

Nível de Enlace

Profa. Débora Christina Muchaluat Saade

Laboratório MídiaCom - UFF

debora@midiacom.uff.br

<http://www.midiacom.uff.br/debora>

Nível de Enlace

- **Funcionalidades principais:**
 - *Oferecer serviços de transmissão de quadros ao nível de rede*
 - *Delimitação de quadros*
 - *Controle de erros:*
 - Detecção de erros - obrigatório
 - Correção de erros - opcional
 - *Controle de fluxo – opcional*

Detecção e Correção de Erros

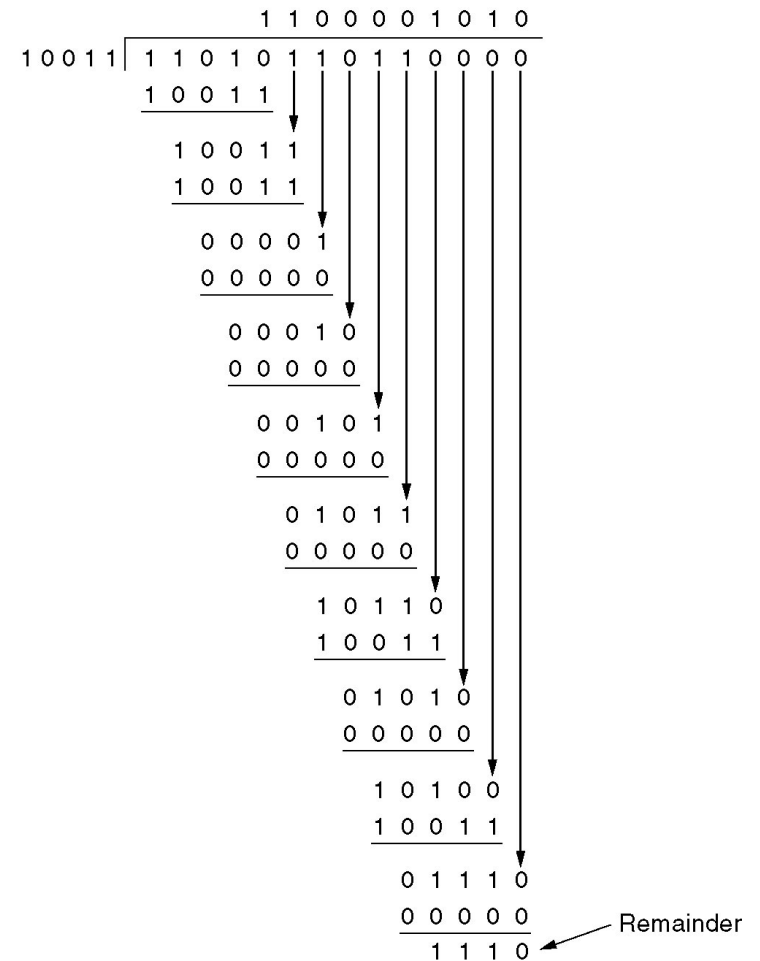
- **Detecção de erro - obrigatória**
 - *checksum*
 - *CRC – Cyclic Redundancy Check*
- **Correção de erros - opcional**
 - *Código corretor de erro*
 - Ex.: Hamming Code
 - *Protocolo de Controle de Erros*
 - Retransmissão do quadro com erro

