



Data Link Protocol

Computer Networks
Bachelors in Informatics and Computing Engineering

3LEIC03_G3

Tiago Rodrigues up201907021@fe.up.pt
Mário Travassos up201905871@fe.up.pt

December 8, 2021

Summary

This report will cover the first work proposed for the Computer Networks Curricular Unit, with the objective of creating a small application that could transfer data through two computers asynchronously, through a serial port.

The application is capable of transferring files whilst maintaining their integrity, and detect errors in transmission, resolving them if possible.

Introduction

This report is the result of an examination to the practical component, which was the development of a data transfer protocol. A serial port was used to transfer the files in an asynchronous fashion.

The report is organized as follows:

1. Architecture- Functional blocks and interfaces
2. Code Structure - APIs, main code structures and their relation with the architecture
3. Main use cases - Identification and Call Stack Sequence
4. Data link Protocol - Main functional aspects and implementation strategy
5. Application Protocol - Main functional aspects and implementation strategy
6. Validation - Description of the tests conducted
7. Efficiency - Statistical characterization of efficiency, against a Stop&Wait protocol
8. Conclusion - Summary of the above descriptions, reflection on the learning objectives

1 Architecture

The application consists of two main layers, one to interact with the file to be sent and another to interact with the hardware. They are the Application Layer, and the Data-Link Layer, and they are detailed below.

1.1 Application Layer

This layer can be found in the **rcom-ftp.c** file, and it encompasses everything related to interaction with the files, be it opening, closing, reading and writing to and from it. Besides that, this is the layer through which the user interacts with the application.

1.2 Data-Link Layer

This layer can be found in the **ll.c** file and it is responsible for ensuring a smooth data transmission over the hardware, including opening, closing, writing and reading from the serial port, with the help of the auxiliary functions present in **config.c**, **read.c**, **send.c**, **state.c** and **utils.c**.

2 Code Structure

The code is divided into seven source code files, separated by responsibility (reading from or writing to the serial port), and by layer. Also, each of them has a corresponding header file. Finally, there is a dedicated header file that hosts several common constants.

2.1 Application - rcom-ftp.c

This module contains the entire application layer developed.

Main Functions

- main - Interacts with the user and passes the arguments given to the rest of the program.
- sendFile - Sends the file requested by the user.
- readFile - Retrieves the file sent by the user.

Main Data Structures

- fileData - Responsible for holding some metadata of the file.

2.2 Config - config.c

This module contains the functions required for setting up the serial port for proper file transferring.

Main Functions

- `set_config` - Sets up the initial configuration for the serial port.
- `reset_config` - Restores the serial port to its initial state.

2.3 Link Layer - ll.c

This module contains the interface for the Link Layer of the protocol.

Main Functions

- `llopen` - Opens the serial port from frame transmission.
- `llwrite` - Writes a frame to the serial port.
- `llread` - Reads a frame from the serial port and checks its integrity.
- `llclose` - Closes the serial port after communication ceases.

2.4 Reading - read.c

This module contains the functions responsible for reading from the serial port.

Main Functions

- `readSupervisionFrame` - Reads a supervision frame and checks if the information is received correctly.
- `readInformationMessage` - Reads an information message and saves the data, which includes the BCC, in a buffer.

2.5 Writing - send.c

This module contains the functions responsible for writing to the serial port.

Main Functions

- `writeSupervisionAndRetry` - Attempts to write a supervision Frame in 3 attempts. If it succeeds, it stops.
- `writeInformationAndRetry` - Attempts to write an information Frame in 3 attempts. If it succeeds, it stops.

2.6 State Management - state.c

This module contains the functions responsible for managing the state of the application.

Main Functions

- `handle_state` - Function responsible for managing the state of the application, according to the data received.

Main Data Structures

- `state_t` - Enumeration containing the possible states of the application.
- `state_machine` - Besides the state of the machine, holds some of the main pieces of information from each frame.

2.7 Utilities - utils.c

This module contains some auxiliary functions that help the others with their operations.

Main Functions

- `stuff_data` - This functions stuffs the data given.
- `unstuff_frame` - This functions unstuffs the frame given.

2.8 Constants - defines.h

This module contains some of the more meaningful constants used throughout the application, like Supervision Frame constants, Rejection Messages, among others.

- 3 Main use cases
- 4 The Data link Layer
- 5 The Application Layer
- 6 Validation
- 7 Efficiency
- 8 Conclusion