2.15	NTP	90 NTP Version 4, server
661	1063.9254242..	10.0.2.15	162.159.200.123	NTP	90 NTP Version 4, client
662	1063.9348884..	162.159.200.123	10.0.2.15	NTP	90 NTP Version 4, server
663	1071.4623162..	10.0.2.15	192.168.1.254	DNS	86 Standard query 0xb6fb A marco.uminho.pt OPT
664	1071.4239269..	10.0.2.15	192.168.1.254	DNS	86 Standard query 0x95a6 AAAA marco.uminho.pt OPT
665	1071.4491582..	192.168.1.254	10.0.2.15	DNS	149 Standard query response 0x95a6 AAAA marco.uminho.pt SOA dns.u...
666	1071.4491585..	192.168.1.254	10.0.2.15	DNS	102 Standard query response 0xb6fb A marco.uminho.pt A 193.136.9...
667	1071.4496932..	10.0.2.15	193.136.9.240	TCP	74 45400 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 T...
668	1071.4674241..	193.136.9.240	10.0.2.15	TCP	60 80 → 45400 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
669	1071.4674569..	10.0.2.15	193.136.9.240	TCP	54 45400 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
670	1071.4678935..	10.0.2.15	193.136.9.240	HTTP	215 GET /disciplinas/CC-LEI/ HTTP/1.1
671	1071.4678928..	193.136.9.240	10.0.2.15	TCP	60 80 → 45400 [ACK] Seq=1 Ack=162 Win=65535 Len=0
672	1071.4823142..	193.136.9.240	10.0.2.15	TCP	2974 80 → 45400 [ACK] Seq=1 Ack=162 Win=65535 Len=2920 [TCP segmen...
673	1071.4823454..	10.0.2.15	193.136.9.240	TCP	54 45400 → 80 [ACK] Seq=162 Ack=2921 Win=62780 Len=0
674	1071.4824907..	193.136.9.240	10.0.2.15	TCP	4434 80 → 45400 [ACK] Seq=2921 Ack=162 Win=65535 Len=4380 [TCP seg...
675	1071.4825919..	10.0.2.15	193.136.9.240	TCP	54 45400 → 80 [ACK] Seq=162 Ack=7301 Win=59860 Len=0
676	1071.4826188..	193.136.9.240	10.0.2.15	HTTP	1773 HTTP/1.1 200 OK (text/html)
677	1071.4826189..	10.0.2.15	193.136.9.240	TCP	54 45400 → 80 [ACK] Seq=162 Ack=9020 Win=58400 Len=0
678	1071.4832168..	10.0.2.15	193.136.9.240	TCP	54 45400 → 80 [FIN, ACK] Seq=162 Ack=9020 Win=62780 Len=0
679	1071.4834210..	193.136.9.240	10.0.2.15	TCP	60 80 → 45400 [ACK] Seq=9020 Ack=163 Win=65535 Len=0
680	1071.5002487..	193.136.9.240	10.0.2.15	TCP	60 80 → 45400 [FIN, ACK] Seq=9020 Ack=163 Win=65535 Len=0
681	1071.5002873..	10.0.2.15	193.136.9.240	SSHv2	54 45400 → 80 [ACK] Seq=163 Ack=9021 Win=62780 Len=0
Frame 670: 215 bytes on wire (1720 bits), 215 bytes captured (1720 bits) on interface enp8s3, id 0					
Ethernet II, Src: PcsCompu_06:03:48 (08:00:27:06:03:48), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)					
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 193.136.9.240					
Transmission Control Protocol, Src Port: 45400, Dst Port: 80, Seq: 1, Ack: 1, Len: 161					
Hypertext Transfer Protocol					

