

---

# Comunicações por Computador - TP1

---

## TRABALHO REALIZADO POR:

CARLOS MIGUEL LUZIA DE CARVALHO

GONÇALO DE SÁ QUENTAL ROSA MEDEIROS

ANTONIO JORGE NANDE RODRIGUES



A89605  
Carlos Carvalho



A89514  
Gonçalo Medeiros



A89585  
António Rodrigues

GRUPO 7  
PL 6  
2020/2021  
UNIVERSIDADE DO MINHO

# Índice

<b>1 Pergunta 1</b>	<b>1</b>
1.1 Tabela . . . . .	1
1.2 Auxiliares à resposta . . . . .	1
1.2.1 Ping . . . . .	1
1.2.2 Traceroute . . . . .	2
1.2.3 telnet . . . . .	2
1.2.4 ftp . . . . .	3
1.2.5 tftp . . . . .	3
1.2.6 browser/http . . . . .	4
1.2.7 nslookup . . . . .	4
1.2.8 ssh . . . . .	5
<b>2 Pergunta 2</b>	<b>6</b>
2.1 FTP . . . . .	6
2.2 TFTP . . . . .	7
<b>3 Pergunta 3</b>	<b>8</b>
<b>4 Pergunta 4</b>	<b>9</b>
<b>5 Conclusão</b>	<b>12</b>
<b>6 Anexo</b>	<b>13</b>
6.1 Ping . . . . .	13
6.2 SFTP . . . . .	14
6.3 FTP . . . . .	16
6.4 TFTP . . . . .	18
6.5 HTTP . . . . .	21

# 1 Pergunta 1

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

## 1.1 Tabela

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	DNS	UDP	53	8
tracert/route	TRACEROUTE	UDP	33451	8
telnet	TELNET	TCP	53420	20
ftp	FTP	TCP	46476	20
Tftp	TFTP	UDP	69	8
browser/http	HTTP	TCP	80	20
nslookup	DNS	UDP	56776	8
ssh	SSH	TCP	58382	41*

O protocolo de transporte TCP tem 20 bytes porém se tiver flags esse tamanho poderá ser maior, podendo variar conforme usando o campo options ou não, por outro lado o protocolo UDP têm, como podemos observar nas imagens abaixo, valores no **length** superiores a 8, no entanto o valor do overhead é fixo e é sempre 8, sendo que os bytes restantes correspondem à informação transportada.

## 1.2 Auxiliares à resposta

### 1.2.1 Ping

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	198.211.97.130	ICMP	98	Echo (ping) request id=0x0005, seq=1/256, ttl=64 (reply in 2)
2	0.121592390	198.211.97.130	10.0.2.15	ICMP	98	Echo (ping) reply id=0x0005, seq=1/256, ttl=49 (request in...)
3	0.122055545	10.0.2.15	192.168.1.254	DNS	98	Standard query 0x9516 PTR 130.97.211.198.in-addr.arpa.0pi
4	0.358109545	192.168.1.254	10.0.2.15	DNS	165	Standard query response 0x9516 No such name PTR 130.97.211.19...
5	0.358251788	10.0.2.15	192.168.1.254	DNS	87	Standard query 0x9516 PTR 130.97.211.198.in-addr.arpa
6	0.636551206	192.168.1.254	10.0.2.15	DNS	154	Standard query response 0x9516 No such name PTR 130.97.211.19...
7	1.001520623	10.0.2.15	198.211.97.130	ICMP	98	Echo (ping) request id=0x0005, seq=2/512, ttl=64 (reply in 8)
8	1.121859577	198.211.97.130	10.0.2.15	ICMP	98	Echo (ping) reply id=0x0005, seq=2/512, ttl=49 (request in...)
9	2.003439971	10.0.2.15	198.211.97.130	ICMP	98	Echo (ping) request id=0x0005, seq=3/768, ttl=64 (reply in 1...)
10	2.125640288	198.211.97.130	10.0.2.15	ICMP	98	Echo (ping) reply id=0x0005, seq=3/768, ttl=49 (request in...)
11	5.050013144	PcsCompu_d1:8b:d0	RealtekU_12:35:02	ARP	42	Who has 10.0.2.2? Tell 10.0.2.15
12	5.050152525	RealtekU_12:35:02	PcsCompu_d1:8b:d0	ARP	60	10.0.2.2 is at 52:54:00:12:35:02

Frame 3: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface enp0s3, id 0

Ethernet II, Src: PcsCompu\_d1:8b:d0 (08:00:27:dd:8b:d0), 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: 38931, Dst Port: 53

Source Port: 38931

Destination Port: 53

Length: 64

Checksum: 0xc0f6 [unverified]

Checksum Status: Unverified

Stream index: 0

Timestamps

Domain Name System (query)

Transaction ID: 0x9516

Flags: 0x0100 Standard query

Questions: 1

Answer RRs: 0

Authority RRs: 0

Additional RRs: 1

Queries

Additional records

Response In: 4

## 1.2.2 Traceroute

No.	Time	Source	Destination	Protocol	Length	Info
18	0.073305836	10.0.2.15	193.137.196.247	UDP	74	57386 → 33444 Len=32
19	0.073375013	10.0.2.15	193.137.196.247	UDP	74	41489 → 33445 Len=32
20	0.073514433	10.0.2.15	193.137.196.247	UDP	74	46705 → 33446 Len=32
21	0.073544791	10.0.2.15	193.137.196.247	UDP	74	41463 → 33447 Len=32
22	0.073633516	10.0.2.15	193.137.196.247	UDP	74	46553 → 33448 Len=32
23	0.073666497	10.0.2.15	193.137.196.247	UDP	74	44814 → 33449 Len=32
24	0.074052707	10.0.2.15	192.168.1.254	DNS	92	Standard query 0x7f81 PTR 2.2.0.10.in-addr.arpa OPT
25	0.075266897	10.0.2.2	10.0.2.15	ICMP	70	Destination unreachable (Port unreachable)
26	0.075266956	10.0.2.2	10.0.2.15	ICMP	70	Destination unreachable (Port unreachable)
27	0.075267004	10.0.2.2	10.0.2.15	ICMP	70	Destination unreachable (Port unreachable)
28	0.099294092	10.0.2.2	10.0.2.15	ICMP	70	Destination unreachable (Port unreachable)
29	0.099294178	10.0.2.2	10.0.2.15	ICMP	70	Destination unreachable (Port unreachable)
30	0.099650919	10.0.2.2	10.0.2.15	ICMP	70	Destination unreachable (Port unreachable)
31	0.099838725	10.0.2.2	10.0.2.15	ICMP	70	Destination unreachable (Port unreachable)
32	0.100165336	10.0.2.2	10.0.2.15	ICMP	70	Destination unreachable (Port unreachable)
33	0.100339082	10.0.2.2	10.0.2.15	ICMP	70	Destination unreachable (Port unreachable)
34	0.100566113	10.0.2.2	10.0.2.15	ICMP	70	Destination unreachable (Port unreachable)
35	0.109565299	192.168.1.254	10.0.2.15	DNS	151	Standard query response 0x7f81 No such name PTR 2.2.0.10.in-addr.arpa
36	0.109700231	10.0.2.15	192.168.1.254	DNS	81	Standard query 0x7f81 PTR 2.2.0.10.in-addr.arpa
37	0.139591280	192.168.1.254	10.0.2.15	DNS	140	Standard query response 0x7f81 No such name PTR 2.2.0.10.in-addr.arpa
38	0.140407738	10.0.2.15	193.137.196.247	UDP	74	39251 → 33450 Len=32
39	0.140437576	10.0.2.15	193.137.196.247	UDP	74	41001 → 33451 Len=32
40	0.140536644	10.0.2.15	193.137.196.247	UDP	74	49719 → 33452 Len=32
41	0.140872625	10.0.2.15	192.168.1.254	DNS	92	Standard query 0xe8e2 PTR 2.2.0.10.in-addr.arpa OPT
42	0.163862950	10.0.2.2	10.0.2.15	ICMP	70	Destination unreachable (Port unreachable)
43	0.163941832	10.0.2.2	10.0.2.15	ICMP	70	Destination unreachable (Port unreachable)
44	0.164791727	10.0.2.2	10.0.2.15	ICMP	70	Destination unreachable (Port unreachable)
45	0.170068953	192.168.1.254	10.0.2.15	DNS	151	Standard query response 0xe8e2 No such name PTR 2.2.0.10.in-addr.arpa
46	0.170258126	10.0.2.15	192.168.1.254	DNS	81	Standard query 0xe8e2 PTR 2.2.0.10.in-addr.arpa
47	0.194973808	192.168.1.254	10.0.2.15	DNS	140	Standard query response 0xe8e2 No such name PTR 2.2.0.10.in-addr.arpa

Frame 39: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface enp0s3, id 0  
Ethernet II, Src: PcsCompu\_d1:8b:d0 (08:00:27:d1:8b:d0), Dst: RealtekU\_12:35:02 (52:54:00:12:35:02)  
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 193.137.196.247  
User Datagram Protocol, Src Port: 41001, Dst Port: 33451  
Source Port: 41001  
Destination Port: 33451  
Length: 40  
Checksum: 0x92c9 [unverified]  
[Checksum Status: Unverified]  
[Stream index: 20]  
[Timestamps]  
Data (32 bytes)

## 1.2.3 telnet

No.	Time	Source	Destination	Protocol	Length	Info
7	0.163355607	10.0.2.15	193.136.9.183	TCP	54	53420 → 23 [ACK] Seq=1 Ack=1 Win=64240 Len=0
8	0.163549822	10.0.2.15	193.136.9.183	TELNET	81	Telnet Data ...
9	0.164002562	193.136.9.183	10.0.2.15	TCP	60	23 → 53420 [ACK] Seq=1 Ack=28 Win=65535 Len=0
10	0.205567669	193.136.9.183	10.0.2.15	TELNET	60	Telnet Data ...
11	0.205582279	10.0.2.15	193.136.9.183	TCP	54	53420 → 23 [ACK] Seq=28 Ack=13 Win=64228 Len=0
12	0.275108947	193.136.9.183	10.0.2.15	TELNET	93	Telnet Data ...
13	0.275122965	10.0.2.15	193.136.9.183	TCP	54	53420 → 23 [ACK] Seq=28 Ack=52 Win=64189 Len=0
14	0.275279379	10.0.2.15	193.136.9.183	TELNET	150	Telnet Data ...
15	0.275531654	193.136.9.183	10.0.2.15	TCP	60	23 → 53420 [ACK] Seq=52 Ack=133 Win=65535 Len=0
16	0.303059302	193.136.9.183	10.0.2.15	TELNET	60	Telnet Data ...
17	0.304557417	10.0.2.15	193.136.9.183	TCP	54	53420 → 23 [ACK] Seq=133 Ack=55 Win=64186 Len=0
18	0.304670205	10.0.2.15	193.136.9.183	TELNET	57	Telnet Data ...
19	0.304944858	193.136.9.183	10.0.2.15	TCP	60	23 → 53420 [ACK] Seq=55 Ack=136 Win=65535 Len=0
20	0.332962952	193.136.9.183	10.0.2.15	TELNET	60	Telnet Data ...

