

Redes de Computadores

Camada de Enlace de Dados

Kalinka Regina Lucas Jaquie Castelo Branco

Camada de Enlace

Aplicação
Apresentação
Sessão
Transporte
Rede
Enlace
Físico

Função da Camada de Enlace

- ❑ Transferir dados da camada de rede da máquina fonte para a camada de rede da máquina destino.
- ❑ Essa transferência é feita nó a nó por um único enlace do caminho, de forma que os dados poderão ser manipulados por diferentes protocolos de enlace nos diferentes enlaces do caminho.

A Camada de Enlace

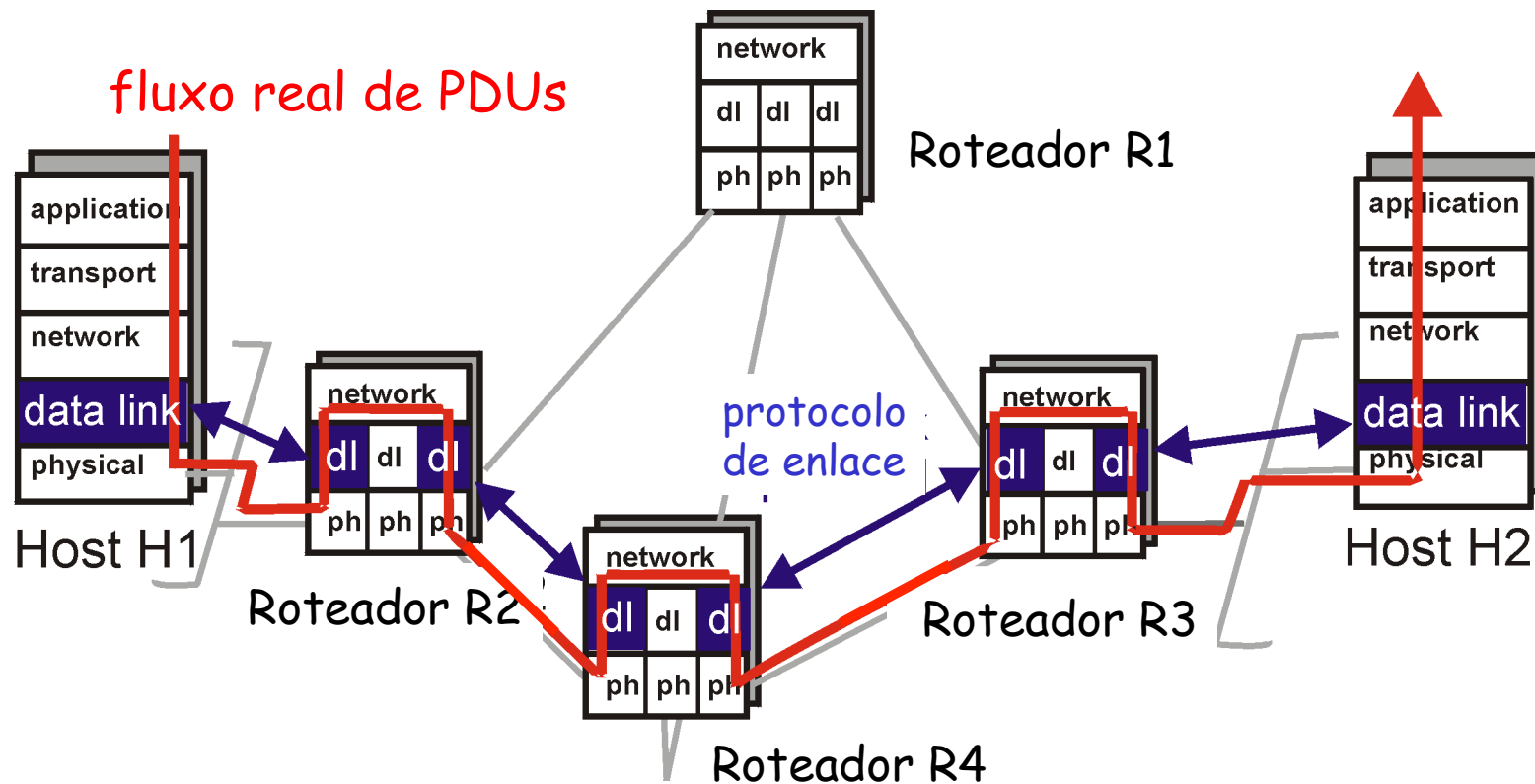
Nossos objetivos:

- ❑ entender os princípios por trás dos serviços da camada de enlace:
 - detecção de erros, correção
 - compartilhando um canal broadcast: acesso múltiplo
 - endereçamento da camada de enlace
 - transferência de dados confiável, controle de fluxo
- ❑ instanciação e implementação de várias tecnologias da camada de enlace

Visão Geral:

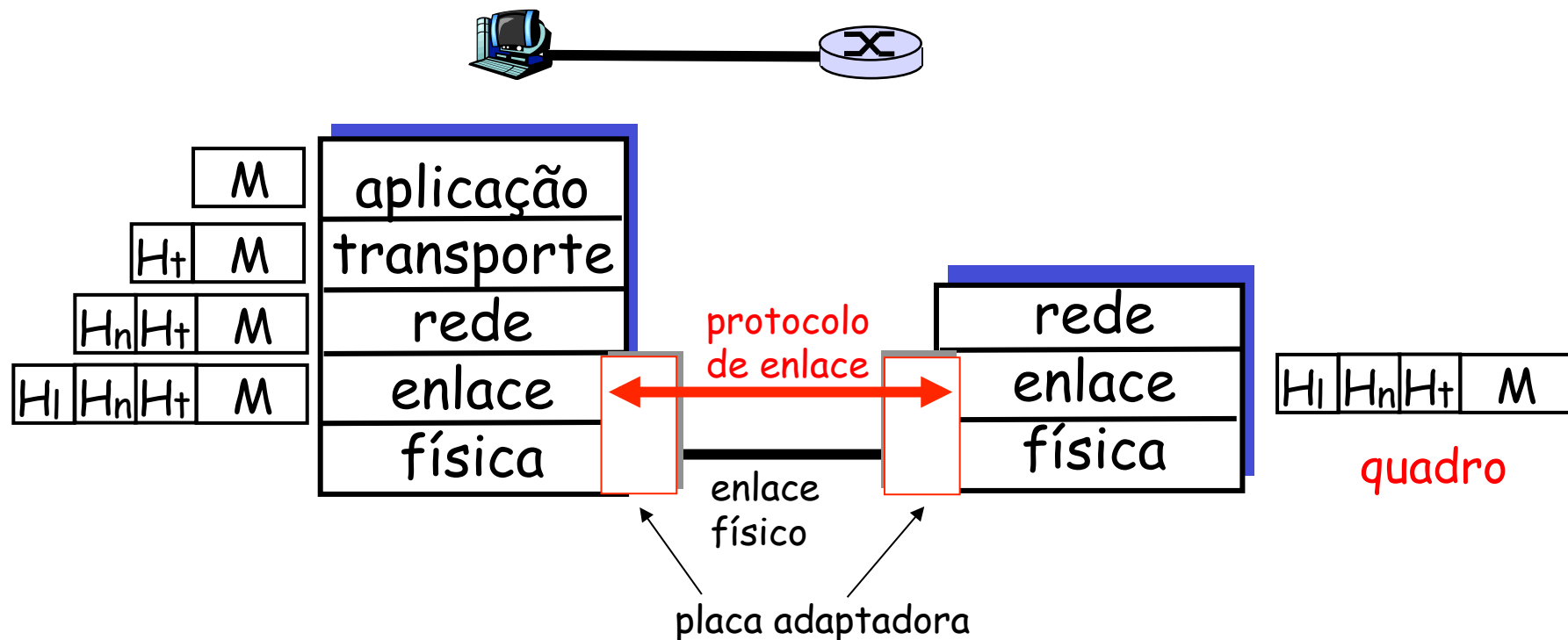
- ❑ serviços da camada de enlace
- ❑ detecção de erros, correção
- ❑ protocolos de acesso múltiplo e LANs
- ❑ endereçamento da camada de enlace, ARP
- ❑ tecnologias específicas da camada de enlace:
 - Ethernet
 - hubs, pontes, switches
 - IEEE 802.11 LANs
 - PPP
 - ATM

