

Redes de Computadores

Camada de Enlace

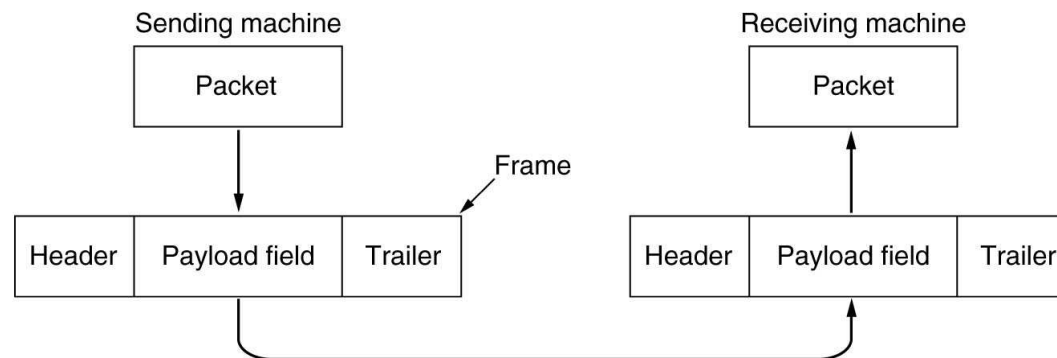
Luiz Filipe Menezes Vieira
lfvieira@dcc.ufmg.br

Departamento de Ciência da Computação
Universidade Federal de Minas Gerais



Camada de enlace

- Responsável pela comunicação confiável e eficiente entre dois computadores adjacentes
 - ◆ Adjacente significa que dois computadores estão fisicamente ligados por um canal de comunicação FIFO (*first-in-first-out*), ou seja, que preserva a ordem que os bits foram enviados



Camada de enlace

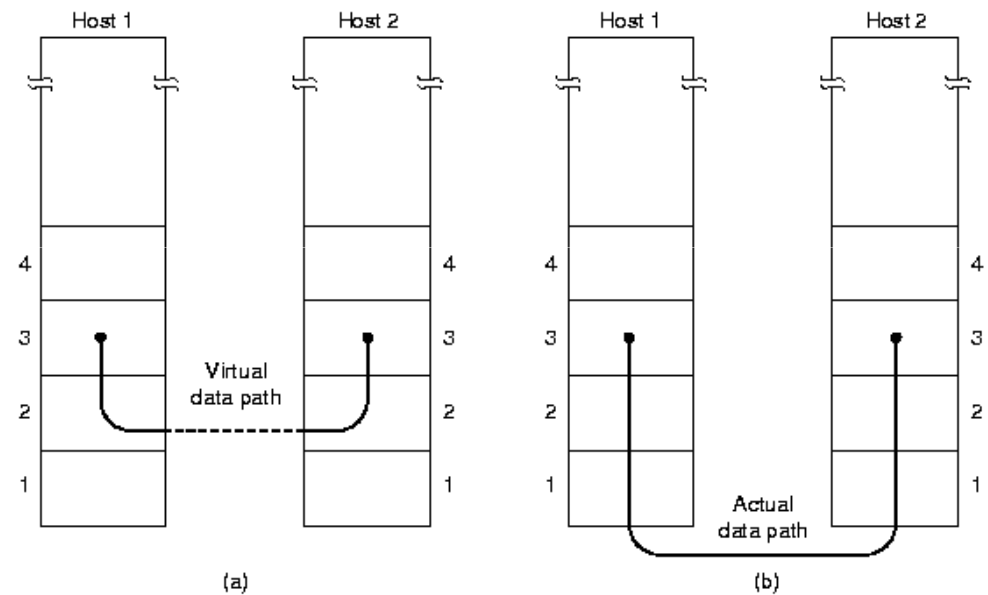
- O que torna interessante o estudo de protocolos desta camada é o “ambiente” com o qual eles interagem:
 - ◆ Canais podem introduzir erros
 - ◆ Permitem uma taxa máxima de transferência
 - ◆ Possuem um tempo de propagação diferente de zero

Questões de projeto relacionadas com a camada de enlace

- Serviços oferecidos para a camada de rede
- Formas de agrupar os bits da camada física em quadros
- Tratamento de erros de transmissão
- Controle de fluxo

Serviços oferecidos para a camada de rede

- Serviço principal:
 - ◆ Transferir dados entre as camadas de rede origem e destino



(a) Virtual communication. (b) Actual communication.

Serviços oferecidos para a camada de rede

- Os três principais tipos de serviços são:
 - ◆ Sem conexão não confirmado
 - ◆ Sem conexão confirmado
 - ◆ Com conexão confirmado

Serviço sem conexão não confirmado

- Conexão não é estabelecida a priori
- Quadros independentes são enviados da origem para o destino que não envia nenhuma confirmação de volta
- Quadros perdidos são ignorados e tratados pelas camadas superiores

Serviço sem conexão não confirmado

- Classe de serviço apropriada para
 - ◆ Baixa taxa de erro
 - ◆ Tráfego de tempo real como voz
- Serviço normalmente usado em LANs

Serviço sem conexão confirmado

- Conexão não é estabelecida a priori
- Quadros enviados pela origem são confirmados pelo destino
- Origem usa um mecanismo de temporização para reenviar quadros não confirmados
- Serviço apropriado para canais não confiáveis como comunicação sem fio

Serviço com conexão confirmado

- Serviço mais sofisticado
- É necessário estabelecer uma conexão antes de transferir dados
- Quadros são recebidos corretamente
- Camada de enlace pode entregar os quadros em ordem para a camada de rede

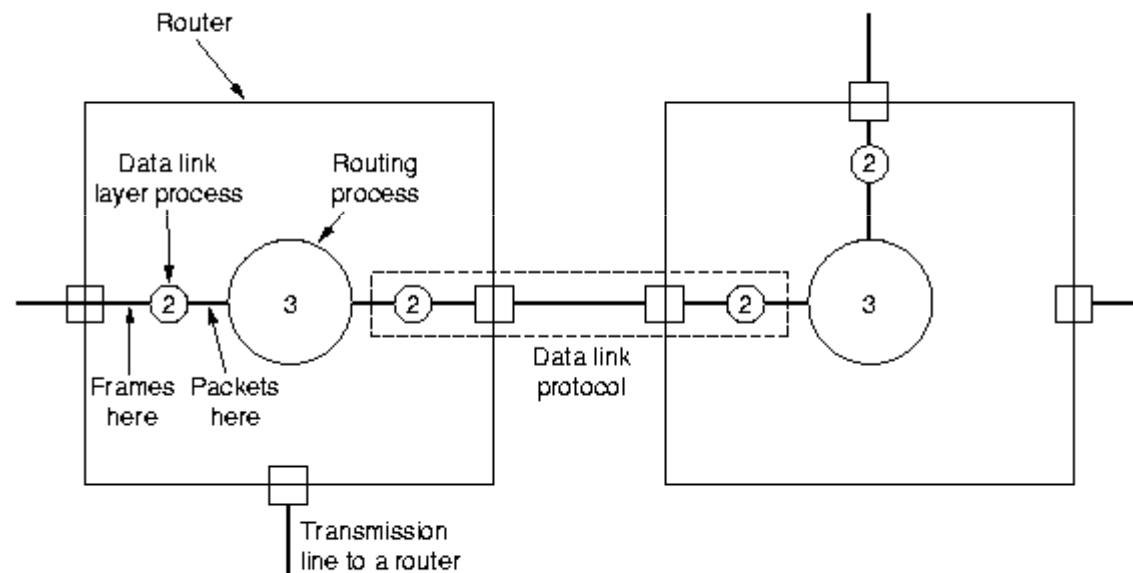
Serviço com conexão confirmado

- O serviço oferecido para a camada de rede é de uma seqüência de bits corretos
- O serviço passa por três fases
 - ◆ Estabelecimento da conexão
 - ◆ Transferência de dados
 - ◆ Término da conexão

Comentários sobre os diferentes tipos de serviços

- Confirmação na camada de enlace é uma otimização e não um requisito
 - ◆ Pode ser deixada para a camada de transporte (camada fim-a-fim)
 - ◆ Se existe na camada de enlace ou não depende do meio de comunicação
- O serviço a ser oferecido para a camada de rede depende, dentre outros fatores, da aplicação que utilizará esse serviço

A camada de enlace na sub-rede de comunicação



Placement of the data link protocol.

