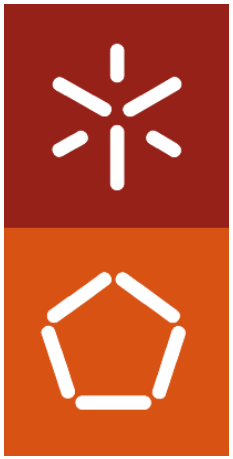
Nível Lógico

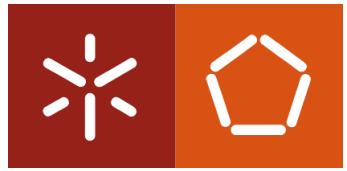
Mestrado Integrado em Engenharia de Comunicações

3º ano, 1ºsemestre

2012/2013



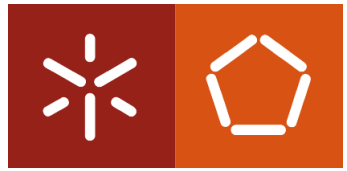
Sumário



- Funções da camada 2 do modelo de referência
- Controlo de Fluxo
  - Stop and Wait, Sliding Window
- Detecção de Erros
  - Paridade, Checksum, CRC
- Controlo de Erros
  - Stop and Wait ARQ, Go-Back-N ARQ, Selective-Reject ARQ
- Disciplina da linha
- Técnicas de Acesos ao Meio Físico

Este módulo é maioritariamente baseado no livro
William Stallings, *Data and Computer Comunications*

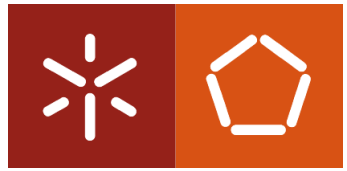
Controlo da ligação de dados



- A existência de ligações físicas e a transmissão de sinais analógicos ou digitais, por si só, não garantem a comunicação de dados entre entidades residentes em diferentes estações.
- A troca de dados entre entidades que pretendem comunicar deve ser regulada a fim de se criar um contexto comum e um sincronismo entre elas.
- As *regras* resultantes constituem o que se designa por *protocolo de comunicação*.
- Os *protocolos de ligação lógica* ou *ligação de dados constituem* o primeiro nível de troca ordenada, controlada e fiável de dados entre sistemas interligados por meio de uma ligação física.

Controlo da ligação de dados

introdução: funções distintivas dos níveis físico e lógico



Nível físico

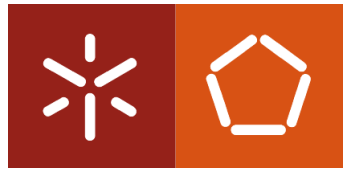
- envio de um sinal sobre um meio de transmissão
- sincronismo (nível do bit)
- codificação de linha
- modulação do sinal
- multiplexagem física
- interface com o meio

Nível de ligação lógica

- estrutura das tramas
- configuração e acesso à linha
- endereçamento
- controlo de fluxo
- controlo de erros
- gestão da ligação (controlo da troca de dados)

Controlo da ligação de dados

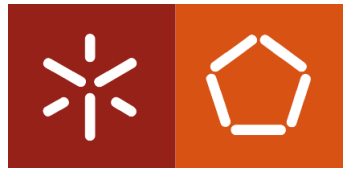
principais funções de um protocolo de ligação



- **definição da trama** - formato da unidade de dados (PDU)
- **configuração da linha** - considera a topologia, define a disciplina de acesso à linha e a sua duplexidade
- **endereçamento** - identifica os interfaces das estações que podem enviar e receber tramas
- **controlo de fluxo** - regula a cadência de tramas enviadas
- **controlo de erros** - detecta erros de transmissão e executa procedimentos de recuperação
- **gestão da ligação** - define como se faz o estabelecimento, a manutenção e a terminação da associação lógica.

Controlo da ligação de dados

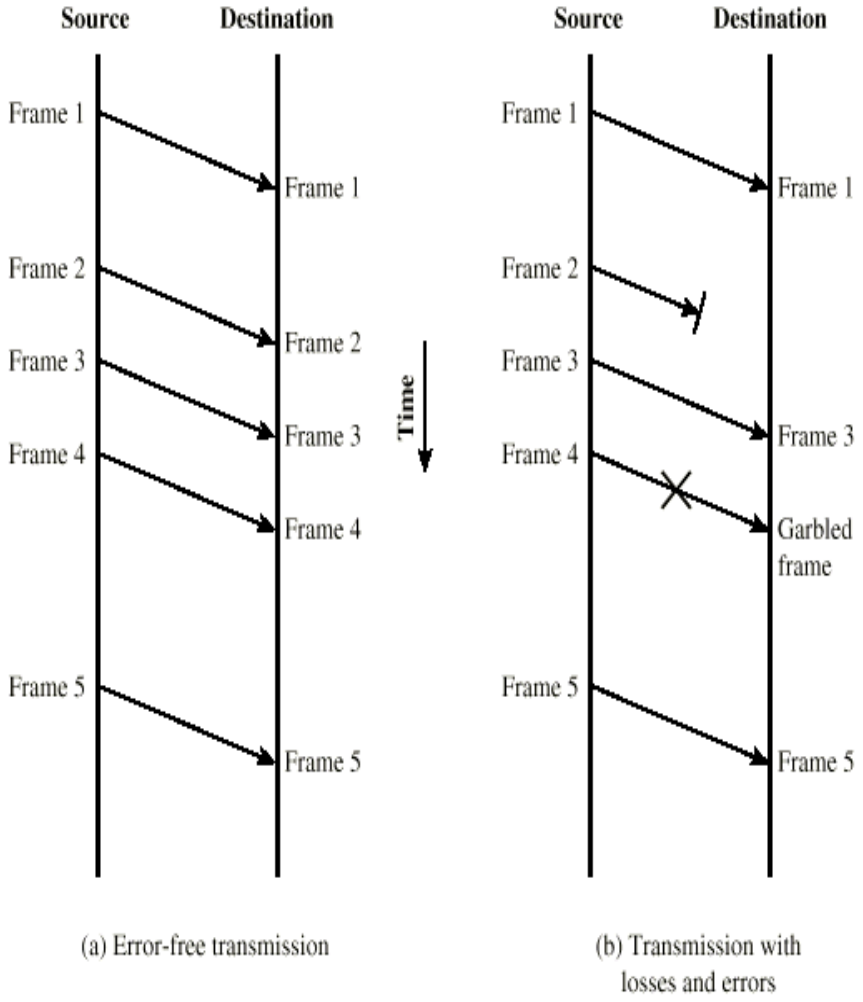
controlo de fluxo



- Técnica para assegurar que a estação que transmite não sobrecarrega a que recebe, evitando perda de tramas.
- Em geral, a existência de *buffers* na estação de recepção, reduz mas não elimina a necessidade de controlar o fluxo.
- A perda de tramas pode ocorrer, também, na(s) rede(s) de interligação das estações quando estas se encontram congestionadas nalgum ponto do percurso entre a estação que transmite e a que recebe.
- **Técnicas mais comuns de controlo de fluxo**:
  - *stop-and-wait*
  - *sliding window* (janela deslizante)

Controlo da ligação de dados

modelo de transmissão de tramas

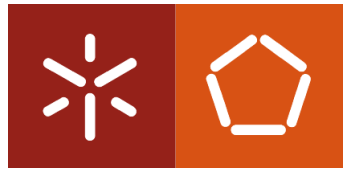


(a) Error-free transmission

(b) Transmission with losses and errors

Controlo da ligação de dados

controlo de fluxo



- ***Stop-and-Wait***
  - Após a transmissão de uma trama, a fonte aguarda confirmação da sua recepção (ACK) antes de transmitir a trama seguinte.
  - A recepção pode parar o fluxo de dados suspendendo temporariamente as confirmações.
- Esta técnica funciona bem quando uma mensagem é fragmentada em poucas tramas de grande dimensão.
- Contudo, se o tamanho das tramas é grande...
  - é maior a probabilidade de erro na trama,
  - é maior a ocupação de recursos (buffers, processadores),
  - o desempenho da ligação tende a piorar, principalmente nas ligações multiponto

Controlo da ligação de dados

controlo de fluxo

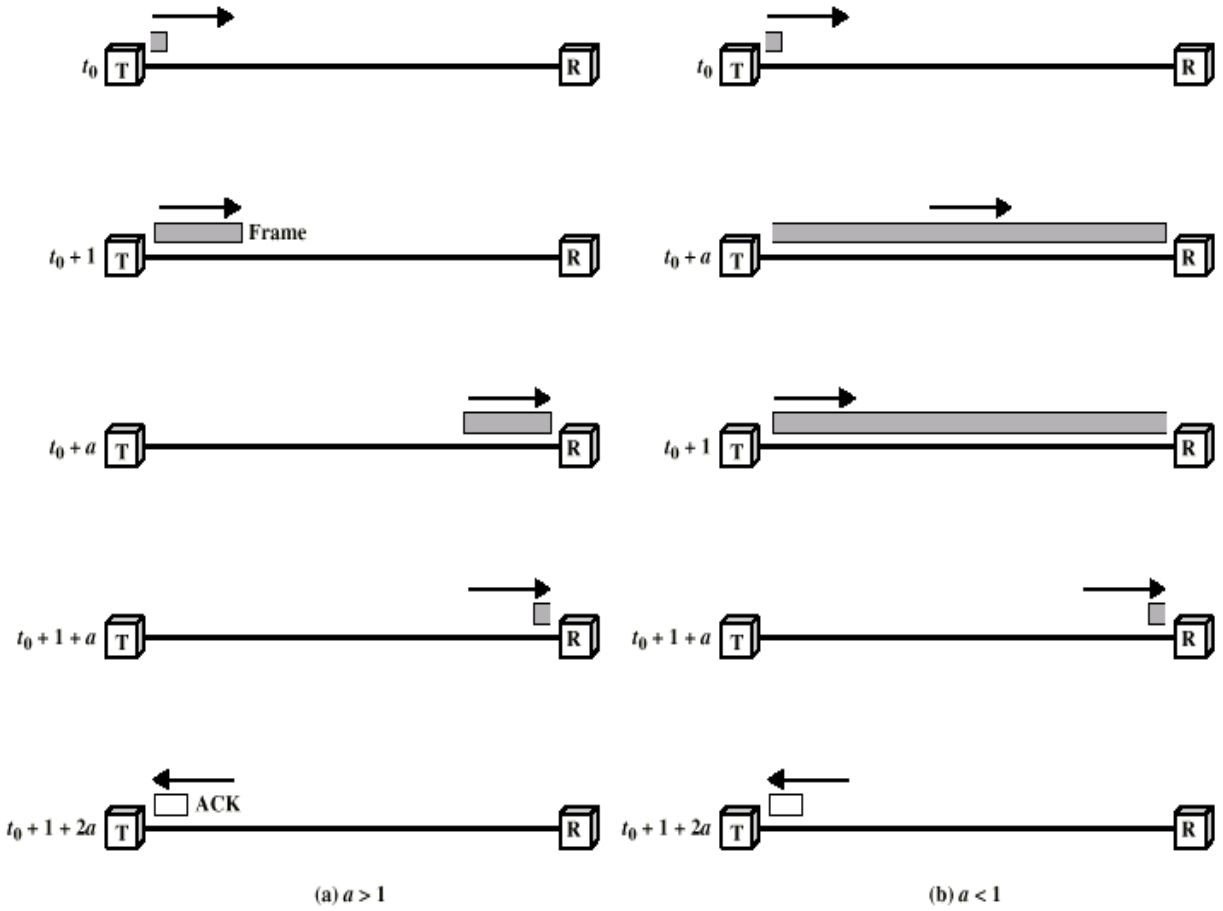
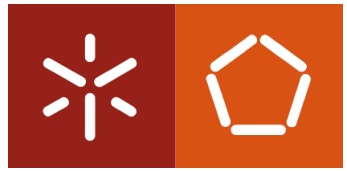


Figure 7.2 Stop-and-Wait Link Utilization (transmission time = 1; propagation time = a)

[DCC,Stallings99]

Controlo da ligação de dados

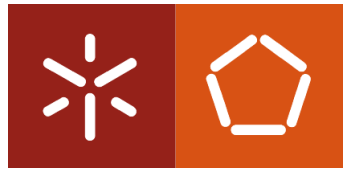
controlo de fluxo



- ***Stop-and-Wait***
  - Tempo de transmissão: tempo que o transmissor demora a emitir todos os bits para o meio de transmissão
  - Tempo de propagação: tempo necessário à propagação de um bit desde o emissor até atingir o receptor
  - **a = tempo de propagação / tempo de transmissão**
  - Quando o tempo de propagação é maior que o tempo de transmissão, o emissor completa a transmissão da trama antes de o receptor ter começado a recebê-la (taxas de transmissão altas ou elevadas distâncias entre o emissor e o receptor). Nestes casos, a técnica de *stop-and-wait* conduz a uma utilização baixa do meio de transmissão e consequentemente uma transmissão pouco eficiente.

