

CAMADA DE ENLACE

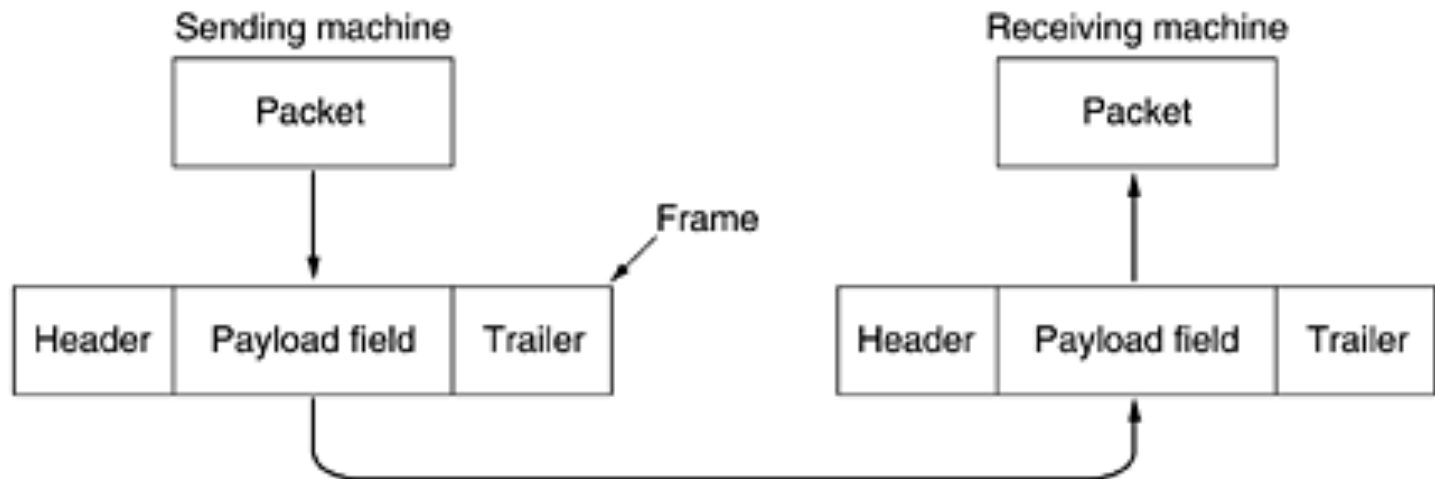
Prof. Marcelo A. Rauh Schmitt

Funções

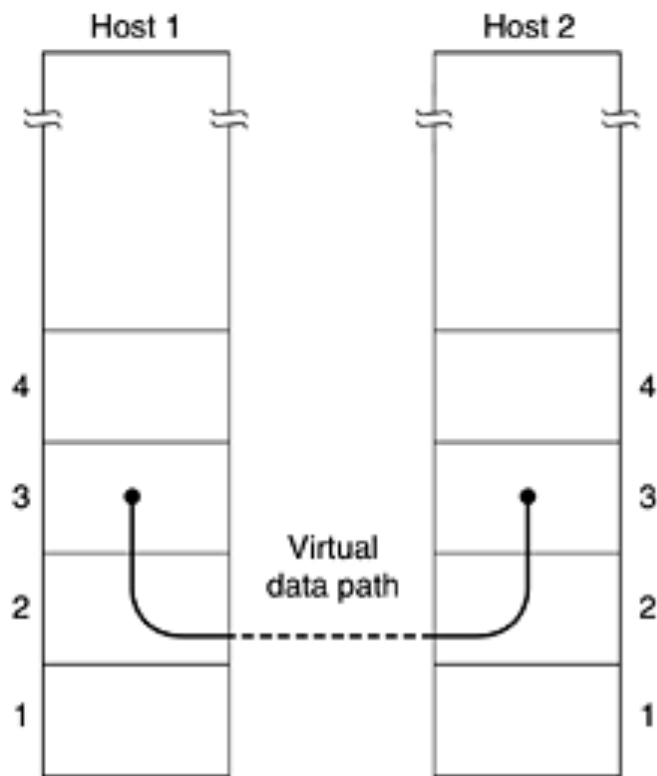
- Proporcionar uma comunicação confiável e eficiente entre duas máquinas adjacentes
 - ▣ Prover uma interface para o serviço de rede
 - ▣ Tratar os erros de transmissão
 - ▣ Regular o fluxo de dados a fim de que receptores lentos não sejam inundados por transmissores rápidos