Camada de enlace: definindo o contexto



Camada de enlace: definindo o contexto

- ❑ dois elementos físicos *fisicamente conectados*:
 - host-roteador, roteador-roteador, host-host
- ❑ unidade de dados: *quadro (frame)*



Serviços da Camada de Enlace

- ❑ **Enquadramento, acesso ao enlace:**
 - encapsula datagramas em quadros, acrescentando cabeçalhos e trailer
 - implementa acesso ao canal se o meio é compartilhado
 - ‘endereços físicos’ usados nos cabeçalhos dos quadros para identificar a fonte e o destino dos quadros
- ❑ **Entrega confiável entre dois equipamentos fisicamente conectados:**
 - raramente usado em enlaces com baixa taxa de erro (fibra, alguns tipos de par trançado)
 - enlaces sem-fio (wireless): altas taxas de erro
 - Q: porque prover confiabilidade fim-a-fim e na camada de enlace?

Protocolos da Camada de Enlace

- ❑ Como exemplos de protocolos da camada de enlace; temos:
 - Ethernet;
 - Token Ring;
 - FDDI e;
 - PPP.

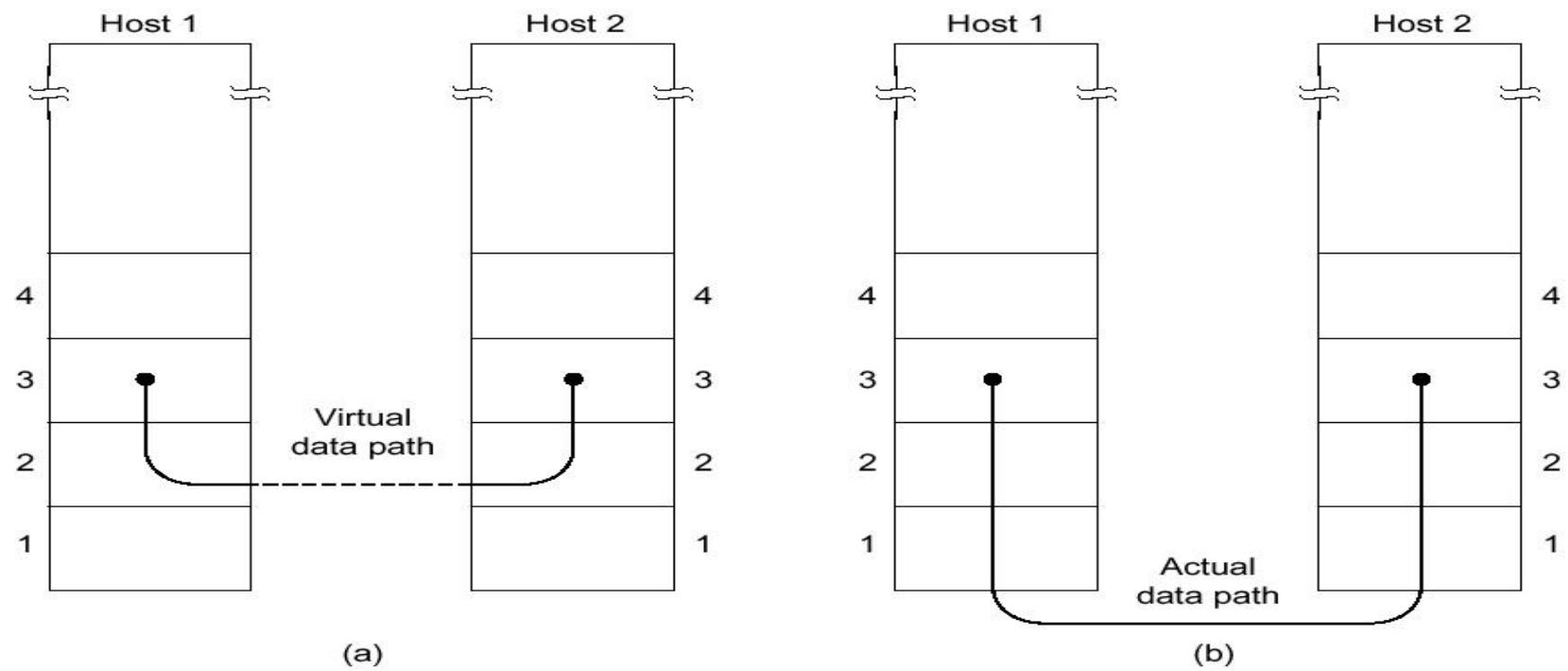


Fig. 3-1. (a) Virtual communication. (b) Actual communication.

Principais Serviços

- ❑ Delimitação de Quadros (Framing)
- ❑ Controle de Erros
- ❑ Controle de Fluxo

Serviços da Camada de Enlace (cont.)

❑ *Controle de Fluxo:*

- limitação da transmissão entre transmissor e receptor

❑ *Detecção de Erros:*

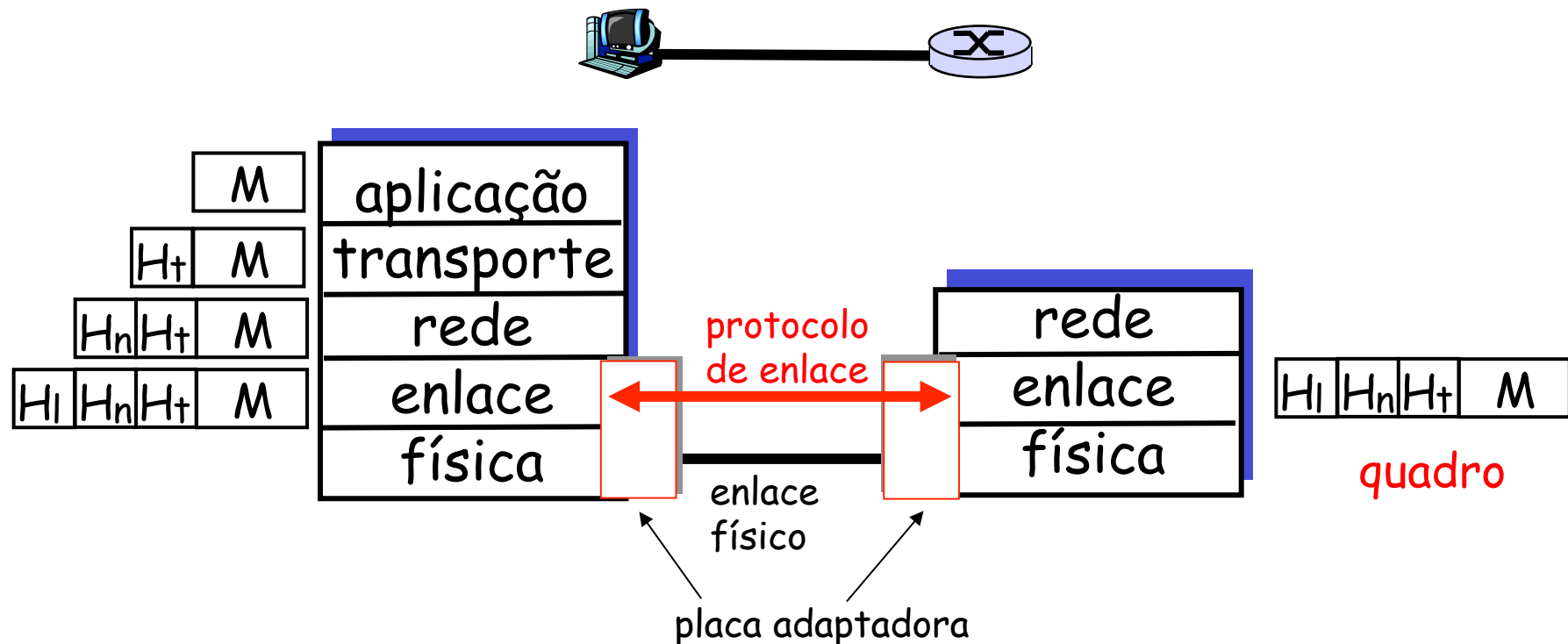
- erros causados pela atenuação do sinal e por ruídos.
- o receptor detecta a presença de erros:
 - avisa o transmissor para reenviar o quadro perdido

