

Unidade III:

Camada de Enlace

Prof. Max do Val Machado



PUC Minas

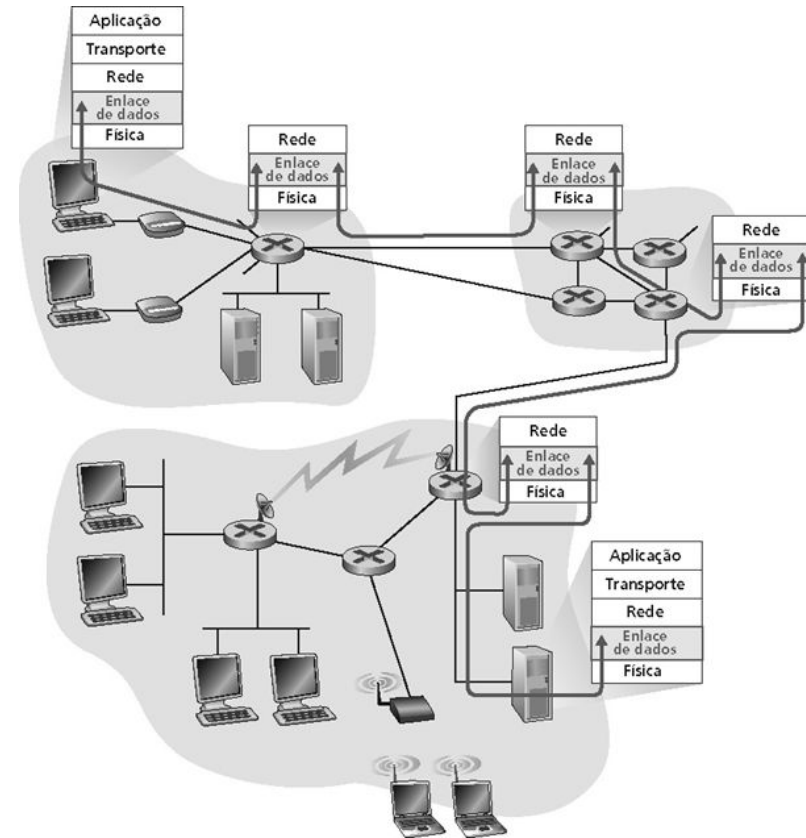
Instituto de Ciências Exatas e Informática
Departamento de Ciência da Computação
Disciplina Redes de Computadores I

- Introdução
- Orientação à Conexão e Existência de Confirmação
- Enquadramento
- Controle de Fluxo
- Controle de Erro
- Camada de Enlace da Internet

- **Introdução**
- Orientação à Conexão e Existência de Confirmação
- Enquadramento
- Controle de Fluxo
- Controle de Erro
- Camada de Enlace da Internet

Conceitos Básicos

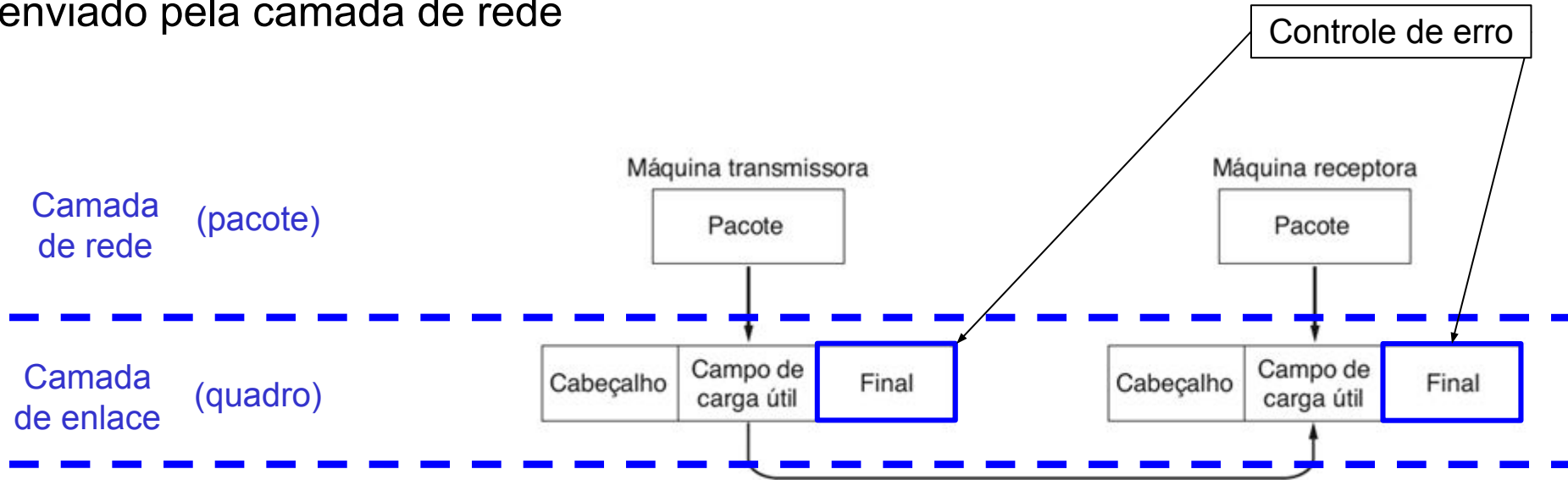
- **Nós**: os hospedeiros e os roteadores
- **Enlaces**: os canais de comunicação que se conectam a nós adjacentes pelo caminho de comunicação (por exemplo, os enlaces com fio, os sem e as LANs)

