UNIVERSIDADE DE BRASÍLIA

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Simulação do funcionamento da camada de enlace por meio da implementação dos protocolos: enquadramento de dados, detecção de erros e correção de erros

Andressa de Araújo Pereira, 190024771

Pedro Brazil Suffert, 200049682

Wallace Ben Teng Lin Wu, 200028880

Introdução

O presente relatório busca descrever e prover uma visão geral da simulação do funcionamento da camada de enlace de dados por meio da implementação dos protocolos de enquadramento, detecção de erros e correção de erros para a disciplina de Teleinformática e Redes 1. Dessa forma, será implementado a simulação do funcionamento da camada de enlace por meio da implementação destes três protocolos mencionados anteriormente, e o simulador será desenvolvido na linguagem C + +.

A camada de enlace de dados que será implementada, é a segunda camada do modelo de referência em redes OSI (Open Systems Interconnection) que tem como suas principais funções a facilitação da transmissão de dados de forma eficiente e confiável, dividir o fluxo contínuo de bits recebidos da camada de rede em quadros de dados menores e transmitir esses quadros através do meio físico até o dispositivo de destino.

É importante enfatizar que o desenvolvimento do simulador foi dividido em três sub etapas, e cada uma delas aborda um conjunto específico dos protocolos. Além disso, o simulador da camada física desenvolvido no primeiro trabalho será utilizado como base, permitindo a integração e complementação da codificação da camada de enlace. Dessa forma, os protocolos de enquadramento, detecção de erros e correção de erros serão incorporados ao código existente, ampliando a funcionalidade do simulador e possibilitando uma simulação mais completa do funcionamento das camadas física e de enlace.

A primeira sub etapa consiste em adicionar o código do enquadramento de dados, que é uma técnica utilizada na camada de enlace para dividir um fluxo contínuo de bits em unidades chamadas de quadros. Essa técnica facilita a transmissão e recepção de dados e permite que o receptor identifique onde começa e termina cada quadro. Existem diversos métodos de enquadramento, no simulador serão implementados, os seguintes: de contagem de caracteres e o de inserção de bytes ou caracteres.

A segunda sub etapa abordará o protocolo de detecção de erros, que é o responsável por identificar a ocorrência de erros durante a transmissão de dados. Neste cenário, serão implementados no simulador os métodos do bit de paridade par e o CRC (Cyclic Redundancy Check) com o polinômio CRC-32 baseado no padrão IEEE 802.

Já na terceira sub etapa, será implementado no simulador o protocolo de correção de erros Hamming que é um método utilizado na camada de enlace de dados para detectar e corrigir erros de transmissão em quadros de dados.

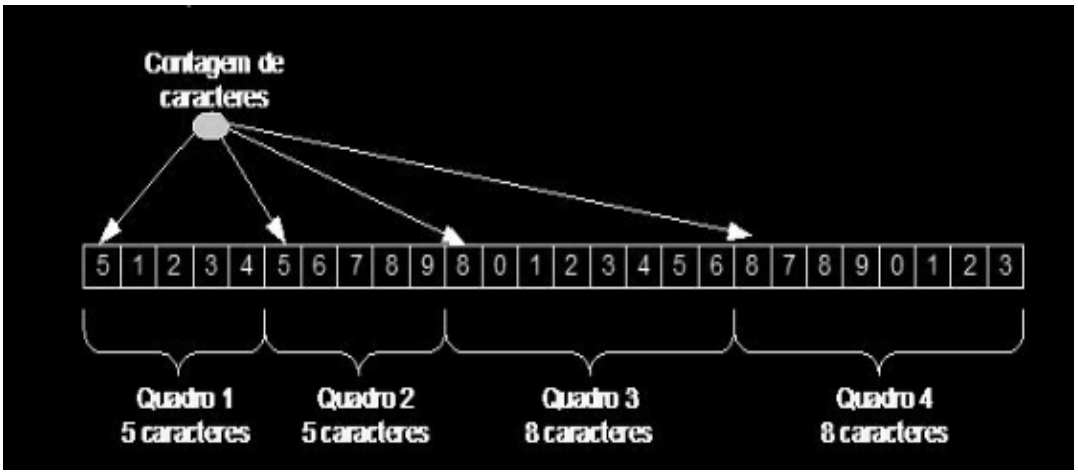
Além da camada física, agora o simulador também contará com a camada de enlace de dados e a interface de interação com o usuário permitirá que se escolha o tipo de codificação, o tipo de enquadramento e o método de tratamento de erro (sendo possível escolher a porcentagem de erro do meio físico de 0 até 100).