Funcionamento geral

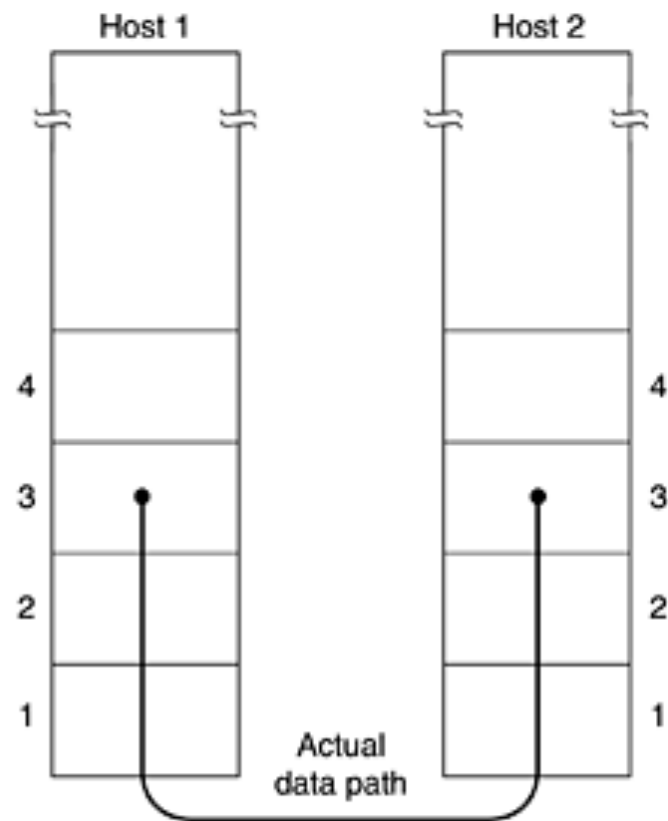
- ❑ Encapsulamento dos pacotes do nível de rede em quadros para transmissão.



Abstração!



(a)



(b)

Tipos de serviços

- Serviço sem conexão e sem confirmação
 - ▣ Quando pode ser utilizado
 - baixa taxa de erros
 - tráfego real time
 - ▣ Qualquer recuperação é deixada para as camadas superiores
 - ▣ Utilizado em LANs (redes locais)

- Serviço sem conexão e com confirmação
 - ▣ Adequado para canais menos confiáveis
 - ▣ Interessante para redes sem fio

Tipos de serviços (continuação)

- Serviço orientado à conexão e com confirmação
 - ▣ Estabelecimento de conexão antes das transmissões
 - inicialização de variáveis e contadores
 - ▣ Transmissão dos quadros
 - Numeração dos quadros
 - Garantia de entrega
 - Garantia de apenas um recebimento
 - Garantia de entrega na ordem
 - ▣ Fechamento da conexão
 - Liberação de variáveis, buffers e outros recursos

