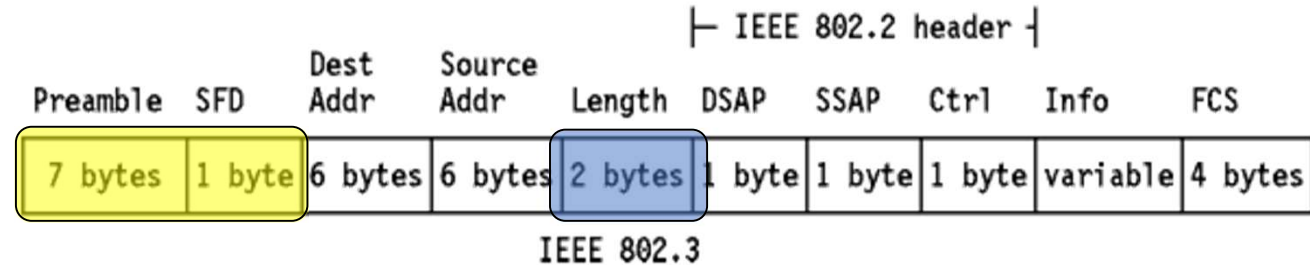


CP3: Nível de Ligação de Dados (refs.)

- Delimitação de tramas
(Peterson 2021, 2.3; Tanenbaum 2011, 3.1; Stallings Ch. 6)
- Controlo de erros e de fluxo
(Peterson 2021, 2.4; Tanenbaum 2011, 3.1; Stallings Ch. 7)
- Códigos detectores e correctores de erros
(Peterson 2021, 2.4; Tanenbaum 2011, 3.2; Stallings Ch. 6)
- Protocolos de controlo de erros e de fluxo e sua análise
(Peterson 2021, 2.5; Tanenbaum 2011, 3.3, 3.4 ; Stallings Ch. 7)
 - “Stop-and-Wait”
 - “Go-back-N”,
 - “Selective Repeat”
 - “Piggyback”
- Protocolos HDLC e PPP
(Peterson 2021, 2.3 ; Tanenbaum 2003, 3.6 ; Stallings Ch. 7)

CP3: Nível de Ligação de Dados (questões.)

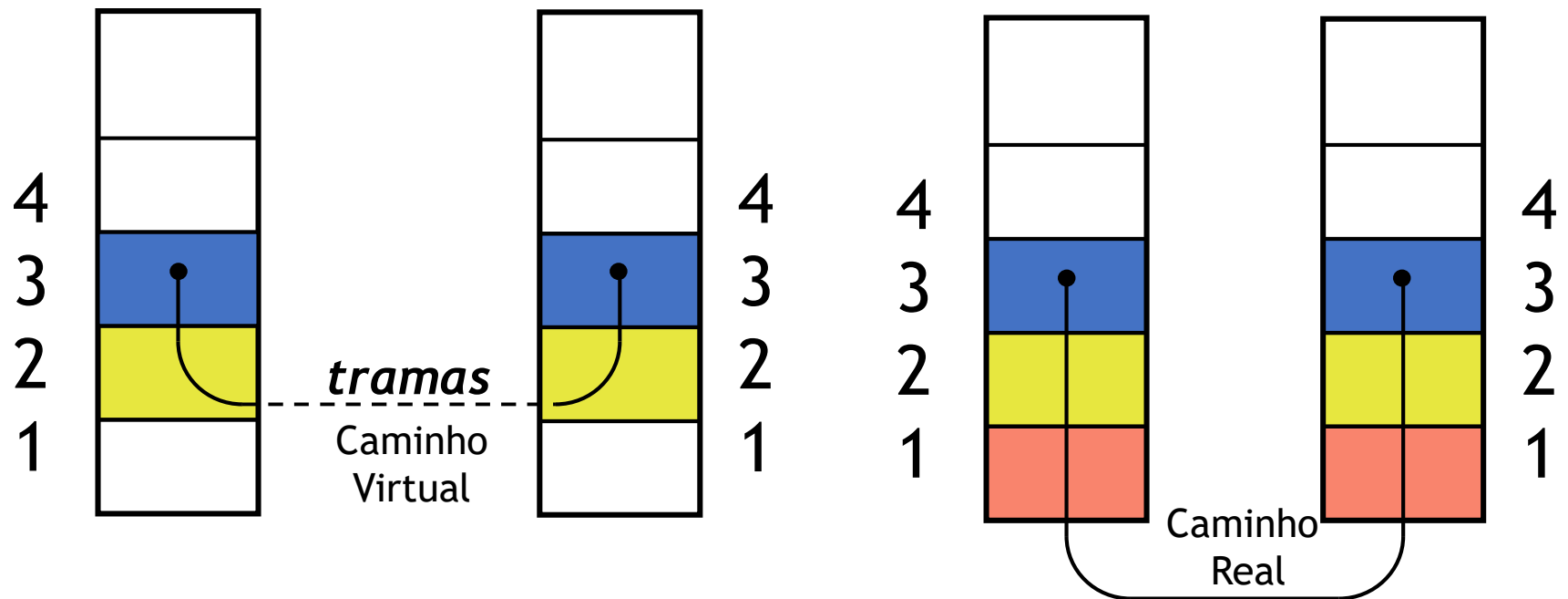
- Onde se situa na pilha protocolar OSI? (Funcionalidades.)
- Blocos de bits.
 - Sequências infinitas de bits ou blocos de bits?
(Como se assinala o principio e o fim de um bloco?)



- Como se detectam blocos de bits com erros?
- Como se corrigem blocos de bits com erros?
- Como pode o receptor controlar o ritmo do emissor?

Nível de Ligação de Dados (*Data Link Layer*)

Transporte de *tramas* entre nós adjacentes na rede. Presta serviços ao nível rede (L3) usando os serviços do nível físico (L1) - transporte tendo em conta as características do nível físico (e.g., atrasos, erros de bit, partilha do meio de transmissão).

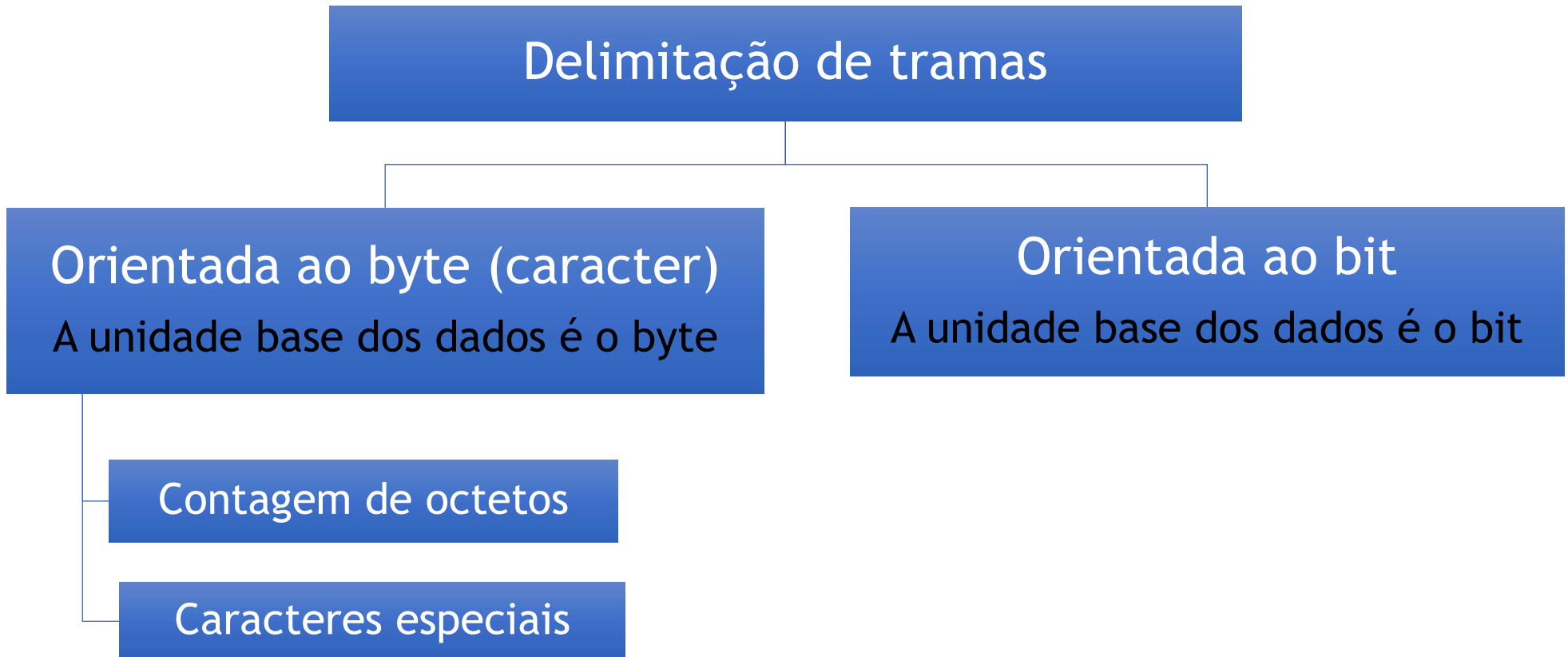


Funções do Nível 2 OSI

- Delimitação de tramas
- Transparência dos dados
- Controlo de erros: detecção e correcção
- Controlo de fluxo

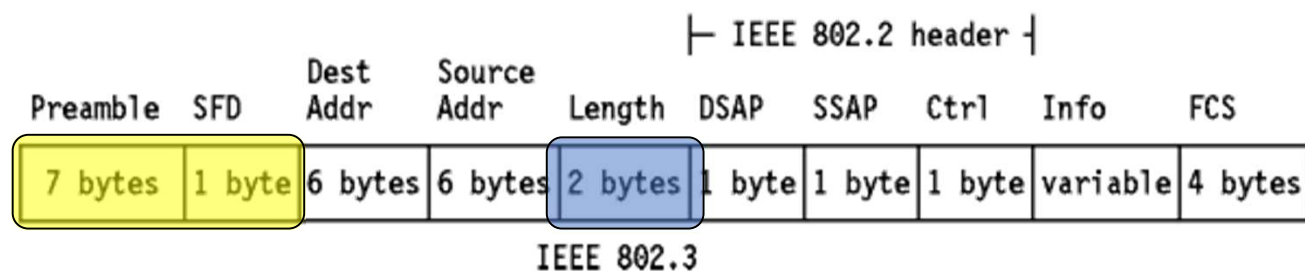
Delimitação de Tramas

Transmissão de dados (blocos de bits) em modo assíncrono (i.e., não sincronizada com um relógio) e sem um tamanho fixo pré-determinado.



Nível Ligação de Dados (*Data Link Layer*)

- Como se definem blocos de bits?
 - Delimitação de tramas.

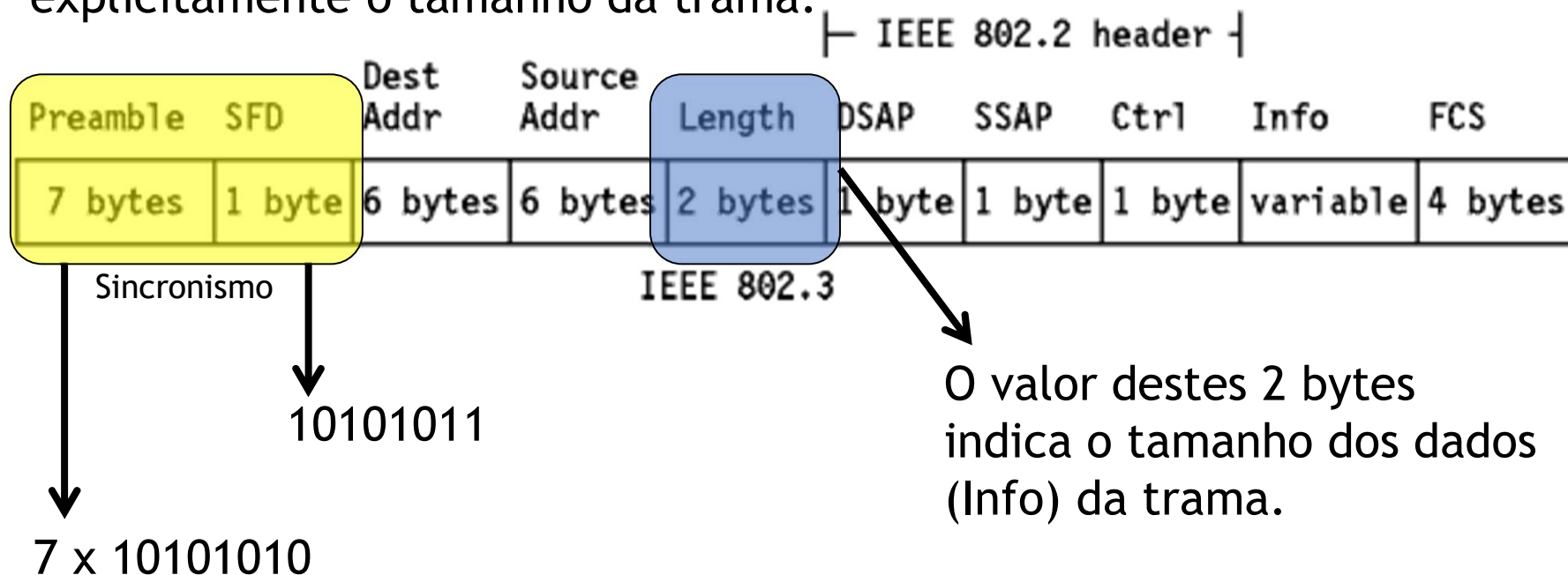


Delimitação de Tramas orientada ao byte(1)

- Contagem de octetos (e.g., Ethernet, IEEE 802.3)



O valor de um campo da informação de controlo (*header*) indica explicitamente o tamanho da trama.



Delimitação de Tramas orientada ao byte (2)

- Caracteres especiais



Caracteres especiais delimitam a trama e/ou parte desta.

Os caracteres dos dados (i.e., texto) possuem um conjunto de valores distinto dos caracteres especiais (e.g., Dados ASCII [32 - “ ” a 126 - “~”]).

Dec	Hx	Oct	Char
0	0	000	NUL (null)
1	1	001	SOH (start of heading)
2	2	002	STX (start of text)
3	3	003	ETX (end of text)
4	4	004	EOT (end of transmission)
5	5	005	ENQ (enquiry)
6	6	006	ACK (acknowledge)
7	7	007	BEL (bell)
8	8	010	BS (backspace)
9	9	011	TAB (horizontal tab)
10	A	012	LF (NL line feed, new line)
11	B	013	VT (vertical tab)
12	C	014	FF (NP form feed, new page)
13	D	015	CR (carriage return)
14	E	016	SO (shift out)
15	F	017	SI (shift in)
16	10	020	DLE (data link escape)
17	11	021	DC1 (device control 1)
18	12	022	DC2 (device control 2)
19	13	023	DC3 (device control 3)
20	14	024	DC4 (device control 4)
21	15	025	NAK (negative acknowledge)
22	16	026	SYN (synchronous idle)
23	17	027	ETB (end of trans. block)
24	18	030	CAN (cancel)
25	19	031	EM (end of medium)
26	1A	032	SUB (substitute)
27	1B	033	ESC (escape)

Delimitação de Tramas orientada ao bit

- Delimitação por bandeiras (*flags*)

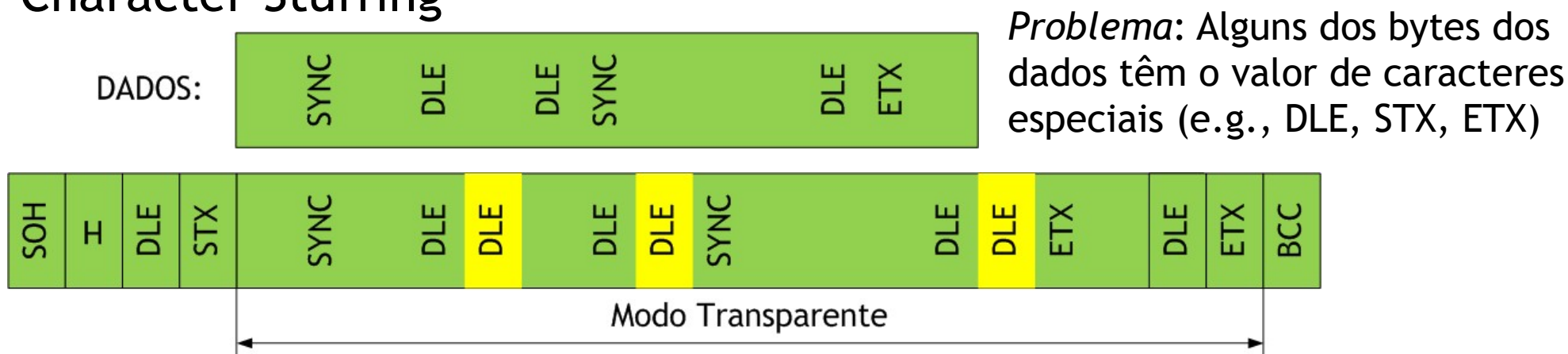


- Sequências pré-definidas de bits (*flags*) sinalizam o início e o fim de uma trama.
- O valor mais comum para esses bits é 01111110
- A sequência de bits dos dados não pode conter a “*flag*”.
Representa uma limitação grave!

Transparência

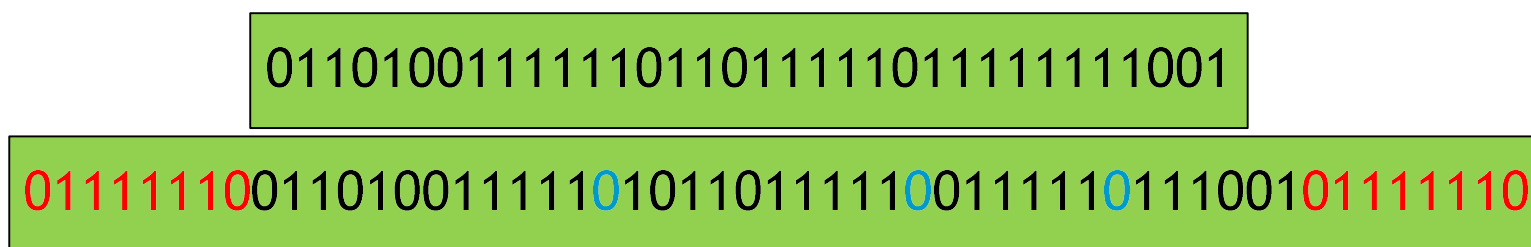
A propriedade de transparência dos dados permite que não haja qualquer restrição aos dados transportados (i.e., os bytes transportados podem ter qualquer valor de 0x00 a 0xFF, ou conter a sequência 01111110).

•Character Stuffing



Solução: Garantir uma interpretação única dos delimitadores (pode implicar *stuffing*.)

•Bit Stuffing (HDLC)



Nível Ligação de Dados (*Data Link Layer*)

- Como se detectam tramas com erros?

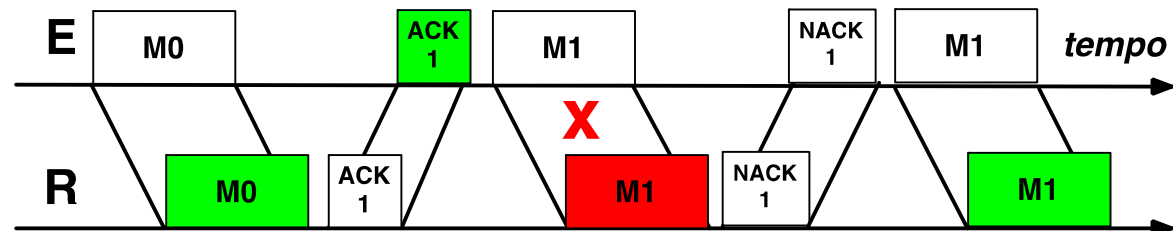
Códigos
e
Redundância

Mensagem ($m=1$)	Palavra código ($n=2$)
0	00
1	11

- Como se corrigem tramas detectadas como tendo erros?

Pelo receptor (FEC)

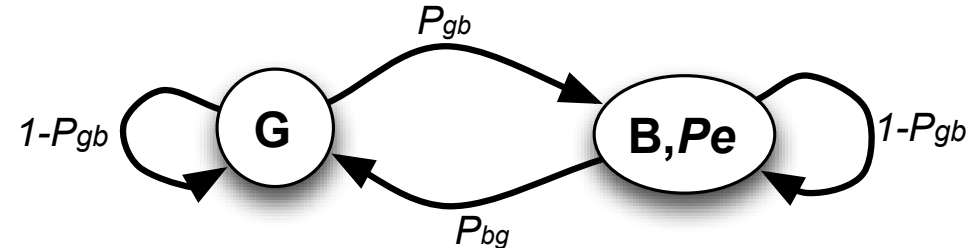
Pelo emissor (ARQ)



Detecção de Erros

- **Tipos de erros:**

- “Simples” e “Em rajada (*burst*)”



- **Técnicas para detecção:**

- Eco no receptor

- e.g., Terminal com eco

- Envio por dois caminhos distintos (diversidade)

- Informação redundante:

- Bit de paridade (paridade horizontal, vertical e cruzada)
- Código de bloco
- Código polinomial (CRC)

Correcção de Erros

- ***Forward Error Correction - FEC***

- Correção dos erros por parte do receptor através de informação redundante recebida.

- ***Automatic Repeat Request - ARQ***

- Correção dos erros por parte do emissor através de repetição, envolve as seguintes fases:
 - 1º: Detecção dos erros pelo receptor usando informação redundante (e.g., CRC)
 - 2º: Indicação do receptor para o emissor das tramas certas/erradas
 - 3º: Retransmissão automática das tramas erradas pelo emissor
- Exemplo:
 - Stop-and-Wait
 - Go-Back-N
 - Selective Repeat

Nível Ligação de Dados (*Data Link Layer*)

- Como se detectam/corrigem tramas com erros?
 - Tipos de erros de bit (recordar).
 - Códigos detectores/correctores de erros.
- Como comparar códigos para detectar/corrigir erros?
 - Taxa de redundância.
 - Distância de Hamming de um código.

Princípios fundamentais: Redundância

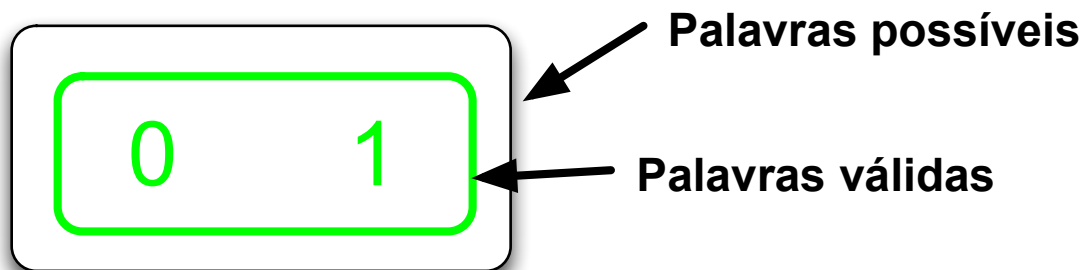
Para enviar m bits de mensagem, transmite-se $n = m + r$ bits, r bits são **redundantes**.

Taxa de redundância = r/n .

Taxa do código = m/n .

- Os n bits transmitidos definem as 2^n **palavras possíveis** de serem recebidas.
- Os m bits de mensagem definem as 2^m mensagens possíveis de serem transmitidas.
- A cada mensagem possível de ser transmitida corresponde uma palavra do código => Existem 2^m **palavras válidas** nesse código.
- E quando não há redundância? ($r=0$).

Exemplo $n=m=1$.



