

Detecção e Controle de Erro

Prof. Dr. S. Motoyama

Problema 3: Detecção de erro (ED)

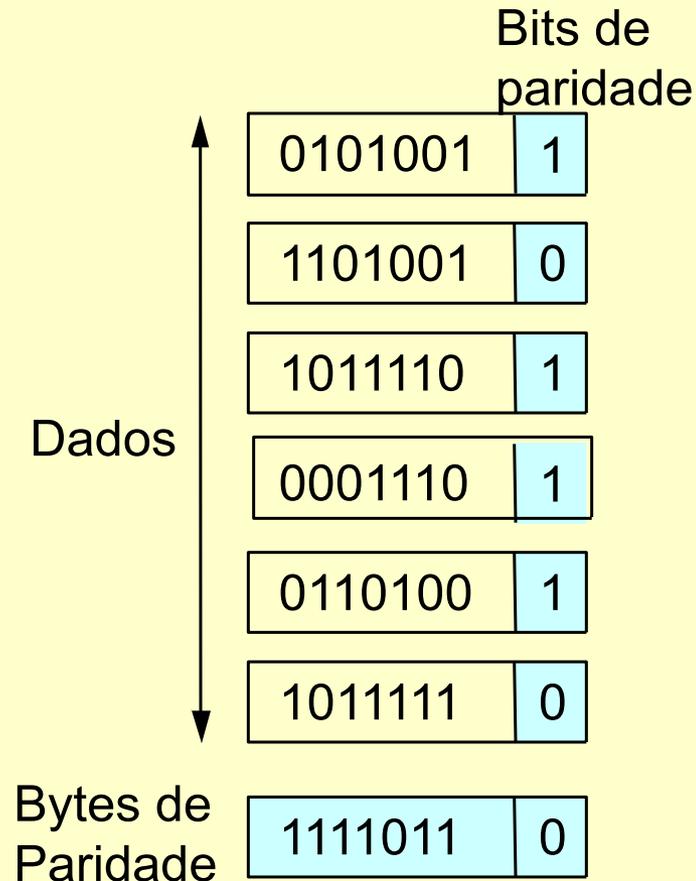
- Em meios físicos de transmissão, erros de transmissão podem ocorrer. As probabilidades de erro são diferentes para cada meio.
- Duas abordagens gerais para tratamento de erros:
 - Código corretor de erro
 - Código detector de erro + um mecanismo de correção de erro quando os erros são detectados.
 - Inserção de redundâncias para detecção ou correção de erro.
- Métodos de detecção de erro:
 - Teste de redundância cíclica (Cyclic redundancy check, CRC)
 - Soma de verificação (Checksum)

ED: Detecção de erro

- Códigos detectores de erro são normalmente inseridos em mais do que uma camada, por ex.,
 - HTTP
 - TCP (16-bits checksum para o cabeçalho de TCP e dados)
 - IPv4 (16-bits checksum para o cabeçalho de IP)
 - PPP/Ethernet (CRC-16, CRC-32 para o todo o quadro)

ED: paridade bi-dimensional

- A paridade 2-dim detecta todos os erros de 1, 2 e 3 bits, e a maioria dos erros de 4 bits.



ED: Checksum

- Idéia básica:
 - Some todas as palavras que são transmitidas e depois transmite o resultado daquela soma. Se quaisquer dos dados transmitidos, incluindo o checksum, sofrerem algum erro, então, no receptor, o resultado da operação soma não será correto.
- Checksum na Internet:
 - Um transmissor soma palavras de 16 bits usando a aritmética complemento de 1, e guarda o resultado no campo checksum.
 - Um receptor realiza a operação de checksum em todas as palavras de 16 bits. Se não há erro, o resultado deve ser tudo 1.

ED: Checksum

- A aritmética complemento de 1:
 - Um inteiro negativo $-x$ é representado como o complemento de x . Cada bit de x é invertido.
 - O “vai 1” do bit mais significativo deve ser somado ao resultado
- Por exemplo: checksum de palavras de 4 bits
 - Dados: 1010 1100, e o seu checksum: 1000.
(0101+0011=1000)
 - Dados enviados: 1010 1100 1000
 - O receptor soma todas as palavras de 4 bits em complemento de 1, inclusive o checksum, resultando em 1111.
(0101+0011=1000, 1000+0111=1111)

ED: Checksum

- Vantagens:
 - Uso relativamente pequeno de número de bits redundantes.
 - Fácil de implementar em software.
- Desvantagens:
 - Não é um algoritmo detector de erro tão robusto como CRC.

