



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Trabalho 2

Teleinformática e Redes 1

Simulador Manchesto 2.0

André Filipe da Conceição | 150005547

Gabriel Matheus da Rocha de Oliveira | 170103498

Guilherme Braga Pinto | 170162290

Introdução

O objetivo deste trabalho é, baseando-se no trabalho previamente executado relativo à simulação de Camada Física, simular a camada de Enlace, que para a simulação deste trabalho se localiza entre a aplicação do usuário e a camada Física propriamente dita. Utilizou-se estruturas de funções providas pelo pedido do trabalho para adicionar a funcionalidade de Enquadramento e Desenquadramento nesta camada.

O funcionamento do simulador é semelhante ao do trabalho anterior porém devido a adição da camada de Enlace a mensagem que antes era transmitida diretamente da Camada de Aplicação Transmissora para a Camada Física Transmissora agora é transmitida para a Camada de Enlace de Dados Transmissora onde ocorre um processo de enquadramento a partir de um dos três protocolos: Contagem de caracteres, Inserção de bytes e Inserção de bits antes de ser transmitida para a Camada Física transmissora.

Da mesma forma, durante o processo de recepção a mensagem que antes era transmitida da Camada Física Receptora para a Camada de aplicação receptora agora é transmitida para a Camada de Enlace de Dados receptora onde ocorre o processo de desenquadramento antes de ser transmitida para a Camada de Aplicação Receptora. Descrição dos protocolos implementados:

- **Contagem de Caracteres:** funciona por meio da anexação de um header indicativo de tamanho para cada quadro que vai ser transmitido mais adiante pela Camada Física. Se um quadro está composto então de 3 números representados em binários de 8 bits, logo um header deverá conter um número “4” representando que aquele quadro contém 3 espaços alocados para a mensagem útil a ser repassada e 1 espaço relativo ao header, totalizando o número 4. O receptor deverá então ler o header para cada quadro, identificar o tamanho do espaço a ser lido, para então retirar o header e passar a mensagem para a próxima camada.
- **Inserção de Bytes:** a inserção de bytes no enquadramento funciona por meio da adição de um byte de flag no começo e no fim de um quadro determinado. Duas flags consecutivas indicam logo que um quadro está terminando e em seguida outro está começando. A leitura de uma flag sozinha no trem de bits indica o final dos quadros. A mensagem com os headers adicionados acaba sendo então transmitida para o receptor, que terá que retirar as flags recompondo a mensagem original.
- **Inserção de Bits:** a inserção de bits apresenta um funcionamento parcialmente análogo ao da inserção de bytes, porém a atenção no enquadramento se volta à repetição de vários bits. Se adiciona também uma flag no começo e no fim do quadro, porém no campo da carga útil de cada quadro se busca checar sequências de bits determinados: a cada 5 repetições de um bit “1” é adicionado um bit “0” a fim de que não ocorram

grandes áreas com repetições de bits. A flag é escolhida de tal forma a se respeitar esse princípio, delimitando o espaço de cada quadro e evitando grandes cadeias de bits repetidos.

Exemplo de resultado de execução do simulador:

```
Aplicacao Transmissora:
Digite uma mensagem:
TR1

Mensagem a ser transmitida: TR1

Camada de Aplicacao Transmissora
010101000101001000110001

Camada de Enlace de dados Transmissora
Insercao de Bits:

0111111001010100010100100011000101111110

Camada Fisica Transmissora:
Codificacao Manchester Diferencial:
01100110011001010110100101101010010110101001010110010101101001100110011010

Meio de Transmissao
011001100110010101101001011010101001011010100101011001010101101001100110011010
```

```
Camada Fisica Receptora
Decodificacao Manchester Diferencial:
0111111001010100010100100011000101111110

Camada de Enlade de dados receptora
Desenquadramento da Insercao de Bits:

010101000101001000110001

Camada de Aplicação Receptora
010101000101001000110001

Aplicacao Receptora
A mensagem recebida foi: TR1
```