Conclusão:

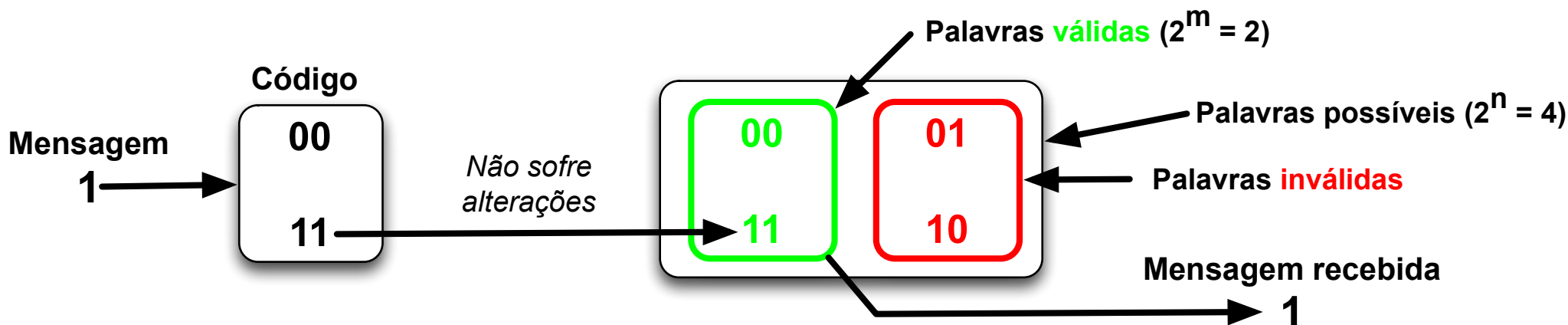
Todas as palavras possíveis são válidas, logo não se consegue detectar qualquer situação de erro.

Princípios fundamentais: Códigos (1)

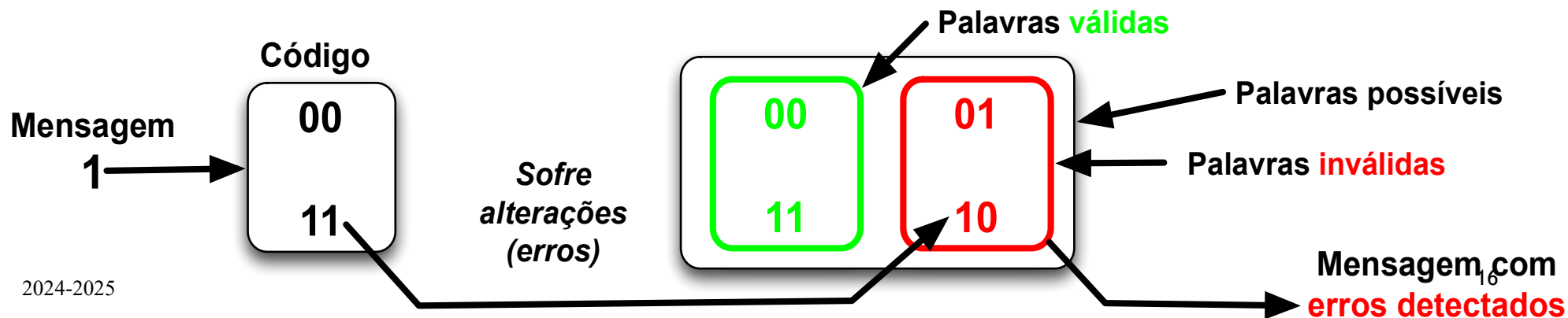
Mensagem ($m=1$)	Palavra código ($n=2$)
0	00
1	11

A cada mensagem passível de ser transmitida corresponde uma palavra do código.
=> Existem **2^m palavras válidas** no código.

- Palavra válida do código não é transformada/alterada.

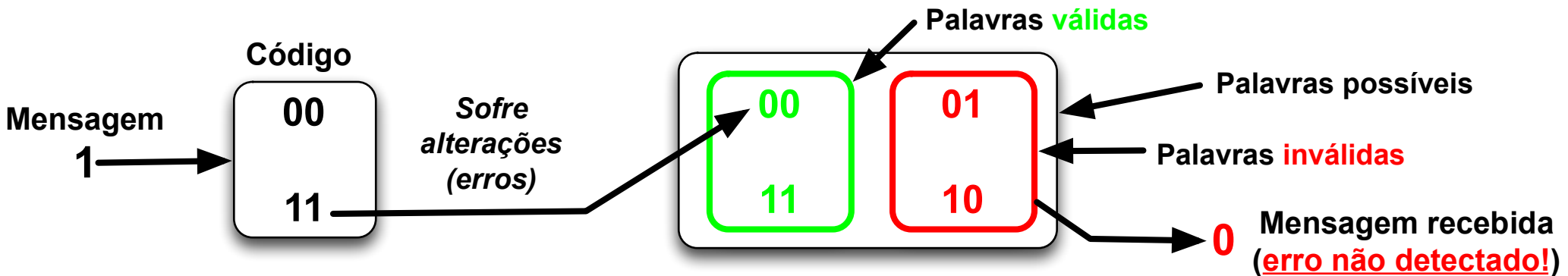


- Palavra válida do código é transformada numa palavra não válida.

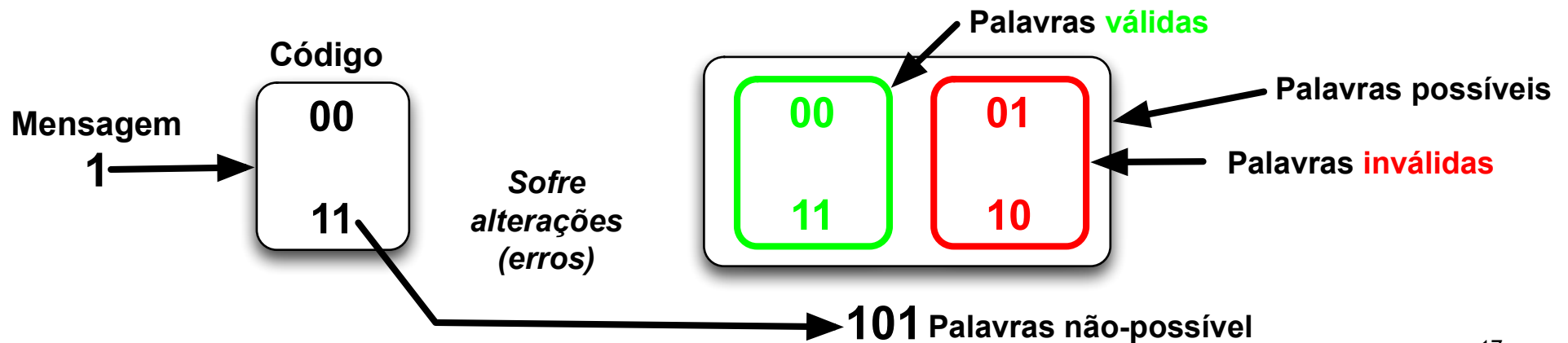


Princípios fundamentais: Códigos (2)

- Palavra válida do código é transformada numa palavra válida.



- Palavra válida do código é transformada numa palavra não possível.
(Situação não considerada!)



Distância de Hamming

- *Distância de Hamming entre duas palavras binárias:*

- **Definição:** Número de bits que se tem de alterar para passar de uma palavra para outra:

$$\begin{array}{r} 0110 \ 1100 \\ \oplus \ 1100 \ 1101 \\ \hline 1010 \ 0001 \Rightarrow d = 3 \end{array}$$

- No caso particular em que se compara a sequência de bits transmitida, T , e recebida, R , a distância de Hamming representa o número de bits errados.



Richard Hamming (1915-1998) é um dos pioneiros no estudo das questões relacionadas com a detecção e correcção de erros.

A distância de Hamming representa uma ferramenta teórica importante na definição destes processos.

Distância de Hamming de um código

Mensagem	Palavra código
00	0000000000
01	0000011111
10	1111100000
11	1111111111

$$(m = 2, n = 10) \Rightarrow r = n - m = 8$$

- 0000011111 => Palavra possível e válida (i.e. do código)
(Estará correcta?)
- 0010011101 => Palavra possível mas não válida
(Qual será a correcta?)

Distância de *Hamming* de um código, d :

- **Definição:** *Distância de Hamming mínima entre qq duas palavras do código.*

No caso exemplo $d = 5$.

- **Detecção de erro:** Não há nenhuma palavra válida à distância de Hamming 0 da palavra recebida, i.e., corresponde a uma palavra não válida
 - Consegue detectar **de certeza** até $(d-1)$ bits errados, em situações com d ou mais erros não se pode ter a certeza quanto à sua detecção embora esta possa ocorrer.
Ex: Recebe 1000011110 -> tem de certeza erros!
Ex: Recebe 0000011111 -> deve estar certa!
- **Correcção:** Escolhe a palavra válida à menor distância de *Hamming* da palavra recebida
 - Consegue corrigir **correctamente** até $(d-1)/2$ bits errados.
Ex: Envia 0000011111 e recebe 1100111111 -> tem erros de certeza, a palavra mais próxima da recebida é 1111111111 => “correção” errada!

Nível Ligação de Dados (*Data Link Layer*)

- Exemplo de código para detectar tramas com erros?
 - Código “Bit de paridade”.
 - Códigos de redundância cíclica (CRC).
- Exemplo de código para detectar e corrigir tramas com erros?
 - Códigos de Hamming.

Bit de Paridade ($r=1$)

Paridade Par

e.g., 10110101
101101011
100101011 (5 1s => erro!)

e.g., 00001111
000011110
000011101 ~~correcto~~

Paridade Ímpar

e.g., 10110101
101101010

e.g., 00000000
000000001

e.g., 11111111
111111111

- Expressão para o cálculo do bit paridade par:

$$P = X_1 \oplus X_2 \oplus X_3 \oplus \cdots \oplus X_m$$

- Distância de *Hamming* do Código ‘Bit de Paridade’: $d = 2$

Códigos Polinomiais (*Cyclic Redundancy Code* - CRC)

- Baseia-se na associação entre sequências binárias e polinómios: em que os coeficientes do polinómio são os dígitos da sequência binária.

Exemplo (o polinómio de M):

$$M = 101001 \rightarrow M(x) = 1 \cdot x^5 + 0 \cdot x^4 + 1 \cdot x^3 + 0 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0 = x^5 + x^3 + 1$$

- As operações realizadas são em aritmética módulo 2.

Adição:

$$\begin{aligned} M1(x) + M2(x) &= (x^5 + x^3 + 1) + (x^5 + x^4 + x^2) \\ &= (1+1)x^5 + x^4 + x^3 + x^2 + 1 \\ &= x^4 + x^3 + x^2 + 1 \end{aligned}$$

$$\begin{array}{r} M1 \oplus M2 \quad 101001 \\ \oplus 110100 \\ \hline 011101 \end{array}$$

Multiplicação:

$$\begin{aligned} M(x) \cdot x^3 &= (x^5 + x^3 + 1) \times x^3 \\ &= x^8 + x^6 + x^3 \end{aligned}$$

$$\begin{array}{r} 3 \text{ bits} \leftarrow 101001 \\ 101001000 \end{array}$$

Códigos Polinomiais (*Cyclic Redundancy Code* - CRC)

- Método CRC:
 - Dada uma trama M com m bits.
 - O **emissor** gera uma sequência de r bits (**Frame Check Sequence - FCS**) com base numa determinada sequência de bits: o polinómio gerador G de grau r (G é conhecido pelo **emissor** e **receptor**, faz parte do protocolo usado por ambos);

$$FCS(x) = \text{Resto} \left[\frac{M(x) \cdot x^r}{G(x)} \right]$$

- A trama a transmitir, T , resultante da concatenação de M e FCS , tem $m+r$ bits e é divisível pela sequência de bits do polinómio gerador G ;

$$T = M \parallel FCS \quad \leftrightarrow \quad T(x) = M(x) \cdot x^r + FCS(x)$$

$$\text{Resto} \left[\frac{T(x)}{G(x)} \right] = 0$$

- O **receptor** divide a trama recebida R pela mesma sequência de bits, G , e se o resto for zero assume que não há erros.

$$\text{Resto} \left[\frac{R(x)}{G(x)} \right] = 0 \rightarrow \text{assume-se que } R(x) \text{ não tem erros.}$$

Códigos Polinomiais (*Cyclic Redundancy Code* - CRC)

- Detalhes do critério no receptor (assumir que não há erro):

