

Arquitetura de Redes de Computadores

Prof. Macêdo Firmino

Aula 08

Camada de Enlace: Endereçamento, Controle de Erro e Fluxo

“Um coração magoado tende a ver tudo sob as lentes da própria dor. Essa é uma das maiores barreiras para o perdão. Para se libertar da mágoa, é preciso trocar as lentes.”
(Rossandro Klinjey)



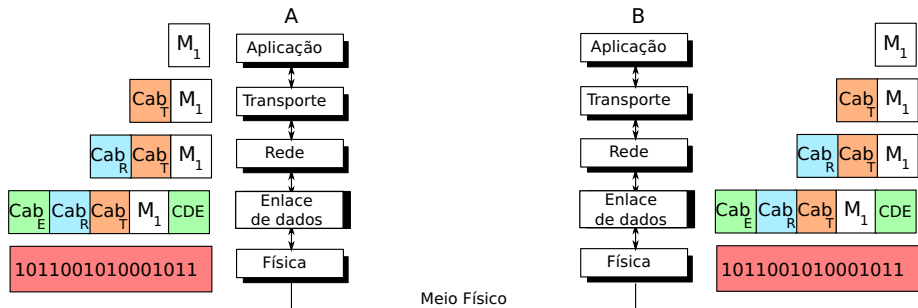
INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

O que Aprenderemos?

- Apresentar os conceitos fundamentais da camada de enlace, explorando sua importância para garantir a entrega confiável de informações entre dispositivos vizinhos.
- Aprender como é realizado o endereçamento na camada de enlace.
- Apresentar umas das principais funções da camada de enlace, como detecção e correção de erros, e endereçamento físico (endereços MAC).



Relembrando.....



Camada de Enlace de Dados

As funções da camada de Enlace são:

- Dividir o fluxo de *bits* em unidades de dados (Enquadramento).
- Realizar o endereçamento físico;
- Implementar um mecanismo de detecção de erros e retransmissão de quadros (Controle de erro);
- Controlar o meio físico para evitar que um transmissor não interfira em outro (Controle de acesso).



Endereçamento Físico (MAC)

O endereço físico (ou endereço MAC) é uma identificação única atribuída a cada dispositivo de rede que se conecta a uma rede local (LAN). Esse endereço é gravado na placa de rede do dispositivo e é usado na camada de enlace para garantir que os dados sejam entregues ao dispositivo correto dentro de uma rede.

O endereço MAC é composto por 48 *bits*, representados por 12 dígitos hexadecimais (por exemplo, 00:1A:2B:3C:4D:5E). Ele é dividido em duas partes: os primeiros 24 *bits* identificam o fabricante da placa de rede, e os últimos 24 *bits* identificam o número de série da placa de rede.



Endereçamento Físico (MAC)

Cada dispositivo de rede tem um endereço MAC único no mundo, o que significa que não existem dois dispositivos com o mesmo endereço MAC, garantindo a identificação correta de cada dispositivo dentro de uma rede.

O endereço MAC só é útil dentro de uma rede local. Quando os dados precisam ser enviados para fora da LAN, outros tipos de endereços (como endereços IP) são usados em conjunto com o endereço MAC para garantir a entrega correta em redes mais amplas, como a Internet.



Endereçamento Físico (MAC)



Controle de Erros

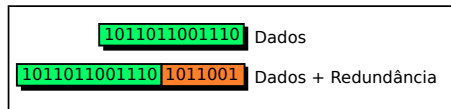
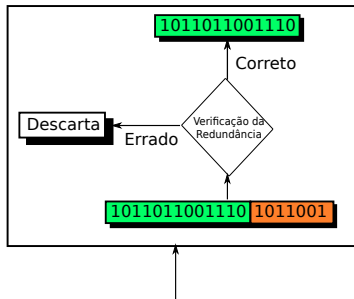
Durante a transmissão de dados em uma rede, diversos fatores podem causar erros, resultando na alteração, perda ou corrupção da informação. Um dos principais fatores: atenuação do sinal, interferência eletromagnética, falhas na placa de rede, em cabos ou conectores mal encaixados, etc.