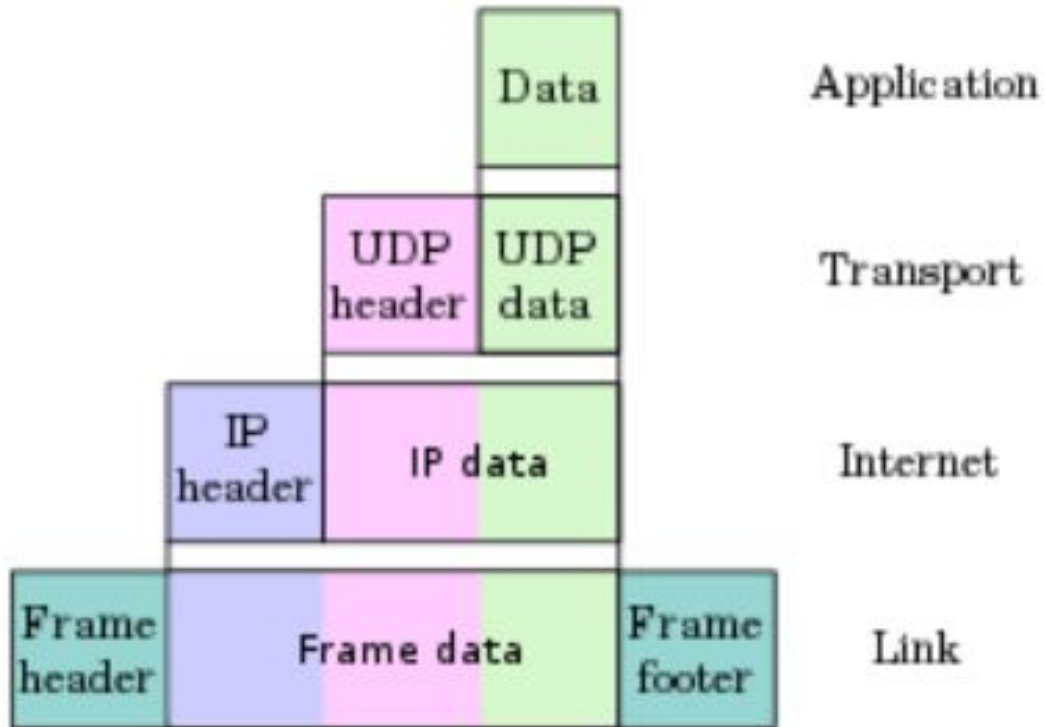


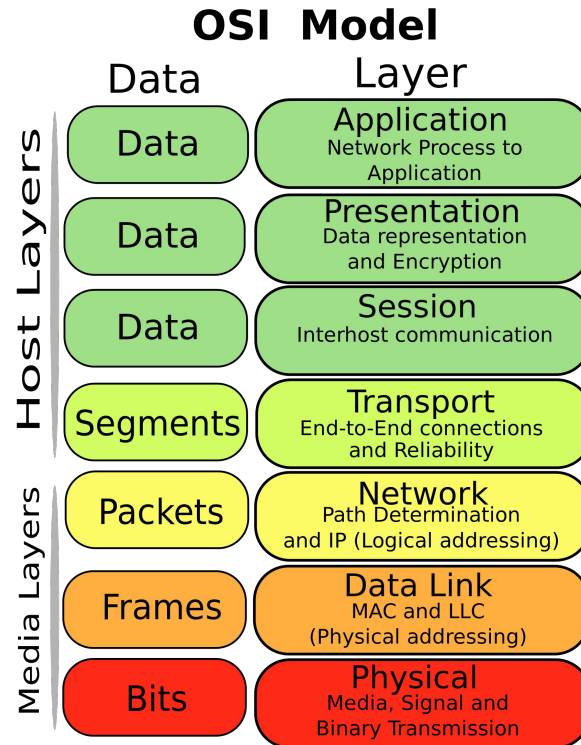
Tipos de Enlaces

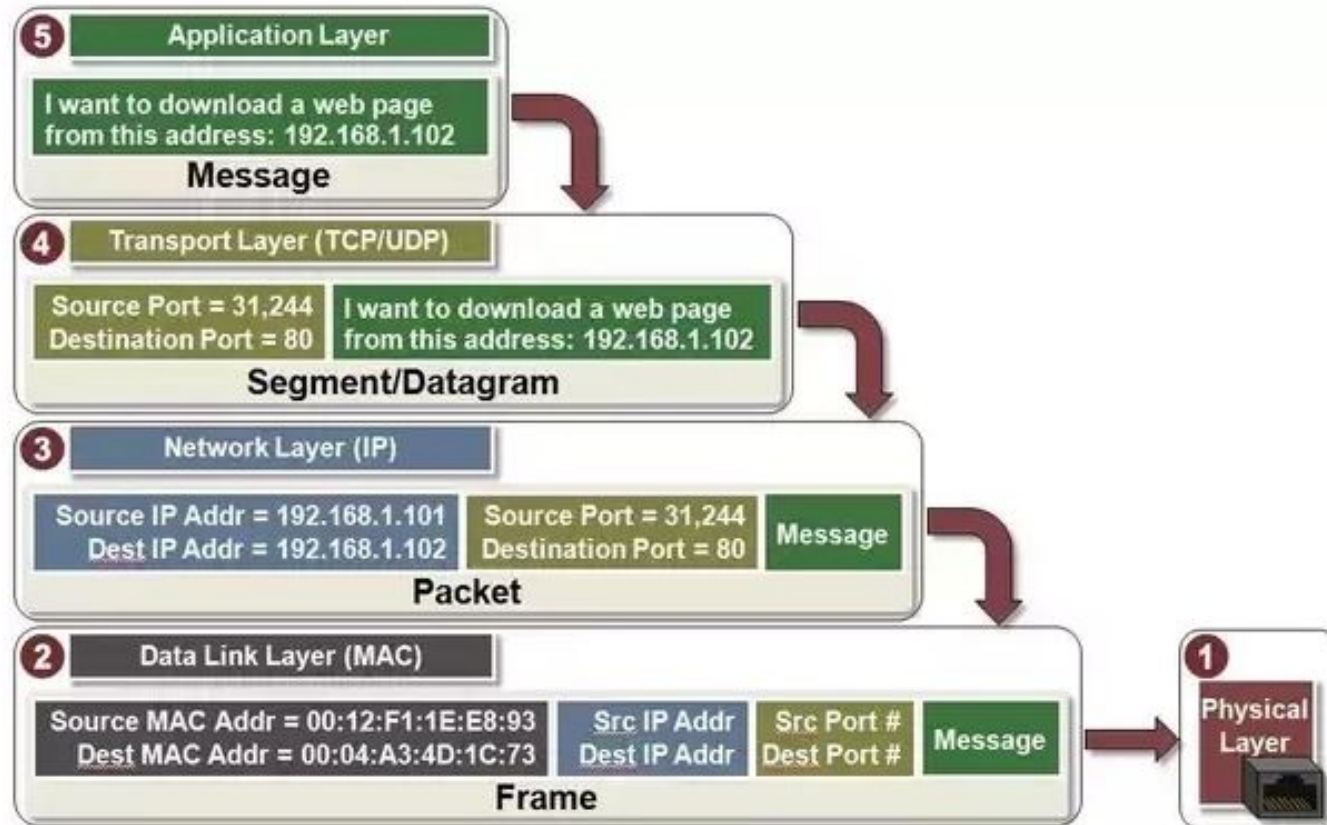
- **Full-duplex**: podem ser usados nos dois sentidos ao mesmo tempo, como uma estrada de mão dupla
- **Half-duplex**: usados em qualquer sentido, mas apenas um deles de cada vez, como uma linha férrea de trilho único
- **Simplex**: permitem o tráfego em apenas uma direção, como uma rua de mão única

- Na camada de enlace, a mensagem é chamada de quadro e encapsula o pacote enviado pela camada de rede









Relação entre as Camadas de Física e de Enlace

- A física fornece um fluxo de bits bruto para a de enlace
- Se o canal tiver ruído, a física pode inserir alguma redundância para reduzir a taxa de erro para um nível tolerável
- A enlace transforma o fluxo de bits bruto em quadros
- A enlace pode detectar e, eventualmente, corrigir erros

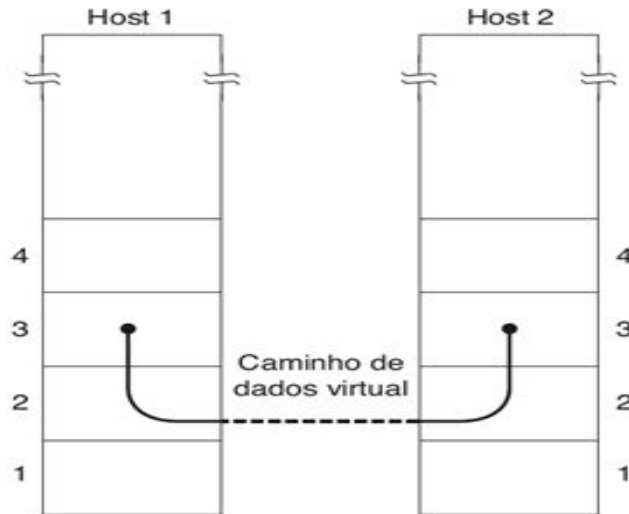
- Dado duas máquinas conectadas diretamente através de um canal de comunicação, podemos ter algumas limitações:
 - Os canais podem produzir erros
 - A taxa de dados é finita
 - O atraso de propagação é diferente de zero

Questões de Projeto da Camada de Enlace

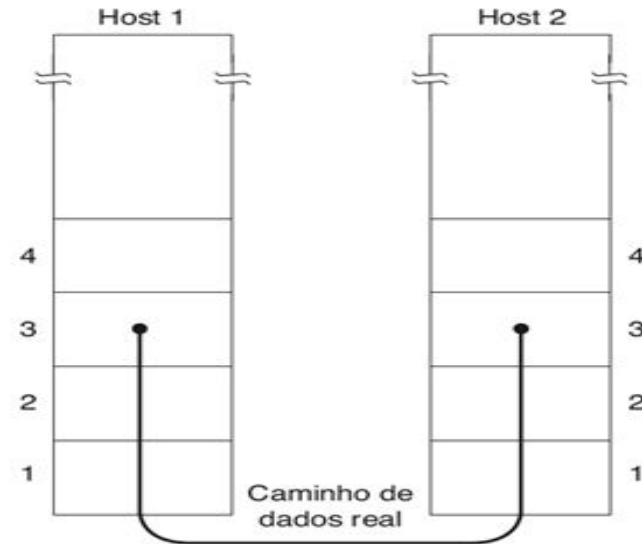
- Fornecer uma interface de serviços bem definida para a camada de rede
- Tarefas da camada de enlace
 - Enquadramento
 - Controle de erros: detecção ou correção
 - Controle de fluxo, permitindo que receptores mais lentos não sejam atropelados por transmissões rápidas

Principal Serviço da Camada de Enlace

- Transferência de dados da camada de rede de uma máquina origem para a mesma camada de uma máquina de destino



(a) Comunicação virtual



(b) Comunicação real

- Introdução
- **Orientação à Conexão e Existência de Confirmação**
- Enquadramento
- Controle de Fluxo
- Controle de Erro
- Camada de Enlace da Internet

Serviços Oferecidos à Camada de Rede

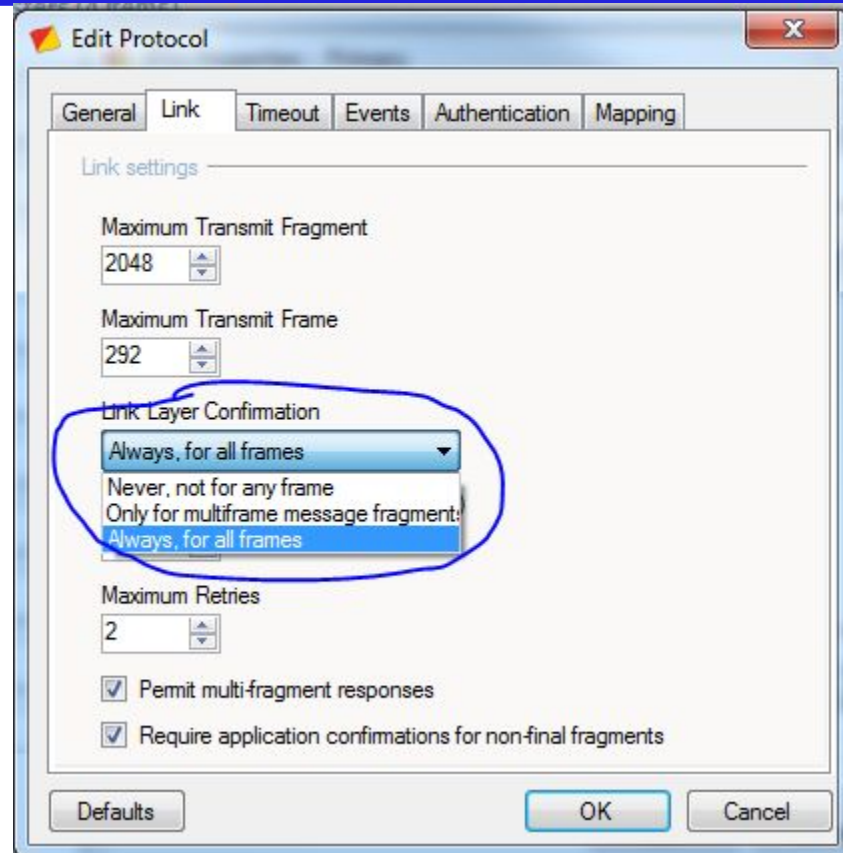
- Classificados quando a:
 - Orientação à conexão
 - Existência de confirmação