Detecção de Erro - CRC

Frame : 1 1 0 1 0 1 1 0 1 1

Generator: 1 0 0 1 1

Message after 4 zero bits are appended: 1 1 0 1 0 1 1 0 1 1 0 0 0 0

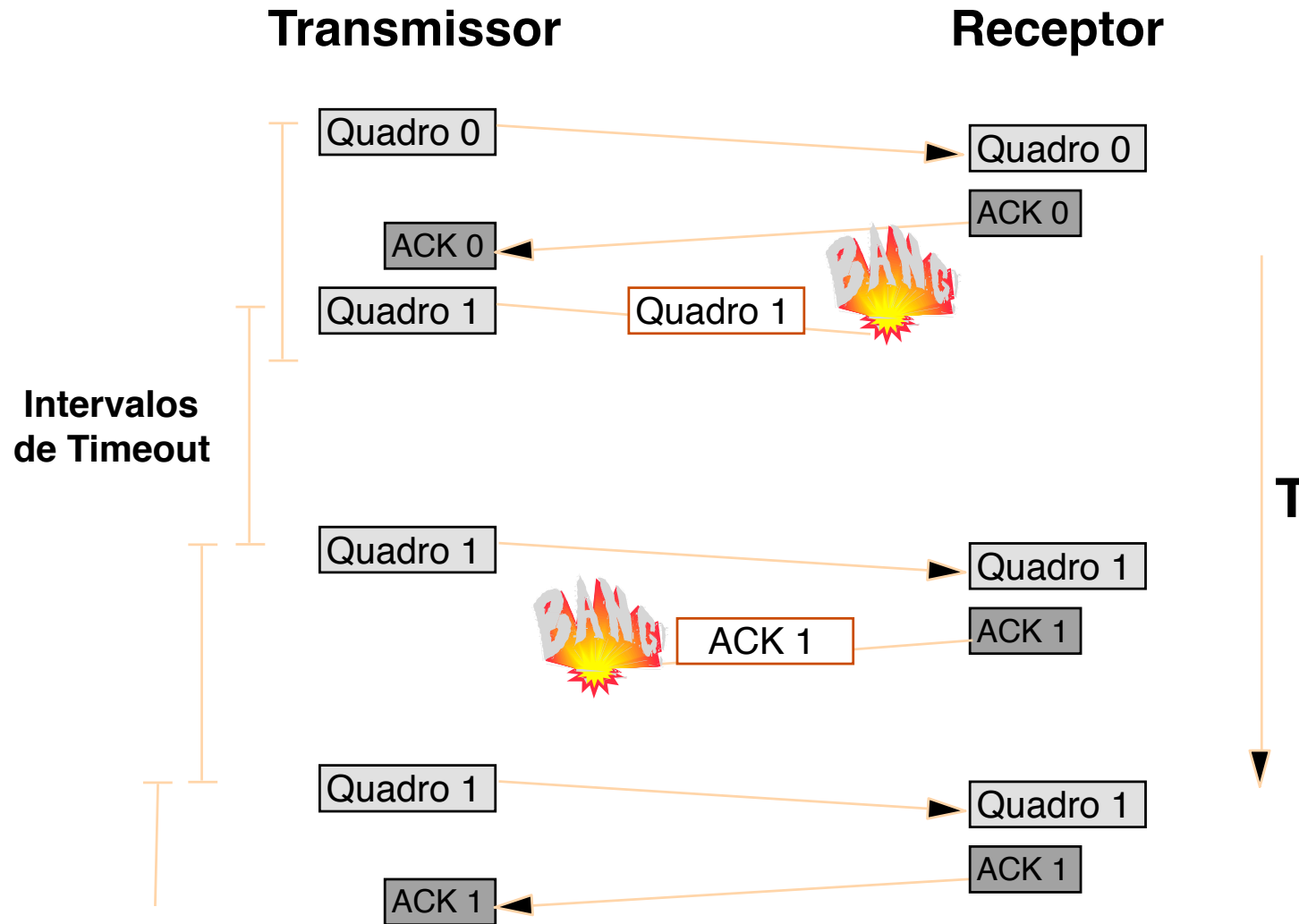


Cálculo do CRC

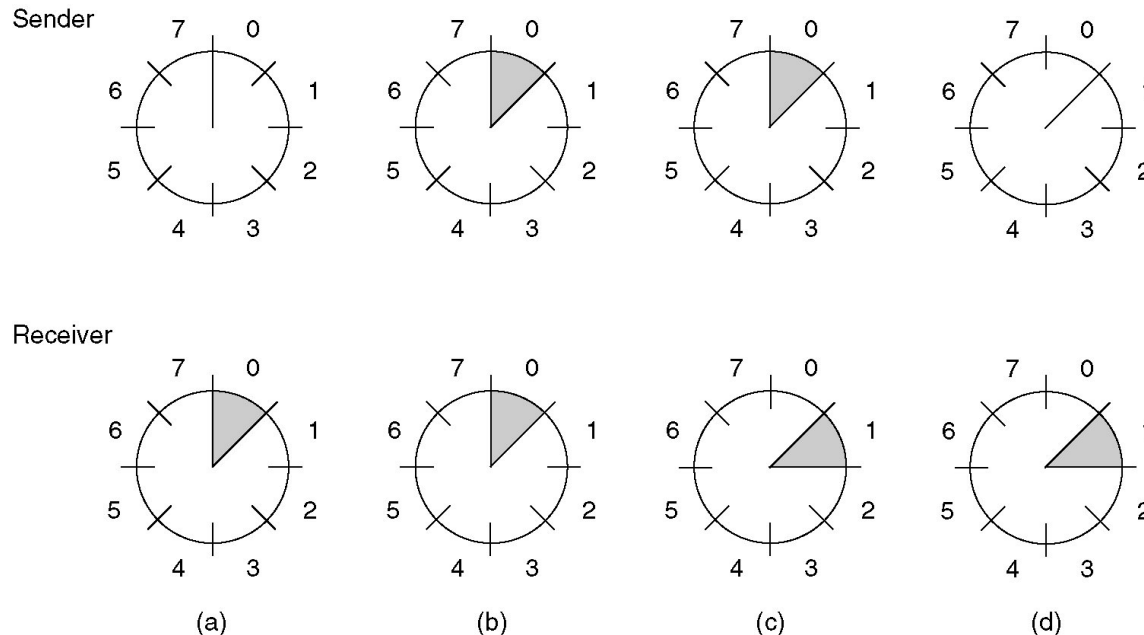
Protocolos de Controle de Erros

- ✓ **Todo quadro recebido deve ser reconhecido (ACK – acknowledgement)**
 - *Se ACK não chegar, retransmite depois do timeout*
- ✓ **Janela de transmissão e janela de recepção**
- ✓ **Protocolos baseados em Janela Deslizante (*Sliding Windows*)**
 - *Stop-and-Wait (One-Bit Sliding Window Protocol)*
 - *Go Back N Sliding Window*
 - *Selective Repeat Sliding Window*

Protocolo *Stop-and-Wait*



Protocolo Sliding Window



Sliding window de tamanho 1, com número de sequência de 3 bits:

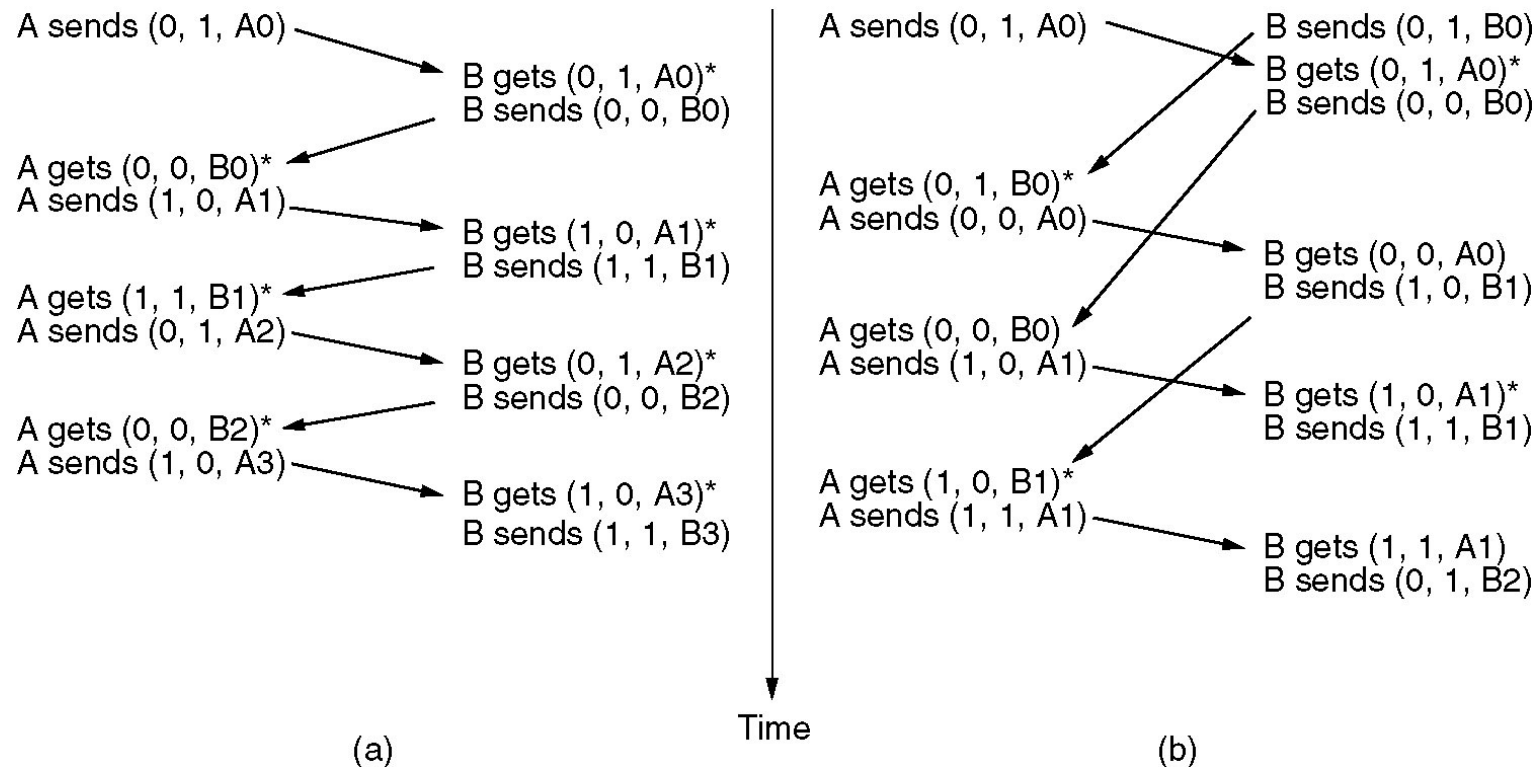
(a) Situação inicial

(b) Depois de transmitir o primeiro quadro

(c) Depois de receber o primeiro quadro

(d) Depois de receber o primeiro ACK

One-Bit Sliding Window Protocol



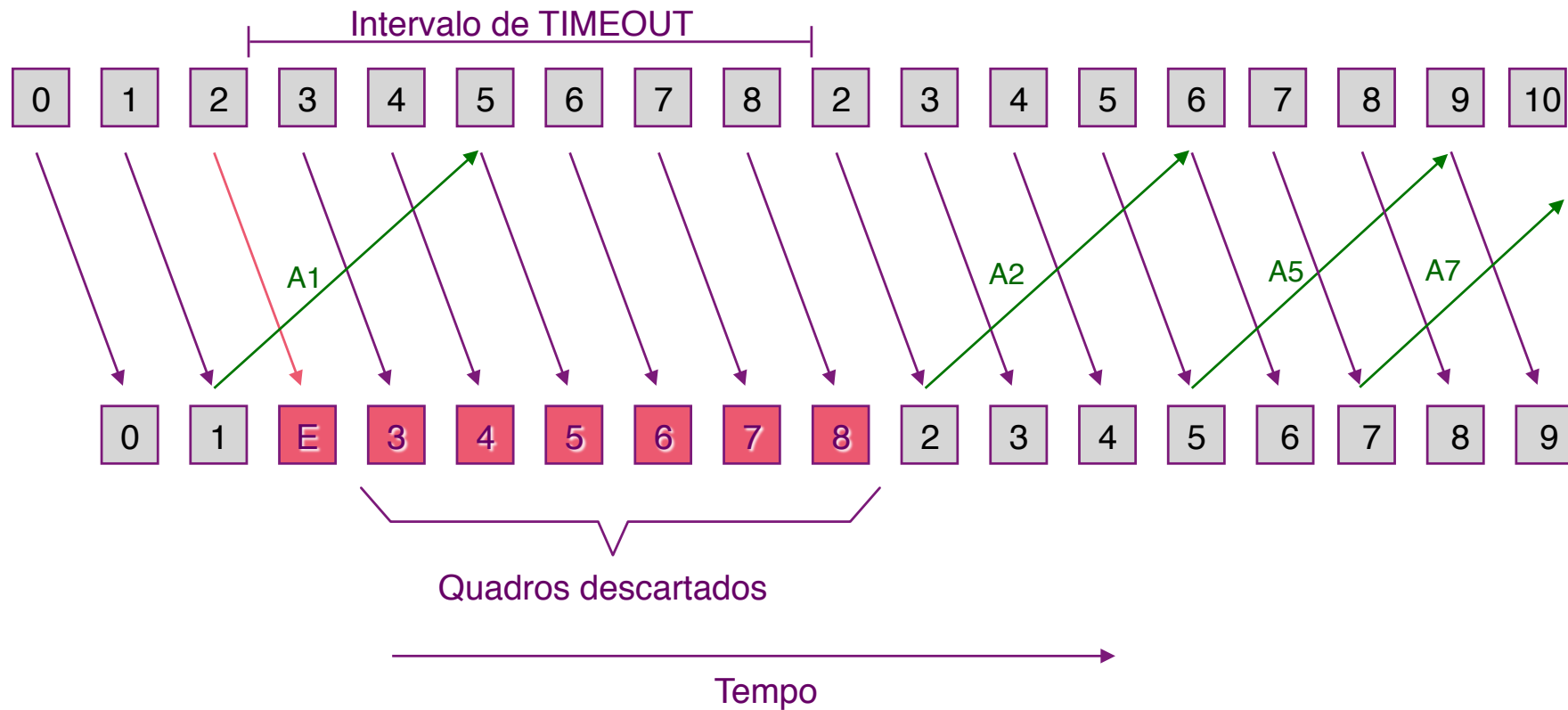
Dois cenários (a) caso normal (b) caso anormal

Notação (seq, ack, packet number).