Framing

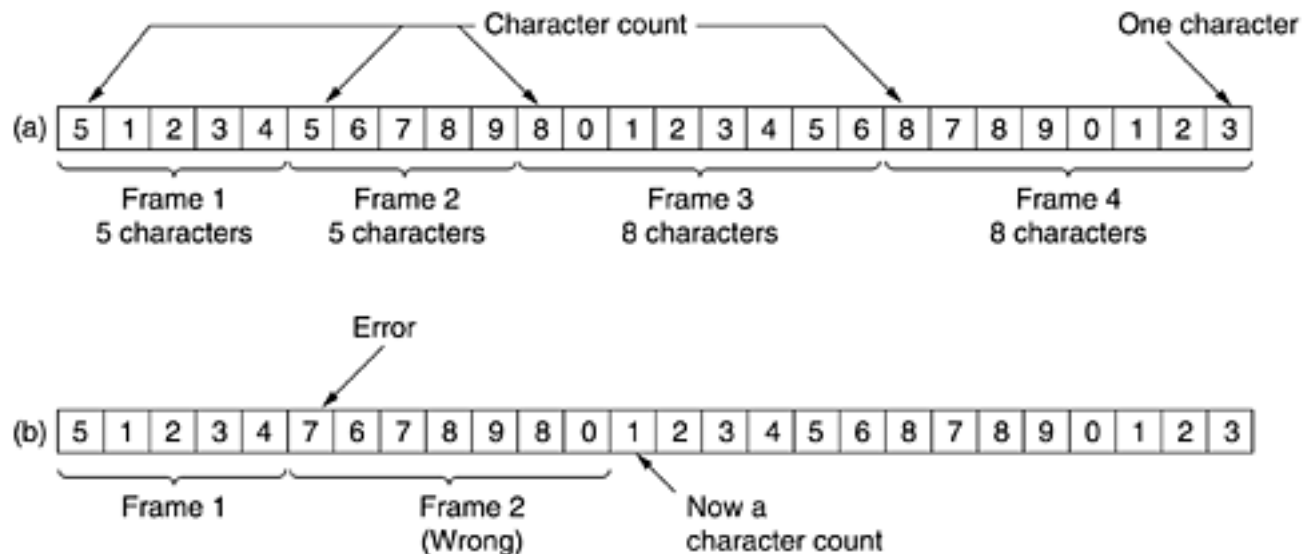
- Objetivo de montar um quadro (frame)
 - ▣ Detectar erros
 - Transmissão de bits a mais
 - Transmissão de bits a menos
 - Modificação de bits
- Cada quadro apresenta um checksum
 - ▣ O checksum é recomputado na recepção
 - ▣ O checksum recomputado é comparado com o enviado

Como marcar o início e fim do quadro

□ Contagem de caracteres

▣ Problema

- Se um erro de transmissão causar modificação no contador
- Não utilizado

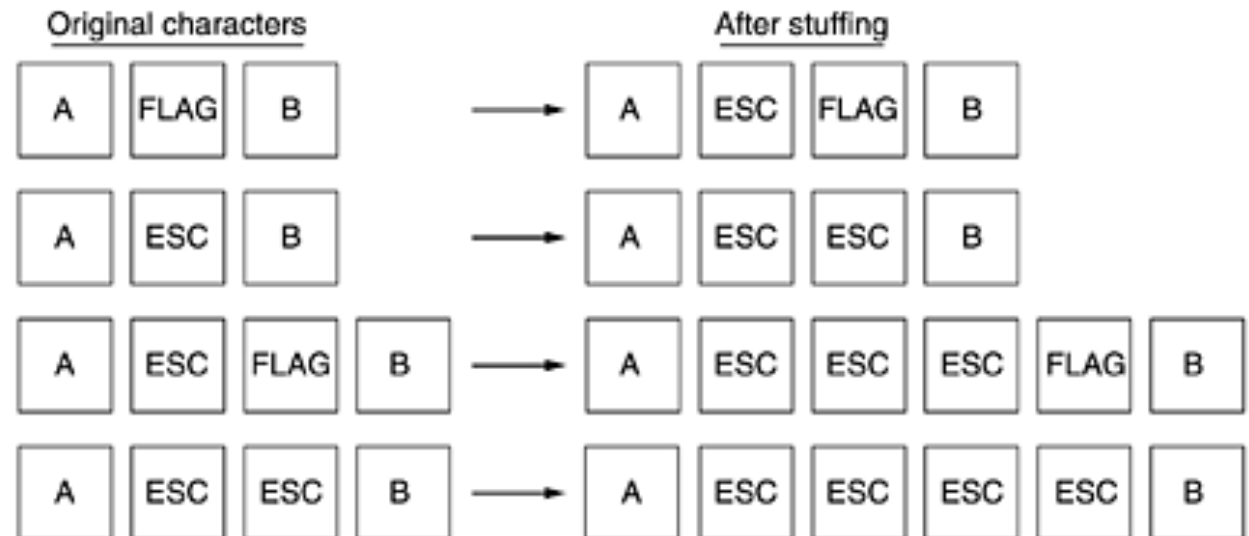
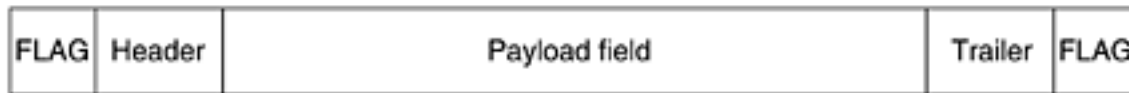


Como marcar o início e fim do quadro

□ *Byte flags com byte stuffing*

▣ Problema

- Dado igual ao flag -> escape byte
- Baseado em byte



Como marcar o início e fim do quadro

□ *Bit Flag com Bit Stuffing*

▣ Flags de início e fim com bit stuffing

(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0

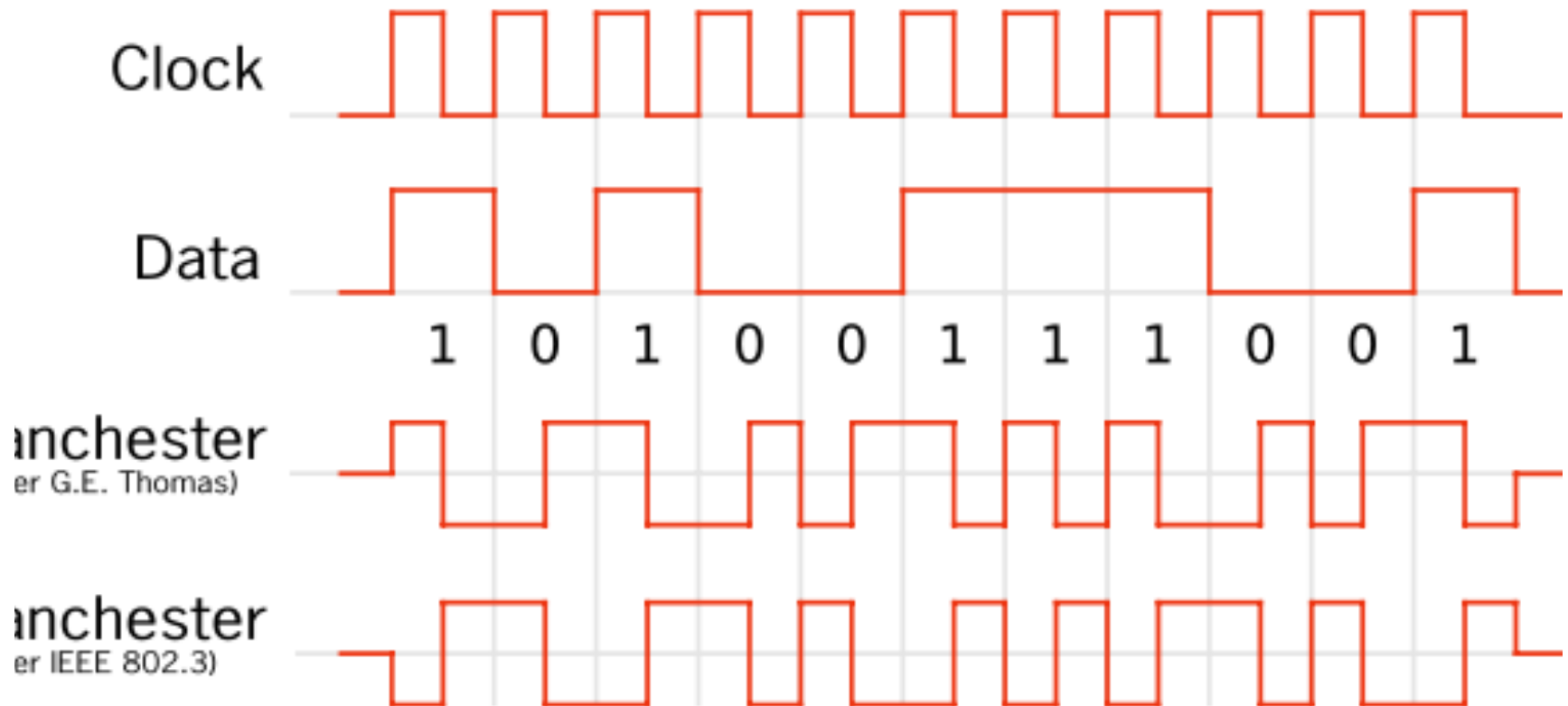
Stuffed bits

(c) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

Como marcar o início e fim do quadro

- Violação de condição física
 - ▣ Em algumas LANs, cada bit da camada de enlace é representada por dois bits da camada física (codificação Manchester)
 - ▣ $1 = 10$
 - ▣ $0 = 01$
 - ▣ Pode-se usar 11 ou 00 para delimitar frames.

