

Redes de Computadores

*1º Trabalho Laboratorial*

*´7 de novembro de 2017*

Índice

|  |  |
| --- | --- |
| **Sumário** | 0 |
| **Introdução** | 2 |
| **Arquitetura** | 2 |
| **Estrutura do código** | 2 |
| **Casos de uso principais** | 2 |
| **Protocolo de ligação lógica** | 2 |
| Identificação dos principais aspetos funcionais | 2 |
| Descrição da estratégia | 2 |
| **Protocolo de aplicação** | 2 |
| **Validação** | 2 |
| **Eficiência do protocolo de ligação de dados** | 2 |
| **Conclusões** | 2 |
| **Anexo I** | 2 |

Sumário

No âmbito do primeiro trabalho laboratorial de Redes de Computadores, este relatório tem como objetivo realizar uma análise estatística ao comportamento do protocolo implementado para a transmissão e receção de ficheiros através da porta série.

Assim, o trabalho foi realizado com sucesso, uma vez que todos objetivos propostos foram cumpridos, e estando desta maneira o protocolo capaz de assegurar uma transmissão correta dos dados, mesmo que ocorram alterações nos pacotes durante a mesma.

(dois parágrafos: um sobre o contexto do trabalho; outro sobre as principais conclusões do relatório, falar sober a )

Camada de Aplicação (Application Layer), a Camada de Ligação de Dados (Data Link Layer) e a Camada Física (Physical Layer). A proposta de trabalho baseou-se em realizar uma transferência de ficheiros entre dois computadores

Introdução

Arquitetura e Estrutura do código

O trabalho está organizado em duas principais camadas lógicas que permitem uma correta organização e funcionamento do protocolo, tornando-o modular.

Camada de Ligação (Data Link Layer)

A camada de Ligação é a camada de software “mais baixa” e é responsável pela comunicação e interpretação dos resultados provenientes da porta de série.

Deste modo, esta camada contém todas as funções relativas ao estabelecimento da ligação entre os terminais, escrita e leitura da informação utilizando a porta série, e respetiva análise em relação à sua coesão, e fecho da ligação no fim da comunicação.

FRASE E ESTRUTUA DA STRUCT

Camada de Aplicação (Application Layer)

A camada de Aplicação é a camada “mais alta” e é responsável pela inicialização do programa (função main) e pela criação dos pacotes que contém os dados a enviar, tanto as tramas de start e end como as tramas de informação – tramas I.

Assim, esta camada contém, entre outras as funções, a função de leitura e a função de criação/escrita do ficheiro a transmitir, no caso de o programa ser executado no modo de escritor ou no modo de leitor respetivamente.

Para melhor uma melhor estruturação esta camada contém duas estruturas de dados (structs) essenciais para guardar informação importante relativa ao ficheiro de leitura/escrita e dados a enviar para a camada de ligação para a comunicação com a porta de série.

ESTRUTUA DA STRUCT

Casos de usos principais

Protocolo de ligação logica

A camada de ligação é uma das camadas implementadas neste projeto. Esta camada é responsável pela comunicação e coesão da mesma através de verificações. Assim esta camada tem as seguintes funcionalidades:

* Estabelecer e terminar uma ligação através da porta de série (Envio e receção das tramas SET e UA).
* Escrever e ler mensagens através da porta série (Envio e receção das tramas RR, REJ e I)
* Adicionar e remover os cabeçalhos necessários para o envio das tramas S e I referentes ao envio dos pacotes de informação, e respetivo stuff/destuff.
* Analisar os cabeçalhos das mensagens recebidas enviando a respetiva trama de resposta.

Deste modo as principais funções desta camada são as seguintes:

LLOPEN

*int LLOPEN(char\* port, char\* mode)*

A função llopen é responsável por estabelecer a ligação através da porta de série. Como tal, esta função ao ser invocada abre e configura a porta de série com os respetivos parâmetros necessários, incluindo o baudrate, e posteriormente é enviado pelo emissor o comando SET e como resposta o recetor envia UA (acknowledge). Se tudo correr bem a ligação fica estabelecida, e a função retorna o valor do descritor da ligacao através da porta de série. Caso contrário, se UA não for enviado a trama SET é reenviada passados timeout segundos ate um máximo de num\_retries tentativas, sendo que timeout e num\_retries são duas variáveis cujos valores são recebidos como argumentos na chamada da função do emissor. Caso seja atingido o número máximo de tentativas a função retorna -1 indicando que a ligação não foi estabelecida. Por outro lado, caso exista resposta por parte do recetor mas esta não seja afirmativa o emissor reenvia a trama SET até que a resposta seja afirmativa.

LLWRITE

*int LLWRITE(int fd, unsigned char\* msg, int\* length);*

A função llwrite é responsável pelo envio dos pacotes de Informação. Assim esta função recebe como argumento, entre outros, a mensagem a ser enviada (pacotes de controlo+dados) e o respetivo tamanho. Deste modo, com recurso à função create\_package() cria a função com os cabeçalhos necessário ao envio da trama(flags, endereço, controlo(c1) e bcc1) e envia a mesma. O envio das tramas de informação é similar ao estabelecimento da ligação por parte do emissor, ou seja, a mensagem é enviada e fica-se a aguardar resposta sendo reenviada a mesma quando for atingido o tempo de timeout. Caso a resposta recebida seja o comando RR(C1 = 0x05/0x85), a mensagem foi recebida corretamente pode-se enviar o próximo pacote; caso a resposta seja o comando REJ(C1 = 0x01/0x81) o ultimo pacote é reenviado.

LLREAD