



Universidade do Minho
Escola de Engenharia

Protocolos da Camada de Transporte

Comunicações por Computador G6.10

Licenciatura em Engenharia Informática

Gonçalo Freitas, José Martins
e-mail: [a96136, a97903]@alunos.uminho.pt



Universidade do Minho
Departamento de Informática
4710-057
Braga, Portugal

Conteúdo

Questões e respostas3

1. Pergunta 13

2. Pergunta 24

3. Pergunta 35

4. Pergunta 46

5. Pergunta 57

Questões e respostas

1. Pergunta 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.

Resposta: Após a verificação da conectividade dos nós Portátil1(LAN1) e Grilo(LAN4), registamos o computador Grilo foi mais lento, uma vez que essa rede tinha associada uma probabilidade de perda e duplicações, tal foi o que aconteceu. Quando efetuamos o *ping* no Grilo registamos uma perda de informação de 5% - 5% *packet loss*. O mesmo não se verificou no Portátil 1 que não registou qualquer perda de informação – 0% *packet loss*. A camada que lidou com os problemas referidos foi a camada de transporte pois é o TCP que lida com as perdas e retransmite o pacote e pertence à camada de aplicação.

```
<45103/Portatil1.conf# ping -c 20 10.2.2.1 | tee file-ping-output
PING 10.2.2.1 (10.2.2.1) 56(84) bytes of data.
64 bytes from 10.2.2.1: icmp_seq=1 ttl=61 time=1.30 ms
64 bytes from 10.2.2.1: icmp_seq=2 ttl=61 time=32.2 ms
64 bytes from 10.2.2.1: icmp_seq=3 ttl=61 time=0.866 ms
64 bytes from 10.2.2.1: icmp_seq=4 ttl=61 time=0.516 ms
64 bytes from 10.2.2.1: icmp_seq=5 ttl=61 time=0.701 ms
64 bytes from 10.2.2.1: icmp_seq=6 ttl=61 time=21.6 ms
64 bytes from 10.2.2.1: icmp_seq=7 ttl=61 time=21.9 ms
less 64 bytes from 10.2.2.1: icmp_seq=8 ttl=61 time=0.728 ms
64 bytes from 10.2.2.1: icmp_seq=9 ttl=61 time=21.5 ms

64 bytes from 10.2.2.1: icmp_seq=10 ttl=61 time=17.8 ms
64 bytes from 10.2.2.1: icmp_seq=11 ttl=61 time=6.61 ms
64 bytes from 10.2.2.1: icmp_seq=12 ttl=61 time=0.656 ms
64 bytes from 10.2.2.1: icmp_seq=13 ttl=61 time=23.5 ms
64 bytes from 10.2.2.1: icmp_seq=14 ttl=61 time=0.908 ms
64 bytes from 10.2.2.1: icmp_seq=15 ttl=61 time=32.3 ms
64 bytes from 10.2.2.1: icmp_seq=16 ttl=61 time=2.13 ms
64 bytes from 10.2.2.1: icmp_seq=17 ttl=61 time=32.4 ms
64 bytes from 10.2.2.1: icmp_seq=18 ttl=61 time=0.574 ms
64 bytes from 10.2.2.1: icmp_seq=19 ttl=61 time=31.8 ms
64 bytes from 10.2.2.1: icmp_seq=20 ttl=61 time=1.26 ms

--- 10.2.2.1 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 1919ms
rtt min/avg/max/mdev = 0.516/12.558/32.425/12.860 ms
```