Orientação à Conexão

- A origem pode enviar quadros de forma numerada, garantindo a entrega e a ordem de entrega de cada um deles
- Adequada para canais longos ou não confiáveis (e.g., satélite ou circuitos telefônicos interurbano)
- Três fases:
 - estabelecimento da conexão
 - transmissão de um ou mais quadros
 - término da conexão

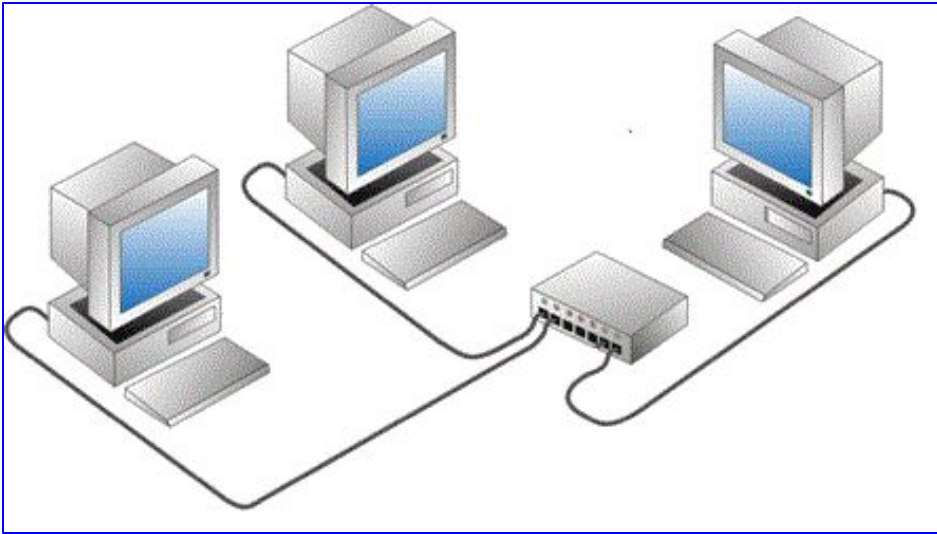
Existência de Confirmação

- Confirmação quadro a quadro



Existência de Confirmação

- É um *overhead* em canais confiáveis (e.g., fibra óptica) e interessante em não confiáveis (e.g., canais sem fio)

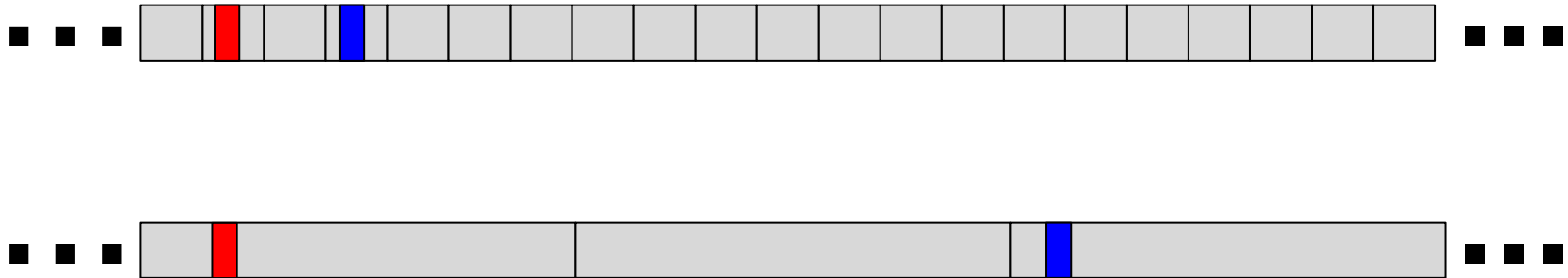


Existência de Confirmação

- Relacionada à questão de otimização (nunca uma exigência)
 - As camadas de rede ou transporte podem efetuar a confirmação, contudo, um pacote pode ser dividido em vários quadros
 - Por exemplo, se um pacote é dividido em 10 quadros e 20% dos quadros são perdidos, o tempo para reenviar o pacote é maior que o de dois quadros

Existência de Confirmação

- Conhecendo o tamanho dos quadros e parâmetros de atraso, a confirmação pode evitar a transmissão de dados já comprometidos



Serviços Oferecidos à Camada de Rede

- Serviço sem conexão nem confirmação
- Serviço sem conexão com confirmação
- Serviço com conexão e confirmação
- ~~Serviço com conexão e sem confirmação (não)~~

Serviço sem Conexão nem Confirmação

- Conexão não é estabelecida à priori
- Origem envia quadros independentes para o destino que não os confirma
- Quadros perdidos são ignorados

Serviço sem Conexão nem Confirmação

- Classe de serviço apropriada quando:
 - taxa de erros é “baixa”
 - alguma camada superior faz o processo de recuperação de erros
 - dados atrasados são piores que falhas (e.g., sistemas de tempo real (voz))
- Serviço normalmente usado em LANs

Serviço sem Conexão com Confirmação

- Conexão não é estabelecida a priori
- Destino confirma os quadros recebidos
- Origem usa mecanismo de temporização para reenviar quadros não confirmados
- Uma confirmação perdida pode acarretar diversas retransmissões de um quadro e, conseqüentemente, faça com que ele seja recebido várias vezes
- Serviço apropriado para canais não são confiáveis (e.g., comunicação sem fio)

Serviço com Conexão e Confirmação

- Serviço mais sofisticado
- Origem e destino estabelecem uma conexão antes da transmissão de dados
- Quadros recebidos corretamente
- Camada de enlace pode entregar os quadros em ordem para a de rede

- Introdução
- Orientação à Conexão e Existência de Confirmação
- **Enquadramento**
- Controle de Fluxo
- Controle de Erro
- Camada de Enlace da Internet

Enquadramento

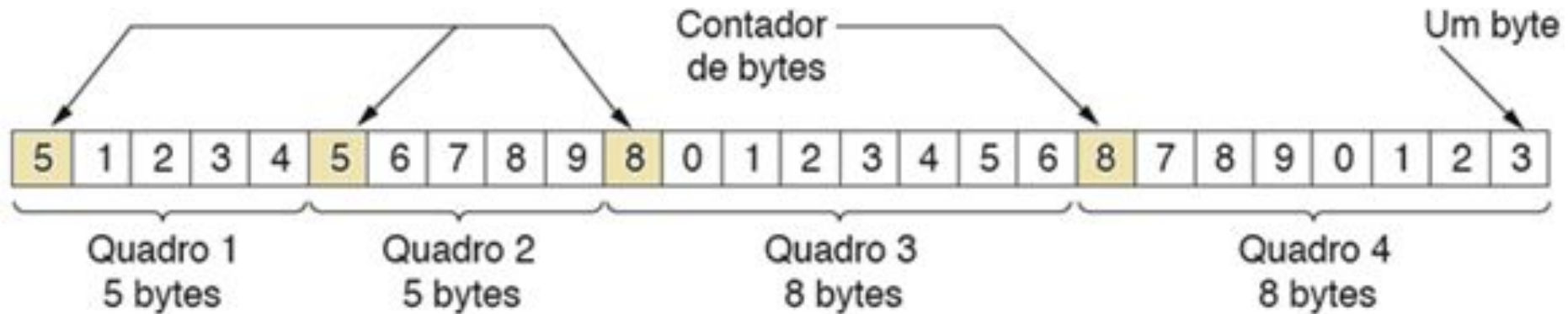
- É a divisão do fluxo de bits em quadros
 - Quando um nó envia um quadro, ele calcula e insere seu *checksum* no quadro
 - Quando um nó recebe um quadro, ele verifica o *checksum* desse quadro
- Permite que o receptor identifique o início de novos quadros consumindo pouco *overhead*

Técnicas para Enquadramento

- Contagem de *bytes*
- *Byte stuffing*
- *Bit stuffing*
- Violação de código da camada física

