

UNIVERSIDADE DO MINHO

DEPARTAMENTO DE INFORMÁTICA

Comunicação por Computadores
Fase 1 - Grupo 7

José Ferreira (A83683)

João Teixeira (A85504)

Miguel Solino (A86435)

4 de Março de 2020

Capítulo 1

Questão 1

<i>Comando Usado (Aplicação)</i>	Protocolo de Aplicação	Protocolo de transporte	Porta de atendimento	Overhead de transporte em bytes
Ping	-	-	-	-
Tracerout	-	UDP	33446	8
telnet	telnet	TCP	23	20
ftp	ftp	TCP	21	20
Tftp	tftp	UDP	69	8
browser/http	http	TCP	80	20
ssh	sshv2	TCP	22	20

1	0.000000	10.0.2.15	193.137.16.65	DNS	75 Standard query A marco.uminho.pt
2	0.003654	193.137.16.65	10.0.2.15	DNS	347 Standard query response A 193.136.9.240
3	0.004209	10.0.2.15	193.136.9.240	ICMP	98 Echo (ping) request id=0x3409, seq=1/256, ttl=64
4	0.007377	193.136.9.240	10.0.2.15	ICMP	98 Echo (ping) reply id=0x3409, seq=1/256, ttl=63
5	0.007721	10.0.2.15	193.137.16.65	DNS	86 Standard query PTR 240.9.136.193.in-addr.arpa
6	0.011754	193.137.16.65	10.0.2.15	DNS	402 Standard query response PTR marco.uminho.pt
7	1.005435	10.0.2.15	193.136.9.240	ICMP	98 Echo (ping) request id=0x3409, seq=2/512, ttl=64
8	1.008049	193.136.9.240	10.0.2.15	ICMP	98 Echo (ping) reply id=0x3409, seq=2/512, ttl=63
9	1.008372	10.0.2.15	193.137.16.65	DNS	86 Standard query PTR 240.9.136.193.in-addr.arpa
10	1.009995	193.137.16.65	10.0.2.15	DNS	402 Standard query response PTR marco.uminho.pt

.....

▸ Frame 3: 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: 193.136.9.240 (193.136.9.240)

▸ Internet Control Message Protocol

.....

0000 52 54 00 12 35 02 08 00 27 78 e5 64 08 00 45 00 RT..S... 'x.d..E.

0010 00 54 4d 3c 40 00 40 01 15 e6 0a 00 02 0f c1 88 .TM<@.e.

0020 09 f0 08 00 a0 55 34 09 00 01 08 25 4d 5e d9 19U4. ...%M^..

0030 0a 00 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15

eth0: <live capture in progress> File: /tmp/... Packets: 10 Displayed: 10 Marked: 0 Profile: ...

Figura 1.1: PING

22	0.067139	10.0.2.15	193.136.19.254	UDP	74	Source port: 45796	Destination port: 33446
23	0.067335	10.0.2.15	193.136.19.254	UDP	74	Source port: 44098	Destination port: 33447
24	0.067559	10.0.2.15	193.136.19.254	UDP	74	Source port: 54326	Destination port: 33448
25	0.067765	10.0.2.15	193.136.19.254	UDP	74	Source port: 36756	Destination port: 33449
26	0.068224	10.0.2.15	193.137.16.65	DNS	81	Standard query PTR 2.2.0.10.in-addr.arpa	
27	0.073617	172.26.254.254	10.0.2.15	ICMP	74	Time-to-live exceeded (Time to live exceeded in transit)	
28	0.074240	172.26.254.254	10.0.2.15	ICMP	74	Time-to-live exceeded (Time to live exceeded in transit)	
29	0.074353	172.26.254.254	10.0.2.15	ICMP	74	Time-to-live exceeded (Time to live exceeded in transit)	
Frame 22: 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: 45796 (45796), Dst Port: 33446 (33446)							
Source port: 45796 (45796)							
Destination port: 33446 (33446)							
Length: 40							
Checksum: 0xe1ce [validation disabled]							
Data (32 bytes)							
0020 13 fe b2 e4 82 a6 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 WXYZ[\] ^_							
User Datagram Protocol (udp), 8 bytes							
Packets: 159 Displayed: 159 Marked: 0							
Profile: ...							