Figura 1: Resultados obtidos em Portátil1

```
<9/Grilo.conf# ping -c 20 10.2.2.1 | tee file-ping-output
PING 10.2.2.1 (10.2.2.1) 56(84) bytes of data.
64 bytes from 10.2.2.1: icmp_seq=1 ttl=61 time=35.4 ms
64 bytes from 10.2.2.1: icmp_seq=2 ttl=61 time=8.77 ms
64 bytes from 10.2.2.1: icmp_seq=3 ttl=61 time=16.3 ms
64 bytes from 10.2.2.1: icmp_seq=4 ttl=61 time=26.2 ms
64 bytes from 10.2.2.1: icmp_seq=5 ttl=61 time=6.06 ms
64 bytes from 10.2.2.1: icmp_seq=6 ttl=61 time=22.0 ms
64 bytes from 10.2.2.1: icmp_seq=7 ttl=61 time=32.9 ms
64 bytes from 10.2.2.1: icmp_seq=8 ttl=61 time=23.5 ms
64 bytes from 10.2.2.1: icmp_seq=9 ttl=61 time=20.3 ms
64 bytes from 10.2.2.1: icmp_seq=10 ttl=61 time=20.5 ms
64 bytes from 10.2.2.1: icmp_seq=11 ttl=61 time=42.6 ms
64 bytes from 10.2.2.1: icmp_seq=12 ttl=61 time=23.1 ms
64 bytes from 10.2.2.1: icmp_seq=13 ttl=61 time=9.98 ms
64 bytes from 10.2.2.1: icmp_seq=14 ttl=61 time=24.7 ms
64 bytes from 10.2.2.1: icmp_seq=15 ttl=61 time=16.4 ms
64 bytes from 10.2.2.1: icmp_seq=16 ttl=61 time=42.7 ms
64 bytes from 10.2.2.1: icmp_seq=16 ttl=61 time=42.7 ms (DUP!)
64 bytes from 10.2.2.1: icmp_seq=17 ttl=61 time=27.1 ms
64 bytes from 10.2.2.1: icmp_seq=19 ttl=61 time=30.7 ms
64 bytes from 10.2.2.1: icmp_seq=20 ttl=61 time=6.67 ms

--- 10.2.2.1 ping statistics ---
20 packets transmitted, 19 received, +1 duplicates, 5% packet loss, time 1916ms
rtt min/avg/max/mdev = 6.059/23.928/42.654/11.202 ms
```

Figura 2: Resultados obtidos em Grilo

2. Pergunta 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.

Resposta: Como se pode observar pelas imagens abaixo representadas, a transferência do *file1* tem um número relativamente baixo de trocas de pacotes.

Dentro das imagens abaixo representadas podemos ver o fluxo de tempo da transferência do *file1*, e as ações requisitadas e efetuadas nesta mesma sequência.

Já no print do *wireshark*, podemos ver toda a ação, desde o login até à transferência do ficheiro, e posterior saída.

