

TP1: Protocolos da Camada de Transporte  
Comunicações por Computador  
Universidade do Minho

André Peixoto, Filipe Cunha, João Monteiro



# 1 Questões e Respostas

## 1.1 Pergunta 1

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	—	—	—	—
tracert	—	UDP	33434-33500	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
sftp	ssh	TCP	22	32

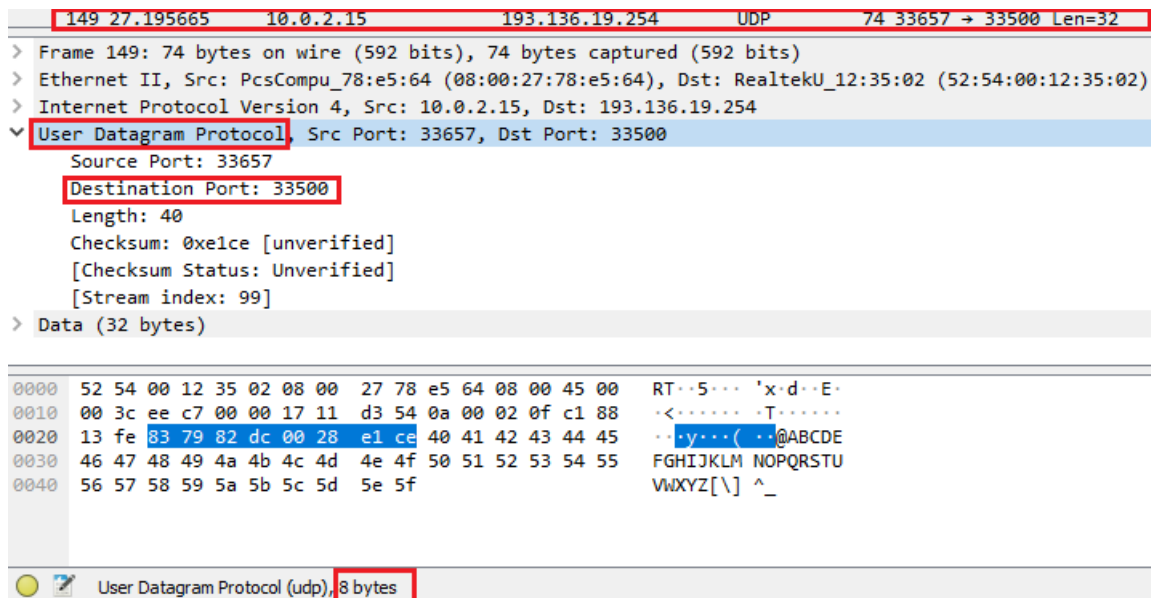


Figure 1: TraceRoute: UDP tem 8 bytes de overhead de transporte e porta de atendimento de 33500 (ultima porta usada)

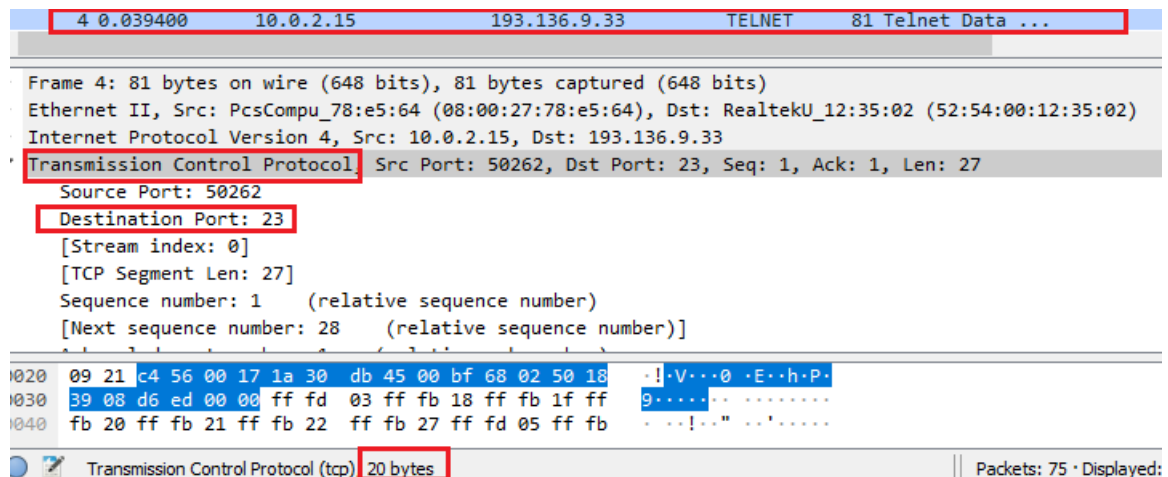


Figure 2: Telnet: TCP tem 20 bytes de overhead de transporte e porta de atendimento 23.

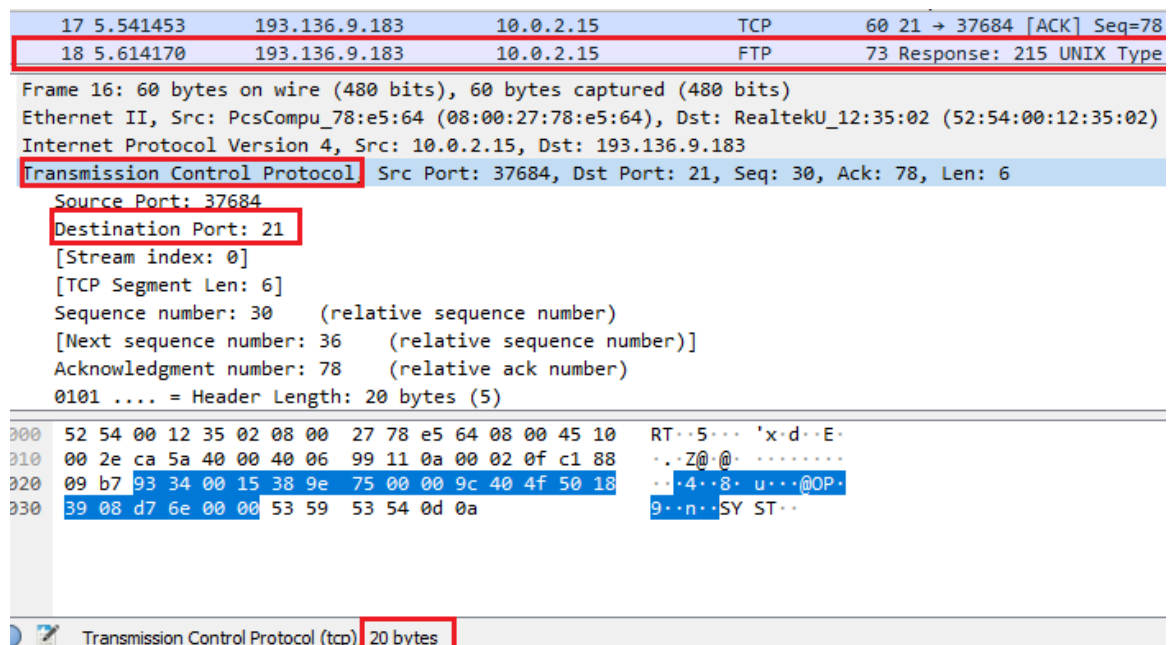


Figure 3: FTP: A porta de atendimento de TCP é 21 e overhead de transporte é 20 bytes.