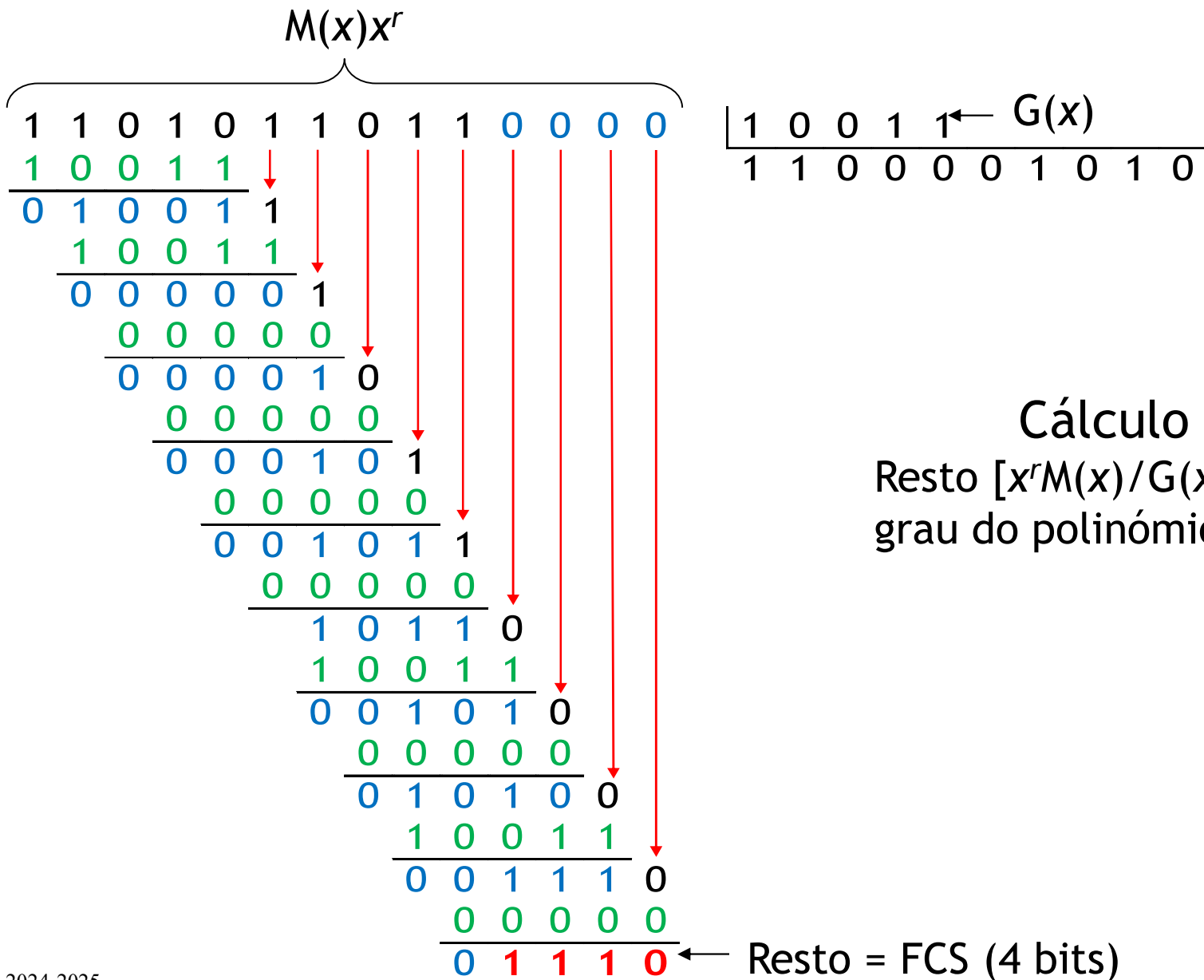
$$\text{Resto} \left[\frac{R(x)}{G(x)} \right] = \text{Resto} \left[\frac{T(x) + E(x)}{G(x)} \right], \quad E(x) \text{ padrão de erros}$$

$$\text{Resto} \left[\frac{T(x)+E(x)}{G(x)} \right] = \text{Resto} \left[\frac{T(x)}{G(x)} \right] + \text{Resto} \left[\frac{E(x)}{G(x)} \right] \text{ (necessita demonstração)}$$

$$\text{Resto} \left[\frac{R(x)}{G(x)} \right] = \text{Resto} \left[\frac{E(x)}{G(x)} \right]$$

$$\text{Resto} \left[\frac{R(x)}{G(x)} \right] \begin{cases} \neq 0 \rightarrow E(x) \neq 0, \text{ a trama recebida } R \text{ tem erro(s) de certeza} \\ \\ 0 \begin{cases} E(x) = 0, \text{ a trama recebida } R \text{ não tem erros} \\ E(x) = Z(x)G(x) \neq 0 \text{ a trama recebida } R \text{ tem erros não detectados!} \end{cases} \end{cases}$$

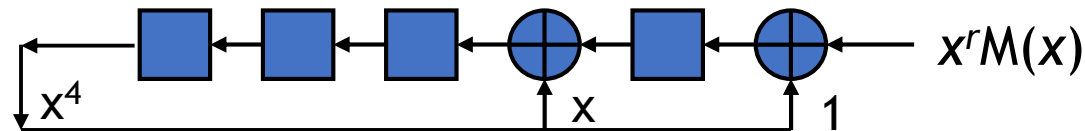
CRC: Exemplo



Cálculo da FCS:
Resto $[x^r M(x)/G(x)]$ em que r é o grau do polinómio $G(x)$.

CRC - Implementação

- Bloco a transmitir (M): 1101011011
- Polinómio gerador $G(x) = x^4 + x + 1 \rightarrow 10011$ (grau de $G(x) = 4$)
- Dados + FCS: 11010110111110
- Registo de deslocamento síncrono:



- O registo de deslocamento contém r bits (nº de bits do FCS);
- Nº de bits da FCS é igual ao grau de $G(x)$;
- Existem no máximo r ou-exclusivos;
- A presença ou ausência de um ou-exclusivo corresponde à presença ou à ausência do termo no polinómio $G(x)$.

Exercício 3.2

Considere uma ligação lógica conforme com a família de protocolos HDLC, em que se adoptou o polinómio $G(x) = x^4 + x^3 + 1$ para o cálculo dos campos CRC das tramas.

- a) Calcule a informação para controlo de erros a acrescentar à mensagem: '0111 1011'. Escreva a sequência a transmitir, admita que o primeiro bit a transmitir é o da esquerda.
- b) Considere o seguinte padrão erros: '0011 0010 0000'.
 - i. Indique o conjunto de bits recebido.
 - ii. Justifique, sem efectuar a divisão, se esta trama seria detectada como estando errada (ou não) pelo receptor.
 - iii. Considerando que os erros na trama ocorrem de forma independente e que a probabilidade de erro de bit, $P_b = 0,25$, indique qual a probabilidade de ocorrência deste padrão de erros.
- c) Expresse como uma sequência binária o padrão de erros $E(x) = (x^3 + x^2)G(x)$ e calcule a sua probabilidade de ocorrência.

Exercício 3.4

Considere que recebeu a seguinte sequência binária:

‘0011 1111 0011 1110 1011 1101 1111 1011’

- a) Assinale as bandeiras (*flags*) idênticas ao HDLC (01111110) presentes na sequência recebida.
- b) Remova os bits de enchimento (*stuffing*) da mensagem, delimitada pelas bandeiras assinaladas na alínea anterior.
- c) Decida sobre a validade da mensagem tendo em conta que o polinómio gerador de código usado foi $G(x) = x^3 + x + 1$.

Nota: Desenhe o registo de deslocamento que implementa o polinómio gerador usado.

Nível Ligação de Dados (*Data Link Layer*)

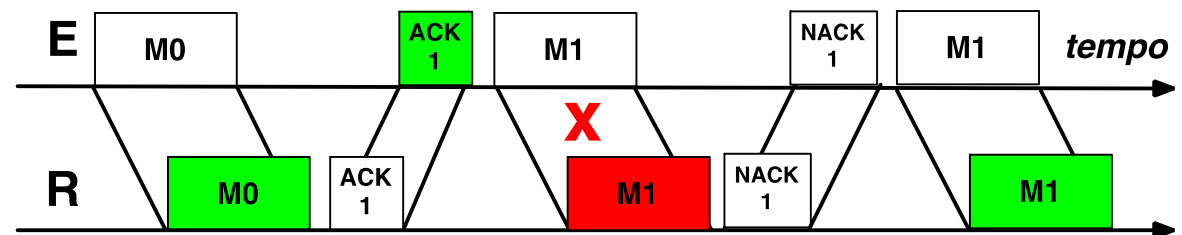
•Como se corrigem tramas detectadas como tendo erros?

- Caso o código permita corrigir erros (e.g., Hamming)

Efectuado no receptor (**FEC - Forward Error Correction**)

- Caso o código apenas permita detectar erros (e.g., CRC)

Pelo emissor (**ARQ**)
colaborando com o
receptor - **protocolo!**



Protocolos do Nível de Ligação de Dados

- **Objectivos:**

- **Controlo de erros por ARQ:** mecanismo para controlar a retransmissão de dados em caso de erro
- **Controlo de fluxo:** mecanismo para restringir a quantidade de dados que o emissor pode enviar para o receptor

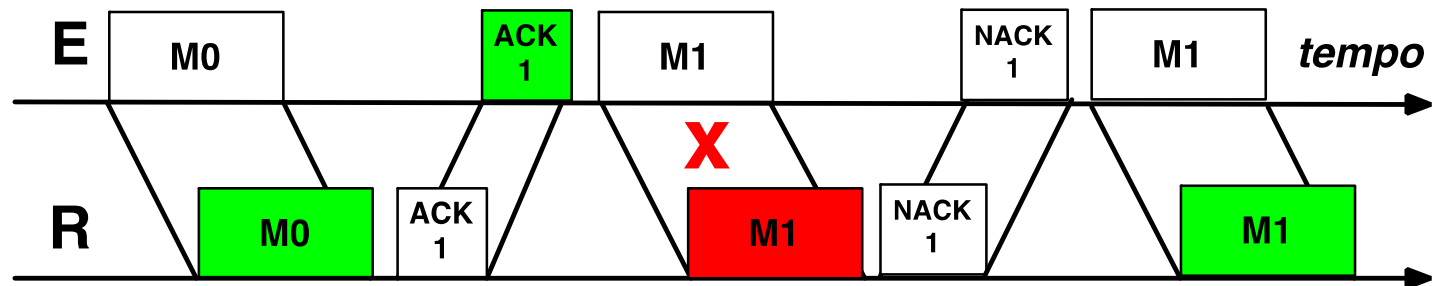
- **Famílias de protocolos:**

- *Stop-and-Wait*
- *Sliding Window:*
 - *Go-Back-N*
 - *Selective Repeat*

Erros na Transmissão e ARQ

Correcção dos erros por retransmissão automática por parte do emissor.
Utilização de avisos de recepção (positivos - ACK e negativos - NACK) e temporizadores
(*Nota*: num ACK o valor n indica o próximo número de sequência esperado).

