



Universidade do Minho
Escola de Engenharia

Mestrado Integrado em Engenharia Informática

Comunicações por Computador

Relatório do Trabalho Prático 1

A78890 Alexandre Costa
A75248 Ana Sofia Gomes Marques
A65277 Flávio Manuel Machado Martins
A79799 Gonçalo Costeira

Grupo 8

3 Março 2020

Conteúdo

1	Questões e Respostas	i
1.1	Questão 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:	i
1.1.1	Ping	ii
1.1.2	Traceroute	iii
1.1.3	Telnet	iv
1.1.4	ftp	iv
1.1.5	Tftp	v
1.1.6	browser/http	vi
1.1.7	nslookup	vii
1.1.8	ssh	viii
1.2	Questão 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.	ix
1.2.1	FTP	ix
1.2.2	TFTP	xi
1.3	Questão 3 - Com base nas experiências realizadas, distinga e compare sucintamente as quatro aplicações de transferência de ficheiros que usou nos seguintes pontos	xii
1.3.1	FTP - File Transfer Protocol	xii
1.3.2	SFTP -Secure File Transfer Protocol	xii
1.3.3	TFTP - Trivial File Transfer Protocol	xiii
1.3.4	HTTP - Hypertext Control Protocol	xiii
1.4	Questão 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)	xiv
2	Conclusão	xv

1 Questões e Respostas

- 1.1 Questão 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	-	-	-	-
tracert	-	UDP	33434 a 33534	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

Por forma a justificar os valores de overhead acima referidos para os diferentes comandos, sabemos que o segmento TCP possui 20 bytes de overhead de cabeçalho em cada segmento, enquanto o UDP possui apenas 8 bytes de overhead.

1.1.1 Ping

No caso do ping, este não possui protocolo de aplicação, protocolo de transporte e nem porta de atendimento. Por consequência, não terá nenhum overhead de transporte, uma vez que este opera ao mandar apenas mensagens ICMP para o destino à espera de obter um ICMP echo reply.

Este comportamento poderá ser observado no seguinte printscreen onde foi efetuado um ping para o endereço IP 193.137.16.65.

No.	Time	Source	Destination	Protocol	Length	Info
230	585.772692	10.0.2.15	193.137.16.65	ICMP	98	Echo (ping) request id=0x1be5, seq=1/256, ttl=64
231	585.778684	193.137.16.65	10.0.2.15	ICMP	98	Echo (ping) reply id=0x1be5, seq=1/256, ttl=62
232	586.775253	10.0.2.15	193.137.16.65	ICMP	98	Echo (ping) request id=0x1be5, seq=2/512, ttl=64
233	586.777944	193.137.16.65	10.0.2.15	ICMP	98	Echo (ping) reply id=0x1be5, seq=2/512, ttl=62
234	587.778137	10.0.2.15	193.137.16.65	ICMP	98	Echo (ping) request id=0x1be5, seq=3/768, ttl=64
235	587.780132	193.137.16.65	10.0.2.15	ICMP	98	Echo (ping) reply id=0x1be5, seq=3/768, ttl=62
236	588.779711	10.0.2.15	193.137.16.65	ICMP	98	Echo (ping) request id=0x1be5, seq=4/1024, ttl=64
237	588.781906	193.137.16.65	10.0.2.15	ICMP	98	Echo (ping) reply id=0x1be5, seq=4/1024, ttl=62
238	589.781942	10.0.2.15	193.137.16.65	ICMP	98	Echo (ping) request id=0x1be5, seq=5/1280, ttl=64
239	589.784127	193.137.16.65	10.0.2.15	ICMP	98	Echo (ping) reply id=0x1be5, seq=5/1280, ttl=62
▶ Frame 230: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)						
▶ Ethernet II, Src: CadmusCo_78:e5:64 (08:00:27:78:e5:64), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)						
▶ Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 193.137.16.65 (193.137.16.65)						
▶ Internet Control Message Protocol						
0000	52 54 00 12 35 02 08 00	27 78 e5 64 08 00 45 00	RT..5... 'x.d..E.			
0010	00 54 5d 07 40 00 40 01	ff c8 0a 00 02 0f c1 89	.T].@.@.			
0020	10 41 08 00 0a 77 1b e5	00 01 46 b8 4b 5e 4e 89	.A...w... ..F.K^N.			
0030	06 00 08 09 0a 0b 0c 0d	0e 0f 10 11 12 13 14 15			
0040	16 17 18 19 1a 1b 1c 1d	1e 1f 20 21 22 23 24 25 !"#\$\$%			
0050	26 27 28 29 2a 2b 2c 2d	2e 2f 30 31 32 33 34 35	&'()*+,- ./012345			
0060	36 37		67			

Figura 1: Tramas capturadas usando o comando ping.