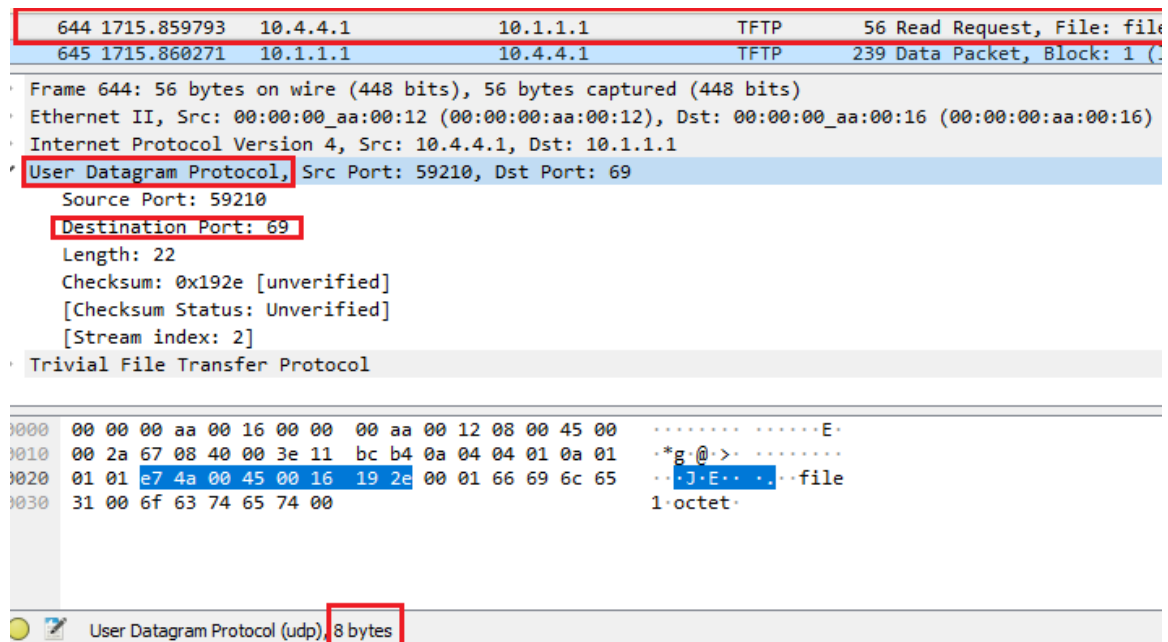


Figure 4: TFTP: UDP tem 8 bytes de overhead de transporte e porta de atendimento 69.

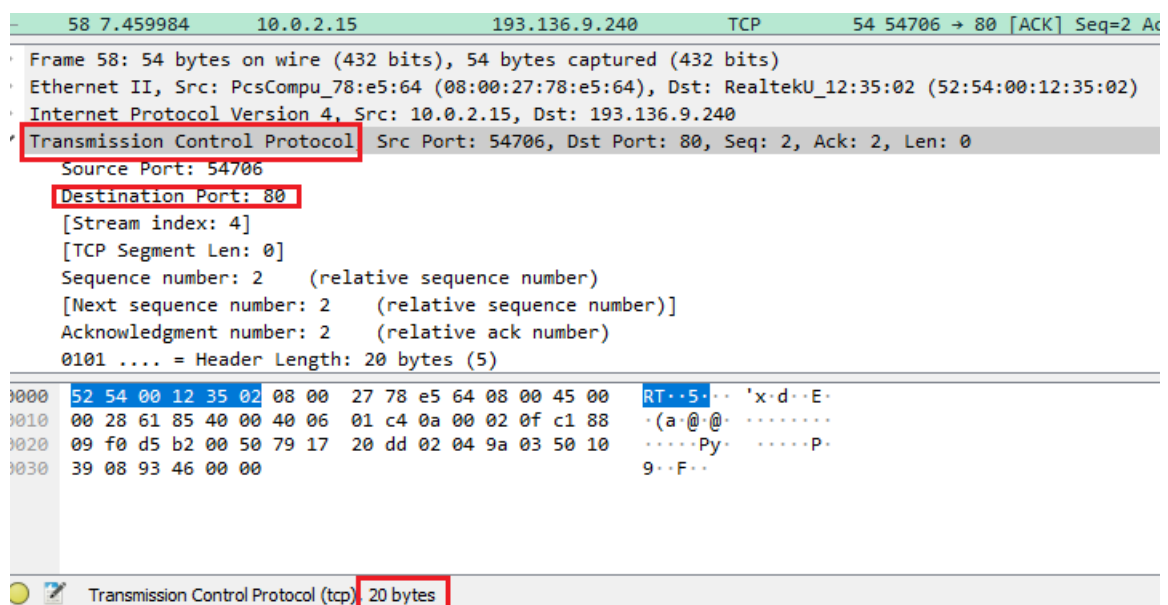


Figure 5: Browser/Http: TCP tem 20 bytes de overhead de transporte e porta de atendimento 80.

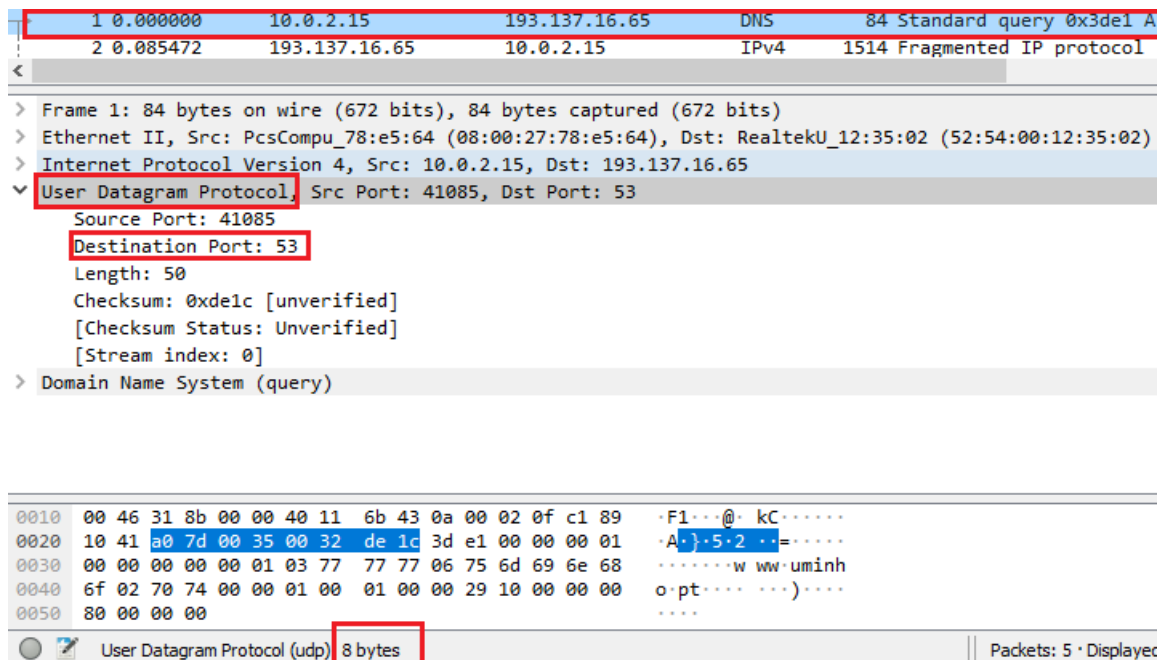


Figure 6: nslookup: UDP tem 8 bytes de overhead de transporte e porta de atendimento 53.