Em redes de computadores, são utilizadas técnicas matemáticas para o receptor descobrir se os *bits* recebidos estão corretos. Se o receptor detectar um erro, ele solicita ao remetente que retransmita o quadro com erro.



Detecção de Erros

Bits de redundância são *bits* adicionais incluídos em uma mensagem transmitida para permitir a detecção de erros. Esses *bits* não fazem parte dos dados originais, mas são calculados com base nos dados e anexados à mensagem antes da transmissão. Quando a mensagem chega ao destino, o receptor usa os *bits* de redundância para verificar se a mensagem foi alterada durante a transmissão.



Algoritmos de Detecção de Erros

Os dois principais algoritmos que utilizam *bits* de redundância para detecção de erro são:

- Bit de Paridade: consiste em adicionar um bit extra ao final de uma sequência de bits antes de transmiti-la, de forma que o número total de bits “1” na sequência (incluindo o bit de paridade) seja par ou ímpar.
- CRC (Verificação de Redundância Cíclica): utiliza divisão binária. O processo de divisão é semelhante à divisão de números decimais, mas usando operações AND e XOR;
- *Checksum*: os dados são somados, utilizando aritmética de complemento de dois, e resultado da soma é então complementado (inversão dos *bits*) para obter o *checksum*.

Bit de Paridade

O bit de paridade é uma técnica de detecção de erros utilizada na transmissão de dados. O seu funcionamento consiste em adicionar um bit extra ao final de uma sequência de bits antes de transmiti-la. Existem dois tipos principais de paridade:

- Paridade Par: o bit de paridade é escolhido de modo que o número total de bits “1” na sequência (incluindo o bit de paridade) seja par.
- Paridade Ímpar: o bit de paridade é escolhido de forma que o número total de bits “1” (incluindo o bit de paridade) seja ímpar.



Bit de Paridade

Por exemplo, suponha que temos 4 bits e uma paridade par. Desta forma, as possibilidades são:

<i>Palavras de dados</i>	<i>Palavras de código</i>	<i>Palavras de dados</i>	<i>Palavras de código</i>
0000	00000	1000	10001
0001	00011	1001	10010
0010	00101	1010	10100
0011	00110	1011	10111
0100	01001	1100	11000
0101	01010	1101	11011
0110	01100	1110	11101
0111	01111	1111	11110



Checksum

O *checksum* é uma técnica mais simples que o CRC, usada para detectar erros em blocos de dados, principalmente em transmissões de pacotes ou em armazenamento de dados. Os dados a serem transmitidos são divididos em blocos. Os valores binários de cada bloco são somados, utilizando aritmética de complemento de dois. Se a soma exceder o valor máximo que pode ser representado, o valor excedente é somado de volta ao resultado. O resultado da soma é então complementado (inversão dos *bits*) para obter o *checksum*. Este valor é transmitido junto com os dados.



Checksum - Aritmética Binária

Função Adição:

$$\begin{array}{r} 0 + 0 = 0 \\ 0 + 1 = 1 \\ 1 + 0 = 1 \\ 1 + 1 = 10 \end{array}$$

Observação: se durante o processo da soma, o resultado extrapolar o tamanho do segmento, n bits, os bits extras mais à esquerda precisam ser adicionados aos n bits mais à direita.



Checksum - Gerador (Exemplo)

1010100110111001 Dados

10101001 10111001 Dividir o bloco de dados com 8bits

10101001	
10111001	
101100010	← Soma
01100010	
1	← Complemento de um
01100011	
10011100	← Inversão
10011100	← Checksum

1010100110111001 **10011100**

Dados + Checksum

Checksum - Verificador

O receptor realiza a mesma soma nos blocos de dados recebidos e adiciona o *checksum*. Se o resultado for zero, os dados serão aceitos, caso contrário, será rejeitado.



