EE15
Comunicação de Dados

Aula 20-21:
TRATAMENTO DE ERROS

TRATAMENTO DE ERROS

Por que adotar algoritmos de tratamento de erros?

- Por melhor projetado/contruído que seja um sistema de comunicação, ele irá falhar.
- O que significa uma "falha" em um sistema de comunicação?
  Significa que:

  INFO RECEBIDA ≠ INFO ENVIADA.

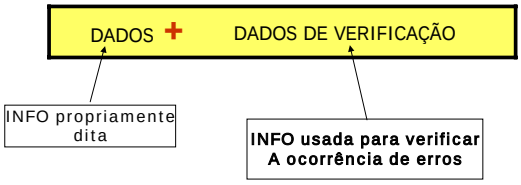
- Pode também significar que a INFO simplesmente não foi recebida.

CAUSAS DOS ERROS

- O meio físico de transmissão.
  - Ruído;
  - Limitação em banda;
  - Atenuação;
  - etc.
- Combinação dos fatores acima:
  Distorção do sinal portador da INFO = corrupção da INFO= ERRO.

Princípio Básico dos Algoritmos de Tratamento

- Envio de INFO REDUNDANTE que permita ao receptor DETECTAR ou CORREGIR os erros ocorridos.
- Formato geral dos dados transmitidos:



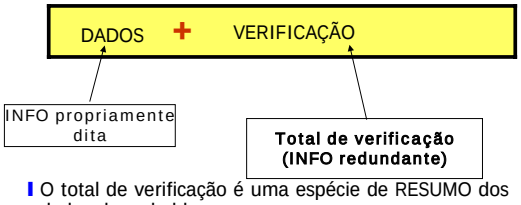
Duas estratégias principais de Tratamento

- **Correção de erros**
  Os dados de verificação enviados permitem a CORREÇÃO dos erros.
  Os algoritmos são complexos e exigem o envio de grandes quantidades de INFO REDUNDANTE.
  Um algortimo de correção bastante conhecido é o CÓDIGO HAMMING.
  Os erros são detectados e corrigidos pelo receptor.

- **Detecção de erros**
  Os dados de verificação enviados permitem que o receptor detecte a ocorrência de erros.
  Em havendo erros, o receptor requisita ao transmissor o reenvio do bloco de dados que foi corrompido

Algoritmos de Detecção de Erros

- Formato geral dos blocos de dados:



- O total de verificação é uma espécie de RESUMO dos dados de cada bloco.

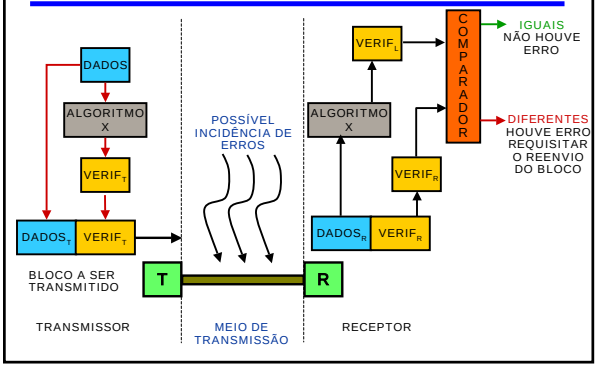
- A partir dos dados a serem transmitidos, o transmissor calcula o total de verificação (o resumo) e o anexa aos dados, formando o bloco que será transmitido.
- Ao chegar ao receptor, este (o receptor) recalcula o total de verificação a partir dos DADOS RECEBIDOS, gerando o Total de Verificação Local (TVL).
- O receptor compara, então, o TVL com o Total de Verficação Recebido (TVR), que veio junto com os dados (o TVR é calculado pelo transmissor, lá no outro lado do meio de transmissão).

- Se houver diferença entre TVL e TVR, só há uma possibilidade:

  DADOS ENVIADOS ≠ DADOS RECEBIDOS

- Isto é porque o algoritmo usado no receptor e no transmissor é o MESMO.
- Para que os Totais de Verificação sejam diferentes, supondo que se use o mesmo algoritmo, a única possibilidade é que o ponto de partida foi diferente, ou seja:

  DADOS ENVIADOS ≠ DADOS RECEBIDOS



Algoritmos Detectores mais comuns

- VRC = Vertical Redundancy Checking
- LRC = Longitudinal Redundancy Checking
- CRC = Cyclic Redundancy Checking

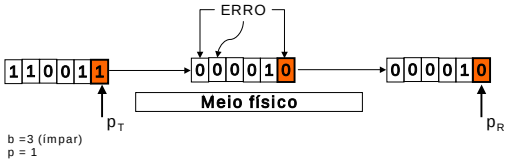
- Os dois primeiros algoritmos baseiam-se na verificação da PARIDADE dos dados; e o último baseia-se na divisão de polinômios.