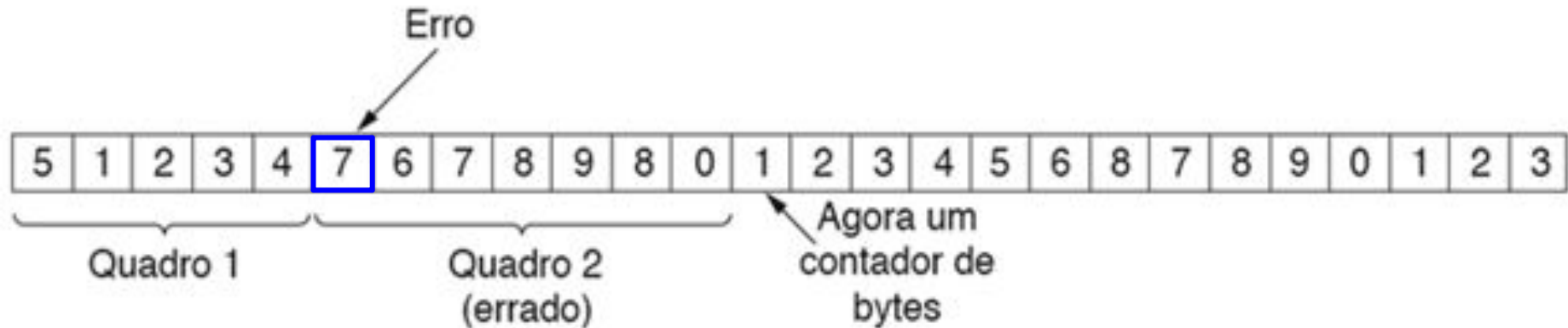
Contagem de Bytes

- Cada quadro terá um campo indicando seu número de bytes



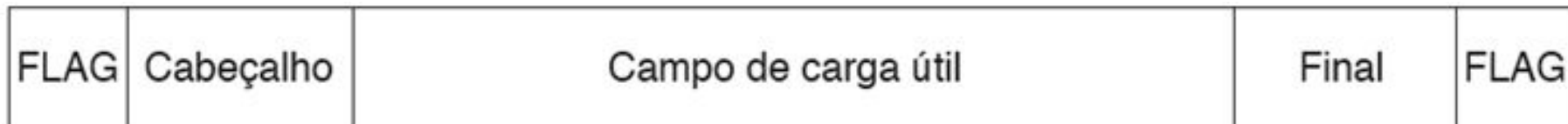
Contagem de Bytes

- Desvantagem: fortemente sensível a erros
 - No exemplo, um erro no contador do segundo quadro (se tornou 7), faz com que o receptor não identifique o início dos demais quadros



Byte Stuffing

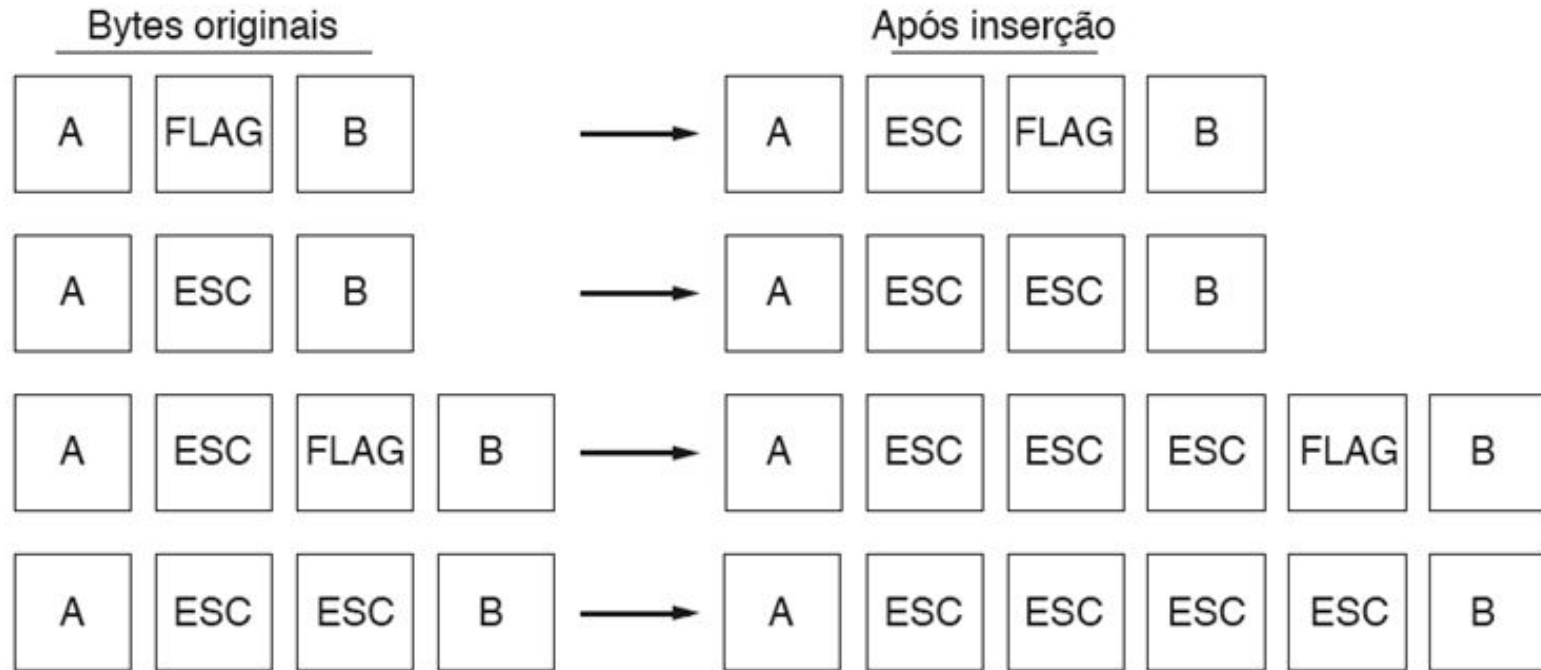
- Contorna o problema de ressincronização dos quadros (no caso de erros), inserindo bytes de flag para delimitar o início e fim dos quadros



Quadro limitado com bytes de flag

Byte Stuffing

- Quatro exemplos de sequências de bytes antes e depois do *byte stuffing*



Byte Stuffing

- Desvantagens:
 - *overhead* com a inserção de bytes que pode ser minimizado inserindo caracteres menos frequentes
 - depende da utilização de caracteres de 8 bits sendo que existem sistemas diferentes (e.g., o UNICODE emprega caracteres de 16 bits)

Exercício (1)

• Sabendo o caractere de flag é o @ e o de escape, #, quais serão as mensagens resultantes da aplicação do *byte stuffing* em:

- a) ABC
- b) A@C
- c) @@A@@
- d) @##A
- e) @DC##

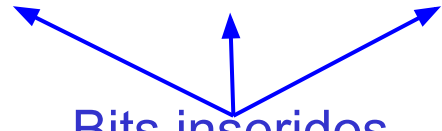
Bit Stuffing

- Minimiza o *overhead* do *byte stuffing*, permitindo flags com número arbitrário de bits
- Cada quadro começa e termina com o padrão 0111 1110 (0x7E)
- Quando o emissor identifica cinco bits com 1, ele insere um bit com 0 e o receptor efetua o processo contrário
- O comprimento do quadro depende de sua carga útil

Exemplo da Técnica de *Bit Stuffing*

Dados originais: 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

Dados com *bit stuffing* **7E** 0 1 1 0 1 1 1 1 1 **0** 1 1 1 1 1 **0** 1 1 1 1 1 **0** 1 0 0 1 0 **7E**



Bits inseridos

Dados armazenados pelo receptor 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

Violação de Código da Camada Física

- Uma das técnicas da camada física é a inserção de redundância, fazendo com que alguns sinais não ocorram em dados regulares
- Esses sinais podem ser explorados para início e fim de quadro pela camada de enlace
- Usado no padrão IEEE 802.11

Observações sobre as Técnicas de Enquadramento

- Protocolos de enlace, por segurança, podem usar uma ou mais técnicas
- O IEEE 802.11 e o Ethernet fazem com que cada quadro tenha um preâmbulo e seja seguido por um campo de comprimento

- Introdução
- Orientação à Conexão e Existência de Confirmação
- Enquadramento
- **Controle de Fluxo**
- Controle de Erro
- Camada de Enlace da Internet

Controle de Fluxo

- O que fazer quando um transmissor é mais rápido que o receptor?
- Protocolos de enlace normalmente consideram técnicas baseadas em:
 - *Feedback*: o receptor envia *feedbacks* sobre sua capacidade de processamento de quadros
 - Velocidade: o protocolo tem um mecanismo interno que limita a velocidade de transmissão sem usar *feedback*

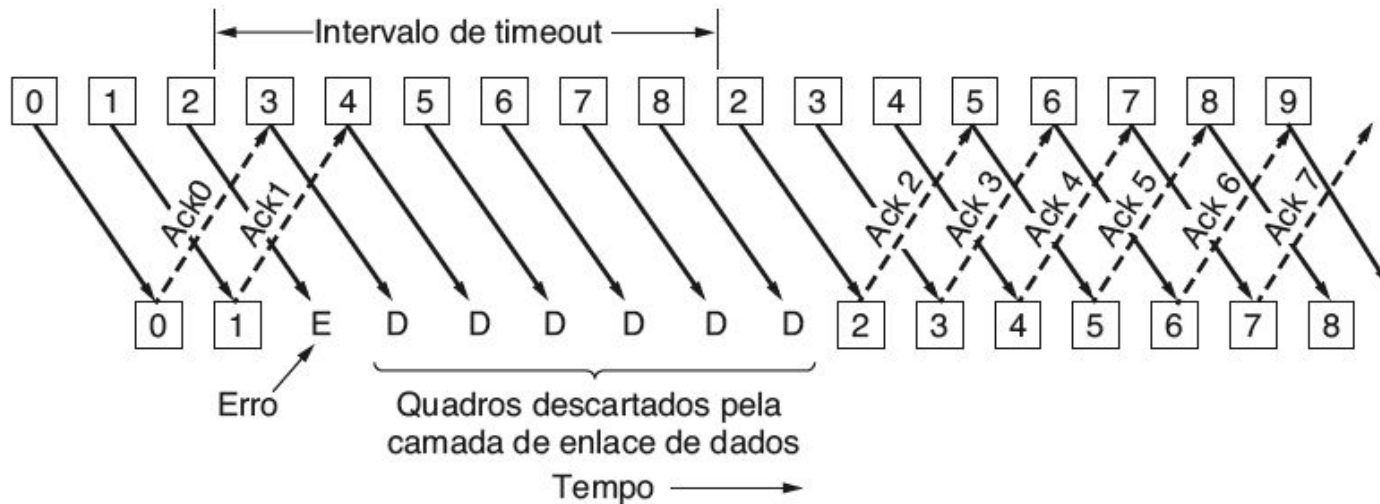
- Introdução
- Orientação à Conexão e Existência de Confirmação
- Enquadramento
- Controle de Fluxo
- **Controle de Erro**
- Camada de Enlace da Internet