Figura 1.2: TRACEROUTE

10	0.053753	10.0.2.15	193.136.9.183	TELNET	81	Telnet Data ...	
11	0.053931	193.136.9.183	10.0.2.15	TCP	60	telnet > 46077 [ACK] Seq=1 Ack=28 Win=65535 Len=0	
12	15.101883	193.136.9.183	10.0.2.15	TELNET	66	Telnet Data ...	
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: 46077 (46077), Dst Port: telnet (23), Seq: 1, Ack: 1, Len: 27							
Source port: 46077 (46077)							
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]							
Telnet							
0020 09 b7 b3 fd 00 17 d7 64 96 51 02 39 56 02 50 18d .0.9V.P.							
0030 39 08 d7 83 00 00 ff fd 03 ff fb 18 ff fb 1f ff 9.....							
0040 fb 20 ff fb 21 ff fb 22 ff fb 27 ff fd 05 ff fb"							
0050 23 #							
Transmission Control Protocol (tcp), 20 bytes							
Packets: 198 Displayed: 198 Marked: 0							
Profile: ...							

Figura 1.3: TELNET

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help							
Filter: Expression...							
No.	Time	Source	Destination	Protocol	Length	Info	
20	16.190671	193.137.16.65	10.0.2.15	DNS	172	St	
21	16.190994	10.0.2.15	193.137.16.65	DNS	86	St	
22	16.192769	193.137.16.65	10.0.2.15	DNS	410	St	
23	16.192966	10.0.2.15	193.136.9.183	TCP	74	45	
24	16.195564	193.136.9.183	10.0.2.15	TCP	60	ft	
25	16.195610	10.0.2.15	193.136.9.183	TCP	54	45	
26	16.228557	193.136.9.183	10.0.2.15	FTP	74	Re	
27	16.228818	10.0.2.15	193.136.9.183	TCP	54	45	
28	21.117883	10.0.2.15	193.136.9.183	FTP	63	Re	
29	21.118439	193.136.9.183	10.0.2.15	TCP	60	ft	
30	21.121164	193.136.9.183	10.0.2.15	FTP	88	Re	
31	21.121226	10.0.2.15	193.136.9.183	TCP	54	45	
Frame 25: 54 bytes on wire (432 bits), 54 bytes captured (432 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: 45008 (45008), Dst Port: ftp (21)							
Source port: 45008 (45008)							
Destination port: ftp (21)							
[Stream index: 7]							
Sequence number: 1 (relative sequence number)							
Acknowledgement number: 1 (relative ack number)							
Header length: 20 bytes							
Flags: 0x010 (ACK)							
Window size value: 14600							
[Calculated window size: 14600]							
[Window size scaling factor: -2 (no window scaling used)]							
Checksum: 0xd768 [validation disabled]							
[SEQ/ACK analysis]							

Figura 1.4: FTP

7	0.011147	10.0.2.15	193.136.9.183	TFTP	86 Read Request, File: file1, Transfer type: octet, tsize\000=0\000, blksize\000=512
8	0.014450	10.0.2.2	10.0.2.15	UDP	76 Source port: 36381 Destination port: 43550
9	0.014563	10.0.2.15	10.0.2.2	UDP	46 Source port: 43550 Destination port: 36381
10	0.019017	10.0.2.2	10.0.2.15	UDP	239 Source port: 36381 Destination port: 43550
11	0.019686	10.0.2.15	10.0.2.2	UDP	46 Source port: 43550 Destination port: 36381
12	5.002453	CadmusCo_78:e5:64	RealtekU_12:35:02	ARP	42 Who has 10.0.2.2? Tell 10.0.2.15
13	5.002968	RealtekU_12:35:02	CadmusCo_78:e5:64	ARP	60 10.0.2.2 is at 52:54:00:12:35:02
▶ Frame 7: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0 ▶ 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: 43550 (43550), Dst Port: tftp (69)					
Source port: 43550 (43550) Destination port: tftp (69) Length: 52 ▶ Checksum: 0xd793 [validation disabled]					
▶ Trivial File Transfer Protocol					
0020 09 b7 aa 1e 00 45 00 34 d7 93 00 01 66 69 6c 65E.4...file 0030 31 00 6f 63 74 65 74 00 74 73 69 7a 65 00 30 00tsize.0. 0040 62 6c 6b 73 69 7a 65 00 35 31 32 00 74 69 6d 65blksize. 512.time 0050 6f 75 74 00 36 00out.6.					
[User Datagram Protocol (udp), 8 bytes] [Packets: 13 Displayed: 13 Marked: 0] [Profile: ...]					