Figura 11: Wget

### 2.1.2 ssh

300	521.753137354	10.0.2.15	193.136.9.201	SSHv2	95 Client: Protocol (SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.3)
301	521.753662064	193.136.9.201	10.0.2.15	TCP	60 22 → 57022 [ACK] Seq=1 Ack=42 Win=65535 Len=0
302	521.812155173	193.136.9.201	10.0.2.15	SSHv2	95 Server: Protocol (SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.4)
303	521.812193548	10.0.2.15	193.136.9.201	TCP	54 57022 → 22 [ACK] Seq=42 Ack=42 Win=64199 Len=0
304	521.812962444	10.0.2.15	193.136.9.201	SSHv2	1566 Client: Key Exchange Init
305	521.813495275	193.136.9.201	10.0.2.15	TCP	60 22 → 57022 [ACK] Seq=42 Ack=1502 Win=65535 Len=0
306	521.813495866	193.136.9.201	10.0.2.15	TCP	60 22 → 57022 [ACK] Seq=42 Ack=1554 Win=65535 Len=0
307	521.815447170	193.136.9.201	10.0.2.15	SSHv2	1134 Server: Key Exchange Init
308	521.820018963	10.0.2.15	193.136.9.201	SSHv2	102 Client: Diffie-Hellman Key Exchange Init
309	521.829589595	193.136.9.201	10.0.2.15	TCP	60 22 → 57022 [ACK] Seq=1122 Ack=1602 Win=65535 Len=0
310	521.847035663	193.136.9.201	10.0.2.15	SSHv2	650 Server: Diffie-Hellman Key Exchange Reply, New Keys, Encrypte...
311	521.854443078	10.0.2.15	193.136.9.201	SSHv2	70 Client: New Keys
312	521.855358549	193.136.9.201	10.0.2.15	TCP	60 22 → 57022 [ACK] Seq=1718 Ack=1618 Win=65535 Len=0
313	521.856012181	10.0.2.15	193.136.9.201	SSHv2	98 Client: Encrypted packet (len=44)
314	521.856567611	193.136.9.201	10.0.2.15	TCP	60 22 → 57022 [ACK] Seq=1718 Ack=1662 Win=65535 Len=0
315	521.879539319	193.136.9.201	10.0.2.15	SSHv2	98 Server: Encrypted packet (len=44)
316	521.880148995	10.0.2.15	193.136.9.201	SSHv2	114 Client: Encrypted packet (len=60)
317	521.880617176	193.136.9.201	10.0.2.15	TCP	60 22 → 57022 [ACK] Seq=1762 Ack=1722 Win=65535 Len=0
318	521.909774928	193.136.9.201	10.0.2.15	SSHv2	106 Server: Encrypted packet (len=52)
319	521.967404896	10.0.2.15	193.136.9.201	TCP	54 57022 → 22 [ACK] Seq=1722 Ack=1814 Win=63720 Len=0
320	526.093840284	10.0.2.15	193.136.9.201	SSHv2	138 Client: Encrypted packet (len=84)
321	526.094507082	193.136.9.201	10.0.2.15	TCP	60 22 → 57022 [ACK] Seq=1814 Ack=1806 Win=65535 Len=0
322	526.230284919	193.136.9.201	10.0.2.15	SSHv2	82 Server: Encrypted packet (len=28)
323	526.230327863	10.0.2.15	193.136.9.201	TCP	54 57022 → 22 [ACK] Seq=1806 Ack=1842 Win=63720 Len=0
324	526.230852083	10.0.2.15	193.136.9.201	SSHv2	166 Client: Encrypted packet (len=112)
325	526.231327327	193.136.9.201	10.0.2.15	TCP	60 22 → 57022 [ACK] Seq=1842 Ack=1918 Win=65535 Len=0
326	526.272838375	193.136.9.201	10.0.2.15	SSHv2	682 Server: Encrypted packet (len=628)
327	526.316380452	10.0.2.15	193.136.9.201	TCP	54 57022 → 22 [ACK] Seq=1818 Ack=1918 Win=63720 Len=0
Frame 304: 1566 bytes on wire (12528 bits), 1566 bytes captured (12528 bits) on interface enp8s3, id 0					
Ethernet II, Src: PcsCompu_06:03:48 (08:00:27:06:03:48), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)					
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 193.136.9.201					
Transmission Control Protocol, Src Port: 57022, Dst Port: 22, Seq: 42, Ack: 42, Len: 1512					
Source Port: 57022					
Destination Port: 22					
[Stream index: 7]					
[TCP Segment Len: 1512]					
Sequence number: 42 (relative sequence number)					
Sequence number (raw): 3880304592					
[Next sequence number: 1554 (relative sequence number)]					
Acknowledgment number: 42 (relative ack number)					
Acknowledgment number (raw): 26368043					
0101 .... = Header Length: 20 bytes (5)					