- Uso de NACK para sinalizar tramas com erro (em falta) e que devem ser repetidas



- Perda/erro na trama
Utilização de *timers* (temporizadores)



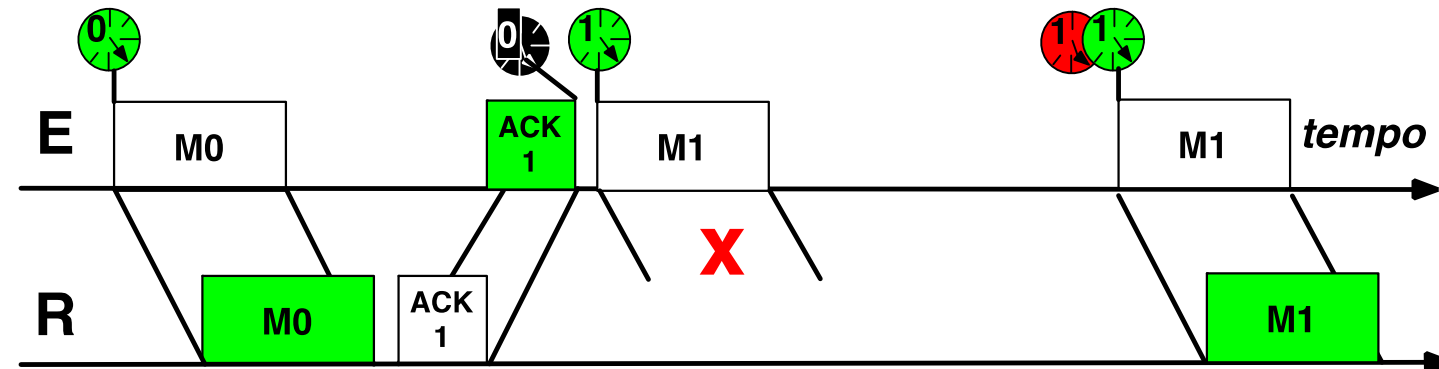
Início do *timer*



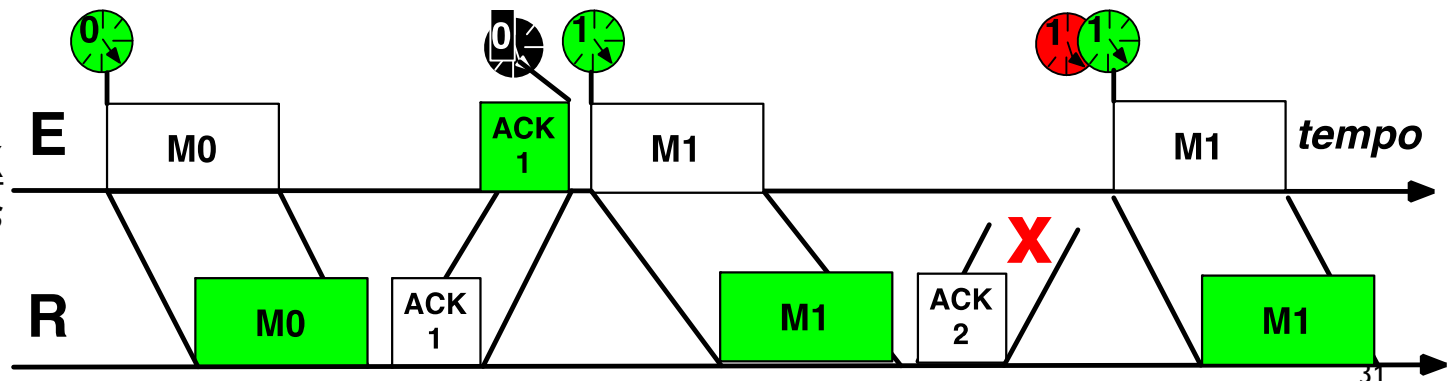
Desligar o *timer*



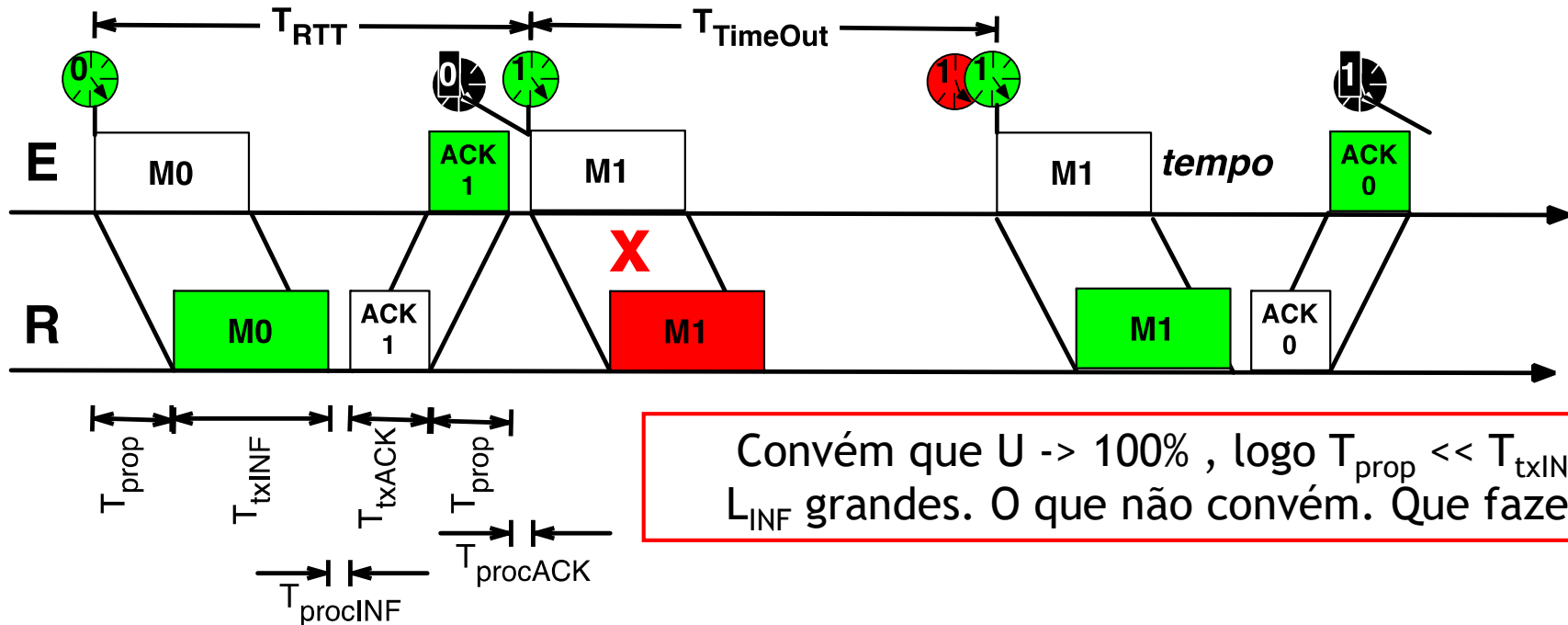
Disparo do *timer*



- Perda/erro no ACK
Utilização de *timers* (temporizadores)

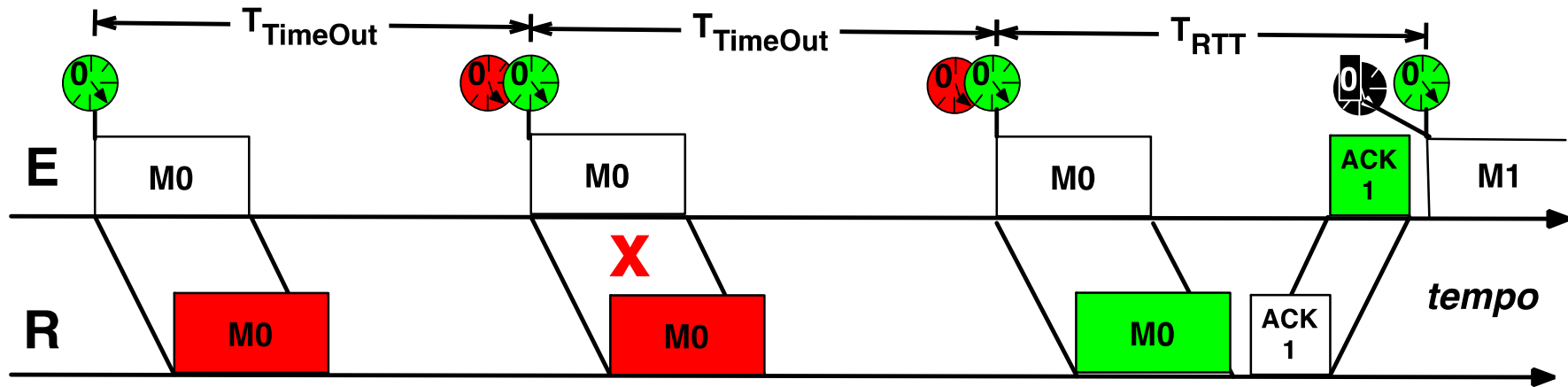


ARQ: Protocolo *Stop-and-Wait (S&W)*



- Necessita apenas um bit (0,1) para numeração/identificação das tramas
- *Round-Trip-Time* (T_{RTT}): $T_{RTT} = T_{txINF} + T_{prop} + T_{procINF} + T_{txACK} + T_{prop} + T_{procACK}$
 T_{txINF} : tempo tx tramas de informação, T_{txACK} : tempo tx tramas de ACK
 $T_{procINF}$: tempo *processamento* tramas de informação, $T_{procACK}$: tempo *processamento* tramas de ACK
- *Valor mínimo do temporizador* ($T_{TimeOut}$): $T_{TimeOut} > T_{RTT}$
- *Taxa de utilização do canal* ($U_{S\&W}$): $U_{S\&W} = T_{txINF} / T_{RTT}$
 $U_{S\&W} = T_{txINF} / T_{RTT} = T_{txINF} / (T_{txINF} + T_{prop} + T_{procINF} + T_{txACK} + T_{prop} + T_{procACK})$
 se $T_{procINF}$, T_{txACK} , $T_{procACK} \approx 0$ então $U_{S\&W} \approx 1/(1+2a)$ com $a = T_{prop} / T_{txINF}$

Desempenho protocolo *Stop&Wait* com erros



- Taxa de utilização do canal com erros ($U_{S\&W_erros}$):

$$U_{S\&W_erros} = \frac{T_{tx}}{(E[N_{tx}] - 1) \cdot T_{TimeOut} + T_{RTT}}$$

$\xrightarrow{\text{Tempo útil}}$
 $\xrightarrow{\text{Tempo de um ciclo}}$

- Valor esperado do número de transmissões ($E[N_{tx}]$):

$$\hat{N}_{tx} = E[N_{tx}] = \sum_{i=1}^{\infty} i \cdot p_i = \sum_{i=1}^{\infty} i \cdot (1 - p_{sucesso})^{i-1} \cdot p_{sucesso} = 1/p_{sucesso}$$

p_i , probabilidade de serem necessárias i transmissões
 $i-1$, insucessos
 sucesso na i -ésima tentativa

Exercício 3.5

Considere uma comunicação com um protocolo Stop-and-Wait com um fluxo unidireccional de tramas de informação. Admita que o comprimento das tramas de informação é de 100 bytes e as de confirmação (ACK) é de 10 bytes.

- Calcule a probabilidade de corrupção de cada um dos tipos de trama ($P_{erroINF}$ e $P_{erroACK}$) considerando que a probabilidade de erro de bit é $P_b = 10^{-4}$. Assuma que a probabilidade de erro de bit é independente.
- Calcule a probabilidade de à transmissão de uma trama de informação não suceder a recepção correcta de uma trama de confirmação (P_R).
- Calcule o número médio de retransmissões de uma trama de informação.
- Calcule a probabilidade de o terminal emissor terminar as tentativas de retransmissão em virtude de se terem verificado 16 “time-outs” sucessivos (P_T).

Nota: se necessário use o facto de $\sum_{n=1}^{\infty} n \times p^n = \frac{p}{(1-p)^2}$

Exercício 3.6

Considere um protocolo do tipo “*stop-and-wait*” com tramas de informação de 1000 bit e de acknowledge de 100 bit em que apenas à fluxo de informação num sentido, e de confirmação no outro. Os ACK são enviados imediatamente após a recepção de uma trama de informação. A distância da ligação é de 200 km ($v_p = 200\,000$ km/s) e o débito da transmissão é de 2Mbit/s.