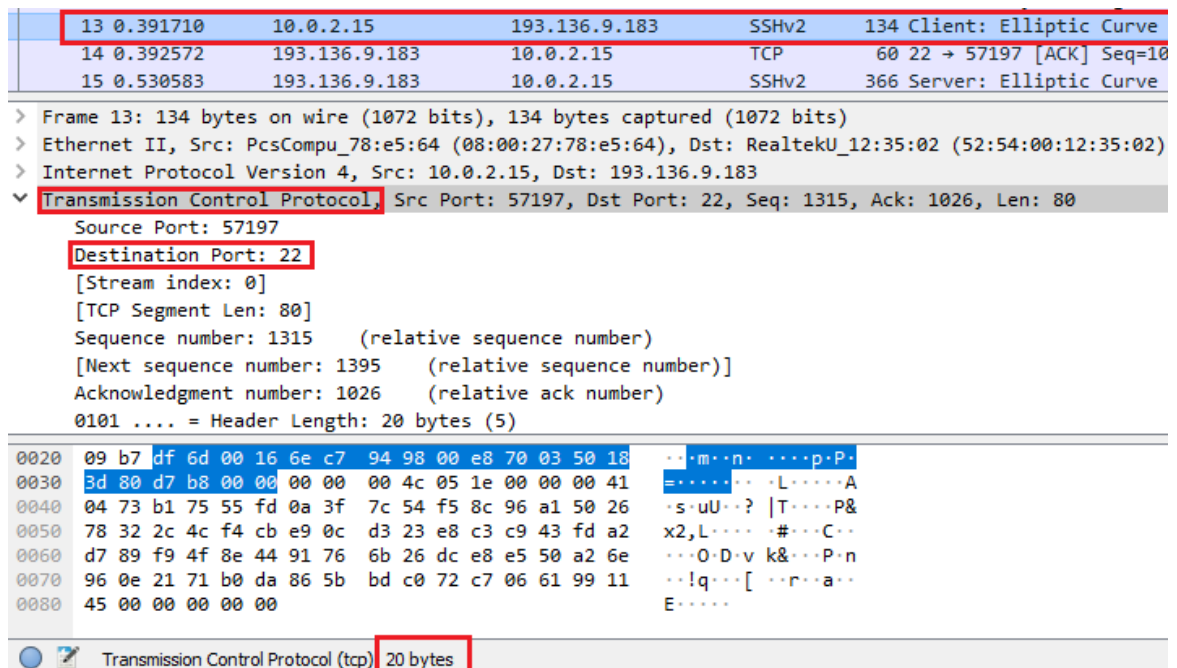


Figure 7: SSH: TCP tem 20 bytes de overhead de transporte e porta de atendimento 22.

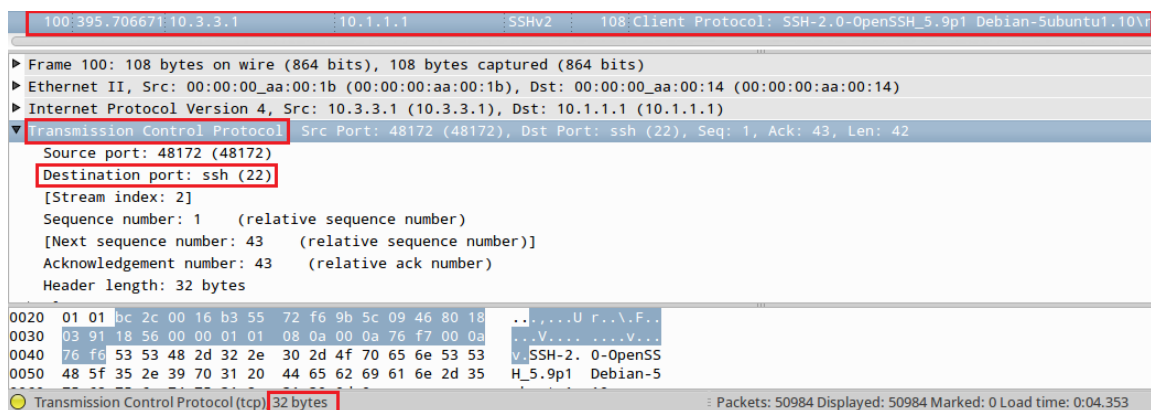


Figure 8: SFTP: TCP tem 32 bytes de overhead de transporte e porta de atendimento 22.

## 1.2 Pergunta 2

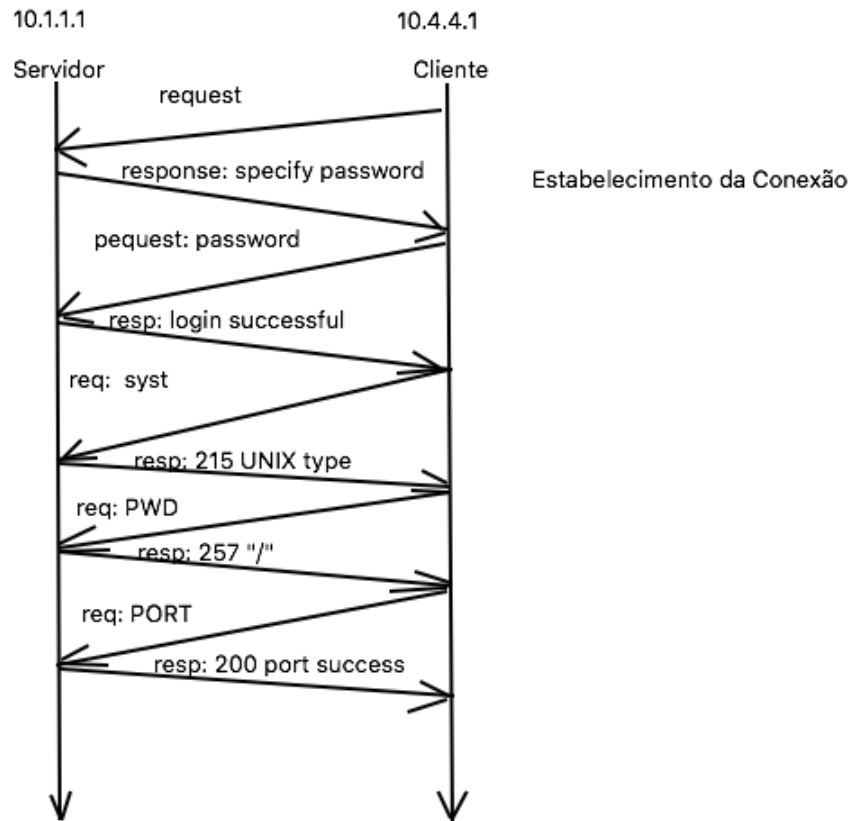


Figure 9: Diagrama Temporal FTP (1/3: estabelecimento de conexão).

