

TP1: Protocolos da Camada de Transporte (Resolução)

Grupo PL59

Alberto Campinho Faria
Bernardo Manuel Ribeiro Marques Soares Silva
Nuno Filipe Maranhão dos Reis

University of Minho, Department of Informatics, 4710-057 Braga, Portugal
e-mail: {a79077,a77230,a77310}@alunos.uminho.pt

Introdução

Este documento apresenta uma resolução do primeiro trabalho prático desenvolvido no âmbito da Unidade Curricular de Comunicações por Computador do curso de Mestrado Integrado em Engenharia Informática, no ano letivo de 2018/2019, da Universidade do Minho.

O trabalho em questão foca o modo de utilização dos protocolos de transporte TCP e UDP por várias aplicações. A secção seguinte apresta a resolução às várias questões do enunciado. A última secção conclui o relatório.

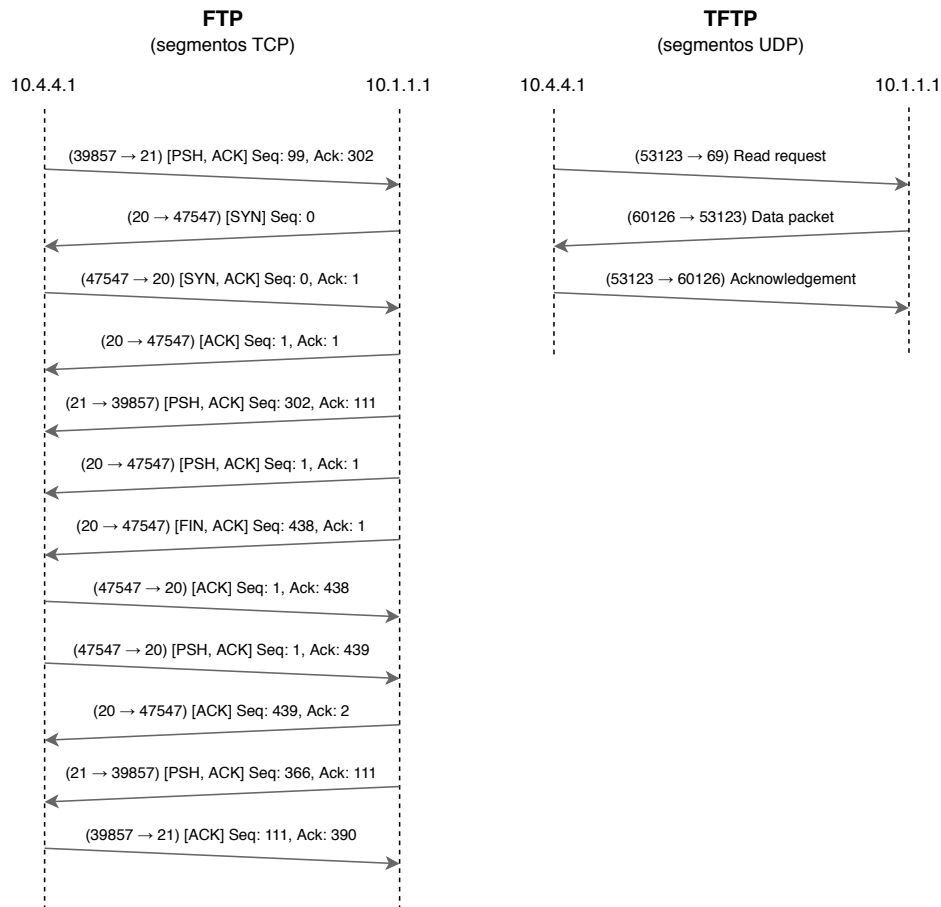
Questões e Respostas

Questão 1. *Inclua no relatório uma tabela em que identifique, para cada comando executado, qual o protocolo de aplicação, o protocolo de transporte, porta de atendimento e overhead de transporte.*

A tabela em questão é apresentada de seguida. Note-se que o comando `ping` utiliza o protocolo ICMP, o qual opera na camada de rede. Como tal, nenhuma das colunas correspondentes na tabela se aplica. De forma similar, o comando `traceroute` não utiliza um protocolo de aplicação específico.

| Comando usado | Protocolo de aplicação | Protocolo de transporte | Porta de atendimento | Overhead de transporte |
|---------------------------|------------------------|-------------------------|----------------------|------------------------|
| <code>ping</code> | — | — | — | — |
| <code>traceroute</code> | — | UDP | 33440 – 33455 | 8 bytes |
| <code>telnet</code> | DNS | TCP | 53 (Telnet) | 20 bytes |
| <code>ftp</code> | FTP | TCP | 21 (FTP) | 20 bytes |
| <code>tftp</code> | TFTP | UDP | 69 (TFTP) | 8 bytes |
| <code>browser/http</code> | HTTP | TCP | 80 (HTTP) | 20 bytes |
| <code>nslookup</code> | DNS | UDP | 53 (DNS) | 8 bytes |
| <code>ssh</code> | SSH | TCP | 22 (SSH) | 20 bytes |
| <code>sftp</code> | SFTP | TCP | 22 (SSH) | 32 bytes |

Questão 2. Uma representação num diagrama temporal das transferências da file1 por FTP e TFTP respectivamente. Se for caso disso, identifique as fases de estabelecimento de conexão, transferência de dados e fim de conexão. Identifica também claramente os tipos de segmentos trocados e os números de sequência usados quer nos dados como nas confirmações.



