



Departamento de Ciência da Computação
Universidade de Brasília

Simulador da Camada Física

David Herbert - 200057405

João Pedro Rodrigues. - 170121348

Mateus Lúcio - 170151727

Vanessa Paixão - 200028286

Brasília

2023

1. Introdução

Neste projeto, iremos demonstrar de maneira prática e simples a implementação de uma transmissão de dados. Faremos isso por meio de simulação, utilizando camadas de aplicação e uma camada física mais sofisticada, juntamente com um meio de comunicação simbólico.

O simulador é composto por camadas menores, que incluem a Aplicação Transmissora, a Camada de Aplicação Transmissora, a Camada Física Transmissora, o Meio de Transmissão, a Camada Física Receptora, a Camada de Aplicação Receptora e a Aplicação Receptora.

O funcionamento do simulador é o seguinte: ele exibe uma mensagem ao usuário, perguntando qual mensagem ele deseja transmitir a partir da camada de aplicação. Essa mensagem é convertida em bits pela Camada de Aplicação Transmissora e, em seguida, é enviada para a Camada Física Transmissora. A Camada Física Transmissora transmite o fluxo de bits para o Meio de Comunicação, onde os bits são transferidos para a Camada Física Receptora. Por fim, a mensagem é enviada para a Camada de Aplicação Receptora, decodificada e exibida corretamente.

Foram utilizados três tipos de codificação: Codificação Binária, Codificação Manchester e Codificação Bipolar.

O comportamento da codificação Binária, Manchester e Bipolar estão descritas nas Figuras 1 e 2. A explicação e implementação dessas será detalhada na Seção 2.