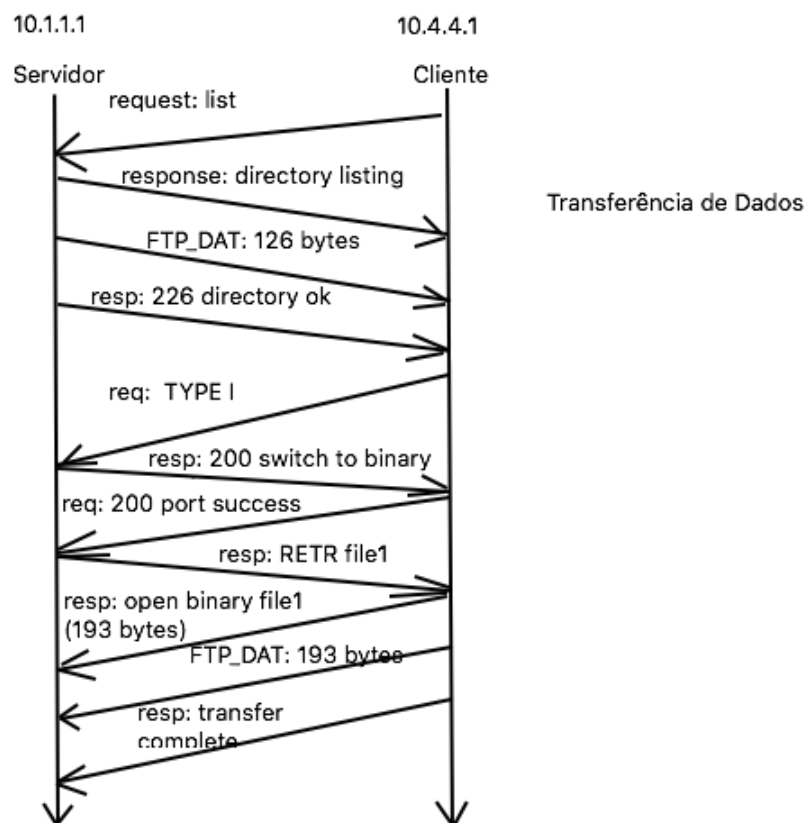


Figure 10: Diagrama Temporal FTP (2/3: transferência de dados).



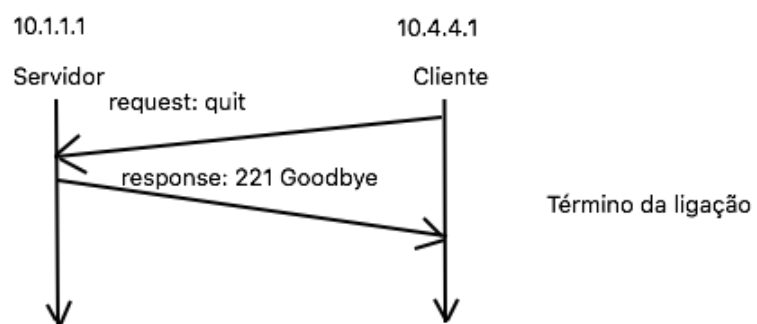


Figure 11: Diagrama Temporal FTP (3/3: fim de conexão).

541	1462.40412	10.4.4.1	10.1.1.1	FTP	82 Request: USER anonymous
542	1462.40426	10.1.1.1	10.4.4.1	TCP	66 ftp > 44040 [ACK] Seq=21 Ack=17 Win=14480 Len=0 TSval=1021570 TSecr=1021570
543	1462.40427	10.1.1.1	10.4.4.1	FTP	100 Response: 331 Please specify the password.
544	1462.40549	10.4.4.1	10.1.1.1	TCP	66 44040 > ftp [ACK] Seq=17 Ack=55 Win=14608 Len=0 TSval=1021570 TSecr=1021570
545	1469.95035	fe80::200:ff:feaa:1ff02::5	224.0.0.5	OSPF	90 Hello Packet
546	1470.10382	10.1.1.254	224.0.0.5	OSPF	78 Hello Packet
547	1479.77342	10.4.4.1	10.1.1.1	FTP	96 Request: PASS a82260@alunos.uminho.pt
548	1479.79205	10.1.1.1	10.4.4.1	FTP	89 Response: 230 Login successful.
549	1479.79341	10.4.4.1	10.1.1.1	TCP	66 44040 > ftp [ACK] Seq=47 Ack=78 Win=14608 Len=0 TSval=1025917 TSecr=1025917
550	1479.79342	10.4.4.1	10.1.1.1	FTP	72 Request: SYST
551	1479.79423	10.1.1.1	10.4.4.1	FTP	85 Response: 215 UNIX Type: L8
552	1479.83270	10.4.4.1	10.1.1.1	TCP	66 44040 > ftp [ACK] Seq=53 Ack=97 Win=14608 Len=0 TSval=1025928 TSecr=1025918
553	1480.00062	fe80::200:ff:feaa:1ff02::5	224.0.0.5	OSPF	90 Hello Packet
554	1480.10492	10.1.1.254	224.0.0.5	OSPF	78 Hello Packet
555	1490.03834	fe80::200:ff:feaa:1ff02::5	224.0.0.5	OSPF	90 Hello Packet
556	1490.10494	10.1.1.254	224.0.0.5	OSPF	78 Hello Packet
557	1490.64060	10.4.4.1	10.1.1.1	FTP	71 Request: PWD

Frame 541: R2_hvtac_on_wire (656 bits) R2_hvtac_captured (656 bits)	
0000	00 00 00 00 aa 00 16 00 00 00 aa 00 12 08 00 45 10 .....E.
0010	00 44 16 53 40 00 3e 06 0d 4b 0a 04 04 01 0a 01 .D.S@>. .K.....
0020	01 01 ac 08 00 15 89 1d 90 b6 54 03 14 45 80 18 .....T..E..
0030	03 91 19 3d 00 00 01 01 08 0a 00 0f 96 82 00 0f ...F.....
0040	86 e8 55 53 45 52 20 61 6e 6f 6e 79 6d 6f 75 73 ..USER a n o n y m o u s

File: "/home/core/Desktop/cc/tudo" 194 KB 00:34:40      Packets: 890 Displayed: 890 Marked: 0 Load time: 0:00.277      Profile

Figure 12: Pergunta 2: Transferência FTP (1/4).