Checksum - Verificador (Exemplo)

1010100110111001 10011100

Dados + Checksum

10101001 10111001 10011100

Dividir o bloco de dados com 8bits

10101001
10111001
10011100

11111110

← Soma

11111110

1

← Complemento de um

11111111

00000000

← Complemento

00000000

← Checksum



Comparação entre CRC e *Checksum*

- CRC: É mais complexo e oferece maior robustez na detecção de erros. É capaz de detectar padrões mais variados de erros. É o mais utilizado na camada de enlace.
- *Checksum*: É mais simples de implementar, mas menos robusto. Pode detectar erros simples, como *bits* invertidos, mas pode falhar em detectar erros mais complexos. Usado em aplicações onde a simplicidade e a velocidade são mais importantes que a robustez, como em protocolos da camada de rede e transporte.
- Bit de paridade: é o mais simples e pouco utilizada na transmissão de dados, pois ele é capaz de detectar melhor um número ímpar de erros.



Atividade

01. Calcule o *checksum* dos conjuntos de *bits* a seguir. Utilize blocos de 4 *bits*.

a) 01001110

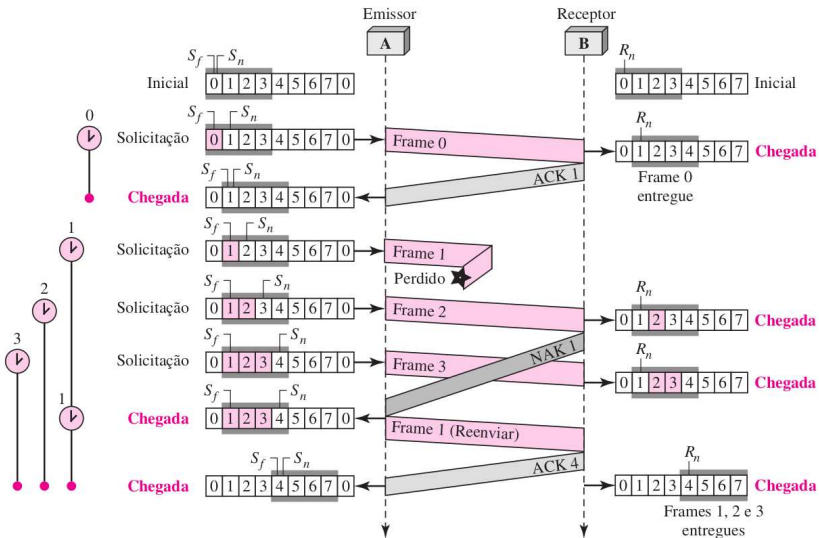
b) 11011000

c) 11011101

Controle de Fluxo e Erro

O controle de fluxo é responsável por gerenciar o ritmo de envio de dados entre dois dispositivos conectados, garantindo que o receptor não seja sobrecarregado com mais dados do que ele pode processar ou armazenar. Ele é implementado em conjunto com o controle de erro, com as técnicas:

- Código de detecção de erro: CRC ou Checksum.
- Confirmação (ACK - *Acknowledgment*): o receptor envia uma confirmação para cada pacote de dados recebido com sucesso. Este ACK informa ao transmissor que o pacote foi recebido corretamente e que ele pode continuar enviando o próximo pacote.
- Temporizador: um temporizador é iniciado pelo transmissor sempre que um pacote de dados é enviado. Se o temporizador expirar antes que o transmissor receba o ACK correspondente, o transmissor assume que o pacote foi perdido ou corrompido e o retransmite.



Repetição Seletiva ARQ

O principal mecanismo de controle de fluxo é o Repetição Seletiva ARQ. Ele funciona da seguinte forma:

- O transmissor pode enviar múltiplos pacotes consecutivos desde que não ultrapasse o tamanho da memória do receptor. Cada pacote é numerado sequencialmente para que o receptor possa identificar a ordem correta dos pacotes.
- O receptor verifica a integridade de cada pacote recebido. Se um pacote for recebido corretamente, o receptor o armazena. Se um pacote for perdido ou corrompido, o receptor simplesmente não envia o ACK correspondente para aquele pacote.



Repetição Seletiva ARQ

- Como os pacotes podem chegar fora de ordem, o receptor os armazena em memória e, uma vez que todos os pacotes esperados tenham sido recebidos, ele os reordena antes de passá-los para a camada superior.
- Os ACKs são acumulativos, ou seja, um ACK pode confirmar a recepção de múltiplos pacotes.

Dúvidas



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE