

# Arquitetura de Redes de Computadores

Prof. Macêdo Firmino

Camada de Enlace

# Camada de Enlace de Dados

- Enlace é um canal de comunicação entre nós adjacentes. Para isso ela:
  - Divide o fluxo de *bits* recebidos em unidades de dados denominados quadros ou *frames* (Enquadramento).
  - Acrescenta um cabeçalho ao frame para definir o transmissor e o receptor (Endereçamento físico);
  - Adiciona um campo com a finalidade de propor confiabilidade aos dados recebidos, através de um mecanismo de detecção, perdas e retransmissão de quadros (Controle de erro);
  - Controla a velocidade da transmissão caso a velocidade de processamento do receptor for menor do que a velocidade que eles são produzidos (Controle de fluxo);
  - Controla o meio físico se existirem muitos computadores compartilhando o mesmo meio físico e todos desejarem enviar os dados ao mesmo tempo (Controle de acesso).

Pergunta???

Como é feito o enquadramento?

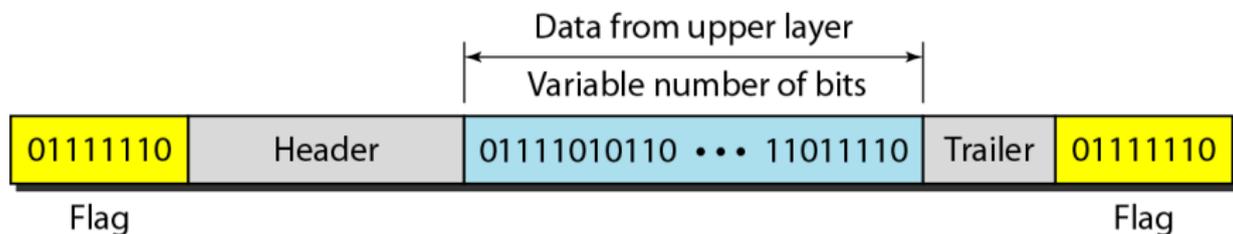
# Enquadramento

A camada física transmite bits na forma de sinal, cabendo a camada de enlace empacotar os bits em frames (informação) A enquadramento separa uma mensagem de outras mensagens.

- Existem duas maneiras de delimitar um *frame*:
  - Tamanho fixo: o tamanho do *frame* é suficiente para determinar a sua delimitação.
  - Tamanho variável: são utilizados padrões de bits pré-determinados para definir o início e o final do *frame*.

# Enquadramento - Tamanho Variável

A maioria dos protocolos utilizam um padrão de 8 bits (01111110) como delimitador de início e fim do *frame*.

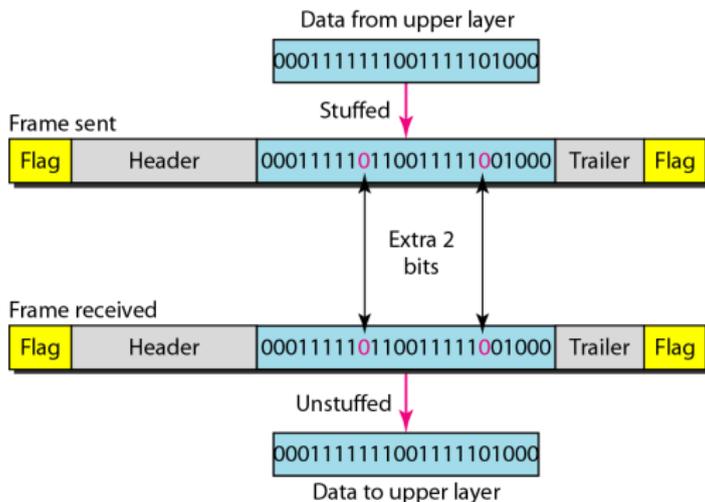


## Pergunta???

Se o padrão de delimitador (01111110) aparecer no meio dos dados?

# Enquadramento - Tamanho Variável

Enxertamos um bit (stuffing) para diferenciar esse padrão do delimitador. Esse bit é eliminado pelo receptor. O bit extra é o acrescentado se for encontrado os bits 011111, será acrescentado o bit 0 (extra), independente do valor do bit que viria posteriormente.



Pergunta???

Como é feito o endereçamento físico?

## Endereçamento Físico (MAC)

Cada estação numa rede possui seu próprio adaptador de rede. Cada adaptador possui um endereço de 6-bytes (48 bits). Estes endereços que são escritos normalmente em notação hexadecimal (12 dígitos) separada por dois pontos entre os bytes. Por exemplo:

0A : 13 : 78 : B3 : FF : 02

Os três primeiros octetos são destinados à identificação do fabricante, os 3 posteriores são fornecidos pelo fabricante. Trata-se de um endereço único, i.e., não existem, em todo o mundo, duas placas com o mesmo endereço.

Pergunta???

Como é feito o controle de erro?