Figura 12: ssh



### 2.1.3 ftp

485	760.505832718	193.137.214.36	10.0.2.15	FTP	74	Response: 220 (vsFTpd 3.0.3)
486	760.505880711	10.0.2.15	193.137.214.36	TCP	54	50020 → 21 [ACK] Seq=1 Ack=21 Win=64220 Len=0
487	765.526737054	PcsCompu_06:03:48	RealtekU_12:35:02	ARP	42	Who has 10.0.2.2? Tell 10.0.2.15
488	765.527356961	RealtekU_12:35:02	PcsCompu_06:03:48	ARP	60	10.0.2.2 is at 52:54:00:12:35:02
489	766.364277345	10.0.2.15	193.137.214.36	FTP	64	Request: USER ftp
490	766.365004441	193.137.214.36	10.0.2.15	TCP	60	21 → 50020 [ACK] Seq=21 Ack=11 Win=65535 Len=0
491	766.379413733	193.137.214.36	10.0.2.15	FTP	88	Response: 331 Please specify the password.
492	766.379449022	10.0.2.15	193.137.214.36	TCP	54	50020 → 21 [ACK] Seq=11 Ack=55 Win=64186 Len=0
493	791.580466682	10.0.2.15	193.137.214.36	FTP	68	Request: PASS cc2023
494	791.581241542	193.137.214.36	10.0.2.15	TCP	60	21 → 50020 [ACK] Seq=55 Ack=25 Win=65535 Len=0
495	791.600131693	193.137.214.36	10.0.2.15	FTP	465	Response: 230-Welcome, archive user of ftp.eq.uc.pt!
496	791.600172633	10.0.2.15	193.137.214.36	TCP	54	50020 → 21 [ACK] Seq=25 Ack=466 Win=63784 Len=0
497	791.600432118	10.0.2.15	193.137.214.36	FTP	60	Request: SYST
498	791.601133244	193.137.214.36	10.0.2.15	TCP	60	21 → 50020 [ACK] Seq=466 Ack=31 Win=65535 Len=0
499	791.620393917	193.137.214.36	10.0.2.15	FTP	73	Response: 215 UNIX Type: L8
500	791.663043072	10.0.2.15	193.137.214.36	TCP	54	50020 → 21 [ACK] Seq=31 Ack=485 Win=63784 Len=0
501	796.758423883	PcsCompu_06:03:48	RealtekU_12:35:02	ARP	42	Who has 10.0.2.2? Tell 10.0.2.15
502	796.758857287	RealtekU_12:35:02	PcsCompu_06:03:48	ARP	60	10.0.2.2 is at 52:54:00:12:35:02
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 193.137.214.36						
Transmission Control Protocol, Src Port: 50020, Dst Port: 21, Seq: 1, Ack: 21, Len: 10						
Source Port: 50020						
Destination Port: 21						
[Stream Index: 9]						
[TCP Segment Len: 10]						
Sequence number: 1 (relative sequence number)						
Sequence number (raw): 1779297805						
[Next sequence number: 11 (relative sequence number)]						
Acknowledgment number: 21 (relative ack number)						
Acknowledgment number (raw): 29632022						
0101 .... = Header Length: 20 bytes (5)						
Flags: 0x018 (PSH, ACK)						
Window size value: 64220						
[Calculated window size: 64220]						
[Window size scaling factor: -2 (no window scaling used)]						
Checksum: 0xa3e1 [unverified]						
[Checksum Status: Unverified]						
Urgent pointer: 0						
[SEQ/ACK analysis]						
[Timestamps]						
TCP payload (10 bytes)						
File Transfer Protocol (FTP)						