Implementação

Decisões relativas ao desenvolvimento

- **Uso de vetores auxiliares**

Em diversos momentos do código optou-se pelo uso de um vetor auxiliar para guiar o tamanho dos quadros que seriam codificados na Camada De Enlace. Como por exemplo no momento da Inserção de Bytes. Para cada quadro "i" (escolheu-se um tamanho fixo), o vetor auxiliar na posição "[i]" recebe o tamanho do quadro que será feito (o valor corresponde à quantidade em bytes). Isso é útil para controlar um possível quadro que recebera um tamanho menor do que o tamanho fixo, o que foi denominado "quadro de resto". Se fosse utilizado apenas 1 valor fixado para todos os quadros, em algum momento um quadro que não possuísse uma carga útil suficiente para preencher o tamanho escolhido receberia em seu header um valor de quantidade de bytes equivocado. O uso do vetor auxiliar mostra, no loop no qual se itera quadro por quadro, o tamanho daquele quadro específico. No momento da recepção desse quadro, será possível executar o desenquadramento de maneira como é de se esperar, no caso de um grande fluxo bruto de bits será desenquadrando o quadro de tamanho fixo quantas vezes for necessário, até que chegue o "quadro de resto". Estratégias análogas são aproveitadas em todos os enquadramentos da Camada de Enlace, e todos os desenquadramentos utilizam estratégias similares.

- **Update de envio entre Camada Física e Camada de Enlace**

A implementação da Camada Física previamente deixou como legado uma decisão que acabou dificultando o trabalho na Camada de Enlace. Como havia a necessidade de duplicar bits para cada "batida de clock" na codificação Manchester e Manchester Invertido, tomou-se a decisão de duplicar esses bits em um momento onde não existia a necessidade, a fim de facilitar e unificar a função que terminaria decodificando um fluxo bruto de bits para caracteres em ASCII. Com a adição de uma camada nova onde não havia nada antes, descobriu-se a necessidade de corrigir essa duplicação por meio da adição de uma função denominada "simplifica", que justamente vem corrigir esse erro de duplicação de bits. Atualizou-se também a Camada Física, onde havia a simplificação original desse fluxo de bits. Basicamente o update antecipou a simplificação a fim de fazer com que a Camada de Enlace possa receber os dados corretos. Antes de se resolver a simplificação, a duplicação de bits causava diversos erros de implementação, resultando na decodificação de diversos caracteres inexistentes em meio a caracteres existentes, porém no final a solução dada resolveu o problema de maneira satisfatória, com o objetivo de impactar de maneira mínima a Camada Física.

- **Técnica de Desenvolvimento**

Utilizou-se reuniões remotas para a melhor comunicação entre membros da equipe, além do uso do Github para o compartilhamento de artefatos criados no desenvolvimento do Trabalho. A modularização das funções permitiu que o trabalho fosse eventualmente

executado de maneira individual e paralela, apenas quando necessário. O repositório utilizado se encontra privado no momento da entrega deste trabalho.

- **Teste específico para o Enquadramento de Inserção de Bits**

A Inserção de Bits apresenta como "header" o seguinte conjunto de bits: "01111110". Este conjunto de bits se diferencia de qualquer outro conjunto na camada da carga útil pois existe o cuidado para não ocorrer uma sequência grande de bits "1". Dessa forma, para esta implementação houve o cuidado para testar se o programa sabia de fato lidar com essa inserção de bits que evitam as sequências de "1". Para isso, o símbolo "ö" foi extremamente conveniente, por se tratar do conjunto "11110000" que, por exemplo, junto de um símbolo dado por um conjunto de bytes que termina em muitos zeros (como por exemplo a letra "G" que é dada por "01000111" em ASCII), formaria uma grande cadeia de bits "1". Assim pode-se testar de maneira eficiente a inserção de bits na carga útil, o que de fato faz com que essa implementação tenha um diferencial maior quando comparada com uma Inserção de Bytes menos complexa, que não trabalha em atualizar o campo dos bits da mensagem.

- **Estrutura de Arquivos**

- **CamadaFisica.cpp e CamadaFisica.h:** Arquivos responsáveis por tratar da Camada Física. Para esse trabalho, esse conjunto de arquivos passou por um leve update relativo à refatoração e otimização, apenas o que foi considerado crucialmente necessário para que a Camada de Enlace fosse funcional.
- **CamadaEnlace.cpp e CamadaEnlace.h:** Arquivos responsáveis por tratar da Camada de Enlace. Estes são os arquivos de fato inéditos para este Trabalho.
- **Makefile:** Responsável pela compilação dos arquivos. Seu uso é dado pelo comando "make" no diretório do código, que resulta na criação do arquivo executável. Detalhes relativos à esse funcionamento também se encontram no arquivo README.md, que acompanha os arquivos do trabalho.
- **Simulador.cpp:** Onde se encontra a função "int main" do programa. Essa parte acaba abstraindo a Camada de Aplicação onde de fato está um usuário simulando o envio de uma mensagem.