Framing

- Problema a ser resolvido:
 - ◆ Como agrupar seqüências de bits em quadros para que possam ser processados como unidades de informação?

ou, de outra forma,

- ◆ Como fazer delimitação de quadros?

Framing: Soluções

- Inserir intervalos de tempo entre transmissões de quadro
 - ◆ Em comunicação assíncrona não é viável
- Fazer contagem de caracteres

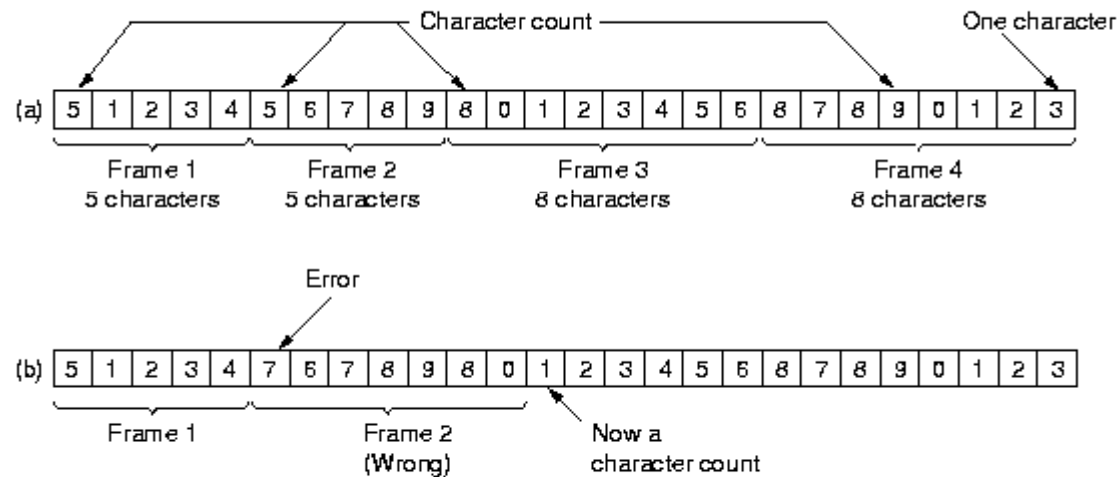
Framing: Soluções

- Inserir caracteres de início e fim de quadro com preenchimento (*stuffing*) de caracteres
- Inserir *flags* (seqüências especiais de bits) de início e fim de quadro com preenchimento de bits
- Usar violações de código da camada física

Contagem de caracteres

- Princípio:
 - ◆ Usa um campo no cabeçalho para especificar o número de caracteres no quadro
- Problema: erro nesse campo faz com que o receptor perca a sincronização
- Não é usado na prática para protocolos da camada de enlace