Sinalização Manchester



Controle de fluxo

- O transmissor não pode ser mais rápido do que o receptor
- Métodos para o controle de fluxo
 - ▣ *feedback-based flow control*
 - o receptor autoriza o transmissor a enviar mais quadros
 - ▣ *rate-based flow control* (nunca utilizado na camada de enlace)
 - mecanismos do protocolo que limita a taxa de transmissão

Controle de fluxo tem de explicar melhor

- *Sliding Window*

- Não precisa confirmar cada quadro
- Há uma janela de transmissão que não exige confirmação

- *Piggybacking*

- Coloca a confirmação junto com o dado

Controle de erro

□ Estratégias

▣ Correção de erros

- muita redundância
- pouca necessidade de retransmissão
- canais menos confiáveis

▣ Detecção de erros

- pouca redundância
- maior necessidade de retransmissão
- canais mais confiáveis

Controle de erro

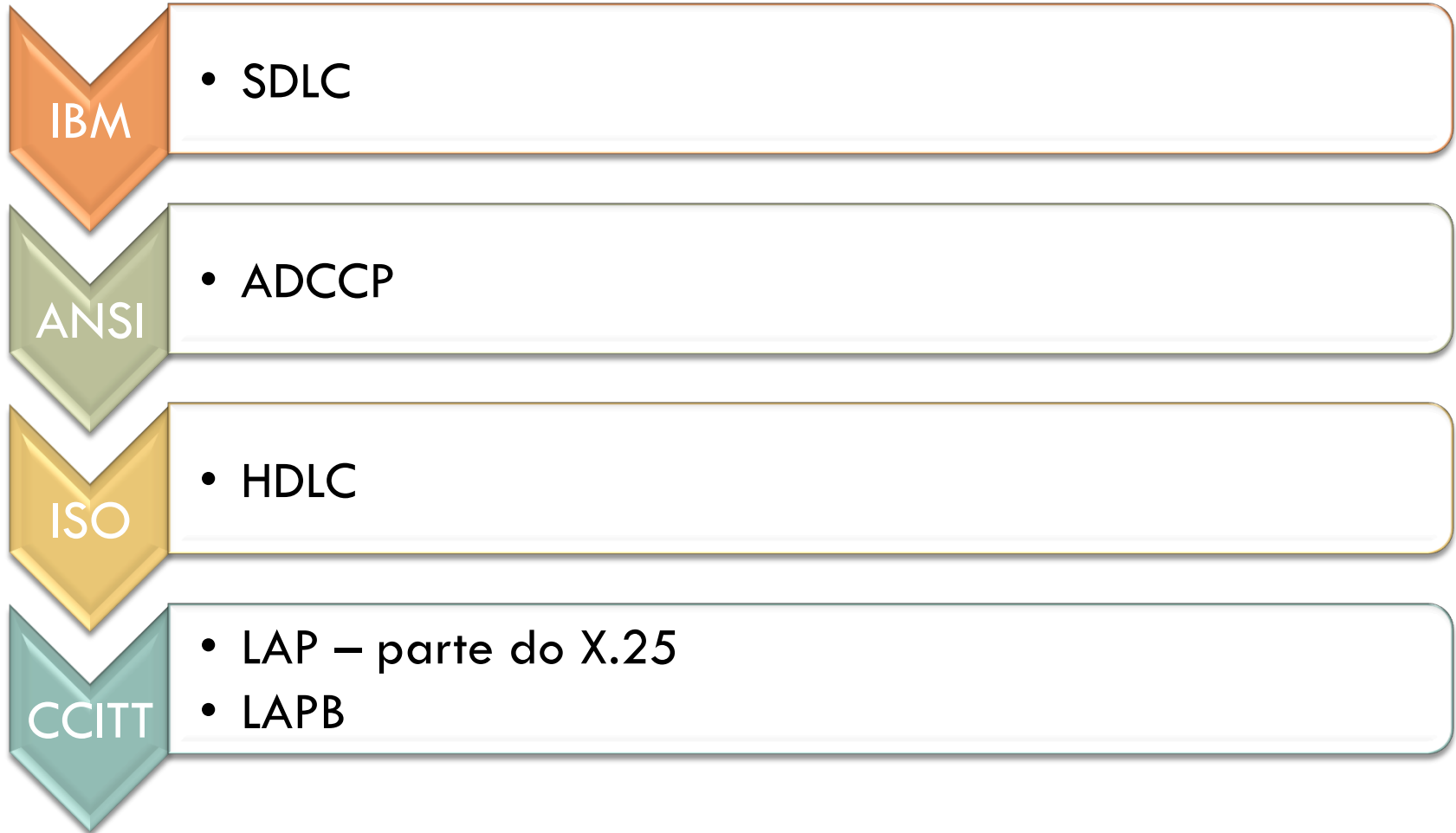