Frame 16: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface enp0s3, id 0  
Ethernet II, Src: RealtekU\_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu\_d1:8b:d0 (08:00:27:d1:8b:d0)  
Internet Protocol Version 4, Src: 193.136.9.183, Dst: 10.0.2.15  
Transmission Control Protocol, Src Port: 23, Dst Port: 53420, Seq: 52, Ack: 133, Len: 3  
Source Port: 23  
Destination Port: 53420  
[Stream index: 0]  
[TCP Segment Len: 3]  
Sequence number: 52 (relative sequence number)  
Sequence number (raw): 516800953  
[Next sequence number: 55 (relative sequence number)]  
Acknowledgment number: 133 (relative ack number)  
Acknowledgment number (raw): 3802507699  
0101 .... = 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: 0x9d5d [unverified]  
[Checksum Status: Unverified]  
Urgent pointer: 0  
[SEQ/ACK analysis]  
[Timestamps]  
TCP payload (3 bytes)  
Telnet  
Do Echo  
Command: Do (253)  
Subcommand: Echo

## 1.2.4 ftp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	192.168.1.254	DNS	86	Standard query 0x66d8 A cc2021.ddns.net OPT
2	0.000166895	10.0.2.15	192.168.1.254	DNS	86	Standard query 0x8262 AAAA cc2021.ddns.net OPT
3	0.026460193	192.168.1.254	10.0.2.15	DNS	146	Standard query response 0x8262 AAAA cc2021.ddns.net SOA nf1.n...
4	0.061947602	192.168.1.254	10.0.2.15	DNS	102	Standard query response 0x66d8 A cc2021.ddns.net A 193.136.9...
5	0.082432771	10.0.2.15	193.136.9.183	TCP	74	46476 → 21 [SYN] Seq=9 Win=64240 Len=0 MSS=1460 SACK PERM=1 T...
6	0.097831461	193.136.9.183	10.0.2.15	TCP	60	21 → 46476 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
7	0.097062114	10.0.2.15	193.136.9.183	TCP	54	46476 → 21 [ACK] Seq=1 Ack=1 Win=64240 Len=0
8	0.145594852	193.136.9.183	10.0.2.15	FTP	74	Response: 220 (vsFTPD 2.3.5)
9	0.145614777	10.0.2.15	193.136.9.183	TCP	54	46476 → 21 [ACK] Seq=1 Ack=21 Win=64220 Len=0
10	0.926584864	10.0.2.15	193.136.9.183	FTP	63	Request: USER cc
11	0.926984842	193.136.9.183	10.0.2.15	TCP	60	21 → 46476 [ACK] Seq=21 Ack=10 Win=65535 Len=0
12	0.956915677	193.136.9.183	10.0.2.15	FTP	88	Response: 331 Please specify the password.
13	0.956942558	10.0.2.15	193.136.9.183	TCP	54	46476 → 21 [ACK] Seq=10 Ack=55 Win=64186 Len=0
14	3.435373860	10.0.2.15	193.136.9.183	FTP	67	Request: PASS cc2021
15	3.435739604	193.136.9.183	10.0.2.15	TCP	60	21 → 46476 [ACK] Seq=55 Ack=23 Win=65535 Len=0
16	3.541042647	193.136.9.183	10.0.2.15	FTP	77	Response: 230 Login successful.
17	3.541058632	10.0.2.15	193.136.9.183	TCP	54	46476 → 21 [ACK] Seq=23 Ack=78 Win=64163 Len=0
18	3.541216758	10.0.2.15	193.136.9.183	FTP	60	Request: SYST
19	3.541552881	193.136.9.183	10.0.2.15	TCP	60	21 → 46476 [ACK] Seq=78 Ack=29 Win=65535 Len=0
20	3.569817505	193.136.9.183	10.0.2.15	FTP	73	Response: 215 UNIX Type: L8
21	3.569829368	10.0.2.15	193.136.9.183	TCP	54	46476 → 21 [ACK] Seq=29 Ack=97 Win=64144 Len=0
						Frame 20: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface enp0s3, id 0
						Ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_d1:8b:d0 (08:00:27:d1:8b:d0)
						Internet Protocol Version 4, Src: 193.136.9.183, Dst: 10.0.2.15
						Transmission Control Protocol, Src Port: 21, Dst Port: 46476, Seq: 78, Ack: 29, Len: 19
						Source Port: 21
						Destination Port: 46476
						[Stream index: 0]
						[TCP Segment Len: 19]
						Sequence number: 78 (relative sequence number)
						Sequence number (raw): 534272079
						[Next sequence number: 97 (relative sequence number)]
						Acknowledgment number: 29 (relative ack number)
						Acknowledgment number (raw): 3586395024
						0101 .... = 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: 0x61fd [unverified]
						[Checksum Status: Unverified]
						Urgent pointer: 0
						[SEQ/ACK analysis]
						[Timestamps]
						TCP payload (19 bytes)
						File Transfer Protocol (FTP)
						215 UNIX Type: L8\r\n
						Response code: NAME system type (215)
						Response arg: UNIX Type: L8
						[Current working directory: ]

## 1.2.5 tftp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	192.168.1.254	DNS	86	Standard query 0x5797 A cc2021.ddns.net OPT
2	0.000196034	10.0.2.15	192.168.1.254	DNS	86	Standard query 0x8a50 AAAA cc2021.ddns.net OPT
3	0.061769624	192.168.1.254	10.0.2.15	DNS	146	Standard query response 0x5797 A cc2021.ddns.net A 193.136.9...
4	0.108742453	192.168.1.254	10.0.2.15	DNS	146	Standard query response 0x8a50 AAAA cc2021.ddns.net SOA nf1.n...
5	0.109097545	10.0.2.15	193.136.9.183	TFTP	86	Read Request, File: file1, Transfer type: octet, tsize=0, blk...
6	7.346927589	10.0.2.15	193.136.9.183	TFTP	86	Read Request, File: file1, Transfer type: octet, tsize=0, blk...
7	14.359085123	10.0.2.15	193.136.9.183	TFTP	86	Read Request, File: file1, Transfer type: octet, tsize=0, blk...
8	21.378327820	10.0.2.15	193.136.9.183	TFTP	86	Read Request, File: file1, Transfer type: octet, tsize=0, blk...
						Frame 6: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface enp0s3, id 0
						Ethernet II, Src: PcsCompu_d1:8b:d0 (08:00:27:d1:8b:d0), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
						Internet Protocol Version 4, Src: 10.0.2.15, Dst: 193.136.9.183
						User Datagram Protocol, Src Port: 52002, Dst Port: 69
						Source Port: 52002
						Destination Port: 69
						Length: 52
						Checksum: 0xd793 [unverified]
						[Checksum Status: Unverified]
						[Stream index: 2]
						[Timestamps]
						Trivial File Transfer Protocol
						Opcode: Read Request (1)
						Source File: file1
						Type: octet
						Option: tsize = 0
						Option: blksize = 512
						Option: timeout = 6
						-----
						[Stream index: 0]
						[TCP Segment Len: 19]
						Sequence number: 78 (relative sequence number)
						Sequence number (raw): 534272079
						[Next sequence number: 97 (relative sequence number)]
						Acknowledgment number: 29 (relative ack number)
						Acknowledgment number (raw): 3586395024
						0101 .... = 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: 0x61fd [unverified]
						[Checksum Status: Unverified]
						Urgent pointer: 0
						[SEQ/ACK analysis]
						[Timestamps]
						TCP payload (19 bytes)
						File Transfer Protocol (FTP)
						215 UNIX Type: L8\r\n
						Response code: NAME system type (215)
						Response arg: UNIX Type: L8
						[Current working directory: ]

## 1.2.6 browser/http

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	192.168.1.254	DNS	86	Standard query 0x39fa AAAA marco.uminho.pt OPT
2	0.042009826	192.168.1.254	10.0.2.15	DNS	140	Standard query response 0x39fa AAAA marco.uminho.pt SOA dns.u...
3	0.042230686	10.0.2.15	193.136.9.240	TCP	74	35794 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 T...
4	0.070544686	193.136.9.240	10.0.2.15	TCP	60	80 → 35794 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
5	0.070570148	10.0.2.15	193.136.9.240	TCP	54	35794 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
6	0.070603444	10.0.2.15	193.136.9.240	HTTP	216	GET /disciplinas/CC-MIEI/ HTTP/1.1\r\n
7	0.070945757	193.136.9.240	10.0.2.15	TCP	60	80 → 35794 [ACK] Seq=1 Ack=163 Win=65535 Len=0
8	0.099538297	193.136.9.240	10.0.2.15	TCP	1514	80 → 35794 [PSH, ACK] Seq=1 Ack=163 Win=65535 Len=1460 [TCP s...
9	0.099550007	10.0.2.15	193.136.9.240	TCP	54	35794 → 80 [ACK] Seq=163 Ack=1461 Win=62780 Len=0
10	0.100100212	193.136.9.240	10.0.2.15	TCP	2974	80 → 35794 [PSH, ACK] Seq=1461 Ack=163 Win=65535 Len=2920 [TC...
11	0.100105135	10.0.2.15	193.136.9.240	TCP	54	35794 → 80 [ACK] Seq=163 Ack=4381 Win=62780 Len=0
12	0.100516004	193.136.9.240	10.0.2.15	HTTP	4424	HTTP/1.1 200 OK (text/html)
13	0.100587766	10.0.2.15	193.136.9.240	TCP	54	35794 → 80 [ACK] Seq=163 Ack=8751 Win=61320 Len=0
14	0.101105266	10.0.2.15	193.136.9.240	TCP	54	35794 → 80 [FIN, ACK] Seq=163 Ack=8751 Win=62780 Len=0
15	0.101380204	193.136.9.240	10.0.2.15	TCP	60	80 → 35794 [ACK] Seq=8751 Ack=164 Win=65535 Len=0
16	0.129223808	193.136.9.240	10.0.2.15	TCP	60	80 → 35794 [FIN, ACK] Seq=8751 Ack=164 Win=65535 Len=0

Frame 6: 216 bytes on wire (1728 bits), 216 bytes captured (1728 bits) on interface enp0s3, id 0  
Ethernet II, Src: PcsCompu\_d1:8b:d0 (08:00:27:d1:8b:d0), 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: 35794, Dst Port: 80, Seq: 1, Ack: 1, Len: 162  
Source Port: 35794  
Destination Port: 80  
[Stream index: 0]  
[TCP Segment Len: 162]  
Sequence number: 1 (relative sequence number)  
Sequence number (raw): 1102550542  
[Next sequence number: 163 (relative sequence number)]  
Acknowledgment number: 1 (relative ack number)  
Acknowledgment number (raw): 581248002  
0101 .... = Header Length: 20 bytes (5)  
Flags: 0x018 (PSH, ACK)  
Window size value: 64240  
[Calculated window size: 64240]  
[Window size scaling factor: -2 (no window scaling used)]  
Checksum: 0xd843 [unverified]  
[Checksum Status: Unverified]  
Urgent pointer: 0  
[SEQ/ACK analysis]  
[Timestamps]  
TCP payload (162 bytes)  
Hypertext Transfer Protocol  
GET /disciplinas/CC-MIEI/ HTTP/1.1\r\n  
User-Agent: Wget/1.20.3 (linux-gnu)\r\n  
Accept: /\*\r\n  
Accept-Encoding: identity\r\n  
Host: marco.uminho.pt\r\n  
Connection: Keep-Alive\r\n  
\r\n  
[Full request URI: http://marco.uminho.pt/disciplinas/CC-MIEI/]  
[HTTP request 1/1]  
[Response in frame: 12]

## 1.2.7 nslookup

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	192.168.1.254	DNS	84	Standard query 0x9818 AAAA www.uminho.pt OPT
2	0.029038748	192.168.1.254	10.0.2.15	DNS	138	Standard query response 0x9818 AAAA www.uminho.pt SOA dns.umi...
3	5.139098733	PcsCompu_d1:8b:d0	RealtekU_12:35:02	ARP	42	Who has 10.0.2.2? Tell 10.0.2.15
4	5.139501048	RealtekU_12:35:02	PcsCompu_d1:8b:d0	ARP	60	10.0.2.2 is at 52:54:00:12:35:02

Frame 2: 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits) on interface enp0s3, id 0  
Ethernet II, Src: RealtekU\_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu\_d1:8b:d0 (08:00:27:d1:8b:d0)  
Internet Protocol Version 4, Src: 192.168.1.254, Dst: 10.0.2.15  
User Datagram Protocol, Src Port: 53, Dst Port: 56776  
Source Port: 53  
Destination Port: 56776  
Length: 104  
Checksum: 0x2f43 [unverified]  
[Checksum Status: Unverified]  
[Stream index: 0]  
[Timestamps]  
Domain Name System (response)  
Transaction ID: 0x9818  
Flags: 0x8180 Standard query response, No error  
Questions: 1  
Answer RRs: 0  
Authority RRs: 1  
Additional RRs: 1  
Queries  
Authoritative nameservers  
Additional records  
[Request in: 1]  
[Time: 0.029038748 seconds]  
[Calculated window size: 64240]  
[Window size scaling factor: -2 (no window scaling used)]  
Checksum: 0xd843 [unverified]  
[Checksum Status: Unverified]  
Urgent pointer: 0  
[SEQ/ACK analysis]  
[Timestamps]  
TCP payload (162 bytes)  
Hypertext Transfer Protocol  
GET /disciplinas/CC-MIEI/ HTTP/1.1\r\n  
User-Agent: Wget/1.20.3 (linux-gnu)\r\n  
Accept: /\*\r\n  
Accept-Encoding: identity\r\n  
Host: marco.uminho.pt\r\n  
Connection: Keep-Alive\r\n  
\r\n  
[Full request URI: http://marco.uminho.pt/disciplinas/CC-MIEI/]  
[HTTP request 1/1]  
[Response in frame: 12]

