Simulação de Erros com Verificação CRC

Dados Originais

byte[] entrada = { (byte) 0b01111100, (byte) 0b01001100 };
Estes dados são enviados com um byte adicional que representa o CRC calculado:
01111100 01001100 [CRC]

Simulações de Erro

O programa aplica diferentes tipos de erros e verifica se o CRC consegue detectar a corrupção. A função CRC.verificarCRC() é utilizada para validar os dados.

Original

- Nenhuma alteração feita.
- Bits: 01111100 01001100 [CRC]
- Resultado CRC: Válido
- Conclusão: Os dados estão íntegros.

Erro de 1 bit

- Inversão de 1 bit (mais significativo do primeiro byte).
- 01111100 -> 11111100
- Resultado CRC: Inválido
- Conclusão: CRC detecta até mesmo erros mínimos.

Erro de múltiplos bits

- Inversão de 3 bits não consecutivos.
- Resultado CRC: Inválido
- Conclusão: CRC continua eficaz mesmo com múltiplas alterações.

Substituição de byte completo

- Substituição do segundo byte por 0xFF.
- 01001100 -> 11111111
- Resultado CRC: Inválido
- Conclusão: Erro grosseiro facilmente detectado.

Simulação de Erros com Verificação CRC

Erro burst (rajada de 4 bits)

- Inversão de 4 bits consecutivos.
- Resultado CRC: Inválido
- Conclusão: CRC protege contra erros sequenciais comuns em canais ruidosos.

Conclusão Geral

Tipo de Erro	Detectado pelo CRC?	Observação Técnica
Nenhum (original)	Sim	Dados íntegros
1 bit	Sim	Detecção precisa
Múltiplos bits	Sim	Alta sensibilidade
Substituição de byte	Sim	Erro grosseiro
Erro em rajada (burst)	Sim	Proteção robusta

O CRC é altamente eficaz na detecção de erros comuns de transmissão, mesmo que mínimos ou localizados.