Controle de Erros

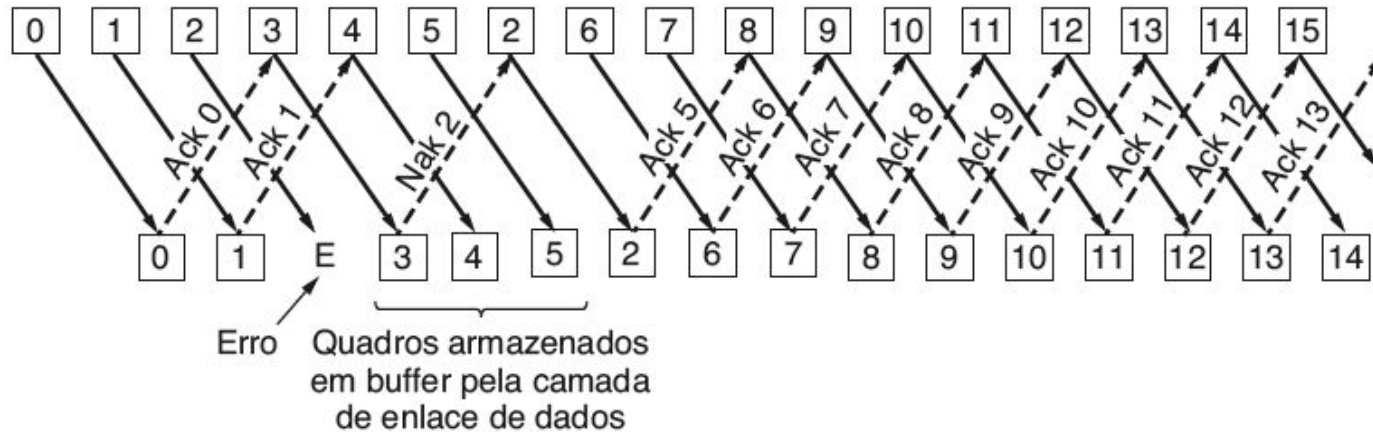
- Como garantir que os quadros serão entregues na camada de rede do destino e na ordem/forma correta?
 - Uma solução é não garantir
 - Outra é dar um *feedback* ao emissor

- Enviado pelo receptor, pode ser uma confirmação positiva ou negativa
 - Positivo: quando o receptor recebe um quadro, ele envia um ACK. Se o emissor não receber o ACK após um tempo, ele retransmite o quadro
 - Negativo: quando um nó recebe o quadro n sendo que ele não recebeu o $(n-1)$, o receptor envia um NACK (n)

Recuperação de Erros com ACK



Recuperação de Erros com NACK



Algumas Considerações

- Não existe mundo perfeito
- Os erros normalmente acontecem em rajadas o que minimiza a quantidade de blocos com erro, contudo, dificulta a detecção ou a correção dos erros
- A correção de erros é chamada de Correção Adiantada de Erros ou *Forward Error Correction* (FEC)

Inserção de Redundância nos Quadros

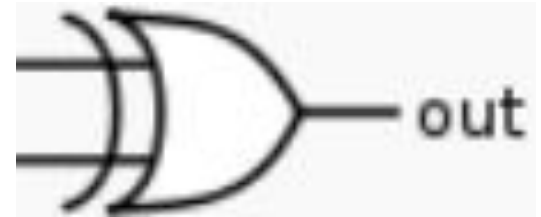
- Permite que o receptor seja capaz de detectar ou corrigir erros
- A correção normalmente demanda mais redundância que a de detecção
- Cada quadro é composto por $n = m + r$ bits, onde temos m bits de dados e r bits de redundância sendo r calculado a partir de m
 - Taxa de código: fração da palavra que representa os dados não redundantes

Estratégias para Inserção de Redundância

- **Estratégia sistemática**: os próprios m bits de dados são enviados
- **Estratégia linear**: os r bits de redundância são calculados como uma função (popularmente, a operação XOR) dos m de dados

Operação XOR

p	q	$p \oplus q$
F	F	F
F	V	V
V	F	V
V	V	F



Dependência do Meio

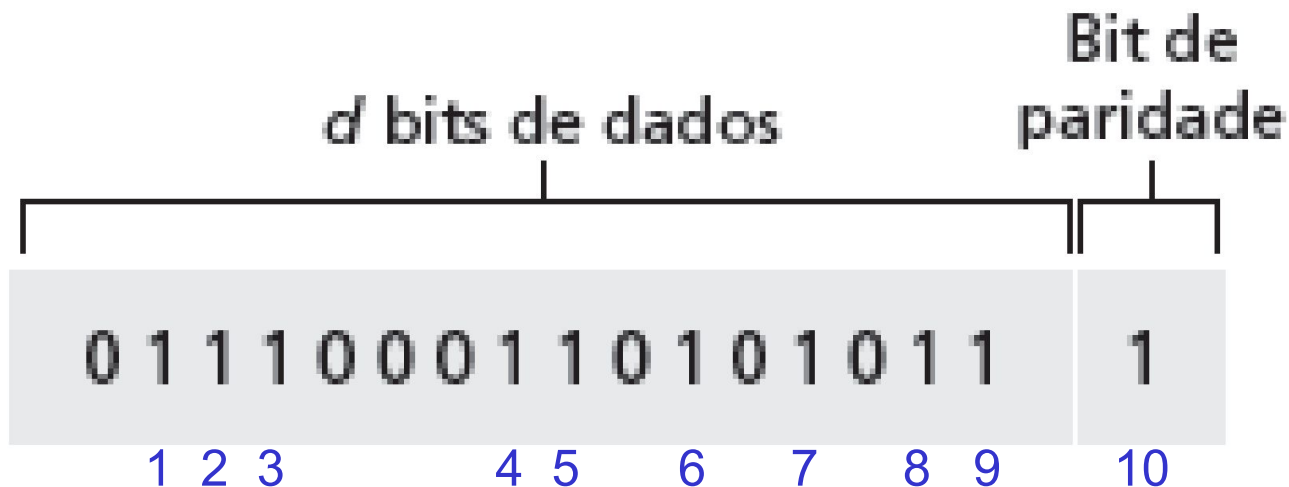
- Códigos de detecção de erros:
 - Usados em canais altamente confiáveis, como as fibras
 - Bloco defeituoso é retransmitido
- Códigos de correção de erros:
 - Usados em canais como enlaces sem fio que geram muitos erros

Algumas Técnicas para Detecção de Erros

- Verificação de paridade
- Checksum
- Verificação de redundância cíclica (CRC, *Cyclic Redundancy Check*)

Verificação de Paridade

- Insere um bit de paridade no quadro para garantir que o número total 1s seja par. No exemplo, o valor desse bit é 1, pois os dados têm um número ímpar de 1s



Esquemas de Paridade

- **Esquema de paridade par**: quando o número total 1s é par
- **Esquema de paridade ímpar**: quando o número total 1s é ímpar

Verificação Bidimensional de Paridade

- Enxerga o fluxo de bits como uma matriz e verifica a paridade de cada linha / coluna, aumentando o *overhead* e qualidade da verificação, e permitindo a correção de 1 bit

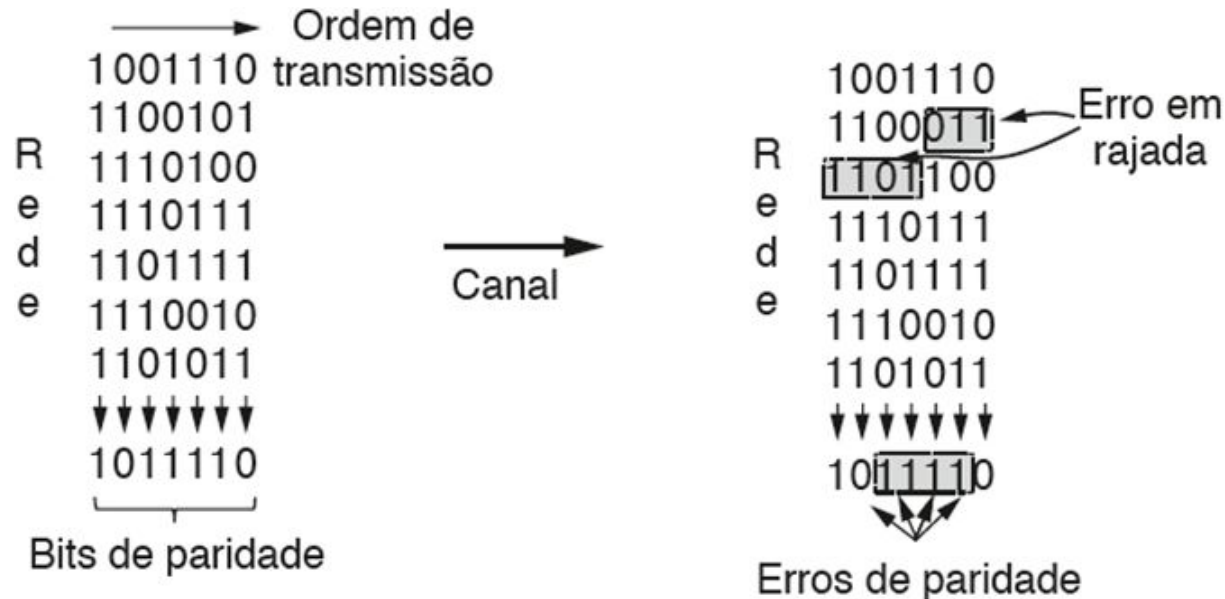
Nenhum erro							Erro de bit único corrigível						
1	0	1	0	1		1	1	0	1	0	1		1
1	1	1	1	0		0	1	0	1	1	0		0
0	1	1	1	0		1	0	1	1	1	0		1
<hr/>							<hr/>						
0	0	1	0	1		0	0	0	1	0	1		0