Figura 1.5: TFTP

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help							▶ Frame 10: 188 bytes on wire (1504 bits), 188 bytes captured (1504 bits)	
[Icons]							▶ Ethernet II, Src: CadmusCo_78:e5:64 (08:00:27:78:e5:64), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)	
Filter: [] Expression: []							▶ Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 193.136.9.240 (193.136.9.240)	
							▶ Transmission Control Protocol, Src Port: 49868 (49868), Dst Port: http (80)	
							Source port: 49868 (49868)	
							Destination port: http (80)	
							[Stream index: 3]	
							Sequence number: 1 (relative sequence number)	
							[Next sequence number: 135 (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: 0xd827 [validation disabled]	
							▶ [SEQ/ACK analysis]	
							▶ Hypertext Transfer Protocol	

No.	Time	Source	Destination	Protocol	Length	Info
5	0.445758	10.0.2.15	193.137.16.65	DNS	75	St
6	0.469214	193.137.16.65	10.0.2.15	DNS	347	St
7	0.469341	10.0.2.15	193.136.9.240	TCP	74	49
8	0.476571	193.136.9.240	10.0.2.15	TCP	60	ht
9	0.476605	10.0.2.15	193.136.9.240	TCP	54	49
10	0.476867	10.0.2.15	193.136.9.240	HTTP	188	GE
11	0.477148	193.136.9.240	10.0.2.15	TCP	60	ht
12	0.490955	193.136.9.240	10.0.2.15	HTTP	667	HT
13	0.490986	10.0.2.15	193.136.9.240	TCP	54	49
14	0.491618	10.0.2.15	193.136.9.240	HTTP	189	GE
15	0.491958	193.136.9.240	10.0.2.15	TCP	60	ht
16	0.701828	193.136.9.240	10.0.2.15	TCP	1292	IT

Figura 1.6: HTTP/browser

10	1.770431	193.136.9.183	10.0.2.15	SSHv2	95 Server Protocol: SSH-2.0-OpenSSH_5.9p1 Debian-Subuntu1.4\r
11	1.770548	10.0.2.15	193.136.9.183	TCP	54 48638 > ssh [ACK] Seq=1 Ack=42 Win=14600 Len=0
12	1.771019	10.0.2.15	193.136.9.183	SSHv2	95 Client Protocol: SSH-2.0-OpenSSH_5.9p1 Debian-Subuntu1.4\r
▶ Frame 10: 95 bytes on wire (760 bits), 95 bytes captured (760 bits) on interface 0 ▶ 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: 48638 (48638), Seq: 1, Ack: 1, Len: 41					
Source port: ssh (22) Destination port: 48638 (48638) [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 [Calculated window size: 65535] [Window size scaling factor: -2 (no window scaling used)] ▶ Checksum: 0x6c4d [validation disabled] ▶ [SEQ/ACK analysis]					
0020 02 0f 00 16 bd fe 02 ac 92 02 8f b8 97 6c 50 18lp. 0030 ff ff 6c 4d 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 69enSSH_5. 9p1 Debi 0050 61 6e 2d 35 75 62 75 6e 74 75 31 2e 34 0d 0aan-Subun tu1.4..					

Figura 1.7: SSH

1	0.000000	10.0.2.15	193.137.16.65	DNS	73 Standard query A www.uminho.pt
2	0.002952	193.137.16.65	10.0.2.15	DNS	345 Standard query response A 193.137.9.114

▶ 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: 53118 (53118), Dst Port: domain (53) Source port: 53118 (53118) Destination port: domain (53) Length: 39 ▶ Checksum: 0xde11 [validation disabled] ▶ Domain Name System (query)

0020	10 41	cf 7e 00 35 00 27 de 11	5d db 01 00 00 01	.A.-.5..].....
0030	00 00 00 00 00 00 03 77	77 77 06 75 6d 69 6e 68	w ww.uminh
0040	6f 02 70 74 00 00 01 00	01		o.pt.... .

○	User Datagram Protocol (udp), 8 bytes	Packets: 2 Displayed: 2 Marked: 0	Profile: ...
---	---------------------------------------	-----------------------------------	--------------

Figura 1.8: nslookup

Capítulo 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.

60	75.354375	10.1.1.1	10.3.3.1	FTP	78 Request: RETR file1
61	75.355592	10.3.3.1	10.1.1.1	TCP	74 ftp-data > 54792 [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=1396260 TSecr=1396261
62	75.355885	10.1.1.1	10.3.3.1	TCP	74 54792 > ftp-data [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval=1396260 TSecr=1396261
63	75.356066	10.3.3.1	10.1.1.1	TCP	66 ftp-data > 54792 [ACK] Seq=1 Ack=1 Win=14608 Len=0 TSval=1396261 TSecr=1396261
64	75.356118	10.3.3.1	10.1.1.1	FTP	130 Response: 150 Opening BINARY mode data connection for file1 (193 bytes).
65	75.356753	10.3.3.1	10.1.1.1	FTP-DATA	259 FTP Data: 193 bytes
66	75.356753	10.3.3.1	10.1.1.1	TCP	66 ftp-data > 54792 [FIN, ACK] Seq=194 Ack=1 Win=14608 Len=0 TSval=1396261 TSecr=1396261
67	75.357153	10.1.1.1	10.3.3.1	TCP	66 54792 > ftp-data [ACK] Seq=1 Ack=194 Win=15552 Len=0 TSval=1396261 TSecr=1396261
68	75.357230	10.1.1.1	10.3.3.1	TCP	66 54792 > ftp-data [FIN, ACK] Seq=1 Ack=195 Win=15552 Len=0 TSval=1396261 TSecr=1396261
69	75.357373	10.3.3.1	10.1.1.1	TCP	66 ftp-data > 54792 [ACK] Seq=195 Ack=2 Win=14608 Len=0 TSval=1396261 TSecr=1396261
70	75.357437	10.3.3.1	10.1.1.1	FTP	90 Response: 226 Transfer complete.

Figura 2.1: FTP

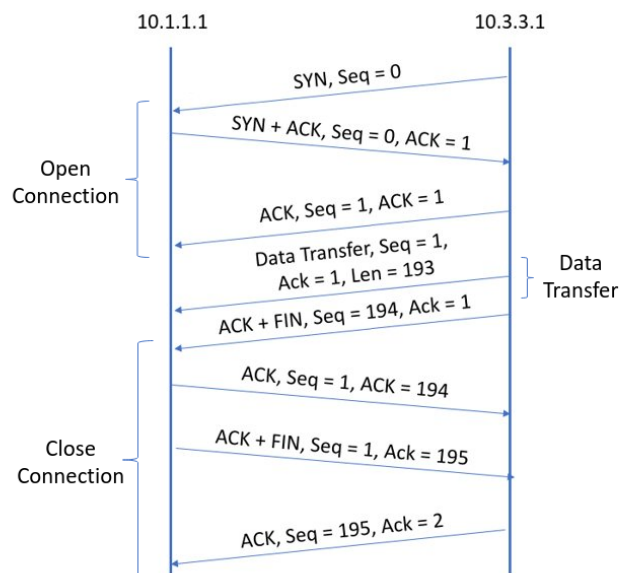


Figura 2.2: FTP- Diagrama Temporal

5	10.792420	10.1.1.1	10.3.3.1	TFTP	56 Read Request, File: file1, Transfer type: octet
8	10.795116	10.3.3.1	10.1.1.1	TFTP	239 Data Packet, Block: 1 (last)
9	10.795736	10.1.1.1	10.3.3.1	TFTP	46 Acknowledgement, Block: 1

Figura 2.3: TFTP

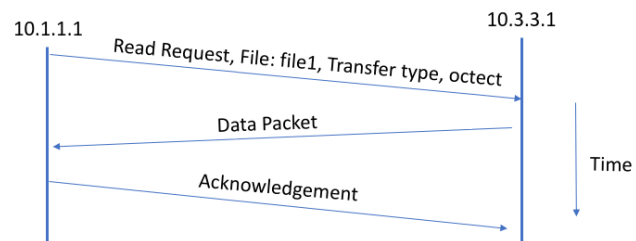


Figura 2.4: TFTP - Diagrama Temporal

Capítulo 3

Questão 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;

3.1 SFTP

3.1.1 Uso da camada de transporte

Protocolo TCP.

3.1.2 Eficiência na transferência

Devido à sua fiabilidade não é muito eficiente.

3.1.3 Complexidade

Disponibiliza várias funcionalidades. logo torna-se muito complexo.

3.1.4 Segurança

Recorre à autenticação e à encriptação dos dados, ou seja pode se dizer que é seguro.

3.2 FTP

3.2.1 Uso da camada de transporte

Protocolo TCP.

3.2.2 Eficiência na transferência

Tem um maior overhead e isso deve-se a ser fiável.

3.2.3 Complexidade

Sendo que garante a segurança de transferência de dados, torna-se complexo.

3.2.4 Segurança

É pouco seguro, contudo utiliza autenticação.

3.3 TFTP

3.3.1 Uso da camada de transporte

Protocolo UDP.

3.3.2 Eficiência na transferência

Possui um overhead menor não se responsabilizando pela entrega dos dados.

3.3.3 Complexidade

Sendo que utiliza um protocolo UDP, as funcionalidades existentes não são várias, logo não é muito complexo.

3.3.4 Segurança

Não utiliza qualquer mecanismo de autenticação ou encriptação, sendo assim pouco seguro.

3.4 HTTP

3.4.1 Uso da camada de transporte

Protocolo TCP.

3.4.2 Eficiência na transferência

Possui uma grande eficiência.

3.4.3 Complexidade

Mesmo utilizando um protocolo TCP, trata-se de uma aplicação pouco complexa.

3.4.4 Segurança

Sendo que só utiliza autenticação, não é muito seguro.

Capítulo 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).

```
root@Alfa:/tmp/pycore.4/503/Alfa.conf# cat file-output-ping
PING 10.3.3.1 (10.3.3.1) 56(84) bytes of data.
64 bytes from 10.3.3.1: icmp_req=3 ttl=61 time=5.66 ms
64 bytes from 10.3.3.1: icmp_req=4 ttl=61 time=5.50 ms
64 bytes from 10.3.3.1: icmp_req=5 ttl=61 time=5.38 ms
64 bytes from 10.3.3.1: icmp_req=6 ttl=61 time=5.50 ms
64 bytes from 10.3.3.1: icmp_req=7 ttl=61 time=5.54 ms
64 bytes from 10.3.3.1: icmp_req=8 ttl=61 time=5.39 ms
64 bytes from 10.3.3.1: icmp_req=9 ttl=61 time=5.53 ms
64 bytes from 10.3.3.1: icmp_req=10 ttl=61 time=5.56 ms
64 bytes from 10.3.3.1: icmp_req=11 ttl=61 time=5.63 ms
64 bytes from 10.3.3.1: icmp_req=11 ttl=61 time=5.64 ms (DUP!)
64 bytes from 10.3.3.1: icmp_req=12 ttl=61 time=5.55 ms
64 bytes from 10.3.3.1: icmp_req=13 ttl=61 time=5.65 ms
64 bytes from 10.3.3.1: icmp_req=14 ttl=61 time=5.52 ms
64 bytes from 10.3.3.1: icmp_req=15 ttl=61 time=5.38 ms
64 bytes from 10.3.3.1: icmp_req=16 ttl=61 time=7.96 ms
64 bytes from 10.3.3.1: icmp_req=17 ttl=61 time=6.30 ms
64 bytes from 10.3.3.1: icmp_req=19 ttl=61 time=5.53 ms
64 bytes from 10.3.3.1: icmp_req=20 ttl=61 time=6.40 ms

--- 10.3.3.1 ping statistics ---
20 packets transmitted, 17 received, +1 duplicates, 15% packet loss, time 19048ms
rtt min/avg/max/mdev = 5.386/5.761/7.967/0.603 ms
```

Figura 4.1: Ping

Observando a figura 4.1, podemos ver que existiu 15% de perda de pacotes e 1 foi duplicado. Maior parte das vezes isto deve-se a existirem problemas ao nível da ligação de rede. Quando se trata de protocolos de transporte existem dois principais que tratam deste problema de formas diferentes, sendo eles TCP e UDP. O primeiro é o mais complexo devido a utilizar mecanismos de deteção e recuperação, garantindo que todos os pacotes são enviados na ordem correta e sem erros. Contudo, esta complexidade também tem os seus defeitos. Como são trocados imensos pacotes por ambas as partes, quando nos encontramos numa rede de menor qualidade são corrompidos e perdidos. Com isso, são enviadas mensagens de erro e reenviados os pacotes originando numa maior sobrecarga. Já a segunda é um protocolo muito menos complexo o que permite a própria aplicação garantir que os dados são corretamente recebidos. No entanto, sendo pouco complexo não existe muito controlo caso os pacotes sejam perdidos, necessitando que os protocolos acima do UDP identifiquem e corrijam o problema.

Capítulo 5

Conclusão

Na realização deste trabalho concluímos que maior parte dos conteúdos lecionados nas aulas teóricas foram colocados em prática.

Na primeira parte foram vistos e aplicados vários protocolos de transporte e aplicativos.

Já na segunda, os conceitos TCP e UDP receberam uma maior atenção garantindo que a nossa percepção dos mesmos ficasse clara.

As principais ferramentas utilizadas para que isto fosse possível foram o Core e o Wireshark, sendo que maior parte do tempo utilizado para este TP foi na análise de tráfego colocando em uso os vários protocolos de Aplicação e Transporte.