Figura 13: ftp

### 2.1.4 tftp

903	1715.9746276...	185.125.190.57	10.0.2.15	NTP	90	NTP Version 4, server
904	1729.9254286...	10.0.2.15	88.157.128.22	NTP	90	NTP Version 4, client
905	1729.9254623...	10.0.2.15	91.209.16.78	NTP	90	NTP Version 4, client
906	1729.9356776...	88.157.128.22	10.0.2.15	NTP	90	NTP Version 4, server
907	1729.9356780...	91.209.16.78	10.0.2.15	NTP	90	NTP Version 4, server
908	1735.9251938...	10.0.2.15	193.136.152.72	NTP	90	NTP Version 4, client
909	1735.9353248...	193.136.152.72	10.0.2.15	NTP	90	NTP Version 4, server
910	1737.9259744...	10.0.2.15	162.159.200.123	NTP	90	NTP Version 4, client
911	1737.9351686...	162.159.200.123	10.0.2.15	NTP	90	NTP Version 4, server
912	1738.9255116...	10.0.2.15	185.125.190.56	NTP	90	NTP Version 4, client
913	1738.9682979...	185.125.190.56	10.0.2.15	NTP	90	NTP Version 4, server
914	1739.9259394...	10.0.2.15	91.189.91.157	NTP	90	NTP Version 4, client
915	1739.9259774...	10.0.2.15	185.125.190.58	NTP	90	NTP Version 4, client
916	1739.9669258...	185.125.190.58	10.0.2.15	NTP	90	NTP Version 4, server
917	1745.0783562...	PcsCompu_06:03:48	RealtekU_12:35:02	ARP	42	Who has 10.0.2.2? Tell 10.0.2.15
918	1745.0787274...	RealtekU_12:35:02	PcsCompu_06:03:48	ARP	60	10.0.2.2 is at 52:54:00:12:35:02
919	1763.8658835...	10.0.2.15	192.168.1.254	DNS	86	Standard query 0x3400 A cc2023.ddns.net OPT
920	1763.8661049...	10.0.2.15	192.168.1.254	DNS	86	Standard query 0x68c3 AAAA cc2023.ddns.net OPT
921	1763.9079622...	192.168.1.254	10.0.2.15	DNS	146	Standard query response 0x68c3 AAAA cc2023.ddns.net SOA nfi.n...
922	1763.9079626...	192.168.1.254	10.0.2.15	DNS	102	Standard query response 0x3400 A cc2023.ddns.net A 193.136.9...
923	1763.9092608...	10.0.2.15	193.136.9.201	TFTP	86	Read Request, File: file1, Transfer type: octet, tsize=0, blk...
924	1770.1173812...	10.0.2.15	193.136.9.201	TFTP	86	Read Request, File: file1, Transfer type: octet, tsize=0, blk...
925	1770.9254198...	10.0.2.15	194.8.30.16	NTP	90	NTP Version 4, client
926	1770.9424231...	194.8.30.16	10.0.2.15	NTP	90	NTP Version 4, server
Frame 919: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface enp0s3, id 0						
Ethernet II, Src: PcsCompu_06:03:48 (08:00:27:06:03:48), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)						
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 192.168.1.254						
User Datagram Protocol, Src Port: 53038, Dst Port: 53						
Source Port: 53038						
Destination Port: 53						
Length: 52						
Checksum: 0xcefa [unverified]						
[Checksum Status: Unverified]						
[Stream Index: 80]						
[Timestamps]						
Domain Name System (query)						