Algoritmo VRC

- Dada uma seqüência de **n** bits que se queira transmitir, anexa-se um bit (pode ser 0 ou 1), tal que os **n+1** bits apresentem o número de bits 1 com a paridade pré-definida (par ou ímpar).
- Para isso, o receptor e transmissor devem concordar quanto à paridade.

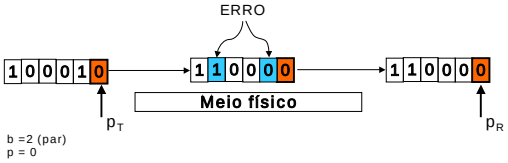
- Exemplo 1: Geração do bit de paridade. Paridade PAR

- Exemplo 2: Detecção de erros. Paridade PAR



O VRC é um algoritmo orientado a bit, sendo indicado para protocolos de baixo nível, em que se tem controle bit a bit: implementação em hardware, etc.

- Exemplo 3: Uma falha do algoritmo. Paridade PAR.

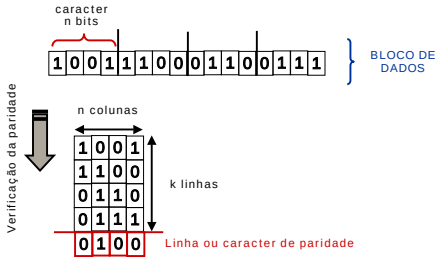


Neste caso houve erro, mas o algoritmo NÃO foi capaz de detectar.

- REGRA GERAL: O algoritmo VRC não é capaz de detectar a troca de números pares de bits (número erros=número PAR), independente da paridade usada.

Algoritmo LRC

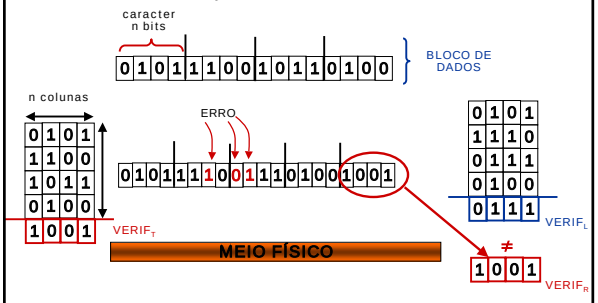
- Semelhante ao VRC.
- Os dados são organizados sob a forma de uma matriz [**k**x**n**]: a paridade é verificada ao longo de cada uma das **n** colunas; é gerado um caracter de paridade, que é a linha **k**+**1** da matriz:



- Este método consegue detectar RAJADAS de erros de até **n** bits. Por rajadas entende-se até **n** bits consecutivos errados, podendo todos estarem errados ou alguns ($\leq$ n).
- Rajadas com n+1 bits ou mais não são, em princípio, detectadas: mesmo problema do VRC.
- Havendo discrepâncias na paridade, requisitá-se o reenvio do bloco de dados (todos os **k** caracteres e o caracter de paridade: **k+1** caracteres ao total).

- É um algoritmo orientado a byte: pode ser usado em protocolos de mais alto nível, mas nos quais é viável/possível **desmontar** os caracteres anexando o bit de paridade a cada caracter.
- Dados **k** caracteres (**k** linhas da matriz), gera-se **1** caracter adicional de paridade.

- Exemplo 1: Detecção de erros. Paridade IMPAR.



- COMO $VERIF_L \neq VERIF_R$: HOUVE ERRO!
- REQUISITAR A RETRANSMISSÃO DO BLOCO!

- Exemplo 1: falha do algoritmo LRC. Paridade IMPAR, com o mesmo bloco de dados.

$VERIF_L = VERIF_R$, por tanto o algoritmo LRC calcula erroneamente que NÃO HOUVE ERRO!, mas houve!