Erro de paridade

Erro de paridade

Verificação em Colunas de Paridade

- Solução intermediária entre as duas anteriores e que é robusta para identificar rajadas de erros (situação comum na transmissão de bits)



Checksum

- Termo normalmente usado para indicar um grupo de bits de verificação associados a uma mensagem, independente de como são calculados
- Na literatura, observa-se diversas variações dessa técnica
- Utilizado no protocolo IP (camada de rede)

Cyclic Redundancy Check (CRC)

- Tipo mais forte de detecção de erros
- Tem uso generalizado na camada de enlace
- Efetua a divisão de polinômios e faz com que o resto da divisão seja usado na detecção de erros

Representação usando Polinômio

- Considera que os **m** bits de dados são os coeficientes de um polinômio com **m** termos, variando desde **x^{m-1}** até **x^0** (grau **$m-1$**)
- Exemplo: **110001** representa o polinômio **$x^5 + x^4 + x^0$**
5 4 3 2 1 0

Polinômio Gerador - $G(x)$

- Definido antecipadamente por emissor e receptor
- Seus bits de maior e menor ordem sempre serão um (por definição)

Polinômios Geradores Comuns

CRC	$G(x)$
CRC-8	$x^8 + x^2 + x + 1$
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x + 1$
CRC-12	$x^{12} + x^{11} + x^3 + x^2 + 1$
CRC-16	$x^{16} + x^{15} + x^2 + 1$
CRC-CCITT	$x^{16} + x^{12} + x^5 + 1$
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

Fonte: PETERSON; DAVIE, 2004, p.70

Polinômios Geradores Comuns

- Redes Ethernet utilizam CRC-32
- Redes *Asynchronous Transfer Mode* (ATM) utilizam CRC-8, CRC-10 e CRC-32

Exercício (2)

• Mostre a sequência de bits dos polinômios geradores abaixo:

a) CRC-8: $x^8 + x^2 + x + 1$

b) CRC-10: $x^{10} + x^9 + x^5 + x^4 + x + 1$

Ideia Básica do CRC

- O emissor adiciona um conjunto de bits de controle no final do quadro de forma que o quadro (agora, dados + bits de controle) é divisível pelo polinômio gerador $G(x)$
- O receptor efetua a divisão do polinômio representado pelos bits do quadro pelo polinômio gerador $G(x)$ e se o resto dessa divisão for zero, temos sucesso, ou seja, não detecção de erros

Polinômio Dividendo - D(x)

$$\begin{array}{rclcl} \text{dividendo} & \leftarrow & \mathbf{6} & \big| \mathbf{3} & \rightarrow \text{divisor} \\ & & & \hline \text{resto} & \leftarrow & \mathbf{0} & \mathbf{2} & \rightarrow \text{quociente} \end{array}$$

Polinômio Dividendo - $D(x)$

- Produto da multiplicação do polinômio de dados da mensagem $M(x)$ por x^R onde R representa o maior grau do polinômio gerador (seu número de bits menos um)
- Por exemplo, se $M(x) = 10011010 = x^7 + x^4 + x^3 + x$ e $G(x) = 1101 = x^3 + x^2 + 1$, e $R = 3$ e o polinômio dividendo $D(x) = (x^7 + x^4 + x^3 + x) * (x^3) = (x^{10} + x^7 + x^6 + x^4) = 10011010000$
- Ou seja, o polinômio dividendo $D(x) = M(x)$ mais R zeros inseridos à direita

Aritmética Polinomial

- É feita em módulo 2 segundo as regras da teoria algébrica, logo, não há transportes para adição nem empréstimos para subtração
- Nesse caso, tanto a adição como a subtração são como o XOR

$$\begin{array}{r} 10011011 \\ + 11001010 \\ \hline 01010001 \end{array}$$

$$\begin{array}{r} 00110011 \\ + 11001101 \\ \hline 11111110 \end{array}$$

$$\begin{array}{r} 11110000 \\ - 10100110 \\ \hline 01010110 \end{array}$$

$$\begin{array}{r} 01010101 \\ - 10101111 \\ \hline 11111010 \end{array}$$

- A divisão será realizada através de várias subtrações

Polinômio a ser Transmitido $T(x)$

- $T(x) = D(x)$ **or** resto da divisão de $D(x)$ por $G(x)$

- Seja $M(x) = 10011010$ e $G(x) = 1101$, calcule $T(x)$
 - 1º passo: calcular o polinômio dividendo $D(x)$
 - 2º passo: encontrar o resto da divisão de $D(x)$ pelo polinômio gerador $G(x)$
 - 3º passo: $D(x)$ or com o resto da divisão de $D(x)$ por $G(x)$

• Seja $M(x) = 10011010$ e $G(x) = 1101$, calcule $T(x)$

- **1º passo: calcular o polinômio dividendo $D(x)$**

$$D(x) = 10011010\mathbf{000}$$

- 2º passo: encontrar o resto da divisão de $D(x)$ pelo polinômio gerador $G(x)$
- 3º passo: $D(x)$ or com o resto da divisão de $D(x)$ por $G(x)$

Exemplo

• Seja $M(x) = 10011010$ e $G(x) = 1101$, calcule $T(x)$

- 1º passo: calcular o polinômio dividendo $D(x)$

$$D(x) = 10011010000$$

- **2º passo: encontrar o resto da divisão de $D(x)$ pelo polinômio gerador $G(x)$**
- 3º passo: $D(x)$ or com o resto da divisão de $D(x)$ por $G(x)$

Exemplo: Divisão de $D(x)$ por $G(x)$

10011010000 1101

Exemplo: Divisão de $D(x)$ por $G(x)$

$$\begin{array}{r}
 10011010000 \mid 1101 \\
 \text{xor } 1101 \\
 \hline
 100
 \end{array}$$

Exemplo: Divisão de $D(x)$ por $G(x)$

$$\begin{array}{r}
 10011010000 \mid 1101 \\
 \text{xor } 1101 \\
 \hline
 1001
 \end{array}$$

The diagram illustrates the first step of polynomial division. The dividend is 10011010000 and the divisor is 1101. The first four bits of the dividend (1001) are aligned under the divisor. The operation 'xor 1101' is performed on these bits, resulting in 1001. The next bit of the dividend (1) is brought down to the right of the result.

Exemplo: Divisão de $D(x)$ por $G(x)$

$$\begin{array}{r}
 10011010000 \mid 1101 \\
 \text{xor } 1101 \\
 \hline
 1001 \\
 \text{xor } 1101 \\
 \hline
 100
 \end{array}$$

Exemplo: Divisão de $D(x)$ por $G(x)$

$$\begin{array}{r}
 10011010000 \mid 1101 \\
 \text{xor } 1101 \\
 \hline
 1001 \\
 \text{xor } 1101 \\
 \hline
 1000
 \end{array}$$

Exemplo: Divisão de $D(x)$ por $G(x)$

$$\begin{array}{r}
 10011010000 \mid 1101 \\
 \text{xor } 1101 \\
 \hline
 1001 \\
 \text{xor } 1101 \\
 \hline
 1000 \\
 \text{xor } 1101 \\
 \hline
 101
 \end{array}$$

Exemplo: Divisão de $D(x)$ por $G(x)$

$$\begin{array}{r}
 10011010000 \mid 1101 \\
 \text{xor } 1101 \\
 \hline
 1001 \\
 \text{xor } 1101 \\
 \hline
 1000 \\
 \text{xor } 1101 \\
 \hline
 1011
 \end{array}$$

Exemplo: Divisão de $D(x)$ por $G(x)$

$$\begin{array}{r}
 10011010000 \mid 1101 \\
 \text{xor } 1101 \\
 \hline
 1001 \\
 \text{xor } 1101 \\
 \hline
 1000 \\
 \text{xor } 1101 \\
 \hline
 1011 \\
 \text{xor } 1101 \\
 \hline
 110
 \end{array}$$

Exemplo: Divisão de $D(x)$ por $G(x)$

```

10011010000 | 1101
xor 1101      1111
1001
xor 1101
1000
xor 1101
1011
xor 1101
1100

```

Exemplo: Divisão de $D(x)$ por $G(x)$

$$\begin{array}{r}
 10011010000 \mid 1101 \\
 \text{xor } 1101 \\
 \hline
 1001 \\
 \text{xor } 1101 \\
 \hline
 1000 \\
 \text{xor } 1101 \\
 \hline
 1011 \\
 \text{xor } 1101 \\
 \hline
 1100 \\
 \text{xor } 1101 \\
 \hline
 001
 \end{array}$$

Exemplo: Divisão de $D(x)$ por $G(x)$

$$\begin{array}{r}
 1001101000 \text{ } | \text{ } 1101 \\
 \text{xor } 1101 \\
 \hline
 1001 \\
 \text{xor } 1101 \\
 \hline
 1000 \\
 \text{xor } 1101 \\
 \hline
 1011 \\
 \text{xor } 1101 \\
 \hline
 1100 \\
 \text{xor } 1101 \\
 \hline
 0010
 \end{array}$$