Contagem de caracteres

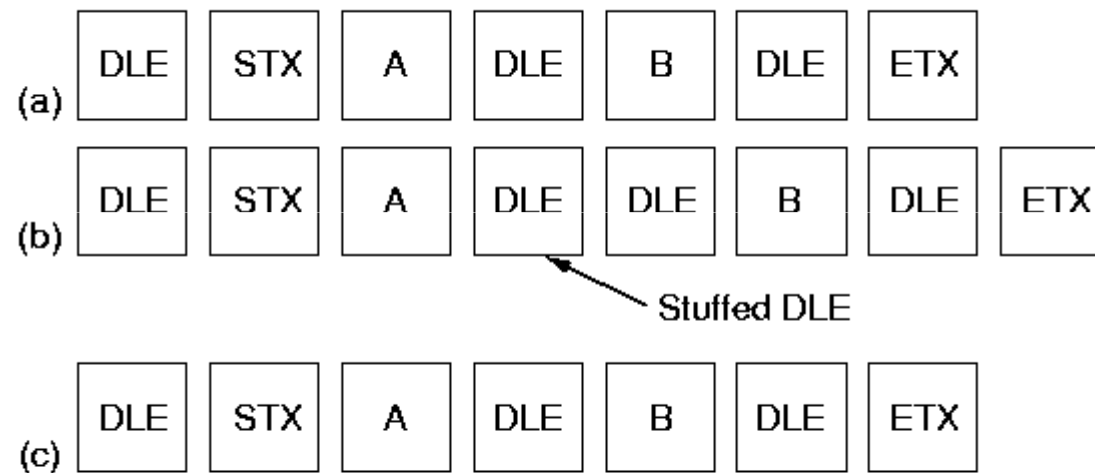


A character stream. (a) Without errors. (b) With one error.

Caracteres de início e fim de quadro

- Quadro é delimitado por caracteres especiais: DLE STX e DLE ETX
- Um DLE no meio de um quadro é prefixado por outro DLE (*character stuffing*) para distinguir do fim de quadro
- Método usado em protocolos orientados a caracteres

Caracteres de início e fim de quadro



(a) Data sent by the network layer. (b) Data after being character stuffed by the data link layer. (c) Data passed to the network layer on the receiving side.

Flags de início e fim de quadro

- Permite enviar um número arbitrário de bits
- Quadros são delimitados por uma seqüência especial de bits (*flag*) que possui o seguinte padrão:
 - ➡ 01111110

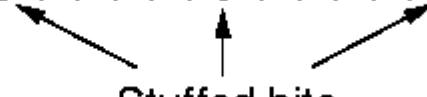
Flags de início e fim de quadro

- Bits são transmitidos de forma transparente:
 - ◆ TX ao encontrar cinco bits 1 consecutivos insere um bit 0
 - ◆ RX ao receber cinco bits 1 seguido de um bit 0 remove o bit 0
 - ▶ Processo conhecido como *bit stuffing*

Flags de início e fim de quadro

(a) 0 1 1 0 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 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

Bit stuffing. (a) The original data. (b) The data as they appear on the line. (c) The data as they are stored in the receiver's memory after destuffing.

Violações de código da camada física

- Método é baseado numa característica da camada inferior
- Existem códigos de transmissão que possuem uma transição no meio do período de transmissão de um bit
- O início e fim de quadro são determinados por um código de transmissão inválido
 - ◆ Usado no padrão IEEE 802

Controle de erro

- Objetivo (serviço):
 - ◆ Entregar em ordem e sem repetição os dados recebidos da camada física para a camada de rede

- Mecanismos utilizados para oferecer esse tipo de serviço (regras de procedimento):
 - ◆ Confirmação positiva e negativa de quadros pelo destinatário
 - ◆ Temporização de quadros enviados pela origem
 - ◆ Número de seqüência de quadros
 - ◆ Retransmissão de quadros um número finito de vezes

Controle de fluxo

- Objetivo:

- ◆ Evitar que TX envie mais quadros que a capacidade de processamento de RX

- Idéia geral:

- ◆ Usar algum mecanismo de realimentação para que o TX saiba sobre o estado do RX
- ◆ Normalmente existem regras que usam um mecanismo explícito ou implícito para fazer o controle de fluxo

Comentários sobre as questões de projeto

- Várias dessas questões se repetem em outras camadas
- A solução a ser adotada para cada questão depende da camada, protocolo e aplicação
- Essas questões são consideradas fundamentais no projeto de qualquer protocolo

Confirmação na carona: *Piggybacking*

- Seja o seguinte protocolo ponto-a-ponto entre entidades A e B:
 - ◆ Usa confirmação
 - ◆ A transmissão de dados é *full-duplex*
 - ◆ É possível embutir numa PDU de dados enviada de B para A a confirmação de uma PDU de dados enviada de A para B já recebida (o mesmo para o caso contrário)
 - ◆ Isto é conhecido como confirmação na carona ou *Piggybacking*

Confirmação na carona: *Piggybacking*

- Melhor utilização do canal
- Utiliza apenas alguns bits ao contrário de uma PDU de controle
- Menos PDUs a processar
- Possivelmente menos *buffers* no RX
- Se não há uma PDU para ser enviada de B → A, quanto tempo deve-se esperar para confirmar uma PDU já enviada e recebida de A → B?

Protocolos de janela deslizante

■ Princípio:

- ◆ Cada PDU tem um campo de número de seqüência de n bits
- ◆ O transmissor mantém um conjunto de números de seqüência que pode enviar
 - ▶ Janela de transmissão (*sending window*)
- ◆ O receptor mantém um conjunto de números de seqüência que pode receber
 - ▶ Janela de recepção (*receiving window*)

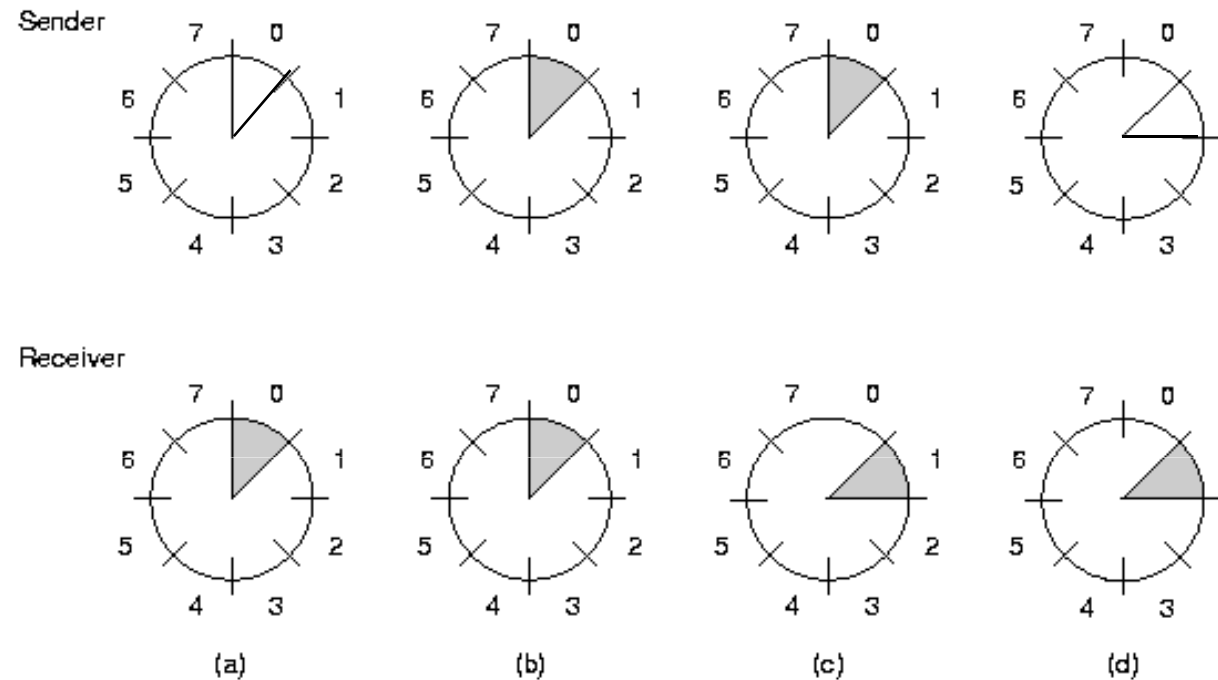
Protocolos de janela deslizante

- Os tamanhos das janelas de transmissão e recepção não precisam ser os mesmos
- Números dentro da janela de transmissão representam PDUs
 - ◆ Enviadas mas não confirmadas, ou
 - ◆ Não transmitidas ainda

Protocolos de janela deslizante

- PDUs transmitidas e não confirmadas devem ser mantidas em *buffers*
- Deve haver um temporizador associado a PDU transmitida a mais tempo
- Números dentro da janela de recepção representam PDUs que podem ser aceitas
- Confirmação:
 - ◆ PDUs aceitas são confirmadas na carona de uma PDU de dados (*piggybacking*) ou por uma PDU de controle

Protocolos de janela deslizante



A sliding window of size 1, with a 3-bit sequence number. (a) Initially. (b) After the first frame has been sent. (c) After the first frame has been received. (d) After the first acknowledgement has been received.