# Controle de Erros

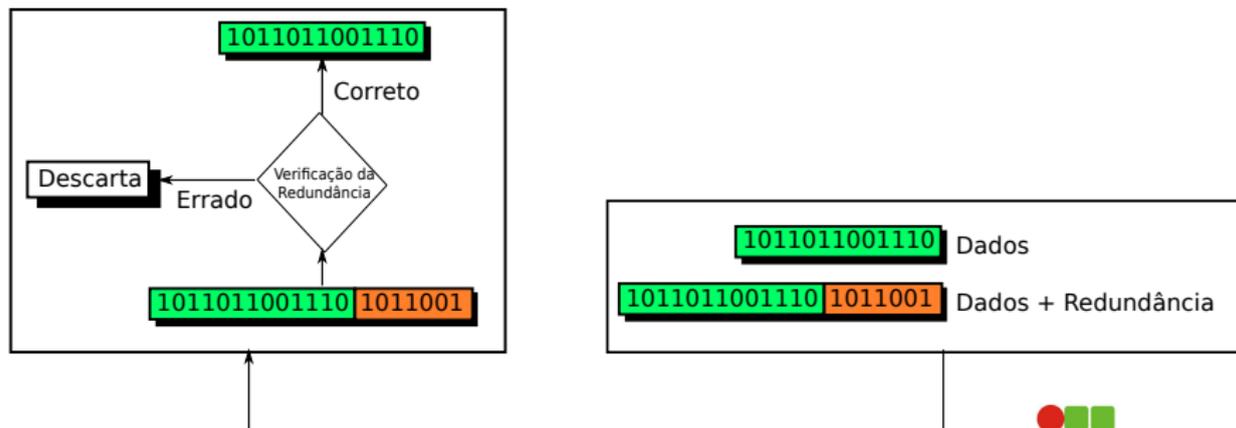
- O controle de erro permite ao receptor informar ao transmissor sobre quais quadros não foram recebidos ou recebidos de forma corrompida, solicitando a retransmissão;
- O controle de erros implementado na camada de enlace é baseado na **detecção** e retransmissão.

Pergunta???

Como é feita a detecção de erro?

# Detecção de Erros

As detecções de erros utilizam o conceito de redundância, que é a técnica de adicionar *bits* extras na unidade de informação. Dessa forma, enviamos os *bits* extras junto com os dados par que o receptor detecte corrompidos.

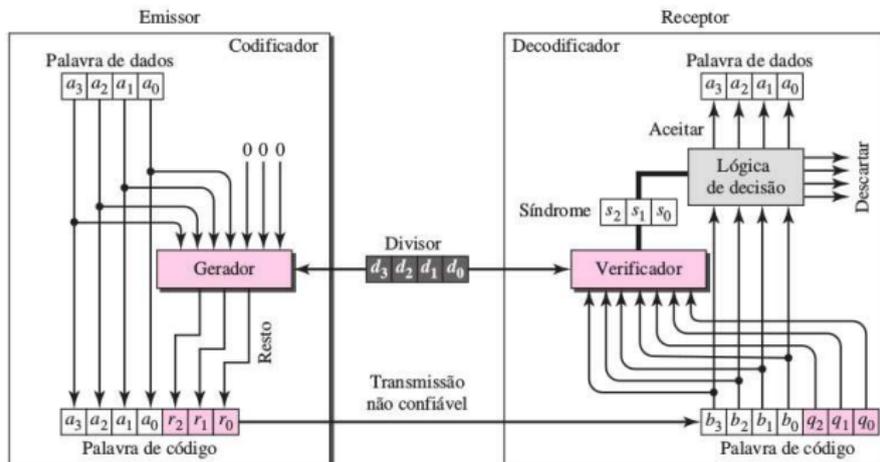


# Detecção de Erros

- Os dois principais tipos de detecção de erros são:
  - CRC;
  - *Checksum*.

# CRC

A técnica CRC (Verificação de Redundância Cíclica) baseia-se em uma divisão binária. Uma seqüência de *bits* de redundância, denominados *bits* de CRC, são acrescentados no final do bloco de dados de maneira a tornar todo o bloco resultante divisível por um número binário predeterminado (divisão modulo 2). No lado receptor, o bloco é dividido pelo mesmo número binário.

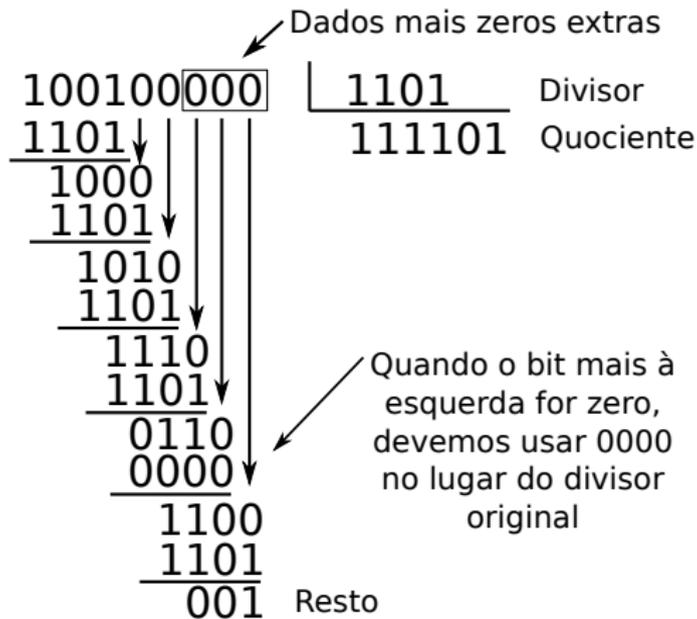


## CRC - Aritmética Binária

O cálculo do CRC é realizado através de uma operação de divisão aplicada a números binários, porém as operações de subtração são substituídas por operações de ou exclusivo (XOR). O que importará, no final, será obter o resto desta divisão.

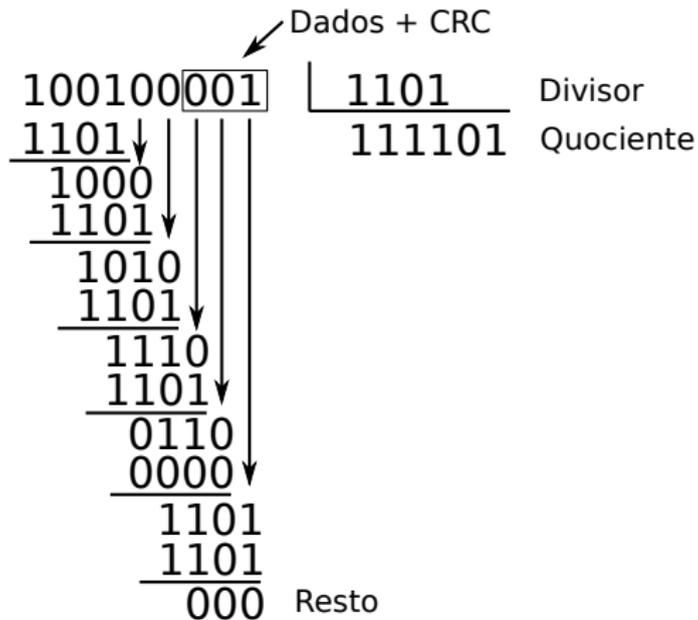
O receptor realiza o mesmo processo de divisões (dados com o CRC ). Se o resto da divisão for zero, não existe erros, caso contrário, tudo é descartado.