Terminal - core@core-VirtualBox: ~/Desktop

core@core-VirtualBox: ~/Desktop

core@core-VirtualBox: ~/Desktop

core@core-VirtualBox: ~/Desktop\$ nslookup www.uminho.pt

Server: 127.0.0.53  
Address: 127.0.0.53#53

Non-authoritative answer:  
Name: www.uminho.pt  
Address: 193.137.9.114

core@core-VirtualBox: ~/Desktop\$

## 1.2.8 ssh

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	192.168.1.254	DNS	86	Standard query 0xf946 A cc2021.ddns.net OPT
2	0.000321351	10.0.2.15	192.168.1.254	DNS	86	Standard query 0xb345 AAAA cc2021.ddns.net OPT
3	0.005147450	192.168.1.254	10.0.2.15	DNS	102	Standard query response 0xf946 A cc2021.ddns.net A 193.136.9...
4	0.115409715	192.168.1.254	10.0.2.15	DNS	146	Standard query response 0xb345 AAAA cc2021.ddns.net SOA nfi.n...
5	0.115602513	10.0.2.15	193.136.9.183	TCP	74	58382 → 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 T...
6	0.14401419	193.136.9.183	10.0.2.15	TCP	60	22 → 58382 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
7	0.144439619	10.0.2.15	193.136.9.183	TCP	54	58382 → 22 [ACK] Seq=1 Ack=1 Win=64240 Len=0
8	0.144879613	10.0.2.15	193.136.9.183	SSHv2	95	Client: Protocol (SSH-2.0-OpenSSH.8.2p1 Ubuntu-4ubuntu0.1)
9	0.145212689	193.136.9.183	10.0.2.15	TCP	60	22 → 58382 [ACK] Seq=1 Ack=42 Win=65535 Len=0
10	0.203921112	193.136.9.183	10.0.2.15	SSHv2	95	Server: Protocol (SSH-2.0-OpenSSH_5.9p1 Debian-5ubuntu1.4)
11	0.203962961	10.0.2.15	193.136.9.183	TCP	54	58382 → 22 [ACK] Seq=42 Ack=42 Win=64199 Len=0
12	0.204440003	10.0.2.15	193.136.9.183	SSHv2	1566	Client: Key Exchange Init
13	0.204826464	193.136.9.183	10.0.2.15	TCP	60	22 → 58382 [ACK] Seq=42 Ack=1502 Win=65535 Len=0
14	0.204826538	193.136.9.183	10.0.2.15	TCP	60	22 → 58382 [ACK] Seq=42 Ack=1554 Win=65535 Len=0
15	0.233686437	193.136.9.183	10.0.2.15	SSHv2	1038	Server: Key Exchange Init
16	0.233700302	10.0.2.15	193.136.9.183	TCP	54	58382 → 22 [ACK] Seq=1554 Ack=1026 Win=63960 Len=0
17	0.23377444	10.0.2.15	193.136.9.183	SSHv2	134	Client: Elliptic Curve Diffie-Hellman Key Exchange Init
18	0.234354410	193.136.9.183	10.0.2.15	TCP	60	22 → 58382 [ACK] Seq=1026 Ack=1634 Win=65535 Len=0

▶ Frame 10: 95 bytes on wire (760 bits), 95 bytes captured (760 bits) on interface enp0s3, id 0  
 ▶ Ethernet II, Src: RealtekU 12:35:02 (52:54:00:12:35:02), Dst: PcsCompu\_d1:8b:d0 (08:00:27:d1:8b:d0)  
 ▶ Internet Protocol Version 4, Src: 193.136.9.183, Dst: 10.0.2.15  
 ▶ Transmission Control Protocol, Src Port: 22, Dst Port: 58382, Seq: 1, Ack: 42, Len: 41  
   Source Port: 22  
   Destination Port: 58382  
   [Stream index: 0]  
   [TCP Segment Len: 41]  
   Sequence number: 1 (relative sequence number)  
   Sequence number (raw): 728192002  
   [Next sequence number: 42 (relative sequence number)]  
   Acknowledgment number: 42 (relative ack number)  
   Acknowledgment number (raw): 633580762  
   0101 ... = 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: 0xb009 [unverified]  
   [Checksum Status: Unverified]  
   Urgent pointer: 0  
   [SEQ/ACK analysis]  
   [Timestamps]  
   TCP payload (41 bytes)  
 ▶ SSH Protocol  
   Accept: /\*\*\r\n  
   Accept-Encoding: identity\r\n  
   Host: marco.uminho.pt\r\n  
   Connection: Keep-Alive\r\n  
   \r\n  
   [Full request URI: http://marco.uminho.pt/disciplinas/CC-MIEI/]  
   [HTTP request 1/1]  
   [Response in frame: 12]

```

Terminal - core@core-VirtualBox: ~/Desktop
core@core-VirtualBox: ~/Desktop
core@core-VirtualBox: ~/Desktop
core@core-VirtualBox: ~/Desktop$ ssh cc@cc2021.ddns.net
cc@cc2021.ddns.net's password:
Welcome to Ubuntu 12.04.5 LTS (GNU/Linux 3.2.0-75-generic i686)

 * Documentation:  https://help.ubuntu.com/

New release '14.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Mar  9 13:02:01 2021 from bl12-228-226.dsl.telepac.pt
  
```

Posto isto achamos necessário tomar mais algumas considerações relativamente às diferenças entre o TCP e o UDP. Concluindo assim que apesar de o TCP ser um serviço bastante confiável perde em relação ao UDP no que toca a velocidade.

	Multiplexagem e Desmultiplexagem	Controlo de Erros	Garantia de Ordem	Controlo de Conexão	Controlo de Congestão	Controlo de fluxo
TCP	V	V	V	V	V	V
UDP	V	F	F	F	F	F

---

## 2 Pergunta 2

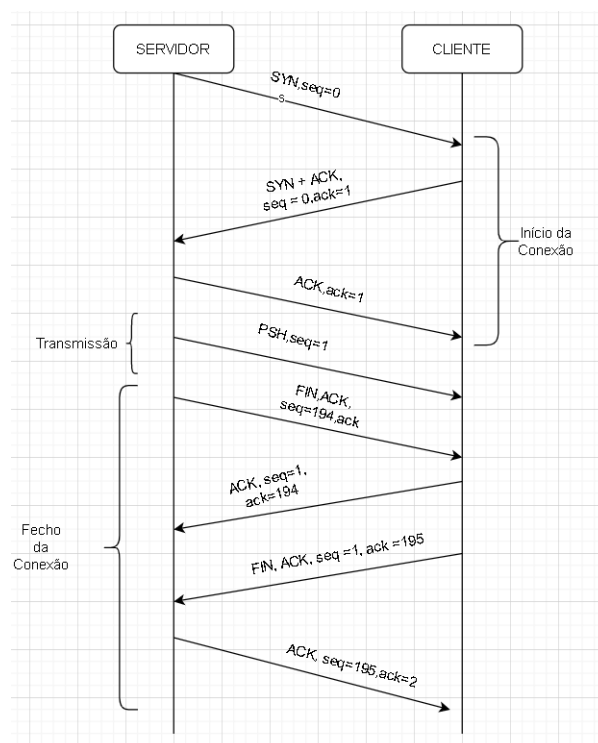
Uma representação num diagrama temporal das transferências da file1 por FTP e TFTP respetivamente. Se for caso disso, identifique as fases de estabelecimento de conexão, transferência de dados e fim de conexão. Identifica também claramente os tipos de segmentos trocados e os números de sequência usados quer nos dados como nas confirmações.

### 2.1 FTP

FTP significa Protocolo de Transferência de Arquivos e é usado para enviar/receber arquivos de um computador remoto. O FTP estabelece duas conexões entre o sistema cliente e o sistema servidor, uma para informações de controlo e outra para transferência de dados, as informações de controlo carregam comandos/respostas.

A autenticação precisa ser feita inicialmente por meio da validação de nome de usuário e senha. Uma vez feito isso, os arquivos podem ser transferidos entre dois sistemas. O FTP lida com arquivos binários e de formato de texto.

Quando um cliente FTP solicita a conexão com o servidor FTP, uma conexão TCP é estabelecida com a porta 21 do servidor FTP reservada para FTP. Após a autenticação, outra conexão TCP é estabelecida para a transferência real de dados na porta número 20, ou seja o FTP funciona em duas portas: 20 e 21 Uma para dados e outra para controle de conexão respetivamente.



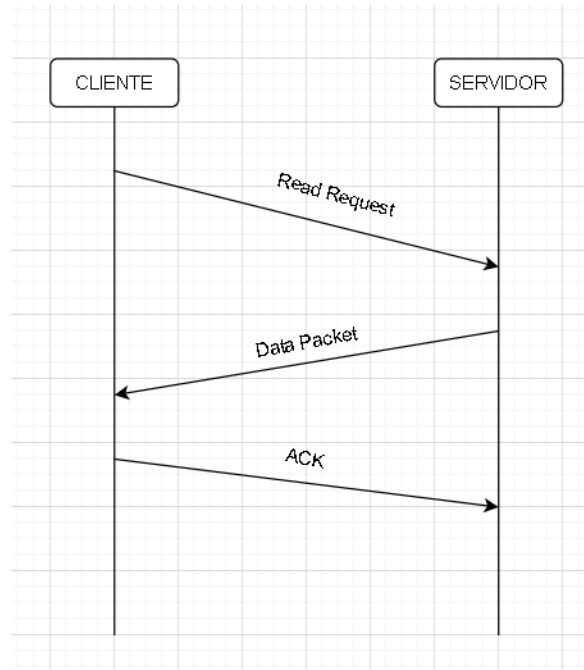
Linha temporal FTP



---

## 2.2 TFTP

TFTP significa Trivial File Transfer Protocol é mais simples que o FTP, pois faz a transferência de arquivos entre o processo do cliente e do servidor, mas não fornece autenticação de usuário e outros recursos úteis suportados pelo FTP. Enquanto FTP usa as portas 20 e 21 TCP, este usa unicamente a porta 69 UDP, acaba por ser mais limitado que o FTP devido ao uso do UDP em vez do TCP, sendo que este último permite significativamente uma complexidade superior.



Linha temporal TFTP

---

### 3 Pergunta 3

Com base nas experiências realizadas, distinga e compare sucintamente as quatro aplicações de transferência de arquivos que usou nos seguintes pontos (i) uso da camada de transporte; (ii) eficiência na transferência; (iii) complexidade; (iv) segurança;

#### (i) Uso da camada de transporte:

O TFTP (Trivial File Transfer Protocol) utiliza ao nível da camada de transporte o protocolo UDP (User Datagram Protocol) e as restantes aplicações FTP, SFTP e browser/HTTP utilizam o protocolo TCP (Transmission Control Protocol)

#### (ii) Eficiência no transporte

- **FTP:** utiliza o protocolo de transporte TCP que é conhecido pela sua fiabilidade na transmissão de pacotes. Após o envio do pacote desejado, este espera pela receção de uma mensagem de Acknowledgement (ACK) que confirma o sucesso da transmissão. Caso isto não se verifique, procede-se ao reenvio do pacote em causa garantindo a integridade dos pacotes mas à custa de eficiência.
- **TFTP:** utiliza o protocolo UDP que não é tão fiável pois não espera para receber uma mensagem de feedback (ACK) que confirma o sucesso da transmissão, todavia é exatamente por isto que o UDP apresenta vantagens na eficiência e consequente aumento de velocidade e redução de latência em comparação ao TCP-
- **SFTP:** À semelhança do FTP, mas neste protocolo a utilização do SSH garante encriptação dos dados.
- **HTTP:** O HTTP é um protocolo que funciona à base de envio de diferentes requests para executar diferentes tarefas, servindo-se do TCP para comunicar entre servidores.

#### (iii) Complexidade

Como já foi referido nas questões anteriores, reparamos que os protocolos SFTP, FTP, HTTP usam TCP e o protocolo TFTP usa UDP. Posto isto e analisando a complexidade dos protocolos de transporte TCP e UDP verificamos que no caso dos que usam o primeiro possuem um maior nível de complexidade já o que usa o segundo tem um grau de complexidade mais baixo.