Figura 14: tftp

## 2.1.5 Telnet

133	34.155430639	193.136.9.33	10.0.2.15	TCP	60 23 → 51246 [ACK] Seq=496 Ack=61 Win=65535 Len=0
134	34.174109313	193.136.9.33	10.0.2.15	TELNET	60 Telnet Data ...
135	34.218083979	10.0.2.15	193.136.9.33	TCP	54 51246 → 23 [ACK] Seq=61 Ack=497 Win=63784 Len=0
136	34.429297863	10.0.2.15	193.136.9.33	TELNET	55 Telnet Data ...
137	34.430958732	193.136.9.33	10.0.2.15	TCP	60 23 → 51246 [ACK] Seq=497 Ack=62 Win=65535 Len=0
138	34.450935061	193.136.9.33	10.0.2.15	TELNET	60 Telnet Data ...
139	34.450935797	10.0.2.15	193.136.9.33	TCP	54 51246 → 23 [ACK] Seq=62 Ack=498 Win=63784 Len=0
140	34.609256159	10.0.2.15	193.136.9.33	TELNET	55 Telnet Data ...
141	34.609866859	193.136.9.33	10.0.2.15	TCP	60 23 → 51246 [ACK] Seq=498 Ack=63 Win=65535 Len=0
142	34.608337369	193.136.9.33	10.0.2.15	TELNET	60 Telnet Data ...
143	34.608378743	10.0.2.15	193.136.9.33	TCP	54 51246 → 23 [ACK] Seq=63 Ack=499 Win=63784 Len=0
144	34.803883977	10.0.2.15	193.136.9.33	TELNET	55 Telnet Data ...
145	34.864485745	193.136.9.33	10.0.2.15	TCP	60 23 → 51246 [ACK] Seq=499 Ack=64 Win=65535 Len=0
146	34.884263196	193.136.9.33	10.0.2.15	TELNET	60 Telnet Data ...
147	34.884294623	10.0.2.15	193.136.9.33	TCP	54 51246 → 23 [ACK] Seq=64 Ack=500 Win=63784 Len=0
148	35.180558355	10.0.2.15	193.136.9.33	TELNET	50 Telnet Data ...
149	35.181109414	193.136.9.33	10.0.2.15	TCP	60 23 → 51246 [ACK] Seq=500 Ack=66 Win=65535 Len=0
150	35.200602585	193.136.9.33	10.0.2.15	TELNET	60 Telnet Data ...
151	35.200636635	10.0.2.15	193.136.9.33	TCP	54 51246 → 23 [ACK] Seq=66 Ack=512 Win=63784 Len=0
152	39.493407254	10.0.2.15	193.136.9.33	TELNET	55 Telnet Data ...
153	39.494349385	193.136.9.33	10.0.2.15	TCP	60 23 → 51246 [ACK] Seq=512 Ack=67 Win=65535 Len=0
154	39.701262282	10.0.2.15	193.136.9.33	TELNET	55 Telnet Data ...
Internet Protocol Version 4, Src: 193.136.9.33, Dst: 10.0.2.15					
Transmission Control Protocol, Src Port: 23, Dst Port: 51246, Seq: 498, Ack: 63, Len: 1					
Source Port: 23					
Destination Port: 51246					
[Stream index: 0]					
[TCP Segment Len: 1]					
Sequence number: 498 (relative sequence number)					
Sequence number (raw): 64499					
[Next sequence number: 499 (relative sequence number)]					
Acknowledgment number: 63 (relative ack number)					
Acknowledgment number (raw): 85365674					
6101 → Header Length: 20 bytes (5)					
Flags: 0x018 (PSH, ACK)					
Window size value: 65535					
[Calculated window size: 65535]					
[Window size scaling factor: -2 (no window scaling used)]					
Checksum: 0xc669 [unverified]					
[Checksum Status: Unverified]					
Urgent pointer: 0					
[SEQ/ACK analysis]					
[Timestamps]					
TCP payload (1 byte)					
Telnet					