❑ *Correção de Erros:*

- o receptor identifica *e corrige* o bit com erro(s) sem recorrer à retransmissão

Implementação: Camada de Enlace

- ❑ Protocolo implementado no “adaptador”
 - ex., placa PCMCIA, placa Ethernet
 - tipicamente inclui: RAM, chips DSP, interface com barramento do host, e interface do enlace



Tipos de Serviços

- ❑ Os serviços são fornecidos nas seguintes combinações:
 - Serviço sem conexão não confirmado
 - Serviço sem conexão confirmado
 - Serviço orientado a conexão confirmado

Serviço sem conexão não confirmado

- ❑ A máquina emissora envia frames sem receber confirmação de recebimento da máquina receptora.
- ❑ Adequado para aplicações onde a taxa de erro é muito baixa.
- ❑ Também para tempo real onde dados atrasados são piores que dados ruins.
- ❑ Conexão não é estabelecida a priori
- ❑ Quadros independentes são enviados da origem para o destino que não envia nenhuma confirmação de volta.
- ❑ Quadros perdidos são ignorados e tratados pelas camadas superiores.
- ❑ Serviço normalmente usado em LANs

Serviço sem conexão confirmado

- ❑ Conexão não é estabelecida a priori.
- ❑ Cada frame enviado é individualmente confirmado. Dessa forma o emissor sabe se o frame foi recebido ou não e poderá enviá-lo novamente.
- ❑ Origem usa um mecanismo de temporização para reenviar quadros não confirmados.
- ❑ Útil para canais não confiáveis: Wireless

Serviço orientado a conexão confirmado

- ❑ Serviço mais sofisticado.
- ❑ Emissor e receptor estabelecem conexão antes do envio dos dados.
- ❑ Cada *frame* enviado é numerado. A camada garante que cada *frame* enviado é recebido.
- ❑ Cada *frame* é recebido exatamente uma vez e todos os *frames* chegam em ordem.
- ❑ O serviço oferecido para a camada de rede é uma sequência de bits corretos.

Comentários sobre os diferentes tipos de serviços

- ❑ Confirmação na camada de enlace é uma otimização e não um requisito
 - Pode ser deixada para a camada de transporte (camada fim-a-fim)
 - Se existe na camada de enlace ou não depende do meio de comunicação
- ❑ O serviço a ser oferecido para a camada de rede depende, dentre outros fatores, da aplicação que utilizará esse serviço.

Camada de Enlace

Delimitação de quadros

Delimitação de Quadros

❑ Problema a ser resolvido:

- Como agrupar sequências de bits em quadros para que possam ser processados como unidades de informação?

ou, de outra forma,

○ Como fazer delimitação de quadros?

- Inserir intervalos de tempo entre transmissões de quadro
 - Em comunicação assíncrona não é viável

Delimitação de quadros (Framing)

- ❑ A camada física transmite uma sequência de bits (*bit stream*)
- ❑ A sequência pode ser grande e os bits podem conter erros.
- ❑ A camada de enlace deve detectar e, se necessário, corrigir os erros de transmissão.
- ❑ Inicialmente a sequência de bits é quebrada em frames.

Delimitação de quadros

- Os principais métodos para delimitação de quadros são:
 - Contagem de caracteres;
 - Caracteres de início e de fim, com caractere de preenchimento;
 - Flags de início e fim com caractere de preenchimento e;
 - Violação de código da camada física.

Contagem de Caracteres

- ❑ Usa um campo no cabeçalho para indicar o número de caracteres no quadro;
- ❑ Problema: o caractere de contagem pode sofrer erro de transmissão, impossibilitando o reconhecimento o início do próximo quadro.
- ❑ Não é usado na prática para protocolos da camada de enlace.

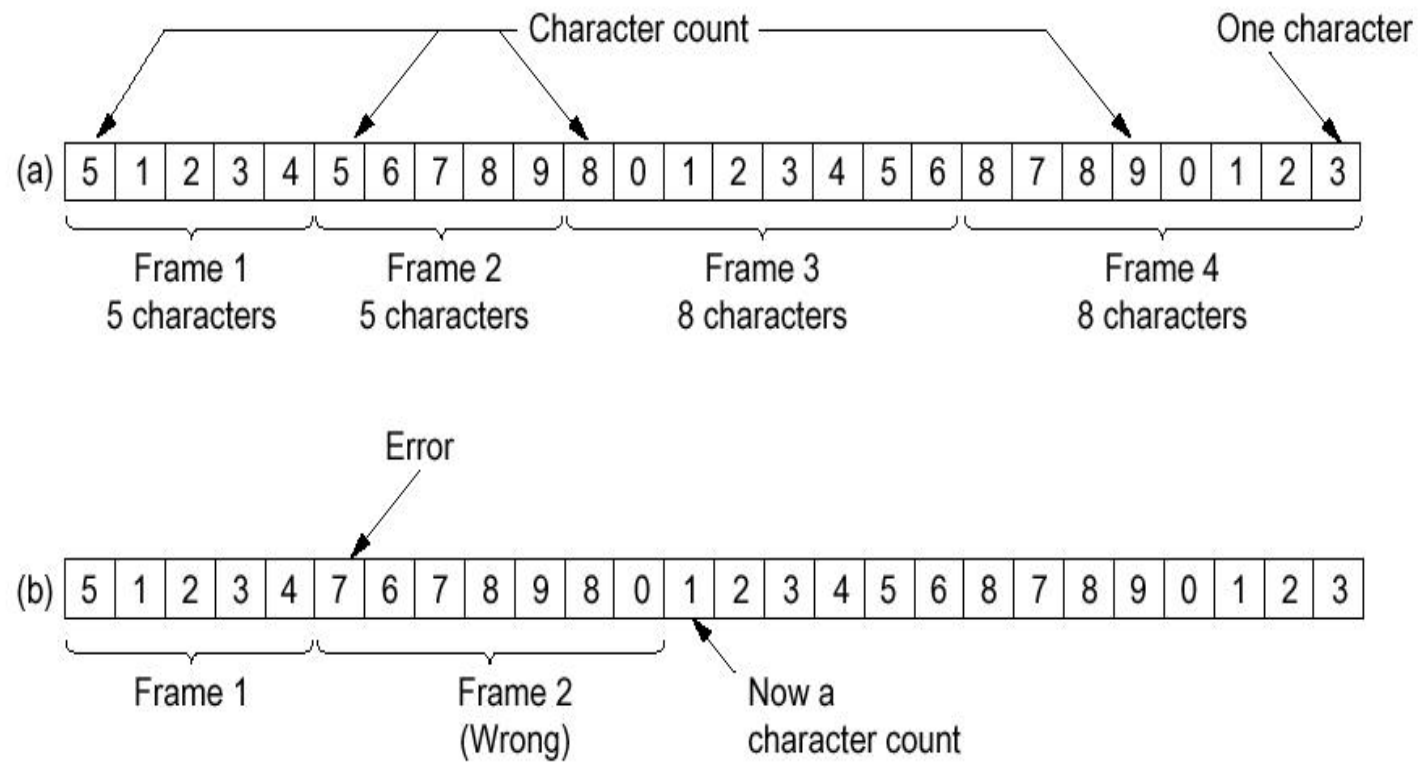


Fig. 3-3. A character stream. (a) Without errors. (b) With one error.

