Faculdade de Engenharia da Universidade do Porto

Data Link Layer Protocol



Projeto 1 – Grupo F

Redes de computadores

Maria Pimentel Pestana Cardoso Ribeiro – up201703844

Miguel…. Silva – up20

Índice

[Sumário 3](#_Toc100651429)

[Introdução 3](#_Toc100651430)

[Arquitetura 3](#_Toc100651431)

[Estrutura do código 3](#_Toc100651432)

[Casos de uso principais 3](#_Toc100651433)

[Protocolo de ligação lógica 4](#_Toc100651434)

[Protocolo de aplicação 4](#_Toc100651435)

[Validação 4](#_Toc100651436)

[Elementos de valorização 4](#_Toc100651437)

[Conclusões 4](#_Toc100651438)

# Sumário

No âmbito do primeiro trabalho laboratorial de Redes de Computadores foi realizada a transferência de dados de um computador para outro através de uma porta de série assíncrona.

Com este trabalho foi possível realizar uma transmissão de dados e a utilizar mecanismos de deteção e controlo de erros de modo a parar a transmissão se forem detetados erros ou realizar as ações necessárias.

# Introdução

Neste trabalho foi pedido para implementar um protocolo de ligação de dados através da porta de série utilizando o protocolo stop & wait de modo a detetar erros e anomalias. Desenvolvemos diversas funções de deteções de erros para que seja possível evitar falhas de receção de tramas.

# Arquitetura

(blocos funcionais e interfaces)

A ligação lógica é realizada pelas funções llopen() e llclose() que abrem e encerram esta ligação.

A função de abertura e configuração da ligação é realizada na primeira função, llopen(), onde inicialmente alteramos as configurações da porta série pretendida. Esta função pode ser chamada de duas formas: “TRANSMITTER” e “RECEIVER”. Aqui também realizamos a configuração do VMIN (=0 para não desbloquear quando lê um numero de bytes na porta série) e de VTIME= 1 para podermos utilizar o timeout se não recebermos tramas ou informar que lemos o que foi transmitido pela porta série.

Implementamos também as funções llread() e llwrite() que realizam a leitura e a escrita da informação. No caso da primeira inicialmente descodifica a informação presente nas tramas e envia uma confirmação dessa receção ao contrário da llwrite() que escreve primeiro e só depois a codifica para poder enviar a trama.

Todas as funções descritas anteriormente utilizam a Struct linkLayer, que nos permite receber valores cruciais definidos pelo main.c, tais como a porta de série, número de tentativas, timeout e se é “TRANSMITTER” ou “RECEIVER”.

A camada de aplicação é responsável pela inicialização do programa???

# 

# Estrutura do código

(APIs, principais estruturas de dados, principais funções e sua relação com a arquitectura)

# Casos de uso principais

(identificação; sequências de chamada de funções

Os casos de uso principais são utilizados na inserção de dados pelo utilizador e transmissão de informação que diferem no caso do emissor e do recetor.

No caso do emissor:

Implementamos uma máquina de estados que inicialmente reinicia um temporizador para podermos utilizar timeout e verifica através de flags se esta informação foi corretamente enviada

No caso do recetor:

Realizamos …

# Protocolo de ligação lógica

(identificação dos principais aspetos funcionais; descrição da estratégia de implementação destes aspetos com apresentação de extratos de código)

A camada de ligação lógica é encarregue da comunicação e confirmação das flags BCC\_T e BBC\_R de modo que não haja erros.

Envia e recebe as tramas: SET, UA de modo a estabelecer e encerrar uma ligação

(continuar)

# Protocolo de aplicação

(identificação dos principais aspetos funcionais; descrição da estratégia de implementação destes aspetos com apresentação de extratos de código)

# Validação

(descrição dos testes efetuados com apresentação quantificada dos resultados, se possível)

# Elementos de valorização

(identificação dos elementos de valorização implementados; descrição da estratégia de implementação com apresentação de pequenos extratos de código)

# Conclusões

(síntese da informação apresentada nas secções anteriores; reflexão sobre os objetivos de aprendizagem alcançados)

# Anexos

- Anexo I - Código fonte  
- Outros anexos, se desejável