Implementação

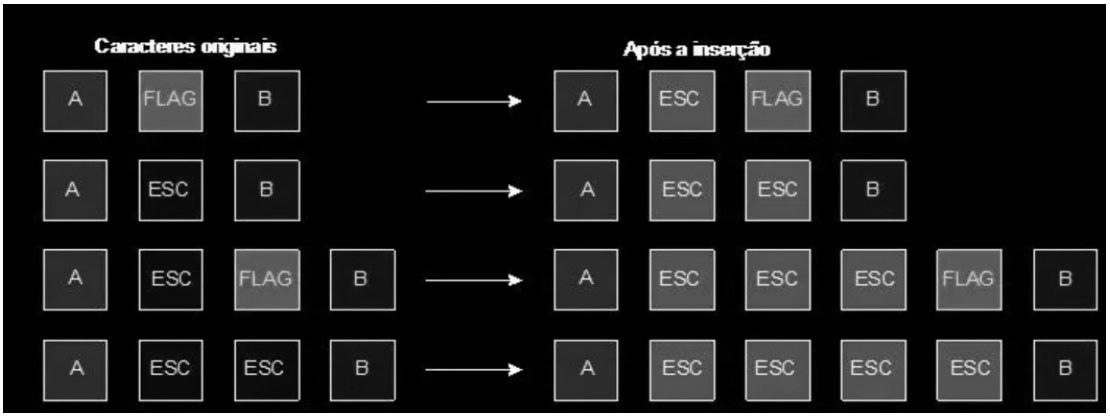
O simulador do funcionamento da camada de enlace será implementado por meio dos protocolos de enquadramento de dados, detecção e correção de erros. Portanto, é importante que se tenha esses protocolos bem definidos em mente para uma melhor compreensão da codificação do simulador.

Tratando-se do protocolo de enquadramento de dados, serão implementados no simulador os métodos de contagem de caracteres e o de inserção de bytes ou caracteres. O método de contagem de caracteres consiste em delimitar cada quadro por uma contagem de caracteres. Já no método de enquadramento por inserção de bytes ou caracteres, são somados bytes ou caracteres especiais no fim e no início de cada quadro para ter sua delimitação, e esses bytes ou caracteres especiais são conhecidos como flags que possuem como propósito marcar o início e fim do quadro.

