

# Redes de Computadores

## Controle de fluxo

Aula 07

## Introdução

- ❑ Comunicação em um enlace envolve a coordenação entre dois dispositivos: emissor e receptor
- ❑ Controle de fluxo:
  - ♦ Quantos dados um emissor pode enviar antes de esperar uma confirmação (ack) do receptor
  - ♦ Está relacionado com capacidade de bufferização no receptor
- ❑ Controle de erros:
  - ♦ O que fazer quando apenas se detecta erros?
    - ♦ Descarte do quadro seguido de retransmissão, porém:
      - ♦ O que retransmitir, como e quando?

## Controle de fluxo

- ❑ Mecanismo de retro-alimentação que informa a fonte (emissor) a capacidade de recepção do destino (receptor)
  - ♦ Objetivo é evitar perda de dados por estouro em buffers de recepção
    - ♦ Destino necessita analisar quadros recebidos antes de enviar as camadas superiores
- ❑ Mecanismos básicos: *stop-and-wait* e janela deslizante
- ❑ Análise considerando transmissão sem erros (por enquanto)
  - ♦ Todos os quadros são transmitidos com sucesso
  - ♦ Nenhum quadro é perdido
  - ♦ Nenhum quadro possui erros
  - ♦ Ordem de chegada é igual a ordem de emissão

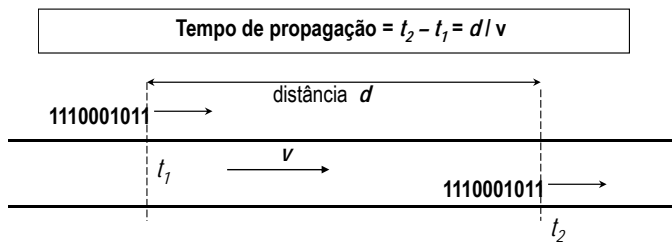


## Questões importantes

- ❑ A duração da transmissão de um quadro depende:
  - ♦ Tempo de propagação do sinal no meio
    - ♦ Função da distância a ser percorrida e da velocidade de transmissão
  - ♦ Tempo de duração do quadro
    - ♦ Função do tamanho do quadro e do tempo de propagação
- ❑ Característica do meio: *half-duplex* versus *full-duplex*
- ❑ Quantidade de dados (quadros) que o emissor pode enviar antes de ser obrigado a parar de transmitir
  - ♦ Protocolo *Stop-and-wait* → apenas um quadro
  - ♦ Protocolo Janela deslizante → capacidade da janela

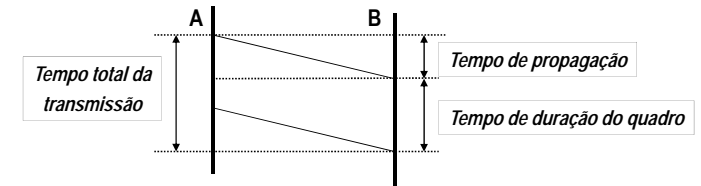
## Tempo de propagação

- ❑ Velocidade de propagação ( $v$ ):
  - ♦ Distância que um sinal percorre em um segundo (e.g.  $3 \times 10^8$  m/s para a luz)
- ❑ Tempo de propagação ( $t$ ):
  - ♦ Tempo necessário para um sinal (*bit*) "viajar" de um ponto a outro
  - ♦ Obviamente é função da distância a ser percorrida



## Duração temporal de um quadro

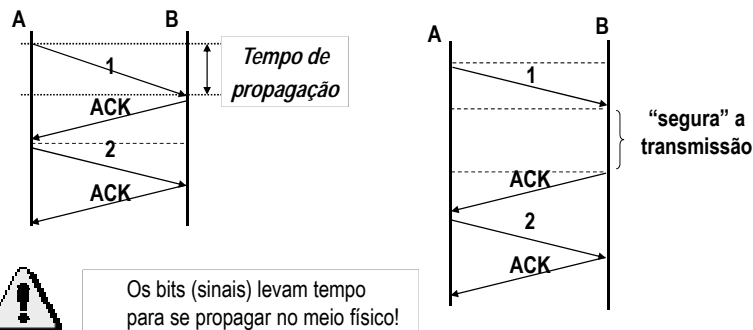
- ❑ Função do tamanho do quadro (*bits*) e da taxa de transmissão (*bps*)
  - ♦ Dado por  $L/R$ , onde  $L$  é tamanho do quadro e  $R$  a taxa de transmissão
- ❑ Tempo total de transmissão ( $s$ )
  - ♦ Duração do quadro + tempo de propagação



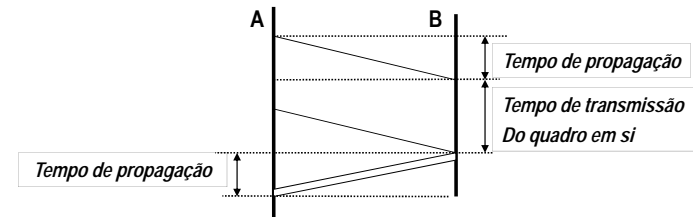
Ex. IEEE802.3: Tempo para envio de um quadro de tamanho mínimo = 76.8  $\mu$ s  
 10 Mbps  $\rightarrow$  0.1  $\mu$ s por bit  
 Quadro com 64 bytes  $\rightarrow$  64 bytes x 8 bits x 0.1  $\mu$ s = 51.2  $\mu$ s  
 Tempo de propagação p/ 2500 m de cabo coaxial = 25.6  $\mu$ s

## Protocolo de controle de fluxo *stop-and-wait*

- ❑ Emissor só pode enviar um quadro  $i$  depois de ter recebido a confirmação da recepção do quadro  $i-1$ 
  - ♦ Definição de dois tipos de quadros: quadro de dados e quadros de controle (ACK)



## Análise do método *stop-and-wait*



- ❑ Inconvenientes do *Stop-and-wait*
  - ♦ Subutiliza o meio de transmissão
  - ♦ Comunicação é sempre *half-duplex* mesmo quando o meio permite *full-duplex*
- ❑ Exemplo: Canal de 1 Mbps e tempo de propagação = 1 ms, quadro de 1000 bits
  - ♦ Se transmite um quadro a cada 2 ms + 1 ms (1000 bits \* 0.000001 s), portanto a cada 3ms do canal, apenas 1 ms é "útil" (considerando quadro ACK de duração desprezível)

## Análise de desempenho: *stop-and-wait*

- ❑ O tempo total para enviar um quadro e receber um *ack* é dado por:

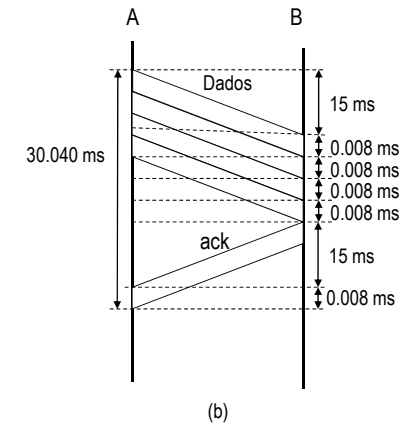
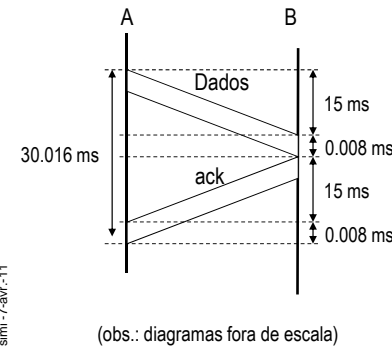
$$T = t_{prop} + t_{frame} + t_{proc} + t_{prop} + t_{ack} + t_{proc}$$

- ❑ Que pode ser aproximado por:  $T = 2 \times t_{prop} + t_{frame}$
- ❑ Eficiência da utilização do meio é:

$$U = \frac{t_{frame}}{2t_{prop} + t_{frame}} \quad \text{fazendo} \quad a = \frac{t_{prop}}{t_{frame}} \quad U = \frac{1}{1+2a}$$

## Uma constatação...

- ❑ Considerando  $t_{prop} = 15$  ms, para quadros de 1000 bytes (8000 bits) em uma rede de 1 Gbps.



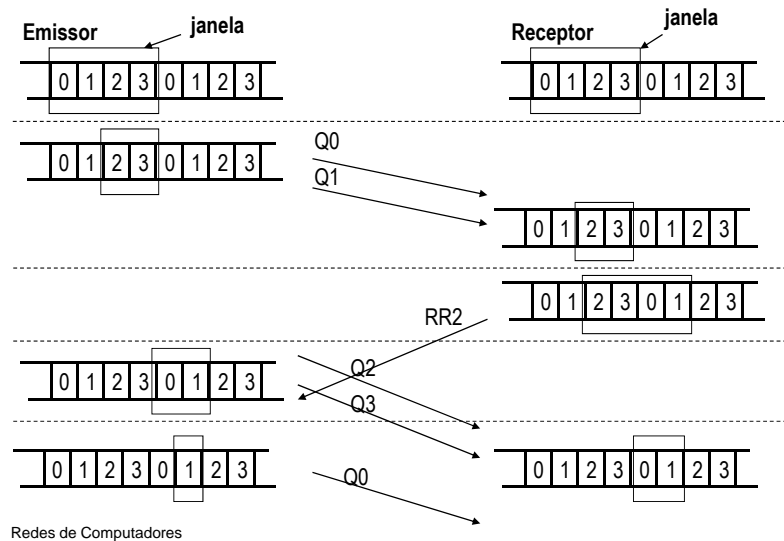
## Protocolo de controle de fluxo por janela deslizante

- ❑ Problema do *stop-and-wait* é ter apenas um quadro em trânsito
- ❑ Melhoria é permitir que a fonte envie  $n$  quadros sem esperar pela confirmação (*ack*) do destino
  - ♦ Necessário que o destino informe sua capacidade de "absorção" de quadros
  - ♦ Número de quadros enviados não deve ultrapassar capacidade de bufferização do destino
- ❑ Princípio básico:
  - ♦ Numerar sequencialmente (módulo  $N$ ) os quadros a serem transmitidos
  - ♦ Controlar os quadros enviados e recebidos
  - ♦ Informar continuamente a capacidade de "absorção" de quadros do destino

## Implementação da janela deslizante

- ❑ Dois tipos de quadros
  - ♦ Quadro de dados: informação e número de sequência
  - ♦ Quadros de controle: *Receive Ready* (RR  $n$ ) e *Receive Not Ready* (RNR  $n$ )
- ❑ Numeração sequencial usando  $k$  bits ( $0$  a  $2^k-1$ )
  - ♦ Limitação em  $k$  bits fornece o efeito módulo (e.g.  $k=2$ ; 0, 1, 2, 3, 0, 1, 2, 3...)
- ❑ Semântica dos quadros de controle:
  - ♦ RR  $n$ : recebi até o quadro  $n-1$ ; pode enviar o quadro  $n$
  - ♦ RNR  $n$ : recebi até o quadro  $n-1$ ; mas não estou pronto para receber o quadro  $n$
- ❑ RR e RNR tem efeito acumulativo

