Protocolos da Camada de Transporte

Filipa Rebelo, Rúben Rodrigues, and Guilherme Gonçalves

Universidade do Minho, Departamento de Informática, 4710-057 Braga, Portugal
email: {a90234,a80960,a88280}@alunos.uminho.pt

1 Questões

1.1 Exercício 1

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

A camada de transporte lidou com as perdas e duplicações de pacotes. Para a resolução deste problema, os mecanismos de controlo de erros do protocolo TCP descarta o pacote e reenvia-o, sem alterar a sua integridade. Por outro lado, o protocolo UDP apenas descarta o pacote.

Para evitar problemas de perdas e duplicações, o uso de protocolos TCP, que são orientados a conexões, passam a ser mais usados apesar de estes criarem possíveis atrasos na chegada e processamento de dados, ocorrendo um overhead e penalizando o desmpenho. No caso dos protocolos UDP, como existe uma probabilidade de que exista informação perdida, este implica que a sua resolução ocorra na camada da aplicação, penalizando o seu empenho.

1.2 Exercício 2

Obtenha a partir do Wireshark, ou desenhe manualmente, um diagrama temporal para a transferência do ficheiro file1 por FTP realizada em A.3. Foque-se apenas na transferência de dados [ftp-data] e não na conexão de controlo (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 tanto nos dados como nas confirmações.

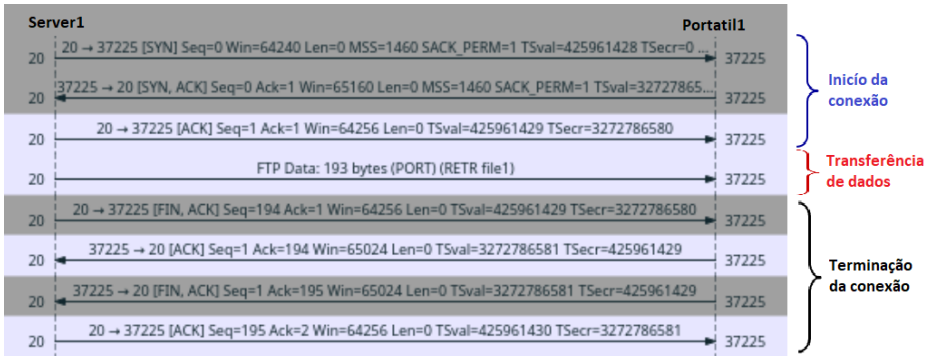

Fig. 1. Diagrama temporal da transferência do file1 por FTP.

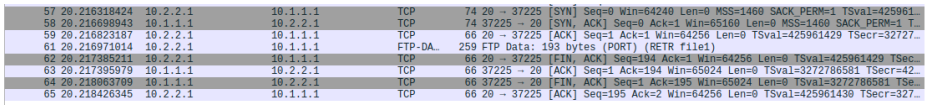

Fig. 2. FTP.

1.3 Exercício 3

Obtenha a partir do Wireshark, ou desenhe manualmente, um diagrama temporal para a transferência do ficheiro file1 por TFTP realizada em A.4. 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 tanto nos dados como nas confirmações.

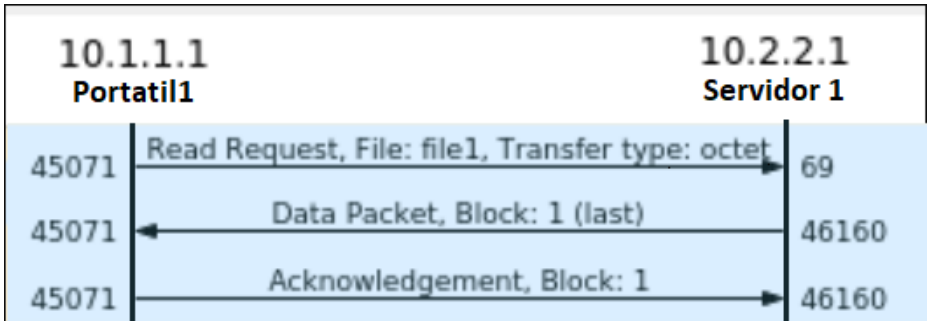

Fig. 3. Diagrama temporal da transferência do file1 por TFTP.

