# Comunicações por Computador (2023/2024) - PL6

Universidade do Minho - Campus de Gualtar, R. da Universidade, 4710-057 Braga Portugal



Trabalho Prático  $N^{o}1$  - Grupo 62



Duarte Leit $\tilde{\mathbf{a}}\mathbf{o}^{[a100550]}$ 



Hugo  $\mathbf{Ramos}^{[a100644]}$ 



Diogo Araújo $^{[a100544]}$ 

#### 2

### 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.

O desempenho das aplicações será afetado por perdas e duplicações caso estas usem TCP. Isto deve-se ao simples facto de as perdas de pacotes resultarem num aumento no tempo de espera dos clientes e dos servidores, porque muito provavelmente será necessário reenviá-los. No caso das duplicações de pacotes, quando estas ocorrem, podem levar a um aumento no tráfego da rede e causar congestionamentos.

Tanto a camada de transporte quanto a da aplicação podem lidar com esses problemas, dependendo do protocolo de transporte em uso. No caso do TCP, as aplicações não se precisam de preocupar com isso, uma vez que este protocolo é altamente confiável, garantindo que os dados chegam ao destino. Caso ocorra alguma falha na entrega, o TCP informa a aplicação. Por outro lado, no caso do UDP, a aplicação é responsável por lidar com as perdas e duplicações, pois este protocolo simplesmente envia os dados, sem se preocupar com as condições da rede ou se alguém está disponível para recebê-los.

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.

Time	10.1	.1.1	1.4.1	Comment
326.227206316	58496	Request: RETR file1	21	FTP: Request: RETR file1
326.227772541	39725	20 → 39725 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2539774988 TSecr=	20	TCP: 20 → 39725 [SYN] Seq=0 Win=64240 Len=0
326.227915640	39725	39725 → 20 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=680222	20	TCP: 39725 → 20 [SYN, ACK] Seq=0 Ack=1 Win=6
326.228034624	39725	20 → 39725 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2539774988 TSecr=680222092	20	TCP: 20 → 39725 [ACK] Seq=1 Ack=1 Win=64256
326.228082982	58496	Response: 150 Opening BINARY mode data connection for file1 (224 bytes).	21	FTP: Response: 150 Opening BINARY mode data co
326.228188667	39725	FTP Data: 224 bytes (PORT) (RETR file1)	20	FTP-DATA: FTP Data: 224 bytes (PORT) (RETR file1)
326.228190464	39725	20 → 39725 [FIN, ACK] Seq=225 Ack=1 Win=64256 Len=0 TSval=2539774988 TSecr=680222092	20	TCP: 20 → 39725 [FIN, ACK] Seq=225 Ack=1 Win=
326.228389524	39725	39725 → 20 [ACK] Seq=1 Ack=225 Win=65024 Len=0 TSval=680222092 TSecr=2539774988	20	TCP: 39725 → 20 [ACK] Seq=1 Ack=225 Win=650
326.228545678	39725	39725 → 20 [FIN, ACK] Seq=1 Ack=226 Win=65024 Len=0 TSval=680222092 TSecr=2539774988	20	TCP: 39725 → 20 [FIN, ACK] Seq=1 Ack=226 Win=
326.228887999	39725	20 → 39725 [ACK] Seq=226 Ack=2 Win=64256 Len=0 TSval=2539774989 TSecr=680222092	20	TCP: 20 → 39725 [ACK] Seq=226 Ack=2 Win=642
326.228890375	58496	Response: 226 Transfer complete.	21	FTP: Response: 226 Transfer complete.
326.229152337	58496	58496 → 21 [ACK] Seq=110 Ack=415 Win=64256 Len=0 TSval=680222093 TSecr=2539774988	21	TCP: 58496 → 21 [ACK] Seq=110 Ack=415 Win=6
329.321012050	58496	Request: QUIT	21	FTP: Request: QUIT
329.321508248	58496	Response: 221 Goodbye.	21	FTP: Response: 221 Goodbye.
329.321510626	58496	21 → 58496 [FIN, ACK] Seq=429 Ack=116 Win=65280 Len=0 TSval=2539778081 TSecr=680225185	21	TCP: 21 → 58496 [FIN, ACK] Seq=429 Ack=116 Wi
329.322500393	58496	58496 → 21 [FIN, ACK] Seq=116 Ack=430 Win=64256 Len=0 TSval=680225186 TSecr=253977808]	21	TCP: 58496 → 21 [FIN, ACK] Seq=116 Ack=430 Wi
329.322800467	58496	21 → 58496 [ACK] Seq=430 Ack=117 Win=65280 Len=0 TSval=2539778083 TSecr=680225186	21	TCP: 21 → 58496 [ACK] Seq=430 Ack=117 Win=6

Figura 1: Diagrama temporal da transferência do file1 para o Portatil1 através de FTP

# Protocolo de Transporte: TCP

**Início de conexão:** (Linhas 2 a 4) No início da conexão existem trocas de [SYN] -> [SYN,ACK] -> [ACK]

Transferência dos dados: A transferência de dados começa na linha 7 através da porta 20.