ED: código de redundância cíclica (CRC)

- Suponha que uma mensagem de $(n+1)$ bits, $M(x)$, possa ser escrita como um polinômio de grau n . Por ex.,
 - 10011010 como
 - $1 \times x^7 + 0 \times x^6 + 0 \times x^5 + 1 \times x^4 + 1 \times x^3 + 0 \times x^2 + 1 \times x^1 + 0 \times x^0 =$
 - $= x^7 + x^4 + x^3 + x^1$
- Dado um polinômio gerador de ordem k , $C(x)$, encontre uma palavra de código de k bits, tal que $M(x)$ concatenado com a palavra de código é divisível por $C(x)$.
 - $P(x)$ é uma concatenação de $M(x)$ e a palavra de código.

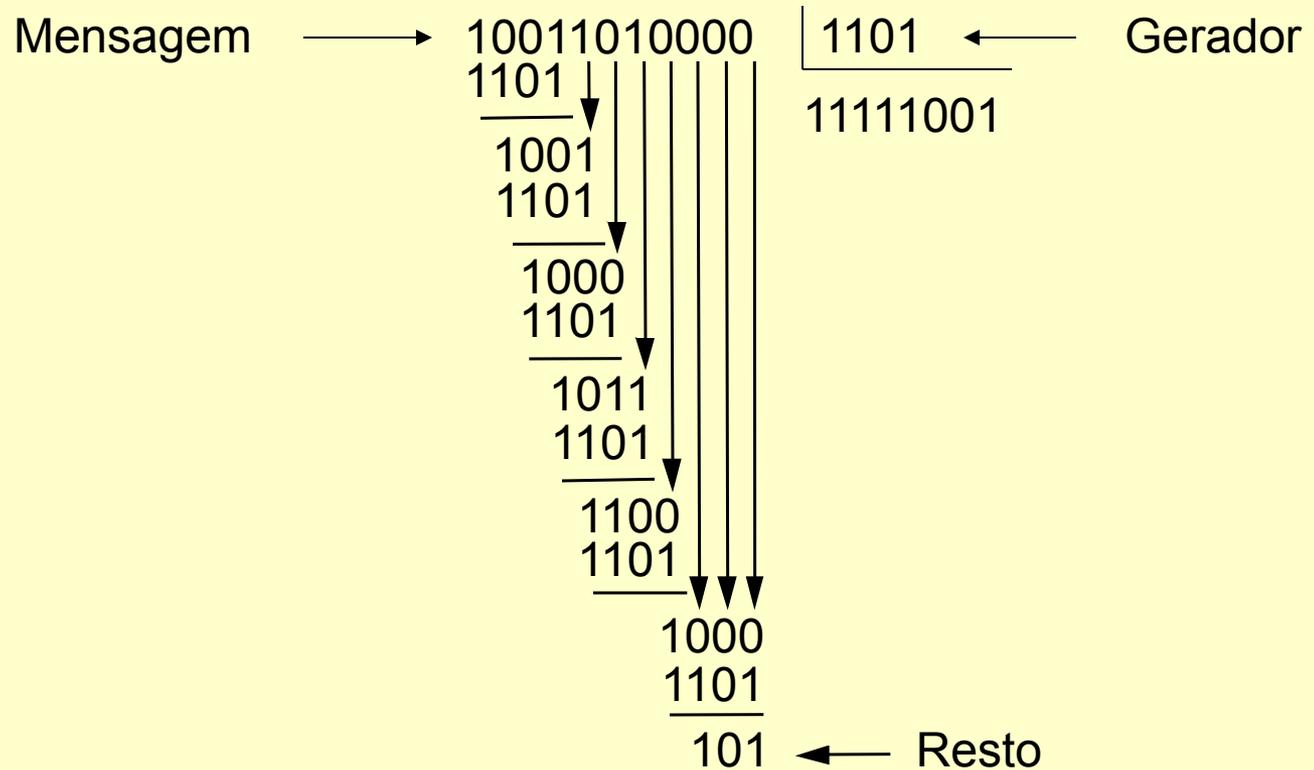
ED: código de redundância cíclica

- Por exemplo, $C(x) = x^3 + x^2 + 1$
 - Seja 101 a palavra de código procurada.
 - $P(x)$ é dado por 10011010 concatenado com 101, ou $1 \times x^{10} + 0 \times x^9 + 0 \times x^8 + 1 \times x^7 + 1 \times x^6 + 0 \times x^5 + 1 \times x^4 + 0 \times x^3 + 1 \times x^2 + 0 \times x^1 + 1 \times x^0 = x^{10} + x^7 + x^6 + x^4 + x^2 + 1$.
 - Pode-se verificar que $P(x)$ é divisível por $C(x)$, isto é, resto zero.
 - Erros podem ser detectados quando o receptor divide $P(x)$ por $C(x)$ e o resto não é zero.

ED: código de redundância cíclica

- Para obter a palavra de código
 - Multiplique $M(x)$ por x^k e divida por $C(x)$
 - No ex. anterior $M(x) = x^7 + x^4 + x^3 + x^1$ e $k = 3$
 - $M(x) \times x^k = 1 \times x^{10} + 0 \times x^9 + 0 \times x^8 + 1 \times x^7 + 1 \times x^6 + 0 \times x^5 + 1 \times x^4 + 0 \times x^3 + 0 \times x^2 + 0 \times x^1 + 0 \times x^0$
 - $C(x) = 1 \times x^3 + 1 \times x^2 + 0 \times x^1 + 1 \times x^0$
 - O resto é 101
 - A subtração é feita por uma operação OR exclusivo (XOR).
 - O resultado é um resto de 3 bits, um bit a menos do que o polinômio gerador.
 - Pode ser implementado eficientemente em hardware.

ED: código de redundância cíclica



Problema 3: Controle de erro (CE)

- Duas abordagens gerais para tratamento de erros:
 - a) ARQ (Automatic-Repeat-reQuest) Solicitação automática de repetição e,
 - b) FEC (Forward Error Correction) Correção de erro adiante.

FEC

Utiliza códigos corretores de erro (códigos de blocos ou convolucionais).

Quando o receptor detecta a presença de erros no quadro de dados, localiza e corrige os erros.

É um esquema complexo que exige muito processamento. É utilizado em aplicações como comunicações por satélite, espacial e celular.

ARQ

Utiliza códigos detectores de erro. Mais utilizado em RC e será visto a seguir.

Solicitação automática de repetição (ARQ)

- Abordagem para conseguir confiabilidade no enlace:
 - Solicita retransmissão quando um quadro corrompido é detectado.
- Mecanismos para solicitar um transmissor para retransmitir:
 - O receptor envia confirmações (ACK acknowledgment) negativas para quadros corrompidos, ou
 - O receptor envia confirmações (ACK) positivas para quadros sem erros.

Solicitação automática de repetição (ARQ)

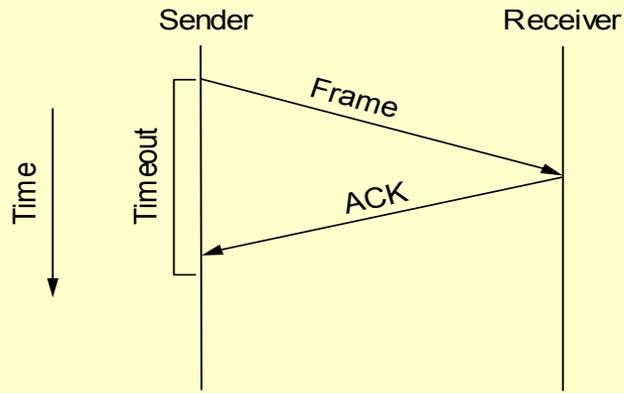
- Ambas abordagens requerem um mecanismo de temporização (timeout).
 - ACK negativo: Um temporizador é iniciado quando uma confirmação negativa é enviada.
 - ACK positivo: Um temporizador é iniciado quando um quadro é enviado.
- São feitas as retransmissões quando ocorrem os esgotamentos dos temporizadores:
 - ACK negativo: o receptor retransmite uma confirmação negativa.
 - ACK positivo: o transmissor retransmite um quadro.

Solicitação automática de repetição (ARQ)

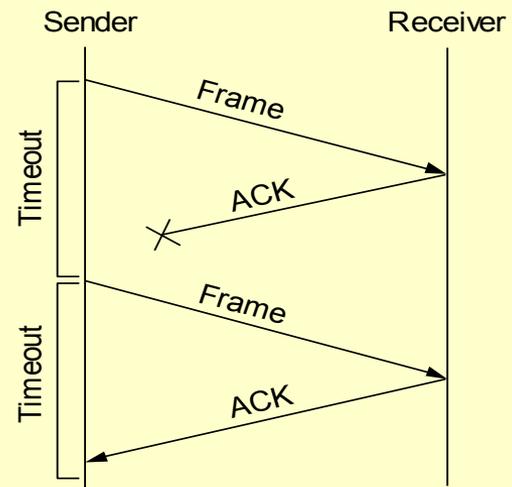
- O ARQ implementa um ACK positivo.
 - Somente após o recebimento de um ACK, um quadro será removido do buffer do transmissor
 - Um ACK pode ser colocado como apêndice (piggybacked) em uma mensagem enviada na direção oposta.
 - Um ACK indica, algumas vezes, o número de seqüência do próximo quadro esperado.
 - ACK positivo é normalmente acumulativo, isto é, a recepção de um ACK do quadro 4 implica que os quadros de 1 a 3 foram todos recebidos corretamente.

Stop-and-wait ARQ

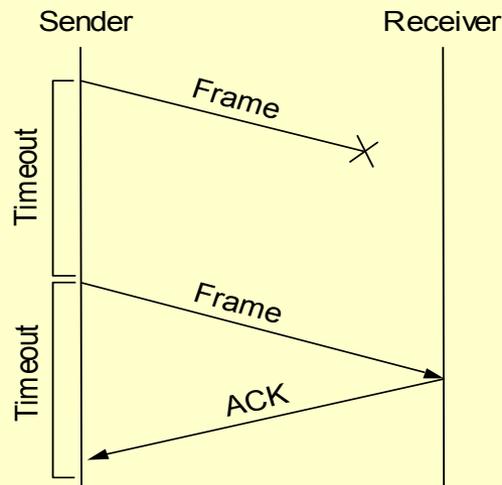
- O número máximo de quadros não confirmados é um.
 - Um transmissor não pode enviar um segundo quadro antes da recepção do ACK do primeiro quadro.
- O número mínimo para enumerar a seqüência de quadros é dois (0,1) i.e., somente um bit.
 - 0: para o quadro enviado e esperando por seu ACK.
 - 1: para o quadro a ser enviado após a recepção do ACK do quadro 0.



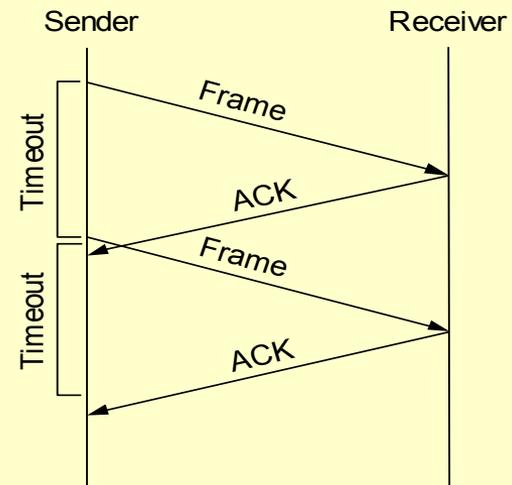
(a)



(c)



(b)



(d)

Stop-and-wait ARQ

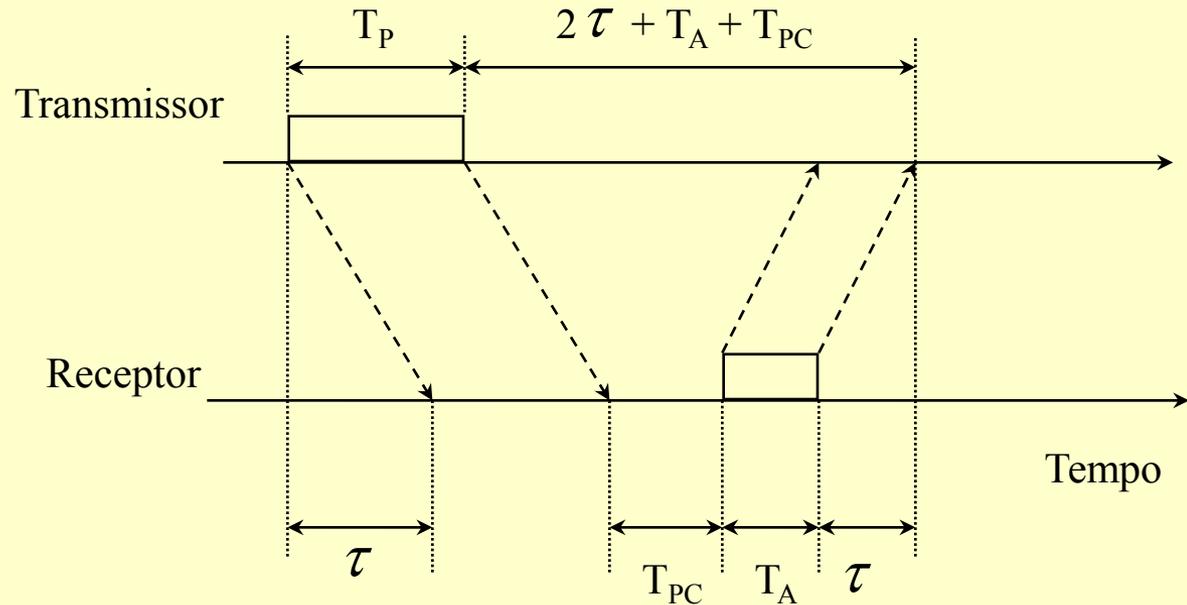
Eficiência do esquema Stop-and-Wait

T_p = tempo de transmissão de um pacote

T_A = tempo de transmissão de uma confirmação.

T_{PC} = tempo de processamento.

τ = atraso de propagação



O Time out deve ser maior do que $2\tau + T_A + T_{PC}$

Supondo que o transmissor tenha sempre pacote para transmitir, a eficiência S pode ser dada por

$$S = \frac{T_p}{T_p + 2\tau + T_{PC} + T_A}$$

Stop-and-wait ARQ

Exemplos numéricos.

Dados:

Quadro de informação = 10 000 bits e quadro de ACK = 100 bits.

Distância entre o transmissor e receptor = 10 km.

Velocidade propagação = 2×10^8 m/s.

$T_{PC} = 10 \mu \text{ seg.}$

a) Taxa de transmissão = 10 Kbps

$$T_P = \frac{10000 \text{ bits}}{10 \text{ Kbps}} = 1 \text{ seg.} \quad T_A = \frac{100}{10 \text{ Kbps}} = 10^{-2} \text{ seg.}$$

$$S = \frac{1}{1 + 10^{-4} + 10^{-2}} \cong 1$$

b) Taxa de transmissão = 100 Mbps

$$T_P = \frac{10000 \text{ bits}}{100 \text{ Mbps}} = 100 \mu \text{seg.} \quad T_A = \frac{100}{100 \text{ Mbps}} = 1 \mu \text{seg.}$$

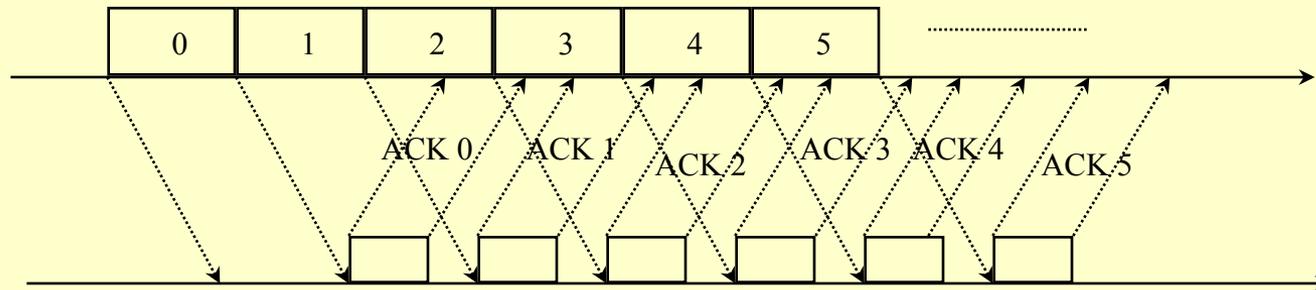
$$S = \frac{100}{100 + 100 + 10 + 1} \cong 0,5$$

O esquema é adequado para a taxa de transmissão baixa. Entretanto para taxas altas o esquema se torna ineficiente, porque gasta muito tempo esperando ACK. É fácil de implementar.

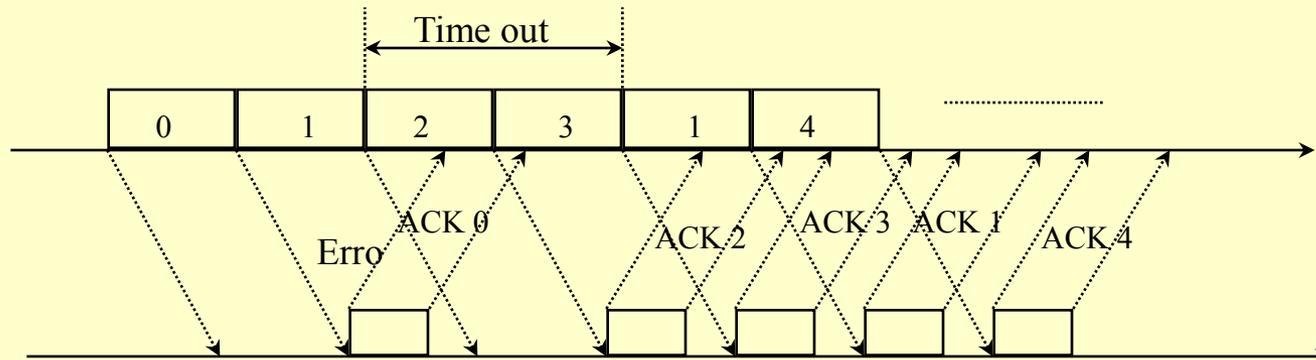
Controle de Erro

ARQ contínuo: o transmissor transmite continuamente quadros de informação sem esperar os ACKs.