557	1490.64060	10.4.4.1	10.1.1.1	FTP	71 Request: PWD
558	1490.64109	10.1.1.1	10.4.4.1	FTP	75 Response: 257 "/"
559	1490.64151	10.4.4.1	10.1.1.1	TCP	66 44040 > ftp [ACK] Seq=58 Ack=106 Win=14608 Len=0 TSval=1028630 TSecr=1028630
560	1494.44774	10.4.4.1	10.1.1.1	FTP	89 Request: PORT 10,4,4,1,215,133
561	1494.44886	10.1.1.1	10.4.4.1	FTP	117 Response: 200 PORT command successful. Consider using PASV.
562	1494.44973	10.4.4.1	10.1.1.1	TCP	66 44040 > ftp [ACK] Seq=81 Ack=157 Win=14608 Len=0 TSval=1029582 TSecr=1029581
563	1494.44973	10.4.4.1	10.1.1.1	FTP	72 Request: LIST
564	1494.45034	10.1.1.1	10.4.4.1	TCP	74 ftp-data > 55173 [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=1029582 TSecr=0 WS=16
565	1494.45100	10.4.4.1	10.1.1.1	TCP	74 55173 > ftp-data [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval=1029582 TSecr=1029582
566	1494.45137	10.1.1.1	10.4.4.1	TCP	66 ftp-data > 55173 [ACK] Seq=1 Ack=1 Win=14608 Len=0 TSval=1029582 TSecr=1029582
567	1494.45137	10.1.1.1	10.4.4.1	FTP	105 Response: 150 Here comes the directory listing.
568	1494.45142	10.1.1.1	10.4.4.1	FTP-DAT	192 FTP Data: 126 bytes
569	1494.45149	10.1.1.1	10.4.4.1	TCP	66 ftp-data > 55173 [FIN, ACK] Seq=127 Ack=1 Win=14608 Len=0 TSval=1029582 TSecr=1029582
570	1494.45249	10.4.4.1	10.1.1.1	TCP	66 55173 > ftp-data [ACK] Seq=1 Ack=127 Win=14480 Len=0 TSval=1029582 TSecr=1029582
571	1494.45250	10.4.4.1	10.1.1.1	TCP	66 55173 > ftp-data [FIN, ACK] Seq=1 Ack=128 Win=14480 Len=0 TSval=1029582 TSecr=1029582
572	1494.45327	10.1.1.1	10.4.4.1	TCP	66 ftp-data > 55173 [ACK] Seq=128 Ack=2 Win=14608 Len=0 TSval=1029582 TSecr=1029582
573	1494.45350	10.1.1.1	10.4.4.1	FTP	90 Response: 226 Directory send OK.
574	1494.45415	10.4.4.1	10.1.1.1	TCP	66 44040 > ftp [ACK] Seq=87 Ack=220 Win=14608 Len=0 TSval=1029583 TSecr=1029582

Frame 557: 71_hvtac_on_wire (568 bits) 71_hvtac_captured (568 bits)	
0000	00 00 00 aa 00 16 00 00 00 aa 00 12 08 00 45 10 .....E.
0010	00 39 16 59 40 00 3e 06 0d 50 0a 04 04 01 0a 01 .9.Y@>. .P.....
0020	01 01 ac 08 00 15 89 1d 90 ea 54 03 14 91 80 18 .....T..E..
0030	03 91 19 32 00 00 01 01 08 0a 00 0f b2 15 00 0f ...2.....
0040	a7 7e 50 57 44 0d 0a .....PMD..

File: "/home/core/Desktop/cc/tudo" 194 KB 00:34:40      Packets: 890 Displayed: 890 Marked: 0 Load time: 0:00.277      Profile: Default

Figure 13: Pergunta 2: Transferência FTP (2/4).

574	1494.45415	10.4.4.1	10.1.1.1	TCP	66.44040 > ftp [ACK] Seq=87 Ack=220 Win=14608 Len=0 TSval=1029583 TSecr=1029582
575	1497.58486	10.4.4.1	10.1.1.1	FTP	74 Request: TYPE I
576	1497.58607	10.1.1.1	10.4.4.1	FTP	97 Response: 200 Switching to Binary mode.
577	1497.58685	10.4.4.1	10.1.1.1	FTP	89 Request: PORT 10,4,4,1,144,251
578	1497.58707	10.1.1.1	10.4.4.1	FTP	117 Response: 200 PORT command successful. Consider using PASV.
579	1497.58753	10.4.4.1	10.1.1.1	FTP	78 Request: RETR file1
580	1497.58857	10.1.1.1	10.4.4.1	TCP	74 ftp-data > 37115 [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=1030366 TSecr=0 WS=16
581	1497.58965	10.4.4.1	10.1.1.1	TCP	74 37115 > ftp-data [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval=1030367 TSecr=1030367
582	1497.59026	10.1.1.1	10.4.4.1	TCP	66 ftp-data > 37115 [ACK] Seq=1 Ack=1 Win=14608 Len=0 TSval=1030367 TSecr=1030367
583	1497.59059	10.1.1.1	10.4.4.1	FTP	130 Response: 150 Opening BINARY mode data connection for file1 (193 bytes).
584	1497.59060	10.1.1.1	10.4.4.1	FTP-DAT	259 FTP Data: 193 bytes
585	1497.59073	10.1.1.1	10.4.4.1	TCP	66 ftp-data > 37115 [FIN, ACK] Seq=194 Ack=1 Win=14608 Len=0 TSval=1030367 TSecr=1030367
586	1497.59116	10.4.4.1	10.1.1.1	TCP	66 37115 > ftp-data [ACK] Seq=1 Ack=194 Win=15552 Len=0 TSval=1030367 TSecr=1030367
587	1497.59127	10.4.4.1	10.1.1.1	TCP	66 37115 > ftp-data [FIN, ACK] Seq=1 Ack=195 Win=15552 Len=0 TSval=1030367 TSecr=1030367
588	1497.59216	10.1.1.1	10.4.4.1	TCP	66 ftp-data > 37115 [ACK] Seq=195 Ack=2 Win=14608 Len=0 TSval=1030367 TSecr=1030367
589	1497.59257	10.1.1.1	10.4.4.1	FTP	90 Response: 226 Transfer complete.
590	1497.59356	10.4.4.1	10.1.1.1	TCP	66 44040 > ftp [ACK] Seq=130 Ack=390 Win=14608 Len=0 TSval=1030368 TSecr=1030367
591	1499.44856	00:00:00:00:00:00:00:00	00:00:00:00:00:00:00:00	ARP	42 Who has 10.1.1.1? Tell 10.1.1.254
592	1499.44876	00:00:00:00:00:00:00:00	00:00:00:00:00:00:00:00	ARP	42 10.1.1.1 is at 00:00:00:aa:00:16
593	1500.00387	fe80::200:ff:feaa:1ff02::5	224.0.0.5	OSPF	90 Hello Packet
594	1500.10517	10.1.1.254	224.0.0.5	OSPF	78 Hello Packet
595	1501.02882	10.4.4.1	10.1.1.1	FTP	72 Request: QUIT
596	1501.03064	10.1.1.1	10.4.4.1	FTP	80 Response: 221 Goodbye.
597	1501.03068	10.1.1.1	10.4.4.1	TCP	66 ftp > 44040 [FIN, ACK] Seq=404 Ack=136 Win=14480 Len=0 TSval=1031227 TSecr=1031226
598	1501.03149	10.4.4.1	10.1.1.1	TCP	66 44040 > ftp [FIN, ACK] Seq=136 Ack=405 Win=14608 Len=0 TSval=1031227 TSecr=1031227
599	1501.03214	10.1.1.1	10.4.4.1	TCP	66 ftp > 44040 [ACK] Seq=405 Ack=137 Win=14480 Len=0 TSval=1031227 TSecr=1031227
600	1509.98922	fe80::200:ff:feaa:1ff02::5	224.0.0.5	OSPF	90 Hello Packet
601	1510.10538	10.1.1.254	224.0.0.5	OSPF	78 Hello Packet
602	1520.00353	fe80::200:ff:feaa:1ff02::5	224.0.0.5	OSPF	90 Hello Packet
603	1520.10559	10.1.1.254	224.0.0.5	OSPF	78 Hello Packet
604	1529.98661	fe80::200:ff:feaa:1ff02::5	224.0.0.5	OSPF	90 Hello Packet

Frame 574: 66 bytes on wire (528 bits) - 66 bytes captured (528 bits) on interface 0

```

0000 00 00 00 aa 00 16 00 00 00 aa 00 12 08 00 45 10 .....E.
0010 00 34 16 5e 40 00 3e 06 0d 50 0a 04 01 0a 01 .4.®>.P.....
0020 01 01 ac 08 00 15 89 1d 91 0c 54 03 15 0c 80 10 .....T.....7..
0030 03 91 19 2d 00 00 01 01 08 0a 00 0f b5 cf 00 0f ...E.....
0040 b5 ce

```

File: "/home/core/Desktop/cc/tudo" 194 KB 00:34:40 : Packets: 890 Displayed: 890 Marked: 0 Load time: 0:00:277 : Profile: Default

Figure 14: Pergunta 2: Transferência FTP (3/4).