□ Exemplo – bit de paridade

Original Data	Even Parity	Odd Parity
0 0 0 0 0 0 0 0	0	1
0 1 0 1 1 0 1 1	1	0
0 1 0 1 0 1 0 1	0	1
1 1 1 1 1 1 1 1	0	1
1 0 0 0 0 0 0 0	1	0
0 1 0 0 1 0 0 1	1	0

Controle de erro

- CRC – Cyclic Redundancy Check
 - ▣ Simples de implementar em hardware
 - ▣ Bom para detectar erros
 - ▣ Inventado por W. Wesley Perterson (EUA – 1961)
 - ▣ Muito usado em comunicação de dados

HDLC - *High-Level Data Link Control*

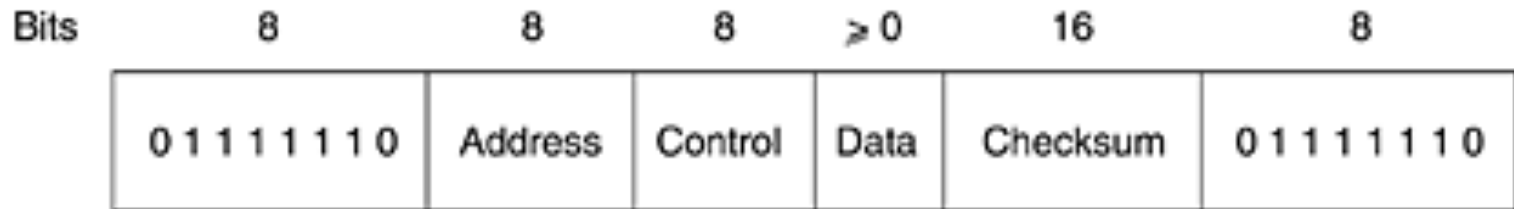


HDLC - *High-Level Data Link Control*

- ▣ Orientado a bit
- ▣ Utiliza *bit stuffing*
- ▣ Utiliza *Sliding window* com número de sequência de 3 bits
 - Até 7 quadros podem estar sem confirmação
- ▣ *Piggyback*
 - número do quadro que ainda não veio

HDLC

- Formato geral do quadro dos protocolos orientados a bit



- *Control*

- Número de sequência
- Número de confirmação (*acknowledgement*)
- Outras funções