Questão 3. Com base nas experiências realizadas, distinga e compare sucintamente as quatro aplicações de transferência de ficheiros que usou nos seguintes pontos: (i) uso da camada de transporte; (ii) eficiência na transferência; (iii) complexidade; (iv) segurança.

Na realização do trabalho foram usadas várias versões do *File Transfer Protocol* no âmbito de registar as diferenças entre estas. As versões usadas foram TFTP (*Trivial File Transfer Protocol*), SFTP (*Secure File Transfer Protocol*) e FTP (*File Transfer Protocol*). Foi também usado o HTTP (*HyperText Transfer Protocol*) que segue um protocolo necessariamente diferente dos anteriores.

- **Uso da camada de transporte.** Foi verificado que, tal como o nome indica apenas TFTP, pela sua tag (*Trivial*), usa UDP como o seu protocolo de transporte. Isto deve-se ao facto deste protocolo de transferência ser indicado para ficheiros ditos de triviais e cuja segurança extra dada pelo TCP não é necessária. FTP, SFTP e HTTP usam todos TCP como o seu protocolo de transporte.
- **Eficiência na transferência.** Em termos de eficiência, o TFTP é o mais rápido de todos os protocolos de transferência devido ao seu uso de UDP em vez de TCP. TFTP também é, por defeito, indicado para a transferência de ficheiros mínimos cuja importância é insignificante. De seguida, temos FTP que pela sua simplicidade é mais rápido

do que SFTP e HTTP. SFTP é mais rápido que HTTP, no entanto é o mais lento de todos os *File Transfer Protocols* devido à segurança adicional presente na elaboração do seu protocolo, no entanto consegue ser mais rápido que HTTP porque este é mais adequado para a transferência de tamanhos elevados de diferentes dados ao mesmo tempo.

- **Complexidade.** Em termos de complexidade, olhando só para o *overhead* de transporte e não para outra complexidade inerente vinda da aplicação em si (maior *overhead* significa, normalmente, maior complexidade devido à existência de controlos), concluímos que o mais complexo é SFTP por juntar as características de SSH ao FTP e, além disso, por usar TCP. Sendo os outros todos iguais em complexidade exceto TFTP, concluímos que este é o menos complexo sendo o seu *overhead* o menor.
- **Segurança.** Todos os protocolos de transferência apresentam uma discrepância em termos de segurança entre si. TFTP não apresenta nenhum controlo de erros ou outro tipo de segurança, indo ao extremo de usar UDP que é *unreliable* e *connectionless* o que torna qualquer erro impossível de corrigir ou detectar. Apresenta também uma total falta no controlo de acesso o que permitirá a exploração nefária por atores maliciosos. De seguida temos FTP e HTTP, que por si só usam autenticação básica e vulnerável. No entanto, existe controlo de erros e controlo de congestão e fluxo. SFTP utiliza SSH, um sistema de autenticação e transmissão confidencial extremamente seguro.

Questão 4. *As características das ligações de rede têm uma enorme influência nos níveis de Transporte e de Aplicação. Discuta, relacionando a resposta com as experiências realizadas, as influências das situações de perda ou duplicação de pacotes IP no desempenho global de Aplicações fiáveis (se possível, relacionando com alguns dos mecanismos de transporte envolvidos).*

A possibilidade e a frequência de ocorrência de situações de perda ou duplicação de pacotes afeta negativamente a *performance* de comunicação ao nível aplicacional.

Por um lado, a perda de pacotes (ou deteção de erros na transmissão) resulta na necessidade de retransmissão dos mesmos. Mesmo não se perdendo um pacote, existe ainda *overhead* introduzido pela lógica de deteção de perda dos mesmos, a qual pode requerer até o envio de informação adicional, contribuindo para o aumento da utilização da rede e consumindo recursos.

Por outro lado, a possibilidade de duplicação de pacotes requer a existência de lógica adicional de deteção desse fenómeno. Note-se que este pode ser consequência da possibilidade de perda de pacotes — pacotes duplicados podem ser o resultado de uma tentativa de retransmissão de um pacote que o transmissor considerou perdido.

Conclusões

Com este trabalho prático, estudou-se o modo de utilização dos protocolos de transporte TCP e UDP por parte de várias aplicações. Foi prestada especial atenção a aplicações de transferência de ficheiros, tendo-se analisado as suas formas de utilização da camada de transporte e comparadas essas aplicações com base em várias características.

Todas as questões propostas pelo enunciado do trabalho prático em questão foram respondidas.