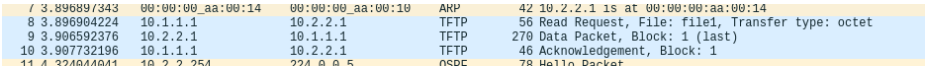

Fig. 4. TFTP.

1.4 Exercício 4

Compare sucintamente as quatro aplicações de transferência de ficheiros que usou, tendo em consideração os seguintes aspetos: (i) identificação da camada de transporte; (ii) eficiência; (iii) complexidade; (iv) segurança.

Camada de Transporte: No que diz respeito ao uso da camada de transporte através da análise das quatro aplicações de transferências de ficheiros utilizadas verificamos que SFTP, FTP e HTTP utilizam o protocolo TCP e que TFTP utiliza o protocolo UDP.

Eficiência: Para compararmos os protocolos utilizados em termos de eficiência recorremos ao wireshark para capturar o tráfego de dados tendo sido verificado que :

FTP: Não é um protocolo seguro uma vez que este não utiliza encriptação ficando desta forma os dados utilizados pelos utilizadores vulneráveis e suscetiveis a ataques.

SFTP: Semelhante com FTP porém neste os dados são encriptados sendo a sua conexão mais segura, é o mais eficaz uma vez que se pode transferir vários ficheiros rapidamente.

HTTP: Eficiente na transferência de ficheiros pequenos uma vez que usa o protocolo TCP que é responsavel pela verificação de erros.

TFTP: é o mais eficaz de todos os protocolos dado que tem um baixo overhead.

Ordem Crescente de Eficiência: HTTP-FTP-SFTP-TFTP

Complexidade: SFTP: É muito complexo dado que usa o protocolo TCP, protocolo mais complexo da camada de transporte, utilizando mais do que uma conexão e tem de cripografar os dados transferidos.

FTP: É muito complexo dado que utiliza o protocolo TCP, protocolo mais complexo da camada de tranporte, e dado que tem permissão para transferir diversos ficheiros em simultâneo cada uma destas transferências cria uma conexão tendo como resultado várias velocidades de conexão.

TFTP: Quando comparado com os outros protocolos é o que apresenta menor complexidade uma vez que este utiliza o protocolo UDP que é um protocolo mais simples.

HTTP: É muito complexo uma vez que utiliza o protocolo TCP, protocolo mais complexo da camada de transporte,mas utiliza apenas uma conexão.

Ordem Crescente de Complexidade: TFTP-HTTP-FTP-SFTP

Segurança: SFTP: É muito seguro uma vez que utiliza o ssh para encriptar os dados de maneira aos dados não serem inrercetados por terceiros.

FTP: É pouco seguro dado que não fornece mecanismos de encriptação, podendo qualquer pessoa fazer a captura de pacotes.

TFTP: Não tem nenhuma segurança nem encriptação associada sendo bastante simples e destinado a receber e enviar ficheiros que não comprometam a privacidade do utilizador.

HTTP: É pouco seguro dado que não é encriptado sendo os dados acessíveis a qualquer pessoa.

1.5 Exercício 5

Com base no trabalho realizado, construa uma tabela informativa identificando, para cada aplicação executada (ping, traceroute, telnet, ftp, tftp, wget/lynx, nslookup, ssh, etc.), qual o protocolo de aplicação, o protocolo de transporte, a porta de atendimento e o overhead de transporte.

Para calcular a percentagem de overhead de transporte fizemos (Header Length / Total Length)*100.

Aplicação Executada	Protocolo de Aplicação	Protocolo de transporte	Porta de atendimento	Overhead de transporte
Ping	--	--	--	--
Tracerout	--	UDP	33434	33%
Telnet	TELNET	TCP	23	35%
Ftp	FTP	TCP	21	38%
Tftp	TFTP	UDP	69	28%
Wget	HTTP	TCP	80	38%
Nslookup	DNS	UDP	53	29%
Ssh	SSHV2	TCP	22	12%

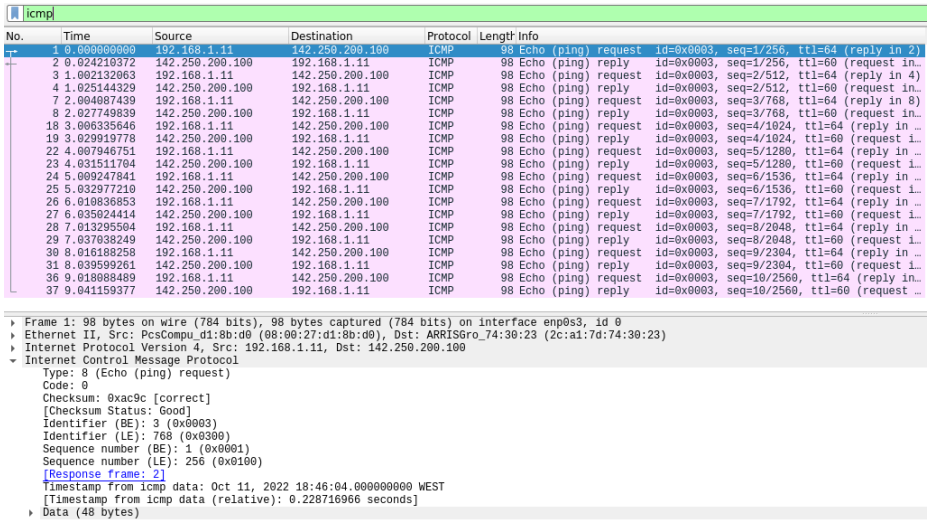

Fig. 5. PING

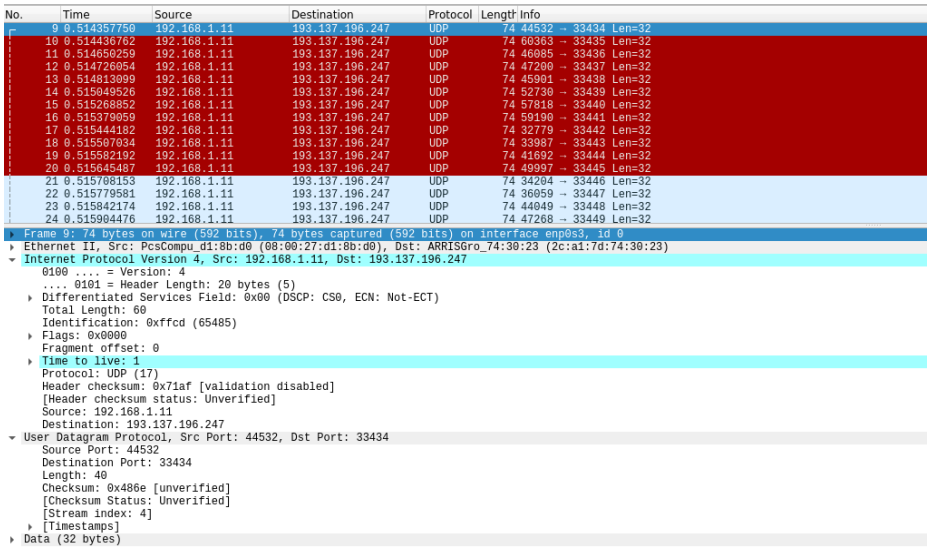

Fig. 6. TRACEROUT

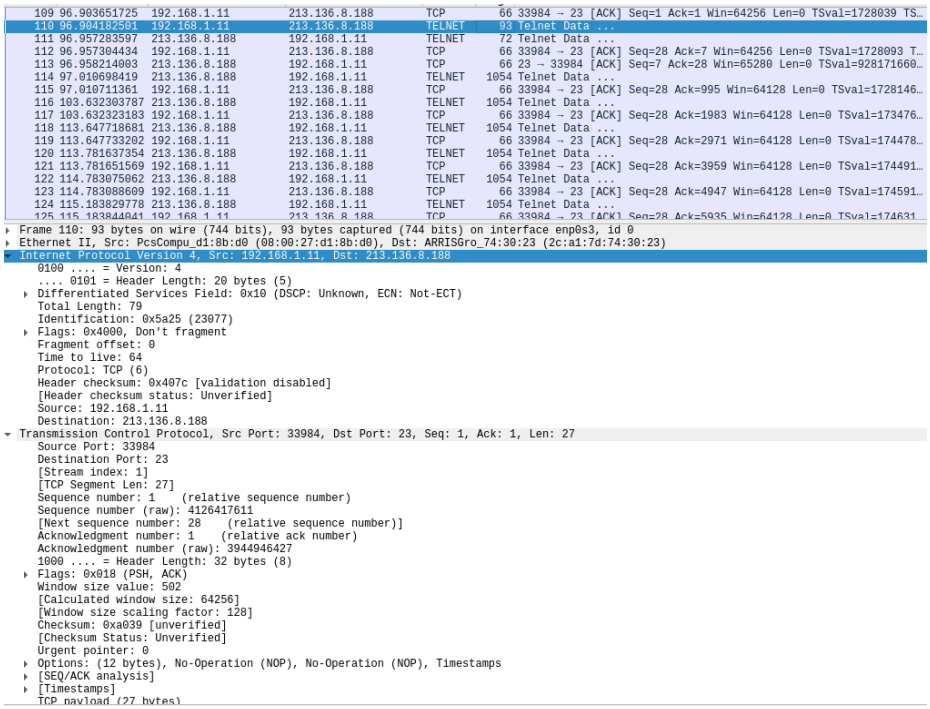
109 96.903651725 192.168.1.11 213.136.8.188 TCP 66 33984 → 23 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1728039 TS…
110 96.904182501 192.168.1.11 213.136.8.188 TELNET 93 Telnet Data ...
111 96.957283597 213.136.8.188 192.168.1.11 TELNET 72 Telnet Data ...
112 96.957304434 192.168.1.11 213.136.8.188 TCP 66 33984 → 23 [ACK] Seq=28 Ack=7 Win=64256 Len=0 TSval=1728093 T…
113 96.958214003 213.136.8.188 192.168.1.11 TCP 66 23 → 33984 [ACK] Seq=7 Ack=28 Win=65280 Len=0 TSval=928171660…
114 97.010698419 213.136.8.188 192.168.1.11 TELNET 1054 Telnet Data ...
115 97.010711361 192.168.1.11 213.136.8.188 TCP 66 33984 → 23 [ACK] Seq=28 Ack=995 Win=64128 Len=0 TSval=1728146…
116 103.632303787 213.136.8.188 192.168.1.11 TELNET 1054 Telnet Data ...
117 103.632323183 192.168.1.11 213.136.8.188 TCP 66 33984 → 23 [ACK] Seq=28 Ack=1983 Win=64128 Len=0 TSval=173476…
118 113.647718681 213.136.8.188 192.168.1.11 TELNET 1054 Telnet Data ...
119 113.647733202 192.168.1.11 213.136.8.188 TCP 66 33984 → 23 [ACK] Seq=28 Ack=2971 Win=64128 Len=0 TSval=174478…
120 113.781637354 213.136.8.188 192.168.1.11 TELNET 1054 Telnet Data ...
121 113.781651569 192.168.1.11 213.136.8.188 TCP 66 33984 → 23 [ACK] Seq=28 Ack=3959 Win=64128 Len=0 TSval=174491…