Comentários sobre o serviço oferecido

- A camada $n+1$ na máquina destino deve receber os dados na mesma ordem em que foram passados para a camada n na máquina origem
 - ◆ Isso pode ser obtido através do número de seqüência
 - ◆ Neste caso, a camada n é a camada de enlace

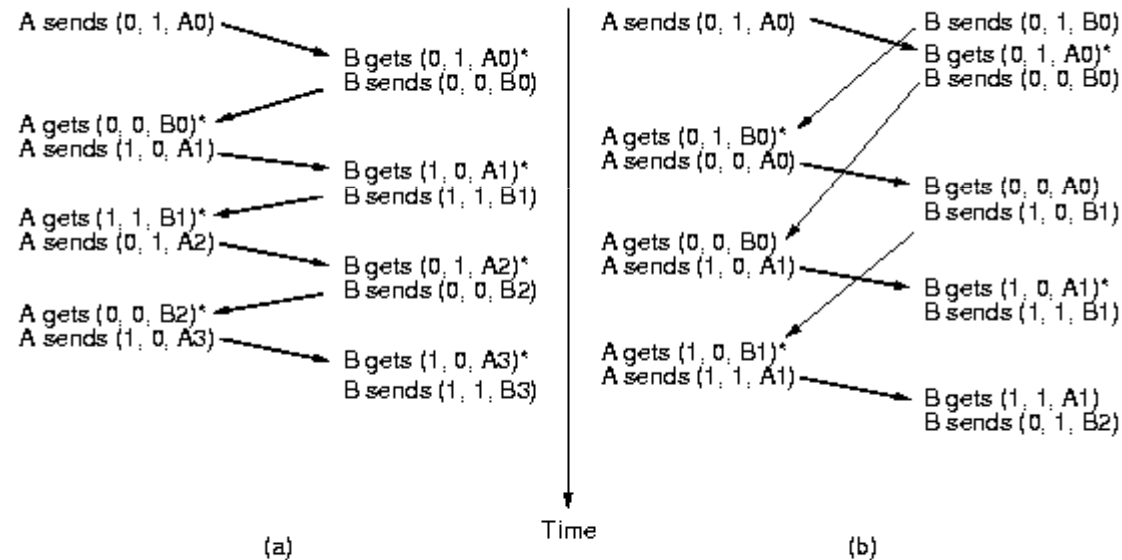
Comentários sobre o serviço oferecido

- A camada física implementa um canal de comunicação FIFO (*first-in-first-out*), ou seja, preserva a ordem das PDUs transmitidas
 - ◆ Isto poderia ser outra forma de oferecer o serviço acima

Colisão em protocolos

- Duas entidades enviam dados “simultaneamente”, cada uma para a outra
- Não é um erro do protocolo mas afeta o seu desempenho

Colisão em protocolos



Two scenarios for protocol 4. The notation is (seq, ack, packet number). An asterisk indicates where a network layer accepts a packet.

