

Universidade do Minho

Comunicações por Computador

Trabalho Prático 1

Grupo 43

José João Cardoso Gonçalves a93204
Bernardo Emanuel Magalhães Saraiva a93189
Daniel Torres Azevedo a93324

28 de outubro de 2021

Questões e Respostas

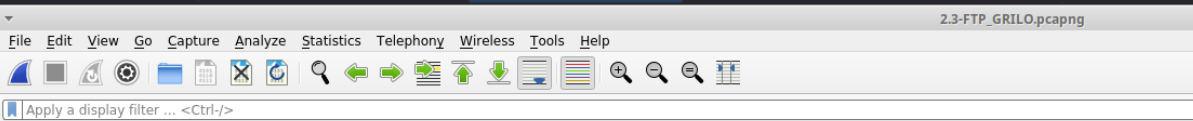
Para a execução do Trabalho Prático 1, foi-nos apresentado um guião contendo algumas perguntas, sendo que procedemos à realização do mesmo e respondemos às perguntas em seguida.

Parte I.

1. De que forma as perdas e duplicações de pacotes afetaram o desempenho das aplicações? Que camada lidou com as perdas e duplicações: transporte ou aplicação? Responda com base nas experiências feitas e nos resultados observados.

Resposta:

No caso de protocolos TCP (**SFTP,FTP,HTTP**), sempre que um pacote é perdido (ou sofre um atraso), obriga a que seja enviado um novo para o seu lugar (característica de protocolos orientados à conexão). Deste modo, de cada vez que isto acontece, ocorrem atrasos na chegada e processamento dos dados, fazendo com que haja um overhead associado e o desempenho sofra. Assim, a camada responsável pelas perdas e duplicações é a camada de transporte.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.2.2.254	224.0.0.5	OSPF	78	Hello Packet
2	1.741914211	fe80::200:ff:feaa:10	ff02::5	OSPF	90	Hello Packet
3	2.001262775	10.2.2.254	224.0.0.5	OSPF	78	Hello Packet
4	3.422254118	10.4.4.1	10.2.2.1	FTP	74	Request: TYPE I
5	3.422434812	10.2.2.1	10.4.4.1	TCP	66	21 → 59238 [ACK] Seq=1 Ack=9 Win=510 Len=0 TSval=1122125557 T...
6	3.422508404	10.2.2.1	10.4.4.1	FTP	97	Response: 200 Switching to Binary mode.
7	3.428079018	10.4.4.1	10.2.2.1	FTP	89	Request: PORT 10,4,4,1,234,105
8	3.428272379	10.2.2.1	10.4.4.1	TCP	66	21 → 59238 [ACK] Seq=32 Ack=32 Win=510 Len=0 TSval=1122125562...
9	3.428452608	10.2.2.1	10.4.4.1	FTP	117	Response: 200 PORT command successful. Consider using PASV.
10	3.433584214	10.4.4.1	10.2.2.1	FTP	78	Request: RETR file1
11	3.433585273	10.4.4.1	10.2.2.1	TCP	78	[TCP Retransmission] 59238 → 21 [PSH, ACK] Seq=32 Ack=83 Win=...
12	3.433834229	10.2.2.1	10.4.4.1	TCP	78	21 → 59238 [ACK] Seq=83 Ack=44 Win=510 Len=0 TSval=1122125568...
13	3.433952583	10.2.2.1	10.4.4.1	TCP	74	20 → 60009 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 T...
14	3.439222602	10.4.4.1	10.2.2.1	TCP	74	60009 → 20 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SA...
15	3.439393168	10.2.2.1	10.4.4.1	TCP	66	20 → 60009 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1122125574...
16	3.439553498	10.2.2.1	10.4.4.1	FTP	130	Response: 150 Opening BINARY mode data connection for file1 (...)
17	3.440120141	10.2.2.1	10.4.4.1	FTP-DA...	290	FTP Data: 224 bytes (PORT) (RETR file1)
18	3.440123157	10.2.2.1	10.4.4.1	TCP	66	20 → 60009 [FIN, ACK] Seq=225 Ack=1 Win=64256 Len=0 TSval=112...
19	3.446260269	10.4.4.1	10.2.2.1	TCP	66	60009 → 20 [ACK] Seq=1 Ack=225 Win=65024 Len=0 TSval=13893286...
20	3.446261379	10.4.4.1	10.2.2.1	TCP	66	60009 → 20 [FIN, ACK] Seq=1 Ack=226 Win=65024 Len=0 TSval=138...
21	3.446427578	10.2.2.1	10.4.4.1	TCP	66	20 → 60009 [ACK] Seq=226 Ack=2 Win=64256 Len=0 TSval=11221255...
22	3.446667930	10.2.2.1	10.4.4.1	FTP	90	Response: 226 Transfer complete.
23	3.451751254	10.4.4.1	10.2.2.1	TCP	66	59238 → 21 [ACK] Seq=44 Ack=171 Win=502 Len=0 TSval=138932861...
24	4.001890548	10.2.2.254	224.0.0.5	OSPF	78	Hello Packet
25	6.002369944	10.2.2.254	224.0.0.5	OSPF	78	Hello Packet
26	6.923615576	10.4.4.1	10.2.2.1	FTP	72	Request: QUIT
27	6.924165477	10.2.2.1	10.4.4.1	TCP	66	21 → 59238 [ACK] Seq=171 Ack=50 Win=510 Len=0 TSval=112212905...
28	6.924169147	10.2.2.1	10.4.4.1	FTP	80	Response: 221 Goodbye.
29	6.924170387	10.2.2.1	10.4.4.1	TCP	66	21 → 59238 [FIN, ACK] Seq=185 Ack=50 Win=510 Len=0 TSval=1122...
30	6.930015122	10.4.4.1	10.2.2.1	TCP	66	59238 → 21 [FIN, ACK] Seq=50 Ack=186 Win=502 Len=0 TSval=1389...
31	6.930172234	10.2.2.1	10.4.4.1	TCP	66	21 → 59238 [ACK] Seq=186 Ack=51 Win=510 Len=0 TSval=112212906...
32	8.002893231	10.2.2.254	224.0.0.5	OSPF	78	Hello Packet
33	10.003474695	10.2.2.254	224.0.0.5	OSPF	78	Hello Packet

Anexo 1 - Screenshot do wireShark durante a transferência do file1 para o grilo usando FTP onde se verifica uma retransmissão.

Já os protocolos UDP (**TFTP**) não fazem essa confirmação da recepção do pacote, sendo que há a possibilidade de não fazermos chegar ao destino a totalidade da informação pretendida. Obrigando por este motivo a que as perdas e duplicações de pacotes sejam tratadas pela camada da aplicação, afetando assim negativamente o desempenho. Deste modo, dizemos que o protocolo UDP não é orientado à conexão e não é confiável.

Capturing from veth1.2.57

No.	Time	Source	Destination	Protocol	Length	Info
5	4.675543505	10.4.4.1	10.2.2.1	TFTP	56	Read Request, File: file2, Transfer type: octet
6	4.676232697	10.2.2.1	10.4.4.1	TFTP	558	Data Packet, Block: 1
7	4.681586166	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 1
8	4.681895959	10.2.2.1	10.4.4.1	TFTP	558	Data Packet, Block: 2
9	4.687354955	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 2
10	4.687621655	10.2.2.1	10.4.4.1	TFTP	558	Data Packet, Block: 3
11	4.693190500	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 3
12	4.693524273	10.2.2.1	10.4.4.1	TFTP	558	Data Packet, Block: 4
13	4.699630053	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 4
14	4.699941242	10.2.2.1	10.4.4.1	TFTP	558	Data Packet, Block: 5
15	4.705284825	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 5
16	4.705536079	10.2.2.1	10.4.4.1	TFTP	558	Data Packet, Block: 6
17	4.710900839	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 6
18	4.711136620	10.2.2.1	10.4.4.1	TFTP	558	Data Packet, Block: 7
19	4.716408902	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 7
20	4.716668641	10.2.2.1	10.4.4.1	TFTP	558	Data Packet, Block: 8
21	4.722043928	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 8
22	4.722327997	10.2.2.1	10.4.4.1	TFTP	558	Data Packet, Block: 9
23	4.727627867	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 9
24	4.727834359	10.2.2.1	10.4.4.1	TFTP	558	Data Packet, Block: 10
25	4.733041125	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 10
26	4.733219635	10.2.2.1	10.4.4.1	TFTP	558	Data Packet, Block: 11
27	4.738508184	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 11
28	4.738681205	10.2.2.1	10.4.4.1	TFTP	558	Data Packet, Block: 12
29	4.743953662	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 12
30	4.744177666	10.2.2.1	10.4.4.1	TFTP	558	Data Packet, Block: 13
31	4.749415836	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 13
32	4.749619170	10.2.2.1	10.4.4.1	TFTP	558	Data Packet, Block: 14
33	4.754843889	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 14
34	4.755923842	10.2.2.1	10.4.4.1	TFTP	558	Data Packet, Block: 15
35	4.760237068	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 15
36	4.760415125	10.2.2.1	10.4.4.1	TFTP	558	Data Packet, Block: 16
37	4.765604908	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 16
38	4.765805591	10.2.2.1	10.4.4.1	TFTP	558	Data Packet, Block: 17
39	4.771144268	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 17
40	4.771420970	10.2.2.1	10.4.4.1	TFTP	558	Data Packet, Block: 18
41	4.776806164	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 18
42	4.776807573	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 18
43	4.777058890	10.2.2.1	10.4.4.1	TFTP	558	Data Packet, Block: 19
44	4.777284599	10.2.2.1	10.4.4.1	TFTP	558	Data Packet, Block: 19
45	4.782427567	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 19
46	4.782429511	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 19
47	4.782431274	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 19
48	4.782773893	10.2.2.1	10.4.4.1	TFTP	558	Data Packet, Block: 20
49	4.782780273	10.2.2.1	10.4.4.1	TFTP	558	Data Packet, Block: 20
50	4.782782210	10.2.2.1	10.4.4.1	TFTP	558	Data Packet, Block: 20
51	4.788051098	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 20
52	4.788325336	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 20
53	4.788327228	10.4.4.1	10.2.2.1	TFTP	46	Acknowledgement, Block: 20