122 114.783075062 213.136.8.188 192.168.1.11 TELNET 1054 Telnet Data ...
123 114.783088609 192.168.1.11 213.136.8.188 TCP 66 33984 → 23 [ACK] Seq=28 Ack=4947 Win=64128 Len=0 TSval=174591…
124 115.183829778 213.136.8.188 192.168.1.11 TELNET 1054 Telnet Data ...
125 115.183844041 192.168.1.11 213.136.8.188 TCP 66 33984 → 23 [ACK] Seq=28 Ack=5935 Win=64128 Len=0 TSval=174631…
Frame 110: 93 bytes on wire (744 bits), 93 bytes captured (744 bits) on interface enp0s3, id 0
Ethernet II, Src: PcsCompu_d1:8b:d0 (08:00:27:d1:8b:d0), Dst: ARRISGro_74:30:23 (2c:a1:7d:74:30:23)
Internet Protocol Version 4, Src: 192.168.1.11, Dst: 213.136.8.188
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x10 (DSCP: Unknown, ECN: Not-ECT)
Total Length: 79
Identification: 0x5a25 (23077)
Flags: 0x4000, Don't fragment
Fragment offset: 0
Time to live: 64
Protocol: TCP (6)
Header checksum: 0x407c [validation disabled]
[Header checksum status: Unverified]
Source: 192.168.1.11
Destination: 213.136.8.188
Transmission Control Protocol, Src Port: 33984, Dst Port: 23, Seq: 1, Ack: 1, Len: 27
Source Port: 33984
Destination Port: 23
[Stream index: 1]
[TCP Segment Len: 27]
Sequence number: 1 (relative sequence number)
Sequence number (raw): 4126417611
[Next sequence number: 28 (relative sequence number)]
Acknowledgment number: 1 (relative ack number)
Acknowledgment number (raw): 3944946427
1000 .... = Header Length: 32 bytes (8)
Flags: 0x018 (PSH, ACK)
Window size value: 502
[Calculated window size: 64256]
[Window size scaling factor: 128]
Checksum: 0xa039 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[SEQ/ACK analysis]
[Timestamps]
TCP payload (27 bytes)