1.1.2 Traceroute

No traceroute não existe protocolo de aplicação, mas possui protocolo de transporte UDP (User Datagram Protocol).

As portas de atendimento variam entre o 33434 e o 33534, como se pode verificar na próxima figura:

No.	Time	Source	Destination	Protocol	Length	Info
7	0.014362	10.0.2.15	193.136.19.254	UDP	74	Source port: 56555 Destination port: traceroute
8	0.014487	10.0.2.2	10.0.2.15	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
9	0.014557	10.0.2.15	193.136.19.254	UDP	74	Source port: 43375 Destination port: 33435
10	0.014675	10.0.2.2	10.0.2.15	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
11	0.014741	10.0.2.15	193.136.19.254	UDP	74	Source port: 43006 Destination port: 33436
12	0.014858	10.0.2.2	10.0.2.15	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
13	0.014911	10.0.2.15	193.136.19.254	UDP	74	Source port: 43629 Destination port: 33437
▶ Frame 7: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)						
▶ Ethernet II, Src: CadmusCo_78:e5:64 (08:00:27:78:e5:64), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)						
▶ Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 193.136.19.254 (193.136.19.254)						
▼ User Datagram Protocol Src Port: 56555 (56555), Dst Port: traceroute (33434)						
Source port: 56555 (56555)						
Destination port: traceroute (33434)						
Length: 40						
▶ Checksum: 0xe1ce [validation disabled]						
▶ Data (32 bytes)						
0000 52 54 00 12 35 02 08 00 27 78 e5 64 08 00 45 00 RT..5... 'x.d..E.						
0010 00 3c b1 d3 00 00 01 11 26 49 0a 00 02 0f c1 88 .<..... &I.....						
0020 13 fe dc eb 82 9a 00 28 e1 ce 40 41 42 43 44 45 ..[.....]@ABCDE						
0030 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 FGHIJKLM NOPQRSTU						
0040 56 57 58 59 5a 5b 5c 5d 5e 5f VWXYZ[\] ^_						

Figura 2: Tramas capturadas usando o comando traceroute.

1.1.3 Telnet

No comando telnet é utilizado o protocolo de aplicação telnet, este é utilizado para providenciar uma comunicação, utilizando uma conexão terminal virtual.

O protocolo de transporte é o TCP (Transmission Control Protocol) e possui uma porta de atendimento 23, como se pode verificar pela figura abaixo apresentada.

No.	Time	Source	Destination	Protocol	Length	Info
10	0.023940	10.0.2.15	193.136.9.183	TELNET	81	Telnet Data ...
11	0.024440	193.136.9.183	10.0.2.15	TCP	60	telnet > 49219 [ACK] Seq=1 Ack=28 Win=65535 Len=0
12	0.024668	10.0.2.15	193.136.9.183	TELNET	66	Telnet Data ...
▶ Frame 10: 81 bytes on wire (648 bits), 81 bytes captured (648 bits)						
▶ Ethernet II, Src: CadmusCo_78:e5:64 (08:00:27:78:e5:64), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)						
▶ Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 193.136.9.183 (193.136.9.183)						
▼ Transmission Control Protocol Src Port: 49219 (49219), Dst Port: telnet (23), Seq: 1, Ack: 1, Len: 27						
Source port: 49219						
Destination port: telnet (23)						
[Stream index: 3]						
Sequence number: 1 (relative sequence number)						
[Next sequence number: 28 (relative sequence number)]						
Acknowledgement number: 1 (relative ack number)						
Header length: 20 bytes						
▶ Flags: 0x018 (PSH, ACK)						
Window size value: 14600						
[Calculated window size: 14600]						
[Window size scaling factor: -2 (no window scaling used)]						
▶ Checksum: 0xd783 [validation disabled]						
▶ [SEQ/ACK analysis]						
0020	09 b7	c0 43 00 17 aa 78	22 f0 08 1f 1a 02 50 18	..C...x "...P.		
0030	39 08	d7 83 00 00 ff fd	03 ff fb 18 ff fb 1f ff	9.....		
0040	f0 20	ff fb 21 ff fb 22	ff fb 27 ff fd 05 ff fb	...!..." ..'....		
0050	23			#		

Figura 3: Tramas capturadas usando o comando telnet.