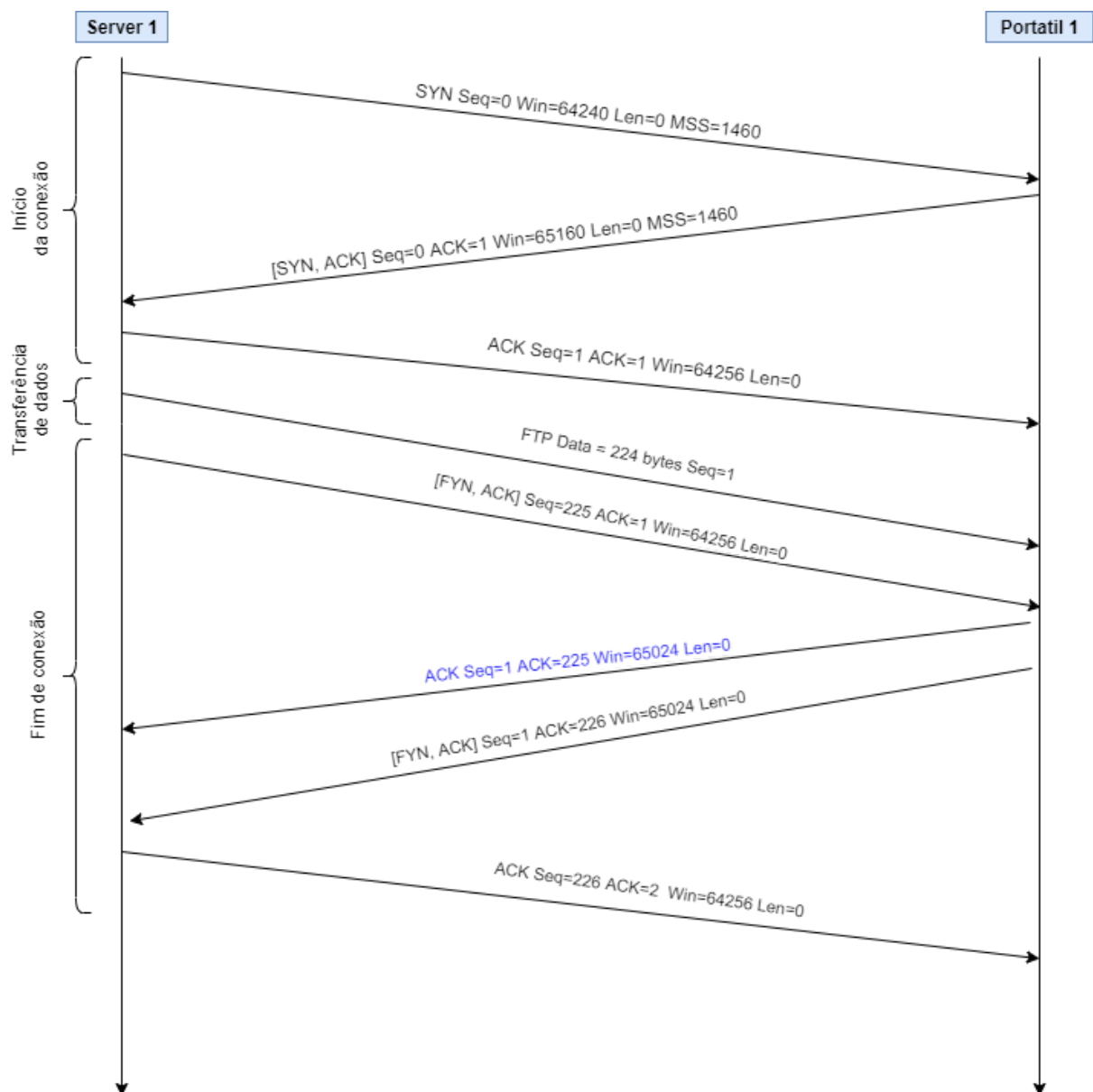
Frame 52: 46 bytes on wire (368 bits), 46 bytes captured (368 bits) on interface veth1.2.57, id 0
 0000 00 00 00 aa 00 14 00 00 00 aa 00 10 08 00 45 00E
 veth1.2.57: <live capture in progress> Packets: 27567 · Displayed: 27567 (100.0%) Profile: Default

Anexo 2- Screenshot do wireShark durante a transferência do file2 para o grilo usando TFTP.

2. Obtenha a partir do wireshark, ou desenhe manualmente, um diagrama temporal para a transferência de file1 por FTP. Foque-se apenas na transferência de dados [ftp-data] e não na conexão de controlo, pois o FTP usa mais que uma conexão em simultâneo. Identifique, se aplicável, as fases de início de conexão, transferência de dados e fim de conexão. Identifique também os tipos de segmentos trocados e os números de sequência usados quer nos dados como nas confirmações.

Resposta:

FTP é um serviço que utiliza duas portas: a 20, que lida com os dados, e a 21, que lida com o estabelecimento da conexão (porta de comando). Por este motivo, e porque apenas desejamos analisar os pacotes referentes aos dados, construímos o diagrama recorrendo ao filtro ***tcp.port==20*** no WireShark.