Asterisco indica que nível de rede recebeu o pacote

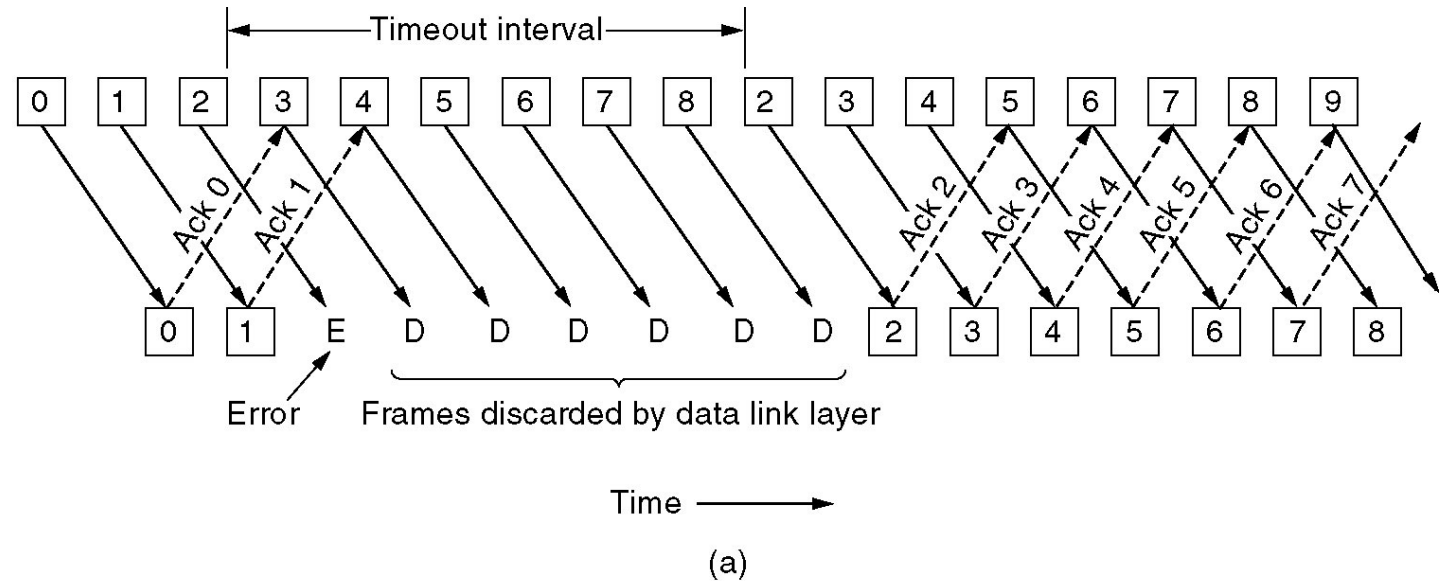
Controle de Erro: Protocolo *Go Back N*

(a) Receptor transmite ACK cumulativo

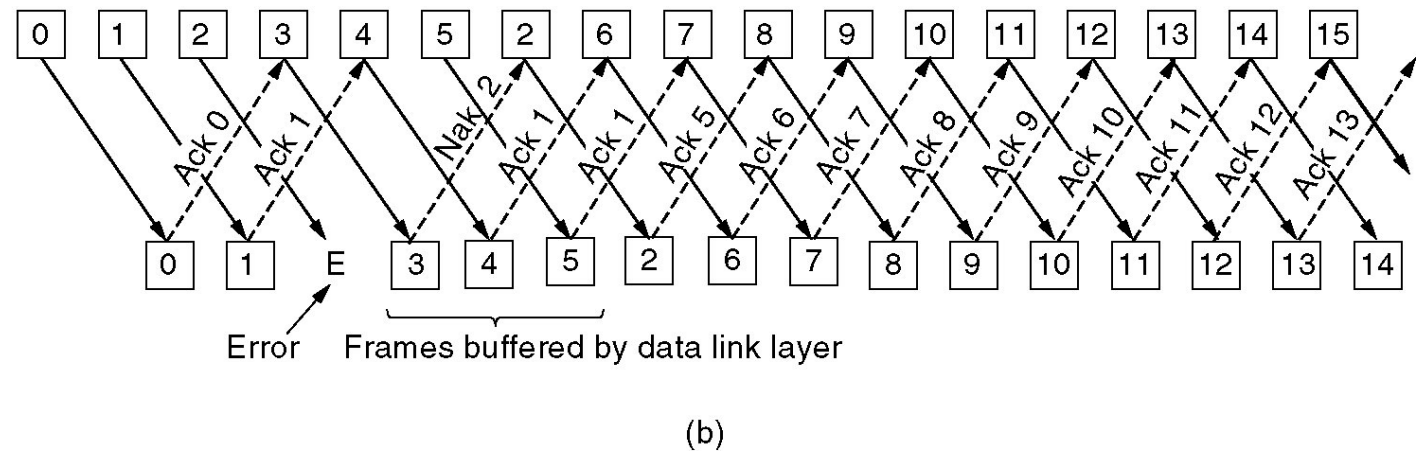


Protocolo Go Back N

(a) Tamanho da janela de recepção é 1

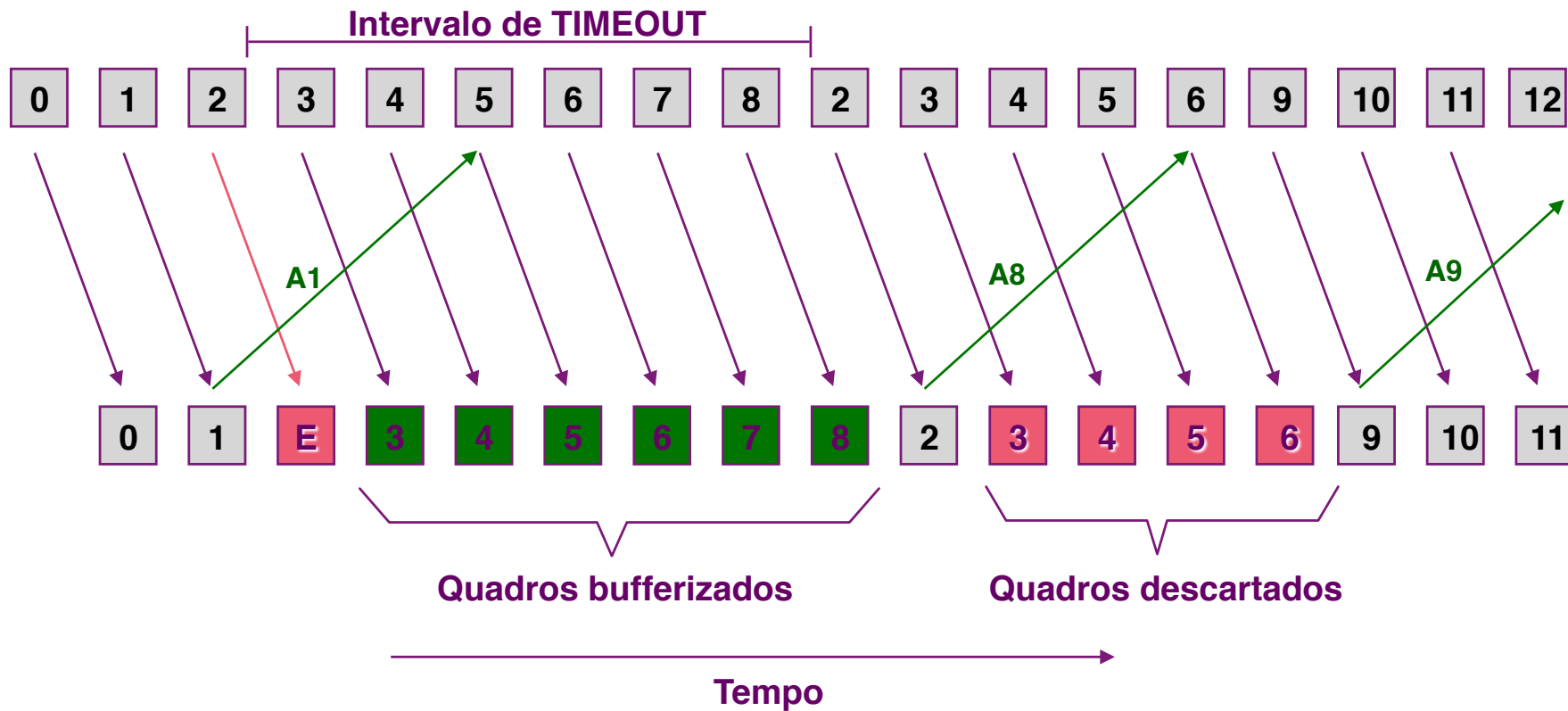


(b) Tamanho da janela de recepção é grande e usa NACK



Protocolo *Selective Repeat*

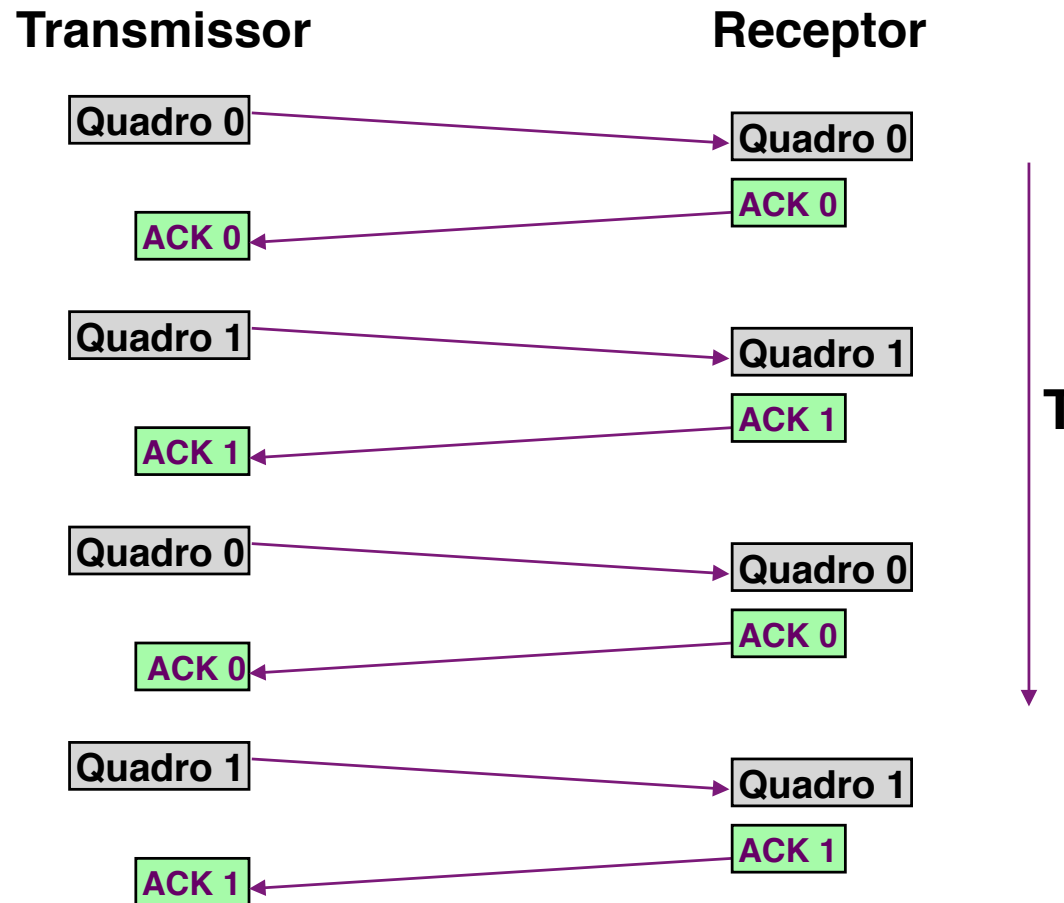
(a) Receptor transmite ACK cumulativo



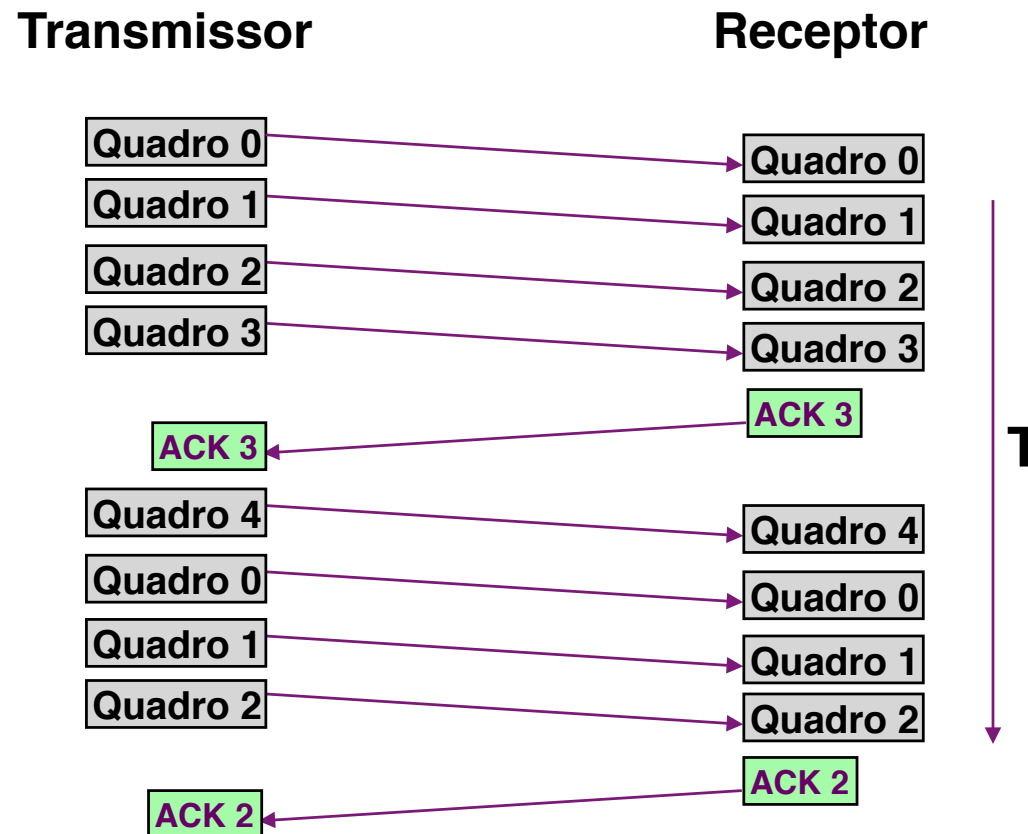
Controle de Fluxo

- ✓ **Regula o fluxo de quadros entre transmissor e receptor**
- ✓ **Resolve o problema de diferença entre velocidade de transmissão e recepção**
- ✓ **Não permite que uma estação transmissora mais rápida sobrecarregue uma estação receptora**
- ✓ **Técnicas:**
 - *Stop-and-Wait*
 - *Sliding Window*