No TFTP, que utiliza o UDP é realizada apenas a transferência de dados relativos ao ficheiro, ao invés dos restantes onde é detetado um mais elevado nível de complexidade, sendo que por exemplo, o FTP permite transferência de dados em paralelo e em cada transferência realiza uma nova conexão de dados, o que leva à necessidade de existirem diferentes velocidades de transferência. No caso do SFTP é um protocolo que permite acesso, transferência e gestão de dados possuindo assim altos custos de processamento, por último no caso do HTTP implementa um estilo que garante confiança, escalabilidade e desacoplamento de sistemas, o que leva a que também ele possuam uma maior complexidade.

---

#### (iv) Segurança

- **FTP:** O protocolo FTP é considerado inseguro dado que ao executar a sua autenticação utiliza informações como passwords em texto sem qualquer encriptação. Assim, se alguém proceder à captura deste tráfego terá acesso a todas estas informações delicadas tornando o FTP um protocolo que não prima pela segurança.
- **TFTP:** Este protocolo é ainda menos seguro que a sua versão mais complexa (o FTP). O TFTP não contempla autenticação do utilizador, não garantindo qualquer protecção de dados.
- **SFTP:** O SFTP utiliza o SSH (Secure Shell) para aumentar a segurança. Este consiste em fazer uma autenticação de dois fatores usando dois pares de chaves públicas, uma para autenticar o host remoto ao host local e o outro par para autenticar o host local ao host remoto. Isto faz com que o SFTP seja seguro, apenas as máquinas intervenientes têm acesso aos dados.
- **HTTP:** O protocolo HTTP (HyperText Transfer Protocol) é normalmente usado para transferência de dados através da internet não requerendo autenticação. À semelhança do FTP também envia informação em texto sem encriptação o que faz com que quem utilize este protocolo esteja sujeito à captura dos seus dados, não garantindo segurança.

## 4 Pergunta 4

As características das ligações de rede têm uma enorme influência nos níveis de Transporte e de Aplicação. Discuta, relacionando a resposta com as experiências realizadas, as influências das situações de perda ou duplicação de pacotes IP no desempenho global de Aplicações fiáveis (se possível, relacionando com alguns dos mecanismos de transporte envolvidos).

No momento da realização de uma transferência de dados, o protocolo de transporte a ser usado na mesma é definido pelo protocolo de aplicação, sendo que a decisão consiste em escolher o protocolo de transporte que mais se adequa à transferência. A duplicação e perda de pacotes IP ao nível das camadas de Transporte e Aplicação são acontecimentos recorrentes, existindo a necessidade de combater este problema. Como podemos verificar na topologia disponibilizada, a LAN3 é um exemplo desta situação, tal como podemos verificar pelas imagens no final desta secção e nas imagens contidas na secção "Anexo". Para solucionar esta dificuldade são usados os protocolos UDP e TCP, já falados anteriormente. Posto isto, temos algumas diferenças entre estes 2 protocolos. O UDP caracteriza-se por ser um protocolo orientado ao datagrama (não há conexão entre emissor e recetor), sem congestão de dados, ou seja, o desempenho não é afetado. Por outro lado, com este protocolo, existe a possibilidade de haver perdas de dados, apesar destas não afetarem o funcionamento da aplicação drasticamente. Esta perda dá-se pois o recetor não confirma que recebeu os dados. Isto é, caso o recetor não tenha recebido um pacote, por razão de algum motivo alheio, esse pacote é descartado e deixa de existir. Assim, não existe pedido de reenvio do pacote, não havendo influência no desempenho, nem a existência de pacotes duplicados. Para solucionar este problema é utilizada uma dependência de protocolo que se encontra acima do protocolo UDP e cuja função é identificar pacotes pelo seu ID ou número de sequência, conseguindo assim saber se todos os pacotes foram recebidos ou até se algum pacote se perdeu. Falando agora do

protocolo TCP, verificamos que existe conexão entre emissor e recetor, garantindo assim que o recetor recebe todos os dados emitidos pelo emissor. Durante o processo de transmissão de dados, existe a probabilidade de haver congestão dos mesmos, resultando assim em atrasos na transferência de pacotes, que podem ser mal interpretados e confundidos com perdas de dados conduzindo a um reenvio desnecessário de dados que afeta o desempenho e leva à duplicação de pacotes. Este protocolo faz a deteção e correção de erros, sendo que quando são encontrados pacotes com erros, existe uma tentativa de retransmissão dos mesmos, que leva a uma redução do débito de transmissão.

No.	Time	Source	Destination	Protocol	Length	Info
192	10.217519232	10.3.3.3	10.1.1.1	TCP	66	47955 → 20 [ACK] Seq=1 Ack=125977 Win=178432 Len=0 TSval=
193	10.217519773	10.3.3.3	10.1.1.1	TCP	66	47955 → 20 [ACK] Seq=1 Ack=127425 Win=181248 Len=0 TSval=
194	10.217653238	10.3.3.3	10.1.1.1	TCP	66	47955 → 20 [ACK] Seq=1 Ack=128873 Win=184192 Len=0 TSval=
195	10.217653857	10.3.3.3	10.1.1.1	TCP	66	47955 → 20 [ACK] Seq=1 Ack=130321 Win=187136 Len=0 TSval=
196	10.217654399	10.3.3.3	10.1.1.1	TCP	66	47955 → 20 [ACK] Seq=1 Ack=131769 Win=189952 Len=0 TSval=
197	10.217710574	10.1.1.1	10.3.3.3	FTP-DA...	1514	FTP Data: 1448 bytes (PORT) (RETR file2)
198	10.217712323	10.1.1.1	10.3.3.3	FTP-DA...	1514	FTP Data: 1448 bytes (PORT) (RETR file2)
199	10.217825233	10.1.1.1	10.3.3.3	FTP-DA...	1514	FTP Data: 1448 bytes (PORT) (RETR file2)
200	10.217844843	10.1.1.1	10.3.3.3	FTP-DA...	1514	FTP Data: 1448 bytes (PORT) (RETR file2)
201	10.217845839	10.1.1.1	10.3.3.3	FTP-DA...	1514	FTP Data: 1448 bytes (PORT) (RETR file2)
202	10.217846675	10.1.1.1	10.3.3.3	FTP-DA...	1514	FTP Data: 1448 bytes (PORT) (RETR file2)
203	10.217847480	10.1.1.1	10.3.3.3	FTP-DA...	1514	FTP Data: 1448 bytes (PORT) (RETR file2)
204	10.222821700	10.3.3.3	10.1.1.1	TCP	66	47955 → 20 [ACK] Seq=1 Ack=133217 Win=192896 Len=0 TSval=
205	10.222822637	10.3.3.3	10.1.1.1	TCP	66	47955 → 20 [ACK] Seq=1 Ack=134665 Win=195840 Len=0 TSval=
206	10.222823228	10.3.3.3	10.1.1.1	TCP	66	[TCP Dup ACK 205#1] 47955 → 20 [ACK] Seq=1 Ack=134665 Win=
207	10.222981182	10.1.1.1	10.3.3.3	FTP-DA...	306	FTP Data: 240 bytes (PORT) (RETR file2)
208	10.223018541	10.3.3.3	10.1.1.1	TCP	66	48036 → 21 [ACK] Seq=75 Ack=246 Win=64256 Len=0 TSval=209
209	10.223020920	10.3.3.3	10.1.1.1	TCP	66	[TCP Dup ACK 208#1] 48036 → 21 [ACK] Seq=75 Ack=246 Win=0
210	10.223021484	10.3.3.3	10.1.1.1	TCP	66	47955 → 20 [ACK] Seq=1 Ack=136113 Win=198656 Len=0 TSval=
211	10.223022027	10.3.3.3	10.1.1.1	TCP	66	47955 → 20 [ACK] Seq=1 Ack=137561 Win=201600 Len=0 TSval=
212	10.223022566	10.3.3.3	10.1.1.1	TCP	66	47955 → 20 [ACK] Seq=1 Ack=139009 Win=204416 Len=0 TSval=
213	10.223023105	10.3.3.3	10.1.1.1	TCP	66	47955 → 20 [ACK] Seq=1 Ack=140457 Win=207360 Len=0 TSval=
214	10.223023646	10.3.3.3	10.1.1.1	TCP	66	47955 → 20 [ACK] Seq=1 Ack=141905 Win=210304 Len=0 TSval=
215	10.229009920	10.3.3.3	10.1.1.1	TCP	66	47955 → 20 [FIN, ACK] Seq=1 Ack=142146 Win=213120 Len=0
216	10.229214278	10.1.1.1	10.3.3.3	TCP	66	20 → 47955 [ACK] Seq=142146 Ack=2 Win=64256 Len=0 TSval=1
217	10.229336238	10.1.1.1	10.3.3.3	FTP	90	Response: 226 Transfer complete.
218	10.234402680	10.3.3.3	10.1.1.1	TCP	66	48036 → 21 [ACK] Seq=75 Ack=270 Win=64256 Len=0 TSval=209
219	10.246631564	10.1.1.254	224.0.0.5	OSPF	78	Hello Packet
220	12.247465421	10.1.1.254	224.0.0.5	OSPF	78	Hello Packet
221	13.816426817	10.3.3.3	10.1.1.1	FTP	72	Request: QUIT
222	13.816723919	10.1.1.1	10.3.3.3	FTP	80	Response: 221 Goodbye.
223	13.816727615	10.1.1.1	10.3.3.3	TCP	66	21 → 48036 [FIN, ACK] Seq=284 Ack=81 Win=65280 Len=0 TSv
224	13.822095934	10.3.3.3	10.1.1.1	TCP	66	48036 → 21 [ACK] Seq=81 Ack=284 Win=64256 Len=0 TSval=209
225	13.822125015	10.3.3.3	10.1.1.1	TCP	66	48036 → 21 [FIN, ACK] Seq=81 Ack=285 Win=64256 Len=0 TSv
226	13.822303831	10.1.1.1	10.3.3.3	TCP	66	21 → 48036 [ACK] Seq=285 Ack=82 Win=65280 Len=0 TSval=118
227	14.250185002	10.1.1.254	224.0.0.5	OSPF	78	Hello Packet

▶ Frame 1: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface veth1.2.ac, id 0  
 ▶ Ethernet II, Src: 00:00:00\_aa:00:10 (00:00:00:aa:00:10), Dst: IPv6mcast\_05 (33:33:00:00:00:05)  
 ▶ Internet Protocol Version 6, Src: fe80::200:ff:feaa:10, Dst: ff02::5  
 ▶ Open Shortest Path First

Figure 1: Wireshark - FTP Corvo

*veth1.2.ac					
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help					
Apply a display filter ... <Ctrl-/>					
No.	Time	Source	Destination	Protocol	Length Info
34266	17.458115510	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34267	17.458116183	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34268	17.458117128	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34269	17.458315213	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34270	17.458385032	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34271	17.458386423	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34272	17.458387114	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34273	17.458448906	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34274	17.458450227	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34275	17.458451014	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34276	17.458511122	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34277	17.458599890	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34278	17.458678502	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34279	17.458710720	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34280	17.458818775	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34281	17.458819930	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34282	17.458885206	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34283	17.458960271	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34284	17.458961333	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34285	17.459007236	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34286	17.459008240	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34287	17.459008784	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34288	17.459009328	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34289	17.4590096188	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34290	17.463079919	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34291	17.463084920	10.1.1.1	10.3.3.3	TFTP	366 Data Packet, Block: 278 (last)
34292	17.463124132	10.3.3.3	10.1.1.1	TFTP	46 Acknowledgement, Block: 278
34293	17.463124584	10.3.3.3	10.1.1.1	ICMP	394 Destination unreachable (Port unreachable)
34294	17.463124992	10.3.3.3	10.1.1.1	ICMP	394 Destination unreachable (Port unreachable)
34295	17.463125396	10.3.3.3	10.1.1.1	ICMP	394 Destination unreachable (Port unreachable)
34296	17.463125802	10.3.3.3	10.1.1.1	ICMP	394 Destination unreachable (Port unreachable)
34297	17.639534563	fe80::200:ff:feaa:10	ff02::5	OSPF	90 Hello Packet
34298	18.011028655	10.1.1.254	224.0.0.5	OSPF	78 Hello Packet
34299	20.011205942	10.1.1.254	224.0.0.5	OSPF	78 Hello Packet
34300	22.011365745	10.1.1.254	224.0.0.5	OSPF	78 Hello Packet
34301	24.012386164	10.1.1.254	224.0.0.5	OSPF	78 Hello Packet

Frame 1: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface veth1.2.ac, id 0  
 Ethernet II, Src: 00:00:00:aa:00:10 (00:00:00:aa:00:10), Dst: IPv4mcast\_05 (01:00:5e:00:00:05)  
 Internet Protocol Version 4, Src: 10.1.1.254, Dst: 224.0.0.5  
 Open Shortest Path First

Figure 2: Wireshark - TFTP Corvo

---

As duas figuras de cima representam o download do file2 a partir do computador "Corvo", usando FTP e TFTP. Analisando a Figura 1 podemos verificar que apesar de existirem segmentos ACK repetidos, o ficheiro acabava sempre por atingir o destino final. Já ao analisar a Figura 2 verificamos que o ficheiro acaba por não atingir o destino final, sendo que o host apenas se apercebia disso após algum tempo. Por outro lado, quando o envio de ficheiros é concluído com sucesso, usando o protocolo TFTP, verificamos que o tempo de transmissão é menor do que o de FTP. Concluimos assim que o protocolo a ser usado deve ser o TFTP, quando a velocidade é prioridade ou o tamanho do ficheiro é pequeno. Já o FTP deve ser utilizado, quando a prioridade é o ficheiro chegar ao destino (para grandes quantidades de dados), garantindo totalmente a transferência dos dados, apesar de ser mais demorado o processo.