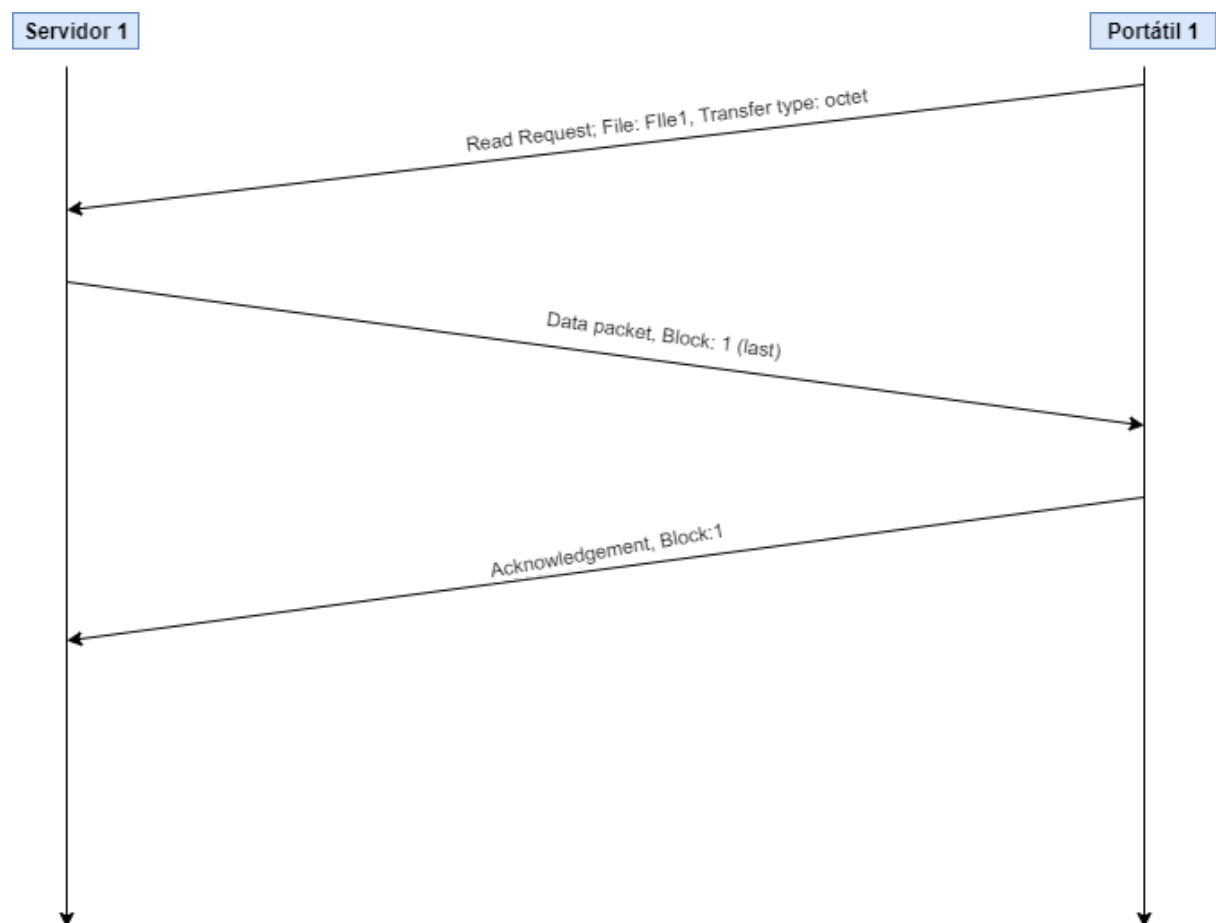


Anexo 3 - Diagrama FTP com as correspondentes fases de início de conexão, transferência de dados e fim de conexão.

Nota: O pacote ACK identificado a azul, apesar de se encontrar na secção de fim de sessão, refere-se à transferência.

3. Obtenha a partir do wireshark, ou desenhe manualmente, um diagrama temporal para a transferência de file1 por TFTP. Identifique, se aplicável, as fases de início de conexão, transferência de dados e fim de conexão. Identifique também os tipos de segmentos trocados e os números de sequência usados quer nos dados como nas confirmações.

Resposta:



Anexo 4 - Diagrama FTP com as correspondentes fases de início de conexão, transferência de dados e fim de conexão.

4. Compare sucintamente as quatro aplicações de transferência de ficheiros que usou nos seguintes pontos (i) uso da camada de transporte; (ii) eficiência; (iii) complexidade; (iv) segurança;

Resposta:

i) Uso da camada de transporte:

O protocolo **TFTP** recorre ao UDP para o seu funcionamento, sendo que a camada de transporte é utilizada para levar a informação, independentemente de fazer essa transmissão com sucesso ou não. Neste exemplo, quem lida com as perdas de pacotes e retransmissões é a camada aplicacional.

Os protocolos **FTP**, **HTTP** e **SFTP** recorrem todos ao TCP, sendo que este se caracteriza pela sua fiabilidade, uma vez que procura garantir o sucesso da transmissão dos pacotes ao seu destino. Ao levar um pacote, caso haja perdas, ele efetua uma retransmissão para que seja entregue no seu destino.

ii) Eficiência:

Para compararmos todos os protocolos pedidos no campo da eficiência vamos analisar a velocidade de transferência utilizando as diferentes tecnologias. Usamos como origem o **servidor1** e como destino o grilo para a transferência do **file2**, capturando o tráfego de dados com o WireShark aberto no grilo.

SFTP

2	1.591	fe80::10ab:5cff:fe55...	ff02::fb	MDNS	107 Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR _ipp...
3	1.759	10.4.4.1	10.2.2.1	SSH	134 Client: Encrypted packet (len=68)
4	1.764	10.2.2.1	10.4.4.1	SSH	142 Server: Encrypted packet (len=76)
5	1.764	10.4.4.1	10.2.2.1	TCP	66 38920 → 22 [ACK] Seq=69 Ack=77 Win=501 Len=0 TSval=1925507702 TSec...
6	1.765	10.4.4.1	10.2.2.1	SSH	134 Client: Encrypted packet (len=68)

Anexo 5 - Início transferência SFTP

153	1.819	10.2.2.1	10.4.4.1	SSH	134 Server: Encrypted packet (len=68)
154	1.819	10.4.4.1	10.2.2.1	SSH	118 Client: Encrypted packet (len=52)
155	1.829	10.2.2.1	10.4.4.1	SSH	134 Server: Encrypted packet (len=68)
156	1.871	10.4.4.1	10.2.2.1	TCP	66 38920 → 22 [ACK] Seq=621 Ack=142945 Win=1768 Len=0 TSval=192550780...

Anexo 6 - Fim transferência SFTP

Tempo decorrido : 0.28 ms

FTP

4	3.962	10.4.4.1	10.2.2.1	FTP	74 Request: TYPE I
5	3.968	10.2.2.1	10.4.4.1	FTP	97 Response: 200 Switching to Binary mode.
6	3.968	10.4.4.1	10.2.2.1	TCP	66 49218 → 21 [ACK] Seq=9 Ack=32 Win=502 Len=0 TSval=1926246101 TSecr...
7	3.968	10.4.4.1	10.2.2.1	FTP	89 Request: PORT 10,4,4,1,210,117
8	3.974	10.2.2.1	10.4.4.1	FTP	117 Response: 200 PORT command successful. Consider using PASV.
9	3.974	10.4.4.1	10.2.2.1	TCP	66 49218 → 21 [ACK] Seq=32 Ack=83 Win=502 Len=0 TSval=1926246107 TSecr...