Figura 15: Telnet

## 2.1.6 nslookup

259	187.094092849	10.0.2.15	193.136.9.33	TCP	54 51246 → 23 [ACK] Seq=70 Ack=553 Win=63784 Len=0
260	187.225483011	10.0.2.15	193.136.9.33	TELNET	55 Telnet Data ...
261	187.225992124	193.136.9.33	10.0.2.15	TCP	60 23 → 51246 [ACK] Seq=553 Ack=77 Win=65535 Len=0
262	187.245249575	193.136.9.33	10.0.2.15	TELNET	60 Telnet Data ...
263	187.245284975	10.0.2.15	193.136.9.33	TCP	54 51246 → 23 [ACK] Seq=77 Ack=554 Win=63784 Len=0
264	187.390025096	10.0.2.15	193.136.9.33	TELNET	56 Telnet Data ...
265	187.390531995	193.136.9.33	10.0.2.15	TCP	60 23 → 51246 [ACK] Seq=554 Ack=79 Win=65535 Len=0
266	187.411372326	193.136.9.33	10.0.2.15	TELNET	60 Telnet Data ...
267	187.411419926	10.0.2.15	193.136.9.33	TCP	54 51246 → 23 [ACK] Seq=79 Ack=556 Win=63784 Len=0
268	187.513784117	193.136.9.33	10.0.2.15	TCP	60 23 → 51246 [FIN, ACK] Seq=556 Ack=79 Win=65535 Len=0
269	187.514034231	10.0.2.15	193.136.9.33	TCP	54 51246 → 23 [FIN, ACK] Seq=79 Ack=557 Win=63784 Len=0
270	187.514477181	193.136.9.33	10.0.2.15	TCP	60 23 → 51246 [ACK] Seq=557 Ack=80 Win=65535 Len=0
271	191.554081132	10.0.2.15	192.168.1.254	DNS	84 Standard query 0xde2a A www.uminho.pt OPT
272	191.566519084	192.168.1.254	10.0.2.15	DNS	100 Standard query response 0xde2a A www.uminho.pt A 193.137.9.11...
273	191.572470494	10.0.2.15	192.168.1.254	DNS	84 Standard query 0x0e2e AAAA www.uminho.pt OPT
274	191.584086667	192.168.1.254	10.0.2.15	DNS	147 Standard query response 0x0e2e AAAA www.uminho.pt SOA dns.umi...
275	192.005022308	fe80::1521:1260:7f1...	ff02::fb	MDNS	107 Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR...
276	192.073836675	10.0.2.15	224.0.0.251	MDNS	87 Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR...
277	207.925433787	10.0.2.15	185.125.190.57	NTP	90 NTP Version 4, client
278	207.975736928	185.125.190.57	10.0.2.15	NTP	90 NTP Version 4, server
279	213.925472918	10.0.2.15	91.189.91.157	NTP	90 NTP Version 4, client
280	214.032270421	91.189.91.157	10.0.2.15	NTP	90 NTP Version 4, server
281	214.926447776	10.0.2.15	185.125.190.58	NTP	90 NTP Version 4, client
Frame 271: 84 bytes on wire (672 bits), 84 bytes captured (672 bits) on interface enp0s3, id 0					
Ethernet II, Src: PcsCompu_06:03:48 (08:00:27:06:03:48), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)					
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 192.168.1.254					
User Datagram Protocol, Src Port: 60914, Dst Port: 53					
Source Port: 60914					
Destination Port: 53					
Length: 50					
Checksum: 0xcef8 [unverified]					
[Checksum Status: Unverified]					
[Stream index: 19]					
[Timestamps]					
Domain Name System (query)					