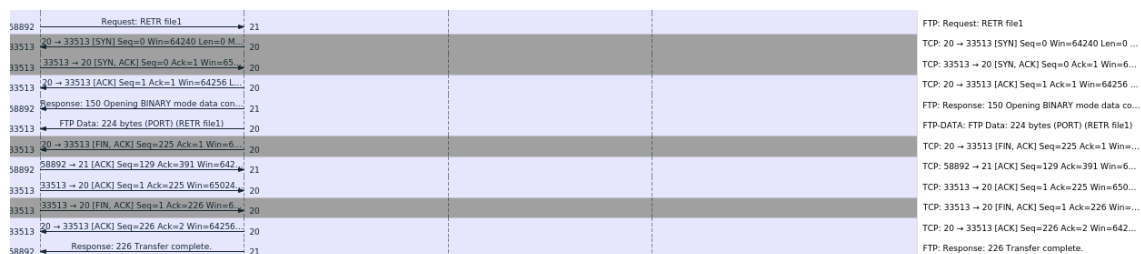


Figura 3: Gráfico de tempo Portatil1

55	78.999108025	10.1.1.1	10.2.2.1	FTP	82 Request: USER anonymous
57	78.999340465	10.2.2.1	10.1.1.1	FTP	100 Response: 331 Please specify the password.
64	88.343541721	10.1.1.1	10.2.2.1	FTP	96 Request: PASS a97903@alunos.uminho.pt
66	88.345471832	10.2.2.1	10.1.1.1	FTP	89 Response: 230 Login successful.
68	88.346523830	10.1.1.1	10.2.2.1	FTP	72 Request: SYST
70	88.346752824	10.2.2.1	10.1.1.1	FTP	85 Response: 215 UNIX Type: L8
78	98.884950621	10.1.1.1	10.2.2.1	FTP	71 Request: PWD
79	98.885187369	10.2.2.1	10.1.1.1	FTP	100 Response: 257 "/" is the current directory
84	102.416464911	10.1.1.1	10.2.2.1	FTP	88 Request: PORT 10,1,1,152,93
85	102.416701169	10.2.2.1	10.1.1.1	FTP	117 Response: 200 PORT command successful. Consider using PASV.
87	102.417181360	10.1.1.1	10.2.2.1	FTP	72 Request: LIST
91	102.417950520	10.2.2.1	10.1.1.1	FTP	105 Response: 150 Here comes the directory listing.
98	102.419550731	10.2.2.1	10.1.1.1	FTP	90 Response: 226 Directory send OK.
102	106.075247528	10.1.1.1	10.2.2.1	FTP	74 Request: TYPE I
103	106.075585493	10.2.2.1	10.1.1.1	FTP	97 Response: 200 Switching to Binary mode.
105	106.075943046	10.1.1.1	10.2.2.1	FTP	88 Request: PORT 10,1,1,132,31
106	106.076519723	10.2.2.1	10.1.1.1	FTP	117 Response: 200 PORT command successful. Consider using PASV.
108	106.077095910	10.1.1.1	10.2.2.1	FTP	78 Request: RETR file1
112	106.078185431	10.2.2.1	10.1.1.1	FTP	130 Response: 150 Opening BINARY mode data connection for file1 (...)
119	106.079540908	10.2.2.1	10.1.1.1	FTP	90 Response: 226 Transfer complete.
122	108.750185512	10.1.1.1	10.2.2.1	FTP	72 Request: QUIT
123	108.751320261	10.2.2.1	10.1.1.1	FTP	80 Response: 221 Goodbye.

Figura 4: Wireshark Portatil1

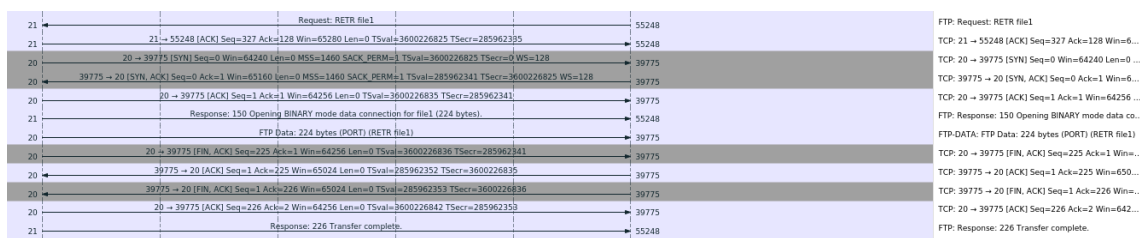


Figura 5: Gráfico de tempo Grilo

148	133.343215721	10.4.4.1	10.2.2.1	FTP	82 Request: USER anonymous
150	133.343919875	10.2.2.1	10.4.4.1	FTP	100 Response: 331 Please specify the password.
158	141.951893500	10.4.4.1	10.2.2.1	FTP	96 Request: PASS a97903@alunos.uminho.pt
160	141.953849460	10.2.2.1	10.4.4.1	FTP	89 Response: 230 Login successful.
162	141.960539485	10.4.4.1	10.2.2.1	FTP	72 Request: SYST
164	141.961075534	10.2.2.1	10.4.4.1	FTP	85 Response: 215 UNIX Type: L8
172	150.050318438	10.4.4.1	10.2.2.1	FTP	71 Request: PWD
173	150.050888954	10.2.2.1	10.4.4.1	FTP	100 Response: 257 "/" is the current directory
177	151.389082032	10.4.4.1	10.2.2.1	FTP	88 Request: PORT 10,4,4,1,149,91
178	151.389740438	10.2.2.1	10.4.4.1	FTP	117 Response: 200 PORT command successful. Consider using PASV.
180	151.394965845	10.4.4.1	10.2.2.1	FTP	72 Request: LIST
184	151.400857993	10.2.2.1	10.4.4.1	FTP	105 Response: 150 Here comes the directory listing.
190	151.407405807	10.2.2.1	10.4.4.1	FTP	90 Response: 226 Directory send OK.
194	154.419110435	10.2.2.1	10.2.2.1	FTP	74 Request: TYPE I
195	154.419660000	10.4.4.1	10.4.4.1	FTP	97 Response: 200 Switching to Binary mode.
197	154.425296944	10.4.4.1	10.2.2.1	FTP	88 Request: PORT 10,4,4,1,174,47
200	154.426112314	10.2.2.1	10.4.4.1	FTP	117 Response: 200 PORT command successful. Consider using PASV.
201	154.431523791	10.4.4.1	10.2.2.1	FTP	78 Request: RETR file1
206	154.437800976	10.2.2.1	10.4.4.1	FTP	130 Response: 150 Opening BINARY mode data connection for file1 (
212	154.446134596	10.2.2.1	10.4.4.1	FTP	90 Response: 226 Transfer complete.
215	157.640905155	10.4.4.1	10.2.2.1	FTP	72 Request: QUIT
217	157.642395874	10.2.2.1	10.4.4.1	FTP	80 Response: 221 Goodbye.