Fig. 7. TELNET

Fig. 8. FTP

Fig. 9. TFTP

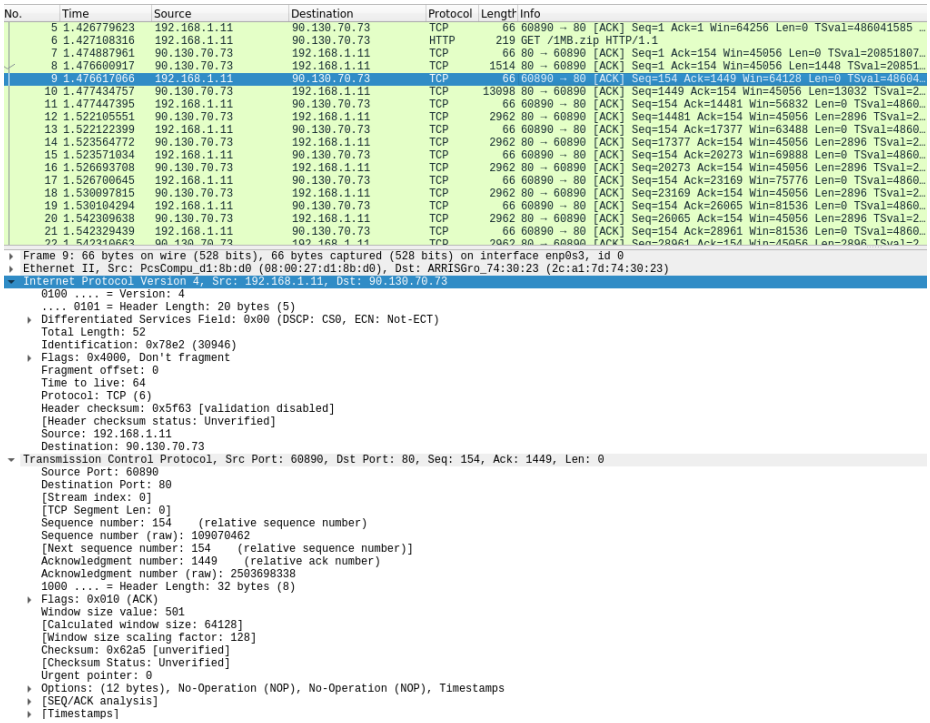

Fig. 10. WGET

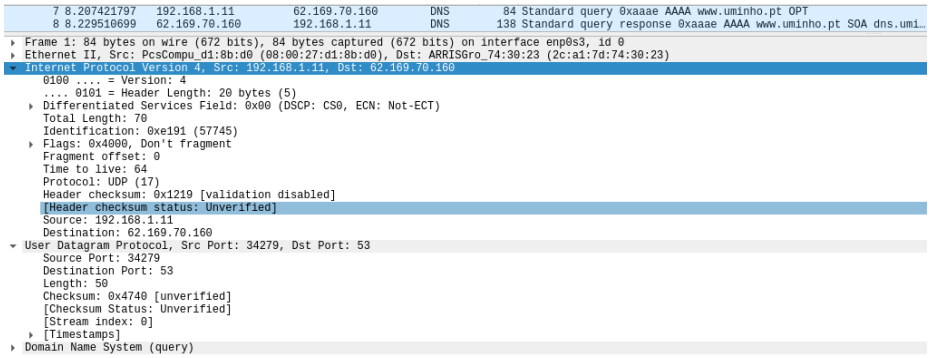

Fig. 11. NSLOOKUP

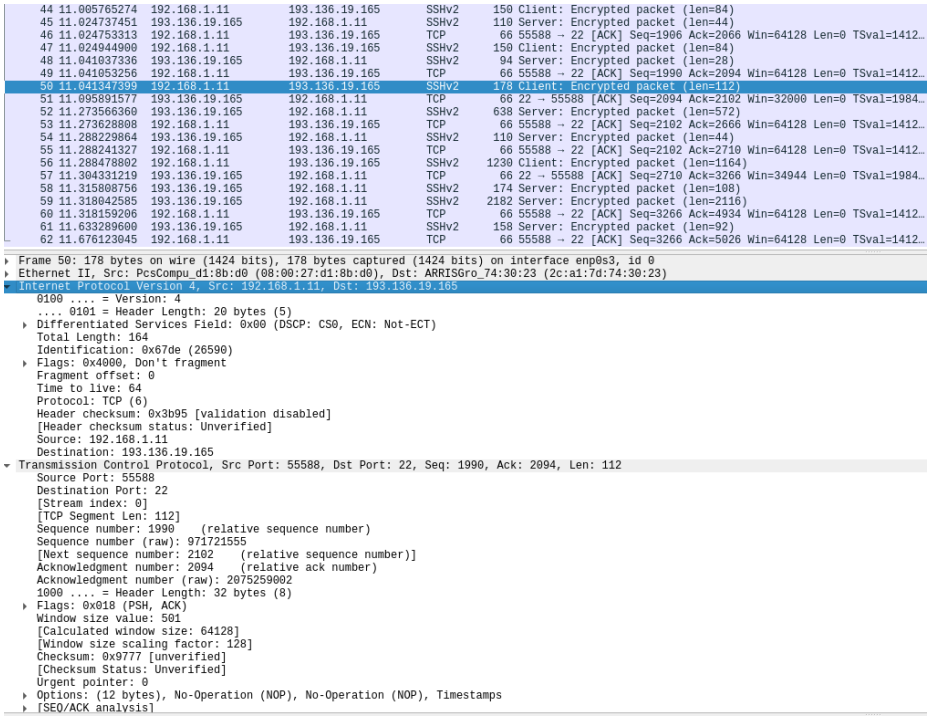

Fig. 12. SSH

2 Conclusão

A realização deste trabalho prático permitiu-nos consolidar alguns dos conhecimentos abordados nas aulas teóricas relativos à camada de transporte e aos seus protocolos. Foi ainda possível identificar as diferenças entre os protocolos TCP e UDP e o seu impacto e analisar outros protocolos tais como TFTP, HTTP, SFTP e FTP. Em conclusão, acreditamos ter obtido o aproveitamento esperado na realização deste trabalho e também conhecimentos importantes relativos ao tema do mesmo.