Protocolo *Stop-and-Wait*

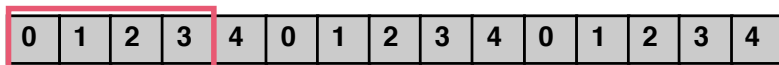


Protocolo *Sliding Window*

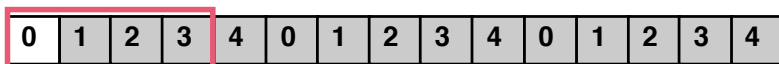


Protocolo *Sliding Window*

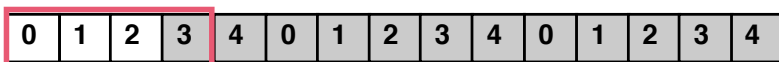
Transmissor



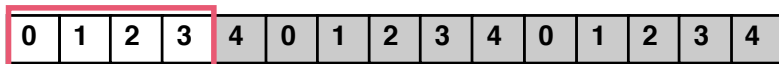
Envia um quadro



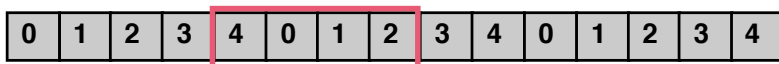
Envia dois quadros



Envia um quadro



Recebe Ack de quatro quadros



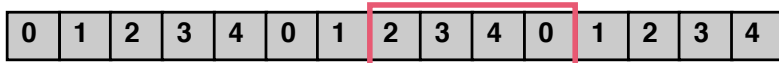
Envia um quadro



Envia dois quadros



Recebe Ack de três quadros



Receptor



Recebe um quadro



Recebe dois quadros



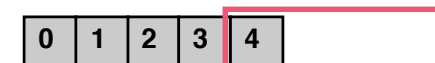
Recebe um quadro



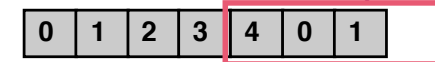
Envia Ack de quatro quadros



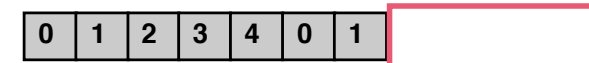
Recebe um quadro



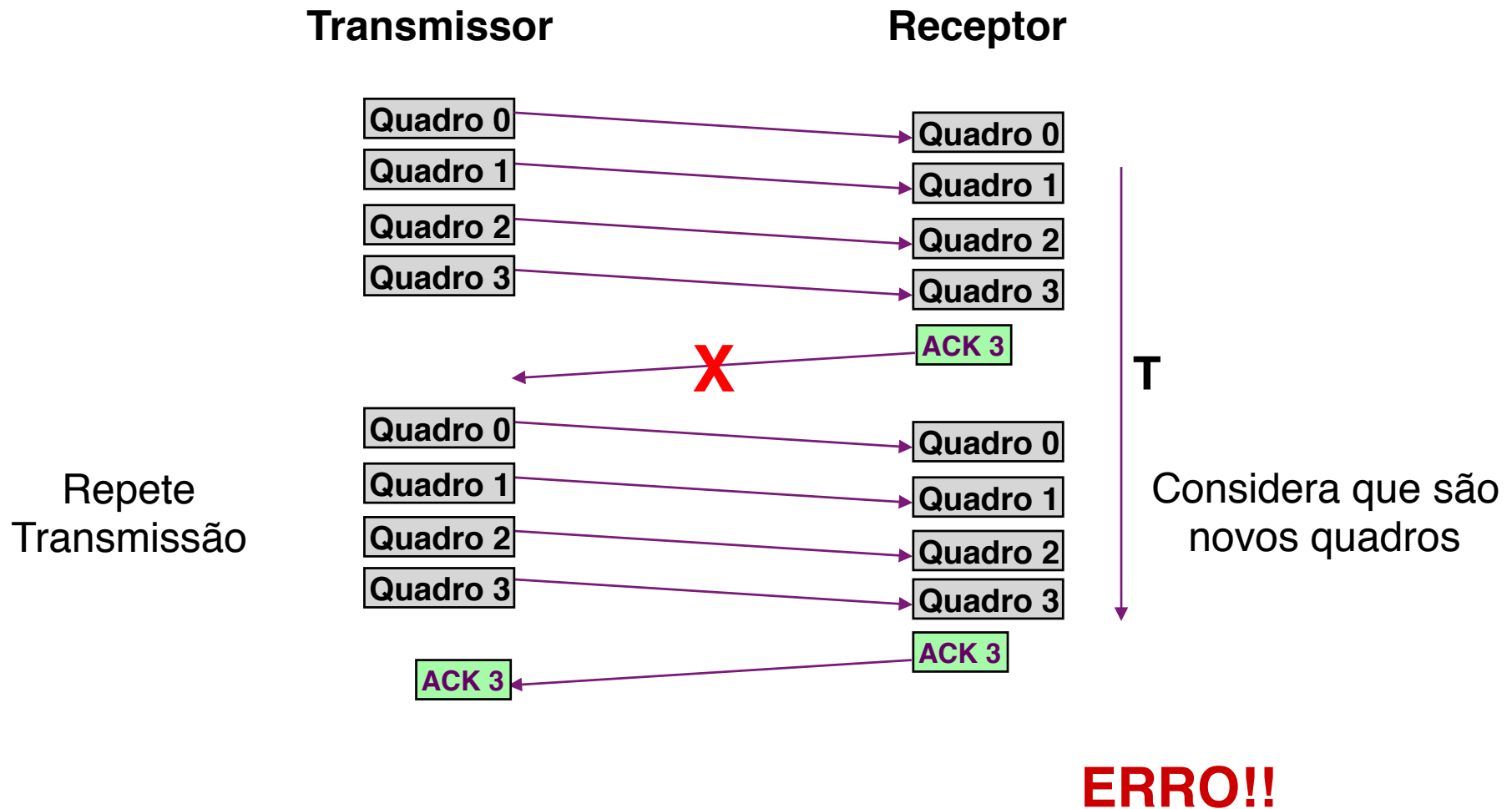
Recebe dois quadros



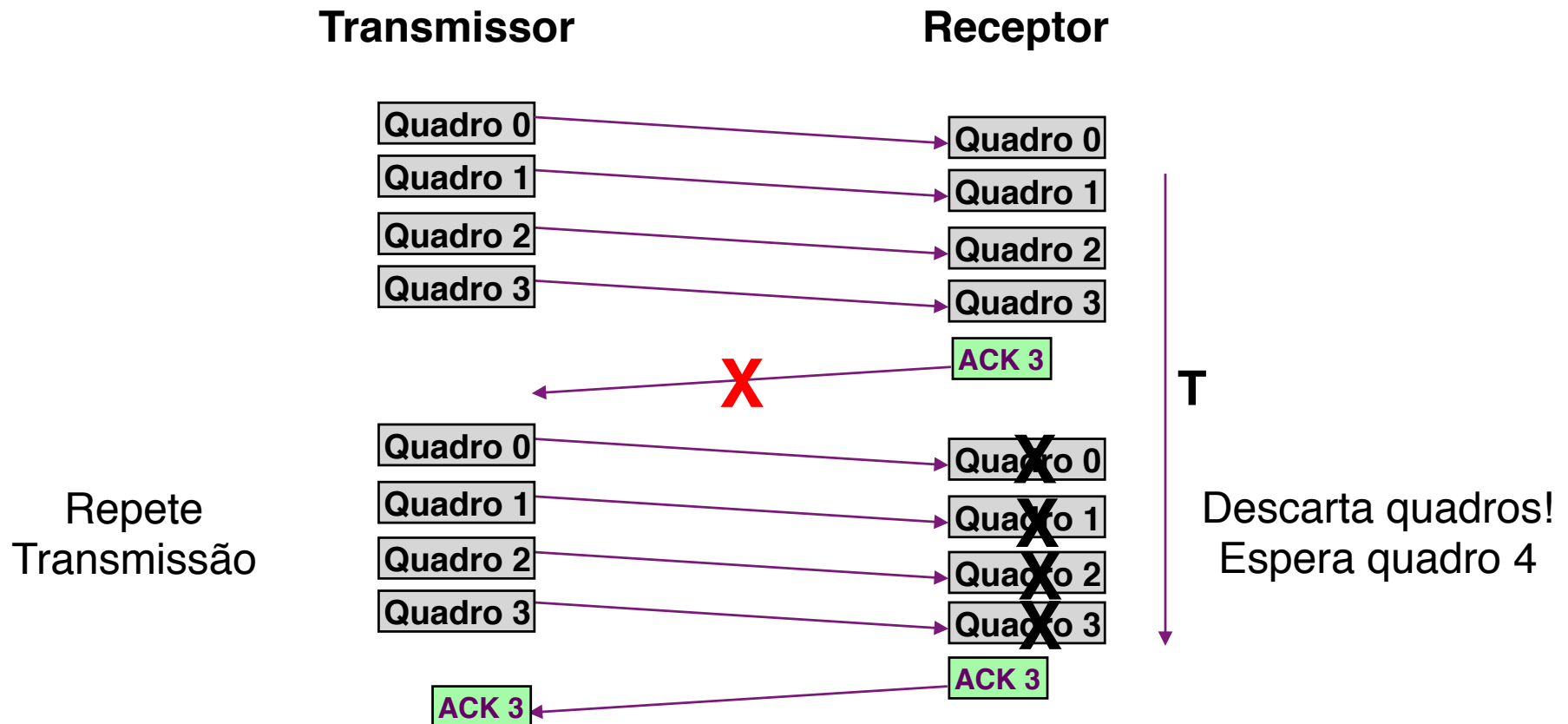
Envia Ack de três quadros



Protocolo *Sliding Window* – *Go Back N*