Caracteres de início e de fim...

- ❑ Reconhecimento do início e do fim de um quadro através dos caracteres ASCII:
 - Início: **DLE STX** (Data Link Scape, Start TeXt)
 - de fim: **DLE ETX** (Data Link Scape, End of TeXt)
 - Método usado em protocolos orientados a caracteres.

Caracteres de início e de fim...

- ❑ E se os caracteres para DLE STX e DLE ETX ocorrem nos dados?
- ❑ Inserir um caracter DLE adicional antes de cada DLE nos dados (*Stuffing*).
- ❑ Técnica presa ao código ASCII e a caracteres de 8 bits. (Caracteres de tamanho arbitrário?)

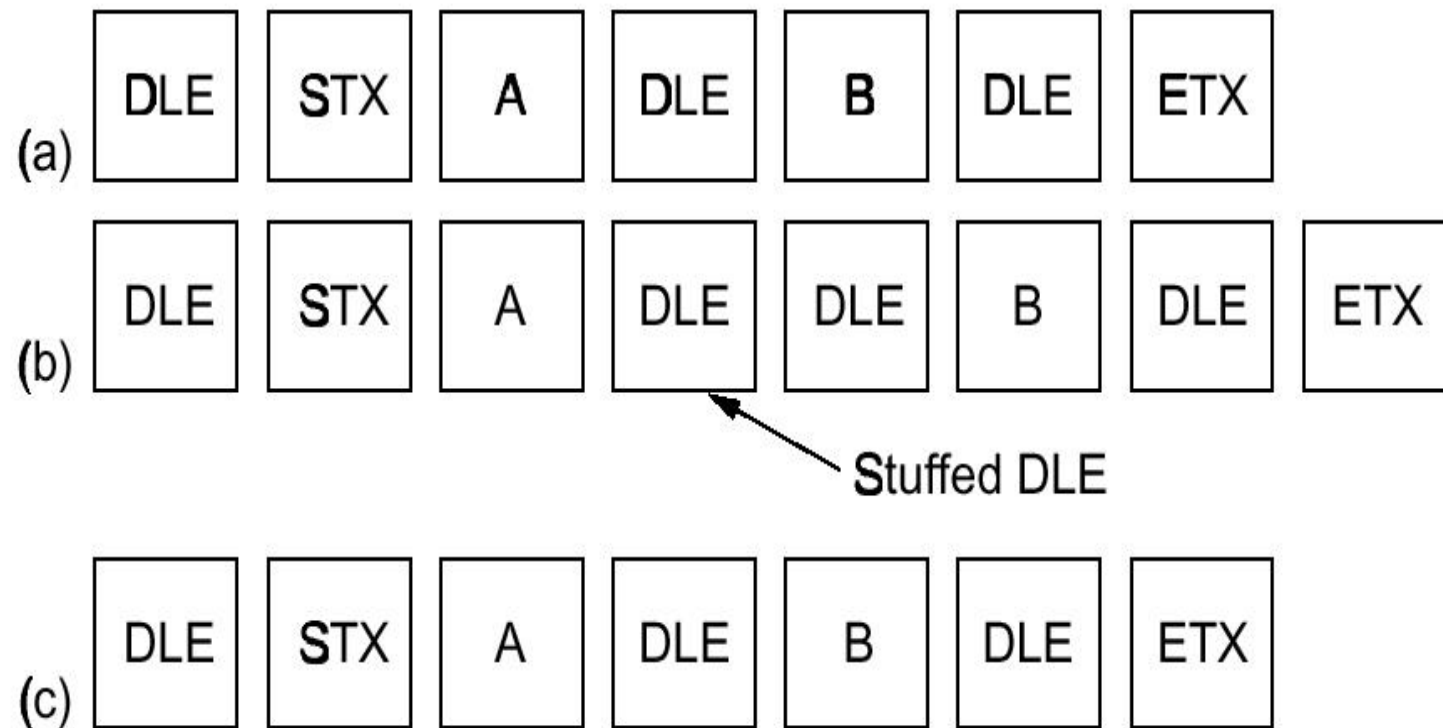


Fig. 3-4. (a) Data sent by the network layer. (b) Data after being character stuffed by the data link layer. (c) Data passed to the network layer on the receiving side.

Flags de início e fim...

- ❑ Permite codificar caracteres com um número arbitrário de bits por caractere
- ❑ Usa padrão especial de bits (*flag*): 01111110
- ❑ Sempre que 5 "ums" (1s) consecutivos são encontrados nos dados, o emissor insere um zero (*stuffing*).

Flags de início e fim...

- ❑ Quando o receptor encontra cinco "ums" seguidos por um zero, o *stuff* é retirado.
- ❑ O Padrão só ocorre nos limites dos quadros, nunca nos dados.
- ❑ Processo conhecido como *bit stuffing*