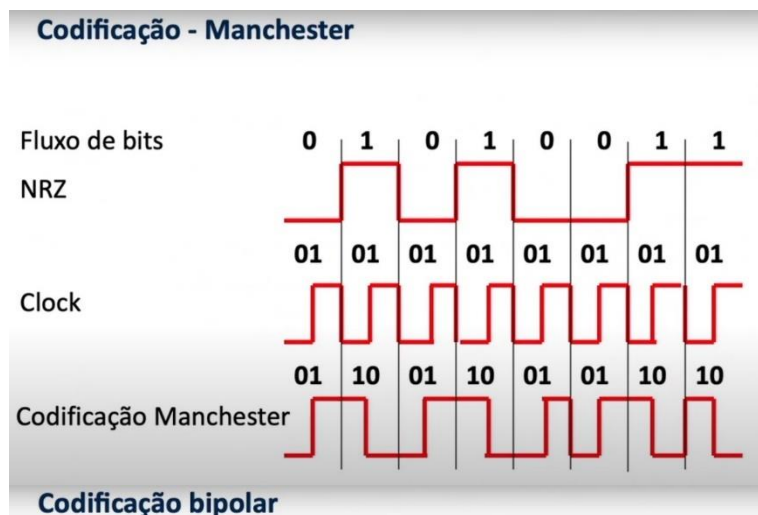


Figura 1 - Codificação Binária e Manchester

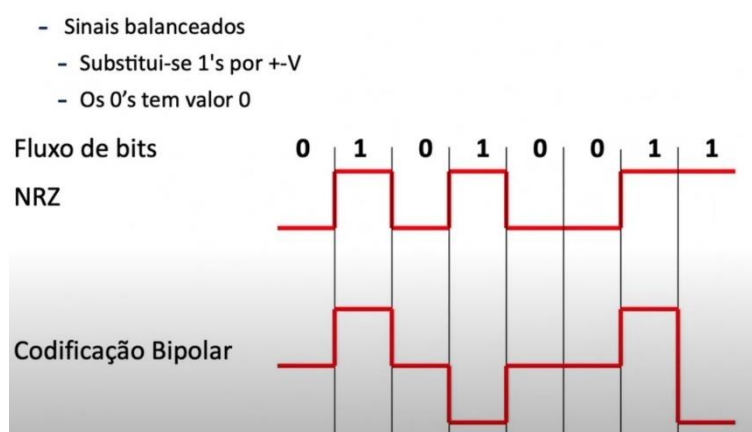


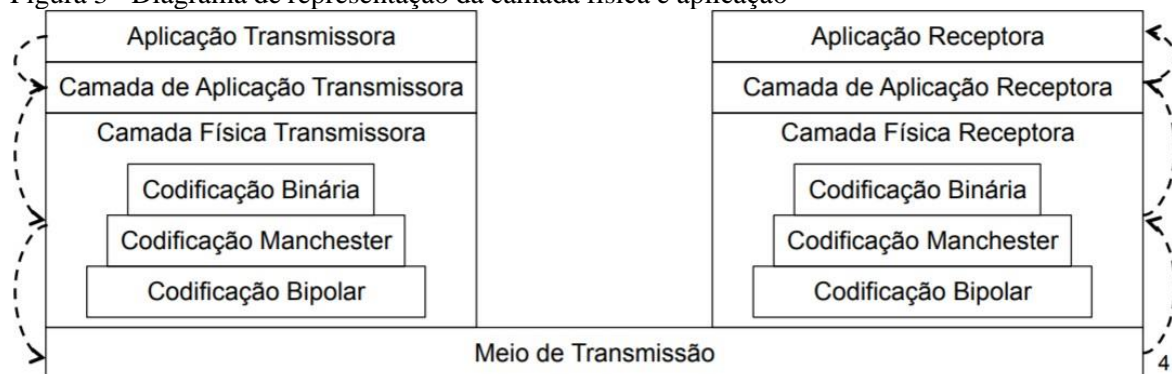
Figura 2- Codificação Binária e Bipolar

A Seção 2 abordará em detalhes a implementação do simulador da camada física, além de fornecer uma apresentação dos resultados obtidos durante a execução. A Seção 3 terá como foco a apresentação dos membros da equipe e suas respectivas atividades realizadas. Por fim, na Seção 4, será apresentada uma conclusão abrangente sobre o trabalho, abordando as dificuldades e limitações enfrentadas ao longo do projeto.

2. Implementação

A implementação do simulador da camada física foi baseada no diagrama apresentado na Figura 3, em que cada quadrado representa um protocolo. Cada protocolo foi implementado como uma sub-rotina no código, e os detalhes de funcionamento de cada função que representa esses protocolos são apresentados no diagrama da Figura 4.