## 5 Conclusão

Após a realização deste trabalho prático consideramos que aprofundamos o nosso conhecimento sobre protocolos da Camada de Transporte e Aplicação. Com o apoio das ferramentas Core e Wireshark realizamos alguns experimentos que permitiram a visualização de vários procedimentos e processos de transferência de dados, em diferentes protocolos. Em suma, concluimos que a nível da camada de transporte escolhemos o protocolo em função da importância dada à perda de dados. No caso de não querermos privilegiar o envio total de dados, optamos pelo protocolo UDP e um protocolo de aplicação que o use, aproveitando a velocidade máxima de transmissão possível. Se quisermos que não haja perda de informação utilizamos o protocolo TCP na camada de transporte e uma aplicação que o use, perdendo assim na eficiência mas garantido que todos os dados enviados são recebidos do lado do recetor. Posto isto o grupo considera que conseguiu realizar o trabalho e explorar as componentes propostas a estudo neste trabalho prático.

## 6 Anexo

### 6.1 Ping

No.	Time	Source	Destination	Proto	Len	Info
87	122.458067390	00:00:00 aa:00:10	00:00:00	ARP	42	Who has 10.1.1.1? Tell 10.1.1.254
88	122.459502324	00:00:00 aa:00:14	00:00:00	ARP	42	Who has 10.1.1.254? Tell 10.1.1.1
89	122.459510609	00:00:00 aa:00:14	00:00:00	ARP	42	10.1.1.1 is at 00:00:00:aa:00:14
90	122.459528489	00:00:00 aa:00:10	00:00:00	ARP	42	10.1.1.254 is at 00:00:00:aa:00:10
91	123.234707380	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request id=0x0030, seq=7/1792, ttl=61 (reply in ...)
92	123.239960260	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply id=0x0030, seq=7/1792, ttl=64 (request i...
93	124.059180195	10.1.1.1.254	224.0.0.5	OSPF	78	Hello Packet
94	124.233955535	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request id=0x0030, seq=8/2048, ttl=61 (reply in ...)
95	124.238753578	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply id=0x0030, seq=8/2048, ttl=64 (request i...
96	125.231703074	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request id=0x0030, seq=9/2304, ttl=61 (reply in ...)
97	125.236392217	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply id=0x0030, seq=9/2304, ttl=64 (request i...
98	126.059338209	10.1.1.1.254	224.0.0.5	OSPF	78	Hello Packet
99	126.238050838	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request
100	126.238291467	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply
101	127.239309943	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request
102	127.239707413	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply
103	128.060799498	10.1.1.1.254	224.0.0.5	OSPF	78	Hello Packet
104	128.283381876	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request
105	128.293059240	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply
106	128.944917500	fe80::200:ff:feaa:10	ff02::5	OSPF	90	Hello Packet
107	129.274278016	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request
108	129.274528379	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply
109	130.060787208	10.1.1.1.254	224.0.0.5	OSPF	78	Hello Packet
110	130.274929878	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request
111	130.275111311	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply
112	131.279966695	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request
113	131.280208987	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply
114	132.061266184	10.1.1.1.254	224.0.0.5	OSPF	78	Hello Packet
115	132.304409527	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request
116	132.304565007	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply
117	133.327987410	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request
118	133.328125474	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply
119	134.061339816	10.1.1.1.254	224.0.0.5	OSPF	78	Hello Packet
120	134.356918052	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request
121	134.357652364	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply
122	135.358685623	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request
123	135.358848968	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply
124	136.061737690	10.1.1.1.254	224.0.0.5	OSPF	78	Hello Packet
125	136.369210874	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request

Frame 1: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface veth1.2.1f, id 0  
Ethernet II, Src: 00:00:00:aa:00:10 (00:00:00:aa:00:10), Dst: IPv4mcast\_05 (01:00:5e:00:00:05)  
Internet Protocol Version 4, Src: 10.1.1.254, Dst: 224.0.0.5  
Open Shortest Path First

```
vcmd
(761/Laptop1.conf# ping -c 20 10.1.1.1 | tee file-ping-output
PING 10.1.1.1 (10.1.1.1) 56(84) bytes of data.
64 bytes from 10.1.1.1: icmp_seq=1 ttl=61 time=0.973 ms
64 bytes from 10.1.1.1: icmp_seq=2 ttl=61 time=5.64 ms
64 bytes from 10.1.1.1: icmp_seq=3 ttl=61 time=5.95 ms
64 bytes from 10.1.1.1: icmp_seq=4 ttl=61 time=22.2 ms
64 bytes from 10.1.1.1: icmp_seq=5 ttl=61 time=3.28 ms
64 bytes from 10.1.1.1: icmp_seq=6 ttl=61 time=6.00 ms
64 bytes from 10.1.1.1: icmp_seq=7 ttl=61 time=13.3 ms
64 bytes from 10.1.1.1: icmp_seq=8 ttl=61 time=3.38 ms
64 bytes from 10.1.1.1: icmp_seq=9 ttl=61 time=4.88 ms
64 bytes from 10.1.1.1: icmp_seq=10 ttl=61 time=5.25 ms
64 bytes from 10.1.1.1: icmp_seq=11 ttl=61 time=0.672 ms
64 bytes from 10.1.1.1: icmp_seq=12 ttl=61 time=21.4 ms
64 bytes from 10.1.1.1: icmp_seq=13 ttl=61 time=0.982 ms
64 bytes from 10.1.1.1: icmp_seq=14 ttl=61 time=0.460 ms
64 bytes from 10.1.1.1: icmp_seq=15 ttl=61 time=0.493 ms
64 bytes from 10.1.1.1: icmp_seq=16 ttl=61 time=0.362 ms
64 bytes from 10.1.1.1: icmp_seq=17 ttl=61 time=0.310 ms
64 bytes from 10.1.1.1: icmp_seq=18 ttl=61 time=1.81 ms
64 bytes from 10.1.1.1: icmp_seq=19 ttl=61 time=0.383 ms
64 bytes from 10.1.1.1: icmp_seq=20 ttl=61 time=1.63 ms
--- 10.1.1.1 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19152ms
rtt min/avg/max/mdev = 0.310/5.293/22.244/6.486 ms
root@Laptop1:/tmp/pycore.46761/Laptop1.conf#
```

Figure 3: Ping Laptop1

No.	Time	Source	Destination	Proto	Len	Info
No.	Time	Source	Destination	Proto	Len	Info
87	122.458067390	00:00:00 aa:00:10	00:00:00	ARP	42	Who has 10.1.1.1? Tell 10.1.1.254
88	122.459502324	00:00:00 aa:00:14	00:00:00	ARP	42	Who has 10.1.1.254? Tell 10.1.1.1
89	122.459510609	00:00:00 aa:00:14	00:00:00	ARP	42	10.1.1.1 is at 00:00:00:aa:00:14
90	122.459528489	00:00:00 aa:00:10	00:00:00	ARP	42	10.1.1.254 is at 00:00:00:aa:00:10
91	123.234707380	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request id=0x0030, seq=7/1792, ttl=61 (reply in ...)
92	123.239960260	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply id=0x0030, seq=7/1792, ttl=64 (request i...
93	124.059180195	10.1.1.1.254	224.0.0.5	OSPF	78	Hello Packet
94	124.233955535	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request id=0x0030, seq=8/2048, ttl=61 (reply in ...)
95	124.238753578	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply id=0x0030, seq=8/2048, ttl=64 (request i...
96	125.231703074	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request id=0x0030, seq=9/2304, ttl=61 (reply in ...)
97	125.236392217	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply id=0x0030, seq=9/2304, ttl=64 (request i...
98	126.059338209	10.1.1.1.254	224.0.0.5	OSPF	78	Hello Packet
99	126.238050838	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request
100	126.238291467	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply
101	127.239309943	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request
102	127.239707413	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply
103	128.060799498	10.1.1.1.254	224.0.0.5	OSPF	78	Hello Packet
104	128.283381876	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request
105	128.293059240	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply
106	128.944917500	fe80::200:ff:feaa:10	ff02::5	OSPF	90	Hello Packet
107	129.274278016	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request
108	129.274528379	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply
109	130.060787208	10.1.1.1.254	224.0.0.5	OSPF	78	Hello Packet
110	130.274929878	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request
111	130.275111311	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply
112	131.279966695	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request
113	131.280208987	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply
114	132.061266184	10.1.1.1.254	224.0.0.5	OSPF	78	Hello Packet
115	132.304409527	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request
116	132.304565007	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply
117	133.327987410	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request
118	133.328125474	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply
119	134.061339816	10.1.1.1.254	224.0.0.5	OSPF	78	Hello Packet
120	134.356918052	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request
121	134.357652364	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply
122	135.358685623	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request
123	135.358848968	10.1.1.1	10.4.4.1	ICMP	98	Echo (ping) reply
124	136.061737690	10.1.1.1.254	224.0.0.5	OSPF	78	Hello Packet
125	136.369210874	10.4.4.1	10.1.1.1	ICMP	98	Echo (ping) request

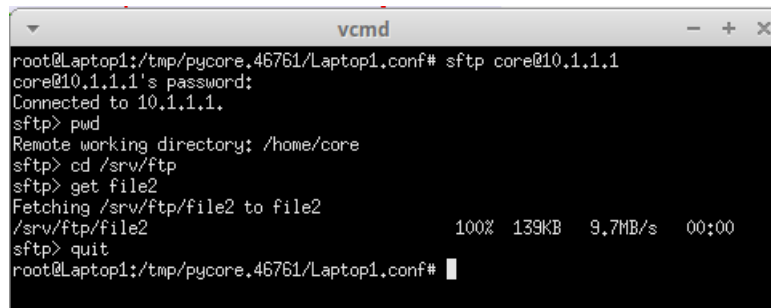
Ethernet II, Src: 00:00:00:aa:00:10 (00:00:00:aa:00:10), Dst: IPv4mcast\_05 (01:00:5e:00:00:05)  
Internet Protocol Version 4, Src: 10.1.1.254, Dst: 224.0.0.5  
Open Shortest Path First

```
vcmd
1/Corvo.conf# ping -c 20 10.1.1.1 | tee file-ping-output
PING 10.1.1.1 (10.1.1.1) 56(84) bytes of data.
64 bytes from 10.1.1.1: icmp_seq=1 ttl=61 time=7.66 ms
64 bytes from 10.1.1.1: icmp_seq=2 ttl=61 time=9.31 ms
64 bytes from 10.1.1.1: icmp_seq=3 ttl=61 time=11.7 ms
64 bytes from 10.1.1.1: icmp_seq=4 ttl=61 time=7.49 ms
64 bytes from 10.1.1.1: icmp_seq=5 ttl=61 time=6.19 ms
64 bytes from 10.1.1.1: icmp_seq=6 ttl=61 time=6.61 ms
64 bytes from 10.1.1.1: icmp_seq=7 ttl=61 time=12.1 ms
64 bytes from 10.1.1.1: icmp_seq=8 ttl=61 time=20.7 ms
64 bytes from 10.1.1.1: icmp_seq=9 ttl=61 time=13.9 ms
64 bytes from 10.1.1.1: icmp_seq=10 ttl=61 time=13.9 ms (DUP!)
64 bytes from 10.1.1.1: icmp_seq=11 ttl=61 time=9.82 ms
64 bytes from 10.1.1.1: icmp_seq=12 ttl=61 time=18.5 ms
64 bytes from 10.1.1.1: icmp_seq=13 ttl=61 time=24.6 ms
64 bytes from 10.1.1.1: icmp_seq=14 ttl=61 time=9.81 ms
64 bytes from 10.1.1.1: icmp_seq=15 ttl=61 time=9.67 ms
64 bytes from 10.1.1.1: icmp_seq=16 ttl=61 time=19.0 ms
64 bytes from 10.1.1.1: icmp_seq=17 ttl=61 time=5.42 ms
64 bytes from 10.1.1.1: icmp_seq=18 ttl=61 time=16.0 ms
64 bytes from 10.1.1.1: icmp_seq=19 ttl=61 time=9.78 ms
64 bytes from 10.1.1.1: icmp_seq=20 ttl=61 time=7.79 ms
--- 10.1.1.1 ping statistics ---
20 packets transmitted, 19 received, +1 duplicates, 5% packet loss, time 19072ms
rtt min/avg/max/mdev = 5.415/11.845/24.544/5.304 ms
root@Corvo:/tmp/pycore.46761/Corvo.conf#
```