Controlo da ligação de dados

controlo de fluxo

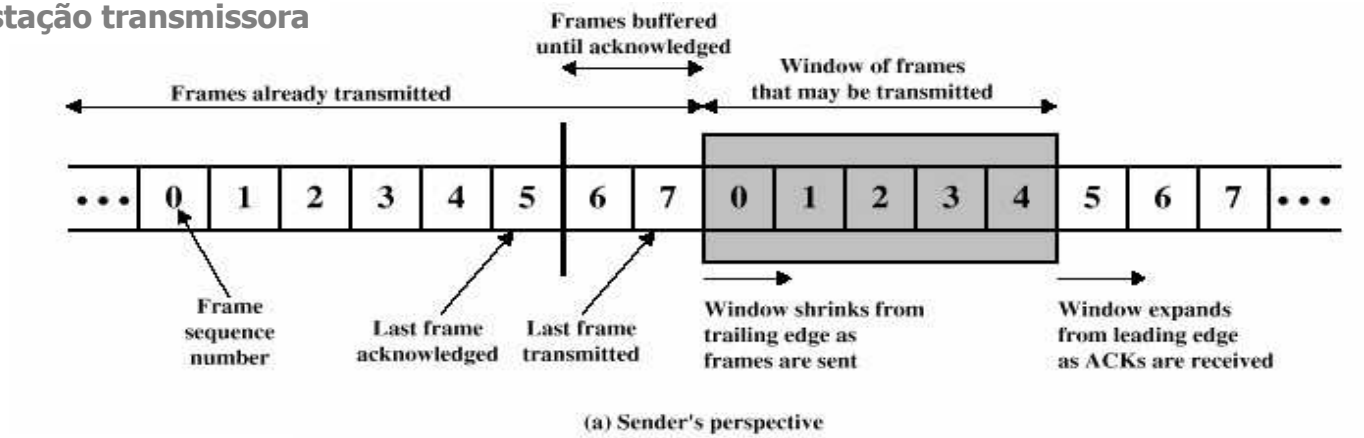


- ***Sliding-Window***
  - permite que existam múltiplas tramas de dados em trânsito
  - o transmissor pode enviar até **W** tramas de dados sem que receba qualquer confirmação da sua recepção
  - obriga o uso de sequenciação (*n* bits, numeração módulo 2^n)
  - cada confirmação positiva indica a próxima trama esperada
  - pode haver confirmação simultânea de múltiplas tramas
  - existem mecanismos distintos para transmitir e receber
  - alguns protocolos de nível 2 suportam *Receive not Ready* e *Piggybacking*
- W **é designado** abertura da janela **($W_{max}=2^n-1$)**

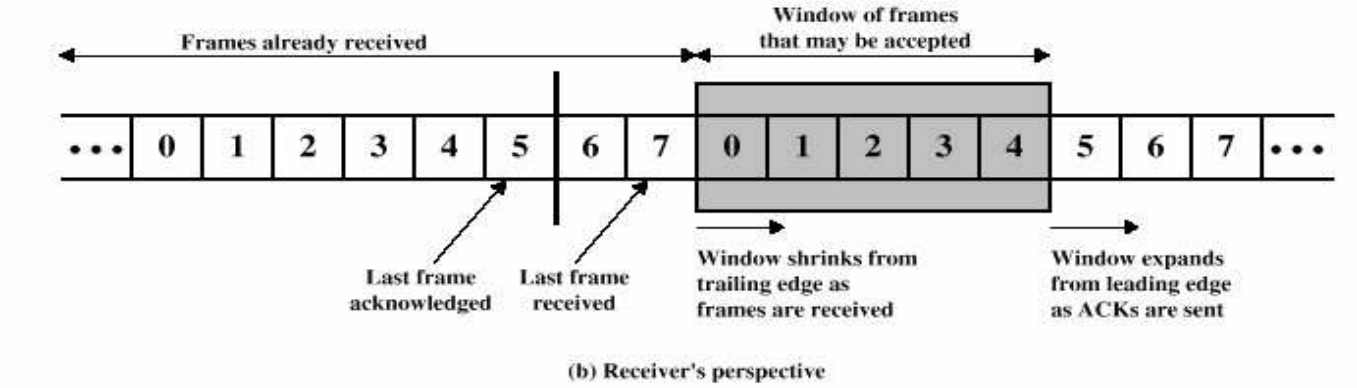
Controlo da ligação de dados

controlo de fluxo: janela deslizante, funcionamento

a) Na perspectiva da estação transmissora



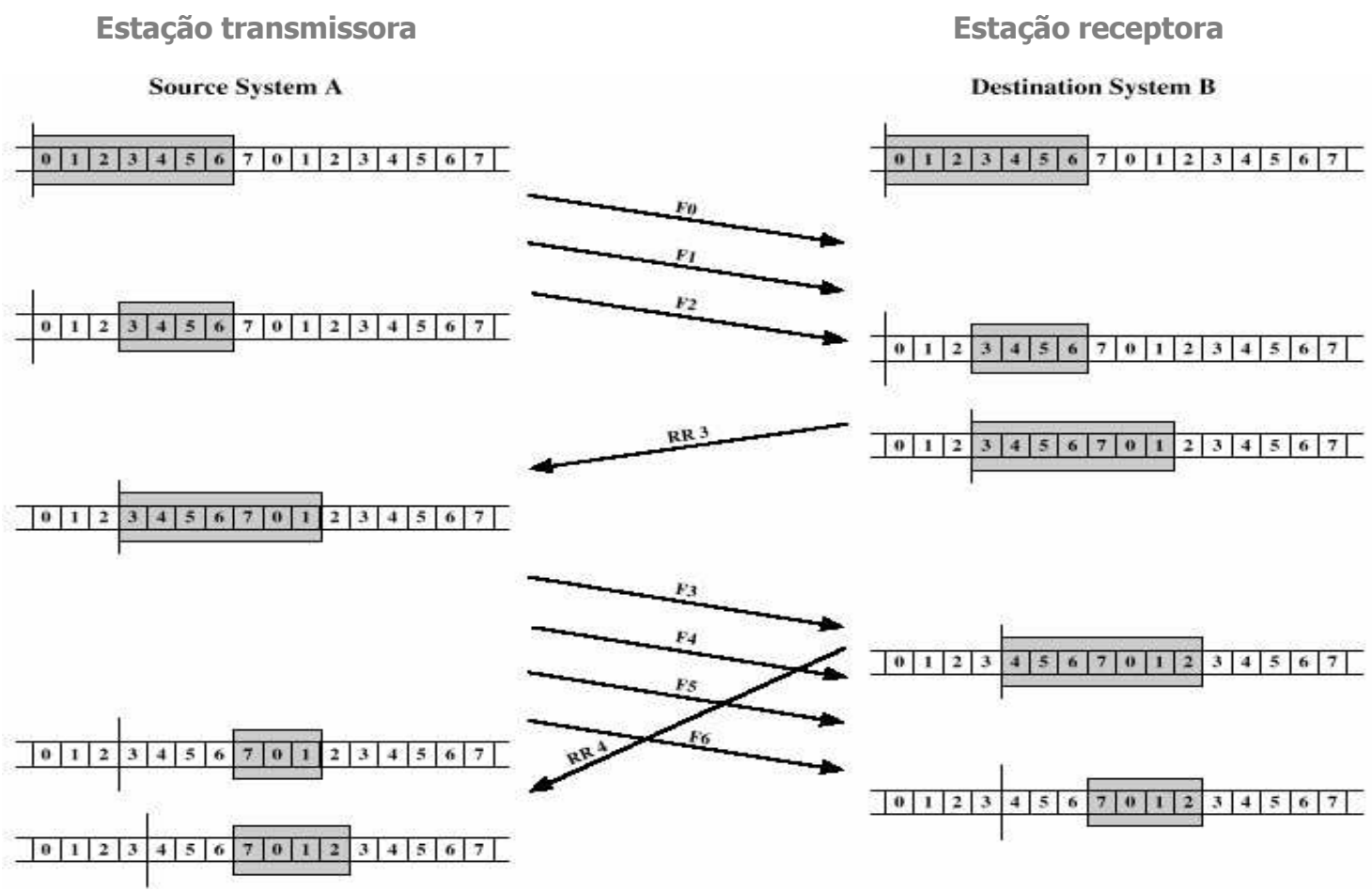
b) Na perspectiva da estação receptora



Janela deslizante com n=3 e W=7 **[DCC,Stallings99]**

Controlo da ligação de dados

controlo de fluxo

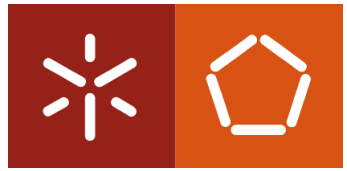


Janela deslizante com n=3 e W=7

[DCC,Stallings99]

Controlo da ligação de dados

controlo de fluxo - utilização da ligação



- A utilização ou rendimento da ligação depende de W e do parâmetro a
- O parâmetro a é a razão entre o tempo de propagação e o tempo de transmissão

$$\mathbf{a = t_{prop} / t_{trama}}$$

$$\mathbf{a = (d/v) \ / (L/r)}$$

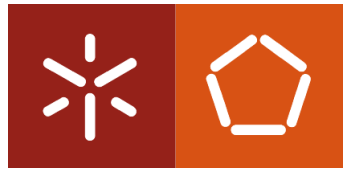
$$\mathbf{a = rd / vL}$$

d - distância (m); v - velocidade de propagação (m/s);

L - comprimento trama (bits); r - rítmo de transmissão (bps)

Controlo de ligação de dados

controlo de fluxo - parâmetro **a**



- **Ligações satélite**

  Valores típicos:

  t_{prop} = 270 ms

  L = 4000 bits; r = 56 Kbps => t_{trama} = 71 ms

  a = 3.8

  Para 125μs < t_{trama} < 6ms os valores de *a* são *a*>>1

- **Redes locais**

  Valores típicos:

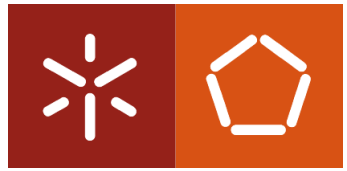
  0.1 < d < 10 Km; V = $2x10^8$ m/s

  L = 500 bits; 0.1 < r < 10 Mbps

  Neste caso o parâmetro *a* tem um valor pequeno, *a*<1

Controlo da ligação de dados

controlo de fluxo - utilização da ligação



- ***Stop-and-Wait***
  - Supondo que uma mensagem é enviada numa sequência de *frames* f1,f2,...,fn, então o tempo total para enviar a mensagem pode ser expresso como $t_{total} = n * t_{frame}$ onde t_{frame} é o tempo necessário para enviar uma frame e receber um ack.