Anexo 7 - Início transferência FTP

185	4.009	10.4.4.1	10.2.2.1	TCP	66 53877 → 20 [FIN, ACK] Seq=1 Ack=142146 Win=249600 Len=0 TSval=1926...
186	4.015	10.2.2.1	10.4.4.1	TCP	66 20 → 53877 [ACK] Seq=142146 Ack=2 Win=64256 Len=0 TSval=379572631 ...
187	4.015	10.2.2.1	10.4.4.1	FTP	90 Response: 226 Transfer complete.
188	4.015	10.4.4.1	10.2.2.1	TCP	66 49218 → 21 [ACK] Seq=44 Ack=174 Win=502 Len=0 TSval=1926246148 TSe...

Anexo 8 - Fim transferência FTP

Tempo decorrido: 0.053 ms

TFTP

3	2.085	10.4.4.1	10.2.2.1	TFTP	56 Read Request, File: file2, Transfer type: octet
4	2.092	10.2.2.1	10.4.4.1	TFTP	558 Data Packet, Block: 1
5	2.092	10.4.4.1	10.2.2.1	TFTP	46 Acknowledgement, Block: 1
6	2.092	10.2.2.1	10.4.4.1	TFTP	558 Data Packet, Block: 0

Anexo 9 - Início transferência TFTP

28412	3.893	10.2.2.1	10.4.4.1	TFTP	366 Data Packet, Block: 278 (last)
28413	3.893	10.2.2.1	10.4.4.1	TFTP	366 Data Packet, Block: 278 (last)
28414	3.893	10.2.2.1	10.4.4.1	TFTP	366 Data Packet, Block: 278 (last)

Anexo 10 - Fim transferência TFTP

Tempo decorrido: 1.808 ms

HTTP

1	0.000	10.4.4.1	10.2.2.1	TCP	74 58424 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=...
2	0.005	10.2.2.1	10.4.4.1	TCP	74 80 → 58424 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PE...
3	0.005	10.4.4.1	10.2.2.1	TCP	66 58424 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1926900170 TSec...
4	0.006	10.4.4.1	10.2.2.1	HTTP	206 GET /file2 HTTP/1.1
5	0.012	10.2.2.1	10.4.4.1	TCP	66 80 → 58424 [ACK] Seq=1 Ack=141 Win=65024 Len=0 TSval=380226660 TSe...

Anexo 11 - Início transferência HTTP

157	0.034	10.4.4.1	10.2.2.1	TCP	66 58424 → 80 [ACK] Seq=141 Ack=140457 Win=150272 Len=0 TSval=1926900...
158	0.034	10.4.4.1	10.2.2.1	TCP	66 58424 → 80 [ACK] Seq=141 Ack=141905 Win=153088 Len=0 TSval=1926900...
159	0.037	10.4.4.1	10.2.2.1	TCP	66 58424 → 80 [FIN, ACK] Seq=141 Ack=142367 Win=156032 Len=0 TSval=19...
160	0.043	10.2.2.1	10.4.4.1	TCP	66 80 → 58424 [ACK] Seq=142367 Ack=142 Win=65024 Len=0 TSval=38022660...

Anexo 12 - Fim transferência HTTP

Tempo decorrido: 0.043ms

Conclusão: HTTP < FTP < SFTP < TFTP

Apesar de teoricamente os protocolos UDP serem mais rápidos que os protocolos TCP, obtivemos na prática um resultado diferente. Este resultado pode ser explicado pelas seguintes razões:

- TFTP é um protocolo muito simples e transmite apenas um bloco de cada vez.
- Como testamos as transferências para o grilo e a conexão entre este e o servidor possui losses e dups definidos, a diferença de tempo de transferência será exacerbado, uma vez que como os protocolos baseados em UDP não são orientados à conexão as perdas de pacotes serão tratadas não na camada de transporte, mas sim na de aplicação gerando um overhead maior.

iii) Complexidade:

Uma vez que o protocolo **TFTP** usa UDP (destes 4 é o único), este é caracterizado por não ser orientado à conexão, sendo que apenas envia o pacote e não se preocupa se este chegou ou não com sucesso ao seu destino, sendo assim mais simples, mas sujeito a mais falhas.

O **SFTP** é mais complexo porque encripta os dados, sendo que para tal há um overhead associado (mas também uma maior segurança) e, dos protocolos falados, é dos mais seguros.

O **HTTP** é simples, uma vez que trabalha com Request's e Response's e não tem nada de encriptação. O cliente envia uma request e o servidor responde com uma response

Por fim, o **FTP** estabelece uma comunicação entre cliente e servidor em que o cliente solicita a conexão para ter acesso aos dados que se encontram na internet e o servidor recebe a solicitação do cliente para a transferência dos arquivos e tem que processar e aceitá-la.

iv) **Segurança:**

SFTP é mais seguro porque encripta os dados usando SSH, sendo por isso que conseguimos encontrar as suas ações ao filtrar a porta 22. Consequentemente, podemos sofrer uma perda na performance por causa desta encriptação. É mais seguro que o FTP.

No **FTP**, a segurança não é a sua maior característica. Quando o cliente solicita a conexão que permite aceder aos dados, é necessário uma autenticação (tal como verificamos no exemplo deste trabalho), mas este protocolo não oferece nenhum sistema de encriptação, sendo os dados expostos em texto.