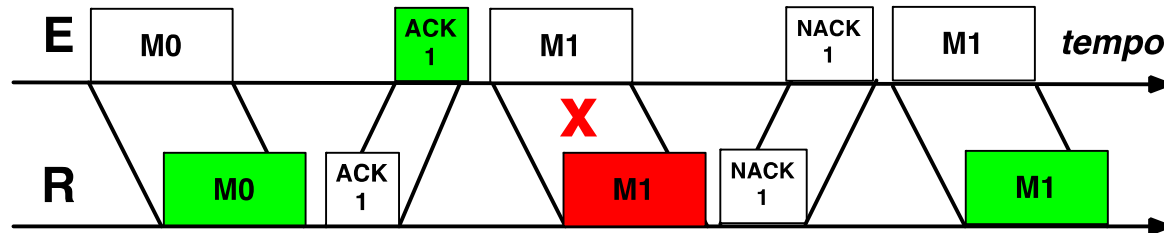
- a) Dimensione o temporizador do emissor (i.e., o valor do “*time-out*”), supondo que este é activado no instante em que se inicia a transmissão da trama.
- b) Quanto tempo demoraria a transmitir 1 Gbit de informação admitindo a ausência de erros neste sistema.
- c) Considere agora que a probabilidade de uma trama de informação chegar com erros é de 0.1 e que a probabilidade de uma trama de confirmação (“ACK”) chegar com erros é de 0.01. Calcule o tempo demoraria a transmitir 1 Gbit e admitindo que as tramas são retransmitidas apenas por “*time-out*”. Considere para esta alínea o valor de 3 ms para o temporizador do emissor.

Nível Ligação de Dados (*Data Link Layer*)

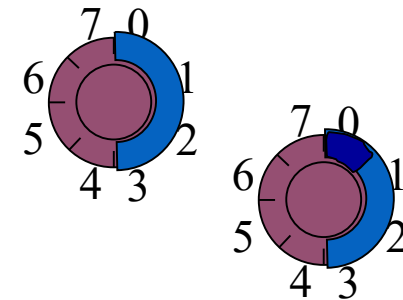
• Como se poderá melhorar a taxa de utilização do *Stop&Wait*?

• O que é o factor mais limitativo?

• Como se poderiam enviar várias tramas sem ter de esperar?



Usar a memória do emissor e receptor:
Janelas de transmissão e recepção.

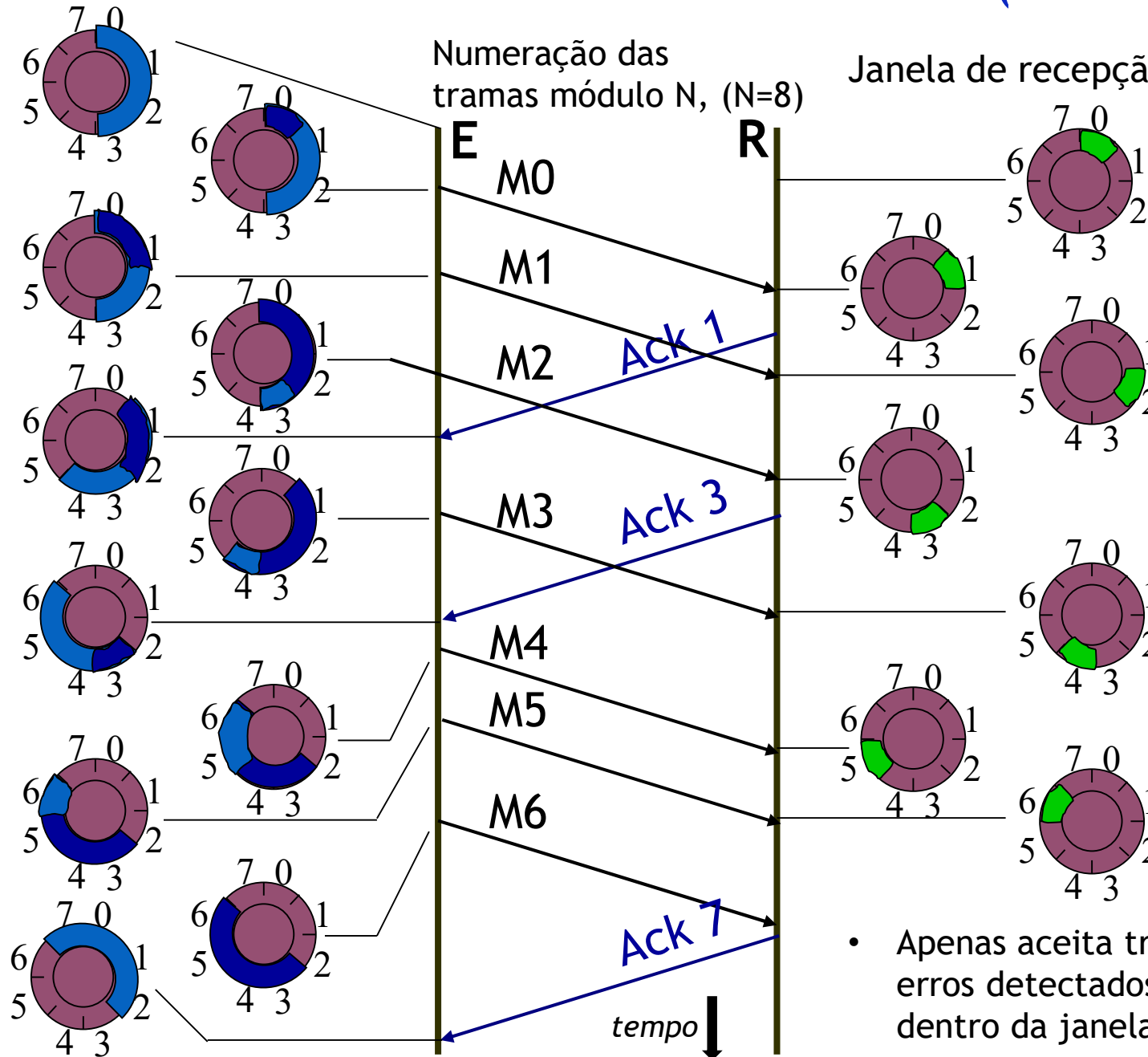


Janela Deslizante (*Sliding Window*)

Janela de transmissão = 4

Numeração das
tramas módulo N, (N=8)

Janela de recepção = 1



Funcionamento do Emissor

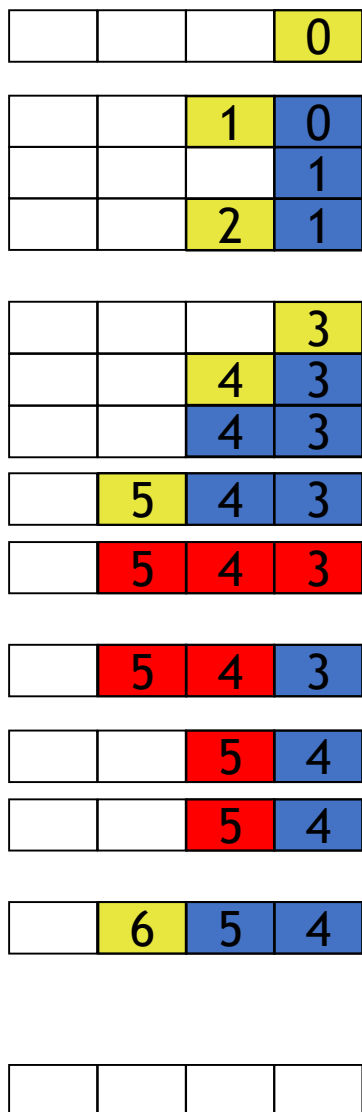
- Pode enviar tramas sem receber Ack até esgotar a janela de transmissão.
- Ao receber um Ack \underline{n} com \underline{n} pertencendo às tramas ainda sem aviso de recepção, retira a trama $\underline{n-1}$ e todas as anteriores da janela (efeito **cumulativo**).

Funcionamento do Receptor

- Apenas aceita tramas que não tenham erros detectados e que se encontrem dentro da janela de recepção.

Indicação de erro implica a retransmissão de todas as tramas deste a sinalizada com “em falta/errada”!

Tamanho da janela de tx, $W = J_{tx} = 4$



Go-Back-N (GB-n)

Tamanho da janela de rx, $J_{rx} = 1$



- 1ª transmissão
- À espera de Ack
- Para retransmissão

- Valor máximo para J_{tx} : $J_{tx} \leq N-1$
- Transmissão sem parar (espera) se:

$$T_{RTT} \leq T_{txINF} \times J_{tx}$$

↓
Chegada ACK, liberta janela

↓
Esgota janela

- Desempenho (sem erros):

$$U_{GB-N} = \begin{cases} 1, & \text{caso } T_{RTT} \leq T_{tx} \cdot J_{tx} \\ \frac{T_{tx} \cdot J_{tx}}{T_{RTT}}, & \text{caso contrário} \end{cases}$$

- Garante a entrega ordenada, sem falhas ou duplicados!

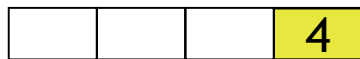
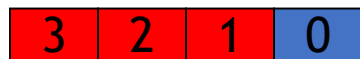
tempo ↓

$$J_{tx} = N-1 \quad \checkmark$$

Limite da janela de Tx do Go-Back-N

Enviar: CatarinaPais

Ack 0



M 0=Ca

M 1=ta

M 2=ri

M 3=na

M 0=Ca

M 1=ta

M 2=ri

M 3=na

M 4=Pa

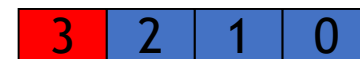
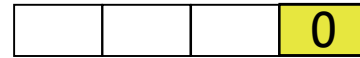
M 0=is