$$t_{frame} = t_{transFrame} + t_{prop} + t_{proc} + t_{transAck} + t_{prop} + t_{proc}$$

Assumindo que $t_{proc} \approx 0$ e $tt_{ransAck} \approx 0$

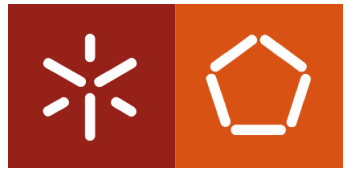
$$t_{total} = n * (t_{transFrame} + 2*t_{prop})$$

  - A **Utilização** da ligação é a fração do tempo total que é útil, ié, que é utilizado a transferir tramas de dados, $U = t_{util} / t_{total}$:

$$U = n * t_{transFrame} / n * (t_{transFrame} + 2*t_{prop}) = 1 / (1 + 2a)$$

Controlo da ligação de dados

controlo de fluxo - utilização da ligação



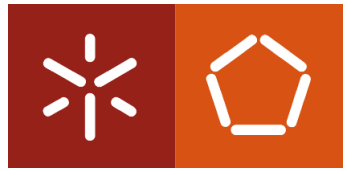
- Exemplo: Considere uma rede de longa distância ATM com duas estações distanciadas 1000 km uma da outra. O tamanho standard de uma frame ATM é 424 bits e a taxa de transmissão standard é 155,52 Mbps. O tempo de transmissão é igual a $424/(155{,}52 \times 10^6) = 2{,}7 \times 10^6$ seg. Se assumirmos que o meio de transmissão é uma fibra óptica e a velocidade de propagação igual a 2/3 velocidade da luz (2×10^8m/seg), temos que o tempo de propagação é igual a $10^6 / (2 \times 10^8) = 0{,}5 \times 10^{-2}$.
- Então $a = 0{,}5 \times 10^{-2} / 2{,}7 \times 10^6 = 1850$, e

$$U = 1/(1 + 2a)$$

$$U = 1 / (1+2*1850) = 0{,}00027 = 0{,}027\%$$

Controlo de ligação de dados

controlo de fluxo - utilização da ligação



- ***Sliding Window (Janela Deslizante)***

  Exemplo: ligação *full-duplex* entre duas estações A e B

  - **Caso 1** - A estação A transmite continuamente. A confirmação de chegada da trama 1 ocorre **antes** da janela se fechar, então

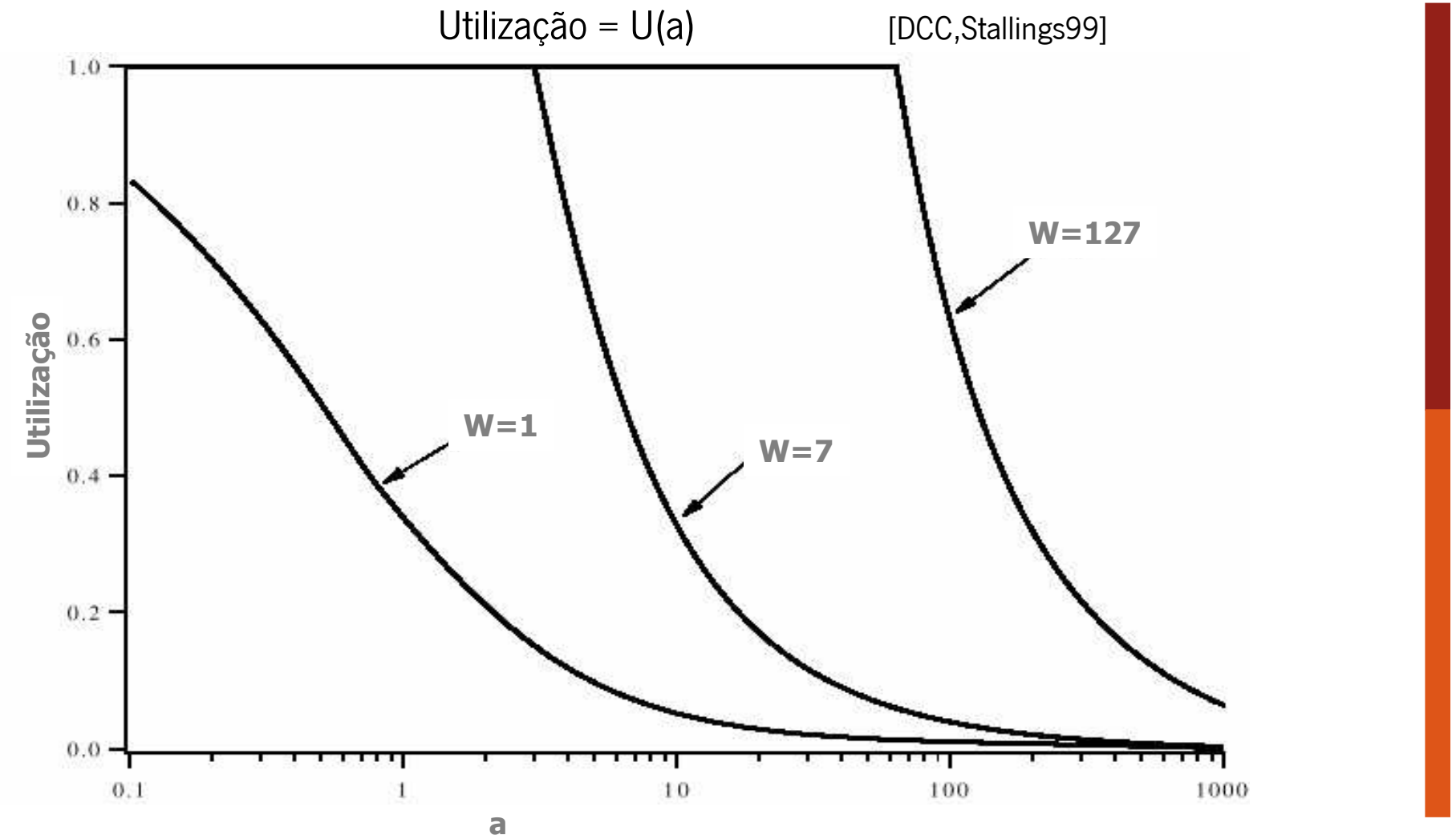
$$\mathbf{U = 1 \quad se \quad W > 2a + 1}$$

  - **Caso 2** - A estação A tem a janela fechada em $t_0 + W$ e não pode enviar tramas até $t_0 + 2a + 1$ (chegada do primeiro ACK), então

$$U = W / (2a + 1) \quad \text{se} \quad W < 2a + 1$$

Controlo de ligação de dados

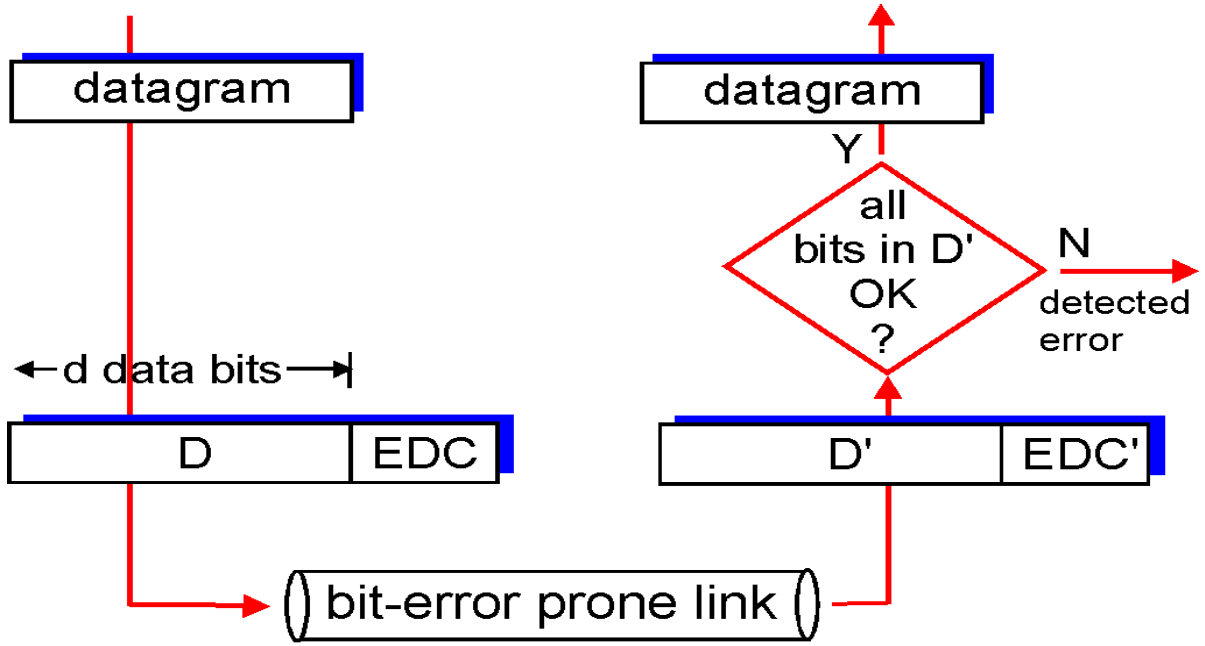
controlo de fluxo - utilização da ligação



Controlo de ligação de dados

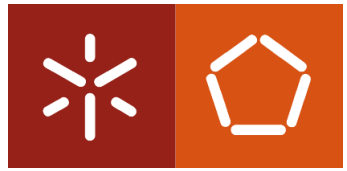
detecção de erros

- A cada trama (D: Datagram), o transmissor adiciona um número de bits (**EDC: Error Detection and Correction bits**) que será usado pelo receptor para detecção de erros.



Controlo de ligação de dados

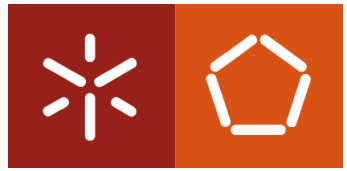
detecção de erros



- A detecção de erros **não é 100% fiável**: o protocolo pode não detectar erros!!
- **Probabilidade de erro residual** - probabilidade de existirem erros em número superior aos que é possível detectar pelo mecanismo utilizado para o efeito.
- **Quanto maior for o número de bits usados no campo EDC, maior é a probabilidade de sucesso na detecção e correcção de erros**

Controlo de ligação de dados

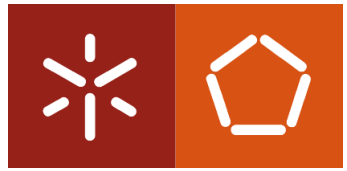
detecção de erros – técnicas



- **Em caso de erro, o receptor corrige o erro ou notifica o transmissor**
- **Técnicas:**
  - Utilização de bit(s) de paridade (paridade vertical e horizontal)
  - Soma de verificação (*Checksum*)
  - Verificação de redundância cíclica (CRC)

Controlo de ligação de dados

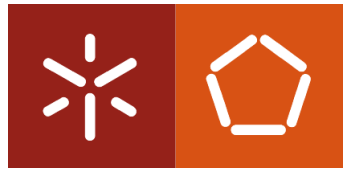
detecção de erros



- **Bit de paridade**
  - processo simples que reduz a probabilidade de aceitação de tramas erradas
  - a taxas de transmissão elevadas podem ocorrer erros em bits consecutivos (erros em rajada)
  - duas variantes: um único bit de paridade, bit de paridade em duas dimensões