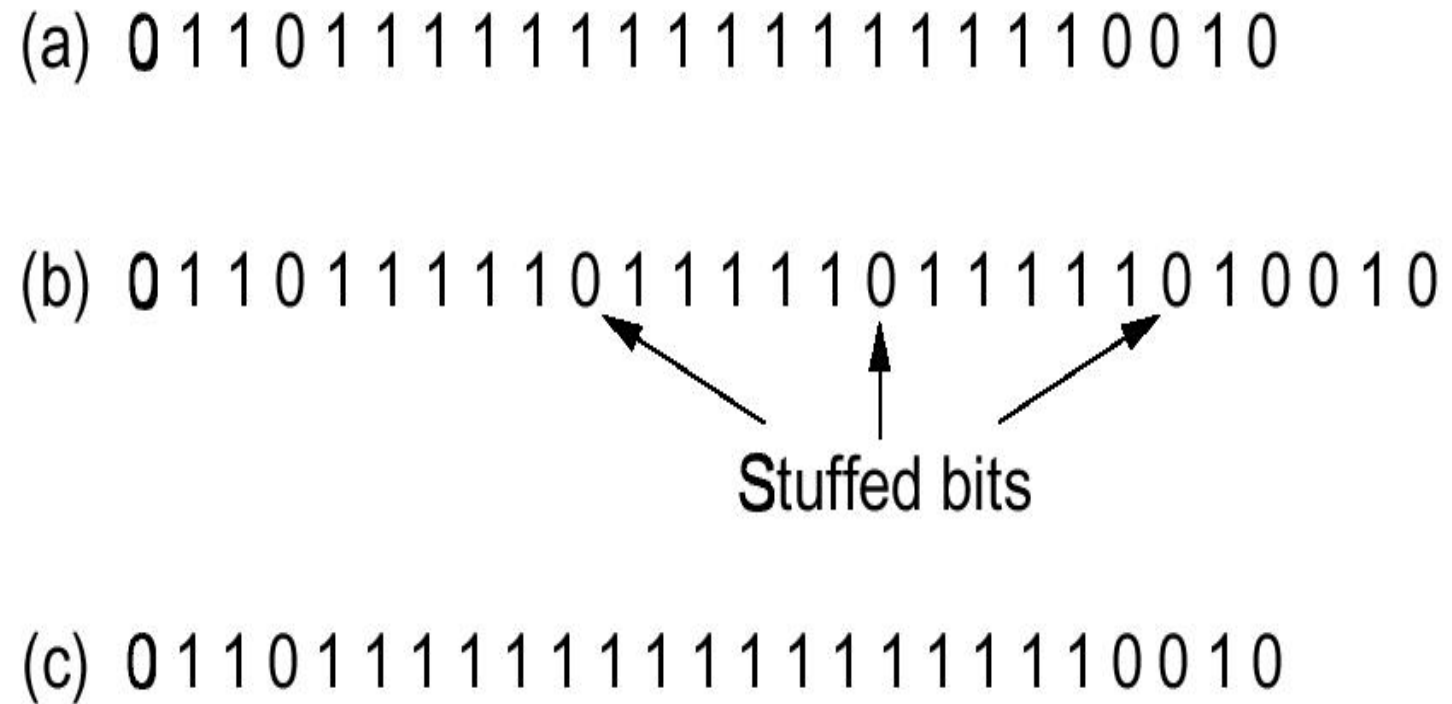


Fig. 3-5. Bit stuffing. (a) The original data. (b) The data as they appear on the line. (c) The data as they are stored in the receiver's memory after destuffing.

Violação de Código

- ❑ Método baseado em uma característica da camada inferior
- ❑ Existem códigos de transmissão que possuem uma transição no meio do período de transmissão de um bit.
- ❑ O início e fim de quadro são determinados por um código de transmissão inválido
 - Usado no padrão IEEE 802.
- ❑ Método utilizado em redes com redundância na codificação física.
- ❑ Algumas LANs codificam um bit de dados usando dois bits físicos:
 - Um bit 1 é codificado como um par alto-baixo.
 - Um bit 0 é codificado como um par baixo-alto.
 - Os pares alto-alto e baixo-baixo não são utilizados

Camada de Enlace

Controle de Erros

Controle de Erros

- ❑ Como ter certeza que todos os quadros transmitidos foram entregues à camada de rede do destino, e na ordem correta?
- ❑ Primeiro passo: detecção
- ❑ Todos os métodos: inserção de bits extras
- ❑ Esses bits podem ser obtidos a partir da informação original
- ❑ O receptor recalcula os bits extras

Controle de erro

❑ Objetivo (serviço):

- Entregar em ordem e sem repetição os dados recebidos da camada física para a camada de rede

❑ Mecanismos utilizados para oferecer esse tipo de serviço (regras de procedimento):

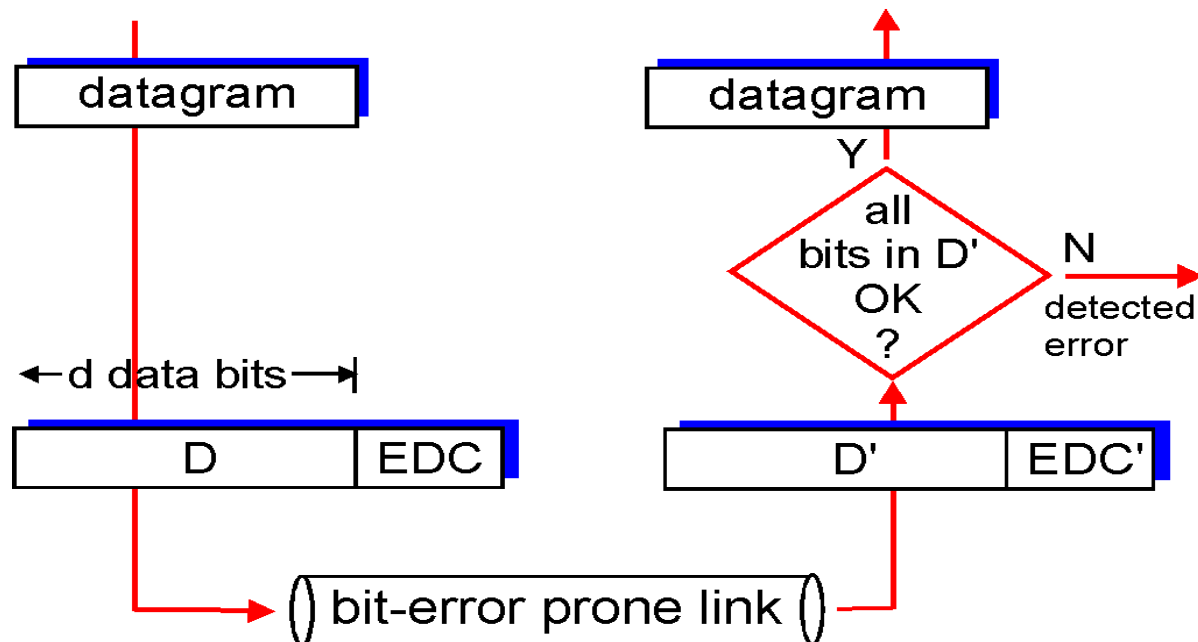
- Confirmação positiva e negativa de quadros pelo destinatário
- Temporização de quadros enviados pela origem
- Número de sequências de quadros
- Retransmissão de quadros em um número finito de vezes.

Detecção de Erros

EDC= Bits de Detecção e Correção de Erros (redundância)

D = Dados protegidos pela verificação de erros, pode incluir os campos de cabeçalho

- A detecção de erros não é 100% confiável!
 - protocolos podem deixar passar alguns erros, mas é raro
 - Quanto maior o campo EDC melhor é a capacidade de detecção e correção de erros



Controle de erros

□ 2 Abordagens:

- Incluir redundância suficiente para que o receptor deduza que algum erro aconteceu.

Detecção de Erro

- Incluir redundância suficiente para que o receptor deduza onde o erro aconteceu.

Correção de Erros

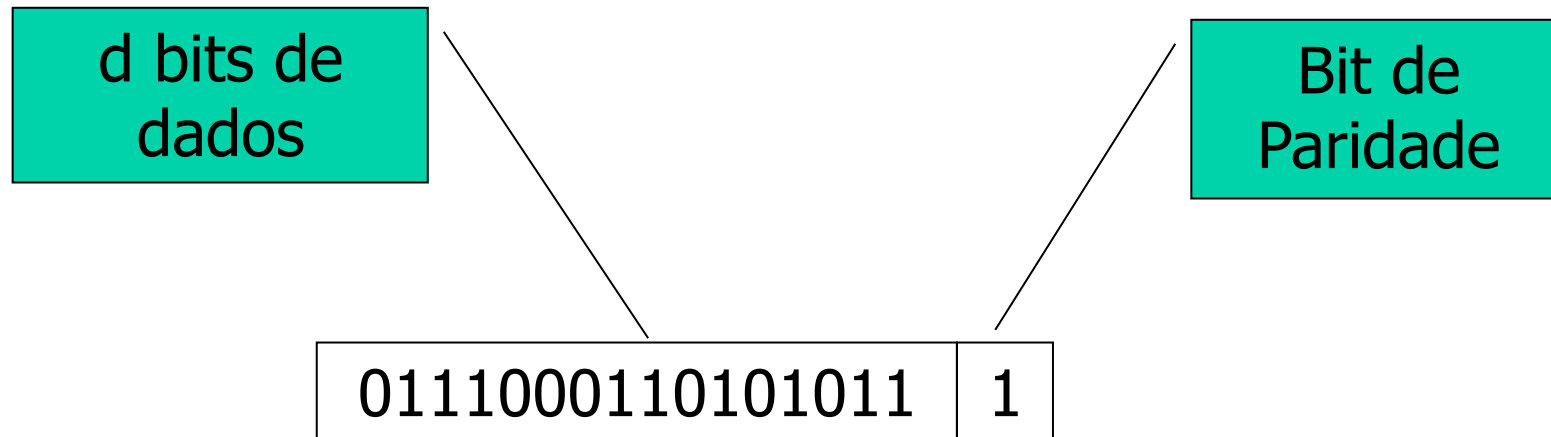
Controle de Erros - Técnicas

- ❑ Bit de paridade
- ❑ Distância de Hamming
- ❑ Redundância Cíclica (CRC)