1.1.4 ftp

O ftp utiliza o protocolo, com o mesmo nome, sendo este usado para a transferência de dados. O protocolo de aplicação é o ftp sendo que o de transporte é o TCP (Transmission Control Protocol), a porta de atendimento é a 21, como se pode verificar na seguinte figura:

No.	Time	Source	Destination	Protocol	Length	Info
592	39.140273	10.0.2.15	193.136.9.183	FTP	63	Request: USER cc
593	39.140779	193.136.9.183	10.0.2.15	TCP	60	ftp > 42825 [ACK] Seq=21 Ack=10 Win=65535 Len=0
594	39.146481	193.136.9.183	10.0.2.15	FTP	88	Response: 331 Please specify the password.
595	39.146592	10.0.2.15	193.136.9.183	TCP	54	42825 > ftp [ACK] Seq=10 Ack=55 Win=14600 Len=0
▶ Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 193.136.9.183 (193.136.9.183)						
▼ Transmission Control Protocol Src Port: 42825 (42825), Dst Port: ftp (21), Seq: 1, Ack: 21, Len: 9						
Source port: 42825						
Destination port: ftp (21)						
[Stream index: 34]						
Sequence number: 1 (relative sequence number)						
[Next sequence number: 10 (relative sequence number)]						
Acknowledgement number: 21 (relative ack number)						
Header length: 20 bytes						
▶ Flags: 0x018 (PSH, ACK)						
Window size value: 14600						
[Calculated window size: 14600]						
[Window size scaling factor: -2 (no window scaling used)]						
▶ Checksum: 0xd771 [validation disabled]						
▶ [SEQ/ACK analysis]						
▶ File Transfer Protocol (FTP)						
0010	00 31	a6 d5 40 00 40 06	bc 93 0a 00 02 0f c1 88	.1..@.		
0020	09 b7	a7 49 00 15 a8 36	be 6f 02 dc 6c 16 50 18	..I...6..o..l.P.		
0030	39 08	d7 71 00 00 55 53	45 52 20 63 6d 0d 0a	9..q..US ER cc..		

Figura 4: Tramas capturadas usando o comando ftp.

1.1.5 Tftp

Neste caso foi utilizado um comando curl que funciona como um protocolo de transferência de dados simples, semelhante ao ftp.

Deste modp, o protocolo de aplicação é o tftp, o protocolo de transporte é o UDP (User Datagram Protocol) com a porta de atendimento 69, assim como se pode verificar no printscreen abaixo apresentado.

No.	Time	Source	Destination	Protocol	Length	Info
7	0.012425	10.0.2.15	193.136.9.183	TFTP	86	Read Request, File: file1, Tran
8	6.015459	10.0.2.15	193.136.9.183	TFTP	86	Read Request, File: file1, Tran
9	12.015569	10.0.2.15	193.136.9.183	TFTP	86	Read Request, File: file1, Tran
▶ Frame 7: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)						
▶ Ethernet II, Src: CadmusCo_78:e5:64 (08:00:27:78:e5:64), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)						
▶ Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 193.136.9.183 (193.136.9.183)						
▼ User Datagram Protocol, Src Port: 51734 (51734), Dst Port: tftp (69)						
Source port: 51734 (51734)						
Destination port: tftp (69)						
Length: 52						
▶ Checksum: 0xd793 [validation disabled]						
▶ Trivial File Transfer Protocol						
0000 52 54 00 12 35 02 08 00 27 78 e5 64 08 00 45 00 RT..5... 'x.d..E.						
0010 00 48 36 2b 40 00 40 11 2d 2c 0a 00 02 0f c1 88 .H6+@.@. -,.....						
0020 09 b7 ca 16 00 45 00 34 d7 93 00 01 66 69 6c 65 ..E.4...file						
0030 31 00 6f 63 74 65 74 00 74 73 69 7a 65 00 30 00 1.octet. tsize.0.						
0040 62 6c 6b 73 69 7a 65 00 35 31 32 00 74 69 6d 65 blksize. 512.time						
0050 6f 75 74 00 36 00 out.6.						

Figura 5: Tramas capturadas usando o comando Tftp.