Stop-And-Wait

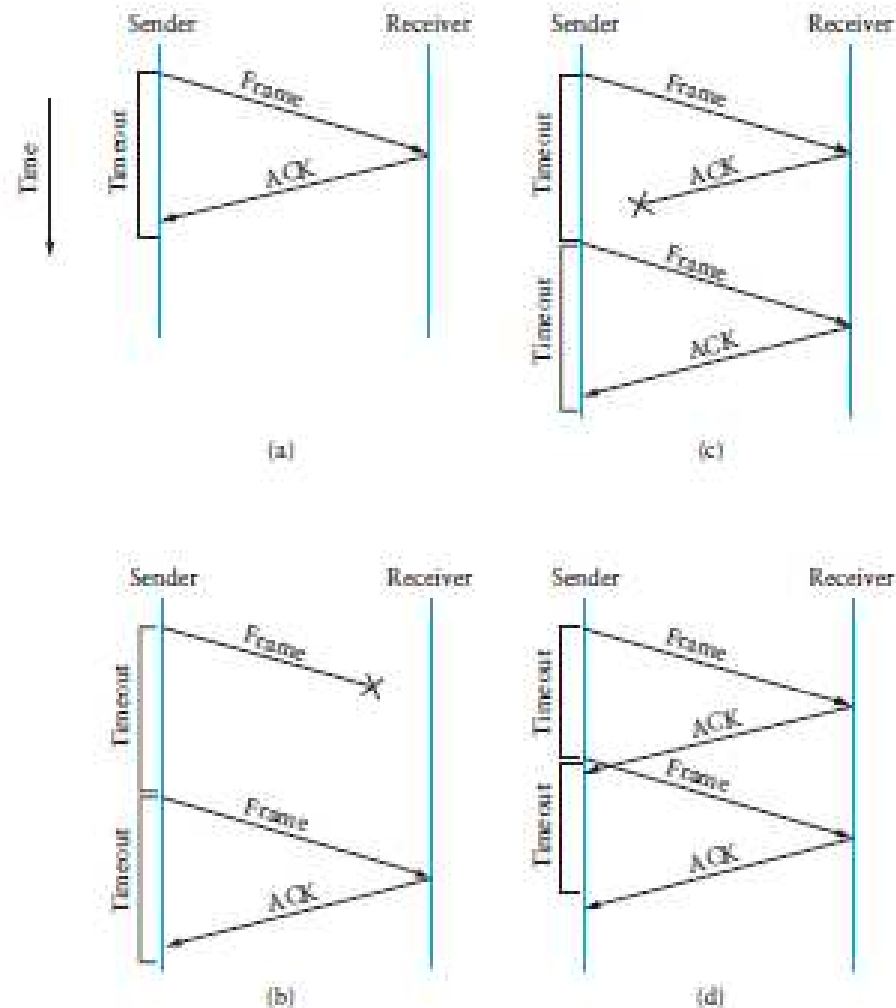


Figure 2.19 Timeline showing four different scenarios for the stop-and-wait algorithm. (a) The ACK is received before the timer expires; (b) the original frame is lost; (c) the ACK is lost; (d) the timeout fires too soon.

O ambiente no projeto de protocolos

- O ambiente onde o protocolo é executado tem um papel importante no projeto de protocolos
- Seja o seguinte cenário:
 - se** Tempo para transmissão de uma PDU +
Tempo de retorno da confirmação é grande
 - então** pode haver uma baixa eficiência do canal de comunicação

O ambiente no projeto de protocolos

■ Exemplo:

- ◆ Canal de satélite de 50 kbps
- ◆ Tempo de propagação de ida e volta (*round trip*) = 500 ms
- ◆ Tamanho do quadro = 1000 bits

O ambiente no projeto de protocolos

■ Cenário

- ◆ $t=0$: início da transmissão
- ◆ $t=20$: fim da TX do quadro
- ◆ $t=270$: receptor recebe todo o quadro
- ◆ $t=520$: transmissor recebe confirmação

■ Análise:

- ◆ Tempo de bloqueio do transmissor = 500 ms ou $500/520 = 96\%$

O ambiente no projeto de protocolos

■ Solução:

- ◆ Permitir que mais quadros sejam transmitidos antes de parar e esperar por confirmação
- ◆ No exemplo acima, pelo menos 26 quadros = 520 ms
- ◆ Técnica conhecida como *pipelining*

Pipelining

- Questão a ser resolvida:
 - ◆ Um quadro com erro no meio de uma seqüência deve ser ignorado ao chegar no RX
 - ◆ O que o RX deve fazer com os quadros corretos restantes?

