

UNIVERSIDADE DE BRASÍLIA
UNB

Gustavo Antônio Souza de Barros 18/0064487
Mayara Chew Marinho 18/0025210

Projeto Camada de Enlace

Simulador DougMay

TELECOMUNICAÇÃO E REDES 1
BRASÍLIA
2021

1 Introdução

A camada de enlace de dados é implementada na interface de rede, ou Adaptador de rede, sendo que o controlador desta camada se encontra no núcleo do adaptador. O objetivo desta camada é assegurar que os dados serão transferidos com segurança por um meio físico, ou seja, que eles serão corretamente recebidos, utilizando mecanismos de enquadramento, detecção e correção de dados.

O objetivo deste trabalho é implementar a camada de enlace de dados utilizando algoritmos de enquadramento de dados, detecção e correção de erros para garantir que a mensagem consiga ser transmitida com sucesso. Tendo em vista que a camada de enlace de dados utiliza alguns atributos da camada física, este projeto foi implementado por cima de funcionalidades do projeto anterior, sobre camada física.

Por meio desse projeto, foi possível visualizar e entender melhor o funcionamento de um sistema de enlace de dados e compreender de forma mais aprofundada o conhecimento acerca dos algoritmos utilizados ao longo do processo.

2 Implementação

Neste projeto, foi implementado um Sistema de Enlace de dados chamado 'DougMay', contendo algoritmos de enquadramento de dados (Contagem de Caracteres e Bytes de Flag com Inserção de Bytes); detecção e correção de erros (Cyclic Redundancy Codes, Bit de Paridade). Sendo possível escolher as alternativas através de uma simples 'interface' com o usuário (GUI) criada no terminal. Para cada uma das codificações citadas, é aplicado um processamento de sinais digitais específico, contendo vantagens ou desvantagens para cada tipo de aplicação.

Uma das dificuldades de transmissão de dados são os erros que podem ocorrer durante o processo, como, por exemplo, a mudança inesperada no valor dos bits e no tamanho do quadro de bits transmitido. Esse problema pode ser resolvido por meio dos métodos: Contagem de Caracteres e Inserção de Bytes.

O método de Contagem de Caracteres tem o objetivo de sinalizar a quantidade de caracteres que serão enviados na transmissão antes de enviar os dados em si. Para implementar este método, foi realizada uma contagem de bits presentes em cada quadro e esse valor foi inserido antes do quadro, dessa forma, assim que os dados chegassem no receptor, este valor é interpretado antes e utilizado para ler os dados a seguir, sinalizando o limite desta leitura.

O método Inserção de Bytes utiliza uma "flag", ou seja, um byte especial, para indicar o início e o fim de um quadro de bites, assim como exemplifica a Fig 1. Esta "flag" deve ser definida previamente pelo projetista e utilizada antes/depois do quadro. Assim, a transmissão tem a garantia de que o número correto de bites foi transmitido. Neste projeto a "flag" utilizada foi: .



Figure 1: Exemplo ilustrativo do método de Inserção de Bytes. Fonte: Aula de Camada de Enlace de Dados: Enquadramento, disciplina de Telecomunicação e Redes 1.

A taxa de erro durante a transmissão varia conforme o canal adotado para transmitir a informação, sendo que em enlaces sem fios as taxas são altas e em fibras ópticas são menores. Sendo assim, foram definidos alguns algoritmos para lidar com a detecção de erros durante a transmissão, tais como Bit de Paridade e Cyclic Redundancy Code (CRC).

O Método de Bit de Paridade calcula o número de bits de paridade e envia este valor depois do quadro. Uma limitação deste método é que ele detecta erros apenas de um único bit. Na implementação deste método, o canal transmissor envia o quadro seguido do bit de paridade e o canal receptor recalcula o bit de paridade e compara com o bit enviado. Assim, pode-se detectar se houve erro na transmissão.

O Método CRC utiliza uma operação de XOR do dado a ser transmitido com um polinômio escolhido previamente pelo projetista. A sobra desta operação é