Bit de paridade

- ❑ Inserção de 1 bit extra ao final de cada caractere de modo a deixar todos os caracteres com um número *par* ou *impar* de bits 1
- ❑ Caso um número par de bits tenha sido invertido não é possível detectar o erro

Bit de paridade



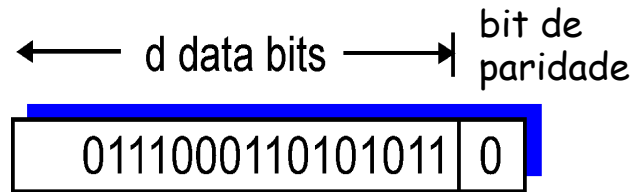
Paridade Bidimensional

- ❑ Os d bits de dados são divididos em i linhas e j colunas.
- ❑ Um valor de paridade é calculado para cada linha e cada coluna.

Verificação de Paridade

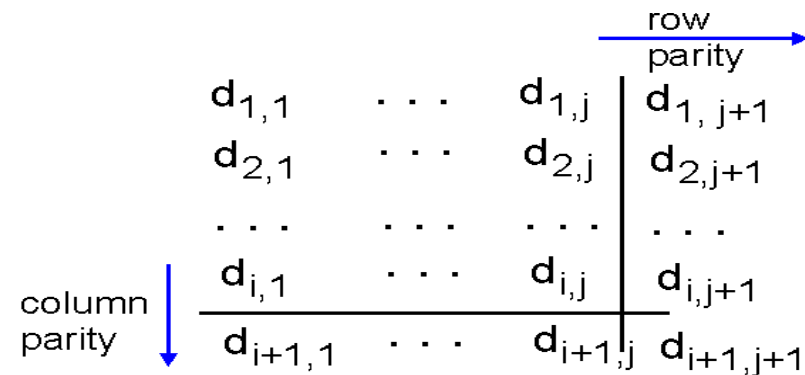
Paridade com Bit único:

Detecta erro de um único bit



Paridade Bi-dimensional:

Detecta e corrige erros de um único bit



1	0	1	0	1	1
1	1	1	1	0	0
0	1	1	1	0	1
0	0	1	0	1	0

sem erros

1	0	1	0	1	1
1	0	1	1	0	0
0	1	1	1	0	1
0	0	1	0	1	0

erro de
paridade

erro de 1 bit
corrigível

erro de
paridade

Distância de Hamming

- ❑ **Código de Hamming** é um código para a detecção de erros em transmissões de dados binários . É geralmente usado para comunicações em que o grau de correção de erros seja importante de ser efetuado.
- ❑ Basicamente o código de hamming emprega o ***bit-stuffing*** para assegurar que eventuais erros sejam detectados. Esses bits inseridos dentro do bloco de dados, vão detectar com alguma eficiência,eventuais erros no envio de dados, bem como em alguns casos, corrigir mediante a posição do bit com erro, determinado pelo receptor.

Distância de Hamming

- ❑ O código de Hamming tem infinitas variantes, no entanto existe universalidade na representação da designação. Usualmente segue a seguinte regra :
 - *Hamming* (n° de bits do bloco de dados COM n° de bits transmitidos , n° de bits de paridade).
- ❑ Exemplo: código Hamming (7,4) . Neste caso existem 7 bits no bloco total. Pretendeu-se transmitir 4 bits e são usados 3 bits para a detecção de erros nesse bloco de 4 bits de dados a enviar. ($7-4=3$)

Distância de Hamming

1. Os bits de controle de dados (também designados por "bits de Hamming") estão em posições 2^k (ou seja: 1, 2, 4, 8, 16, 32, 64, etc..)
2. As restantes posições são para o bloco de dados a enviar (Ou seja: 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 17, etc.)
3. Cálculo XOR
 - 3.1 Representa-se em binário as posições em que o bloco de dados tem o valor binário de 1 . Essas posições são referenciadas da esquerda para a direita.

Distância de Hamming

- ❑ Bloco a enviar - 1110 ; Bloco Total codificado em Hamming(7,4)- ? ? 1 ? 1 1 0 Neste caso calcular-se-ia :

3 : 011

5 : 101

6 : 110

- ❑ Calcula-se o XOR seqüencial dos binários calculados e determina-se os bits de Hamming nas posições referenciadas e em ordem seqüencial da direita para a esquerda.

No exemplo anterior:

3 : 0 1 1

5 : 1 0 1

_____XOR

1 1 0

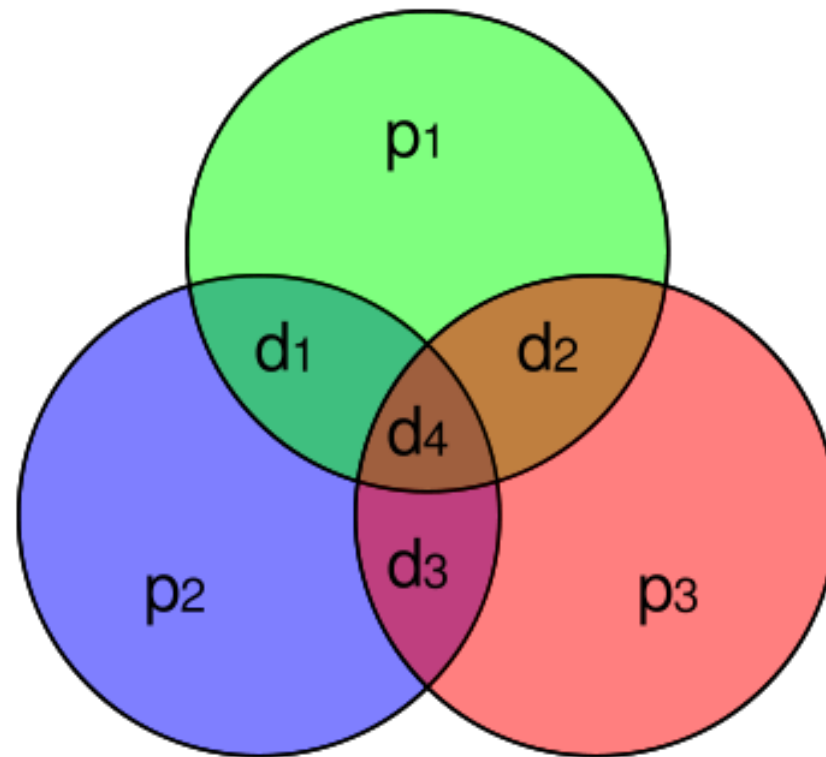
6 : 1 1 0

_____XOR

0 0 0

..3^a2^a1^a -posições Ficaria então, 0 0 1 0 1 1 0.

Distância de Hamming



Checksum da Internet

Objetivo: detectar “erros” (ex. bits trocados) num segmento transmitido (nota: usado *apenas* na camada de transporte)

Sender:

- ❑ trata o conteúdo de segmentos como sequências de números inteiros de 16 bits
- ❑ checksum: adição (soma em complemento de um) do conteúdo do segmento
- ❑ transmissor coloca o valor do checksum no campo checksum do UDP

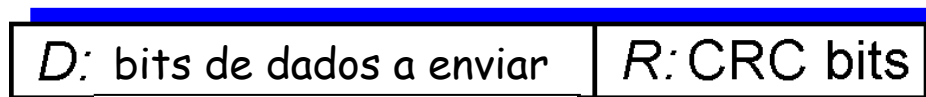
Receptor:

- ❑ computa o checksum do segmento recebido
- ❑ verifica se o checksum calculado é igual ao valor do campo checksum:
 - NÃO - erro detectado
 - SIM - não detectou erro.
Mas talvez haja erros apesar disso? Mais depois....