1.1.6 browser/http

De modo a capturar o tráfego, foi utilizado o comando:

```
wget http://marco.uminho.pt/disciplinas/CC-MIEI/
```

Onde foi possível depois da análise da trama capturada que o protocolo de aplicação é o http e o de transporte o TCP(Transmission Control Protocol) e a porta de atendimento é a 80. Tudo isto pode ser verificado na figura seguinte:

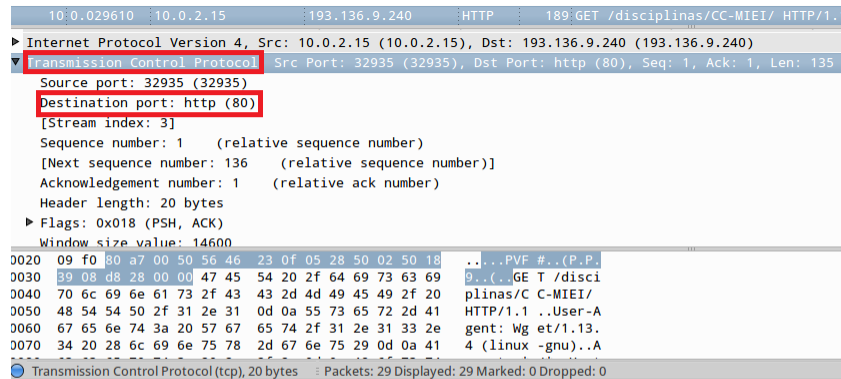


Figura 6: Tramas capturadas usando o comando http.

1.1.7 nslookup

O nslookup funciona como uma ferramenta de resolução de nomes como tal, o protocolo de aplicação é o DNS (Domain Name System), o de transporte é o UDP (User Datagram Protocol) com porta de atendimento 53, como se pode confirmar pela imagem abaixo apresentada.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.2.15	193.137.16.65	DNS	73	Standard query A www.uminho.pt
2	0.005813	193.137.16.65	10.0.2.15	DNS	345	Standard query response A 193.137.9.114
3	5.012632	CadmusCo_78:e5:64	RealtekU_12:35:02	ARP	42	Who has 10.0.2.2? Tell 10.0.2.15
4	5.012783	RealtekU_12:35:02	CadmusCo_78:e5:64	ARP	60	10.0.2.2 is at 52:54:00:12:35:02

▶ Frame 1: 73 bytes on wire (584 bits), 73 bytes captured (584 bits)
▶ Ethernet II, Src: CadmusCo_78:e5:64 (08:00:27:78:e5:64), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
▶ Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 193.137.16.65 (193.137.16.65)
▼ User Datagram Protocol, Src Port: 52014 (52014), Dst Port: domain (53)
Source port: 52014 (52014)
Destination port: domain (53)
Length: 39
▶ Checksum: 0xde11 [validation disabled]
▶ Domain Name System (query)
0000 52 54 00 12 35 02 08 00 27 78 e5 64 08 00 45 00 RT..5... 'x.d..E.
0010 00 3b 9e 9c 00 00 40 11 fe 3c 0a 00 02 0f c1 89 ..@. .<.....
0020 10 41 cb 2e 00 35 00 27 de 11 83 c2 01 00 00 01 .A...5.
0030 00 00 00 00 00 00 03 77 77 77 06 75 6d 69 6e 68w ww.uminh
0040 6f 02 70 74 00 00 01 00 01 o.pt.... .

Figura 7: Tramas capturadas usando o comando nslookup.

1.1.8 ssh

O ssh é um protocolo de rede criptográfico cujo intuito é operar em redes de forma segura através de uma rede insegura, tendo sido efetuado um login remoto. O protocolo de aplicação portanto é ssh, o de transporte o TCP e a porta de atendimento a 22.