Operação sem erro



Operação com erro – Rejeição Seletiva (Selective-reject)



Controle de Erro

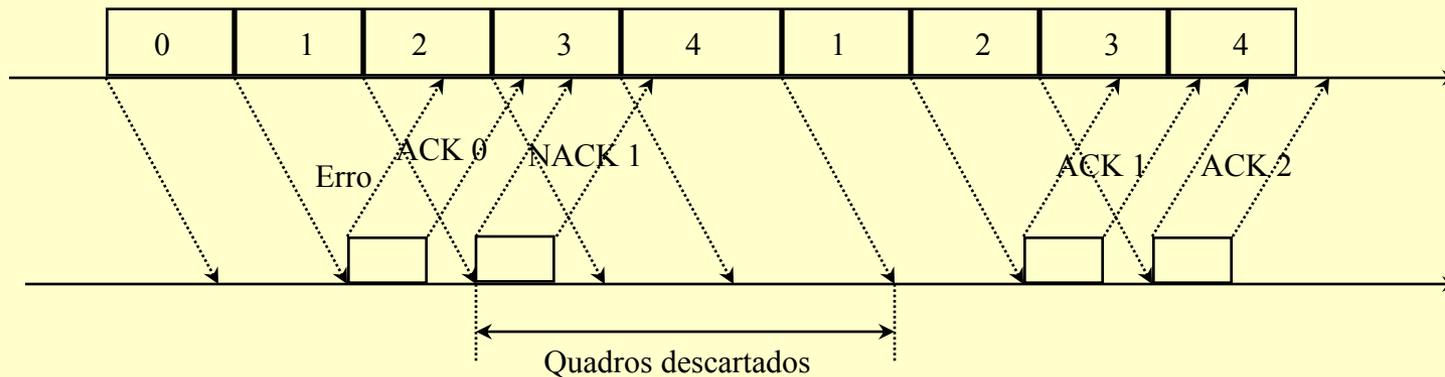
O esquema rejeição seletiva tem as seguintes desvantagens:

a) os quadros chegam fora de ordem

Deve-se esperar o quadro que falta para enviar a camada acima juntamente com outros quadros, pois os quadros em geral não são independentes.

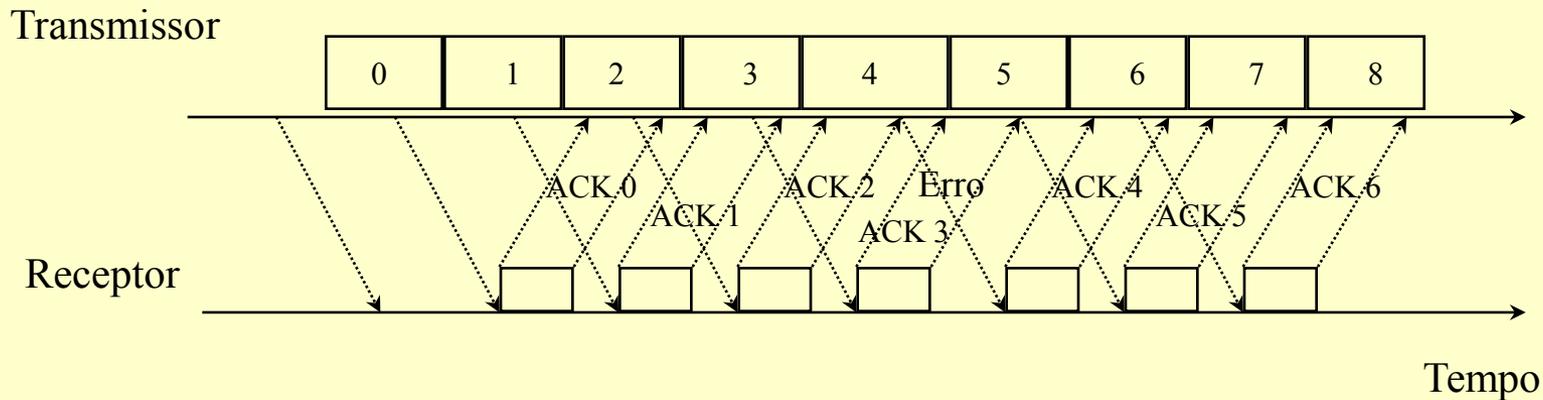
b) necessidade de gerenciamento de buffer.

Go-back-N - Um esquema para evitar armazenamento de quadros.



Controle de Erro

Go-back-N: Erro em ACK



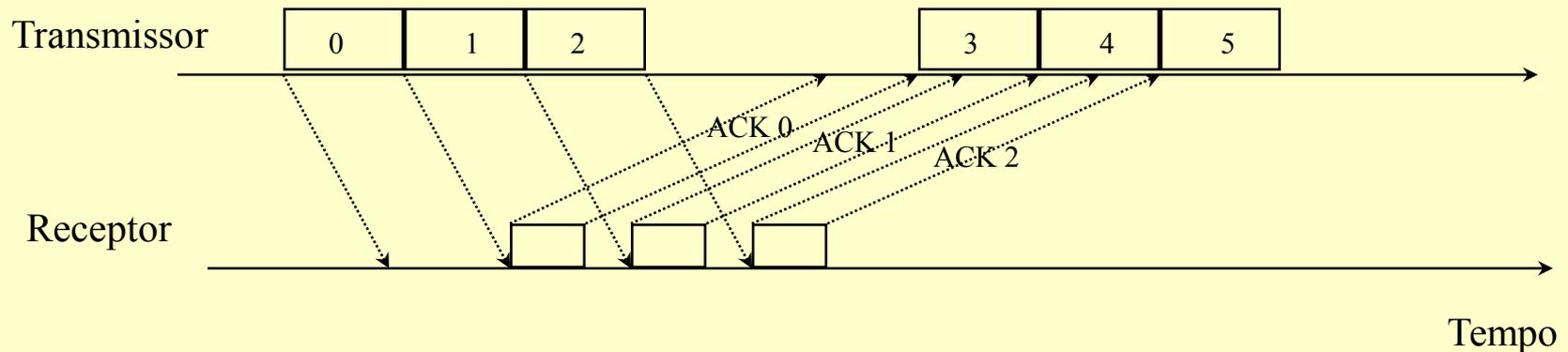
Houve erro no ACK 3 e não foi recebido pelo transmissor. Entretanto, chegou o ACK 4. O transmissor entende que o quadro 3 chegou ao receptor, mas houve erro no ACK 3 e, nenhuma providência será tomada.

Controle de Erro: Janela Deslizante

Neste esquema, o transmissor pode transmitir até W quadros de informação continuamente, antes de receber um ACK do receptor.

Exemplo de Operação

Tamanho da janela $W = 3$

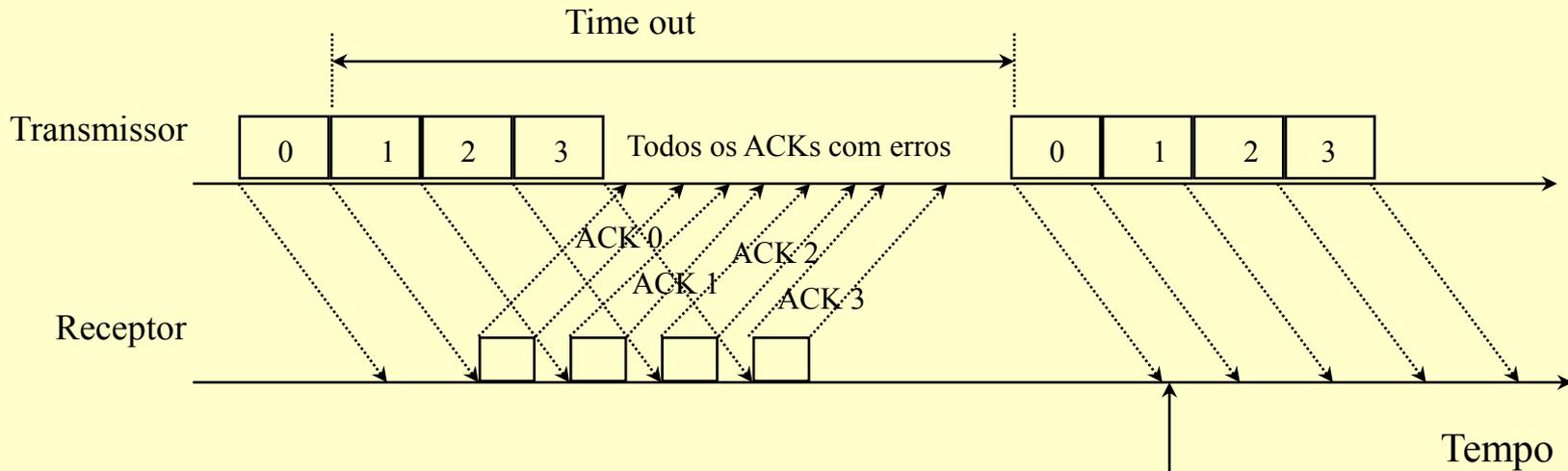


O transmissor pára de transmitir após três quadros sucessivos. Quando receber o ACK 0 pode transmitir um quadro, assim como quando receber os ACK 1 e ACK 2. Neste exemplo, após a transmissão do quadro 5, o transmissor necessita parar, pois não chegou mais ACK.

Controle de Erro: Janela Deslizante

Operação com erro em ACK

Vamos supor uma janela $W = 4$, uma numeração sequencial utilizando 2 bits e o protocolo go-back-N. Seja o exemplo da figura abaixo.