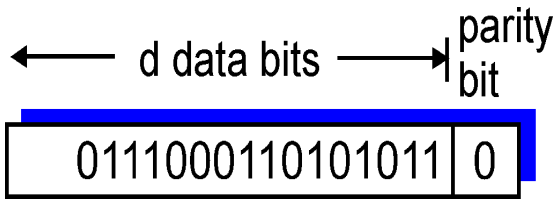
Controlo de ligação de dados

detecção de erros: utilização de bits de paridade



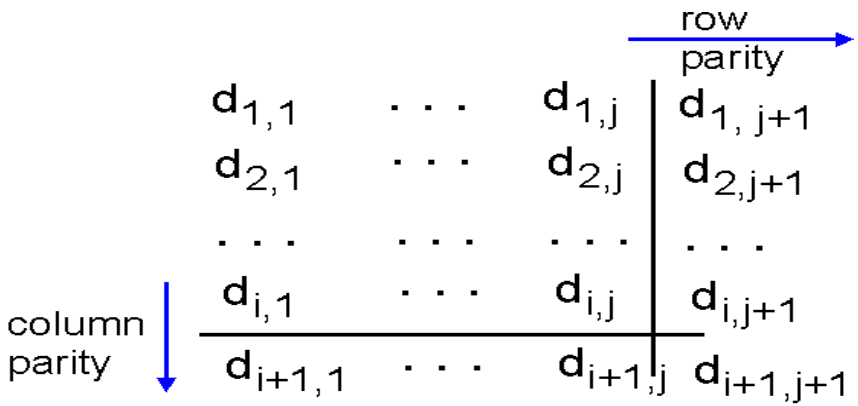
Um único bit de paridade

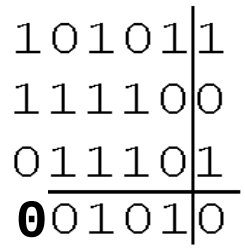
Detecta erros num único bit



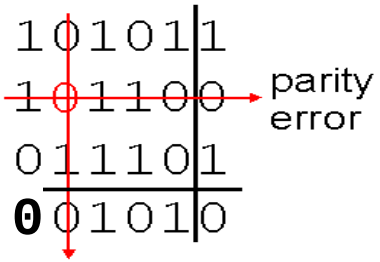
Bits de paridade a duas dimensões

Detecta e corrige erros que ocorram num bit





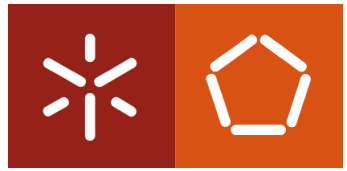
no errors



correctable single bit error

Controlo de ligação de dados

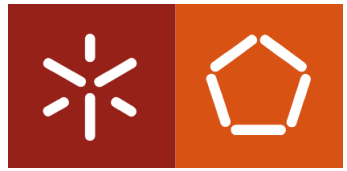
detecção de erros: utilização de bits de paridade



- **1 bit de paridade por caracter**
  - Usado na transmissão série assíncrona
  - Detecta erros num bit, ou qualquer número impar de erros
  - Se os erros ocorrerem em rajada, a probabilidade de erros não detectados pode chegar aos 50%
- **Paridade a duas dimensões**
  - É apenas uma generalização da paridade simples
  - Detecta e corrige erros de um bit (mesmo que esse erro ocorra nos bits de paridade); detecta sem corrigir qualquer combinação de dois erros numa trama;
  - A capacidade de detectar e corrigir erros designa-se FEC (*Forwarding Error Correction*), sendo estas técnicas usadas, por exemplo, no armazenamento e reprodução de áudio digital.

Controlo de ligação de dados

detecção de erros: soma de verificação (checksum)



Checksum: apenas usado no nível de transporte, por exemplo pelo TCP e pelo UDP (16 bits)

Transmissor

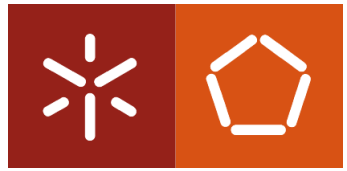
- Encara cada conjunto de dados a enviar como uma sequência de grupos de *k* bits
- Determina o *checksum*: adiciona os grupos de k bits. O complemento para 1 da soma constituí o *checksum*
- O transmissor insere o *checksum* no conjunto de dados a enviar

Receptor:

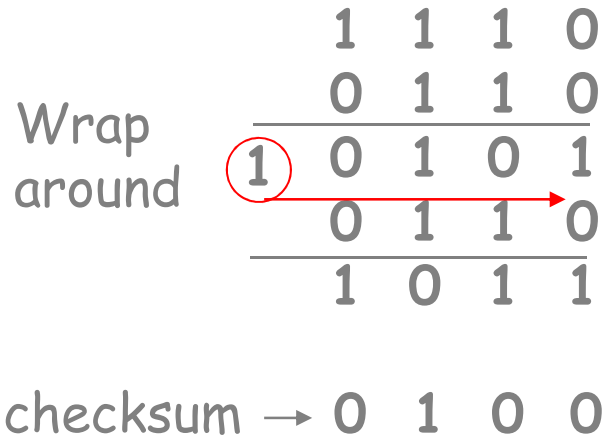
- Encara cada conjunto de dados recebidos como uma sequência de grupos de *k* bits
- Determina o *checksum* (com o *checksum* inserido pelo transmissor incluído)
- O resultado deverá ser igual a zero, senão foi detectado um erro

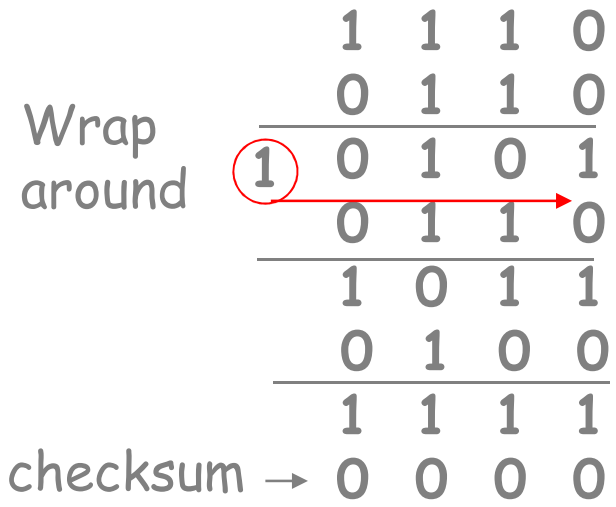
Controlo de ligação de dados

detecção de erros: soma de verificação (checksum)



Exemplo: k=4, mensagem = 111001100110

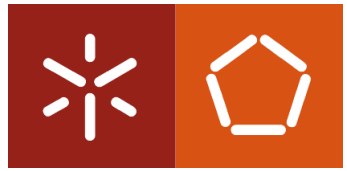




Mensagem a enviar: 1 1 1 0 0 1 1 0 0 1 1 0 0 1 0 0

Controlo de ligação de dados

detecção de erros: verificação de redundância cíclica (CRC)



- **Cyclic Redundancy Check**

  Dada uma mensagem inicial M de *k* bits, o transmissor gera uma sequência R de *n-k* bits (CRC ou FCS *Frame Check Sequence)* tal que, os *n* bits da trama resultante sejam divisíveis por um número pré-determinado G.

$$T_t = 2^{n-k} M + R$$

  T_t - trama total a ser transmitida (*n* bits)

  M - mensagem de *k* bits (parte mais significativa de T)

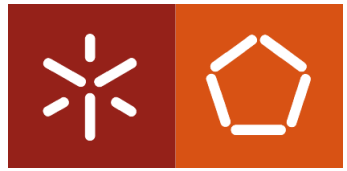
  R - FCS (parte menor significativa de T) de *n-k* bits

  G - divisor de *n-k+1* bits $[2^{n-k} M = Q G + R]$

  (usando aritmética módulo 2)

Controlo de ligação de dados

detecção de erros: verificação de redundância cíclica (CRC)

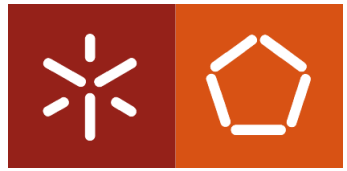


- **Processo:**
  - na transmissão
    - dividir 2^{n-k} M por G
    - usar o resto R como FCS
  - na recepção
    - dividir a trama recebida T_r por G
    - se R = 0 decidir que não há erro, ié, $T_r = T_t$

- Pode falhar se o número de erros for superior à capacidade de detecção do código
- Consegue detectar todos os **n-k** erros consecutivos, e mesmo rajadas com mais de **n-k** erros embora com probabilidade $\mathbf{1 - 0.5^{(n-k)}}$

Controlo de ligação de dados

detecção de erros: verificação de redundância cíclica (CRC)

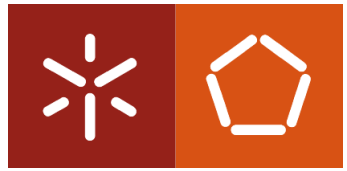


Verificação

- **Pretende-se dividir 2^{n-k} M por G e usar o resto R**
  - $T_t = 2^{n-k} M + R$
- **Será que T_t é divisível por G ?**
  - $T_t/G = 2^{n-k} M/G + R/G$
  - como $2^{n-k} M/G = Q + R/G$,
  - $T_t/G = Q + (R+R)/G$ [sendo R+R=0, na aritmética módulo 2]
  - $T_t/G = Q$
- **T_t/G não tem resto, logo T_t é divisível por G**

Controlo de ligação de dados

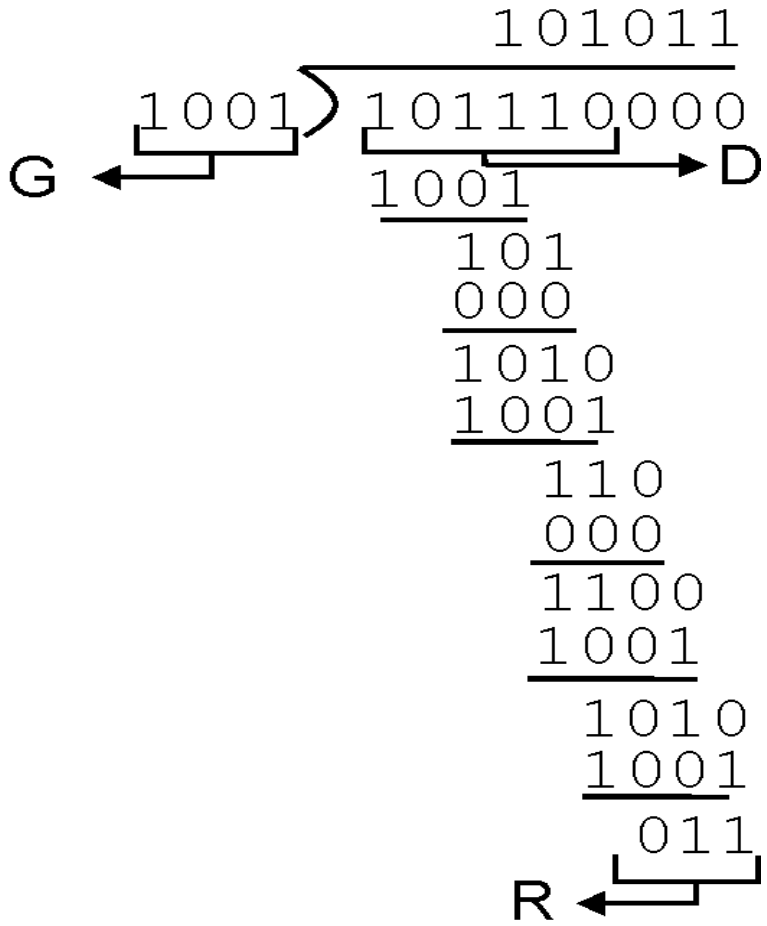
detecção de erros: verificação de redundância cíclica (CRC)



Exemplo no emissor:

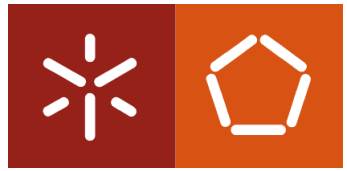
Mensagem a enviar: 101110

Mensagem enviada: 101110011



Controlo de ligação de dados

detecção de erros: verificação de redundância cíclica (CRC)



- **O processo CRC é, em geral, expresso através de polinómios de uma variável, com coeficientes binários.**

  Exemplo de um polinómio gerador G(*x*):

  CRC-32: $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$

  100000100110000010001110110110111 (33bits, para dar resto 32)

  normalizado para transmissão síncrona ponto-a-ponto (IEEE 802.x)

  Outros:

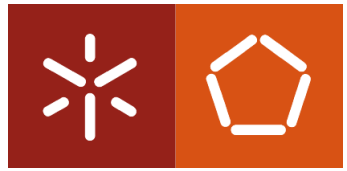
  CRC-12 = $x^{12} + x^{11} + x^3 + x^2 + x + 1$ (1100000001111)

  CRC-16 = $x^{16} + x^{15} + x^2 + 1$ (1100000000000101)

  CRC-CCITT = $x^{16} + x^{12} + x^5 + 1$ (1000100000100001)

Controlo de ligação de dados

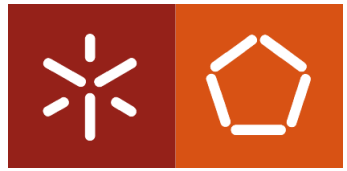
controlo de erros



- **Na comunicação de dados a técnica mais usada no controlo de erros é o *Automatic Repeat Request* (ARQ)**
  - o receptor não tenta corrigir os erros
  - o código de controle de erros é usado no receptor apenas como detector erros
  - detectados erros, o receptor descarta a trama e pode pedir a retransmissão da unidade de dados
  - probabilidades de erro aceitáveis podem ser obtidas com polinómios de menor grau
- **Alternativa: *Forward Error Correction* (FEC)**

Controlo de ligação de dados

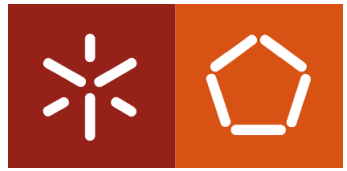
controlo de erros



- **Envolve a detecção de falhas nas tramas trocadas de modo a tornar a ligação de dados fiável.**
- **Tipos de falhas: trama perdida ou trama errada**
- **O ARQ envolve:**
  - detecção de erros na trama recebida através do CRC
  - confirmação positiva: para tramas recebidas sem erros
  - confirmação negativa e retransmissão: para tramas onde é detectado erro
  - retransmissão por limite de tempo - se não é recebida confirmação de trama, dentro do período de tempo *t*

Controlo de ligação de dados

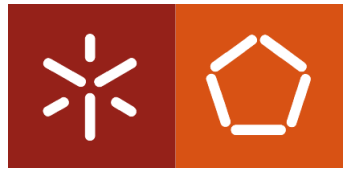
controlo de erros



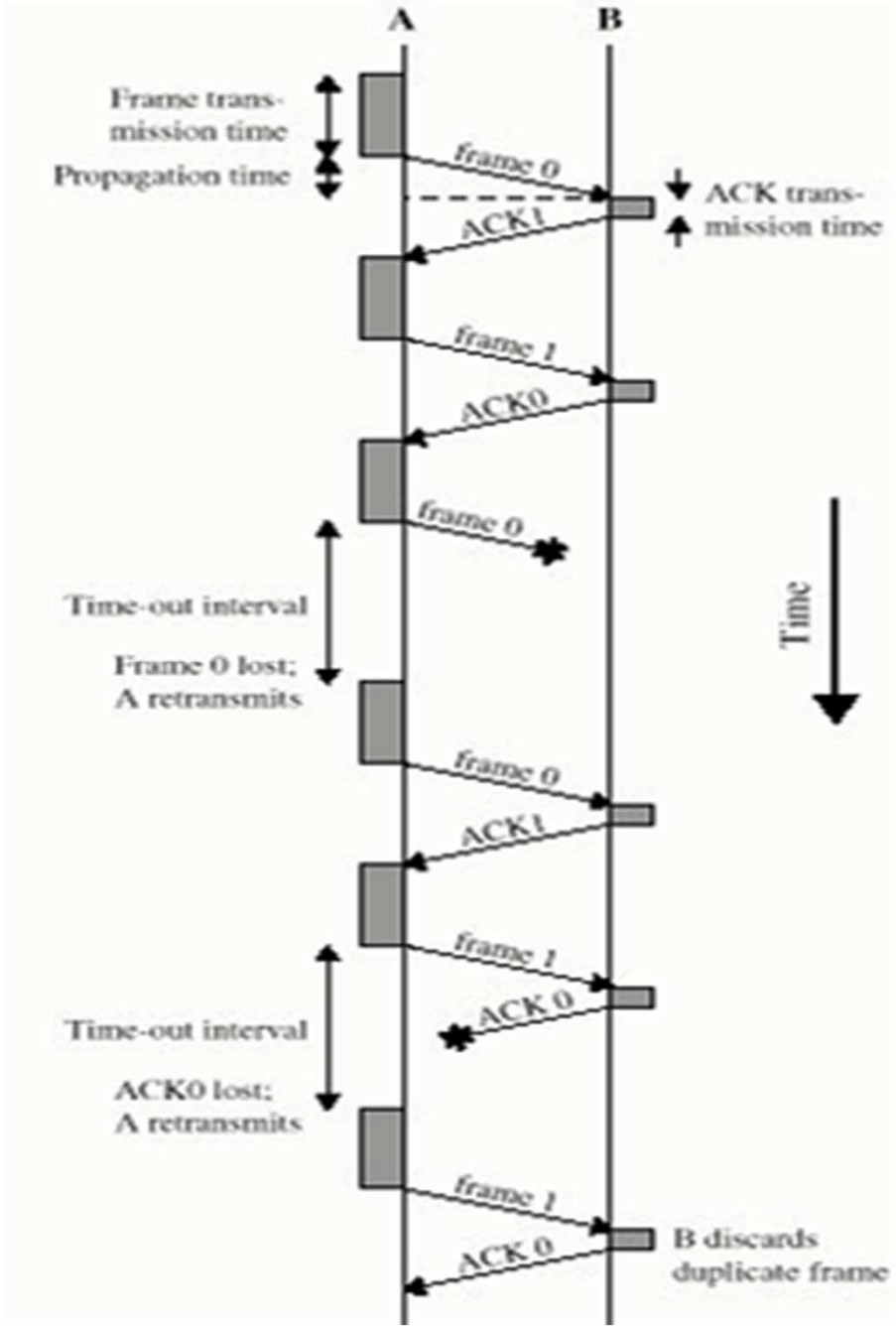
- **Métodos ARQ:**
  - ***Stop-and-wait*** *(Pára-e-espera)*
  - ***Go-back-N*** *(volta-atrás-N)*
  - ***Selective Reject*** *(rejeição selectiva)*

Controlo de ligação de dados

controlo de erros

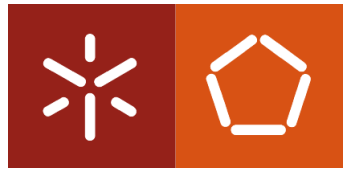


- ***stop-and-wait***
  - semelhante à técnica de controlo de fluxo *stop-and-wait*
  - transmissor:
    - activa temporizador e mantém cópia da trama até obter ACK
    - no máximo espera *timeout* até transmitir de novo
  - receptor:
    - envia ACK, NAK (pedido explícito) ou *no reply* (pedido implícito)
  - sequenciação necessária para resolver a situação de erro na trama de confirmação (duplicação da trama)
  - vantagem: simples; desvantagem: reduzida eficiência


A
B
Frame trans-
mission time
Propagation time
frame 0
ACK trans-
mission time
ACK1
frame 1
ACK0
frame 0
Time-out interval
Frame 0 lost;
A retransmits
Time
frame 0
ACK1
frame 1
ACK 0
Time-out interval
ACK0 lost;
A retransmits
frame 1
B discards
duplicate frame
ACK 0

Controlo de ligação de dados

controlo de erros



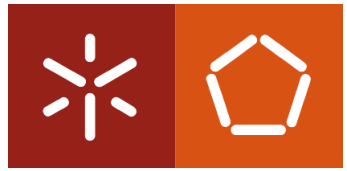
- ***volta-atrás-N***
  - a falta de sequenciação ou erro na recepção implica a retransmissão a partir de uma determinada ordem.

Exemplos:

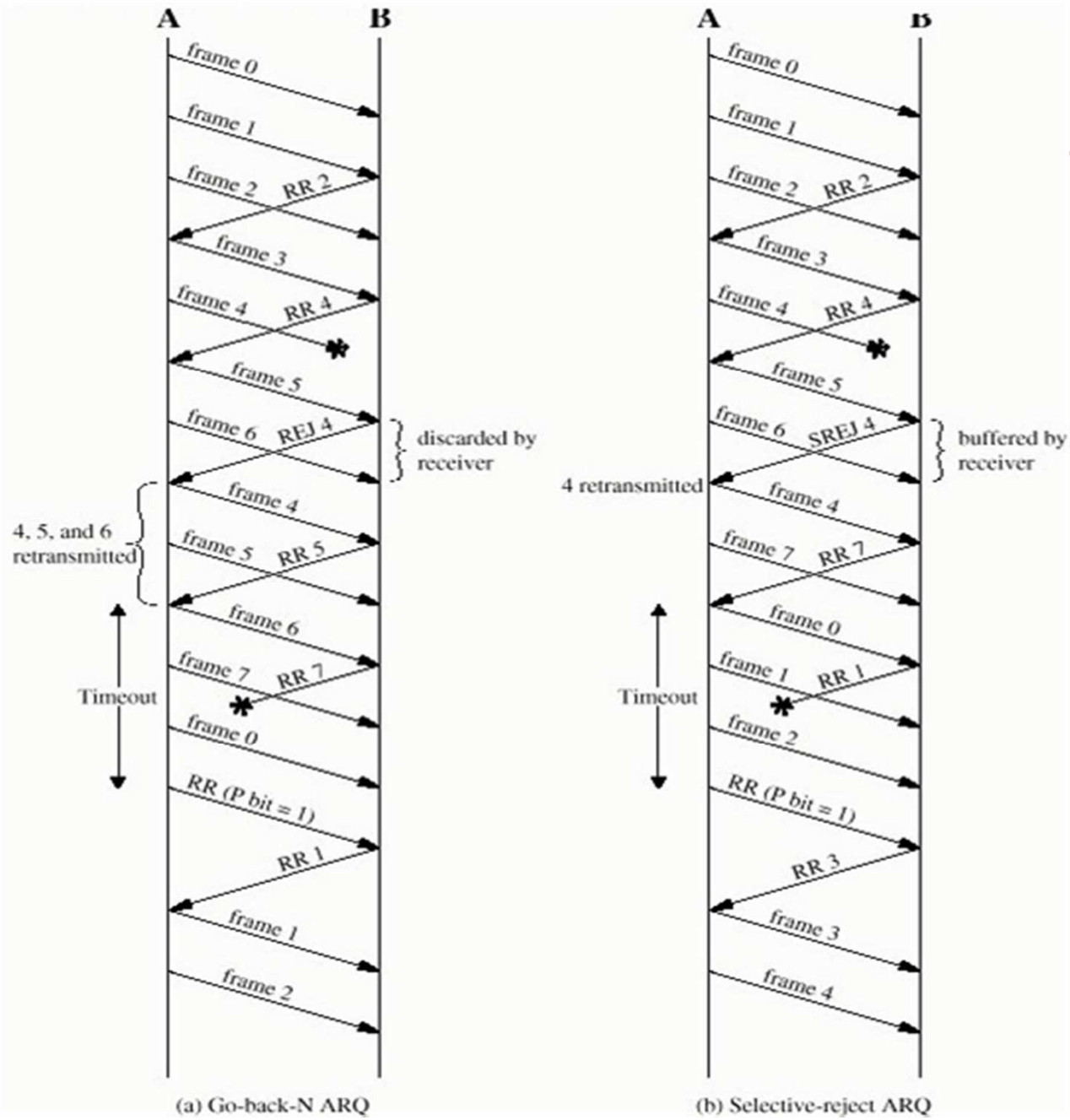
- trama t_i corrompida ou perdida
  - B recebeu $t_{(i-1)}$ e detecta erro em t_i; B envia NAK i
  - t_i é perdida, B recebe $t_{(i+1)}$ e envia NAK i
  - t_i é perdida; A retransmite t_i por *timeout*
- confirmações corrompidas
  - confirmação por ACK seguintes
  - se A expira, A retransmite t_i e todas as tramas subsequentes