Figura 6: Wireshark Grilo 1

3. Pergunta 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.

Resposta: Com base nos gráficos do *Wireshark*, os tipos de segmentos trocados foram: *Read Request*, *Data Packet* e *Acknowledgement*.

18	25.406143736	10.1.1.1	10.2.2.1	TFTP	56 Read Request, File: file1, Transfer type: octet
19	25.406535425	10.2.2.1	10.1.1.1	TFTP	270 Data Packet, Block: 1 (last)
20	25.407108205	10.1.1.1	10.2.2.1	TFTP	46 Acknowledgement, Block: 1
37	44.876329563	10.4.4.1	10.2.2.1	TFTP	56 Read Request, File: file1, Transfer type: octet
38	44.877022837	10.2.2.1	10.4.4.1	TFTP	270 Data Packet, Block: 1 (last)
39	44.883648963	10.4.4.1	10.2.2.1	TFTP	46 Acknowledgement, Block: 1

Figura 7: Wireshark Portatil1 e Grilo

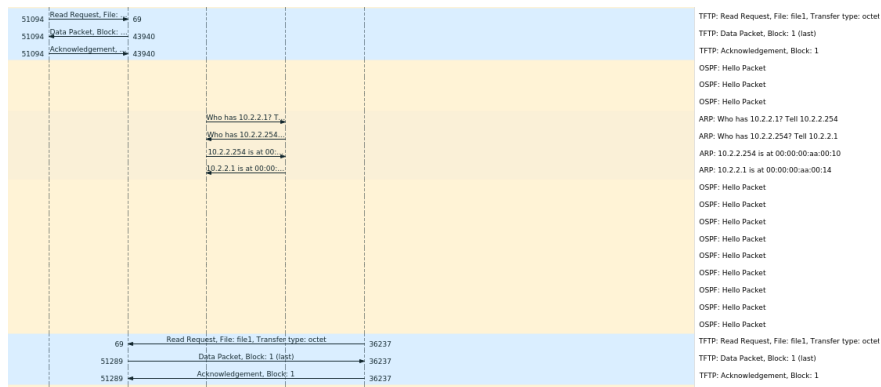


Figura 8: Gráfico temporal portatil1 e Grilo 1

4. Pergunta 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.

Resposta: Nos protocolos FTP, SFTP E HTTP é usado o TCP como camada de transporte. Assim podemos considerar estes protocolos mais eficientes, em relação ao TFTP que usa como camada de transporte o UDP pois o TCP tenta fazer a correção de erros enquanto com o UDP a correção de erros é a nível aplicacional.

Porém, como não existe o protocolo perfeito, o TFTP é considerado menos complexo em relação aos restantes pois é um protocolo de transporte muito mais simples que o TCP, embora o UDP não nos garanta a total fiabilidade dos dados. Podemos considerar que o SFTP é o mais complexo pois também usa SSH para encriptação.

Por outro lado, o SFTP, ao ser o único protocolo a usar SSH para encriptação é também o único protocolo a seguro já que mais nenhum faz encriptação de dados.

5. Pergunta 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.

Resposta:

	Protocolo de aplicação	Protocolo de transporte	Porta de atendimento	Overhead de transporte
PING	PING	-	-	-
TRACEROUTE	TRACEROUTE	UDP	33434	8
TELNET	TELNET	TCP	23	40
FTP	FTP	TCP	21	20
TFTP	TFTP	UDP	69	8
WGET/LYNX	HTTP	TCP	80	20
NSLOOKUP	DNS	UDP	53	8
SSH	SSH	TCP	22	40