Fim de conexão: (Linhas 15 a 17) No fim da conexão existem trocas de [FIN,ACK] -> [FIN,ACK] -> [ACK]

Tipo de segmentos trocados: ACK, SYN e FIN

**Números de sequência:** No inicio da conexão os números de sequencia são 0 e 1. Durante a transferência são 1, 225 e 266. No fim da conexão são 429,116 e 430.



Figura 2: Diagrama temporal da transferência do file1 para o PC1 através de FTP

## Protocolo de Transporte: TCP

Início de conexão: (Linhas 2 a 4) No início da conexão existem trocas de [SYN] -> [SYN,ACK] -> [ACK]

Transferência dos dados: A transferência de dados começa na linha 7 atryés da porta 20.

Fim de conexão: (Linhas 15 a 19) No fim da conexão entre os end-systems podemos reparar que ouve a perda de um pacote que se encontra na linha 15. Esta perda originou uma retransmissão desse mesmo pacote para que o PC1 receba a mensagem de termino de conexão enviado pelo servidor. Uma vez que este [FIN,ACK] é recebido pelo PC1 o mesmo envia um [ACK] e um [FIN,ACK] a confirmar que recebeu a mensagem e que vai fechar a conexão. Por fim o PC1 recebe um [ACK], a confirmar que o servidor recebeu a mensagem de fecho.

# Tipo de segmentos trocados: ACK, SYN e FIN

**Números de sequência:** No inicio da conexão os números de sequencia são o 0 e o 1. Durante a transferência são 225,226 e 1. No fim da conexão são 429, 118,117 e 430.

- 4 Duarte Leitão, Hugo Ramos, Diogo Araújo
- 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.



Figura 3: Diagrama temporal da transferência do file1 para o Portatil1 através de TFTP

# Protocolo de Transporte: UDP

**Início e fim de conexão:** O TFTP usa o UDP como camada de transporte. Este protocolo não é orientado à conexão.

Transferência dos dados: O dados são transferidos a partir da porta 37097.

Tipo de segmentos trocados: Não existe troca de segmentos.

Números de sequência: Não existem.



Figura 4: Diagrama temporal da transferência do file1 para o PC1 através de TFTP

# Protocolo de Transporte: UDP

**Início e fim de conexão:** O TFTP usa o UDP como camada de transporte. Este protocolo não é orientado à conexão.

Transferência dos dados: O dados são transferidos a partir da porta 56167.

Tipo de segmentos trocados: Não existe troca de segmentos.

Números de sequência: Não existem.

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;

Uso da camada de transporte: As aplicações que empregam o protocolo de transporte TCP, como FTP, HTTP e SFTP, fazem uso significativo da camada de transporte, enquanto a aplicação TFTP utiliza o UDP, resultando em um uso bastante limitado dessa camada.

**Eficiência:** Em termos de velocidade e menor overhead, as aplicações que usam o protocolo de transporte UDP (TFTP) são consideradas mais eficientes. Por outro lado, em termos de integridade de dados, as aplicações que usam o protocolo de transporte TCP (FTP, HTTP, SFTP) são consideradas mais eficientes.

Complexidade: A aplicação TFTP é a menos complexa pois não necessita de estabelecer uma ligação para enviar os ficheiros pedidos. Comparada com TFTP, a aplicação HTTP é ligeiramente mais complexa, já que estabelece uma ligação para enviar os ficheiros. A aplicação SFTP é ainda mais complexa que as anteriores, visto que necessita de estabelecer uma ligação para pedidos e transferências. A aplicação FTP é a mais complexa das 4, pois é estabelecida uma ligação para pedidos e uma outra ligação para transferências.

**Segurança:** As aplicações HTTP, TFTP e FTP não fazem qualquer tipo de encriptação da mensagem dos pacote. Na aplicação SFTP é feita encriptação das mensagens dos pacotes, contribuindo assim para uma melhor segurança dos dados transferidos.

#### Parte II

5 Com base no trabalho realizado, tanto na parte I como na parte II, identifique para cada aplicação executada, qual o protocolo de aplicação, o protocolo de transporte, porta de atendimento e overhead de transporte.

	_	Protocolo de trans- porte (se aplicável)	mento (se aplicá-	Overhead de transporte em bytes (se aplicável)
wget, lynx ou via browser	http	tcp	80	54
ssh, sftp	-	tcp/tls	443	14
ftp	ftp	tcp	21	20
Tftp	tftp	udp	69	34
telnet	telnet	tcp	23	20
nslookup ou dig	dns	udp	53	14
Ping	-	udp	53	14
Traceroute	-	udp	53	14
Discord	http	tcp	80	-
Skype	irc	tcp	6667	66
CS:GO	-	udp	27015	-

# 6

# Conclusão

Na realização deste trabalho prático conseguimos compreender melhor o funcionamento dos vários protocolos de transporte através de capturas de pacotes com recurso ao software wireshark. Percebemos que o protocolo de transporte TCP é orientado à conexão enquanto o UDP é usado para pedidos e transferência de dados. Conseguimos ainda perceber que o overhead e o tempo de transmissão do TCP são maiores que os do UDP. Isto deve-se ao facto de existir deteção e correção de erros ligados à transmissão de pacotes no TCP, algo que não acontece com o UDP. Na análise das capturas, conseguimos ainda averiguar quando ocorrem as diferentes fases das transferências de dados (inicio de conexão, fim de conexão e transferência de dados).