Controlo de ligação de dados

controlo de erros

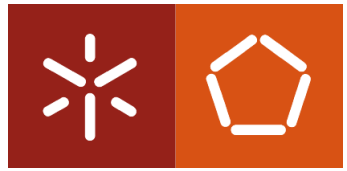


- ***rejeição selectiva***
  - apenas são retransmitidas as tramas que recebem confirmação negativa explícita ou se ocorre *timeout*.
  - obriga a confirmações positivas por ordem
  - W_{max} mais restritivo para não sobrepor as janelas na transmissão e na recepção ($W_{max}=2^{n-1}$ e não $W_{max}=2^{n}-1$)
  - vantagem: menos retransmissões, melhor utilização da ligação
  - desvantagem: requer mais processamento (e controlo) na transmissão e na recepção


A
B
frame 0
frame 1
frame 2
RR 2
frame 3
frame 4
RR 4
frame 5
frame 6
REJ 4
discarded by receiver
4, 5, and 6 retransmitted
frame 4
frame 5
RR 5
frame 6
frame 7
RR 7
Timeout
frame 0
RR (P bit = 1)
RR 1
frame 1
frame 2
(a) Go-back-N ARQ
A
B
frame 0
frame 1
frame 2
RR 2
frame 3
frame 4
RR 4
frame 5
frame 6
SREJ 4
buffered by receiver
4 retransmitted
frame 4
frame 7
RR 7
frame 0
frame 1
RR 1
Timeout
frame 2
RR (P bit = 1)
RR 3
frame 3
frame 4
(b) Selective-reject ARQ

Controlo de ligação de dados

controlo de erros

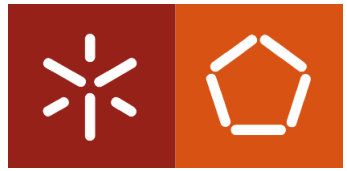


- **No mecanismo de rejeição selectiva <u>a ordem das tramas na recepção não é mantida</u> daí que:**
  - se a ordem das tramas for relevante, o tamanho dos *buffers* pode ser incomportável
  - Transmissor complexo: tem de ser capaz de enviar tramas fora de ordem
  - Receptor complexo: tem de conseguir ordenar tramas
  - em geral, é usado para transmitir tramas "independentes" entre si
  - usado em meios onde a probabilidade de erro é maior (radio links)

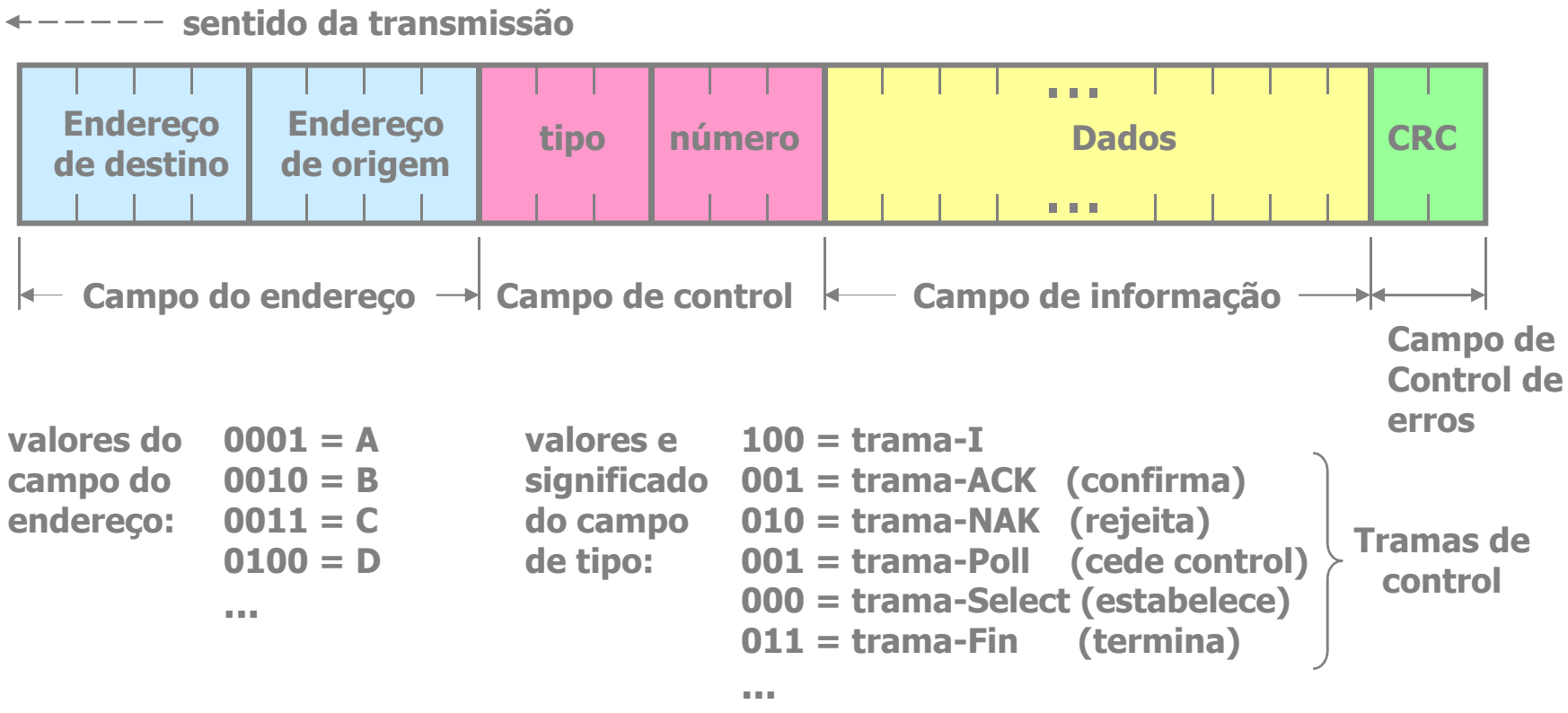
- **O mecanismo volta-atrás-N é mais usado do que o de rejeição selectiva, pois apesar de conduzir a uma pior utilização da ligação, reduz a complexidade do receptor.**

Controlo da ligação de dados

definição da trama: exemplo de um formato e semântica



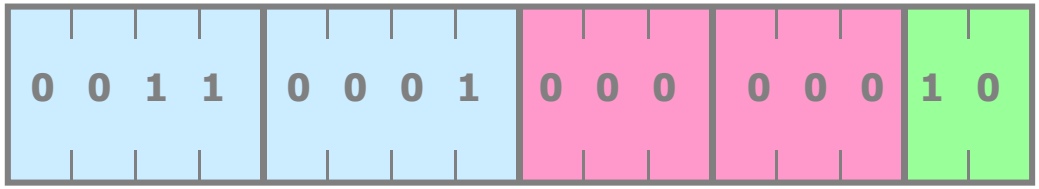
- **Cada protocolo define um formato de PDU, bem como os valores, o significado e o comprimento dos seus campos. Exemplo:**



Controlo da ligação de dados

definição da trama: exemplo de um formato e semântica

- **As tramas de controle não possuem o campo de dados e portanto são tramas curtas.**
- **Exemplo de uma trama-Select:**

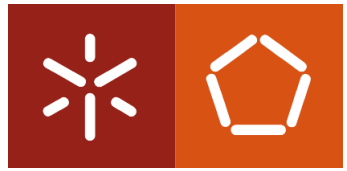


trama-Sel C, A, 0

- **Nesta definição protocolar está-se a pressupor que, nas tramas de resposta (ACK e NAK), o** número **refere-se ao de uma trama de comando do sentido oposto (recebida) que elas estão a confirmar**
- **Nas restantes tramas, o** número **representa a numeração de sequência da própria trama**

Controlo da ligação de dados

protocolos (disciplinas) de linha



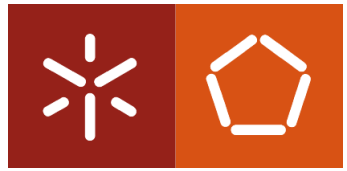
- **Ligações Ponto-a-Ponto (PP)**
  - Em geral são ligações com um canal (circuito ou banda) para transmissão em cada sentido
  - Por usarem canal dedicado (não partilhado), a ligação lógica pode efectuar-se imediatamente porque o canal está naturalmente *adquirido*.
- **Ligações Multiponto (MP)**
  - Em geral são ligações com um único canal de transmissão que é partilhado por várias estações
  - Duas ou mais transmissões simultâneas podem causar interferências
    - **colisão** ocorre quando uma estação recebe dois ou mais sinais ao mesmo tempo
  - A ligação lógica tem de ser precedida pela aquisição do canal através de um *protocolo de acesso ao meio (protocolo MAC)*

Acesso múltiplo partilhado

- **Algoritmo distribuído que determina como é que as diferentes estações acedem ao canal, ou seja, determina quando é que uma estação pode transmitir**
- **Utiliza o próprio canal partilhado para fazer essa coordenação**

Controlo da ligação de dados

protocolo de acesso ao meio - MAC (Medium Access Control)

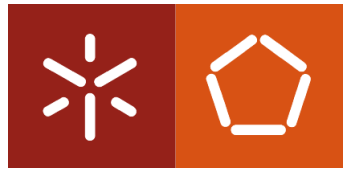


<u>Protocolo Ideal</u> para um canal com uma taxa de transmissão de R bps:

1. Quando apenas uma estação pretende transmitir pode fazê-lo a uma taxa **R**
2. Quando M estações pretendem transmitir cada uma dela transmite a uma taxa **R/M**
3. Completamente **distribuída**
   - Não existe uma única estação responsável por coordenar as transmissões
   - Sem necessidade de sincronizações, *clocks*, etc.
4. **Simples**

Controlo da ligação de dados

protocolo de acesso ao meio - MAC (Medium Access Control)



Taxonomia:

- **Partição do Canal**

  O canal é dividido em "peças" mais pequenas (por tempo ou frequências) e cada uma das "peças" é alocada a um nó para uso exclusivo

- **Acesso Aleatório**

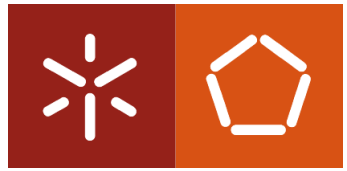
  O canal não é dividido, permite colisões e recupera das colisões

- **Acesso sequencial**

  O nós esperam pela sua vez para transmitir, mas os recursos são usados na mediada das necessidades de cada um (com limites).

