A black sign with white text

Description automatically generated

**Protocolo de Ligação de Dados**

(1º Trabalho Laboratorial)

Redes de Computadores **·** L.EIC025

2023/2024 **·** 1º Semestre

Turma: 3LEIC08

João Correia

up202005015

up201506074

Miguel Charchalis

# Sumário

# No âmbito da unidade curricular Redes de Computadores, realizou-se o 1º projeto que nos foi proposto. O projeto “Protocolo de Ligação de Dados” consiste em implementar a funcionalidade de transmissão e receção para permitir transferir um ficheiro armazenado no disco de um computador para um outro computador, estando ligados através de um cabo série.

# O nosso principal objetivo no projeto foi alcançado com êxito, uma vez que conseguimos desenvolver com sucesso um protocolo de conexão de dados.

# Introdução

# A elaboração deste relatório, tem o objetivo de explicar de uma forma mais detalhada o funcionamento do protocolo de ligação de dados para a transmissão e receção de ficheiros através de uma porta Série RS-232. A estrutura da documentação é a seguinte:

* **Arquitetura**  
  Demonstração dos blocos funcionais e interfaces.
* **Estrutura do código**  
  Representação das APIs, as estruturas de dados usadas, as funções e a sua interação com a arquitetura.
* **Casos de uso principais**Identificação dos principais casos de uso e sequências da chamada de funções.
* **Protocolo de ligação lógica**  
  Identificação dos principais aspetos funcionais e descrição da estratégia de implementação destes aspetos.
* **Protocolo de aplicação**  
  Identificação dos principais aspetos funcionais e descrição da estratégia de implementação destes aspetos.
* **Validação**  
  Descrição dos testes efetuados com apresentação quantificada dos resultados, se possível.
* **Eficiência do protocolo de ligação de dados**  
  Caracterização estatística da eficiência do protocolo.
* **Conclusões**  
  Síntese da informação apresentada nas secções anteriores e reflexão sobre os objetivos de aprendizagem alcançados.

# Arquitetura

# Este projeto está dividido em duas camadas: camada da ligação de dados (*link\_layer*) e a camada de aplicação (*application\_layer*). A camada de ligação de dados é responsável pela delimitação e numeração das tramas, pelo estabelecimento e terminação da ligação às portas série e por controlo de erros e fluxo. A camada de aplicação depende da camada de ligação de dados e é responsável por enviar e receber tramas e processar e transmitir pacotes de dados ou de controlo.

# A camada de aplicação está acima da camada de ligação de dados, então não conhece a estrutura interna do protocolo de ligação de dados, apenas conhece o serviço que desempenha. Esse serviço pode ser acedido a partir da interface do utilizador.

# A interface do utilizador permite que o utilizador escolha a porta série, a função desempenhada pelo sistema (emissor ou recetor) e o ficheiro que pretende transferir entre os dois computadores. Para tal, a execução do programa é realizada em 2 terminais, um em cada computador, sendo um o transmissor e o outro o recetor.

# Estrutura do código

# Camada de aplicação (application\_layer.c e application\_layer.h)

# A camada de aplicação é constituída pela função ‘*application\_layer’* que recebe alguns dados predefinidos e outros introduzidos pelo utilizador. Esses dados são usados para transferir e processar pacotes de dados e de controlo. A camada de ligação dos dados é usada nesta função para comunicar e transferir os ficheiros de acordo com os parâmetros passados. A função ‘tlv’ é uma função adicional à *application\_layer.*

# A computer code with green text Description automatically generated with medium confidence

# Camada de ligação de dados (link\_layer.c e link\_layer.h)

# Na camada de ligação de dados utilizaram-se três estruturas de dados auxiliares: *LinkLayer*, onde são caracterizados os parâmetros associados à transferência dos dados, *LinkLayerRole*, que identifica se o computador está a exercer o papel de recetor ou emissor, *LinkLayerState*, que identifica o estado da leitura e receção das tramas de informação.

# A screenshot of a computer code Description automatically generated A screenshot of a computer code Description automatically generated

As principais funções desta camada são a seguintes:

* ***llopen -*** Abre uma conexão usando os parâmetros da *port* definida na estrutura.
* ***llwrite -*** Envia informação num *buffer*, com um determinado tamanho, e retorna o número de caracteres escritos.
* ***llread -*** Recebe a informação num pacote e retorna o número de caracteres lidos.
* ***llclose -*** Fecha a conexão previamente aberta.

# A screenshot of a computer program Description automatically generated

# Casos de uso principais

Este trabalho explora diversos cenários de aplicação, englobando situações como a transferência de ficheiros entre dois computadores e a implementação de uma interface que permite ao transmissor selecionar o ficheiro a ser enviado. Para executar o programa, é necessário que o recetor utilize o comando make run\_rx, enquanto o transmissor deve empregar make run\_tx.