Recebe: CatarinaPais

Enviar: ManuelCapelo

Ack 0



M 0

M 1

M 2

M 3

M 0

M 1

M 2

M 3

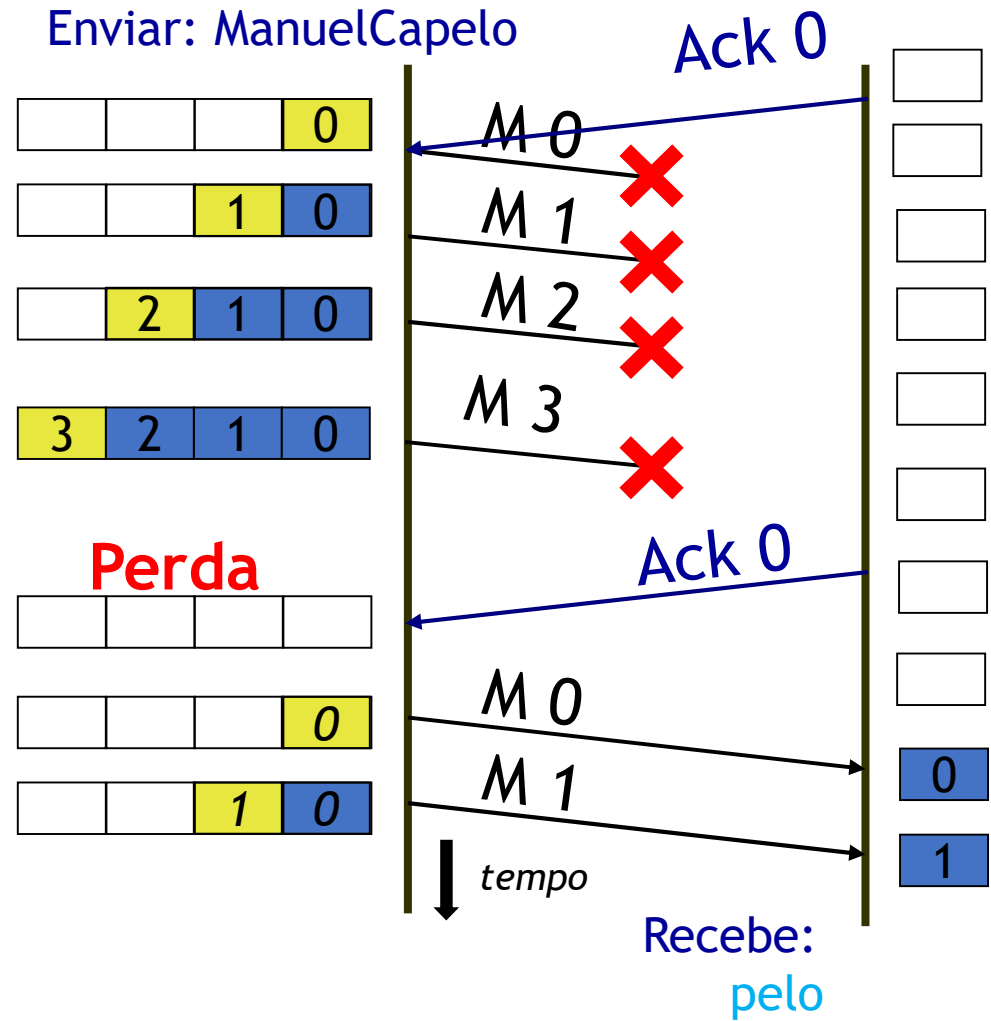
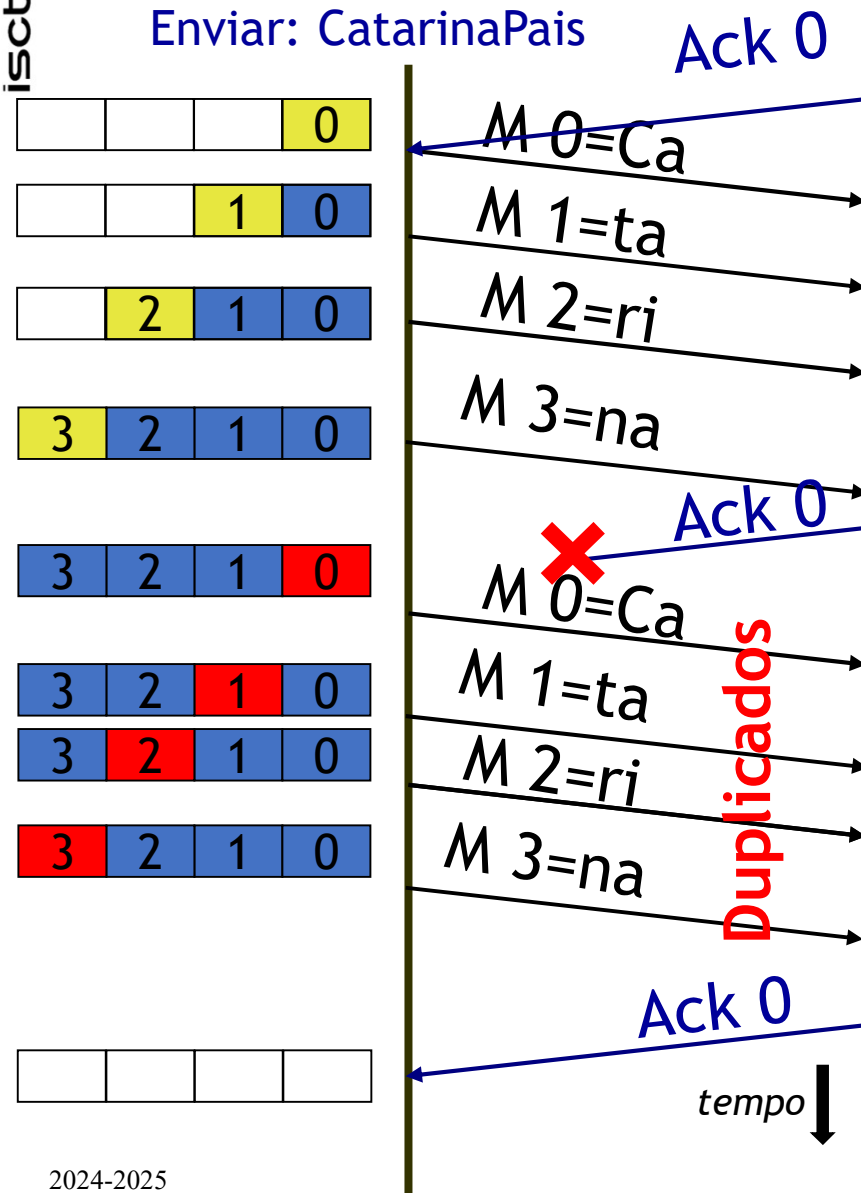
M 4

M 0

Recebe: ManuelCapelo

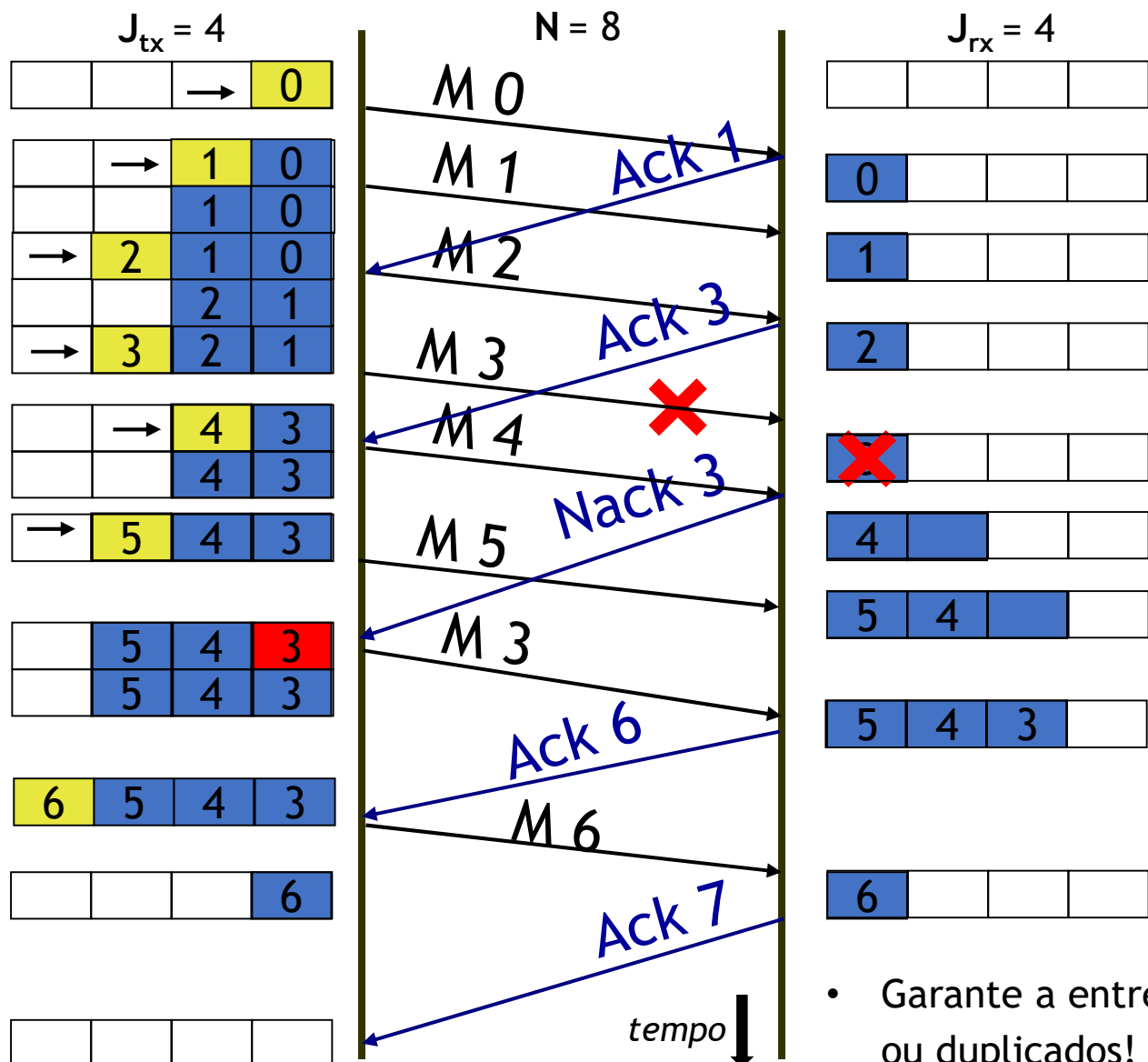
Violação do limite da janela de *tx* Go-Back-N

$$J_{tx} = N \times$$



ARQ: Protocolo *Selective Repeat*

No protocolo GB-n a necessidade de retransmitir todas as tramas na janela tx desde a sinalizada como “errada/em falta” tem implicações negativas no seu desempenho.



Passa a trama para o nível de rede

Valor máximo para J_{tx} :
 $J_{tx} \leq N/2$

Transmissão sem parar:

$$T_{RTT} \leq T_{txINF} \times J_{tx}$$

Desempenho

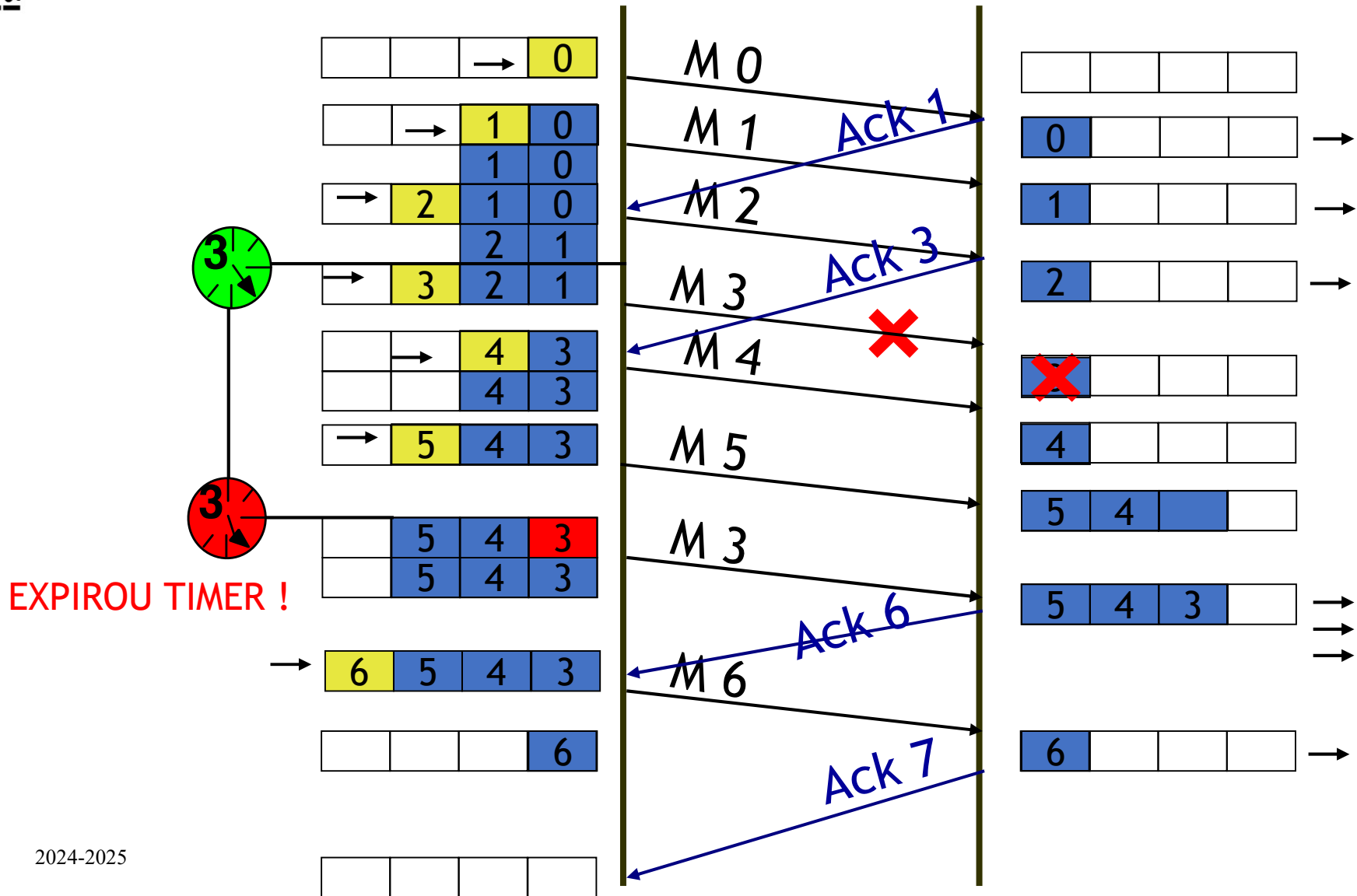
(sem erros):

$$U_{SR} = \begin{cases} 1, & \text{caso } T_{RTT} \leq T_{tx} \cdot J_{tx} \\ \frac{T_{tx} \cdot J_{tx}}{T_{RTT}}, & \text{caso contrário} \end{cases}$$