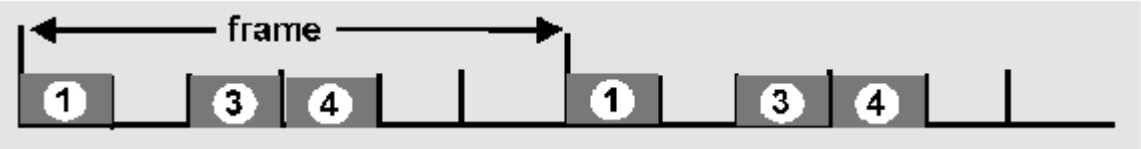
Protocolos de Acesso ao Meio

Partição do canal



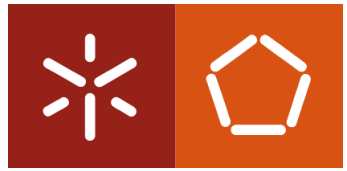
TDMA: time division multiple access

- O acesso ao canal é implementado “em voltas”
- Cada estação recebe um *slot* de tamanho fixo em cada volta (o tamanho do *slot* deve dar para transmitir um pacote)
- Os *slots* que não são usados permanecem desocupados exemplo: 6 estações numa LAN, as estações 1, 3 e 4 tem pacotes para transmitir, a 2,5 e 6 estão em ”silêncio”



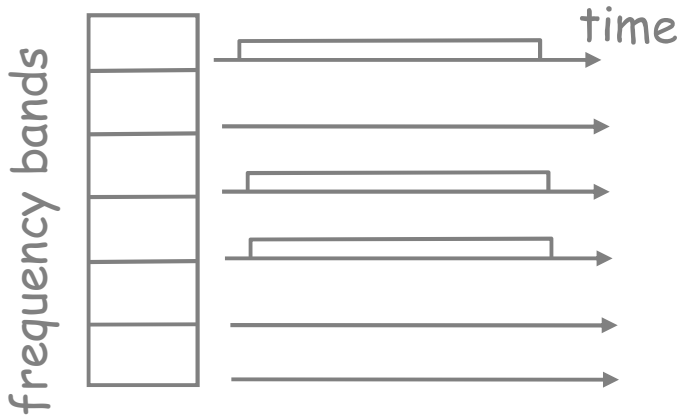
Protocolos de Acesso ao Meio

Partição do canal



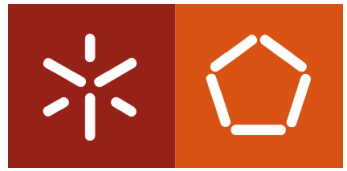
FDMA: frequency division multiple access

- O espectro do canal é dividido em bandas de frequência
- A cada estação é atribuída uma banda de frequência fixa
- O tempo de transmissão que não é utilizado nas bandas de frequência é desperdiçado
- exemplo: 6 estações numa LAN, as estações 1, 3 e 4 tem pacotes para transmitir, a 2,5 e 6 estão em "silêncio"



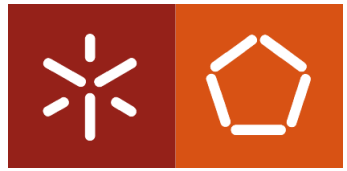
Protocolos de Acesso ao Meio

Acesso aleatório



- **Quando uma estação tem um pacote para enviar**
  - Envia-o usando toda a capacidade do canal (R)
  - Não existe coordenação entre as estações
- **Duas ou mais estações a transmitirem simultaneamente dá "colisão"**
- **Os protocolos MAC de acesso aleatório especificam:**
  - Como detectar colisões
  - Como recuperar de colisões (por exemplo através de retransmissões com atraso aleatório)
- **Exemplos de protocolos MAC de <u>Acesso Aleatório</u>:**
  - slotted ALOHA
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA



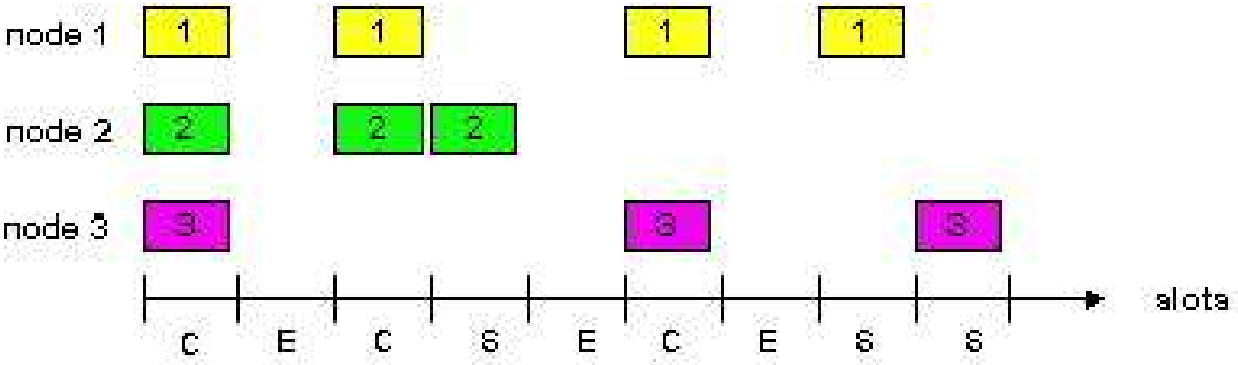
Assumindo que:

- Todas as tramas têm o mesmo tamanho
- O tempo é dividido em *slots* de igual tamanho, em cada *slot* cabe uma trama
- As estações começam a transmitir apenas no inicio dos *slots*
- As estações estão sincronizadas
- Se duas ou mais estações começarem a transmitir num *slot*, todos as estações detectam a colisão

Operação

- Quando uma estação tem uma trama para transmitir espera pelo início do próximo *slot* e transmiti-a
- Se não ocorrer nenhuma colisão a estação pode enviar a próxima trama no *slot* seguinte
- Se ocorrer uma colisão a estação retransmite a trama num dos próximos *slots* com uma probabilidade *p*

Slotted ALOHA



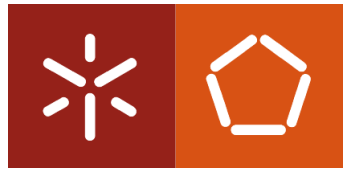
Pros

- Uma estação pode transmitir continuamente utilizando toda a capacidade do canal
- Distribuída: só os *slots* é que têm que ser sincronizados entre as estações
- Simples

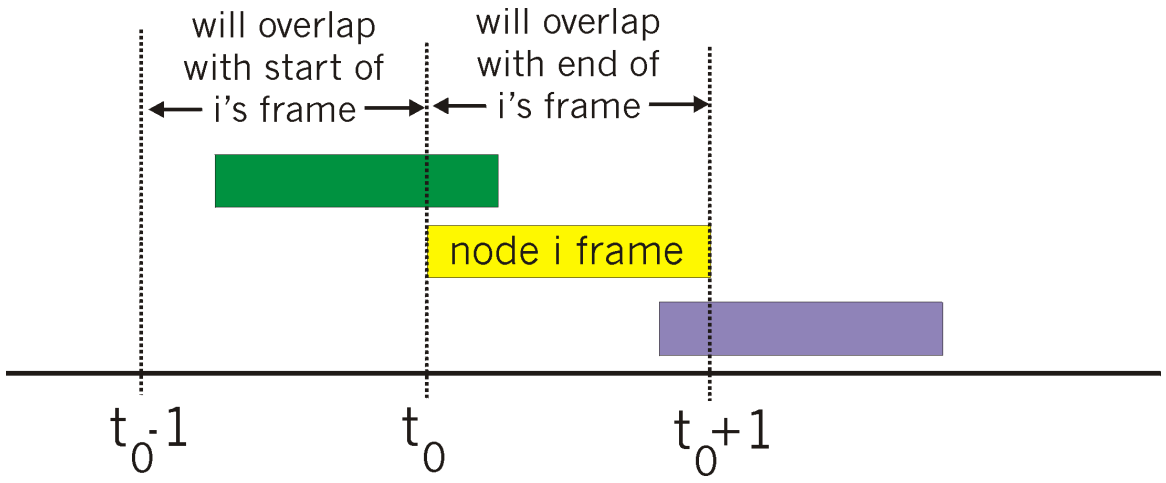
Cons

- As colisões levam a um desperdício *slots*
- *Slots* desocupados
- Os nós podem conseguir detectar uma colisão antes de acabar de transmitir a trama.
- Sincronização de *clocks*

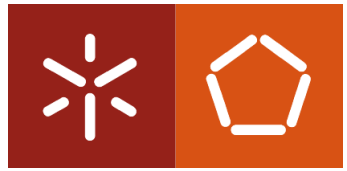
ALOHA Puro (unslotted)



- **<u>Aloha Puro</u>: mais simples e sem necessidade de sincronização**
- **Quando o transmissor tem um trama pronta**
  - transmiti-a imediatamente
- **A probabilidade de ocorrerem colisões aumenta**
  - Uma trama enviada em t_0 colide com outras tramas enviadas em $[t_0-1,t_0+1]$



CSMA (Carrier Sense Multiple Access)



CSMA: **ouvir antes de transmitir**

Se o canal está desocupado

- transmite a trama

Senão

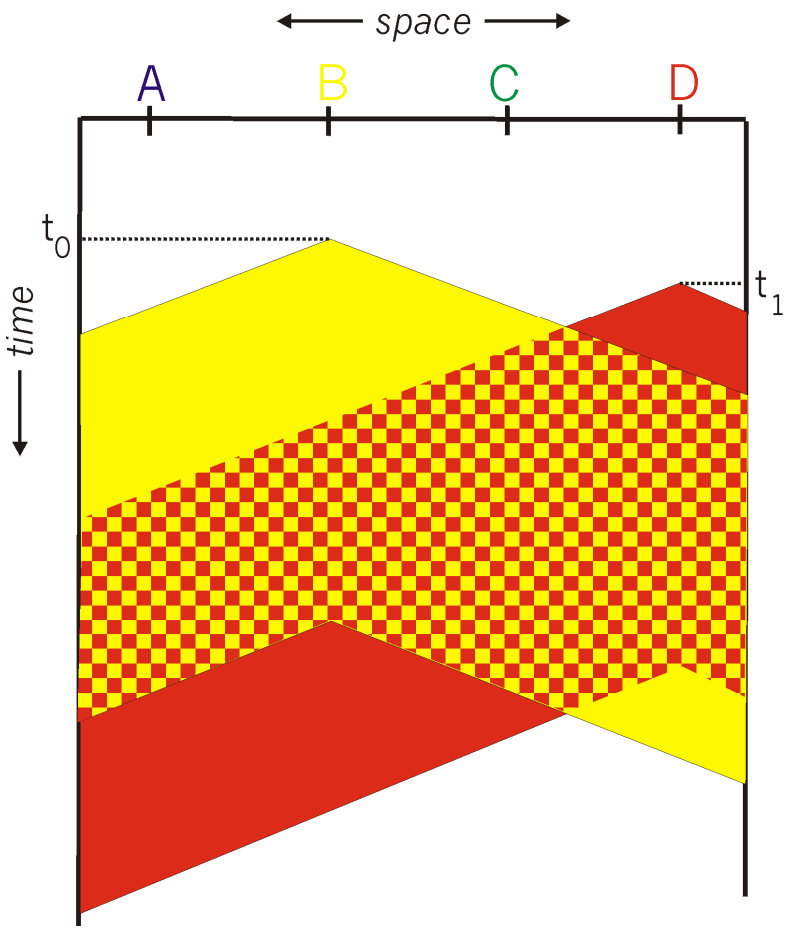
- retarda a transmissão

Colisões CSMA

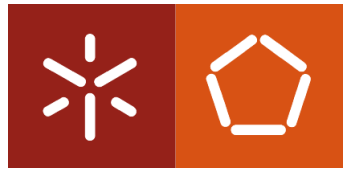
As colisões podem acontecer na mesma: o atraso de propagação faz com que a estação não se aperceba que outra estação já começou a transmitir