# CRC - Gerador (Exemplo)



**100100** **001** Dados + CRC

# CRC - Verificador (Exemplo)

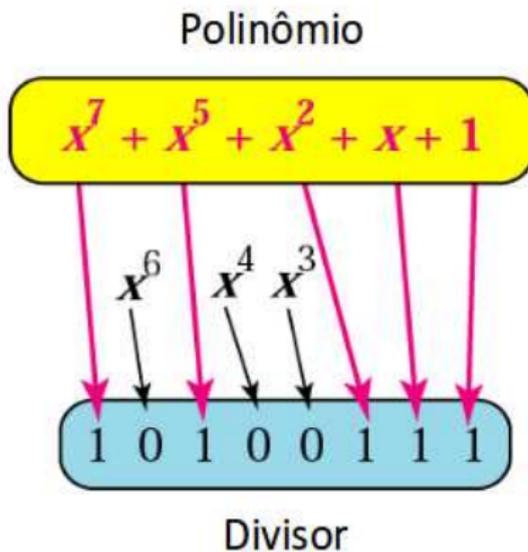


# CRC - Padrões

<i>Name</i>	<i>Polynomial</i>	<i>Application</i>
CRC-8	$x^8 + x^2 + x + 1$	ATM header
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x^2 + 1$	ATM AAL
CRC-16	$x^{16} + x^{12} + x^5 + 1$	HDLC
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	LANs



# CRC - Padrões



## Checksum

A técnica de *Checksum* baseia-se na soma binária de complemento um. Se o número tiver mais de  $n$  bits, os bits extras mais à esquerda precisam ser adicionados aos  $n$  bits mais à direita (wrapping).

Os dados são divididos em segmentos de dados. Todos os segmentos são somados, através das regras da aritmética de complemento de um. Após a soma é feito o complementado (inverte os bits) e o resultado gerado é o *Checksum*.

O *Checksum* é normalmente usado em protocolos das camadas superiores (TCP, UDP e IP).

## Checksum - Gerador

O bloco de dados é dividido segmentos de  $n$ -bits. O valor do checksum é inicializado em 0. Todos os seguimentos são somados, inclusive o checksum, através das regras da aritmética de complemento de um. Finalmente é realizado o complemento do resultado da soma para gerar o *checksum*. O checksum é enviado junto com os dados.

# Checksum - Gerador

**1010100110111001** Dados

10101001 10111001    Dividir o bloco de dados com 8bits

```
10101001
10111001
00000000
-----
101100010 ← Soma
-----
01100010
      1 ← Complemento de um
-----
01100011
10011100 ← Complemento
10011100 ← Checksum
```

**1010100110111001** **10011100**

Dados + Checksum

## Checksum - Verificador

O bloco de dados é dividido em segmentos de  $n$ -bits. Todos os seguimentos são somados, através das regras da aritmética de complemento de um. Finalmente é realizado o complemento do resultado da soma para gerar o *checksum*. Se o resultado é zero, os dados são aceitos, caso contrário, ela é rejeitada.

# Checksum - Verificador

1010100110111001 10011100

Dados + Checksum

10101001 10111001 10011100

Dividir o bloco de dados com 8bits

```
10101001
10111001
10011100
-----
11111110 ← Soma
11111110
      1 ← Complemento de um
-----
11111111 ← Complemento
00000000 ← Checksum
00000000
```

# Atividade

01. Dados o polinômio corretor  $G(x) = x^3 + x + 1$  e a sequência de bits de informação 1110101, forneça a sequência de bits a ser transmitida, conforme a técnica de detecção de erros CRC.
02. Calcule o valor do checksum de 8 bits da sequência 01100111001111011010101000110111

# Controle de Fluxo

Todo dispositivo possui uma quantidade de memória reservada para armazenar dados que chegam, até que sejam processados;

O fluxo de dados não deve permitir que essa memória seja inundado por dados. Para isso, o receptor deverá informar ao transmissor que o limite de capacidade está próximo de ser alcançado e requer uma taxa de transmissão menor ou até mesmo uma parada na transmissão;

# Controle de Fluxo

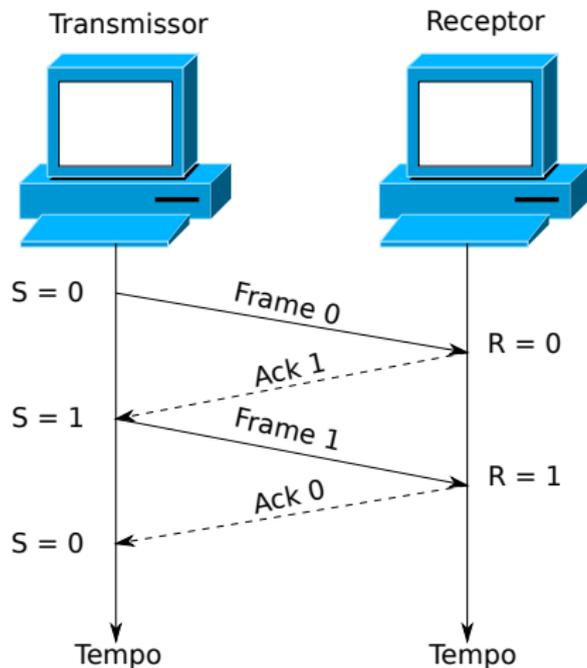
São mecanismos de controle de fluxo:

- *Stop-and-Wait* ARQ;
- *Go-Back-N* ARQ;
- *Selective-Repeat* ARQ.

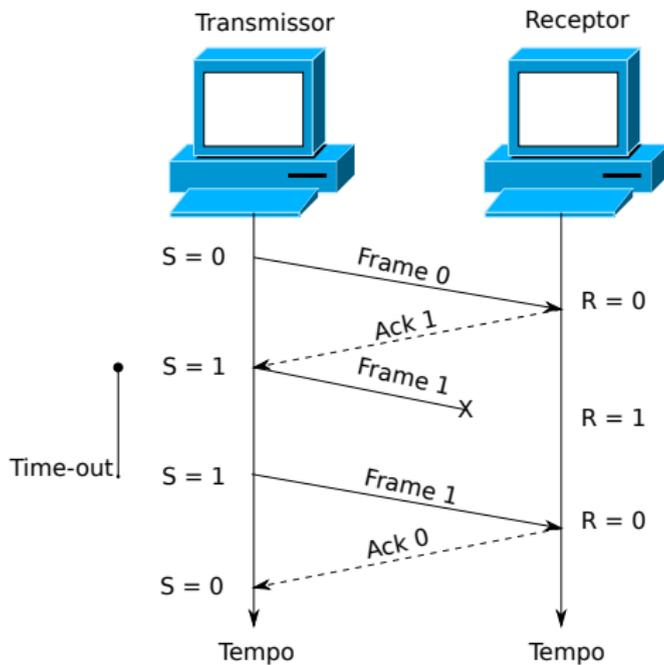
## Stop-and-Wait ARQ

- O dispositivo transmissor mantém uma cópia do último *frame* (quadro) transmitido até receber uma resposta de confirmação (*frame* de ack, *acknowledgment*);
- Os quadros são numerados alternadamente em 0 e 1;
- Um *frame* de dados 0 é confirmado por um *frame* ack 1, indicando que o receptor aceitou o *frame* de dados 0 e espera o *frame* de dados 1. O receptor envia respostas positivas (ACKs) somente para *frame* recebidos e aceitos.
- O transmissor dispara um relógio quando é enviado um *frame*. Se a resposta ACK não for recebida dentro do intervalo de tempo predefinido, o transmissor assume que houve uma perda desse *frame* e o reenvia;

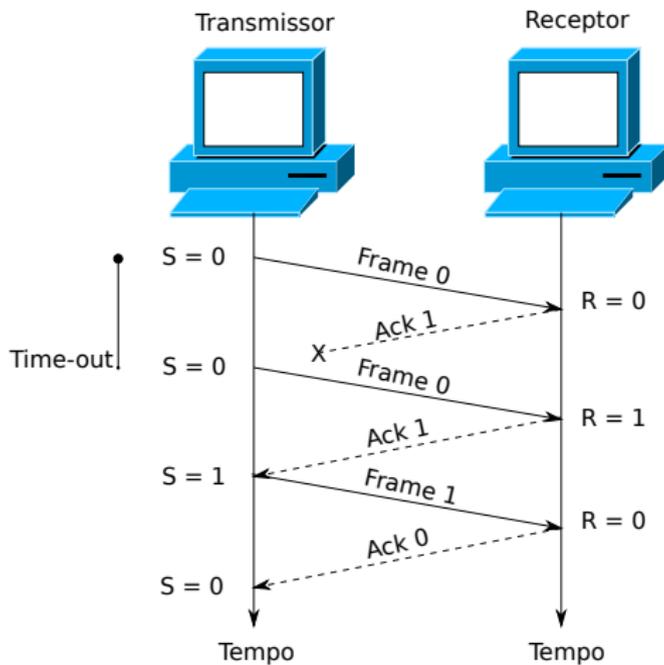
# Stop-and-Wait ARQ - Funcionamento Normal



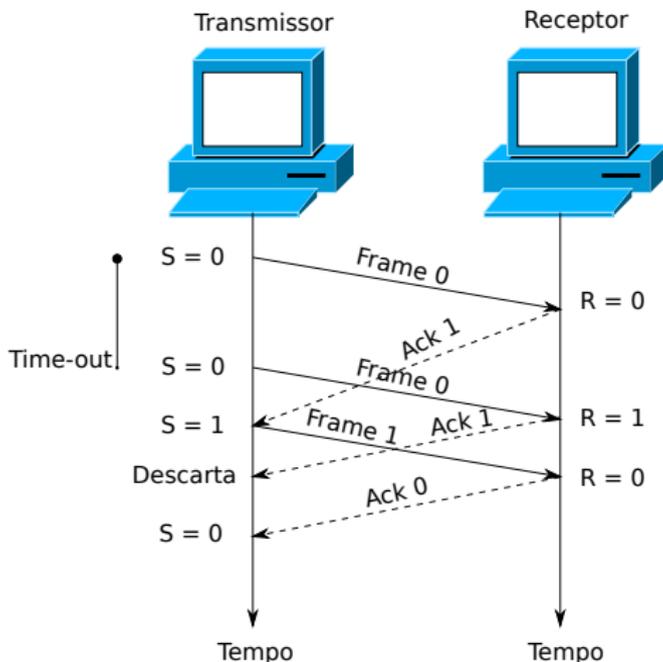
# Stop-and-Wait ARQ - Frame perdido



# Stop-and-Wait ARQ - ACK perdido

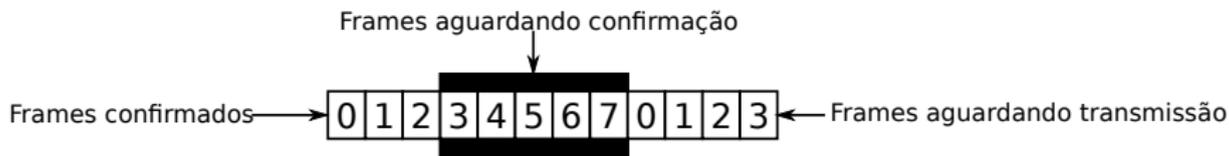


# Stop-and-Wait ARQ - Atraso ACK



## Go-Back-N ARQ

- Para melhorar a eficiência da comunicação é enviado múltiplas *frames* antes da transmissor receber um quadro de confirmação;
- No transmissor usamos o conceito de janela para manter em memória todos os *frames* pendentes até que sejam recebidos os respectivos ACKs. Um *frames* só deixa a janela quando receber um ACK, liberando o espaço para outros *frames*;
- O transmissor dispara um relógio para cada *frame* enviado.



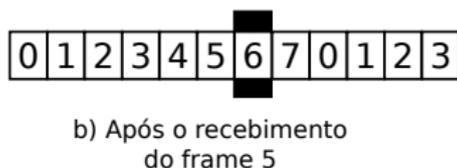
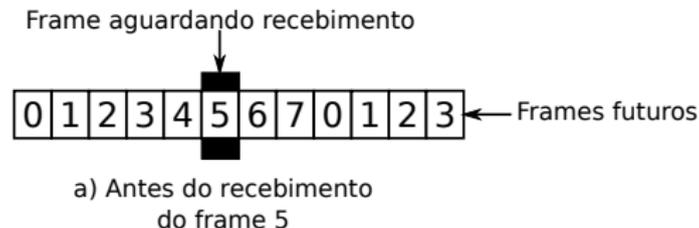
a) Antes da confirmação



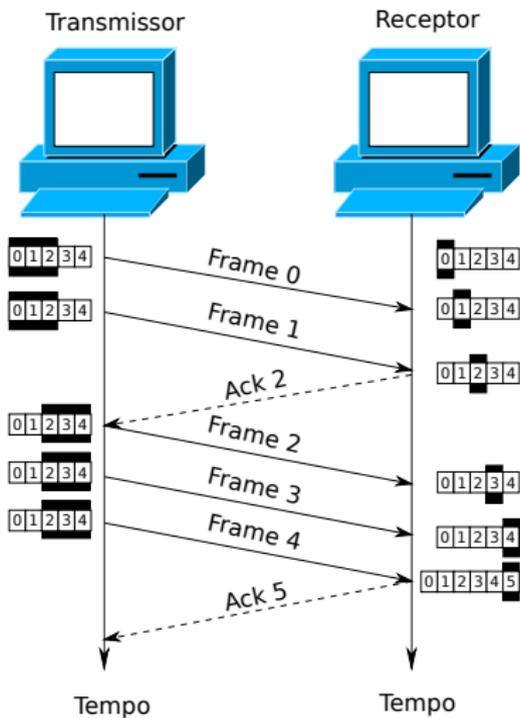
b) Após receber 2 ACKs

## Go-Back-N ARQ

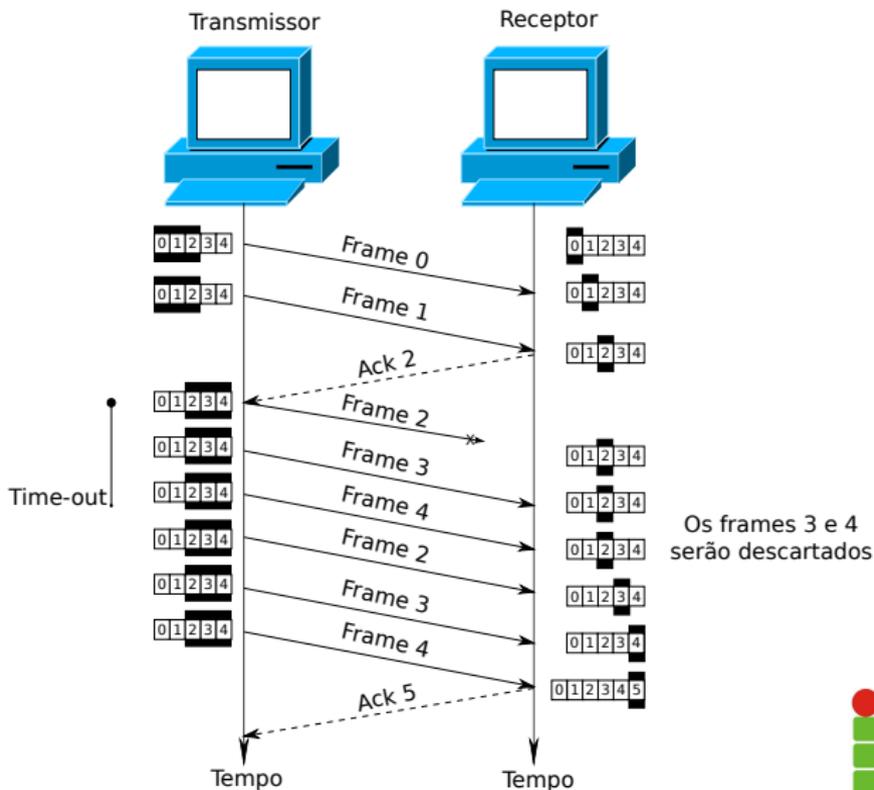
- No receptor a janela é igual a 1, ou seja, o receptor está sempre esperando um quadro específico, recebido em ordem. Os frames recebidos fora de ordem são descartados;
- O receptor não precisa confirmar todos cada *frame* isoladamente. Ele pode ser configurado para enviar um único ACK para confirmar todos os *frames* que ele receber num intervalo de tempo.



# Go-Back-N ARQ - Funcionamento Normal



# Go-Back-N ARQ - Frames Perdidos

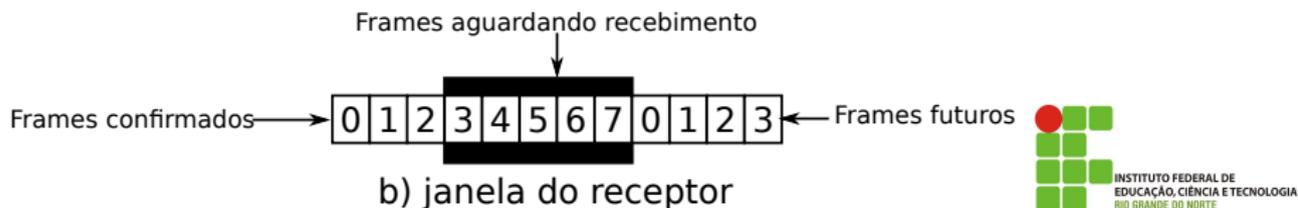
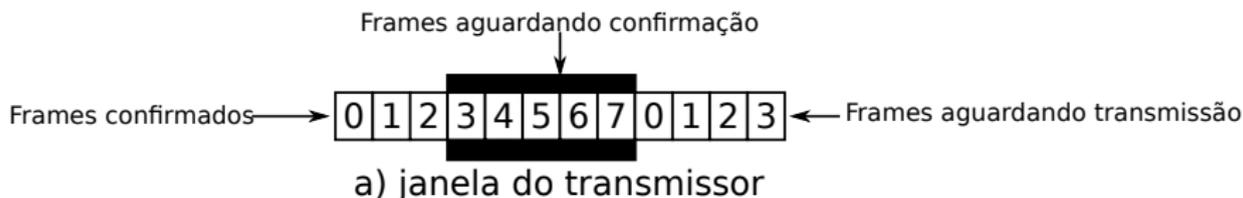


## Selective-Repeat ARQ

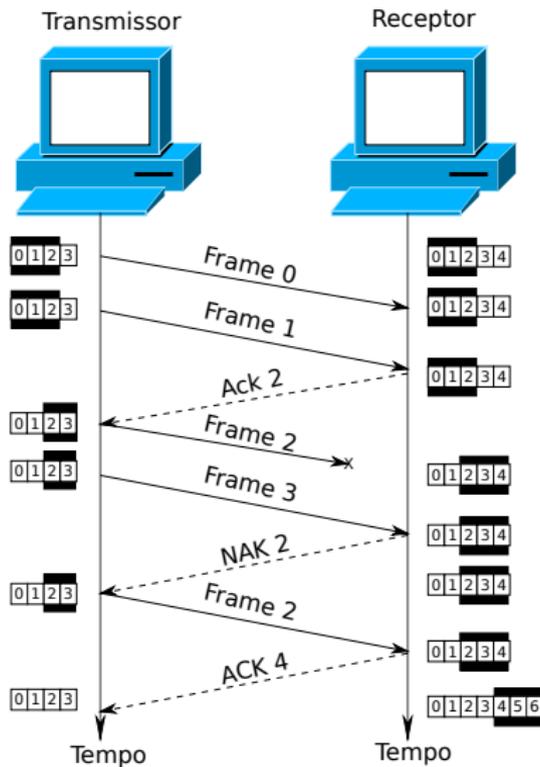
- No *Go-Back-N* o receptor aguarda somente um quadro, descartando os demais. Para melhorar a eficiência, na técnica *Selective-Repeat* não é necessário reenviar  $N$  frames quando apenas um está danificado. Somente o quadro danificado é retransmitido;
- É definida uma mensagem de confirmação negativa (NAK - *Negative Acknowledgment*) que informa ao transmissor que um determinado quadro não foi recebido;
- O transmissor dispara um relógio para cada *frame* enviado.

## Selective-Repeat ARQ

A configuração do transmissor para o mecanismo *Selective-Repeat* são os mesmos mostrados no *Go-Back-N*. Entretanto, a janela de recepção deverá ter o mesmo tamanho da janela de transmissão. Esta janela irá ser utilizada para que o receptor possa receber uma faixa de números de seqüência.



# Selective-Repeat ARQ - Funcionamento



# Controle de Acesso

Quando nós ou estações são conectados e usam um enlace comum, chamado enlace multiponto ou broadcast, precisamos de um protocolo de acesso múltiplo para coordenar o acesso ao meio físico. Se mais de uma estação transmitir ao mesmo tempo ocorrerá um conflito de acesso (colisão) e os quadros são perdidos. Por exemplo, redes locais sem fio, comunicação por satélite e telefonia.

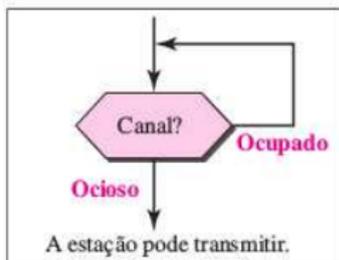
O procedimento deve garantir que todos consigam se comunicar, que duas estações não transmitam ao mesmo tempo, não interrompam a quem está transmitindo, não monopolize a transmissão e assim por diante. As principais abordagens de acesso múltiplo são:

# Acesso Randômico

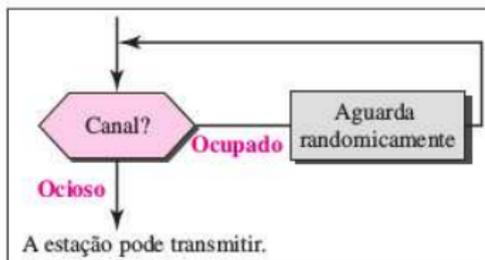
- Um dos principais tipos de controle de acesso é chamado de Acesso Randômico.
- Nele cada estação tem direito ao meio, sem ser controlada por outra estação. A cada instante, uma estação que tenha dados a enviar tomar sua decisão sobre enviar ou não esses dados.
- As principais técnicas de acesso randômico:
  - CSMA;
  - CSMA/CD;
  - CSMA/CA.