Figure 4: Ping Corvo

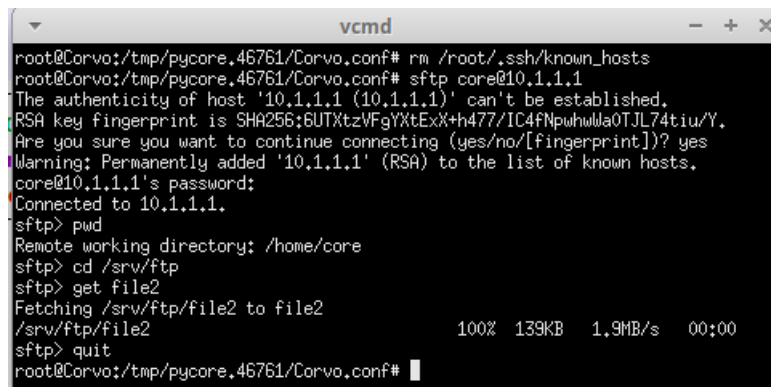
---

## 6.2 SFTP



```
vcmd
root@Laptop1:/tmp/pycore.46761/Laptop1.conf# sftp core@10.1.1.1
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> get file2
Fetching /srv/ftp/file2 to file2
/srv/ftp/file2                                100% 139KB  9.7MB/s   00:00
sftp> quit
root@Laptop1:/tmp/pycore.46761/Laptop1.conf#
```

Figure 5: SFTP Laptop1



```
vcmd
root@Corvo:/tmp/pycore.46761/Corvo.conf# rm /root/.ssh/known_hosts
root@Corvo:/tmp/pycore.46761/Corvo.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 SHA256:6UTXtzVFgYXtExX+h477/IC4fNpwhwMa0TJL74tiu/Y.
Are you sure you want to continue connecting (yes/no/[fingerprint])? 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> get file2
Fetching /srv/ftp/file2 to file2
/srv/ftp/file2                                100% 139KB  1.9MB/s   00:00
sftp> quit
root@Corvo:/tmp/pycore.46761/Corvo.conf#
```

Figure 6: SFTP Corvo



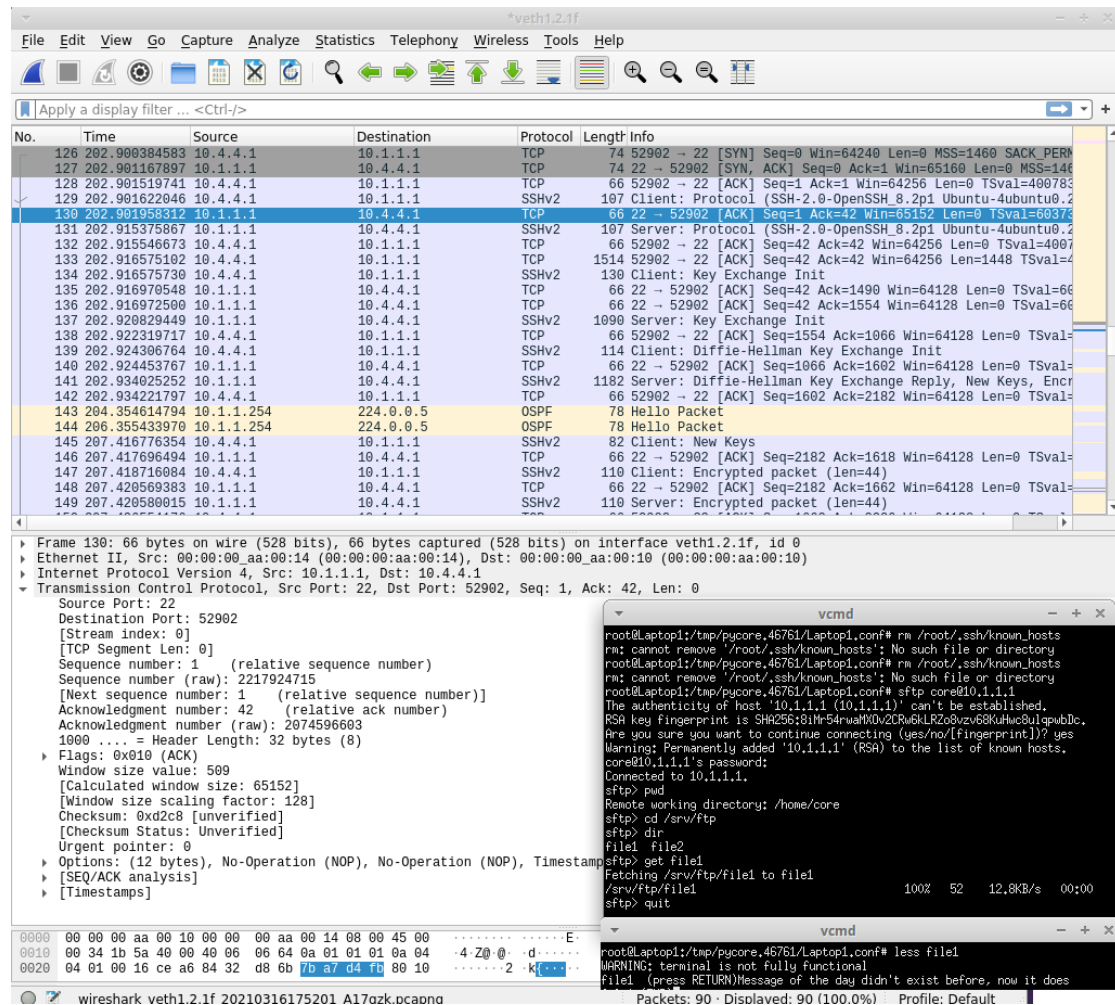
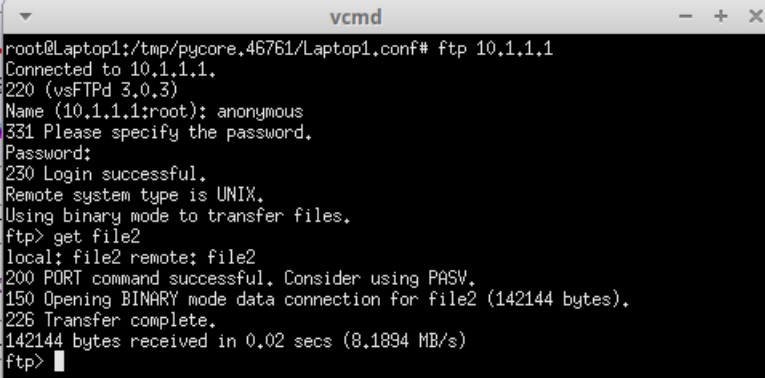


Figure 7: SFTP Wireshark

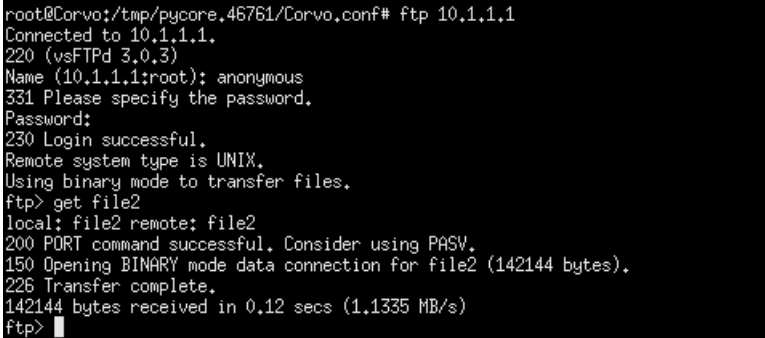
---

## 6.3 FTP



```
vcmd
root@Laptop1:/tmp/pycore.46761/Laptop1.conf# ftp 10.1.1.1
Connected to 10.1.1.1.
220 (vsFTPd 3.0.3)
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> get file2
local: file2 remote: file2
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for file2 (142144 bytes).
226 Transfer complete.
142144 bytes received in 0.02 secs (8.1894 MB/s)
ftp> █
```

Figure 8: FTP Laptop1



```
root@Corvo:/tmp/pycore.46761/Corvo.conf# ftp 10.1.1.1
Connected to 10.1.1.1.
220 (vsFTPd 3.0.3)
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> get file2
local: file2 remote: file2
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for file2 (142144 bytes).
226 Transfer complete.
142144 bytes received in 0.12 secs (1.1335 MB/s)
ftp> █
```