Verificação de Redundância Cíclica

- ❑ encara os bits de dados, **D**, como um número binário
- ❑ escolhe um padrão gerador de $r+1$ bits, **G**
- ❑ objetivo: escolhe r CRC bits, **R**, tal que
 - $\langle D, R \rangle$ é divisível de forma exata por G (módulo 2)
 - receptor conhece G , divide $\langle D, R \rangle$ por G . Se o resto é diferente de zero: erro detectado!
 - pode detectar todos os erros em sequência (burst errors) com comprimento menor que $r+1$ bits
- ❑ largamente usado na prática (ATM, HDCL)

← d bits → ← r bits →



padrão de bits

$$D * 2^r \text{ XOR } R$$

fórmula matemática

Exemplo de CRC

Desejado:

$$D \cdot 2^r \text{ XOR } R = nG$$

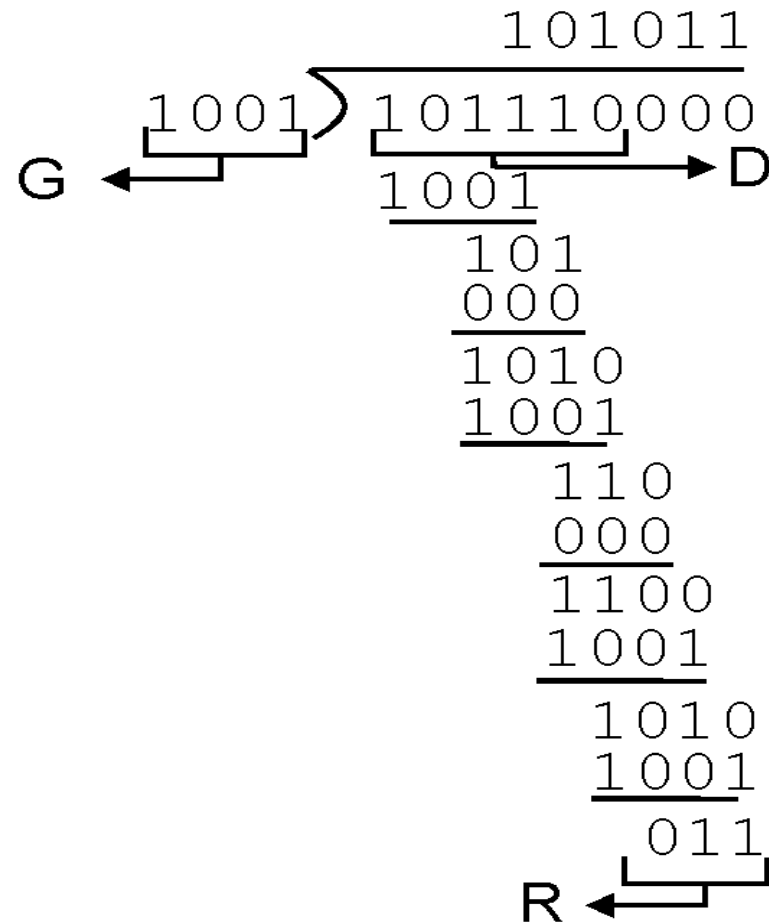
equivalente a:

$$D \cdot 2^r = nG \text{ XOR } R$$

equivalente a:

se nós dividimos D por G , buscamos resto R

$$R = \text{resto} \left[\frac{D \cdot 2r}{G} \right]$$



CRC

- ❑ *Cyclic Redundancy Check*
- ❑ Esquema mais eficiente
- ❑ Quadro de k bits é representado por um polinômio em X de ordem $k-1$
 - quadro = 10110001
 - polinômio = $X^7 + X^5 + X^4 + 1$

$$B(x) = x^5 + x^3 + x^2 + x^0 = (101101)_2$$

- ❑ Uma mensagem deve ser enviada com o código de CRC calculado para que possa ser verificada no receptor.

CRC

- ❑ Emissor/receptor concordam com um **polinômio gerador $G(x)$** ;
- ❑ Idéia: polinômio representado pelo quadro com a soma de verificação seja divisível por $G(x)$.
- ❑ Receptor tenta dividir por $G(x)$. Se houver resto, houve um erro de transmissão

CRC

- ❑ O cálculo de CRC é realizado através de uma operação de divisão aplicada a números binários.
- ❑ Há uma diferença na divisão: as operações de subtração são substituídas por operações lógicas de ou exclusivo (XOR - eXclusive OR).

CRC

- ❑ Relembrando a tabela da operação XOR:

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

CRC

- ❑ Um emissor deseja enviar D com d bits de dados.
- ❑ Um código de CRC R com r bits de comprimento deve ser gerado e anexado aos dados antes do envio.
- ❑ O receptor e o emissor conhecem um padrão de bits denominado G , de gerador.
- ❑ Este gerador tem $(r + 1)$ bits de comprimento.
- ❑ O bit mais significativo do gerador (mais à esquerda) deve ser 1.

D: d bits de dados	R: r bits de CRC
----------------------	--------------------

CRC

- A base para o cálculo de R (código de CRC) é a fórmula:

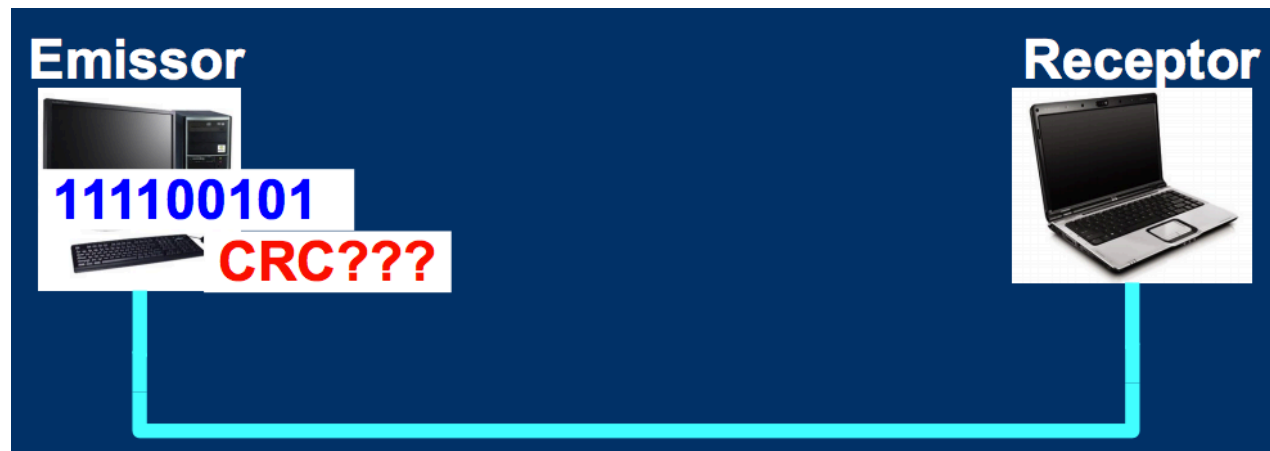
$$R = \text{resto} \frac{D * 2^r}{G}$$

- $D * 2^r$ é o deslocamento dos bits de dados à esquerda r casas.
 - Isso é a adição de r bits 0 no final dos bits de dados. Por exemplo:

$$D = (100101)_2 \text{ e } r = 3 \implies D * 2^r = (100101\mathbf{000})_2$$

CRC - Antes do Envio dos Dados

- ❑ O emissor deseja enviar os bits de dados 111100101.
- ❑ O gerador são os bits 101101.
- ❑ Como calcular o CRC antes de enviar os dados???