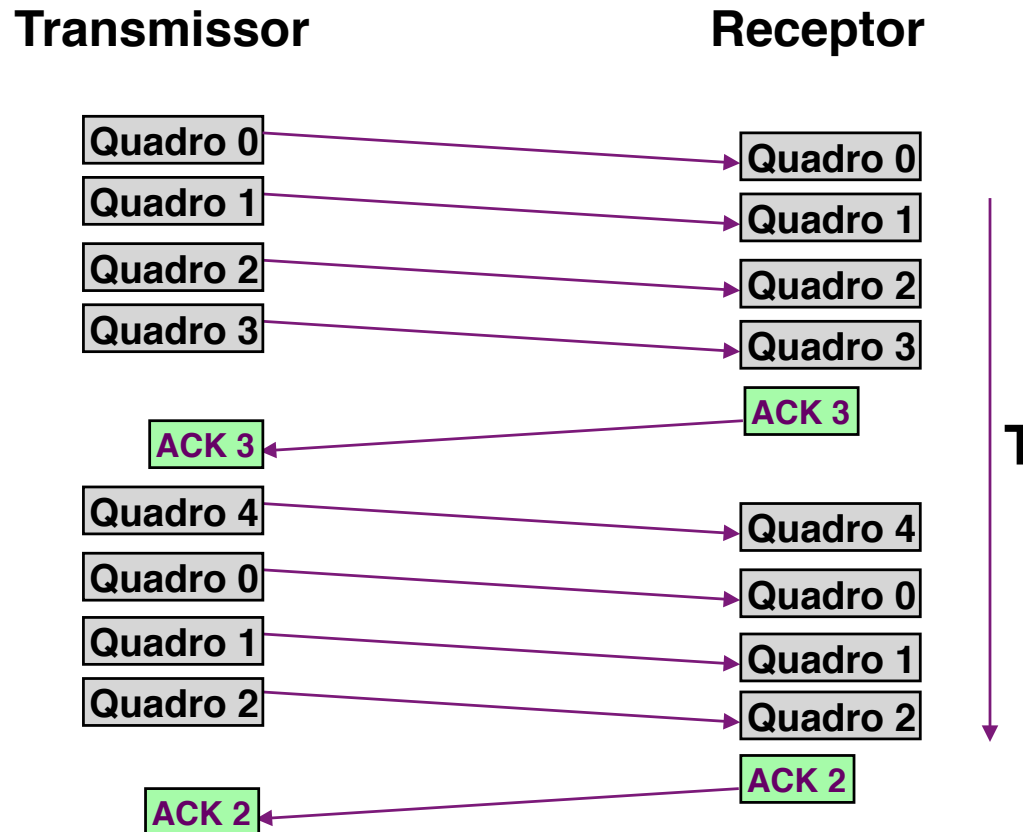


Protocolo *Sliding Window* – *Go Back N*



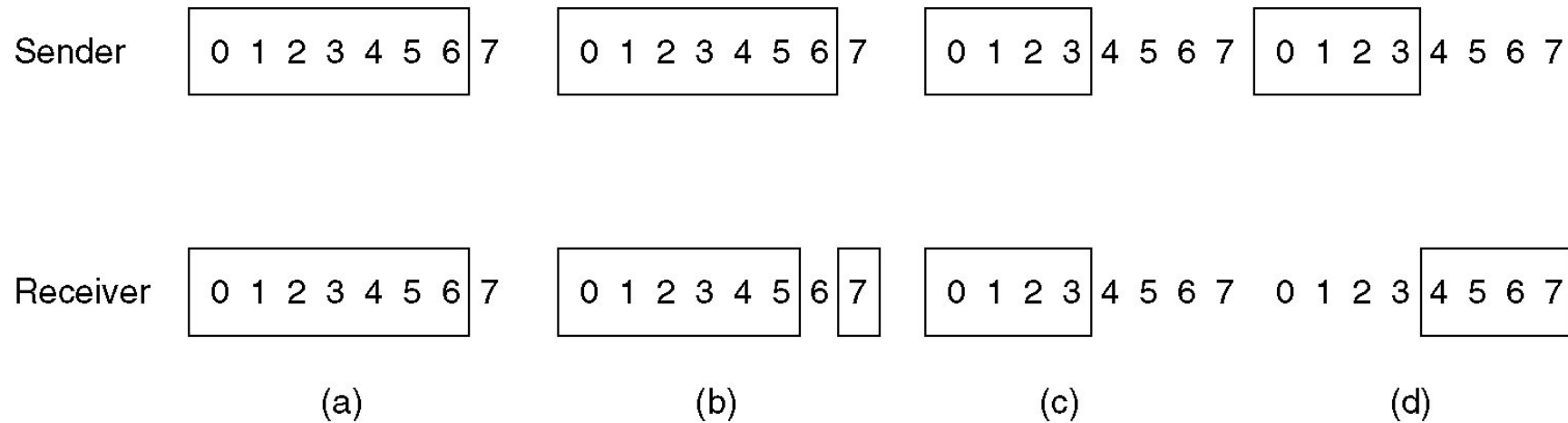
Go Back N: Se janela tem tamanho n , precisa de $(n+1)$ números de sequência

Protocolo *Sliding Window* – *Go Back N*



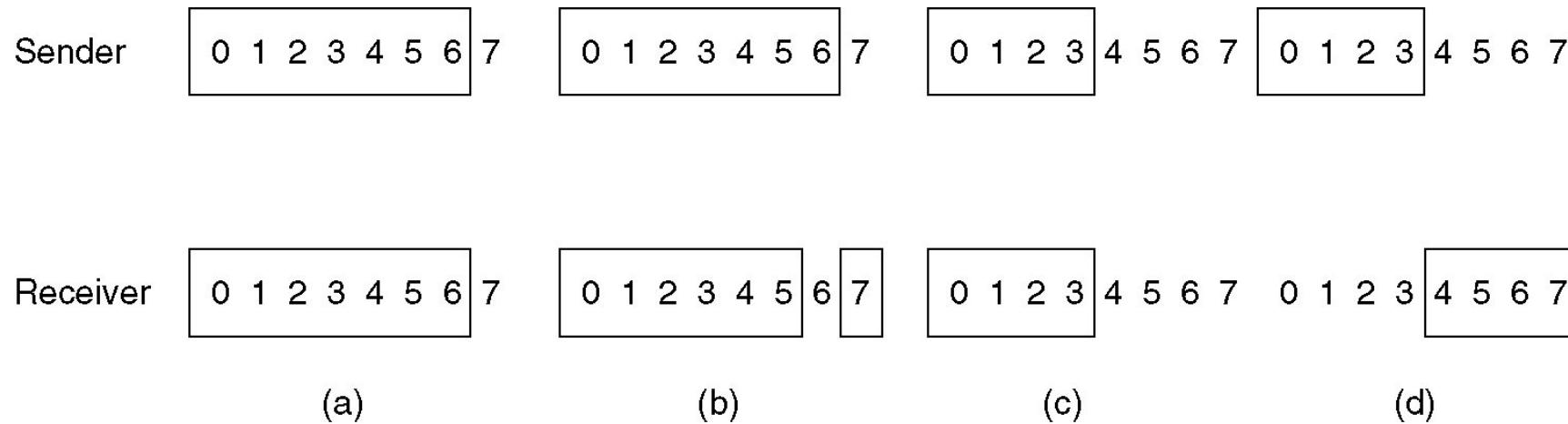
Se janela = 4, precisa de 5 números de sequência (0 a 4)

A Sliding Window Protocol Using Selective Repeat



- (a) Situação inicial com tamanho de janela = 7**
- (b) Depois de transmitir e receber 7 quadros cujos ACKs não chegaram**
- (c) Situação inicial com tamanho de janela = 4**
- (d) Depois de transmitir e receber 4 quadros cujos ACKs não chegaram**

A Sliding Window Protocol Using Selective Repeat



(c) Situação inicial com tamanho de janela = 4

(d) Depois de transmitir e receber 4 quadros cujos ACKs não chegaram – 8 números de sequência (0 a 7)

Selective Repeat: Para janela de tamanho n , precisa de $(2n)$ números de sequência