No método CSMA (*Carrier Sense Multiple Access*) uma estação deve verificar o meio antes de tentar utilizá-lo; Se perceber canal ocupado, adia a transmissão.

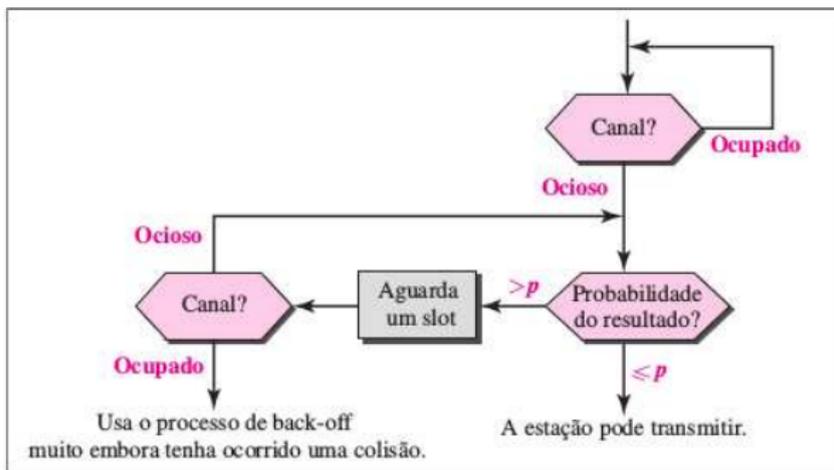
Existem três implementações do CSMA: o método persistente, não persistente e o persistente com probabilidade.



a. 1-persistent

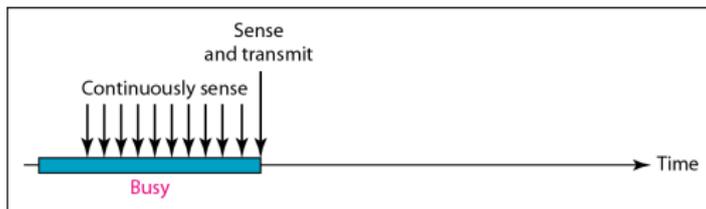


b. Nonpersistent

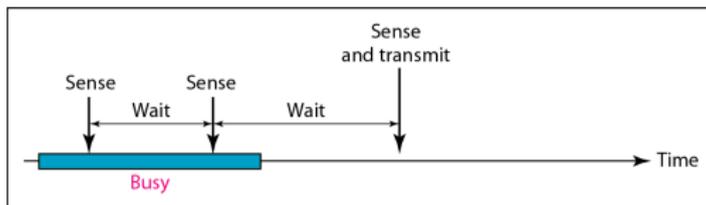


c. p-persistent

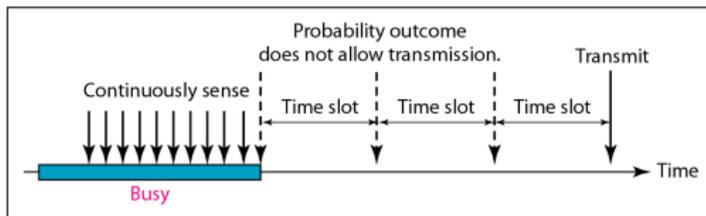
# CSMA



a. 1-persistent



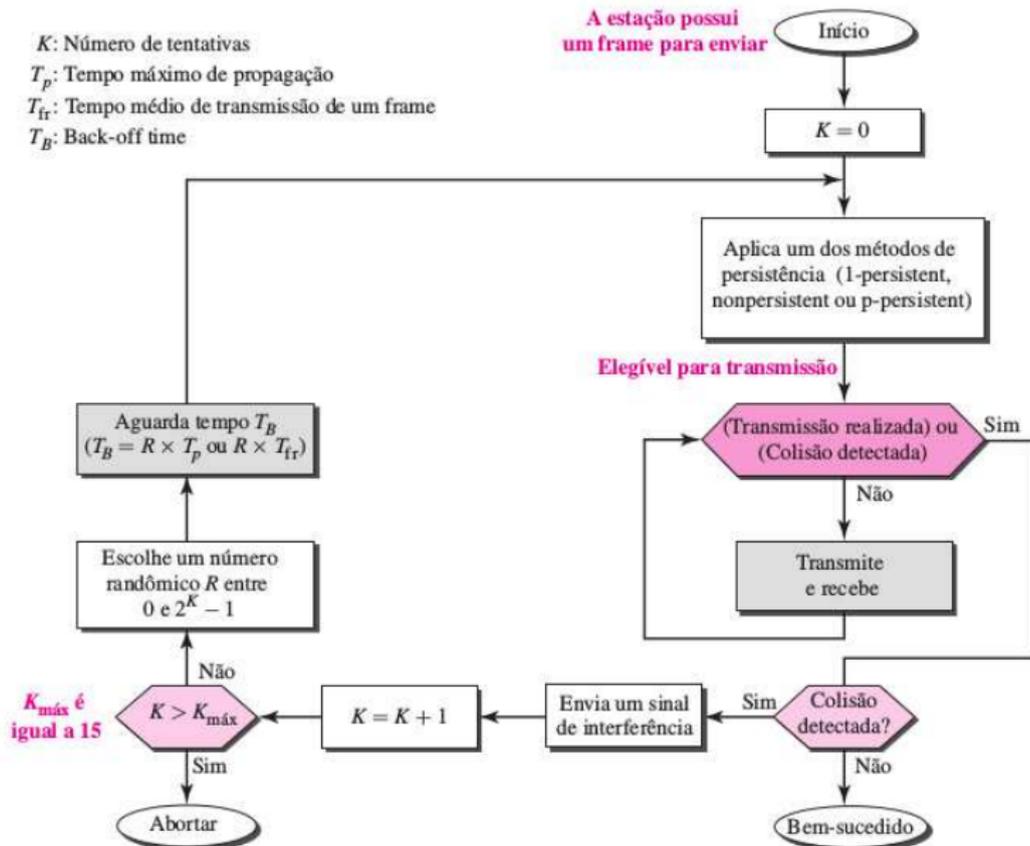
b. Nonpersistent



c. p-persistent

O CSMA/CD (CSMA com *Collision Detection*) estende o algoritmo CSMA para tratar colisão. Nesse método, a estação monitora o meio após ela transmitir um frame para verificar se ocorreu colisão. Caso não tenha ocorrida uma colisão, a estação finaliza. Caso contrário, a estação aborta a transmissão imediatamente, envia um sinal de interferência (*jamming signal*) e posteriormente irá retransmitir o frame.