- **Tamanho dos Vetores e Updates**

Se manteve a convenção adotada na Camada Física para a Camada de Enlace em relação à tamanhos dos vetores utilizados. Elaborando melhor, um tamanho de um vetor de binários é demarcado pelo número "2" em seu final. Diversas vezes existiu-se a necessidade de realocar tamanhos de vetores, então ao longo do código vetores com "áreas de carga inútil" são descartados para não causar ambiguidade para as camadas vizinhas: apenas bits úteis são repassados, incluindo os headers das diferentes camadas e os demarcadores de tamanho de vetores.

- **Modos de Funcionamento**

O funcionamento da seleção de cada tipo de enquadramento e desenquadramento pode ser controlado por meio de um “#define” no arquivo de header CamadaEnlace.h, relativo à declarações de funções responsáveis por implementar a Camada de Enlace. Cada número configura simultaneamente o enquadramento e o desenquadramento a fim de facilitar a configuração do simulador. Essa escolha garantirá que um enquadramento de um determinado tipo jamais seja desenquadrado por uma lógica diferente da original. É adotada a seguinte convenção:

| Enquadramento/Desenquadramento | Código atrelado |
|---------------------------------------|------------------------|
| Contagem de Caracteres | 0 |
| Inserção de Bytes | 1 |
| Inserção de Bits | 2 |

O funcionamento para a configuração da Camada Física é análogo e independente, podendo ser alterado por meio do #define no arquivo CamadaFisica.h sem alterar o funcionamento da Camada de Enlace. Cada codificação e decodificação pode ser usada com cada enquadramento e desenquadramento fazendo as devidas combinações. É adotada a seguinte convenção:

| Codificação/Decodificação | Código atrelado |
|----------------------------------|------------------------|
| Binária | 0 |
| Manchester | 1 |
| Manchester Diferencial | 2 |

Membros

- **André Filipe:** implementação prática da contagem de caracteres, inserção de bytes e inserção de bits; identificação e correção de erros de desenvolvimento.
- **Gabriel Matheus:** implementação prática da contagem de caracteres, inserção de bytes e inserção de bits; modularização e codificação da interface (terminal), desenvolvimento do código e documentação.
- **Guilherme Braga Pinto:** implementação prática da contagem de caracteres, inserção de bytes e inserção de bits; decisões de desenvolvimento; desenvolvimento do código; elaboração de comentários e documentação.

Conclusão

A implementação do simulador permitiu que o grupo ampliasse seu conhecimento acerca da transmissão de dados de forma mais prática a partir da introdução de uma nova camada com novas funcionalidades ao simulador criado no Trabalho 1 para a matéria de TR1. Com a adição da camada de Enlace ao simulador foi possível compreender o processo de transmissão de dados de forma um pouco mais completa mas ainda simplificada a partir da implementação dos processos de enquadramento de Contagem de Caracteres (Inserção de Bytes), Inserção de bits e Inserção de Bytes.

Durante o desenvolvimento do simulador a equipe encontrou um problema que demandou uma quantia considerável de tempo para ser solucionado. Após a implementação do processo de desenquadramento da inserção de Bytes foi notado que os resultados obtidos não eram os esperados e após muita discussão e análise da lógica do código foi descoberto que o problema não se encontrava na camada de enlace, mas em uma decisão adotada durante a implementação da Camada Física no trabalho 1. Após a detecção do problema a equipe realizou algumas mudanças necessárias na Camada Física e após essas alterações foi possível terminar a implementação dos processos da Camada de Enlace sem maiores dificuldades de maneira eficiente.

Em suma, o simulador alcançou resultados satisfatórios mesmo após encontrar obstáculos inesperados e se tornou ferramenta útil para ampliar os conhecimentos obtidos em relação a transmissão de dados utilizando apenas a Camada Física, de Aplicação e a de Enlace, assim alcançando os objetivos previamente estipulados.