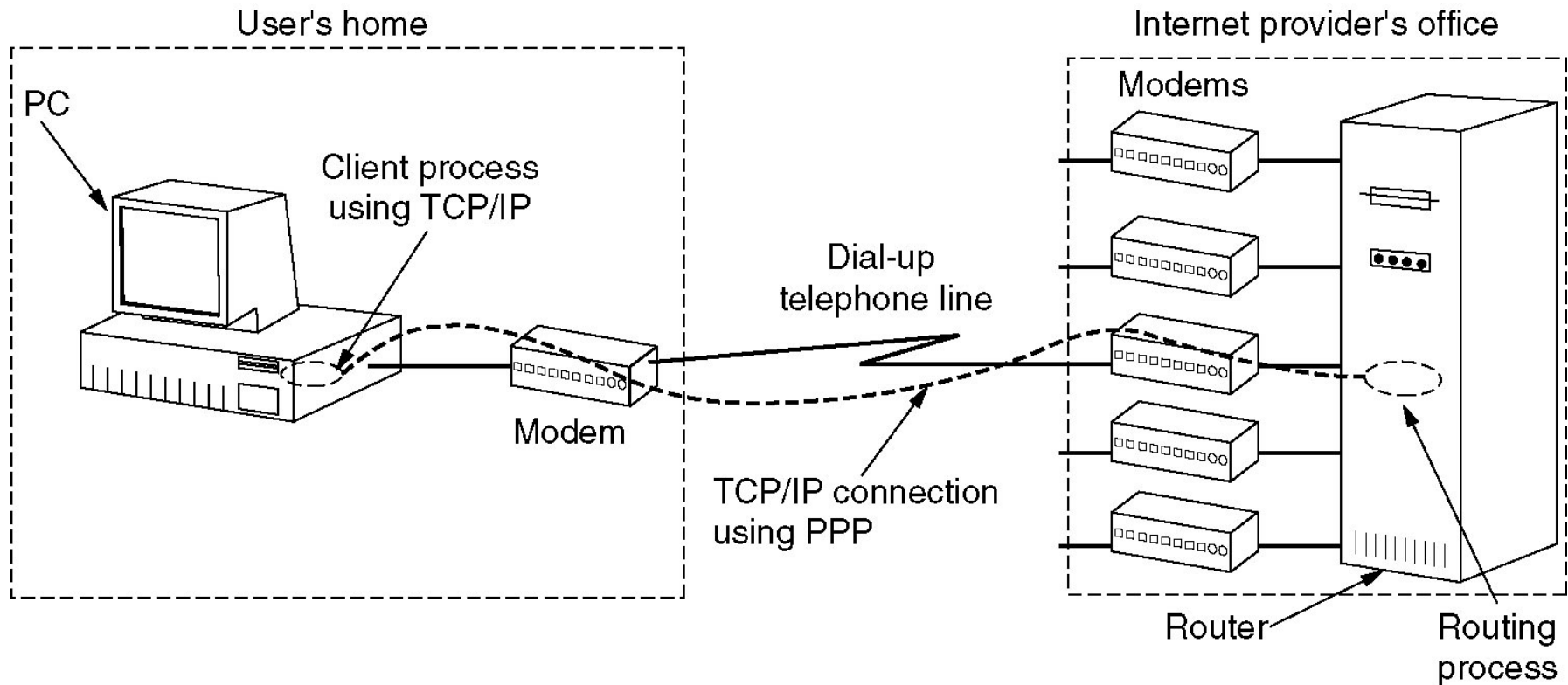
Enlace Ponto a Ponto

Protocolo PPP

Controle de Enlace Ponto-a-Ponto

- ✓ Um transmissor, um receptor, um link: mais fácil que um enlace broadcast:
 - *não há Controle de Acesso ao Meio*
 - *não há necessidade de endereçamento de enlace*
 - *ex.: enlace discado, linha ISDN*
- ✓ protocolos ponto-a-ponto populares para camada de enlace:
 - *SLIP (Serial Line IP)*
 - *PPP (Point-to-Point Protocol)*
 - *HDLC: High Level Data Link Control (A camada de enlace costumava ser considerada de alto nível na pilha de protocolos!)*

Acesso Discado usa PPP



PPP Requisitos de Projeto [RFC 1557]

- ✓ **Delimitação de quadro:** encapsulamento do pacote da camada de rede no quadro da camada de enlace
- ✓ **Múltiplos protocolos de rede:** habilidade para transportar múltiplos protocolos de rede e identificá-los no destino
- ✓ **Múltiplos tipos de enlace:** capacidade de utilização em diferentes tipos de enlace (transmissão serial/paralela, síncrona/assíncrona)
- ✓ **transparência de bits:** deve transportar qualquer padrão de bits no campo de dados
- ✓ **detecção de erros** (mas não correção)
- ✓ **gerenciamento da conexão:** detecta e informa falhas do enlace para a camada de rede
- ✓ **negociação de endereço da camada de rede:** os pontos terminais do enlace podem aprender e configurar o endereço de rede de cada outro
- ✓ **Simplicidade:** protocolo ponto a ponto deve ser simples
 - + *de 50 RFCs!*

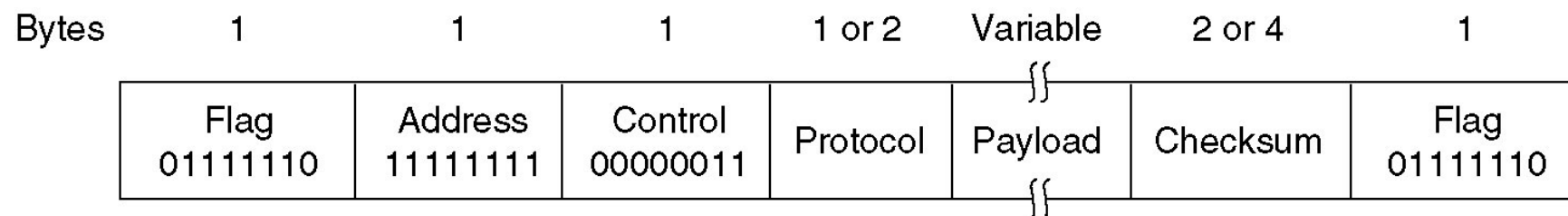
PPP não-requisitos

- ✓ não há correção nem recuperação de erros
- ✓ não há controle de fluxo
- ✓ aceita entregas fora de ordem
- ✓ não há necessidade de suportar enlaces multiponto
 - *HDLC suporta*

Recuperação de erros, controle de fluxo, re-ordenação dos dados são todos deixados para as camadas mais altas!

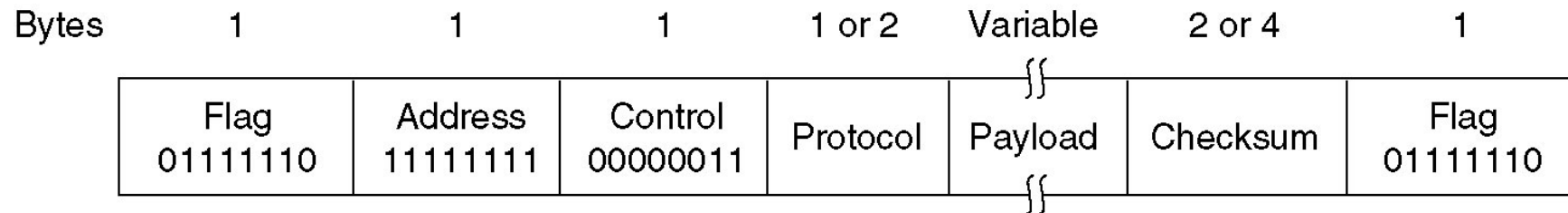
PPP Formato do Quadro

- ✓ **Flag:** delimitador
- ✓ **Endereço:** não tem função (apenas uma opção futura)
- ✓ **Controle:** não tem função; no futuro é possível ter múltiplos campos de controle
- ✓ **Protocolo:** indica o protocolo da camada superior ao qual o conteúdo do quadro deve ser entregue (ex.: IP - 21) (default = 2 bytes)



PPP Formato dos dados

- ✓ **info:** dados da camada superior sendo transportados
- ✓ **CRC:** verificação de redundância cíclica para detecção de erros (default = 2 bytes)

