

Serviço de transferência rápida e fiável de dados sobre UDP

Relatório do trabalho prático

Grupo 9

Joana Cruz(A76270)
Etienne Costa (A76089)
Hugo Moreira (A43148)

Maio 2019



Universidade do Minho
Escola de Engenharia

Comunicações por Computador
Mestrado Integrado em Engenharia Informática
Universidade do Minho

1 Introdução

Este projeto surge no âmbito da Unidade Curricular de Comunicações por Computador e tem como objetivo a implementação de um serviço de transferência rápida e fiável de dados sobre UDP. Sendo que a comunicação deverá ser feita em unidades de dados que caibam dentro de um datagrama UDP e que sejam enviados e recebidos via Sockets UDP. Este serviço de transferência é caracterizado por três fases durante uma conexão: início da conexão, transferência de dados e término da conexão.

2 Especificação do protocolo

Sendo um protocolo uma convenção que controla e possibilita uma conexão, comunicação e transferência de dados entre dois sistemas computacionais, dedicou-se esta secção para explicar o formato das mensagens protocolares bem com as suas interações.

2.1 Proposta do PDU

O serviço de transferência rápida e fiável de dados foi implementado, a nível aplicacional, com recurso ao UDP. Sendo assim houve a necessidade de definir um datagrama UDP que possui as seguintes características :

- **Sequence Number:** Valor utilizado para garantir a entrega ordenada e robustez durante a transferência.
- **Acknowledge:** Valor utilizado para confirmar a entrega de um segmento.
- **Flag Type:** Valores utilizados para indicar um estado particular da conexão ou para fornecer informação adicional. Portanto, podem ser usados para fins de solução de problemas ou para controlar como uma determinada conexão é tratada.

De seguida apresenta-se os possíveis valores para estas flags:

1. **SYN.**
2. **ACK.**
3. **PSH**
4. **SYN+ACK**
5. **JOE DOWN**
6. **JOE UP**
7. **EXIT**
8. **LIST**
9. **NOTHING**

- **Port Number:**

- **Window Size:** Indica em bytes a quantidade de informação que este pode enviar nos próximos pacotes, durante uma conexão.
- **Length Data:** Corresponde em bytes ao tamanho do pacote a ser enviado.
- **Checksum:** Valor utilizado para verificar a integridade de dados transmitidos.
- **File Data:** Corresponde em bytes a informação a ser transferida.

| PDU | | | | | | | |
|-----------------|-------------|-----------|-------------|-------------|-------------|----------|-----------|
| Sequence Number | Acknowledge | Flag Type | Port Number | Window Size | Length Data | Checksum | File Data |

Figura 1: Protocol Data Unit

2.2 Interação

O nosso sistema de transferência é caracterizado por três fases distintas durante uma conexão:

1. **Início da conexão** Para o início de uma conexão, o nosso sistema foi baseado no 3-Way Handshake do TCP. Sendo que esta conexão procede do seguinte modo:

| Início da Conexão |
|--|
| Host A sends a SYN chronize packet to Host B |
| Host B receives A's SYN |
| Host B sends a SYN chronize- ACK nowledgment |
| Host A receives B's SYN-ACK |
| Host A sends ACK nowledgment |
| Host B receives ACK |

Tabela 1: Início da conexão

2. **Transferência de Dados**
3. **Término da conexão** Para o término de uma conexão, o nosso sistema foi baseado no TCP, sendo que este término é um processo de quatro fases, em que cada sistema computacional é responsável pelo encerramento do seu lado da ligação.

| Término da Conexão |
|--|
| Host A sends a FIN ished packet to Host B |
| Host B receives A's FIN ished |
| Host B sends a ACK nowledgment |
| Host A receives ACK nowledgment |
| Host B sends FIN ished packet to Host A |
| Host A receives FIN ished |
| Host A sends a ACK nowledgment |
| Host B receives A's ACK nowledgment |

Tabela 2: Início da conexão

3 Implementação

3.1 Arquitetura da solução

A arquitetura da solução implementada encontra-se descrita na figura apresentada abaixo:

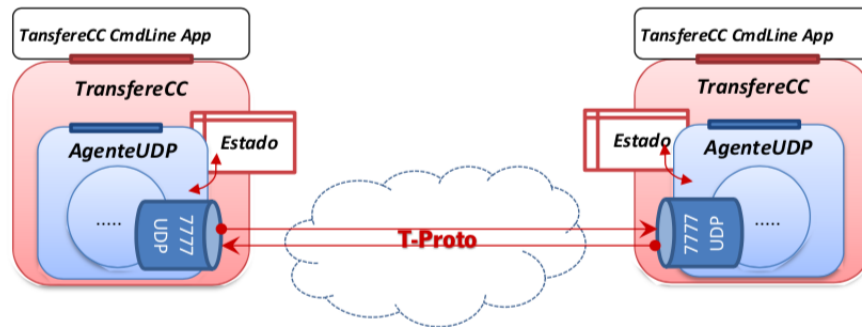


Figura 2: Arquitetura da solução implementada

3.2 Estrutura de dados

Para a implementação destes conceitos, foi utilizada a linguagem de programação orientada a objetivos *Java*, e definidas as seguintes classes:

3.3 Métodos

3.4 Parâmetros

3.5 Bibliotecas de suporte

4 Testes e Resultados

5 Conclusões e trabalho futuro