

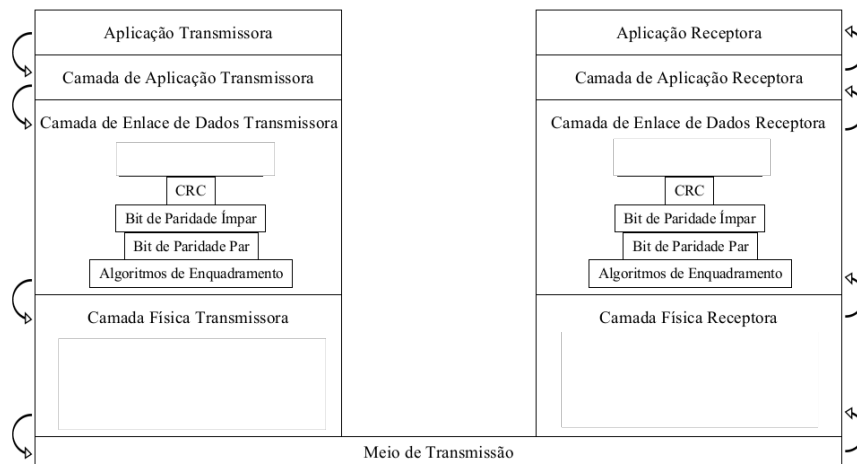
Trabalho em grupo de 4 pessoas

Introdução

Simule o funcionamento de camada de enlace por meio da implementação dos protocolos existentes . Ou seja, faça um programa que simule o envio um dado de um computador A para um computador para um computador B.

Realizando as devidas conversões entre String, da mensagem a ser enviada, e os bytes e bits a serem verificados pela camada de enlace.

Cenário



A camada de aplicação e de recepção são dadas abaixo, bem como do meio de transmissão. Camada de aplicação do transmissor.


```
void main (void) {  
    AplicacaoTransmissora();  
} //fim do metodo main  
  
void AplicacaoTransmissora (void) {  
    string mensagem;  
    cout << "Digite uma mensagem:" << endl;  
    cin >> mensagem;  
    //chama a proxima camada  
    CamadaDeAplicacaoTransmissora(mensagem); //em um exemplo mais  
                                                realistico, aqui seria dado um SEND do SOX  
} //fim do metodo AplicacaoTransmissora  
  
void CamadaDeAplicacaoTransmissora (string mensagem) {  
    //int quadro [] = mensagem //trabalhar com bits!!!  
    //chama a proxima camada
```

A próxima camada é a camada de enlace que por sua vez, este trabalho, envia para o meio de comunicação na qual é recepcionada pela camada de enlace do receptor e as camadas de aplicação do receptor. A camada de aplicação do receptor segue abaixo

```
void CamadaDeAplicacaoReceptora (int quadro []) {  
    //string mensagem = quadro []; //estava trabalhando com bits  
    //chama proxima camada  
    AplicacaoReceptora(mensagem);  
} //fim do metodo CamadaDeAplicacaoReceptora  
  
void AplicacaoReceptora (string mensagem) {  
    cout << "A mensagem recebida foi:" << mensagem << endl;  
} //fim do metodo AplicacaoReceptora
```

A implementação do meio de comunicação segue abaixo

```
void MeioDeComunicacao (int fluxoBrutoDeBits []) {  
    //OBS: trabalhar com BITS e nao com BYTES!!!  
    int erro, porcentagemDeErros;  
    int fluxoBrutoDeBitsPontoA [], fluxoBrutoDeBitsPontoB [];  
  
    porcentagemDeErros = 0; //10%, 20%, 30%, 40%, ..., 100%  
    fluxoBrutoDeBitsPontoA = fluxoBrutoDeBits;  
  
    while (fluxoBrutoDeBitsPontoB.lenght!=  
           fluxoBrutoDeBitsPontoA) {  
        if ((rand()%100)== ... ) //fazer a probabilidade do erro  
            fluxoBrutoBitsPontoB += fluxoBrutoBitsPontoA; //BITS!!!  
        else //ERRO! INVERTER (usa condicao ternaria)  
            fluxoBrutoBitsPontoB==0) ?  
            fluxoBrutoBitsPontoA=fluxoBrutoBitsPontoB++ :  
            fluxoBrutoBitsPontoA=fluxoBrutoBitsPontoB--;  
    } //fim do while  
} //fim do metodo MeioDeTransmissao
```



Ao termino da camada de meio de comunicação chama-se a camada de enlace do receptor.

Desenvolvimento das camadas de enlaces transmissores e receptores, na qual pode seguir a seguinte estrutura

```
void CamadaEnlaceDadosTransmissora (int quadro []) {  
      
    CamadaEnlaceDadosTransmissoraControleDeErro(quadro);  
      
    //chama proxima camada  
      
} //fim do metodo CamadaEnlaceDadosTransmissora  
  
void CamadaEnlaceDadosTransmissoraControleDeErro (int quadro []) {  
    int tipoDeControleDeErro = 0; //alterar de acordo com o teste  
    switch (tipoDeControleDeErro) {  
        case 0 : //bit de paridade par  
            //codigo  
            break;  
        case 1 : //bit de paridade impar  
            //codigo  
            break;  
        case 2 : //CRC  
            //codigo  
              
            //codigo  
            break;  
    } //fim do switch/case  
} //fim do metodo CamadaEnlaceDadosTransmissoraControleDeErro
```


```

void CamadaEnlaceDadosTransmissoraControleDeErroBitParidadePar (int quadro []) {
    //implementacao do algoritmo
} //fim do metodo CamadaEnlaceDadosTransmissoraControleDeErroBitParidadePar

void CamadaEnlaceDadosTransmissoraControleDeErroBitParidadeImpar (int quadro [])
{
    //implementacao do algoritmo
} //fim do metodo CamadaEnlaceDadosTransmissoraControleDeErroBitParidadeImpar

void CamadaEnlaceDadosTransmissoraControleDeErroCRC (int quadro []) {
    //implementacao do algoritmo
    //usar polinomio CRC-32(IEEE 802)
} //fim do metodo CamadaEnlaceDadosTransmissoraControleDeErroCRC

```



O repetir os mesmo procedimento para a camada de enlace do receptor.

O que deve ser entregue

Um arquivo zip com todo o sistema com os códigos a cada módulo, um arquivo Makefile para compilar e executar o código e um README com os nomes dos membros do grupo, a versão do sistema operacional (Linux) e compilador utilizados, além de quaisquer instruções adicionais que julgarem necessárias, tais como principais verificação de falha da conexão do socket e transmissão das informações.

OU

Um repositório do github com todos os arquivos acima.

O que deve ser apresentado.

O funcionamento do Código, mostrando o seu funcionamento correto nos 3 tipos de verificação

O seu funcionamento correto valerá 5, e os comentários bem como a indentação valerá mais 5