Figura 16: nslookup

## 2.1.7 ping

393 535.168428317	192.168.1.254	10.0.2.15	DNS	112 Standard query response 0x48f2 AAAA www.google.pt AAAA 2a00:1...
394 535.168428617	192.168.1.254	10.0.2.15	DNS	100 Standard query response 0x3eaa A www.google.pt A 142.250.185....
395 535.168990031	10.0.2.15	142.250.185.3	ICMP	98 Echo (ping) request id=0x0001, seq=1/256, ttl=64 (reply in 3...
396 535.188553795	142.250.185.3	10.0.2.15	ICMP	98 Echo (ping) reply id=0x0001, seq=1/256, ttl=114 (request 1...
397 535.189041197	10.0.2.15	192.168.1.254	DNS	97 Standard query 0xab80 PTR 3.185.250.142.in-addr.arpa OPT
398 535.202916386	192.168.1.254	10.0.2.15	DNS	135 Standard query response 0xab89 PTR 3.185.250.142.in-addr.arpa...
399 536.170167659	10.0.2.15	142.250.185.3	ICMP	98 Echo (ping) request id=0x0001, seq=2/512, ttl=64 (reply in 4...
400 536.192394750	142.250.185.3	10.0.2.15	ICMP	98 Echo (ping) reply id=0x0001, seq=2/512, ttl=114 (request 1...
401 537.171431187	10.0.2.15	142.250.185.3	ICMP	98 Echo (ping) request id=0x0001, seq=3/768, ttl=64 (reply in 4...
402 537.189164795	142.250.185.3	10.0.2.15	ICMP	98 Echo (ping) reply id=0x0001, seq=3/768, ttl=114 (request 1...
403 538.173311151	10.0.2.15	142.250.185.3	ICMP	98 Echo (ping) request id=0x0001, seq=4/1024, ttl=64 (reply in ...
404 538.190492832	142.250.185.3	10.0.2.15	ICMP	98 Echo (ping) reply id=0x0001, seq=4/1024, ttl=114 (request ...
405 539.174684567	10.0.2.15	142.250.185.3	ICMP	98 Echo (ping) request id=0x0001, seq=5/1280, ttl=64 (reply in ...
406 539.192254770	142.250.185.3	10.0.2.15	ICMP	98 Echo (ping) reply id=0x0001, seq=5/1280, ttl=114 (request ...
407 540.174612597	10.0.2.15	142.250.185.3	ICMP	98 Echo (ping) request id=0x0001, seq=6/1536, ttl=64 (reply in ...
408 540.193646631	142.250.185.3	10.0.2.15	ICMP	98 Echo (ping) reply id=0x0001, seq=6/1536, ttl=114 (request ...
409 541.176499381	10.0.2.15	142.250.185.3	ICMP	98 Echo (ping) request id=0x0001, seq=7/1792, ttl=64 (reply in ...
410 541.193914088	142.250.185.3	10.0.2.15	ICMP	98 Echo (ping) reply id=0x0001, seq=7/1792, ttl=114 (request ...
411 542.178134774	10.0.2.15	142.250.185.3	ICMP	98 Echo (ping) request id=0x0001, seq=8/2048, ttl=64 (reply in ...
412 542.194935513	142.250.185.3	10.0.2.15	ICMP	98 Echo (ping) reply id=0x0001, seq=8/2048, ttl=114 (request ...
413 543.188106724	10.0.2.15	142.250.185.3	ICMP	98 Echo (ping) request id=0x0001, seq=9/2304, ttl=64 (reply in ...
414 543.198114372	142.250.185.3	10.0.2.15	ICMP	98 Echo (ping) reply id=0x0001, seq=9/2304, ttl=114 (request ...
Frame 399: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface enp0s3, id 0				
Ethernet II, Src: PcsCompu, 06:03:48 (08:00:27:06:03:48), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)				
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 142.250.185.3				
Internet Control Message Protocol				