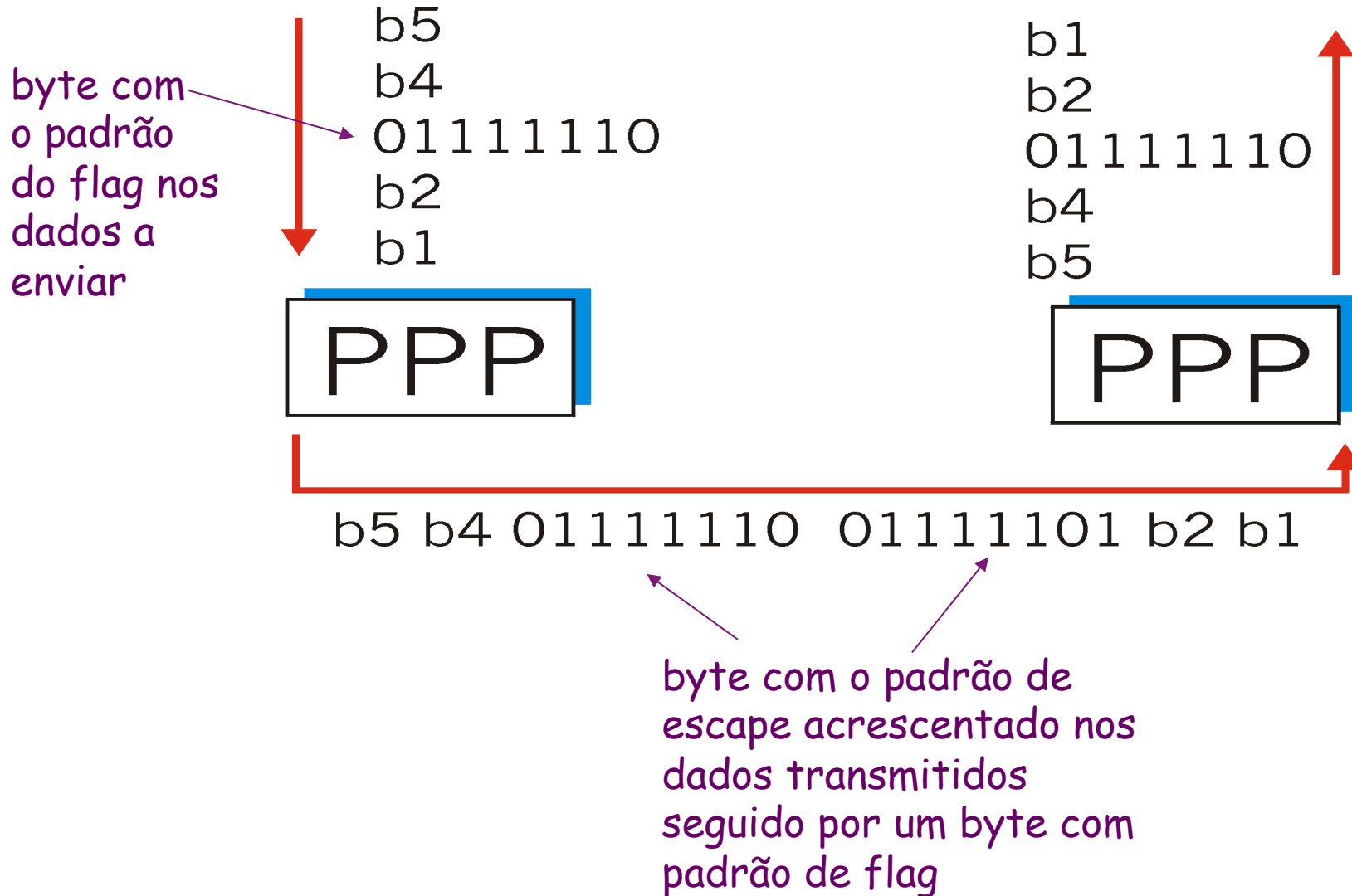


Delimitação do Quadro: Stuffing de Caracter

Redes de Computadores II

- ✓ Requisito de “transparência de dados”: o campo de dados deve poder incluir o padrão correspondente ao flag $\langle 01111110 \rangle$
 - *Q: se for recebido o padrão $\langle 01111110 \rangle$, é dado ou flag?*
- ✓ **Transmissor**: acrescenta (“stuffs”) um byte extra com o padrão $\langle 01111101 \rangle$ (escape) antes de cada byte com o padrão de flag $\langle 01111110 \rangle$ nos *dados*
- ✓ **Receptor**:
 - *um byte 01111101 seguido de 01111110 : descarta o primeiro e continua a recepção de dados*
 - *único byte 01111110 : então é um flag*

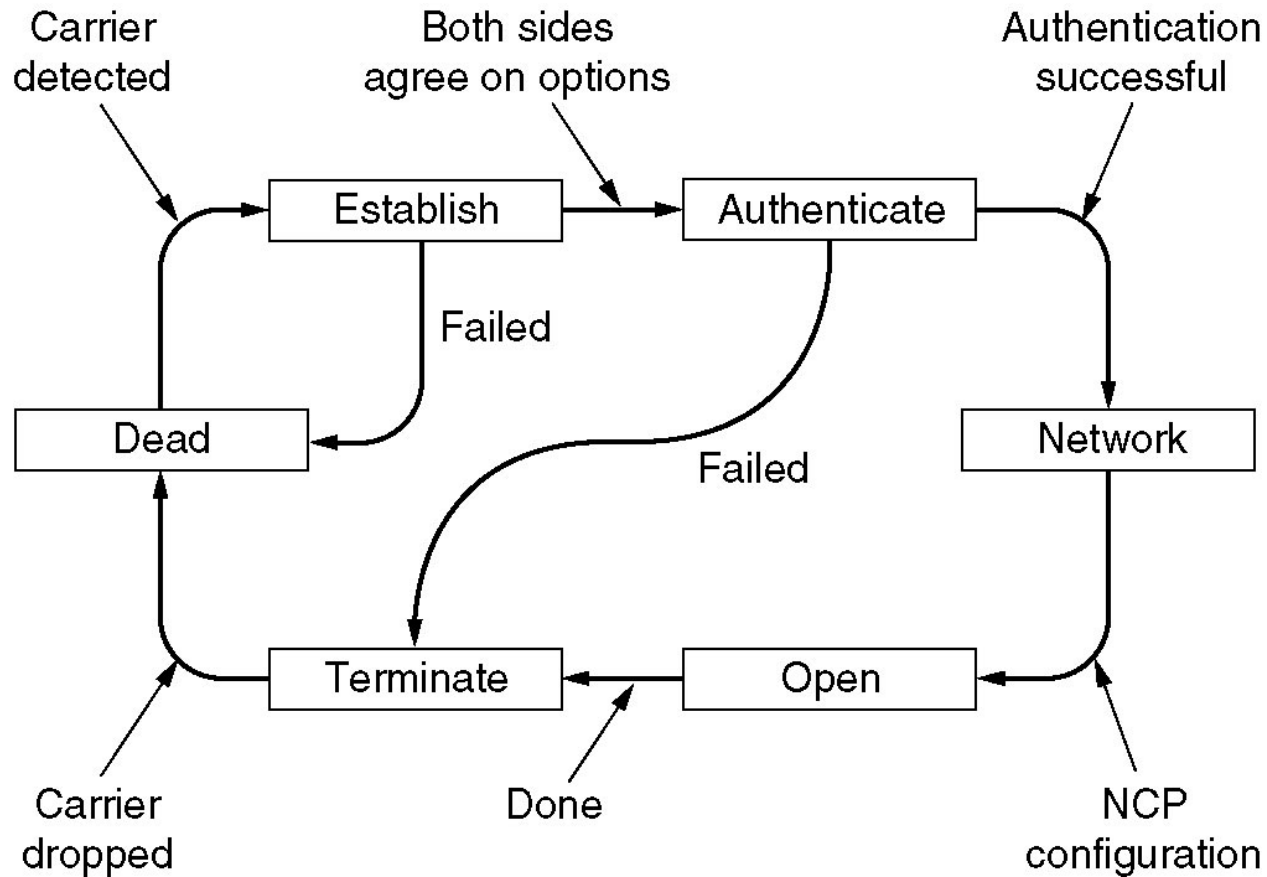
Byte Stuffing



PPP

- ✓ **Antes de trocar dados da camada de rede, os parceiros da camada de enlace devem**
 - ***configurar o enlace PPP (tamanho máximo do quadro, autenticação)***
 - Protocolo LCP – *Link Control Protocol*
 - *Encapsulado no PPP – tipo do protocolo C021*
 - ***aprender/configurar as informações da camada de rede***
 - configuração dinâmica de endereço
 - Protocolo NCP – *Network Control Protocol* – específico para cada protocolo de rede
 - *Ex.: IPCP para o protocolo IP (tipo = 8021)*

Controle do Enlace PPP



Tipos de Pacotes - Protocolo LCP

Nome	Sentido	Descrição
Configure-request	T => R	Lista de opções e valores propostos
Configure-ack	T <= R	Todas opções aceitas
Configure-nack	T <= R	Algumas opções não aceitas
Configure-reject	T <= R	Algumas opções não negociáveis
Terminate-request	T => R	Solicitação de término do enlace
Terminate-ack	T <= R	Confirmação de término
Code-reject	T <= R	Solicitação não reconhecida
Protocol-reject	T <= R	Protocolo não reconhecido
Echo-request	T => R	Por favor envie esse quadro de volta
Echo-reply	T <= R	Aqui está o quadro de volta
Discard-request	T => R	Descarte esse quadro (teste)