Figura 3 - Diagrama de representação da camada física e aplicação



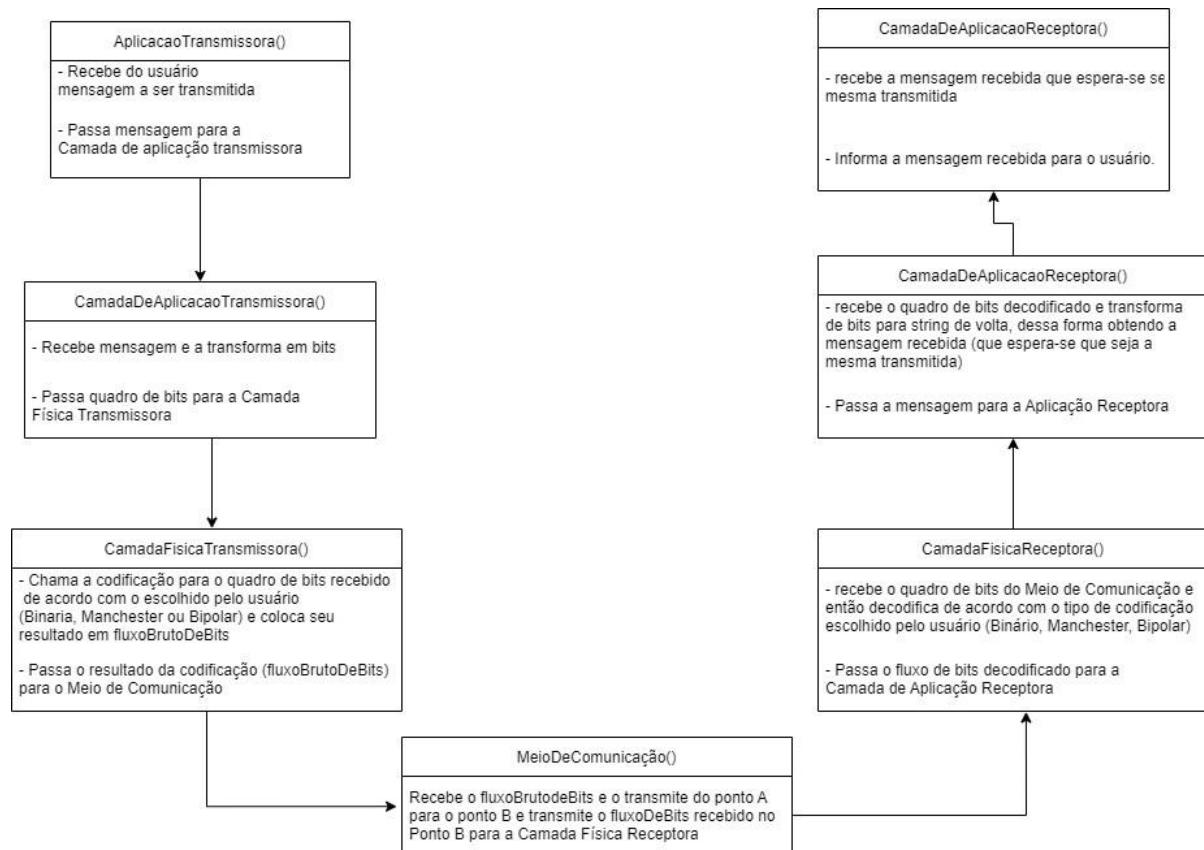


Figura 4 - Diagrama com explicações do funcionamento do código

A primeira função a ser executada é a Aplicação Transmissora, responsável por receber a mensagem que o usuário deseja transmitir. Em seguida, essa mensagem é encaminhada para a próxima função, a Camada de Aplicação, onde ocorre a transformação da mensagem em bits, conforme descrito em 2.1. O quadro de bits resultante é então enviado para a Camada Física Transmissora.

Na função Camada Física Transmissora, é aplicada a codificação escolhida pelo usuário, que pode ser binária, Manchester ou Bipolar, conforme explicado em 2.4. Após a codificação, é obtido um fluxo bruto de bits que é encaminhado para o Meio de Comunicação. O Meio de Comunicação estabelece a comunicação entre as camadas físicas de transmissão e recepção, como mostrado no diagrama da Figura 4. Em seguida, os dados codificados são transmitidos para a Camada Física Receptora.

A função Camada Física Receptora recebe os dados e os decodifica de acordo com a codificação escolhida, novamente seguindo as explicações em 2.4. Os dados decodificados são então repassados para a próxima camada, a Camada de Aplicação Receptora. Na Camada de Aplicação Receptora, ocorre a transformação dos bits decodificados, que estão em formato binário, em uma string, como descrito em 2.2. O resultado dessa transformação é finalmente encaminhado para a última camada, a Aplicação Receptora, que entrega ao usuário a mensagem transmitida e recebida.

Por fim, em 2.5 é apresentado os resultados da execução do código, utilizando astrês codificações, como é possível ver nas Figuras, a mensagem foi transmitida e recebida

corretamente, como esperado.

2.1 - Transformação da mensagem em bits

Para realizar a conversão da string mensagem em bits e permitir a manipulação e codificação dos mesmos, utilizamos a biblioteca do C++ chamada "bitset". Essa biblioteca foi responsável por transformar cada letra da mensagem em uma sequência de 8 bits binários. A partir dessa transformação, cada bit foi armazenado em um vetor utilizando outra biblioteca que nos permitiu uma manipulação fácil da estrutura de dados.