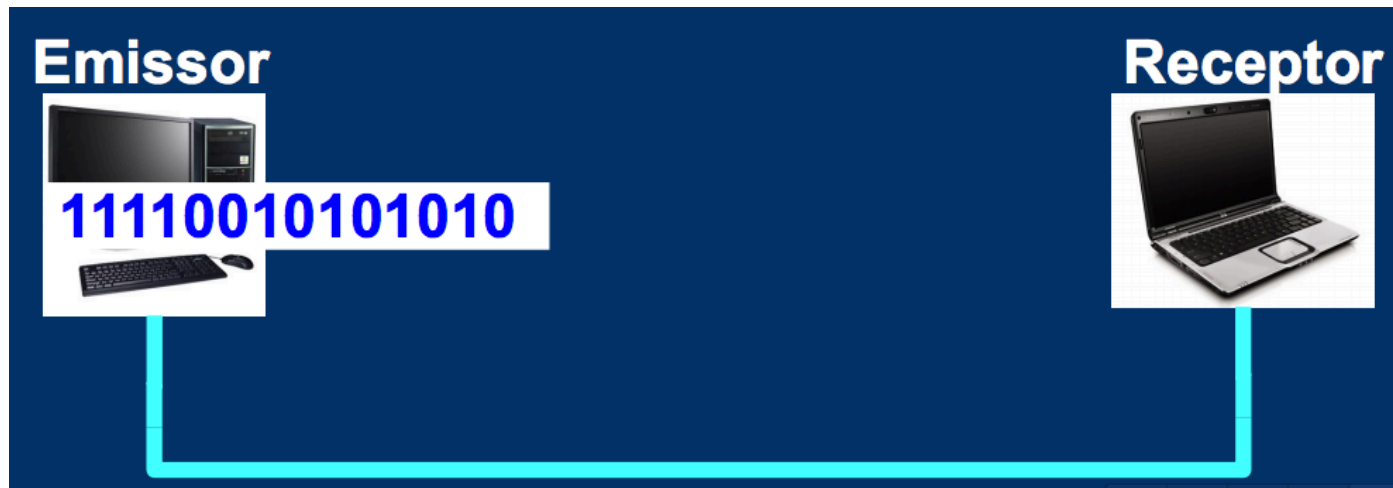


CRC - Antes do Envio dos Dados

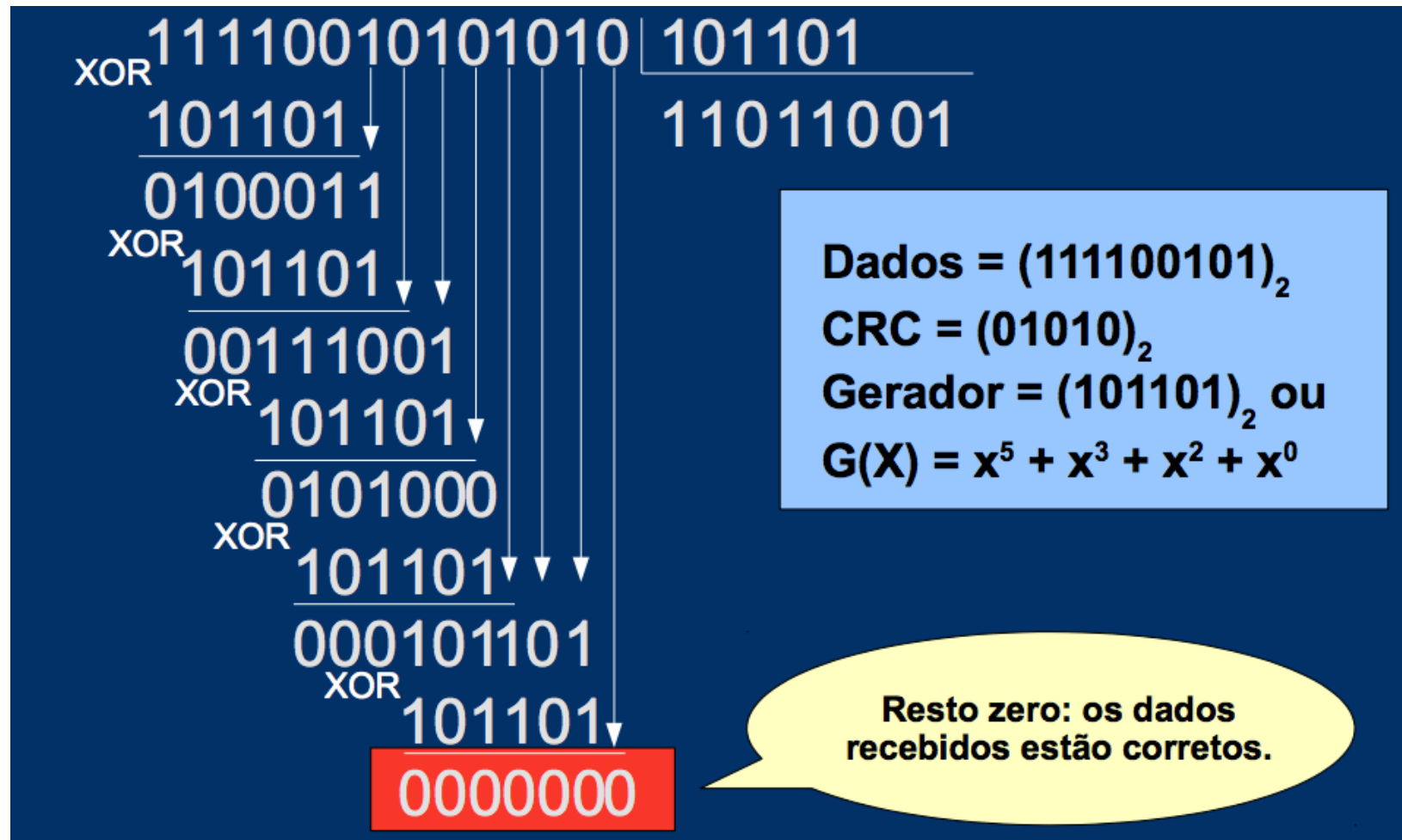


CRC - Envio dos Dados

- ❑ Após calcular o CRC cujos bits são 01010, o emissor envia os bits de dados mais os bits de CRC.
- ❑ O receptor utiliza os bits enviados e divide pelo gerador para verifica se os bits estão corretos.



CRC - Teste no Receptor



CRC - Padrões

CRC-32

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + x^0 = (100000100110000010001110110110111)_2$$

CRC-16

$$G(x) = x^{16} + x^{15} + x^2 + x^0 = (110000000000000101)_2$$

CRC-12

$$G(x) = x^{12} + x^3 + x^1 + x^0 = (1000000001011)_2$$

CRC-8

$$G(x) = x^8 + x^2 + x^1 + 1 = (100000111)_2$$

CRC - Padrões

- ❑ Cada padrão de CRC detecta erros em rajadas menores que $(r + 1)$ bits.
- ❑ Mesmo um erro em rajadas iguais ou superiores a $(r + 1)$ bits é detectada com probabilidade $(1 - 0,5^r)$.
- ❑ Os padrões também detectam qualquer quantidade ímpar de erros de bits.
- ❑ ATM utiliza CRC-8 no cabeçalho das células.
- ❑ CRC-32 é utilizado pelos protocolos da camada de enlace padronizados pelo IEEE.