



UNIVERSIDADE DO MINHO

LICENCIATURA EM ENGENHARIA INFORMÁTICA

Comunicações por Computador - TP1  
Grupo 67

Eduardo Pereira (A94881)      Gonçalo Vale (A96923)  
Pedro Oliveira (A95076)

Ano Letivo 2023/2024



# Questões e Respostas

## Parte I

**Questão 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.

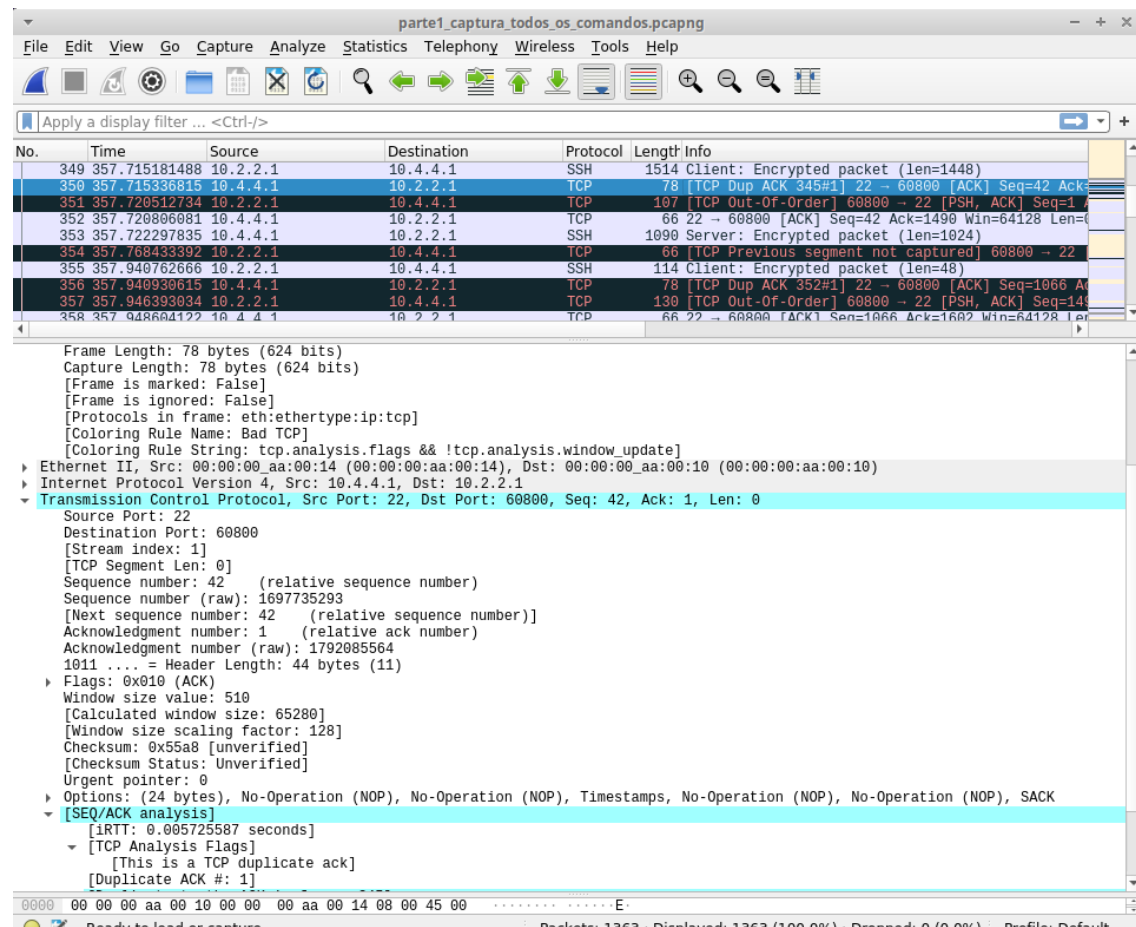


Figura 1: Exemplo de um pacote duplicado

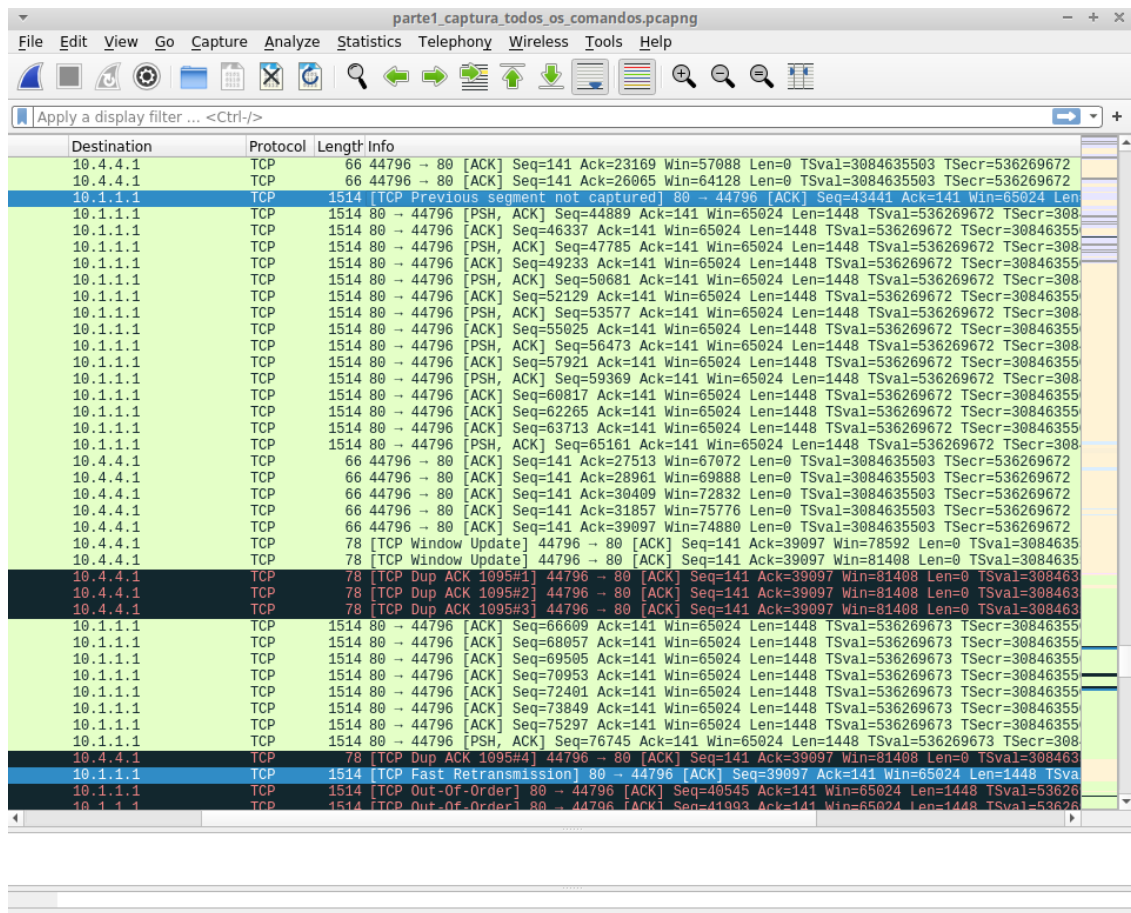


Figura 2: Exemplo de um pacote perdido

As perdas e duplicações de pacotes, que decorrem na **camada de transporte**, afetam o desempenho das aplicações ao reduzir a velocidade e a taxa de transferência da conexão, visto que, quando cada uma destas situações ocorre, a rede é obrigada a resolver esses casos, pedindo a retransmissão no caso da perda e ignorando os pacotes no caso de duplicação. Para diminuir a ocorrência destes casos utilizam-se protocolos de transporte, como o **TCP** e o **UDP**.

**Questão 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.

Utilizando o filtro "ftp-data" no *wireshark*, obteve-se o seguinte diagrama temporal:

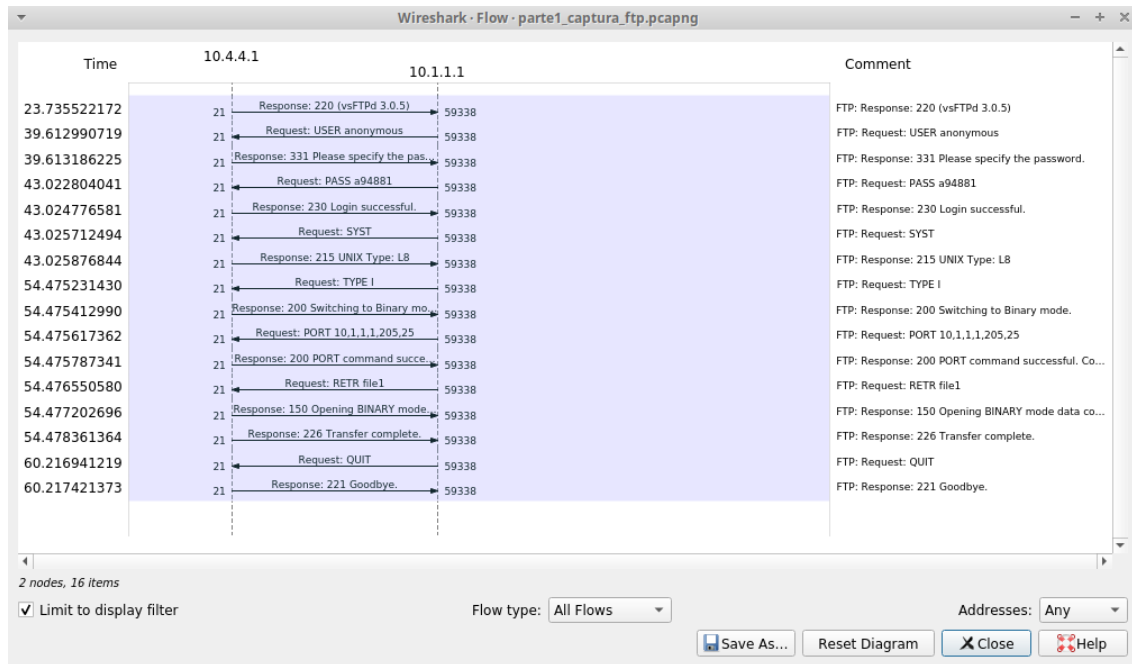


Figura 3: Diagrama temporal para a transferência de file1 por FTP

A fase de início de conexão dá-se ao instante 23.735522172. A fase de transferência de dados dá-se entre os instantes 54.476550580 e 54.478361364. A fase de fim de conexão dá-se entre os instantes 60.216941219 e 60.217421373. Observando as *flags* e os números de sequência no separador *Transmission Control Protocol* dentro de cada pacote, podemos ver que todos os packotes são do tipo **Acknowledgement**, sendo os números de sequência, ordenados do pacote de cima para o pacote de baixo na figura 3: 1, 1, 21, 17, 55, 30, 78, 36, 97, 44, 128, 66, 179, 243, 78, 267.

**Questão 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.

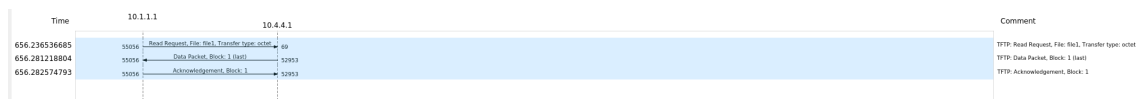


Figura 4: Diagrama temporal para a transferência de file1 por TFTP para o Portatil1

Tal como é possível verificar na figura, que se trata de uma análise da transferência via TFTP do file1 a partir do Servidor1 para o Portatil1, existe:

- Fase de início de conexão (primeiro packet)(instante 656.236536685), que inicializa a conexão e faz o pedido da transferência de dados.

- Fase de transferência de dados (segundo packet)(instante 656.281218804), uma vez que este se trata de um data packet, observável nos comentários.
- Fase de fim de conexão (terceiro packet)(instante 656.282574793), sendo este um acknowledgement que sinaliza a receção do packet.

Os dados são transferidos num block number 1. No entanto, uma vez que TFTP se trata de um protocolo UDP, não existem números de sequência.

**Questão 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;**

Para este trabalho prático foram usadas as seguintes aplicações SFTP, FTP, TFTP e HTTP.

Para começar, analisamos os pacotes transferidos através do SFTP, e pudemos ver que este utiliza o TCP como protocolo de transporte. Este utiliza SSH, logo permite uma segurança boa devido à encriptação dos dados. Apesar disso, diminui a eficiência na transferência, devido ao overhead. Comparativamente ao FTP, é menos eficiente devido à segurança adicional. Falando de complexidade, apresenta um grau elevado desta, já que é capaz de acessar, transferir e gerir dados, incluindo a encriptação dos mesmos.

Quanto ao FTP, tal como o SFTP, utiliza TCP como protocolo de transporte. Este é bastante fiável, contudo, para tal, devido ao *overhead* causado pelo envio de pacotes para confirmação de transferência, a sua eficiência é reduzida. Apesar disso, este é menos seguro que o SFTP, visto que FTP não apresenta qualquer segurança adicional, como encriptação. Em termos de complexidade, este é também um tanto quanto complexo, pois, para além das funcionalidades base de transferência de dados, tem a capacidade adicional de transferir dados em paralelo.

O TFTP utiliza UDP como protocolo de transporte, logo não utiliza qualquer tipo de *acknowledgements*. Assim sendo, a sua eficiência de transferência é superior devido ao baixo *overhead*. Mesmo havendo a possibilidade de existir um envio repetido do mesmo pacote várias vezes, o facto de não necessitar de esperar confirmações faz com que continue a ganhar velocidade comparativamente a FTP e SFTP. Por outro lado, não apresenta qualquer tipo de segurança ou autenticação, devido ao uso deste protocolo. Devido à sua falta de segurança e de funcionalidades, a sua complexidade é bastante reduzida comparado aos anteriores.

Por fim, o HTTP utiliza o TCP, logo possui autenticação, dando alguma segurança. Porém, este não oferece encriptação dos dados, o que o torna menos seguro. Como este é capaz de realizar vários pedidos ao mesmo tempo para uma ligação TCP sem a necessidade de aguardar por uma resposta (através de uma técnica de pipeline), é o mais eficiente. Para finalizar, quanto à sua complexidade, é o mais complexo entre as quatro aplicações, devido à grande variedade de funcionalidades que oferece.

## Parte II

**Questão 1.** 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 Aplicação	Protocolo de Transporte	Porta	Overhead de Transporte
Wget	HTTP	TCP	80	20 bytes
SFTP	SSH	TCP	22	32 bytes
FTP	FTP	TCP	21	32 bytes
TFTP	TFTP	UDP	69	8 bytes
telnet	TELNET	TCP	23	20 bytes
nslookup	DNS	UDP	53	8 bytes
Ping	-	ICMP	-	-
Traceroute	-	UDP	33434	8 bytes

Tabela 1: Tabela informativa sobre as aplicações