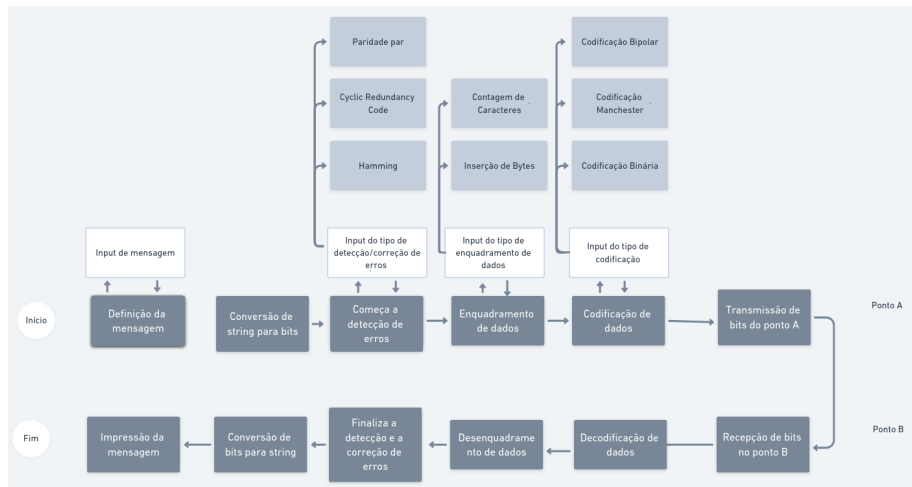


Figure 2: Fluxograma da metodologia aplicada ao projeto.

enviada ao canal receptor, que faz uma operação de XOR do valor recebido com o polinômio novamente. Caso esta operação seja exata, com resto zero, significa que não houve erros durante o processo de transmissão, caso o resto não seja zero, houve erros durante o processo.

O código de Hamming pode ser utilizado para detectar e corrigir erros durante a transmissão. Para isso, é criada a "palavra de código" contendo bits de dados e bits de verificação para identificar se houve erro, sendo todos os bits de potência de dois de verificação e os outros de dados. Os bits de verificação são definidos como 0 ou 1 para manter a paridade par na sequência. A "palavra de código" é enviada ao canal receptor e a paridade é calculada novamente, caso o resultado seja 0, significa que não ocorreu erro, caso ele seja 1, significa que ocorreu erro. Por meio deste último cálculo de paridade, é possível agrupar os resultados e gerar um valor que indica exatamente qual a ordem do bit transmitido com erro. Com essa informação, também existe a possibilidade de corrigir o bit errado, transformando-o de 0 para 1 e de 1 para 0. Este método não foi possível de ser finalizado ao longo desse projeto devido a dificuldades no algoritmo.

Conforme o fluxograma apresentado na Fig 2, a implementação do simulador 'DougMay' pode ser exemplificada em 9 etapas:

- Definição da mensagem: um input é utilizado para coletar a mensagem a ser transferida pelo simulador;
- Definição do tipo de codificação: um input é utilizado para o usuário escolher o tipo de codificação;
- Conversão de string para bites: a mensagem é convertida para bites;

- Detecção de erros: um input é utilizado para definir o algoritmo de detecção de erros.
- Enquadramento de dados: um input é utilizado para definir o tipo de enquadramento a ser aplicado.
- Codificação: é realizada a codificação escolhida;
- Transmissão de bits do ponto A: o ponto A transmite os bits para o ponto B;
- Recepção de bits no ponto B: o ponto B recebe os bits do ponto A;
- Decodificação: os bits são decodificados de acordo com a codificação escolhida anteriormente;
- Desenquadramento de dados de acordo com o algoritmo definido previamente.
- Finalização da detecção de erros e correção de erros de acordo com o algoritmo definido previamente.
- Conversão de bites para string: os bites recebidos são convertidos novamente em string;
- Impressão da mensagem: por meio do terminal, é impressa na tela a mensagem inicial que foi transmitida pelo simulador.

3 Membros

Para a realização do trabalho, foi definido que todos participariam da implementação do projeto. Logo, foram marcadas reuniões para codificar o programa de forma conjunta. Foi utilizada a ferramenta Visual Studio Code com a extensão Live Share para realizar esse trabalho. Também foi criado um repositório no GitHub para os integrantes terem acesso ao código mais recente sempre que necessário. Dessa forma, o programa foi implementado com a participação dos dois membros.

Depois do programa implementado, separaram-se as partes do relatório que cada um escreveria, a fim de agilizar esse processo. A Mayara ficou com a seção de Implementação e Gustavo com as outras. Durante a escrita, os dois revisaram o trabalho um do outro, ajudando no que fosse necessário.

4 Conclusão

O trabalho facilitou o entendimento da camada de enlace e possibilitou uma visão mais prática das codificações utilizadas. Durante o desenvolvimento encontramos algumas dificuldades. A primeira foi a implementação do algoritmo de detecção de erros Cyclic Redundancy Codes (CRC), demoramos bastante para entender e codificar esta parte e a segunda foi que não conseguimos implementar o código de Hamming, responsável por detectar e corrigir erros durante a transmissão de dados.

No início do desenvolvimento, tiramos um tempo para escrever a estrutura base disponibilizada no slide e utilizamos o primeiro projeto da disciplina para implementar a camada de enlace de dados por cima. Uma sugestão seria deixar esse código já disponibilizado para os alunos. Havia também alguns comentários errados e trechos de código que estavam comentados nesse código.