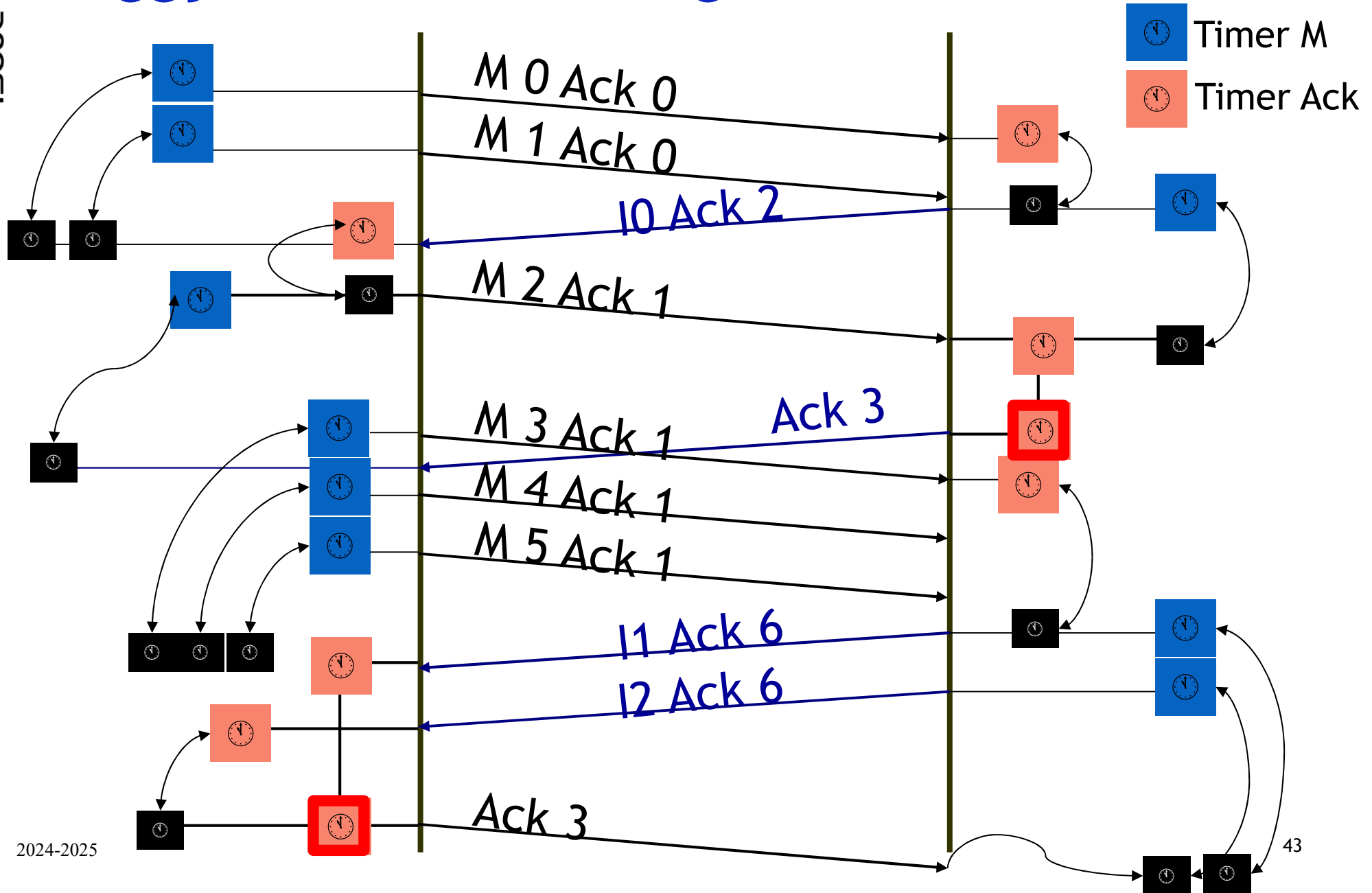
- Garante a entrega ordenada, sem falhas ou duplicados!

Selective Repeat (variante com temporizador)

Usado conjuntamente com NACK o dimensionamento do *timer* é dinâmico e diferente do usado no *S&W* ou *GB-N*.

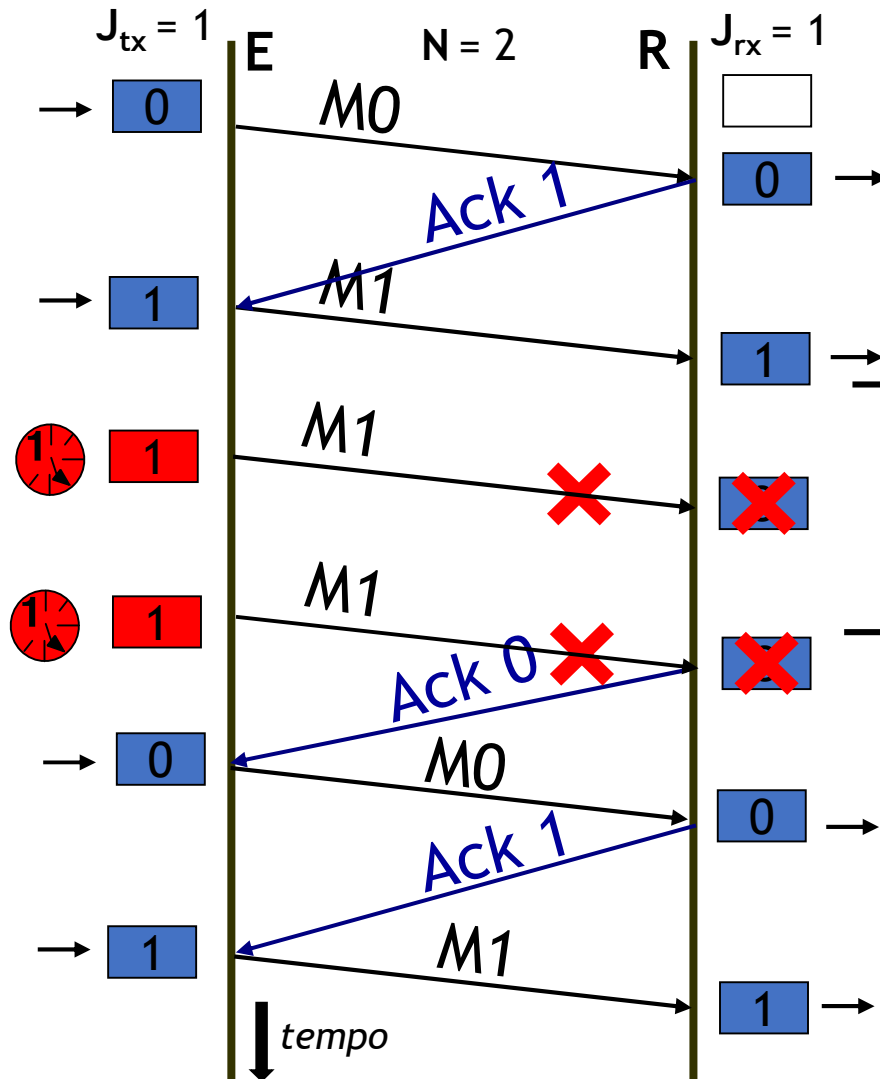


Piggyback Acknowledgement



Controlo de Fluxo

O receptor controla o ritmo a que o emissor envia tramas novas.



Enquanto o receptor pode receber tramas envia Ack ao emissor o que lhe permite enviar novas tramas.

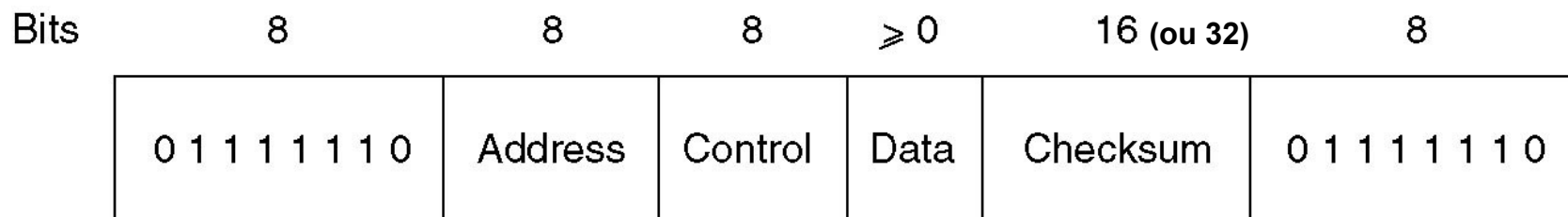
Quanto o receptor não pode/pretende receber tramas não envia Ack ao emissor.

O emissor passa apenas a retransmitir tramas que são descartadas pelo receptor.

Quanto o receptor pretende receber novas tramas volta a enviar Ack ao emissor.

High-Level Data Link Control - HDLC

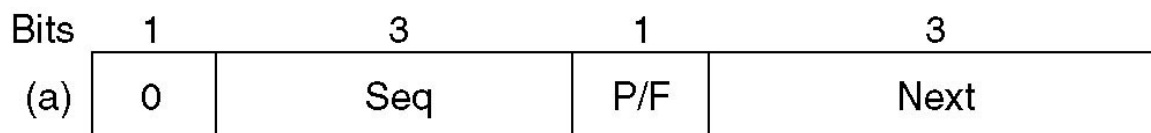
• Formato da trama



CRC-16-CCITT

$$G(x) = x^{16} + x^{12} + x^5 + 1$$

• Campo de controlo



Trama de informação



Trama de supervisão



Trama não numerada

TYPE

00 - ACK

01 - NACK

10 - RECEIVE NOT READY (confirma todas as tramas até Next, mas não quer receber mais)

11 - SELECTIVE REJECT (pede retransmissão da trama Next)

Exercício 3.7

Considere uma ligação lógica “full-duplex” sob uma linha digital a 384 kbit/s, de 3000 km de comprimento, cuja velocidade de propagação é 200 000 km/s. Admita que as tramas de informação têm 1584 bits de comprimento e que as tramas de “Acknowledge” de 48 bits de comprimento são expedidas imediatamente após a recepção de uma trama de informação correcta. Despreze os tempos de processamento dos equipamentos terminais bem como a probabilidade de perda ou corrupção das tramas transmitidas.

- a) Para uma estratégia de controlo de erros “Go-Back-N”, em que a numeração cíclica das tramas de informação seja efectuada com identificadores de 3 bits e em que o fluxo de informação seja unidireccional, calcule a taxa de utilização do meio de transmissão.
- b) Para as duas estratégias de controlo de erros, “Go-Back-N” e “Selective Repeat”, em que o fluxo de informação seja unidireccional, calcule o número mínimo de identificadores e o número de bits necessário para os representar de modo a que o emissor nunca esgote a sua janela de transmissão.

Exercício 3.8

Considere uma ligação lógica “full-duplex” sobre uma linha digital de 2,048 Mbit/s entre dois interlocutores distantes 500 km ($v_p = 250\,000$ km/s).

Admita que se usa a estratégia de controlo de erros “stop-and-wait”, que as tramas de informação tem um comprimento de 1024 bits (dos quais 48 são de controlo) e que as tramas de confirmação (ACK) têm um comprimento de 48 bits.

O temporizador do emissor foi fixado em 20 ms. É utilizado “piggybacking”.

- Explique em que consiste a técnica do “piggybacking” e quando é vantajoso o seu uso.
- Qual o tempo máximo de espera do envio da confirmação de uma mensagem (ACK) contado em relação à recepção de uma trama de informação correcta.
- Admita que o tempo máximo de espera é de 15 ms, e que o tempo médio de espera quando se usa efectivamente o “piggybacking”, é de 10 ms. Verifica-se que em 45% dos casos o “acknowledge” é enviado através de “piggybacking”. Considerando que um dos interlocutores tem sempre informação para transmitir, determine a sua taxa média útil de transmissão.

CP3: Nível de Ligação de Dados (Exercícios)

- Sebenta de Exercícios e Aplicações
 - Capítulo 3
- Questões de provas escritas
 - Ver plataforma *moodle* com provas de anos anteriores.
- Mini-teste
 - Mini-teste para avaliação na plataforma *moodle*.