Relativamente ao **HTTP**, este não é encriptado, sendo mais suscetível de expor dados como passwords e dados pessoais. Para uma versão mais segura seria recomendado o uso de HTTPS. Este protocolo opera na porta 80.

Por fim, **TFTP** não tem qualquer tipo de segurança nem encriptação associada, caracterizando-se por ser muito simples e destinado para enviar e receber ficheiros ou configurações que não comprometam a privacidade do utilizador. Este protocolo opera na porta 69.

Parte II.

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	33446	101*8=808
telnet	TELNET	TCP	23	71*20=1420
ftp	FTP	TCP	21	61*20=1220
tftp	TFTP	UDP	69	13*8=104
http(browser)	HTTP	TCP	80	19*20=380
nslookup	DNS	UDP	53	4*8=32
ssh	SSHv2	TCP	22	37*20=740

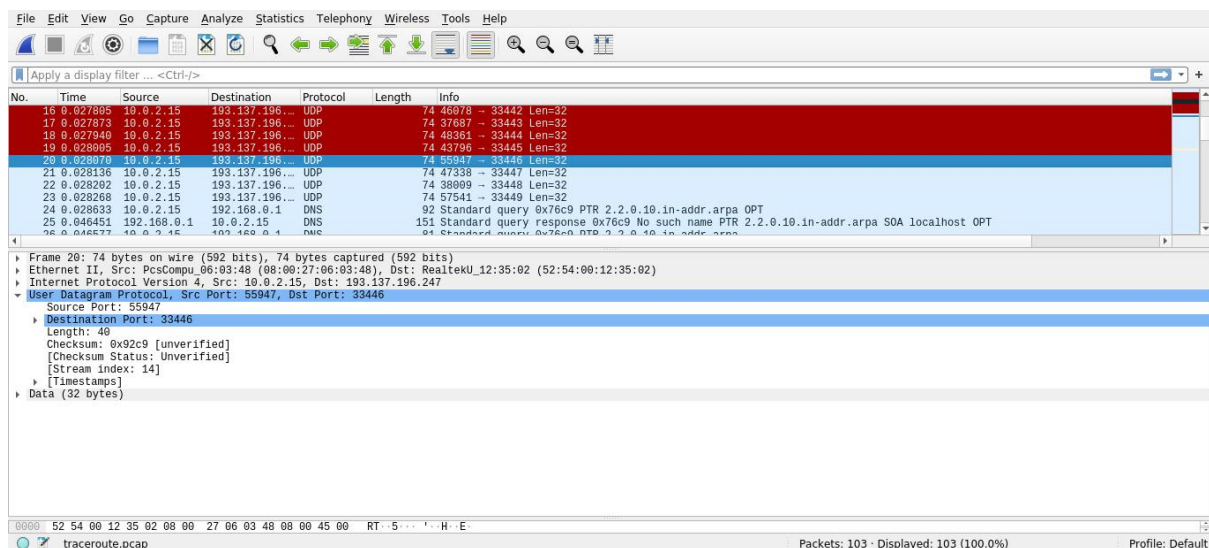


Imagem 1 - TRACEROUTE

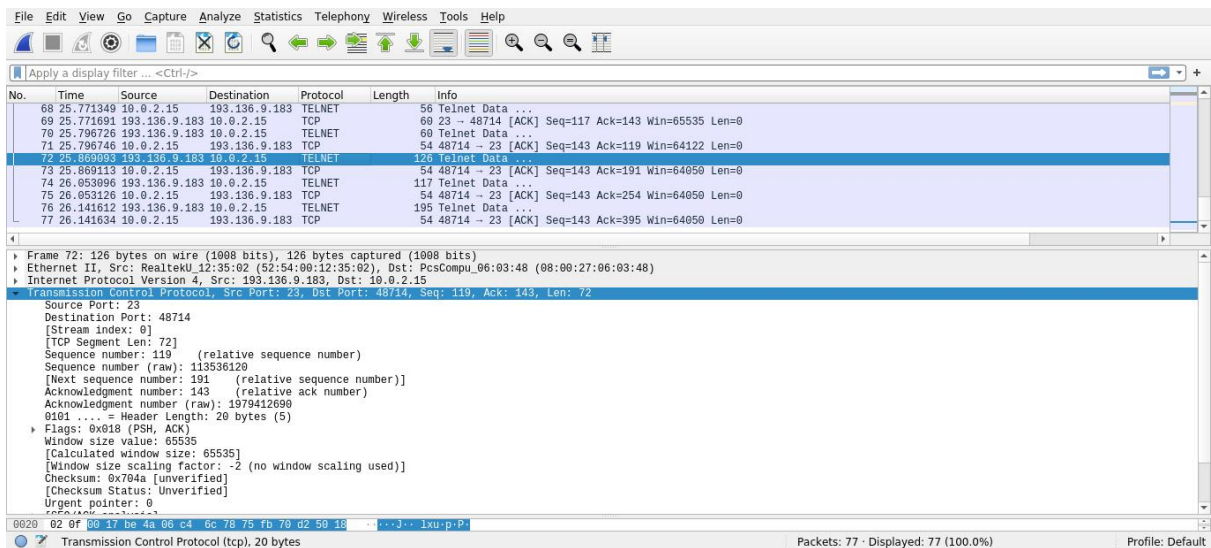


Imagem 2 - TELNET

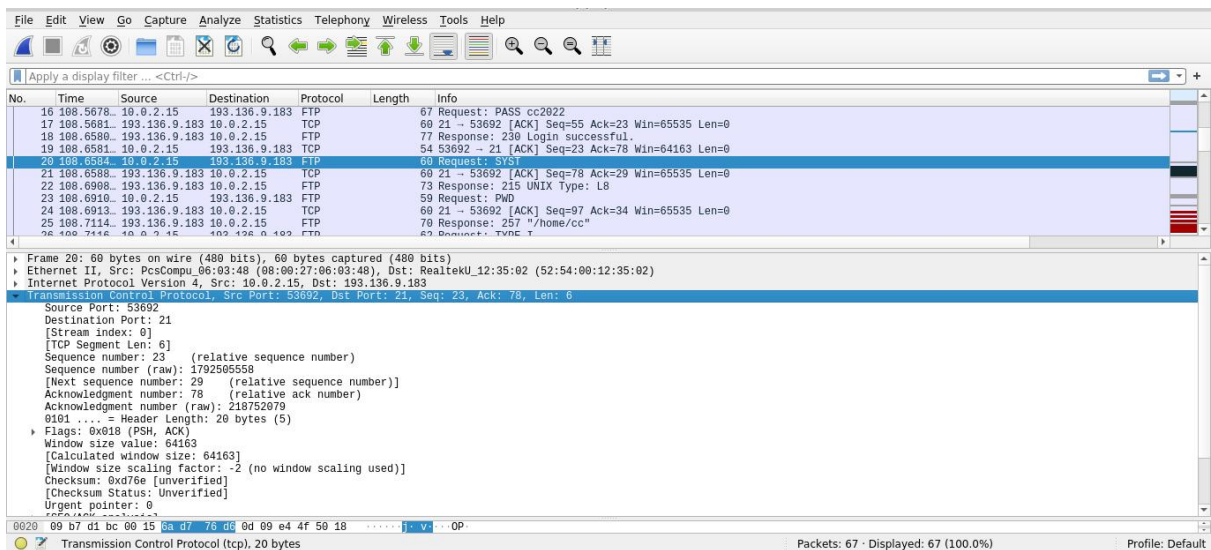


Imagem 3 - FTP

No.	Time	Source	Destination	Protocol	Length	Info
8	19.533619	PcsCompu_06:...	RealtekU_12:...	ARP	42	Who has 10.0.2.2? Tell 10.0.2.15
9	19.534088	RealtekU_12:...	PcsCompu_06:...	ARP	60	10.0.2.2 is at 52:54:00:12:35:02
10	21.387548	10.0.2.15	193.136.9.183	TFTP	86	Read Request, File: file1, Transfer type: octet, tsize=0, blksize=512, timeout=6
11	28.347628	10.0.2.15	193.136.9.183	TFTP	86	Read Request, File: file1, Transfer type: octet, tsize=0, blksize=512, timeout=6
12	35.358935	10.0.2.15	193.136.9.183	TFTP	86	Read Request, File: file1, Transfer type: octet, tsize=0, blksize=512, timeout=6
13	41.365284	10.0.2.15	193.136.9.183	TFTP	86	Read Request, File: file1, Transfer type: octet, tsize=0, blksize=512, timeout=6
14	48.375757	10.0.2.15	193.136.9.183	TFTP	86	Read Request, File: file1, Transfer type: octet, tsize=0, blksize=512, timeout=6
15	53.582446	PcsCompu_06:...	RealtekU_12:...	ARP	42	Who has 10.0.2.2? Tell 10.0.2.15
16	53.582870	RealtekU_12:...	PcsCompu_06:...	ARP	60	10.0.2.2 is at 52:54:00:12:35:02