- Checksum - CRC

HDLC

□ Tipos de quadros

▣ Informação

▣ Supervisão

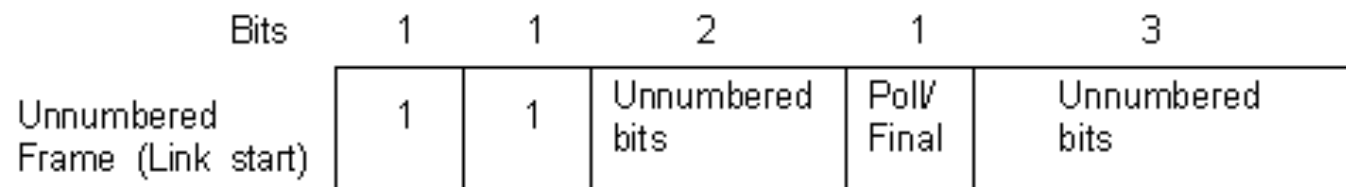
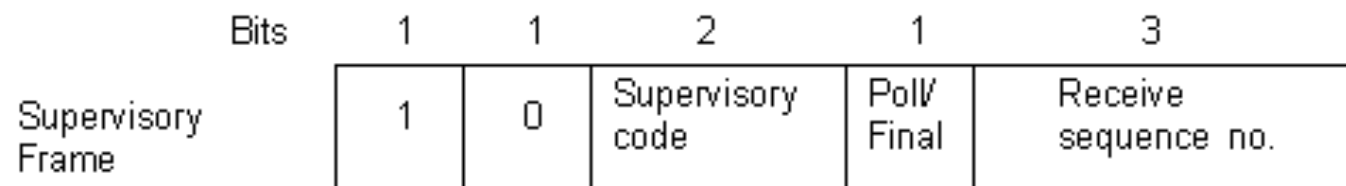
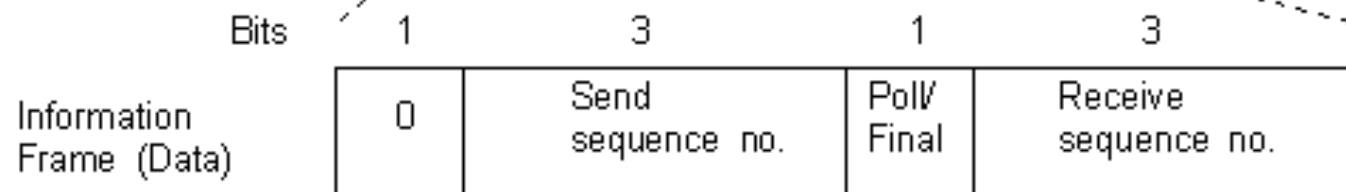
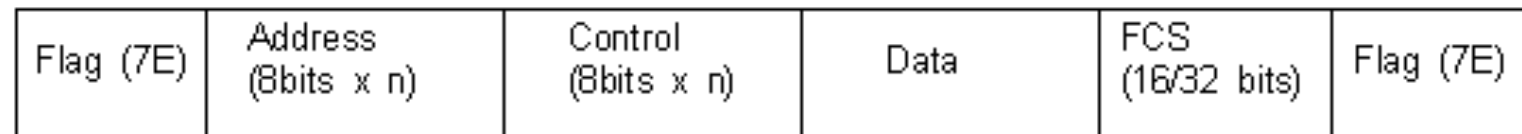
- Tipo 0 – confirmação (RECEIVE READY)
- Tipo 1 – não confirmação (REJECT)
- Tipo 2 – confirma e pede para não mandar (RECEIVE NOT READY)
- Tipo 3 – não confirmação seletiva (SELECTIVE REJECT)