Pipelining

- Duas soluções básicas:
 - ◆ Retransmitir a partir do quadro errado
 - ▶ *Go back n*
 - ◆ Repetir seletivamente
 - ▶ *Selective repeat*

- A eficiência de cada solução depende do tamanho da janela, taxa de erros, etc

Go back n: Recepção

- Ignora todos os quadros recebidos após o quadro errado
- Confirma somente os quadros recebidos corretamente na seqüência

Go back n: Transmissão

- Temporiza o primeiro quadro não confirmado
- Retransmite esse quadro
- Repete o processo até que todos os quadros sejam confirmados, ou
- Até que um quadro seja enviado um número máximo de vezes e não haja confirmação

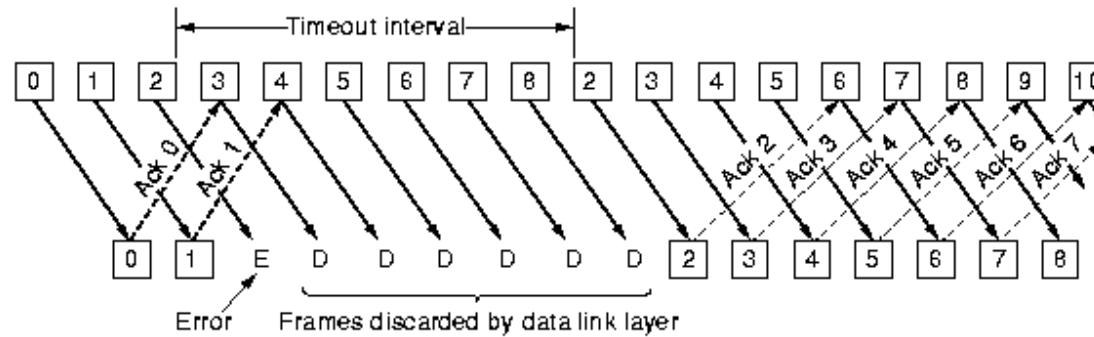
Selective repeat

- Similar ao *go back n*, exceto que RX armazena todos os quadros recebidos corretamente
- Pode necessitar muitos *buffers* no caso do tamanho da janela ser grande

Comentários sobre as soluções

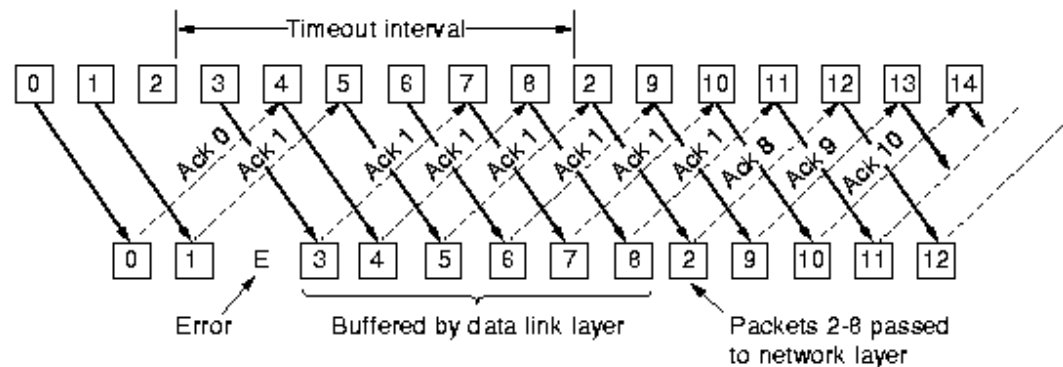
- As duas soluções definem um compromisso entre largura de banda (BW) e *buffers* disponíveis
- A solução a ser adotada pode depender de qual recurso é mais valioso

Go back n e Selective repeat



Time →

(a)



(b)

- (a) Effect of an error when the receiver window size is 1.
 (b) Effect of an error when the receiver window size is large.