No.	Time	Source	Destination	Protocol	Length	Info
9	0.0615467	193.136.9.183	10.0.2.15	TCP	95	37555 > ssh [ACK] Seq=1 Ack=1 Win=14600 Len=0
10	0.061587	193.136.9.183	10.0.2.15	SSHv2	95	Server Protocol: SSH-2.0-OpenSSH_5.9p1 Debian-5ubuntu1.4\r
11	0.061649	10.0.2.15	193.136.9.183	TCP	54	37555 > ssh [ACK] Seq=1 Ack=42 Win=14600 Len=0
12	0.061864	10.0.2.15	193.136.9.183	SSHv2	95	Client Protocol: SSH-2.0-OpenSSH_5.9p1 Debian-5ubuntu1.4\r
13	0.062199	193.136.9.183	10.0.2.15	TCP	60	ssh > 37555 [ACK] Seq=42 Ack=42 Win=65535 Len=0
▶ Frame 10: 95 bytes on wire (760 bits), 95 bytes captured (760 bits)						
▶ Ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: CadmusCo_78:e5:64 (08:00:27:78:e5:64)						
▶ Internet Protocol Version 4, Src: 193.136.9.183 (193.136.9.183), Dst: 10.0.2.15 (10.0.2.15)						
▼ Transmission Control Protocol, Src Port: ssh (22), Dst Port: 37555 (37555), Seq: 1, Ack: 1, Len: 41						
Source port: ssh (22)						
Destination port: 37555 (37555)						
[Stream index: 3]						
Sequence number: 1 (relative sequence number)						
[Next sequence number: 42 (relative sequence number)]						
Acknowledgement number: 1 (relative ack number)						
Header length: 20 bytes						
▶ Flags: 0x018 (PSH, ACK)						
Window size value: 65535						
0000	08 00 27 78 e5 64 52 54	00 12 35 02 08 00 45 00	.. 'x.dRT ..S...E.			
0010	00 51 05 21 00 00 40 06	9e 38 c1 88 09 b7 0a 00	.Q.l...@. .8.....			
0020	02 0f 00 16 92 b3 0a 01	86 02 6f 39 45 95 50 1809E.P.			
0030	ff ff 0e 9a 00 00 53 53	48 2d 32 2e 30 2d 4f 70SSH-2.0-Op			
0040	65 6e 53 53 48 5f 35 2e	39 70 31 20 44 65 62 69	enSSH_5. 9p1 Debi			
0050	61 6e 2d 35 75 62 75 6e	74 75 31 2e 34 0d 0a	an-Subun tu1.4..			

Figura 8: Tramas capturadas usando o comando ssh.

1.2 Questão 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.

1.2.1 FTP

146 422.095457 10.1.1.1	10.3.3.1	FTP	78 Request: RETR file1
147 422.095586 10.3.3.1	10.1.1.1	TCP	74 ftp-data > 59234 [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=2512272 TSecr=0 WS=16
148 422.095756 10.1.1.1	10.3.3.1	TCP	74 59234 > ftp-data [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval=2512272 TSecr=2
149 422.095904 10.3.3.1	10.1.1.1	TCP	66 ftp-data > 59234 [ACK] Seq=1 Ack=1 Win=14608 Len=0 TSval=2512272 TSecr=2512272
150 422.095965 10.3.3.1	10.1.1.1	FTP	130 Response: 150 Opening BINARY mode data connection for file1 (193 bytes).
151 422.095969 10.3.3.1	10.1.1.1	FTP-DAT	259 FTP Data: 193 bytes
152 422.096024 10.3.3.1	10.1.1.1	TCP	66 ftp-data > 59234 [FIN, ACK] Seq=194 Ack=1 Win=14608 Len=0 TSval=2512272 TSecr=2512272
153 422.096264 10.1.1.1	10.3.3.1	TCP	66 59234 > ftp-data [ACK] Seq=1 Ack=194 Win=15552 Len=0 TSval=2512272 TSecr=2512272
154 422.097146 10.1.1.1	10.3.3.1	TCP	66 59234 > ftp-data [FIN, ACK] Seq=1 Ack=195 Win=15552 Len=0 TSval=2512272 TSecr=2512272
155 422.097533 10.3.3.1	10.1.1.1	TCP	66 ftp-data > 59234 [ACK] Seq=195 Ack=2 Win=14608 Len=0 TSval=2512273 TSecr=2512272
156 422.097688 10.3.3.1	10.1.1.1	FTP	90 Response: 226 Transfer complete.

Figura 9: Tramas da transferência do file1 por FTP.

O protocolo de transferência de arquivos(FTP) utiliza como protocolo de transporte o TCP. Este pode ser dividido em três fases distintas:

- Estabelecimento de conexão;
- Transferência de dados;
- Fim da conexão.

Neste caso, da transferência do file1 via FTP, podemos ver que na fase de estabelecimento de conexão o Cliente envia um Request do ficheiro ao Servidor, e o Servidor entao envia um SYN(Synchronize) ao Cliente, sincronizando e iniciando assim uma conexão entre ambos. Durante esta conexão o Servidor envia exactamente um ficheiro para o cliente e no final termina a conexão. Se durante a sessão o cliente iniciar outra transferência é criada uma nova conexão.