17	55.384238	10.0.2.15	193.136.9.183	TFTP	86	Read Request, File: file1, Transfer type: octet, tsize=0, blksize=512, timeout=6

Frame 10: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)
 Ethernet II, Src: PcsCompu_06:03:48 (08:00:27:00:03:48), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
 Internet Protocol Version 4, Src: 10.0.2.15, Dst: 193.136.9.183
 User Datagram Protocol, Src Port: 49887, Dst Port: 69
 Source Port: 40807
 Destination Port: 69
 Length: 52
 Checksum: 0xd793 [unverified]
 [Checksum Status: Unverified]
 [Stream index: 2]
 [Timestamps]
 Trivial File Transfer Protocol

0020 09 b7 9f 67 00 45 00 34 d7 93 00 01 66 69 0c 65 ...g.E.4...file

User Datagram Protocol (udp), 8 bytes Packets: 17 · Displayed: 17 (100.0%) Profile: Default

Imagem 4 - TFTP

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.2.15	192.168.0.1	DNS	86	Standard query 0xe0d0 AAAA marco.uminho.pt OPT
2	0.025945	192.168.0.1	10.0.2.15	DNS	140	Standard query response 0xe0d0 AAAA marco.uminho.pt SOA dns.uminho.pt OPT
3	0.027105	10.0.2.15	193.136.9.240	TCP	74	38888 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3029612325 TSecr=0 WS=128
4	0.060390	193.136.9.240	10.0.2.15	TCP	60	80 → 38888 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
5	0.060447	10.0.2.15	193.136.9.240	TCP	54	38888 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
6	0.060507	10.0.2.15	193.136.9.240	TFTP	214	GET /disciplinas/CC-LEI HTTP/1.1
7	0.061938	193.136.9.240	10.0.2.15	TCP	60	80 → 38888 [ACK] Seq=1 Ack=161 Win=65535 Len=0
8	0.093477	193.136.9.240	10.0.2.15	HTTP	616	HTTP/1.1 301 Moved Permanently (text/html)
9	0.093589	10.0.2.15	193.136.9.240	TCP	54	38888 → 80 [ACK] Seq=161 Ack=563 Win=64068 Len=0
10	0.093981	10.0.2.15	193.136.9.240	HTTP	215	GET /disciplinas/CC-LEI HTTP/1.1
11	0.094590	193.136.9.240	10.0.2.15	TCP	60	80 → 38888 [ACK] Seq=161 Ack=563 Win=65535 Len=0