589	1497.59257	10.1.1.1	10.4.4.1	FTP	90 Response: 226 Transfer complete.
590	1497.59356	10.4.4.1	10.1.1.1	TCP	66 44040 > ftp [ACK] Seq=130 Ack=390 Win=14608 Len=0 TSval=1030368 TSecr=1030367
591	1499.44856	00:00:00:00:00:00:00:00	00:00:00:00:00:00:00:00	ARP	42 Who has 10.1.1.1? Tell 10.1.1.254
592	1499.44876	00:00:00:00:00:00:00:00	00:00:00:00:00:00:00:00	ARP	42 10.1.1.1 is at 00:00:00:aa:00:16
593	1500.00387	fe80::200:ff:feaa:1ff02::5	224.0.0.5	OSPF	90 Hello Packet
594	1500.10517	10.1.1.254	224.0.0.5	OSPF	78 Hello Packet
595	1501.02882	10.4.4.1	10.1.1.1	FTP	72 Request: QUIT
596	1501.03064	10.1.1.1	10.4.4.1	FTP	80 Response: 221 Goodbye.
597	1501.03068	10.1.1.1	10.4.4.1	TCP	66 ftp > 44040 [FIN, ACK] Seq=404 Ack=136 Win=14480 Len=0 TSval=1031227 TSecr=1031226
598	1501.03149	10.4.4.1	10.1.1.1	TCP	66 44040 > ftp [FIN, ACK] Seq=136 Ack=405 Win=14608 Len=0 TSval=1031227 TSecr=1031227
599	1501.03214	10.1.1.1	10.4.4.1	TCP	66 ftp > 44040 [ACK] Seq=405 Ack=137 Win=14480 Len=0 TSval=1031227 TSecr=1031227
600	1509.98922	fe80::200:ff:feaa:1ff02::5	224.0.0.5	OSPF	90 Hello Packet
601	1510.10538	10.1.1.254	224.0.0.5	OSPF	78 Hello Packet
602	1520.00353	fe80::200:ff:feaa:1ff02::5	224.0.0.5	OSPF	90 Hello Packet
603	1520.10559	10.1.1.254	224.0.0.5	OSPF	78 Hello Packet
604	1529.98661	fe80::200:ff:feaa:1ff02::5	224.0.0.5	OSPF	90 Hello Packet

Frame 589: 90 bytes on wire (720 bits) - 90 bytes captured (720 bits) on interface 0

```

0000 00 00 00 aa 00 12 00 00 00 aa 00 16 08 00 45 00 .....E.
0010 00 4c 92 e3 40 00 04 06 8e c2 0a 01 01 01 0a 04 .L.®.®.....
0020 04 01 00 15 ac 08 54 03 15 9e 89 1d 91 37 80 18 .....T.....7..
0030 03 89 19 45 00 00 01 01 08 0a 00 0f b8 df 00 0f ...E.....
0040 b8 de 32 32 36 20 54 72 61 6e 73 66 65 72 20 63 ..226 Transfer c

```

File: "/home/core/Desktop/cc/tudo" 194 KB 00:34:40 : Packets: 890 Displayed: 890 Marked: 0 Load time: 0:00:277 : Profile: Default

Figure 15: Pergunta 2: Transferência FTP (4/4).

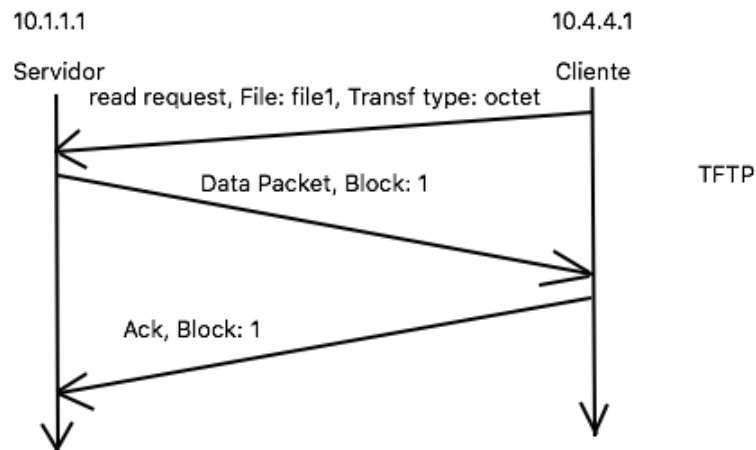


Figure 16: Diagrama Temporal TFTP (1/1)

644	1715.85979	10.4.4.1	10.1.1.1	TFTP	56 Read Request, File: file1, Transfer type: octet
645	1715.86027	10.1.1.1	10.4.4.1	TFTP	239 Data Packet, Block: 1 (last)
646	1715.86058	10.4.4.1	10.1.1.1	TFTP	46 Acknowledgement, Block: 1
647	1720.08300	fe80::200:ff:feaa:1	ff02::5	OSPF	90 Hello Packet
648	1720.11213	10.1.1.254	224.0.0.5	OSPF	78 Hello Packet
649	1720.87268	00:00:00_aa:00:16	00:00:00_aa:00:12	ARP	42 Who has 10.1.1.254? Tell 10.1.1.1
650	1720.87270	00:00:00_aa:00:12	00:00:00_aa:00:16	ARP	42 10.1.1.254 is at 00:00:00_aa:00:12
651	1730.04533	fe80::200:ff:feaa:1	ff02::5	OSPF	90 Hello Packet
652	1730.11225	10.1.1.254	224.0.0.5	OSPF	78 Hello Packet
653	1740.04966	fe80::200:ff:feaa:1	ff02::5	OSPF	90 Hello Packet
654	1740.11247	10.1.1.254	224.0.0.5	OSPF	78 Hello Packet

▶ Frame 644: 56 bytes on wire (448 bits) 56 bytes captured (448 bits)			
0000	00 00 00 aa 00 16 00 00	00 aa 00 12 08 00 45 00	.....E.
0010	00 2a 67 08 40 00 3e 11	bc b4 0a 04 04 01 0a 01	.*g@.>.....
0020	01 01 e7 4a 00 45 00 16	19 2e 00 01 66 69 6c 65	...J.E...file
0030	31 00 6f 63 74 65 74 00		1.octet.

File: "/home/core/Desktop/cc/tudo" 194 KB 00:34:40    Packets: 890 Displayed: 890 Marked: 0 Load time: 0:00.277

Figure 17: Pergunta 2: Transferência TFTP (1/1).

### 1.3 Pergunta 3

Com base nas experiências realizadas, podemos distinguir as 4 aplicações de transferência que usamos: SFTP, FTP, TFTP e HTTP.