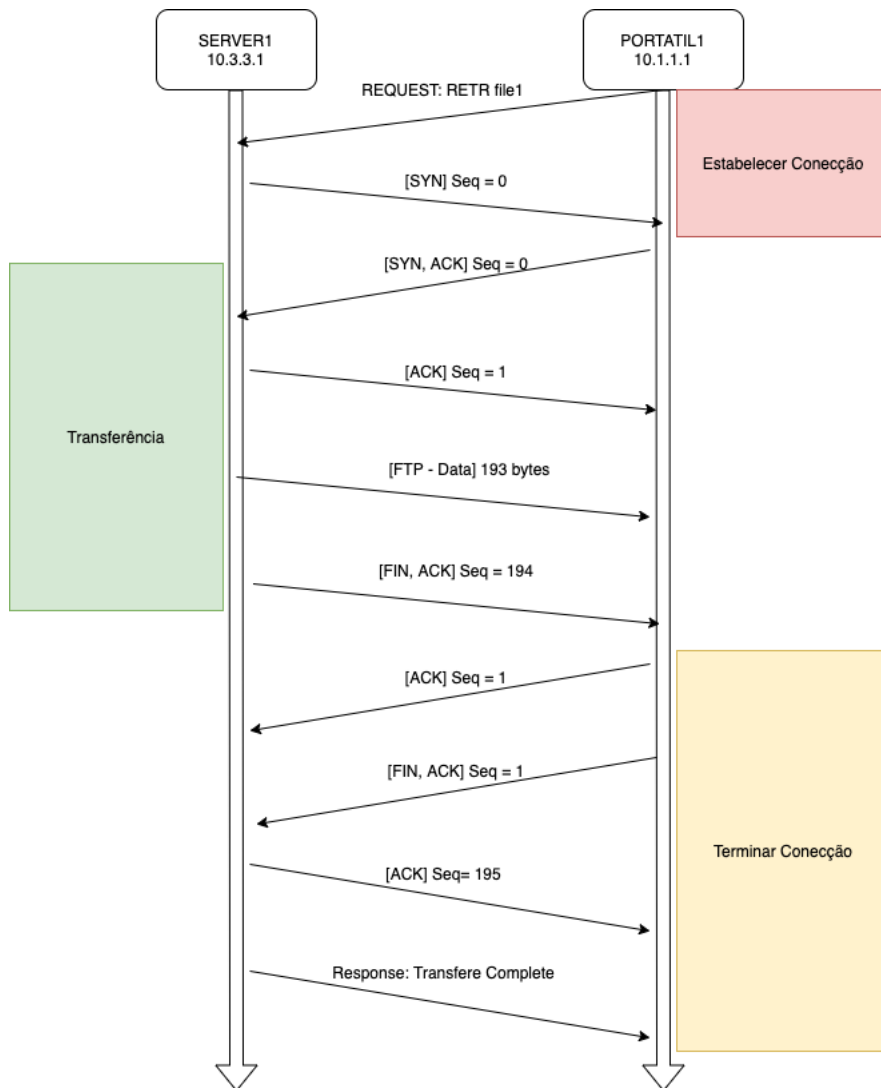


Figura 10: Diagrama Temporal da transferência por FTP

1.2.2 TFTP

43	194.306922	10.1.1.1	10.3.3.1	TFTP	56 Read Request, File: file1, Transfer type: octet
44	194.307381	10.3.3.1	10.1.1.1	TFTP	239 Data Packet, Block: 1 (last)
45	194.307875	10.1.1.1	10.3.3.1	TFTP	46 Acknowledgement, Block: 1

Figura 11: Tramas da transferencia do file1 por TFTP.

Na transferencia por TFTP é utilizado como protocolo de transporte o UDP. Como pode ser observado na figura 11, é inicialmente feito um Read Request pelo cliente ao servidor, de seguida o pacote é transferido do servidor para o cliente, e por fim o cliente envia um acknowledgement ao servidor.

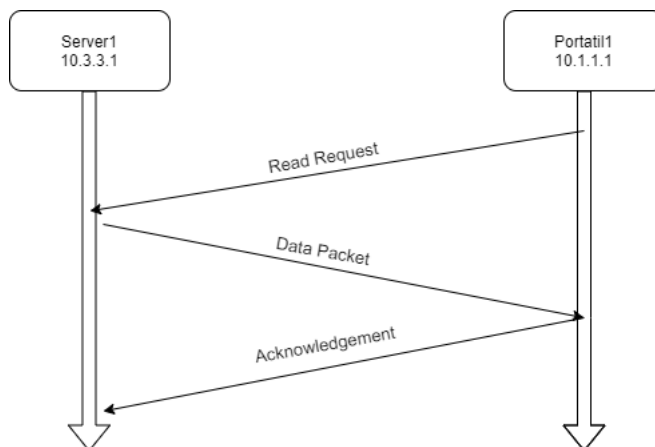


Figura 12: Diagrama temporal da transferência por TFTP.