Figure 9: FTP Corvo

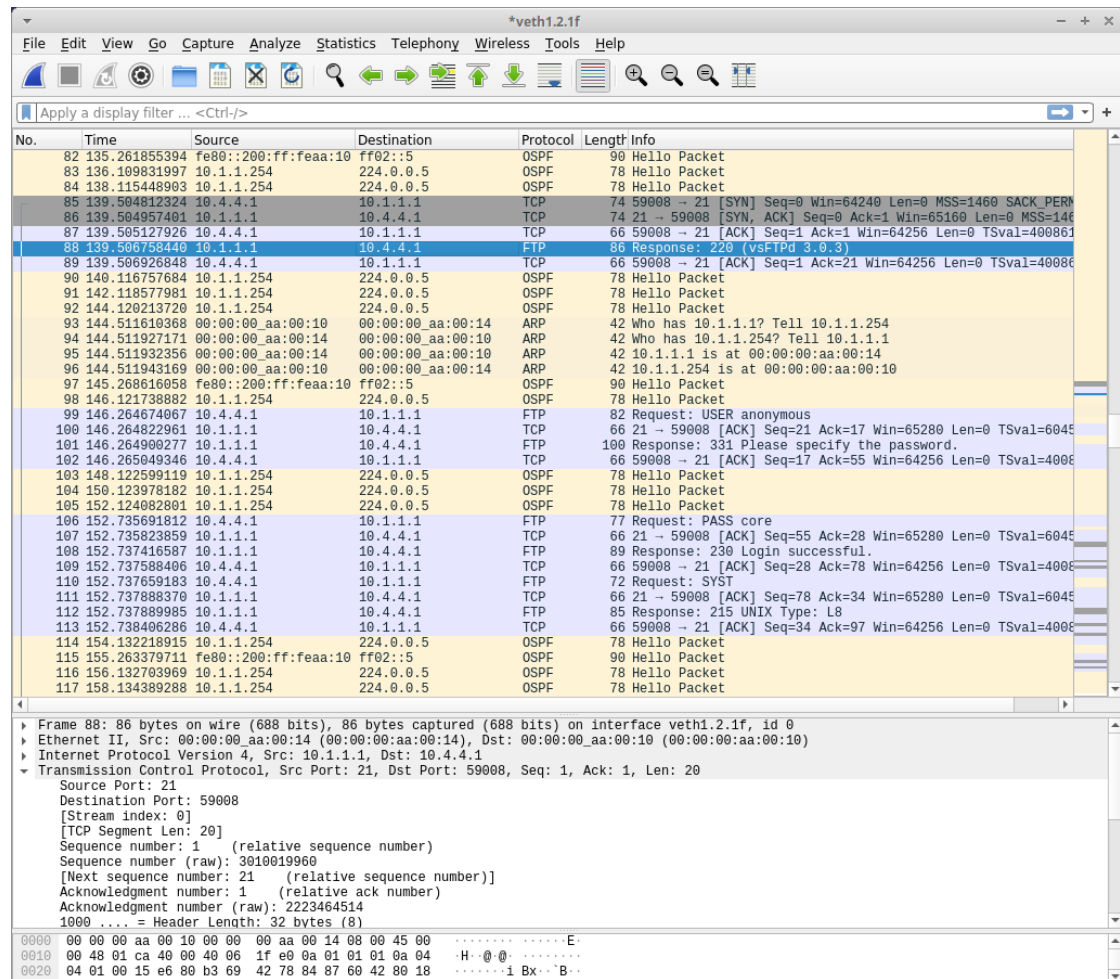
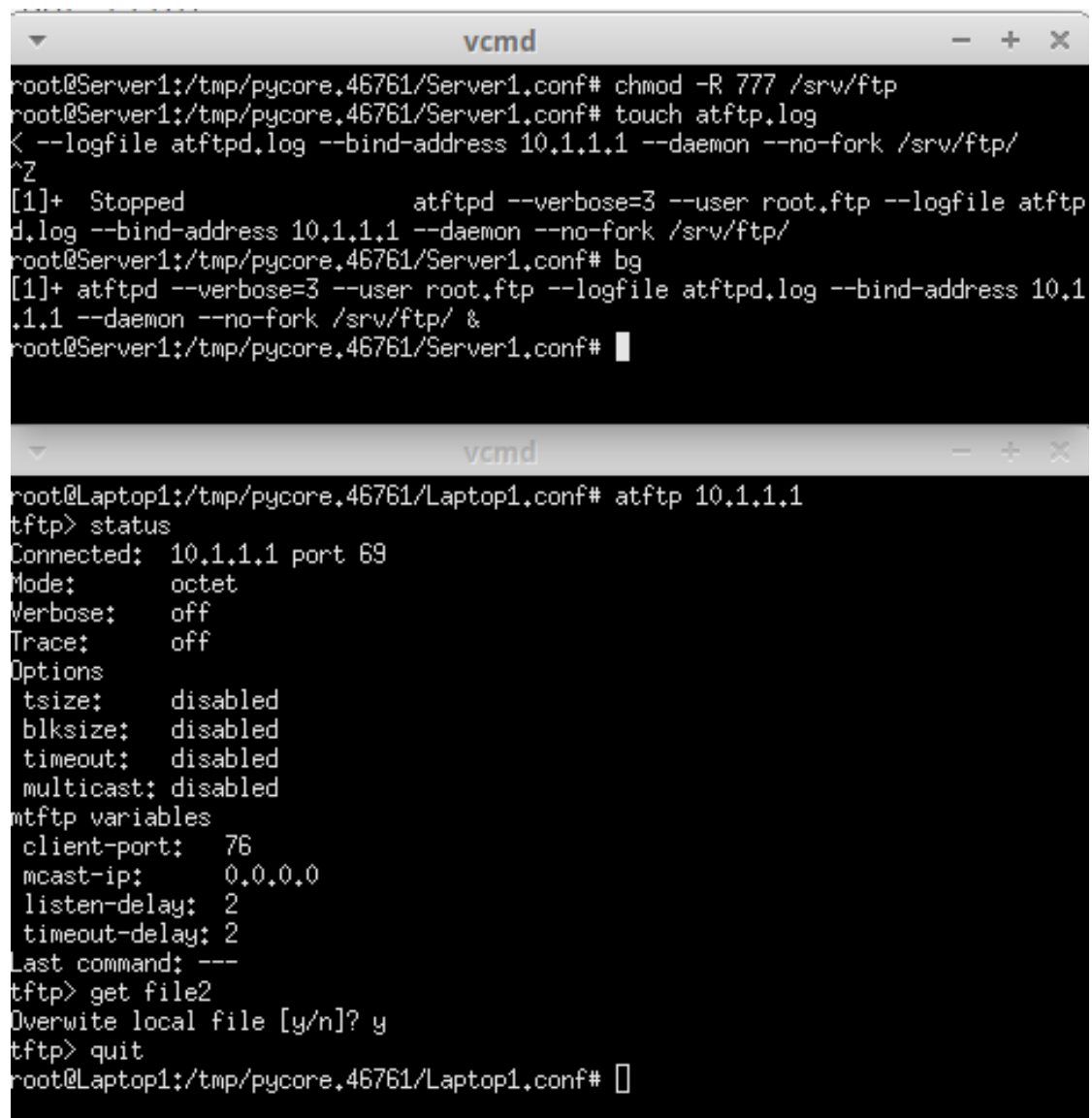


Figure 10: FTP Wireshark

## 6.4 TFTP

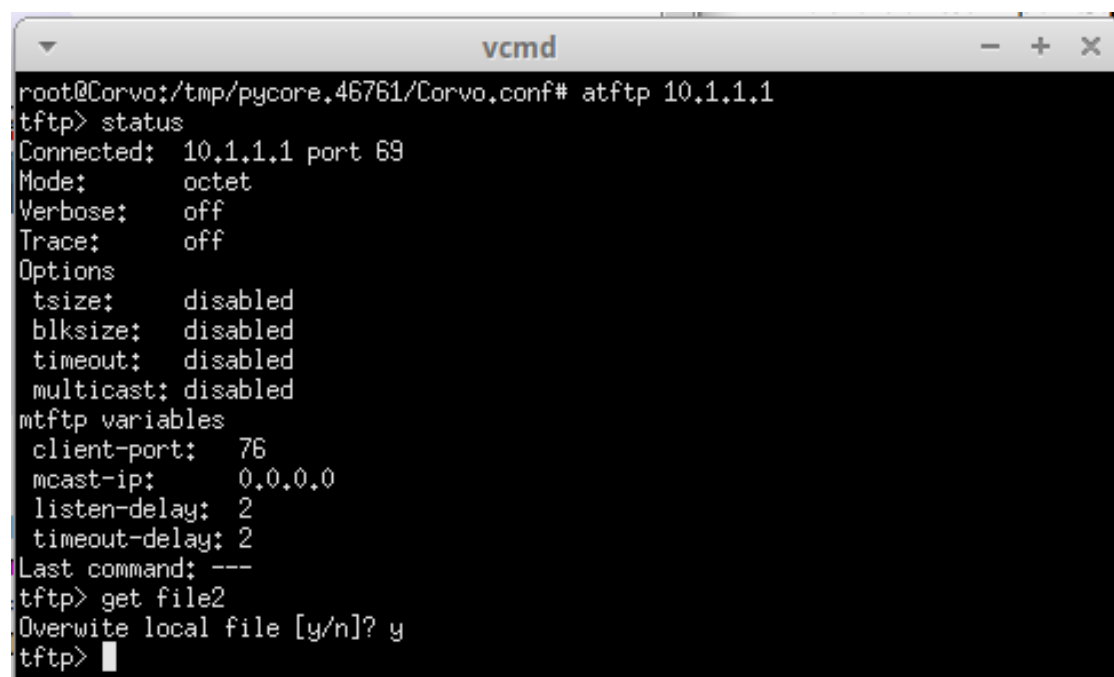


The figure consists of two terminal windows. The top window, titled 'vcmd', shows the configuration of a TFTP server on 'Server1'. The user sets permissions for the directory /srv/ftp, creates a log file, and starts the atftpd daemon in the background. The bottom window, also titled 'vcmd', shows the client-side interaction on 'Laptop1'. The user connects to the server at 10.1.1.1, checks the status, and successfully retrieves a file named 'file2'.

```
root@Server1:/tmp/pycore.46761/Server1.conf# chmod -R 777 /srv/ftp
root@Server1:/tmp/pycore.46761/Server1.conf# touch atftp.log
< --logfile atftpd.log --bind-address 10.1.1.1 --daemon --no-fork /srv/ftp/
^Z
[1]+  Stopped                  atftpd --verbose=3 --user root.ftp --logfile atftp
d.log --bind-address 10.1.1.1 --daemon --no-fork /srv/ftp/
root@Server1:/tmp/pycore.46761/Server1.conf# bg
[1]+  atftpd --verbose=3 --user root.ftp --logfile atftpd.log --bind-address 10.1
.1.1 --daemon --no-fork /srv/ftp/ &
root@Server1:/tmp/pycore.46761/Server1.conf# █

root@Laptop1:/tmp/pycore.46761/Laptop1.conf# atftp 10.1.1.1
tftp> status
Connected: 10.1.1.1 port 69
Mode:      octet
Verbose:   off
Trace:     off
Options
  tsize:    disabled
  blksize:  disabled
  timeout:  disabled
  multicast: disabled
mtftp variables
  client-port: 76
  mcast-ip:    0.0.0.0
  listen-delay: 2
  timeout-delay: 2
Last command: ---
tftp> get file2
Overwrite local file [y/n]? y
tftp> quit
root@Laptop1:/tmp/pycore.46761/Laptop1.conf# █
```

Figure 11: TFTP Laptop1



```
root@Corvo:/tmp/pycore.46761/Corvo.conf# atftp 10.1.1.1
tftp> status
Connected: 10.1.1.1 port 69
Mode:      octet
Verbose:   off
Trace:     off
Options
  tsize:    disabled
  blksize:  disabled
  timeout:  disabled
  multicast: disabled
mtftp variables
  client-port: 76
  mcast-ip:    0.0.0.0
  listen-delay: 2
  timeout-delay: 2
Last command: ---
tftp> get file2
Overwrite local file [y/n]? y
tftp> 
```