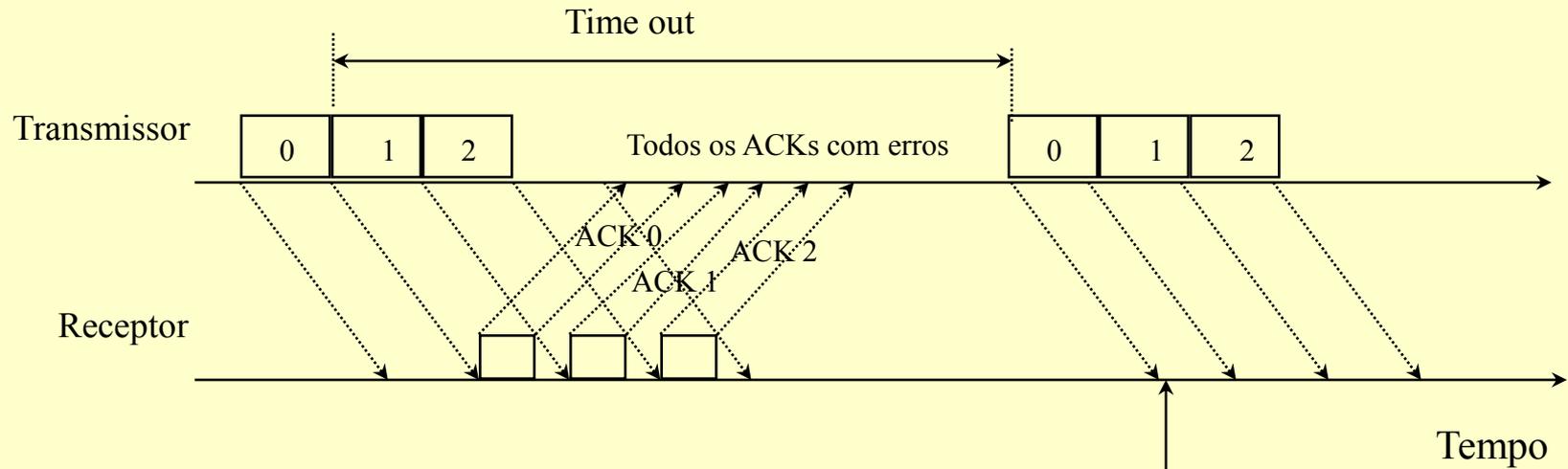


O receptor interpreta como um novo quadro e não de retransmissão.

No exemplo acima, teremos uma operação errada com o protocolo. Para evitar essa situação errônea devemos utilizar um tamanho de janela W menor do que 2^n onde n é número de bits de numeração. No exemplo acima, para $W = 3$, evita-se a operação incorreta.

Controle de Erro: Janela Deslizante

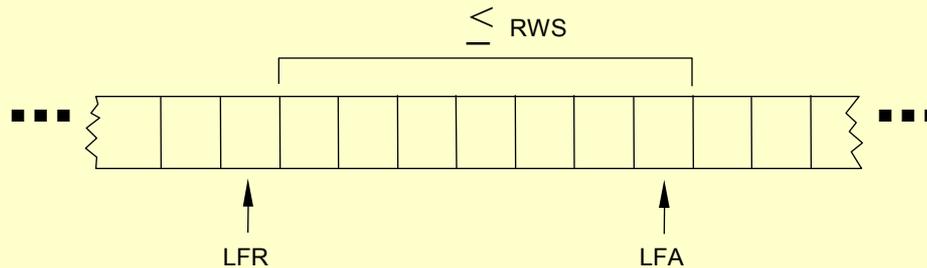
Operação com 2 bits e $W = 3$



O receptor reconhece que são quadros de retransmissão, pois aguardava como a seqüência correta, o número 3

SW: Receptor

- Mantenha três variáveis de estado
 - tamanho da janela de recepção (**RWS**)
 - maior quadro aceitável (**LFA**)
 - último quadro recebido (**LFR**)
- Mantenha a expressão: **LFA - LFR \leq RWS**



- Quadro **SeqNum** chega:
 - se **LFR < SeqNum \leq LFA** \longrightarrow aceita
 - se **SeqNum \leq LFR** ou **SeqNum > LFA** \longrightarrow descartado
- Envie ACKs acumulados

Exercícios

- 4.3 Para transmitir a mensagem 11001001 é utilizado um polinômio gerador $C(x) = x^3 + 1$ para fins de controle de erro.
 - a) Calcule a palavra de código que deve ser concatenada com a mensagem.
 - b) Suponha que o bit mais à esquerda da mensagem seja invertido devido a erro na transmissão. Qual é o resultado do cálculo do CRC do receptor? Como o receptor sabe que ocorreu erro?