Frame 6: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits)
 Ethernet II, Src: PcsCompu_06:03:48 (08:00:27:00:03:48), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
 Internet Protocol Version 4, Src: 10.0.2.15, Dst: 193.136.9.240
 Transmission Control Protocol, Src Port: 38888, Dst Port: 80, Seq: 1, Ack: 1, Len: 160
 Source Port: 38888
 Destination Port: 80
 [Stream index: 0]
 [TCP Segment Len: 160]
 Sequence number: 1 (relative sequence number)
 Sequence number (raw): 2635474941
 [Next sequence number: 161 (relative sequence number)]
 Acknowledgment number: 1 (relative ack number)
 Acknowledgment number (raw): 125184002
 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: 0xd041 [unverified]
 [Checksum Status: Unverified]
 Urgent pointer: 0

0020 09 f0 0f 00 00 50 9d 15 27 fd 07 76 28 02 50 18 ...P...V...P...

Transmission Control Protocol (tcp), 20 bytes Packets: 23 · Displayed: 23 (100.0%) Profile: Default

Imagem 5 - HTTP

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.2.15	192.168.0.1	DNS	84	Standard query 0x2fa6 A www.uminho.pt OPT
2	0.014099	192.168.0.1	10.0.2.15	DNS	190	Standard query response 0x2fa6 A www.uminho.pt A 193.137.9.114 OPT
3	0.014097	10.0.2.15	192.168.0.1	DNS	84	Standard query 0xb3d9 AAAA www.uminho.pt OPT
4	0.046708	192.168.0.1	10.0.2.15	DNS	138	Standard query response 0xb3d9 AAAA www.uminho.pt SOA dns.uminho.pt OPT
5	5.227220	PcsCompu_06:...	RealtekU_12:...	ARP	42	Who has 10.0.2.2? Tell 10.0.2.15
6	5.227457	RealtekU_12:...	PcsCompu_06:...	ARP	60	10.0.2.2 is at 52:54:00:12:35:02

Frame 1: 84 bytes on wire (672 bits), 84 bytes captured (672 bits) Ethernet II, Src: PcsCompu_06:03:48 (08:00:27:06:03:48), Dst: RealtekU_12:35:02 (52:54:00:12:35:02) Internet Protocol Version 4, Src: 10.0.2.15, Dst: 192.168.0.1 User Datagram Protocol, Src Port: 35987, Dst Port: 53
Source Port: 35987 Destination Port: 53 Length: 50 Checksum: 0xc9fb [unverified] [Checksum Status: Unverified] [Stream index: 0] [Timestamps]
Domain Name System (query)

0020	00 01 8c 93 00 35 00 32 cc ff 2f a6 01 00 00 01	...5.2.../.....
------	-------------------------------------------------	-----------------

User Datagram Protocol (udp), 8 bytes Packets: 6 · Displayed: 6 (100.0%) Profile: Default

Imagem 6 - NSLOOKUP

No.	Time	Source	Destination	Protocol	Length	Info
5	0.146313	10.0.2.15	193.136.9.183	TCP	74	45140 → 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=217763438 TSecr=0 WS=128
6	0.166761	193.136.9.183	10.0.2.15	TCP	60	22 → 45140 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
7	0.166799	10.0.2.15	193.136.9.183	TCP	54	45140 → 22 [ACK] Seq=1 Ack=1 Win=64240 Len=0
8	0.167167	10.0.2.15	193.136.9.183	SSHv2	95	Client: Protocol (SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.3)
9	0.167319	193.136.9.183	10.0.2.15	TCP	60	22 → 45140 [ACK] Seq=1 Ack=42 Win=65535 Len=0
10	0.238837	193.136.9.183	10.0.2.15	SSHv2	95	Server: Protocol (SSH-2.0-OpenSSH_5.9p1 Debian-Subuntu1.4)
11	0.238857	10.0.2.15	193.136.9.183	TCP	54	45140 → 22 [ACK] Seq=42 Ack=42 Win=64199 Len=0
12	0.239312	10.0.2.15	193.136.9.183	SSHv2	1566	Client: Key Exchange Init
13	0.239552	193.136.9.183	10.0.2.15	TCP	60	22 → 45140 [ACK] Seq=42 Ack=1502 Win=65535 Len=0
14	0.239563	193.136.9.183	10.0.2.15	TCP	60	22 → 45140 [ACK] Seq=42 Ack=1554 Win=65535 Len=0
15	0.242420	10.0.2.15	193.136.9.183	SSHv2	1030	Server: Key Exchange Init

Frame 8: 95 bytes on wire (760 bits), 95 bytes captured (760 bits) Ethernet II, Src: PcsCompu_06:03:48 (08:00:27:06:03:48), Dst: RealtekU_12:35:02 (52:54:00:12:35:02) Internet Protocol Version 4, Src: 10.0.2.15, Dst: 193.136.9.183 Transmission Control Protocol, Src Port: 45140, Dst Port: 22, Seq: 1, Ack: 1, Len: 41
Source Port: 45140 Destination Port: 22 [Stream index: 0] [TCP Segment Len: 41] Sequence number: 1 (relative sequence number) Sequence number (raw): 1306595371 [Next sequence number: 42 (relative sequence number)] Acknowledgment number: 1 (relative ack number) Acknowledgment number (raw): 65216002 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: 0xd791 [unverified] [Checksum Status: Unverified] Urgent pointer: 0

0020	09 07 00 54 00 18 4d e1 10 2b 03 e3 1e 02 50 18	...T..M..:....P..
------	-------------------------------------------------	-------------------

Transmission Control Protocol (tcp), 20 bytes Packets: 41 · Displayed: 41 (100.0%) Profile: Default

Imagem 9 - SSH