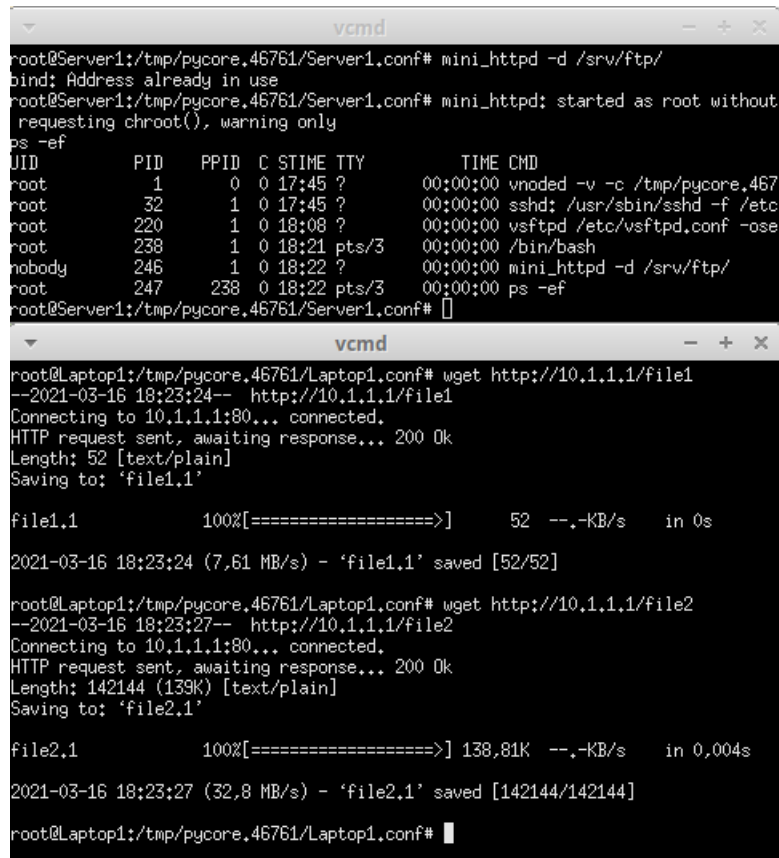
Figure 12: TFTP Corvo

*veth1.2.1f						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
Apply a display filter ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
10	14.011949708	10.1.1.254	224.0.0.5	OSPF	78	Hello Packet
11	16.012857144	10.1.1.254	224.0.0.5	OSPF	78	Hello Packet
12	18.013842616	10.1.1.254	224.0.0.5	OSPF	78	Hello Packet
13	20.014688103	10.1.1.254	224.0.0.5	OSPF	78	Hello Packet
14	20.487183216	fe80::200:ff:feaa:10	ff02::5	OSPF	90	Hello Packet
15	22.015208734	10.1.1.254	224.0.0.5	OSPF	78	Hello Packet
16	24.017014623	10.1.1.254	224.0.0.5	OSPF	78	Hello Packet
17	26.017275309	10.1.1.254	224.0.0.5	OSPF	78	Hello Packet
18	28.018781616	10.1.1.254	224.0.0.5	OSPF	78	Hello Packet
19	30.019251406	10.1.1.254	224.0.0.5	OSPF	78	Hello Packet
20	30.446995305	fe80::200:ff:feaa:10	ff02::5	OSPF	90	Hello Packet
21	31.908327200	10.4.4.1	10.1.1.1	TFTP	56	Read Request, File: file2, Transfer type: octet
22	31.909034427	10.1.1.1	10.4.4.1	TFTP	558	Data Packet, Block: 1
23	31.909262994	10.4.4.1	10.1.1.1	TFTP	46	Acknowledgement, Block: 1
24	31.909424956	10.1.1.1	10.4.4.1	TFTP	558	Data Packet, Block: 2
25	31.909765981	10.4.4.1	10.1.1.1	TFTP	46	Acknowledgement, Block: 2
26	31.910066749	10.1.1.1	10.4.4.1	TFTP	558	Data Packet, Block: 3
27	31.911069137	10.4.4.1	10.1.1.1	TFTP	46	Acknowledgement, Block: 3
28	31.912308287	10.1.1.1	10.4.4.1	TFTP	558	Data Packet, Block: 4
29	31.914347228	10.4.4.1	10.1.1.1	TFTP	46	Acknowledgement, Block: 4
30	31.915983126	10.1.1.1	10.4.4.1	TFTP	558	Data Packet, Block: 5
31	31.916520470	10.4.4.1	10.1.1.1	TFTP	46	Acknowledgement, Block: 5
32	31.918104834	10.1.1.1	10.4.4.1	TFTP	558	Data Packet, Block: 6
33	31.920140469	10.4.4.1	10.1.1.1	TFTP	46	Acknowledgement, Block: 6
34	31.921084655	10.1.1.1	10.4.4.1	TFTP	558	Data Packet, Block: 7
35	31.921369342	10.4.4.1	10.1.1.1	TFTP	46	Acknowledgement, Block: 7
36	31.921777347	10.1.1.1	10.4.4.1	TFTP	558	Data Packet, Block: 8
37	31.923087485	10.4.4.1	10.1.1.1	TFTP	46	Acknowledgement, Block: 8
38	31.926054016	10.1.1.1	10.4.4.1	TFTP	558	Data Packet, Block: 9
39	31.928956730	10.4.4.1	10.1.1.1	TFTP	46	Acknowledgement, Block: 9
40	31.929333378	10.1.1.1	10.4.4.1	TFTP	558	Data Packet, Block: 10
41	31.930917875	10.4.4.1	10.1.1.1	TFTP	46	Acknowledgement, Block: 10
42	31.933521999	10.1.1.1	10.4.4.1	TFTP	558	Data Packet, Block: 11
43	31.936732874	10.4.4.1	10.1.1.1	TFTP	46	Acknowledgement, Block: 11
44	31.938547787	10.1.1.1	10.4.4.1	TFTP	558	Data Packet, Block: 12
45	31.942460815	10.4.4.1	10.1.1.1	TFTP	46	Acknowledgement, Block: 12
▶ Frame 31: 46 bytes on wire (368 bits), 46 bytes captured (368 bits) on interface veth1.2.1f, id 0 ▶ Ethernet II, Src: 00:00:00:aa:00:10 (00:00:00:aa:00:10), Dst: 00:00:00:aa:00:14 (00:00:00:aa:00:14) ▶ Internet Protocol Version 4, Src: 10.4.4.1, Dst: 10.1.1.1 ▶ User Datagram Protocol, Src Port: 42663, Dst Port: 51962 Source Port: 42663 Destination Port: 51962 Length: 12 Checksum: 0x7524 [unverified] [Checksum Status: Unverified] [Stream index: 1] ▶ [Timestamps] ▶ Trivial File Transfer Protocol						
0000	00 00 00 aa 00 14 00 00	00 aa 00 10 08 00 45 00	.....E..			
0010	00 20 a3 7b 40 00 3d 11	81 4b 0a 04 04 01 0a 01	..{0 = .K.....			
0020	01 01 a6 a7 ca ra 00 0c	75 24 00 04 00 05	.....US....			

Figure 13: TFTP Wireshark

---

## 6.5 HTTP



The image shows two terminal windows. The top window, titled 'vcmd', is on a host named 'Server1'. It shows the command 'mini\_httpd -d /srv/ftp/' being executed, which starts the mini\_httpd service as root. Below this, a 'ps -ef' command is run, showing a list of processes including 'vnoded', 'sshd', 'vsftpd', and 'mini\_httpd'. The bottom window, also titled 'vcmd', is on a host named 'Laptop1'. It shows two 'wget' commands being used to download files from 'http://10.1.1.1/file1' and 'http://10.1.1.1/file2'. The output of these commands shows the files being downloaded and saved to 'file1.1' and 'file2.1' respectively, with progress bars and file sizes displayed.

```
vcmd
root@Server1:/tmp/pycore.46761/Server1.conf# mini_httpd -d /srv/ftp/
bind: Address already in use
root@Server1:/tmp/pycore.46761/Server1.conf# mini_httpd: started as root without
requesting chroot(), warning only
ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1        0  0 17:45 ?            00:00:00 vnoded -v -c /tmp/pycore.467
root          32        1  0 17:45 ?            00:00:00 sshd: /usr/sbin/sshd -f /etc
root         220        1  0 18:08 ?            00:00:00 vsftpd /etc/vsftpd.conf -ose
root         238        1  0 18:21 pts/3      00:00:00 /bin/bash
nobody       246        1  0 18:22 ?            00:00:00 mini_httpd -d /srv/ftp/
root         247       238  0 18:22 pts/3      00:00:00 ps -ef
root@Server1:/tmp/pycore.46761/Server1.conf#

vcmd
root@Laptop1:/tmp/pycore.46761/Laptop1.conf# wget http://10.1.1.1/file1
--2021-03-16 18:23:24-- http://10.1.1.1/file1
Connecting to 10.1.1.1:80... connected.
HTTP request sent, awaiting response... 200 Ok
Length: 52 [text/plain]
Saving to: 'file1.1'

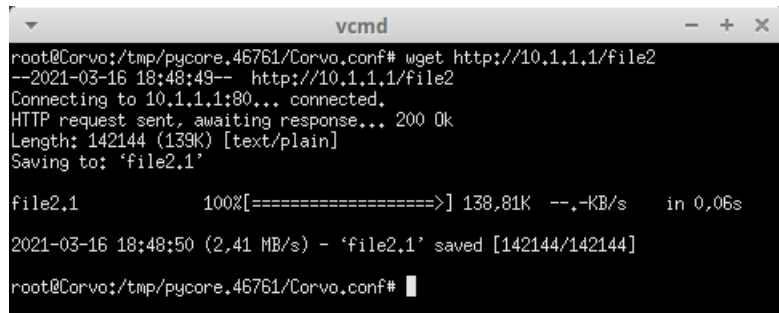
file1.1          100%[=====>]      52  --.-KB/s   in 0s
2021-03-16 18:23:24 (7.61 MB/s) - 'file1.1' saved [52/52]

root@Laptop1:/tmp/pycore.46761/Laptop1.conf# wget http://10.1.1.1/file2
--2021-03-16 18:23:27-- http://10.1.1.1/file2
Connecting to 10.1.1.1:80... connected.
HTTP request sent, awaiting response... 200 Ok
Length: 142144 (139K) [text/plain]
Saving to: 'file2.1'

file2.1          100%[=====>] 138,81K  --.-KB/s   in 0,004s
2021-03-16 18:23:27 (32,8 MB/s) - 'file2.1' saved [142144/142144]

root@Laptop1:/tmp/pycore.46761/Laptop1.conf#
```

Figure 14: HTTP Laptop1



The image shows a terminal window titled 'vcmd' on a host named 'Corvo'. It shows a 'wget' command being used to download a file from 'http://10.1.1.1/file2'. The output of the command shows the file being downloaded and saved to 'file2.1', with progress bars and file sizes displayed.

```
vcmd
root@Corvo:/tmp/pycore.46761/Corvo.conf# wget http://10.1.1.1/file2
--2021-03-16 18:48:49-- http://10.1.1.1/file2
Connecting to 10.1.1.1:80... connected.
HTTP request sent, awaiting response... 200 Ok
Length: 142144 (139K) [text/plain]
Saving to: 'file2.1'

file2.1          100%[=====>] 138,81K  --.-KB/s   in 0,06s
2021-03-16 18:48:50 (2,41 MB/s) - 'file2.1' saved [142144/142144]

root@Corvo:/tmp/pycore.46761/Corvo.conf#
```

Figure 15: HTTP Corvo

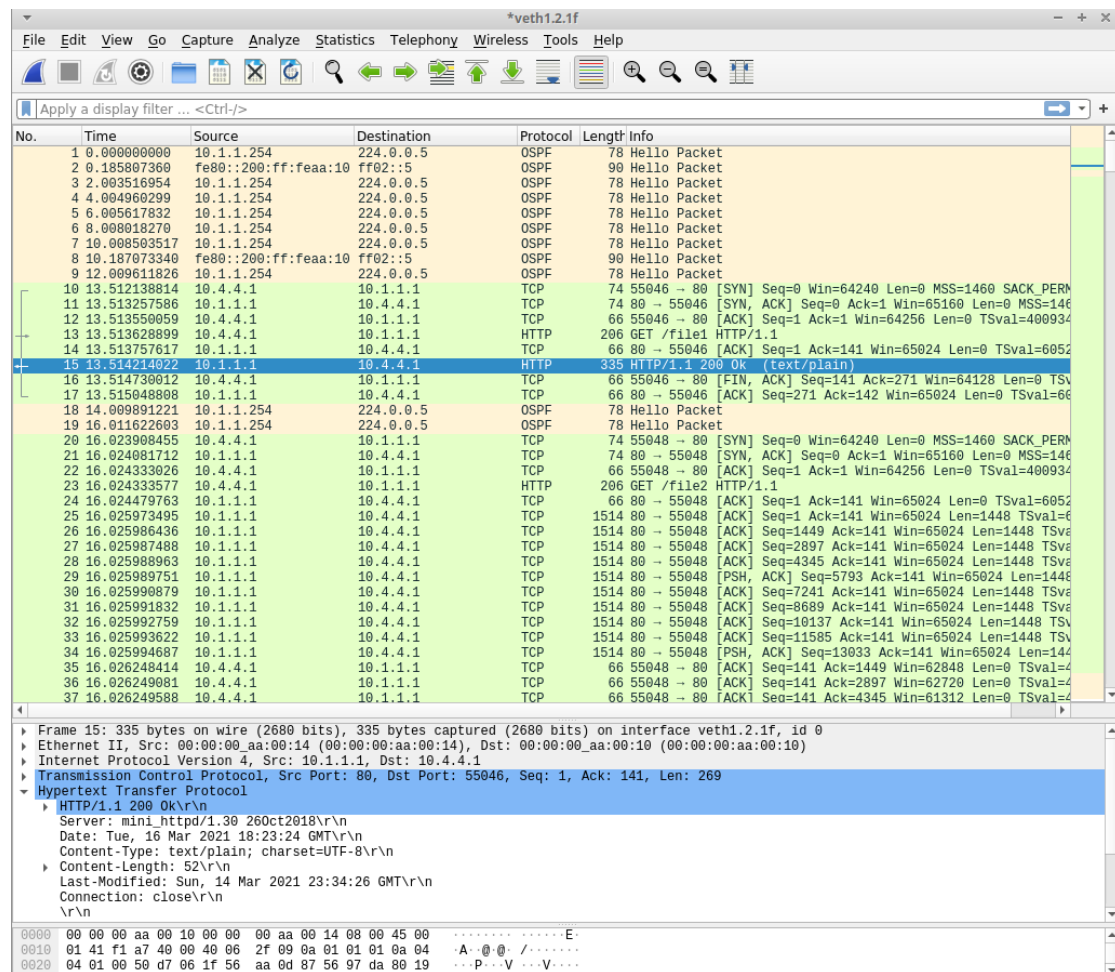


Figure 16: HTTP Wireshark