1.3 Questão 3 - Com base nas experiências realizadas, distinga e compare sucintamente as quatro aplicações de transferência de ficheiros que usou nos seguintes pontos

1.3.1 FTP - File Transfer Protocol

(i) **Uso da camada de transporte:** O servidor remoto aceita uma conexão de controlo do cliente a nível local. O cliente envia comandos para o servidor e a conexão persiste ao longo de toda a sessão (tratando-se assim de um protocolo TCP);

(ii) **Eficiência na transferência:** Não muito eficiente uma vez que envia toda a informação numa só Stream de bits guardada como uma string limitando a capacidade de controlo de erros e até mesmo velocidade de transmissão.

(iii) **Complexidade:** Envia uma Stream de bits guardada como uma única string que contém a informação e toda o overhead necessário para transmissão (ex : nome do ficheiro, tamanho, timestamp).

(iv) **Segurança:** FTP é considerada uma das opções menos seguras no que toca a transmissão de ficheiros em rede uma vez que não encripta os dados e para a transferência destes não considera a segurança desta, havendo diversos tipos de formas de intercetar transferências em FTP como por exemplo : FTP Bounce Attack , FTP Brute Force Attack , Packet Capture (or Sniffing), Spoof Attack ou Port Stealing.

1.3.2 SFTP -Secure File Transfer Protocol

(i) **Uso da camada de transporte:** A transferência dos dados é realizada através de uma conexão previamente assegurada, que utiliza o protocolo SSH (Secure Shell protocol).

(ii) **Eficiência na transferência:** SFTP é um protocolo baseado em transferência de pacotes ao invés de baseados em texto. Assim ao enviar pequenos pacotes com a informação devidamente dividida e preparada entre si consegue ser um método mais rápido e é capaz de produzir um menor número de erros e de perda de informação.

(iii) **complexidade;** Transferências em SFTP são realizadas através de uma conexão SSH e ocorre a preparação da informação em data packages. Além disso, esta transferência é realizada sobre o controlo de conexão principal , eliminando a necessidade de abrir uma nova conexão de dados somente para este evento.

(iv) **segurança:** Uma vez que utiliza o protocolo SSH é considerada intrinsecamente segura.

1.3.3 TFTP - Trivial File Transfer Protocol

(i) **Uso da camada de transporte:** Utiliza o protocolo de transporte UDP.

(ii) **Eficiência na transferência:** Ao utilizar os protocolo UDP o TFTP peca por não possuir tanto controle de erros, ha- vendo assim alguma perda na sua eficiência. Para além disso tem também de suportar a sessão criada para o envio da informação.

(iii) **Complexidade:** Transferência é iniciada pelo cliente pedindo para ler ou escrever um ficheiro num ser- vidor. Se este aceita o pedido o ficheiro e enviado em blocos de tamanho fixo , cada um geralmente possuindo apenas um único pacote IP de modo a evitar a sua fragmentação e deve esperar pelo acknowledgment de modo a poder enviar o pacote seguinte.

(iv) **Segurança:** O TFTP não possui nenhum mecanismo de controlo de acesso. Assim devemos ter cui- dado ao enviar ficheiros de carácter privado e ter em atenção aos direitos garantidos ao servidor TFTP de modo a não violar a segurança do servidor onde o ficheiro se encontra guardado.

1.3.4 HTTP - Hypertext Control Protocol

(i) **Uso da camada de transporte:** Aplicação de camada de transporte desenhada com a framework do protocolo de Internet em mente, assumindo assim capacidade de transporte segura e eficaz.

(ii) **Eficiência na transferência:** - Eficiente no sentido em que a sua arquitetura é projetada para permitir intermediação entre elementos de rede de modo a melhorar e permitir comunicações entre servidores e clientes. Web sites com muita procura e tráfego normalmente apresentam servers cache que permi- tem melhorar o tempo de resposta.

(iii) **Complexidade:** Devido a utilização de servers cache e outros métodos que permitem me- lhorar a eficiência da transmissão, verificamos que existe um aumento a nível de recursos utilizados e também na complexidade da sua gestão que permite este melhoramento.

(iv) **segurança:** O HTTP utiliza vários métodos de autentificação para a ocorrência de transfe- rências, quer seja acessos básicos ou acessos a web sites com informação maus importante os métodos de autentificação e segurança vão melhorando, fornecendo assim um ambiente seguro para o envio de dados.

1.4 Questão 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)

Em transmissão de dados é normal ouvir falar em package loss, este ocorre quando um ou mais pacotes que navegam sobre uma determinada rede falham em alcançar o destinatário, é assim lógico que ao aumentar a percentagem de pacotes a serem enviados na ligação, se verifique que mais pacotes são perdidos ou duplicados.