Dessa forma, para cada letra da mensagem, realizamos sua conversão em binário, gerando assim um conjunto de 8 bits. Em seguida, cada bit foi inserido em um vetor de inteiros. Essa abordagem nos proporcionou o quadro de bits necessário para a codificação e transmissão.

2.2 - Decodificação de bits para mensagem

Para decodificar os bits e convertê-los de volta para uma string, o processo inicia lendo o vetor onde os bits estão armazenados. Durante essa leitura, processamos grupos consecutivos de 8 bits até o final do vetor. Com base nos bits que estão definidos como 1, realizamos a conversão de binário para decimal usando uma variável acumuladora. Essa variável soma-se sempre que um bit é igual a 1. Utilizando a posição do bit definido como 1 no grupo de 8 bits como expoente, realizamos uma potenciação simples com base dois.

Após concluir o processamento do grupo de 8 bits, convertemos o valor inteiro resultante em um caractere. Isso é feito utilizando um type casting para transformar o valor em char. Em seguida, concatenamos esse caractere na string que será apresentada ao final da execução. Ao final da concatenação, repetimos todo o ciclo de processamento até percorrer todo o vetor. Dessa forma, obtemos a mensagem decodificada e no formato de uma string.

2.3 - Codificações

Os protocolos de codificação utilizados são descritos a seguir:

- **Codificação Binária:** Nesse protocolo, implementamos a técnica de codificação NRZ-L (Non Return Zero - Level). Essa técnica é baseada na presença ou ausência de tensão para representar os sinais lógicos 1 e 0, respectivamente. Na implementação, os bits da mensagem são transmitidos diretamente pela camada física de transmissão, sem sofrer alterações devido ao padrão de codificação. Ao serem recebidos na camada física de recepção, os bits não precisam ser decodificados e são repassados à camada de aplicação receptora. Nessa camada, eles são convertidos de bits para uma string e, em seguida, apresentados ao usuário. É importante ressaltar que outros protocolos de codificação também podem ser implementados, como Manchester e Bipolar, porém, seus detalhes não estão incluídos neste trecho.
- **Codificação Manchester:** Nesse protocolo, implementamos a técnica de codificação

Manchester, que mantém o receptor e o transmissor sincronizados por meio de um sinal pulsado que combina o sinal de dados e o sinal de clock. Na implementação, os bits recebidos da camada de aplicação transmissora são codificados utilizando estruturas condicionais que representam uma porta XOR. Essa codificação utiliza o bit atual e os valores de subida do clock (0 e 1) como entradas, resultando em uma saída de dois bits que combina o bit atual e o clock. Após a codificação, os bits são enviados pela camada física transmissora e encaminhados para a camada física receptora por meio da camada de meio de transmissão. Ao chegar na camada receptora, os bits são separados do clock utilizando uma decodificação simples, também utilizando estruturas condicionais. Em seguida, os bits são direcionados para a camada de aplicação receptora, onde são convertidos em uma string para serem apresentados ao usuário.

- **Codificação Bipolar:** Nesse protocolo, utilizamos a técnica de codificação Bipolar, que utiliza três níveis para representar os símbolos binários. A ausência de sinal representa o nível lógico 0, enquanto os sinais que representam pulsos positivos e negativos são associados ao nível lógico 1. Na implementação, os bits recebidos da camada de aplicação transmissora são codificados utilizando estruturas condicionais. Quando o bit é zero, seu valor é mantido. Caso seja um, geramos um sinal de valor $+V$ ou $-V$. Para isso, utilizamos uma flag para verificar se o último sinal codificado tinha valor $+V$ ou $-V$. Em seguida, os bits são enviados pela camada física transmissora e passam pela camada de meio de transmissão. Ao chegar na camada receptora, os bits são decodificados utilizando uma checagem simples com estruturas condicionais. Os bits de sinal 0 são mantidos, enquanto qualquer sinal, seja ele $+V$ ou $-V$, é convertido para um bit de nível lógico 1. Após essa decodificação, os bits seguem para a camada de aplicação receptora, onde são convertidos em uma string para serem apresentados ao usuário

2.4 - Exemplos de resultados do simulador para cada codificação:

Logo abaixo estão os resultados da execução do código para cada codificação, como mensagem utilizamos “mensagem teste”.