Enquadramento de dados por contagem de caracteres

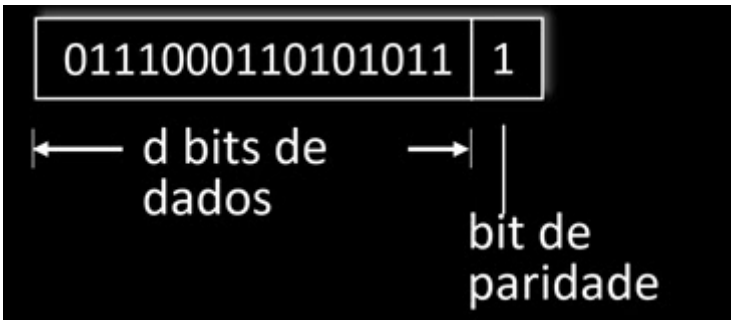


Enquadramento de dados por inserção de bytes ou caracteres

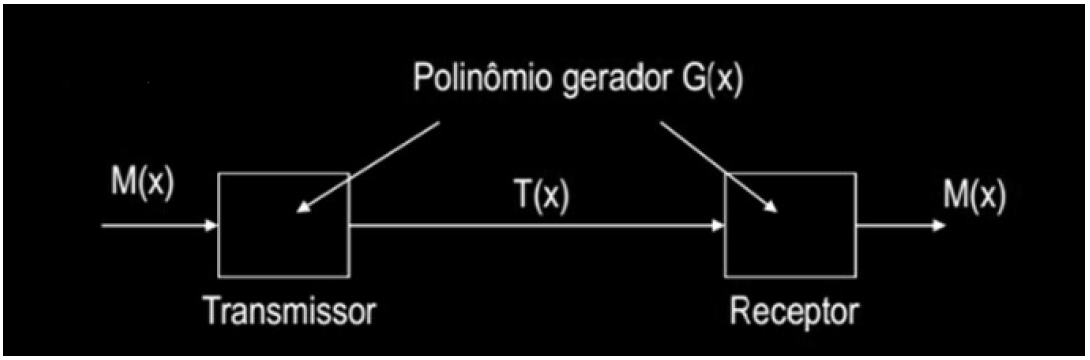


Já a implementação do protocolo de detecção de erros no simulador, abordará os métodos de bit de paridade par e o CRC. A técnica de detecção de erros do bit de paridade, consiste em adicionar um bit ao quadro com o intuito de verificar a paridade dos bits de dados. Sendo assim, esse bit de paridade é calculado de forma que o número total de bits 1 no quadro, inclusive o bit de paridade, seja sempre par. Dessa forma, se o número total de bits 1 for par quando o quadro é recebido, significa que não houve erro e se for ímpar, que houve erro de transmissão. Por outro lado, o CRC (polinômio CRC-32, IEEE 802) é um método mais sofisticado de detecção de erros e utiliza um polinômio gerador para realizar cálculos matemáticos nos dados do quadro, gerando um valor de redundância cíclica. Sendo assim, o valor é anexado ao quadro e quando o receptor o recebe, ele realiza outra vez os cálculos usando o mesmo polinômio gerador (se o valor calculado não for o mesmo do que o valor anexado ao quadro, isso indica a presença de erros). O polinômio CRC-32, baseado no padrão IEEE 802, realiza cálculos usando um polinômio de grau 32, o que torna possível uma capacidade de detecção de erros alta.

Bit de paridade par

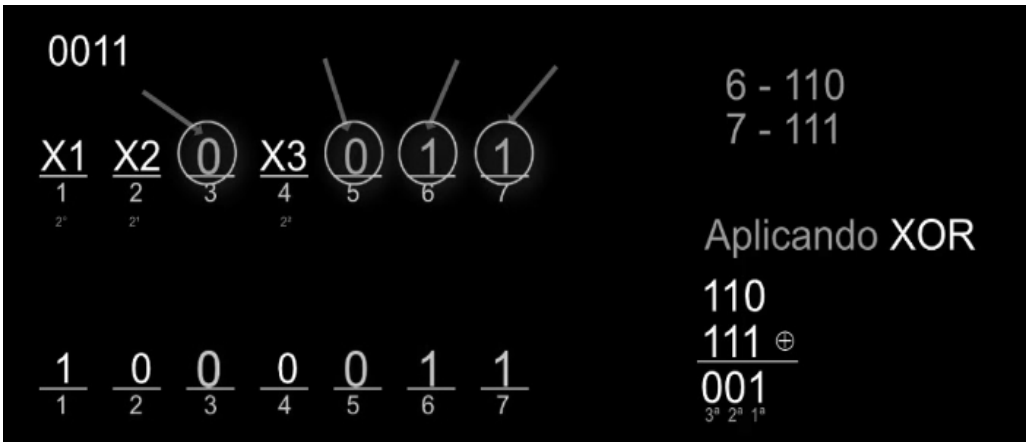


CRC (polinômio CRC-32, IEEE 802)



E por último, o protocolo de correção de erros no simulador, que contará com o método Hamming que se baseia no código de Hamming e adiciona bits de verificação aos dados que são transmitidos para que seja possível detectar e corrigir erros. Esse método, adiciona bits de paridade aos dados originais e a quantidade de bits de paridade adicionados depende do tamanho dos dados e dos bits de verificação necessários. Sendo assim, esses bits são calculados de forma que a paridade de bits de cada conjunto de bits seja verificada. Logo, na recepção de um quadro de dados com bits de verificação de Hamming, se realiza uma operação para verificar a paridade (se houver um erro em um único bit, é possível corrigi-lo automaticamente, porém se houver mais de um erro, o código de Hamming não é capaz da correção).

Código de Hamming



O simulador foi desenvolvido com base nesses protocolos e suas particularidades de enquadramento, correção e detecção de erros e o usuário poderá escolher a porcentagem de erro do meio físico para que a simulação possa ser realizada.

Sendo assim, na camada de enlace transmissora o simulador contará com uma função para representar a chamada da camada de enlace, com funções para os tratar os métodos de enquadramento, detecção de erros e correção de erro usando o método de Hamming.

Por outro ângulo a camada de enlace receptora, contará com uma função para representar o recebimento do quadro pela camada de enlace receptora, uma função para definir como o simulador deve interpretar o tratamento dos erros e uma função para definir o método de desenquadramento.

A codificação do simulador é modularizada e as funções implementadas no arquivo camadaEnlace.cpp na camada de enlace estão declaradas no arquivo de cabeçalho camadaEnlace.hpp.

camadaEnlace.hpp

```hpp
#ifndef CAMADAENLACE_HPP
#define CAMADAENLACE_HPP

#include <bits/stdc++.h>
using namespace std;

namespace CamadaEnlace {

    extern int tipoEnquadramento;
    extern int tipoTratamento;
    namespace Transmissora {

        void chamada(vector<bool> quadro);
        vector<bool> tratamentoErro(vector<bool> quadro);
        vector<bool> paridade(vector<bool> quadro);
        vector<bool> crc(vector<bool> quadro);
        vector<bool> hamming(vector<bool> quadro);
        vector<bool> enquadramento(vector<bool> quadro);
        vector<bool> contagemDeCaracteres(vector<bool> quadro);
        vector<bool> insercaoDeBytes(vector<bool> quadro);

    }

    namespace Receptora {

        void chamada(vector<bool> quadro);
        vector<bool> tratamentoErro(vector<bool> quadro);
        vector<bool> paridade(vector<bool> quadro);
        vector<bool> crc(vector<bool> quadro);
        vector<bool> hamming(vector<bool> quadro);
        vector<bool> desenquadramento(vector<bool> quadro);
        vector<bool> contagemDeCaracteres(vector<bool> quadro);
        vector<bool> insercaoDeBytes(vector<bool> quadro);

    }

}

#endif
```

Membros

Andressa de Araújo Pereira: relatório e testes no simulador
Pedro Brazil Suffert: slide e testes no simulador
Wallace Ben Teng Lin Wu: codificação e testes no simulador

Conclusão

Contudo, ao finalizar a implementação do simulador dos protocolos de enquadramento, detecção e correção de erros em C + +, é importante destacar os cenários ideais para a utilização de cada um desses protocolos.

Se tratando do protocolo de enquadramento de dados, a escolha entre a contagem de caracteres e a inserção de bytes ou caracteres depende fortemente das características da aplicação e do meio de transmissão. Por exemplo, a contagem de caracteres é mais indicada quando os dados são transmitidos em forma de caracteres e é necessário um método simples para delimitar os quadros. Já olhando por outro ponto de vista, a inserção de bytes ou caracteres é mais flexível, o que permite a inserção de bytes especiais como flags para marcar o início e o fim dos quadros.

Agora em relação aos protocolos de detecção de erros, o bit de paridade par é uma opção adequada em cenários com baixa taxa de erros, onde a detecção de erros simples é suficiente, este método é simples de implementar e oferece uma detecção eficiente de erros de transmissão. Porém o CRC, com o polinômio CRC-32 que é baseado no padrão IEEE 802, é mais recomendado para cenários com maiores taxas de erros, pois ele pode proporcionar uma detecção mais potente.

Por último, o protocolo de correção de erros Hamming é recomendado em cenários onde a detecção e correção de erros de um único bit são essenciais.

Sendo assim, durante o desenvolvimento do simulador dos protocolos em C + +, uma das maiores dificuldades encontradas foi a implementação do CRC-32 baseado no padrão IEEE 802. Pois o CRC-32 é um polinômio complexo que demanda uma série de cálculos matemáticos e manipulação de bits para gerar o valor de verificação de redundância cíclica. A implementação correta do algoritmo de CRC-32 exigiu um sólido entendimento dos conceitos teóricos subjacentes, assim como uma abordagem precisa para lidar com a operação de deslocamento de bits e a combinação de bits do quadro de dados.

Logo, a implementação da camada de enlace no simulador permitiu uma compreensão aprofundada do funcionamento da camada física de redes juntamente com a de enlaces, e do funcionamento dos devidos protocolos abordados.