**(i) uso da camada de transporte:** SFTP, FTP e HTTP usam TCP, enquanto que o TFTP usa UDP.

**(ii) eficiência na transferência:** o FTP é mais rápido do que SFTP, que por sua vez é mais rápido do que HTTP (o http seria mais rápido para uma transferência de múltiplos ficheiros). O TFTP é mais rápido do que o FTP por usar UDP que é mais rápido do que TCP porque não usa acknowledgement para controlo de fluxo (como demonstrado no diagrama TFTP da pergunta 2).

```
root@Cliente1:/tmp/pycore.47517/Cliente1.conf# rm /root/.ssh/known_hosts
root@Cliente1:/tmp/pycore.47517/Cliente1.conf# sftp core@10.1.1.1
The authenticity of host '10.1.1.1 (10.1.1.1)' can't be established.
RSA key fingerprint is 22:fc:a5:06:9f:44:74:ed:fe:a5:5c:b4:4f:ad:3b:b7.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.1.1.1' (RSA) to the list of known hosts.
core@10.1.1.1's password:
Connected to 10.1.1.1.
sftp> pwd
Remote working directory: /home/core
sftp> cd /srv/ftp
sftp> dir
file1  file2
sftp> get file2
Fetching /srv/ftp/file2 to file2
/srv/ftp/file2                                100% 102KB 17.0KB/s 00:06
sftp> █
```

Figure 18: Velocidade do SFTP para file2 (6 segundos).

```

Connected to 10.1.1.1.
220 (vsFTPd 2.3.5)
Name (10.1.1.1:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> status
Connected to 10.1.1.1.
No proxy connection.
Connecting using address family: any.
Mode: stream; Type: binary; Form: non-print; Structure: file
Verbose: on; Bell: off; Prompting: on; Globbing: on
Store unique: off; Receive unique: off
Case: off; CR stripping: on
Quote control characters: on
Ntrans: off
Nmap: off
Hash mark printing: off; Use of PORT cmds: on
Tick counter printing: off
ftp> pwd
257 "/"
ftp> dir
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rwxrwxrwx   1 0      0          193 Feb 18 11:51 file1
-rwxrwxrwx   1 0      0       104508 Feb 18 11:51 file2
226 Directory send OK.
ftp> get file2
local: file2 remote: file2
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for file2 (104508 bytes).
226 Transfer complete.
104508 bytes received in 0.63 secs (161.7 kB/s)
ftp> quit
221 Goodbye.
root@Cliente1:/tmp/pycore.47517/Cliente1.conf# █

```

Figure 19: Velocidade do FTP para file2 (0.63 segundos).

```

root@Client1:/tmp/pycore.47517/Client1.conf# wget http://10.1.1.1/file2
--2019-02-25 16:59:37-- http://10.1.1.1/file2
Connecting to 10.1.1.1:80... connected.
HTTP request sent, awaiting response... 200 Ok
Length: 104508 (102K) [text/plain]
Saving to: 'file2.1'

100%[=====] 104,508 26.3K/s in 3.9s

2019-02-25 16:59:40 (26.3 KB/s) - 'file2.1' saved [104508/104508]

root@Client1:/tmp/pycore.47517/Client1.conf# wget http://10.1.1.1/file1
--2019-02-25 16:59:43-- http://10.1.1.1/file1
Connecting to 10.1.1.1:80... connected.
HTTP request sent, awaiting response... 200 Ok
Length: 193 [text/plain]
Saving to: 'file1.1'

100%[=====] 193 --.-K/s in 0s

2019-02-25 16:59:43 (13.6 MB/s) - 'file1.1' saved [193/193]

root@Client1:/tmp/pycore.47517/Client1.conf# █

```

Figure 20: Velocidade do HTTP para file2 (3.9 segundos).

Outro aspeto importante da eficiência é a confiabilidade.

O protocolo TFTP é "unreliable" devido ao facto do UDP ser connectionless, isto é, caso um pacote seja perdido na rede, o protocolo de transporte UDP não apresenta qualquer mecanismo para este ser reenviado. Por outro lado, todos os outros protocolos que usam TCP como protocolo de transporte estabelecem uma conexão mais "reliable" pois este apresenta controlo de fluxo.

**(iii) complexidade:** como já referido anteriormente, o TFTP utiliza o protocolo de transporte UDP e por isso torna esta aplicação a mais simples das 4. Prova disto é também o facto de possuir o menor overhead (8 bytes). Contrariamente, o FTP, HTTP e SFTP implementam controlo de fluxo, controlo de congestão e controlo de erro (TCP), o que, obviamente, implica uma maior quantidade de funcionalidades o que, por sua vez, aumenta o nível de complexidade.

**(iv) segurança:** Efetivamente, estas quatro aplicações de transferência de ficheiros apresentam diferentes níveis de segurança. Em primeiro lugar, temos o TFTP que não apresenta qualquer tipo de sistema de autenticação o que o torna mais vulnerável em relação a todas as outras aplicações impossibilitando o controlo no acesso aos ficheiros. Pelo contrário FTP e HTTP implementam mecanismos de autenticação o que já permite criar sistemas de permissões entre os ficheiros, no entanto, estes protocolos implementam sistemas de autenticação básicos, os quais são bastante vulneráveis a ataques na atualidade. Por fim, temos o SSH e SFTP que implementam mecanismos de autenticação encriptada. Esta implementação de segurança permite uma menor vulnerabilidade a ataques exteriores tornando este tipo de aplicações nos sistemas mais seguros.

## 1.4 Pergunta 4

De facto, quando se aumenta a percentagem de loss package da ligação, verificamos que, obviamente, vários pacotes são perdidos. Nesta situação, como estamos a tratar de aplicações fiáveis, como FTP e HTTP (que utilizam protocolo de transporte TCP), estas são obrigadas a implementar mecanismos que garantam que todos os pacotes cheguem ao destino.

Assim, como vemos nas figuras em baixo, quando se dá a perda de pacotes ou a recessão de pacotes com erros, o servidor irá enviar ao cliente acknowledgements duplicados que referem o pacote em falta até que o cliente o reenvie com sucesso. O cliente, quando recebe vários acknowledgements a referir o mesmo pacote desconfia que o pacote foi mal recebido e reenvia-o. Este processo dará origem à retransmissão do pacote em falta, corrigindo o erro e garantindo a fiabilidade de aplicações que usem o protocolo de transporte TCP.

Por fim, a validação da recessão de todos os pacotes, apesar de garantir a fiabilidade, traduz-se também numa perda de desempenho devido ao aumento do overhead causado pelo protocolo de transporte TCP, como vimos na pergunta 1. Para minimizar este impacto negativo, uma das abordagens que devemos tomar é a seguinte: na ocorrência de um erro na transmissão de um determinado pacote para o destino, este, apesar de enviar acknowledgements para a origem referindo o pacote em erro, continua a receber os pacotes seguintes guardando-os num buffer. Posto isto, quando o pacote em erro é reenviado corretamente, o destino deve ter mecanismos que lhe permita aceitar os outros pacotes que foram recebidos com sucesso (pelo buffer), evitando o reenvio destes por parte do cliente.

Desta forma, conseguimos diminuir o impacto de erros numa conexão TCP.