Nota: A distância e o tempo de propagação têm um papel importante na probabilidade de ocorrerem colisões

Colisão: O tempo de transmissão do pacote é desperdiçado



CSMA/CD (Collision Detection)

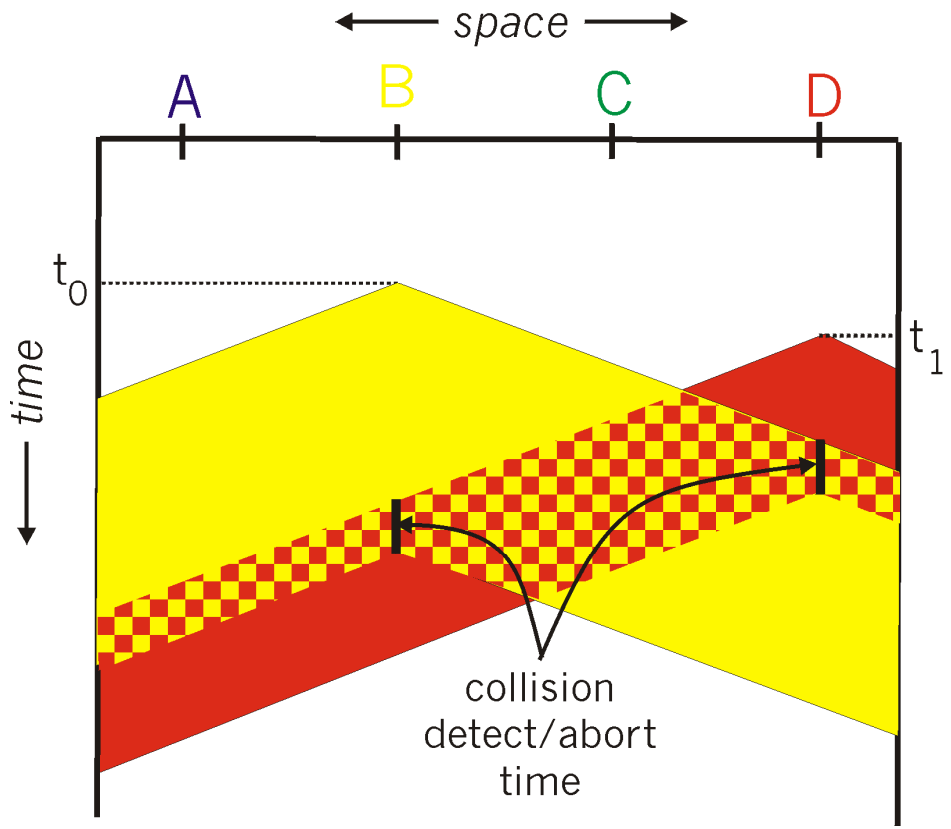


CSMA/CD: o transmissor escuta o canal antes de transmitir como no CSMA, mas depois continua à escuta enquanto transmite

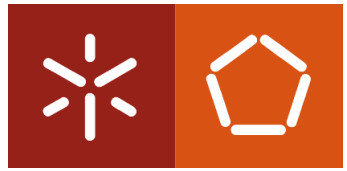
- As colisões são habitualmente detectadas num curto espaço de tempo
- Quando uma colisão é detectada a transmissão é abortada para não desperdiçar a largura de banda do canal

- **A detecção das colisões:**
  - É fácil nas redes LAN com fios: o sinal recebido é comparado com o transmitido.

CSMA/CD collision detection



Protocolos MAC sequenciais



O protocolos MAC baseados na partição dos canais

- Partilham o canal de forma justa e eficiente no caso da carga ser alta
- São ineficientes no caso da carga ser baixa: atraso no acesso ao canal, a largura de banda não excede R/N mesmo que só haja uma estação activa

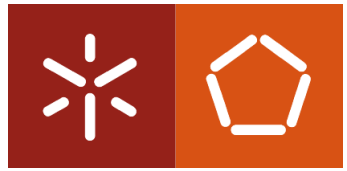
Protocolos MAC de acesso aleatório

- São eficientes no caso da carga ser baixa, uma estação pode disfrutar de toda a capacidade do canal
- No caso da carga ser alta o excesso de colisões traz uma grande sobrecarga

Protocolos MAC de acesso sequencial

- Tentam "apanhar o melhor dos dois mundos"

Protocolos MAC sequenciais

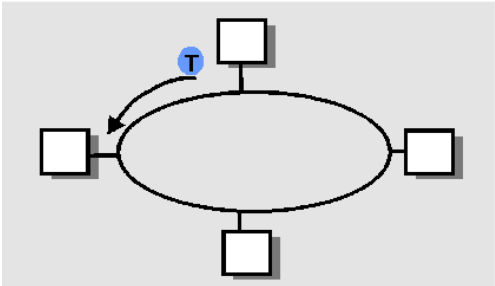


Polling:

- **Estação primária convida as estações secundárias para transmitir à vez**
- **Contras:**
  - Sobrecarga do *polling*
  - Latência
  - Um único ponto de falha (estação primária)

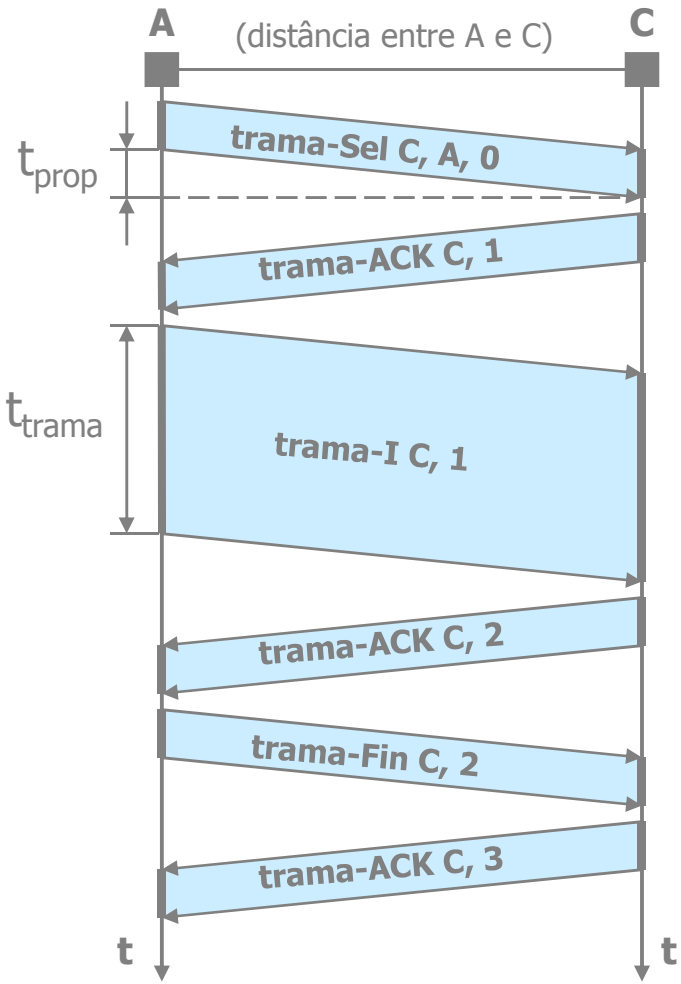
Passagem de testemunho (*token*):

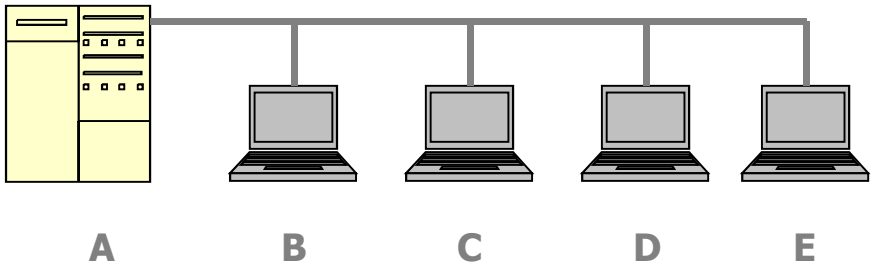
- **O testemunho é passado de uma estação para a estação seguinte de forma sequencial.**
- **Contras:**
  - Sobrecarga do *token*
  - Latência
  - Um único ponto de falha (*token*)



Controlo da ligação de dados

protocolo poll-select

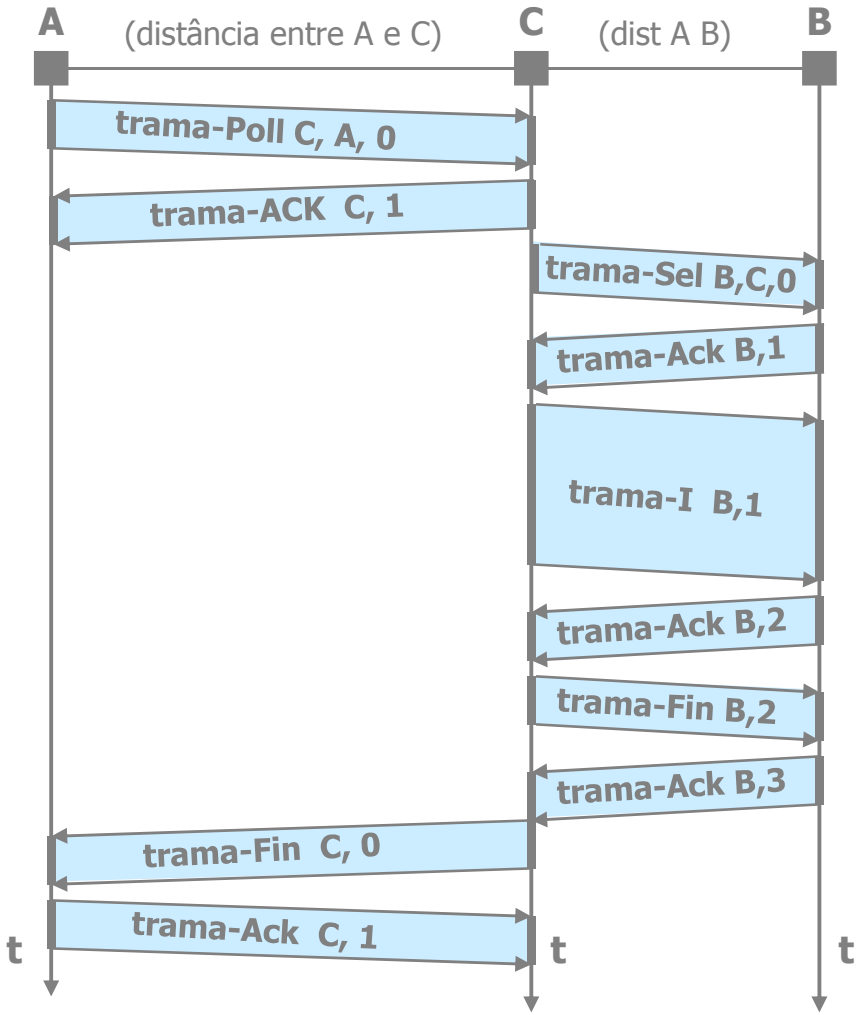




- **Considere-se que a estação (A) é a primária e que as restantes são estações secundárias**
- **A estação primária (A) selecciona a estação secundária (C) para lhe enviar dados**
- **Diz-se que (A) estabelece uma ligação lógica com a estação (C)**

Controlo da ligação de dados

protocolo poll-select



- **A estação primária (A) faz Polling à estação secundária (C) para lhe dar o control da linha**
- **A estação secundária (C) passa a comportar-se como primária e estabelece uma ligação lógica com a estação secundária (B) para lhe enviar dados**
- **Ao terminar a ligação com (B), a estação (C) devolve o control da linha à estação primária (A)**