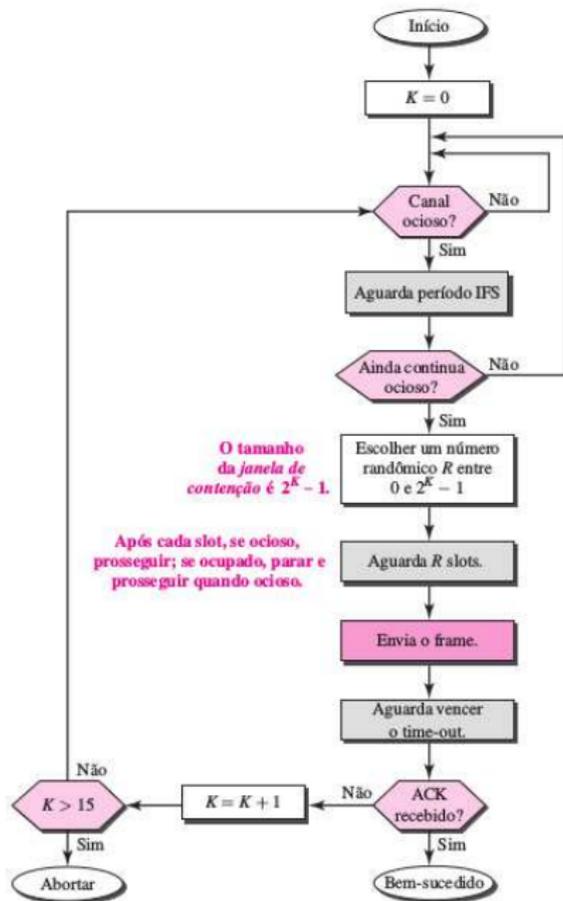
$K$ : Número de tentativas  
 $T_p$ : Tempo máximo de propagação  
 $T_{tr}$ : Tempo médio de transmissão de um frame  
 $T_B$ : Back-off time



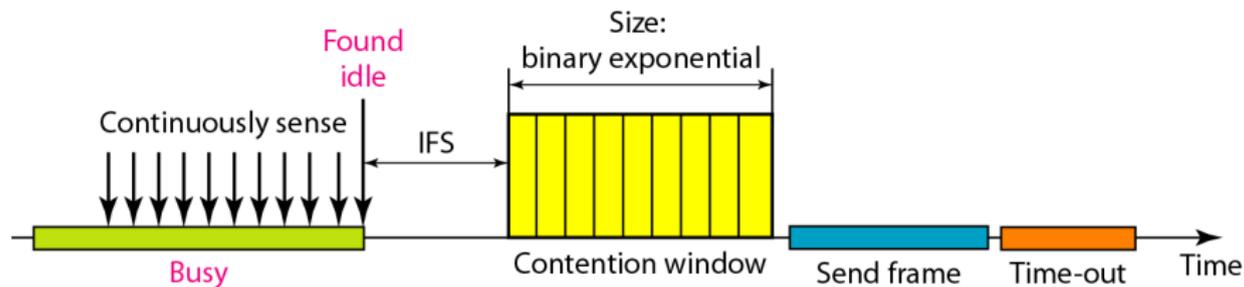
# CSMA/CD

- O CSMA/CD identifica quando o canal está disponível para a transmissão. Neste momento a transmissão é iniciada. O mecanismo CD (detecção de colisão) ao mesmo tempo obriga que os nós escutem a rede enquanto emitem dados. Se o mesmo detecta uma colisão, toda transmissão é interrompida e é emitido um sinal (“jam”) para anunciar que ocorreu uma colisão. Para evitar colisões sucessivas o nó espera um período aleatório e volta a tentar transmitir.
- O método CSMA/CD é utilizado nas redes Ethernet padrão.
- O sinal jam (congestionamento) é uma mensagem para informar as outras estações que ocorreu uma colisão durante a transmissão. Todas as estações irão descartar o *frame* recebido;

Em redes sem fio, o nível do sinal recebido, normalmente tem muito pouca energia. Conseqüentemente, detectar uma colisão é uma tarefa bem mais complexa do que nas redes cabeadas. Para evitar colisão em redes sem fio foi criada o CSMA/CA (CSMA com *Collision Avoidance*). O CSMA/CA procura somente evitar ao máximo colisões e não detecta-las. As colisões são evitadas por meio da utilização de três estratégias: espaçamento entre quadros (IFS), janela de contenção e confirmações (ACKs).



# CSMA/CA



Quando um canal se encontra ocioso, a estação não envia dados imediatamente. Ela aguarda por certo tempo, denominado espaçamento entre frames (*interframe space*) ou IFS. Esse tempo também pode ser utilizado para priorizar estações ou tipos de frame. Por exemplo, uma estação a qual é atribuído um IFS menor tem maior prioridade.

Uma estação que já se encontra pronta para transmitir escolhe um número randômico de slots para seu tempo de espera (janela de contenção). Esse número aleatório é proporcional a quantidade de tentativas de transmissão. Ela possui o tempo na primeira vez e, em seguida, vai dobrando de acordo com o número de tentativas de transmissão.