120	151.0.1003	10.1.1.1	10.3.3.1	FTP-DAT	1514	FTP Data: 1448 bytes
121	151.071081	10.3.3.1	10.1.1.1	TCP	78	[TCP Dup ACK 118#1] 57405 > ftp-data [ACK] Seq=1 Ack=4345 Win=23168 Len=0 TSval=220453 TSecr=220453
122	151.071101	10.3.3.1	10.1.1.1	TCP	78	[TCP Dup ACK 118#2] 57405 > ftp-data [ACK] Seq=1 Ack=4345 Win=23168 Len=0 TSval=220453 TSecr=220453
123	151.077150	10.1.1.1	10.3.3.1	FTP-DAT	1514	[TCP Fast Retransmission] FTP Data: 1448 bytes
124	151.077210	10.1.1.1	10.3.3.1	FTP-DAT	1514	[TCP Retransmission] FTP Data: 1448 bytes
125	151.077362	10.1.1.1	10.3.3.1	FTP-DAT	1514	[TCP Retransmission] FTP Data: 1448 bytes
126	151.077442	10.3.3.1	10.1.1.1	TCP	78	57405 > ftp-data [ACK] Seq=1 Ack=5793 Win=26064 Len=0 TSval=220455 TSecr=220455 SLE=20273 SRE=2316
127	151.077503	10.3.3.1	10.1.1.1	TCP	78	57405 > ftp-data [ACK] Seq=1 Ack=7241 Win=28960 Len=0 TSval=220455 TSecr=220455 SLE=20273 SRE=2316
128	151.077579	10.3.3.1	10.1.1.1	TCP	78	57405 > ftp-data [ACK] Seq=1 Ack=8689 Win=31856 Len=0 TSval=220455 TSecr=220455 SLE=20273 SRE=2316
129	151.083302	10.1.1.1	10.3.3.1	FTP-DAT	1514	[TCP Retransmission] FTP Data: 1448 bytes
130	151.083361	10.1.1.1	10.3.3.1	FTP-DAT	1514	[TCP Retransmission] FTP Data: 1448 bytes
131	151.083485	10.3.3.1	10.1.1.1	TCP	78	57405 > ftp-data [ACK] Seq=1 Ack=10137 Win=34752 Len=0 TSval=220457 TSecr=220456 SLE=20273 SRE=231
132	151.083581	10.3.3.1	10.1.1.1	TCP	78	57405 > ftp-data [ACK] Seq=1 Ack=11585 Win=37648 Len=0 TSval=220457 TSecr=220456 SLE=20273 SRE=231
133	151.090154	10.1.1.1	10.3.3.1	FTP-DAT	1514	[TCP Retransmission] FTP Data: 1448 bytes
134	151.090314	10.1.1.1	10.3.3.1	FTP-DAT	1514	[TCP Retransmission] FTP Data: 1448 bytes
135	151.090486	10.1.1.1	10.3.3.1	FTP-DAT	1514	[TCP Retransmission] FTP Data: 1448 bytes
136	151.090592	10.3.3.1	10.1.1.1	TCP	78	57405 > ftp-data [ACK] Seq=1 Ack=13033 Win=40544 Len=0 TSval=220458 TSecr=220458 SLE=20273 SRE=231

Figure 21: FTP com 80 por cento loss e 50 por cento duplicates.

101	130.571142	10.1.1.1	10.3.3.1	TCP	1514	[TCP segment of a reassembled PDU]
102	130.571210	10.3.3.1	10.1.1.1	TCP	66	59164 > http [ACK] Seq=114 Ack=33531 Win=42256 Len=0 TSval=312905 TSecr=312905
103	130.577809	10.1.1.1	10.3.3.1	TCP	1514	[TCP Previous segment lost] [TCP segment of a reassembled PDU]
104	130.577868	10.1.1.1	10.3.3.1	TCP	1514	[TCP segment of a reassembled PDU]
105	130.577919	10.3.3.1	10.1.1.1	TCP	78	[TCP Dup ACK 102#1] 59164 > http [ACK] Seq=114 Ack=33531 Win=42256 Len=0 TSval=312906 TSecr=312906
106	130.577942	10.3.3.1	10.1.1.1	TCP	78	[TCP Dup ACK 102#2] 59164 > http [ACK] Seq=114 Ack=33531 Win=42256 Len=0 TSval=312906 TSecr=312906
107	130.583673	10.1.1.1	10.3.3.1	TCP	1514	[TCP Fast Retransmission] [TCP segment of a reassembled PDU]
108	130.583727	10.1.1.1	10.3.3.1	TCP	1514	[TCP Retransmission] [TCP segment of a reassembled PDU]
109	130.583779	10.3.3.1	10.1.1.1	TCP	78	59164 > http [ACK] Seq=114 Ack=34979 Win=40816 Len=0 TSval=312908 TSecr=312908 SLE=40771 SRE=436
110	130.583801	10.3.3.1	10.1.1.1	TCP	78	59164 > http [ACK] Seq=114 Ack=36427 Win=39392 Len=0 TSval=312908 TSecr=312908 SLE=40771 SRE=436
111	130.583980	10.1.1.1	10.3.3.1	TCP	1514	[TCP Retransmission] [TCP segment of a reassembled PDU]
112	130.584028	10.3.3.1	10.1.1.1	TCP	78	59164 > http [ACK] Seq=114 Ack=37875 Win=38320 Len=0 TSval=312908 TSecr=312908 SLE=40771 SRE=436
113	130.589599	10.1.1.1	10.3.3.1	TCP	1514	[TCP Retransmission] [TCP segment of a reassembled PDU]
114	130.589650	10.3.3.1	10.1.1.1	TCP	78	59164 > http [ACK] Seq=114 Ack=39323 Win=40464 Len=0 TSval=312909 TSecr=312909 SLE=40771 SRE=436
115	130.589782	10.1.1.1	10.3.3.1	TCP	1514	[TCP Retransmission] [TCP segment of a reassembled PDU]
116	130.589818	10.3.3.1	10.1.1.1	TCP	66	59164 > http [ACK] Seq=114 Ack=43667 Win=39392 Len=0 TSval=312909 TSecr=312909
117	130.589967	10.1.1.1	10.3.3.1	TCP	1514	[TCP segment of a reassembled PDU]

Figure 22: HTTP com 80 por cento loss e 50 por cento duplicates.

## **2 Conclusões**

Neste trabalho explorámos os vários tipos de aplicações e aspetos específicos que fazem uso de protocolos de transporte, analisando as suas vantagens e desvantagens bem como o impacto que apresentam a níveis de segurança, fiabilidade, complexidade e eficiência.

Com a ajuda do wireshark fomos capazes de classificar os protocolos de transporte (UDP e TCP) de cada aplicação verificando o impacto de cada um destes protocolos (overhead).

De seguida, testamos todas as aplicações referidas no enunciado com a ajuda do core através de uma topologia fornecida pelos docentes. Analisamos o seu comportamento e verificamos as diferenças entre as várias aplicações. Posto isto, fomos capazes de perceber as implicações que os protocolos de transporte têm sobre as aplicações que os usam.

Por fim, analisamos uma situação de package loss em algumas aplicações fiáveis (usam TCP) de forma a perceber como os mecanismos de correção de erros funcionam. Por outro lado vimos também formas de diminuir o impacto na performance aquando de erros de transmissão.

Este trabalho permitiu-nos analisar a forma como as aplicações comunicam entre si e observar os mecanismos por de traz dessa comunicação em funcionamento. Desta forma, foi possível obter uma compreensão mais aprofundada nas diversas vertentes sobre os protocolos da camada de transporte enriquecendo os conhecimentos obtidos nas aulas teóricas.