▣ Não numerado

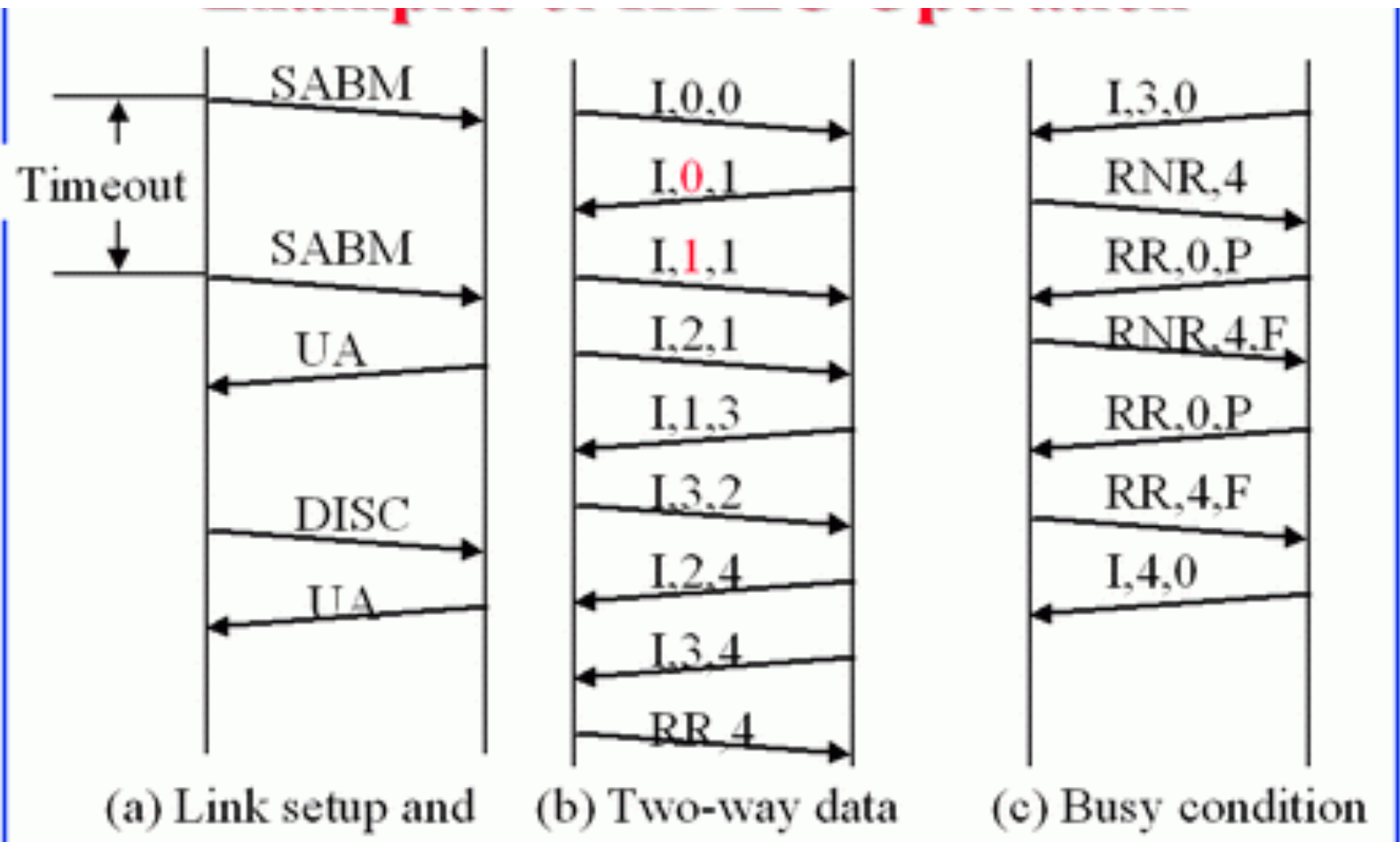
■ Controle

- Anuncia que vai desligar, por exemplo
- Anuncia que voltou

HDLC



HDLC - esempio

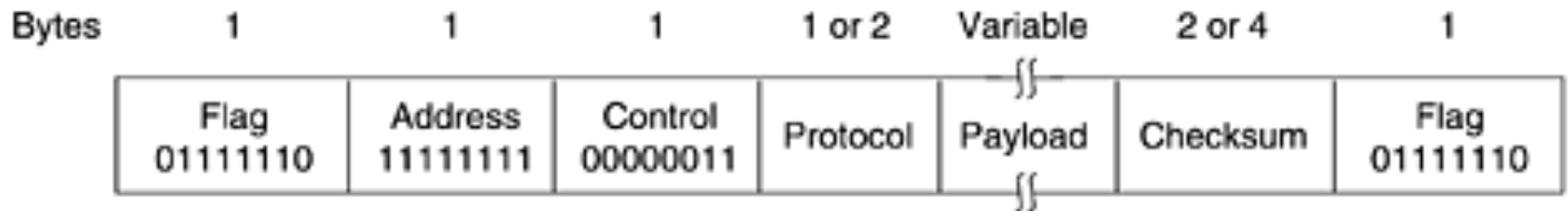


PPP – Point-to-Point Protocol

- ❑ RFC 1661 e outras
- ❑ Orientado a caractere
- ❑ Utiliza *byte stuffing*
- ❑ *Unnumbered frame* – opcionalmente pode-se usar um modo confiável
- ❑ LCP – *Link Control Protocol* – estabelecimento de conexão
- ❑ NCP – *Network Control Protocol* – negociação de parâmetros de rede

PPP

- Uso atuais do PPP em linhas ADSL
 - ▣ PPPoE – PPP over Ethernet
 - ▣ PPPoA – PPP over ATM



Outros protocolos

- ATM
- Frame Relay
- MPLS