Exemplo: Divisão de D(x) por G(x)

$$\begin{array}{r}
 1001101000 \text{ } 00 \mid 1101 \\
 \text{xor } 1101 1111100 \\
 1001 \\
 \text{xor } 1101 \\
 1000 \\
 \text{xor } 1101 \\
 1011 \\
 \text{xor } 1101 \\
 1100 \\
 \text{xor } 1101 \\
 00100
 \end{array}$$

Exemplo: Divisão de $D(x)$ por $G(x)$

$$\begin{array}{r}
 10011010000 \text{ } 0 \quad | \quad 1101 \\
 \text{xor } 1101 \quad \quad \quad 11111000 \\
 \hline
 1001 \\
 \text{xor } 1101 \\
 \hline
 1000 \\
 \text{xor } 1101 \\
 \hline
 1011 \\
 \text{xor } 1101 \\
 \hline
 1100 \\
 \text{xor } 1101 \\
 \hline
 001000 \text{ } 0
 \end{array}$$

Exemplo: Divisão de D(x) por G(x)

$$\begin{array}{r}
 10011010000 \text{ } 0 \quad | \quad 1101 \\
 \text{xor } 1101 \quad \quad \quad 111110001 \\
 \hline
 1001 \\
 \text{xor } 1101 \\
 \hline
 1000 \\
 \text{xor } 1101 \\
 \hline
 1011 \\
 \text{xor } 1101 \\
 \hline
 1100 \\
 \text{xor } 1101 \\
 \hline
 001000 \\
 \text{xor } 1101 \\
 \hline
 101
 \end{array}$$

Exemplo: Divisão de D(x) por G(x)

$$\begin{array}{r}
 10011010000 \mid 1101 \\
 \underline{\text{xor } 1101} \\
 1001 \\
 \underline{\text{xor } 1101} \\
 1000 \\
 \underline{\text{xor } 1101} \\
 1011 \\
 \underline{\text{xor } 1101} \\
 1100 \\
 \underline{\text{xor } 1101} \\
 001000 \\
 \underline{\text{xor } 1101} \\
 101 \leftarrow \text{resto}
 \end{array}$$

Exemplo

• Seja $M(x) = 10011010$ e $G(x) = 1101$, calcule $T(x)$

- 1º passo: calcular o polinômio dividendo $D(x)$

$$D(x) = 10011010000$$

- **2º passo: encontrar o resto da divisão de $D(x)$ pelo polinômio gerador $G(x)$**

$$\text{resto} = 101$$

- 3º passo: $D(x)$ or com o resto da divisão de $D(x)$ por $G(x)$

• Seja $M(x) = 10011010$ e $G(x) = 1101$, calcule $T(x)$

- 1º passo: calcular o polinômio dividendo $D(x)$

$$D(x) = 10011010000$$

- 2º passo: encontrar o resto da divisão de $D(x)$ pelo polinômio gerador $G(x)$

$$\text{resto} = 101$$

- 3º passo: $D(x)$ or com o resto da divisão de $D(x)$ por $G(x)$

$$T(x) = 10011010101$$

Ideia Básica do CRC

- O emissor adiciona um conjunto de bits de controle no final do quadro de forma que o quadro (agora, dados + bits de controle) é divisível pelo polinômio gerador $G(x)$
- O receptor efetua a divisão do polinômio representado pelos bits do quadro pelo polinômio gerador $G(x)$ e se o resto dessa divisão for zero, temos sucesso, ou seja, não detecção de erros

Exemplo na Recepção

- Como $T(x) = 10011010101$ e $G(x) = 1101$, temos:

Exemplo na Recepção

10011010101		1101
<u>xor 1101</u>		xxxxxxxx
1001		
<u>xor 1101</u>		
1000		
<u>xor 1101</u>		
1011		
<u>xor 1101</u>		
1100		
<u>xor 1101</u>		
001101		
<u>xor 1101</u>		
000		

000 ← resto

Exercício (3)

- Seja $M(x) = 111100101$ e $G(x) = 101101$, calcule $T(x)$
 - 1º passo: calcular o polinômio dividendo $D(x)$
 - 2º passo: encontrar o resto da divisão de $D(x)$ pelo polinômio gerador $G(x)$
 - 3º passo: $D(x)$ or com o resto da divisão de $D(x)$ por $G(x)$

Exercício (4)

- Seja $R(x) = 11110010101010$ um quadro recebido e o polinômio gerador $G(x) = 101101$, informe se conseguimos identificar algum erro usando CRC

Exercício (5)

- Seja $M(x) = 1101011011$ e $G(x) = 10011$, calcule $T(x)$
 - 1o passo: calcular o polinômio dividendo $D(x)$
 - 2o passo: encontrar o resto da divisão de $D(x)$ pelo polinômio gerador $G(x)$
 - 3o passo: $D(x)$ or com o resto da divisão de $D(x)$ por $G(x)$

Exercício (6)

- Qual é o resto obtido pela divisão módulo 2 de x^7+x^5+1 pelo polinômio gerador x^3+1 ?

Exercício (7)

- Um fluxo de bits 10011101 é transmitido com a utilização do método de CRC padrão descrito no texto. O polinômio gerador é x^3+1 . Mostre o string de bit real transmitido. Suponha que o terceiro bit a partir da esquerda seja invertido durante a transmissão. Mostre que esse erro é detectado na extremidade receptora.

Algumas Técnicas para Correção de Erros

- Código de Hamming
- Código Convolutacional Binário
- Outros

Código de Hamming

(1) Recodificamos a mensagem de tal forma que os bits potência de dois (1, 2, 4, 8, 16, ...) serão bits de verificação

Exemplo: mensagem = 10101

b1	b2	b3	b4	b5	b6	b7	b8	b9
—	—	1	—	0	1	0	—	1

Nesse caso, observamos que os bits de verificação são: b1, b2, b4 e b8

Código de Hamming

(2) Na nova mensagem (ainda incompleta), selecionamos os bits com 1 e efetuamos um XOR entre os valores de suas respectivas posições

b1	b2	b3	b4	b5	b6	b7	b8	b9
—	—	1	—	0	1	0	—	1

Como os bits com valor um são b3, b6 e b9:

Pos.	Binário
3	0 0 1 1
6	0 1 1 0
9	1 0 0 1

XOR	1 1 0 0

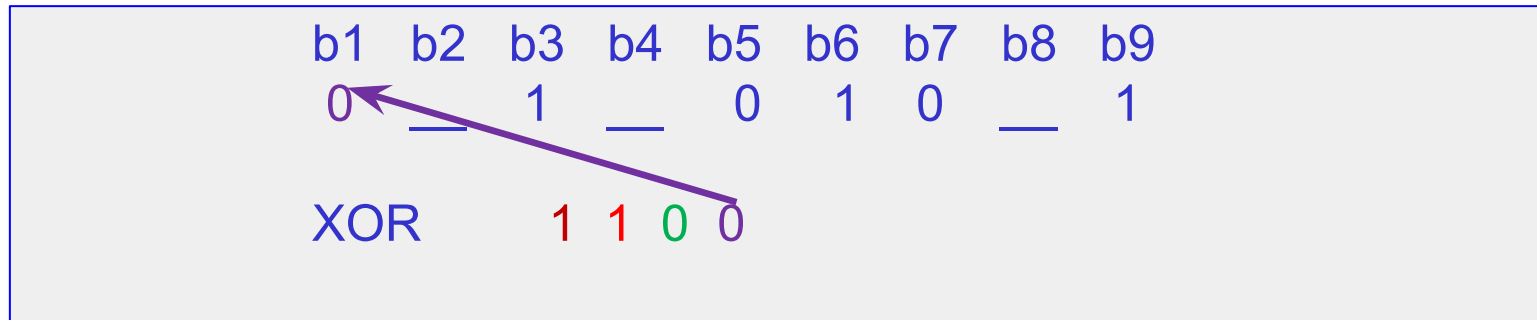
Código de Hamming

(3) Completamos a mensagem a ser enviada com os valores do XOR na ordem inversa

b1	b2	b3	b4	b5	b6	b7	b8	b9
—	—	1	—	0	1	0	—	1
XOR		1	1	0	0			

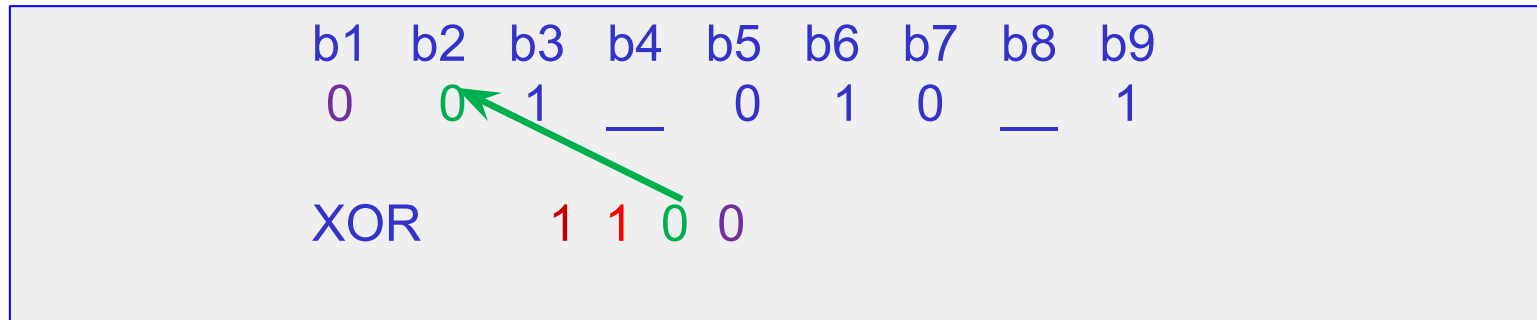
Código de Hamming

(3) Completamos a mensagem a ser enviada com os valores do XOR na ordem inversa



Código de Hamming

(3) Completamos a mensagem a ser enviada com os valores do XOR na ordem inversa



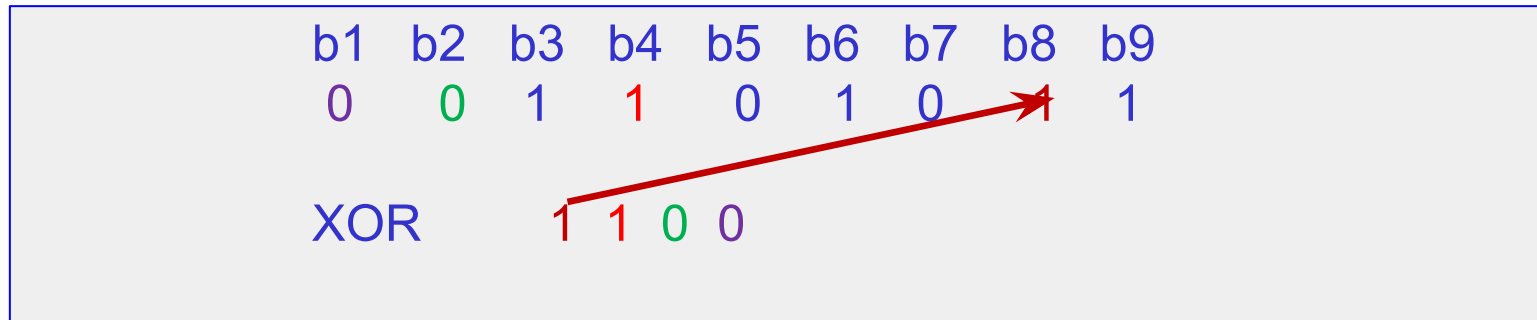
Código de Hamming

(3) Completamos a mensagem a ser enviada com os valores do XOR na ordem inversa

	b1	b2	b3	b4	b5	b6	b7	b8	b9
	0	0	1	1	0	1	0	—	1
XOR			1	1	0	0			

Código de Hamming

(3) Completamos a mensagem a ser enviada com os valores do XOR na ordem inversa



Código de Hamming

(3) Completamos a mensagem a ser enviada com os valores do XOR na ordem inversa

Mensagem a ser enviada:

0 0 1 1 0 1 0 1 1

Código de Hamming

(4) No recebimento da mensagem, efetuamos um XOR com as posições cujo valor seja igual a 1

Mensagem recebida:

0	0	1	1	0	1	0	1	1
b1	b2	b3	b4	b5	b6	b7	b8	b9

Pos.	Binário
3	0 0 1 1
4	0 1 0 0
6	0 1 1 0
8	1 0 0 0
9	1 0 0 1
<hr/>	
XOR	0 0 0 0

Código de Hamming

(5) Se a resposta do XOR for zero, o código de Hamming não detectou qualquer erro

Mensagem recebida:

XOR 0 0 0 0

Código de Hamming

(6) Se a resposta do XOR for diferente de zero, o código de Hamming detectou um erro no bit indicado pelo XOR

Mensagem recebida com erro na posição:

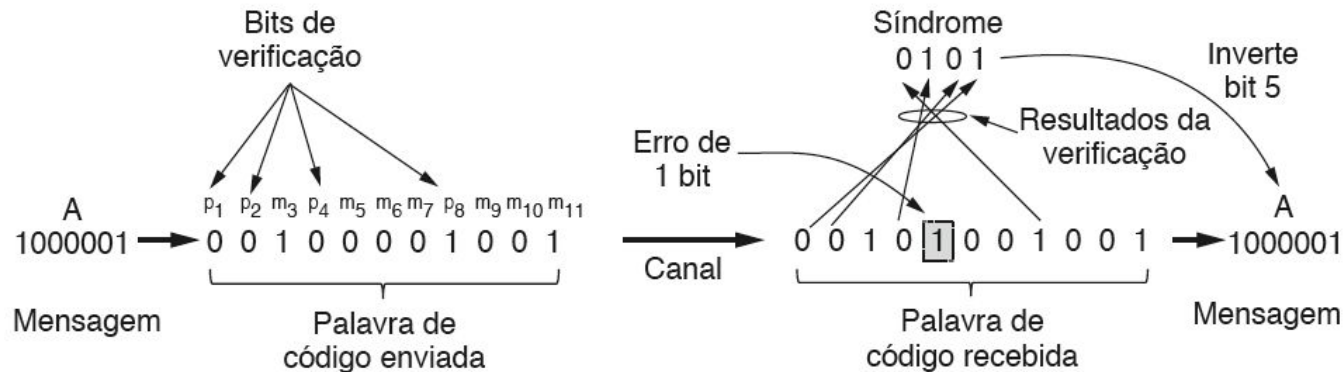
1	0	1	1	0	1	0	1	1
b1	b2	b3	b4	b5	b6	b7	b8	b9

Pos.	Binário
1	0 0 0 1
3	0 0 1 1
4	0 1 0 0
6	0 1 1 0
8	1 0 0 0
9	1 0 0 1

XOR	0 0 0 1
-----	---------

Exercício (8)

- Mostre que a figura abaixo está correta



Exercício (9)

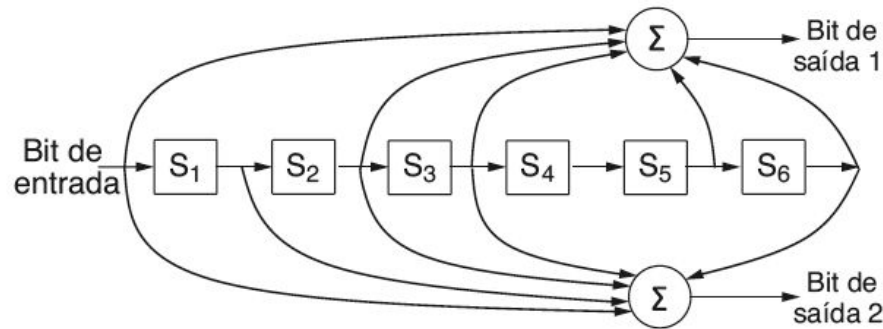
- Um usuário deseja enviar uma mensagem contendo a letra 'G', como ficarão os bits dessa mensagem aplicando o Código de Hamming

Código Convolutacional Binário

- Utilizado pela NASA, ele converte cada bit de entrada em dois de saída, conforme a figura abaixo
- Para cada bit de entrada, suas saídas dependem dos últimos seis bits
- Para decodificação, utilizamos algoritmos probabilísticos que encontram a sequência de bits que tem a maior probabilidade de ser a original

Código Convolutivo Binário

- Para os primeiros bits, considera-se que os anteriores eram 000000, assim, por exemplo, se a entrada for 111, a saída será 111001



•Saída 1: $b_1 \oplus b_3 \oplus b_4 \oplus b_6 \oplus b_7$

•Saída 2: $b_1 \oplus b_2 \oplus b_3 \oplus b_4 \oplus b_7$

Exercício (10)

- Quais são as funções da camada de enlace. Explique cada uma delas.

- Introdução
- Orientação à Conexão e Existência de Confirmação
- Enquadramento
- Controle de Fluxo
- Controle de Erro
- **Camada de Enlace da Internet**

Camada de Enlace da Internet

- A Internet consiste em *hosts* e roteadores conectados através de uma infraestrutura de comunicação
- Boa parte dessa infraestrutura são linhas dedicadas **ponto a ponto**



Point-to-Point Protocol (PPP)

- Protocolo de enlace usado nas comunicações ponto-a-ponto da Internet
- Definido na RFC 1661 e atualizado nas RFCs 1662 e 1663
- Detecta erros
- Aceita vários protocolos
- Permite que endereços IP sejam negociados em tempo de conexão
- Permite a autenticação

- Método de enquadramento não ambíguo
- Protocolo de Controle de Enlace (LCP, *Link Control Protocol*) para ativar linhas, testá-las, negociar opções e desativá-las
- Protocolo de Controle de Rede (NCP, *Network Control Protocol*) para negociar as opções da camada de rede independente do protocolo de rede usado

Exemplo de uma conexão PPP

- Computador A chama o roteador de um provedor através de um modem
- Conexão física é estabelecida
- Quadros LCP são trocados e os parâmetros PPP são selecionados
- Quadros NCP são trocados e os parâmetros de rede são selecionados (e.g., número IP é assinalado a A)
- Conexão definida e A é visto como um computador estático da rede
- Finalização: NCP termina conexão de rede, liberando o número IP
- LCP termina conexão de enlace
- Conexão física é terminada

Exercício (11)

- Explique o funcionamento das técnicas de enquadramento Contagem de bytes, *byte stuffing* e *bit stuffing*.

Exercício (12)

- Explique o funcionamento da técnica de correção de erro denominada matriz de paridade

Exercício (13)

- Explique a recuperação de erros com ACK e NACK