Codificação Binária:

```
PS C:\Users\joao2\OneDrive\UnB\1-2023\TR1\Trabalho 2\simulador> make run
obj/simulador.exe
Digite uma mensagem: mensagem teste
Camada De Aplicacao Transmissora | Mensagem convertida para bits:
01101101011001010110111001110011011000001011001110110010101101101001000000111010001100101011100110111010001100101

=====
Selecione uma opcao de codificacao:
=====
0 - Codificacao Binaria
1 - Codificacao Manchester
2 - Codificacao Bipolar
=====
Opcao: 0

Codificacao BINARIA selecionada
Camada Fisica Transmissora - Mensagem codificada:
01101101011001010110111001110011011000001011001110110010101101101001000000111010001100101011100110111010001100101

Enviando a mensagem, aguarde...

Camada Fisica Receptora
Mensagem recebida com sucesso! Iniciando decodificacao BINARIA:
01101101011001010110111001110011011000001011001110110010101101101001000000111010001100101011100110111010001100101

Mensagem decodificada
+-----+
| mensagem teste |
+-----+
```

Codificação Manchester:

[illegible]

Codificação Bipolar:

```
PS C:\Users\joao2\OneDrive\UnB\1-2023\TR1\Trabalho 2\simulador> make run
obj/simulador.exe
Digite uma mensagem: mensagem teste
Camada De Aplicacao Transmissora | Mensagem convertida para bits:
01101101011001010110111001110011011000010110011101100101011100110111010001100101

=====
      Selecione uma opcao de codificacao:
=====
      0 - Codificacao Binaria
      1 - Codificacao Manchester
      2 - Codificacao Bipolar
=====
      Opcao: 2

Codificacao BIPOLAR selecionada
Camada Fisica Transmissora - Mensagem codificada:
01-101-1010-1100-1010-110-11-1001-1100-110-110000-101-1001-110-1100-1010-110-110-1001000000-11-101000-1100-1010-11-1001-101-110-10001-10010-1

Enviando a mensagem, aguarde...

Camada Fisica Receptora
Mensagem recebida com sucesso! Iniciando decodificacao BIPOLAR:
0110110101100101011011100111001101100001011001110110010101110100100000011101000110011100110111010001100101

Mensagem decodificada
+-----+
| mensagem teste |
+-----+
```

3. Membros e Atividades

David Herbert: Implementação da transformação da string para bits e bits para string, codificação e decodificação bipolar, elaboração dos slides.

João Pedro Rodrigues: Implementação da transformação da string para bits e bits para string, da codificação e decodificação binária, manchester e bipolar, elaboração do relatório.

Mateus Lúcio: Implementação da codificação e decodificação binária, manchester e bipolar, elaboração do relatório e criação do arquivo de automação Makefile.

Vanessa Paixão: Implementação da transformação da string para bits e bits para string, codificação e decodificação manchester, elaboração dos slides e interação com usuário.

4. Conclusão

Ao chegarmos ao término deste projeto, podemos concluir que a abordagem prática complementou de forma interessante os conhecimentos teóricos adquiridos na disciplina. Durante a construção do trabalho, realizamos pesquisas em bibliografias adicionais para reafirmar nosso entendimento das codificações e obter maior clareza na implementação do código-fonte.

Enfrentamos algumas dificuldades ao longo do projeto que merecem destaque. Tivemos desafios na compreensão e implementação da interface gráfica. No entanto, conseguimos concluir o projeto com um resultado satisfatório, graças ao trabalho em grupo e discussões produtivas. Outra dificuldade foi compreender como aplicar a teoria das codificações em código C++, mas com esforço conjunto, chegamos a um consenso e obtivemos resultados práticos na implementação.

Por fim, enfatizamos que os resultados obtidos são satisfatórios e estão de acordo com as especificações do projeto. Estamos satisfeitos com o trabalho realizado.