No contexto do transmissor, a conexão entre os computadores é iniciada através do llopen. Em seguida, o pacote de controlo start é enviado, dando início à execução da função llwrite onde acontece o processo de payload seguido pelo stuffing. Posteriormente, o pacote de controle end é transmitido, culminando no encerramento da conexão entre os computadores por meio da função do llclose.

No que diz respeito ao recetor, o estabelecimento da conexão entre os computadores ocorre por meio do llopen. A recepção do pacote de controlo start é seguida pela obtenção dos pacotes de dados de tamanho de 512 bytes através da função llread. Em seguida, o pacote de controlo end é recebido, finalizando a conexão entre os computadores pela função llclose.

# Protocolo de ligação lógica

# 6.1- llopen

# 

# Esta função é responsável pelo estabelecimento da ligação entre os computadores. A função começa por abrir e configurar a porta série. De seguida, o emissor manda uma trama de supervisão SET (comando) e fica à espera que o recetor lhe responda com outra trama de supervisão. O recetor, recebendo o SET, responde com a trama de supervisão UA (resposta ao comando). Caso o emissor receba a resposta UA, a ligação foi bem estabelecida. A leitura dos pacotes, tal como nas restantes funções, é controlada com uma *state machine* para a verificação da sua integridade e controlo.

# 6.2- llwrite

# 

# Uma vez que a ligação está estabelecida, o emissor começa a enviar informação que será lida pelo recetor. Esse envio de informação é feito pela função *llwrite* e apenas o emissor é que a utiliza. A função recebe um pacote de controlo/dados e executa o processo de *byte stuffing* para evitar conflitos entre bytes. O pacote é transformado para uma trama de informação e envia-se para o recetor, esperando pela sua resposta. No caso de a tama ser rejeitada pelo recetor, é enviada novamente até ser aceite ou exceder o número de tentativas. O emissor aguarda até receber a resposta do recetor, sendo que pode ser rejeitada (REJ0/REJ1) ou aceite (RR0/RR1).

# 6.3- llread

# 

# O recetor procede então à leitura das tramas de informação a partir da função *llread*. A função faz o *destuffing* do campo de dados e procede à leitura. Se os campos BCC1 e

# BCC2 forem válidos, não ocorreram erros na transmissão. No caso de haver erros, a função vai responder ao emissor com uma trama de supervisão REJ. No caso da trama ser aceite, a resposta ao emissor será com um RR.

# 6.4- llclose

# 

A função ***llclose*** é responsável por terminar a ligação através da porta série. O comportamento desta função varia caso se trate de um recetor ou transmissor. Assim, no caso do recetor, espera pela trama de supervisão DISC e quando a recebe, envia um DISC e espera por uma resposta UA final. No caso do emissor, é enviada uma trama de supervisão DISC, recebe outro DISC do recetor e, por fim, envia um UA. A transmissão é depois fechada através da porta série.

# Protocolo de aplicação

A camada de mais alto nível é a de aplicação que é responsável pela leitura e escrita do ficheiro a enviar ou receber e pelo envio e receção dos pacotes de dados e controle. A ***application\_layer*** está dividida em dois blocos, e escolhe qual deles executa, dependendo se se trata do emissor ou do recetor. Ainda antes de executar um desses blocos, é chamada a função ***llopen***, para estabelecer a conexão.

# Validação

# Foram efetuados os seguintes testes para garantir a integridade das transmissões e confiabilidade do programa:

# - Envio de um ficheiro; - Interrupção da ligação da porta série enquanto se envia o ficheiro; -Introduzir ruído na porta série enquanto se envia o ficheiro;

# Os testes foram efetuados na presença do docente quando foi efetuada a apresentação.

# Eficiência do protocolo da ligação de dados

# Neste projeto, não foram efetuadas estatísticas relativamente à eficiência, com as diferentes variáveis.

# Conclusão

# O objetivo principal deste projeto, que consiste no envio de um ficheiro entre dois computadores através do uso da porta série RS-232, foi concluído com sucesso. Este trabalho contribuiu principalmente para uma melhor compreensão acerca da independência de camadas em sistemas e a forma como estes estão organizados. A realização do projeto também fez com que aprendêssemos conceitos novos tais como o stuffing, framing e deu-nos novas formas de implementação de código.

# Anexo I – link\_layer.h

# 

# 

# Anexo II – application\_layer.h

# 

# Anexo III – link\_layer.c

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# Anexo IV – application\_layer.c

# 

# 

# 

# 

# 

# 

# 

# 