Figura 17: ping

## 2.1.8 Traceroute

470 698.790507200	192.168.1.254	10.0.2.15	RIP	90 RIP version 4, server
471 698.792515486	10.0.2.15	192.168.1.254	DNS	89 Standard query 0x071f A cisco.di.uminho.pt OPT
472 698.793037451	10.0.2.15	192.168.1.254	DNS	89 Standard query 0x3410 AAAA cisco.di.uminho.pt OPT
473 698.818843770	192.168.1.254	10.0.2.15	DNS	138 Standard query response 0x3410 AAAA cisco.di.uminho.pt SOA dn...
474 698.820743765	192.168.1.254	10.0.2.15	DNS	105 Standard query response 0x071f A cisco.di.uminho.pt A 193.136...
475 698.820930305	10.0.2.15	193.136.19.254	UDP	74 42659 ~ 33442 Len=32
476 698.821152443	10.0.2.15	193.136.19.254	UDP	74 46333 ~ 33435 Len=32
477 698.821176010	10.0.2.15	193.136.19.254	UDP	74 57010 ~ 33436 Len=32
478 698.821290499	10.0.2.15	193.136.19.254	UDP	74 50059 ~ 33437 Len=32
479 698.821426407	10.0.2.2	10.0.2.15	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
480 698.821426678	10.0.2.2	10.0.2.15	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
481 698.821426778	10.0.2.2	10.0.2.15	ICMP	70 Time-to-live exceeded (Time to live exceeded in transit)
482 698.821521798	10.0.2.15	193.136.19.254	UDP	74 41531 ~ 33438 Len=32
483 698.821681408	10.0.2.15	193.136.19.254	UDP	74 39220 ~ 33439 Len=32
484 698.821719183	10.0.2.15	193.136.19.254	UDP	74 55872 ~ 33440 Len=32
485 698.821708147	10.0.2.15	193.136.19.254	UDP	74 47811 ~ 33441 Len=32
486 698.821850476	10.0.2.15	193.136.19.254	UDP	74 48698 ~ 33442 Len=32
487 698.822014595	10.0.2.15	193.136.19.254	UDP	74 56769 ~ 33443 Len=32
488 698.822050176	10.0.2.15	193.136.19.254	UDP	74 52679 ~ 33444 Len=32
489 698.822073563	10.0.2.15	193.136.19.254	UDP	74 53307 ~ 33445 Len=32
490 698.822093403	10.0.2.15	193.136.19.254	UDP	74 47638 ~ 33446 Len=32
491 698.822324525	10.0.2.15	193.136.19.254	UDP	74 59973 ~ 33447 Len=32
492 698.822358273	10.0.2.15	193.136.19.254	UDP	74 46487 ~ 33448 Len=32
493 698.822465508	10.0.2.15	193.136.19.254	UDP	74 56681 ~ 33449 Len=32
494 698.823055038	10.0.2.15	192.168.1.254	DNS	92 Standard query 0x0864 PTR 2.2.0.0.in-addr.arpa OPT
Frame 471: 89 bytes on wire (712 bits), 89 bytes captured (712 bits) on interface enp0s3, id 0				
Ethernet II, Src: PcsCompu, 06:03:48 (08:00:27:06:03:48), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)				
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 192.168.1.254				
User Datagram Protocol, Src Port: 36153, Dst Port: 53				
Source Port: 36153				
Destination Port: 53				
Length: 55				
Checksum: 0xc6fd [unverified]				
[Checksum Status: Unverified]				
[Stream index: 24]				
[Timestamps]				
Domain Name System (query)				

Figura 18: Traceroute

## 2.2 Conclusão

Neste trabalho, o grupo teve a oportunidade de aplicar e consolidar os conhecimentos adquiridos durante as aulas teóricas, especialmente em relação à camada de transporte e aos diferentes protocolos de transporte e aplicação. Os protocolos de transporte TCP e UDP foram particularmente destacados e estudados em detalhes. Além disso, utilizamos a ferramenta Wireshark para analisar o tráfego gerado ao empregar diversos protocolos.