Sabemos que aplicações de transferência fiáveis, sendo um exemplo a SFTP (SSH File Transfer Protocol), tipicamente utilizado com o protocolo de segurança SSH, que utiliza protocolo de transporte TCP (Transmission Control Protocol), este permite a recuperação de pacotes perdidos e a eliminação de pacotes duplicados, utilizam a retransmissão para garantir que todos os dados chegam ao destino e que estes são confiáveis.

A perda de pacotes em uma conexão TCP também é usada para evitar congestionamentos e, portanto, produz uma taxa de transferência intencionalmente reduzida para a conexão.

A perda de pacotes está intimamente associada a considerações de qualidade de serviço. A quantidade de perda de pacotes aceitável depende do tipo de dados que está sendo enviado, mas ronda em torno de valores bastante baixos.

Assim, ao aumentar o package loss, acabamos por ter um maior overhead, causado pelo facto de que quanto maior for a perda, mais tráfego vai circular na rede por forma a ser capaz de corrigir os erros existentes, o que acaba por diminuir o desempenho global da rede.

30	2.631936	10.3.3.1	10.2.2.1	SSH	1514 [TCP Previous segment lost] Encrypted response packet len=1448
31	2.631936	10.3.3.1	10.2.2.1	SSH	1122 Encrypted response packet len=1056
32	2.637727	10.2.2.1	10.3.3.1	TCP	78 [TCP Dup ACK 29#1] 46705 > ssh [ACK] Seq=273 Ack=14705 Win=2641 Len=0 TSval=431
33	2.637729	10.2.2.1	10.3.3.1	TCP	78 [TCP Dup ACK 29#2] 46705 > ssh [ACK] Seq=273 Ack=14705 Win=2641 Len=0 TSval=431
34	2.637940	10.3.3.1	10.2.2.1	SSH	1514 [TCP Fast Retransmission] Encrypted response packet len=1448
35	2.637941	10.3.3.1	10.2.2.1	SSH	1514 [TCP Retransmission] Encrypted response packet len=1448
36	2.637999	10.3.3.1	10.2.2.1	SSH	1514 [TCP Retransmission] Encrypted response packet len=1448
37	2.643685	10.2.2.1	10.3.3.1	TCP	78 46705 > ssh [ACK] Seq=273 Ack=16153 Win=2551 Len=0 TSval=4316234 TSecr=4316234
▶ Internet Protocol Version 4, Src: 10.3.3.1 (10.3.3.1), Dst: 10.2.2.1 (10.2.2.1)					
▼ Transmission Control Protocol, Src Port: ssh (22), Dst Port: 46705 (46705), Seq: 16153, Ack: 273, Len: 1448					
Source port: ssh (22)					
Destination port: 46705 (46705)					
[Stream index: 0]					
Sequence number: 16153 (relative sequence number)					
[Next sequence number: 17601 (relative sequence number)]					
Acknowledgement number: 273 (relative ack number)					
Header length: 32 bytes					
► Flags: 0x010 (ACK)					
Window size value: 1563					
[Calculated window size: 1563]					
0020	02 01	00 16 b6 71 1f 68	c1 80 6b 36 b3 00 80 10q.h..k6...
0030	06 1b 1e d5 00 00 01 01	08 0a 00 41 dc 4a 00 41A.J.A
0040	dc 48 93 5f 0f 11 2f a2	38 64 75 d1 0d 9a 69 c4/.8du...i.
0050	28 b9 2a b2 77 cd f9 83	09 55 d4 92 58 c8 d3 95	(.w...U.X...

Figura 13: SFTP com perdas e duplicações.

2 Conclusão

Com a realização deste trabalho podemos por em prática alguns dos aspetos que aprendemos nas aulas teóricas, tais como, sermos capazes de classificar os protocolos de transporte, respetivamente UDP e TCP, em diversas aplicações diferentes e o seu respetivo overhead.

De seguida cada uma destas aplicações foi testada utilizando a topologia CORE fornecida. Foi feita uma análise dos resultados entre as várias aplicações, sendo possível estabelecer uma conexão entre a consequência do uso de diferentes protocolos de transporte em diferentes aplicações.

Finalmente, foi feita uma análise numa situação de package loss de forma a perceber como os mecanismos de correção de erros funcionam.

Assim sendo, com a elaboração deste trabalho foram aprofundados os conhecimentos relativos à camada de transporte, cruciais para a melhor compreensão do seu funcionamento e importância.