## Exemplo funcionamento janela deslizante (sem erros)



13

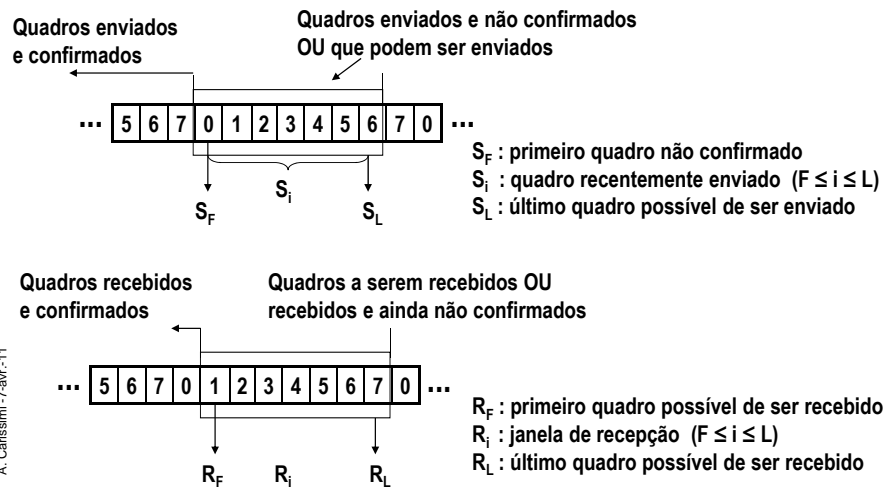
## Exemplo de janela deslizante

<http://www.humboldt.edu/%7eae3/telecom/SlidingWindow.html>

Redes de Computadores

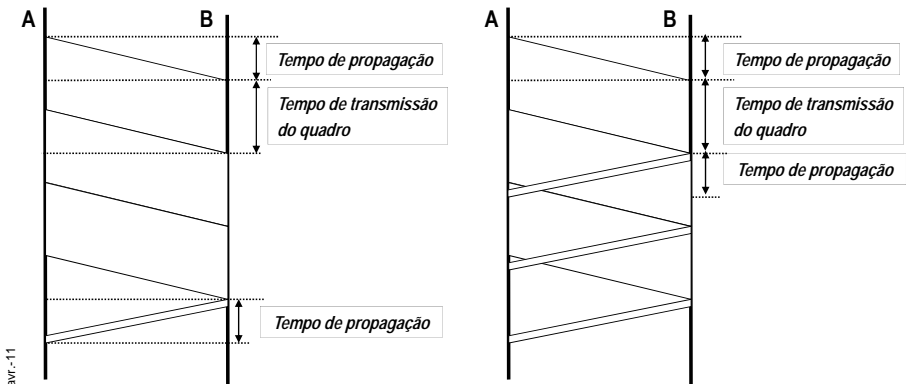
14

## Implementação janela deslizante



15

## Análise do método *janela deslizante*



- Efeito *pipeline* que melhora a eficiência do canal
- Se o canal é *full-duplex* → superposição quadros com ACK

Redes de Computadores

16

## Análise de desempenho: janela deslizante

- ❑ O uso do canal depende do tamanho da janela e do parâmetro  $a$ 
  - ♦ Tempo de transmissão é  $W = n * t_{frame}$
  - ♦ Normalizando tempo de transmissão  $t_{frame} = 1$ ;  $a \rightarrow$  tempo de propagação
- ❑ Caso 1:  $W \geq 2a + 1$ 
  - ♦ O *ack* de um quadro  $i$  é recebido antes da capacidade da janela ter se esgotado, ou seja, é possível enviar continuamente
- ❑ Caso 2:  $W < 2a + 1$ 
  - ♦ O emissor esgota sua capacidade de transmissão em  $t = W$ , ou seja, não pode transmitir mais quadros até receber um *ack*

$$\text{eficiência} = \begin{cases} 1 & \text{para } W \geq 2a + 1 \\ \frac{W}{(2a + 1)} & \text{para } W < 2a + 1 \end{cases}$$

17

## Desempenho da janela deslizante (sem erros)

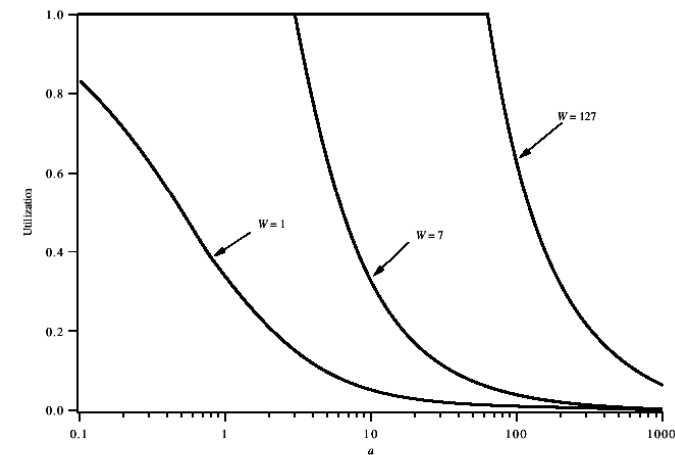


Figure 7.16 Sliding-Window Utilization as a function of  $a$

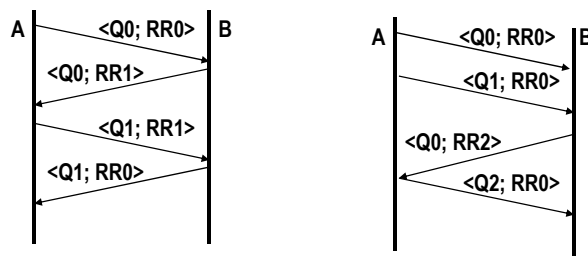
Redes de Computadores

W. Stallings Data & Computer Communications, 6ed, Prentice Hall.

18

## Piggybacking

- ❑ Otimização quando há dados sendo transmitidos nos dois sentidos
- ❑ Envio da confirmação (RR n ou RNR n) junto com dados
  - ♦ Se não existe dados a serem enviados, se envia quadro RR n (RNR n)
  - ♦ Se existe dados a serem enviados, mas não existe ainda uma confirmação a ser enviada, reenvia (repete) o último RR n ou RNR n.



19

## Leituras complementares

- ❑ Stallings, W. *Data and Computer Communications* (6<sup>th</sup> edition), Prentice Hall 1999.
  - ♦ Capítulo 7, seção 7.1 e 7.3
- ❑ Tanenbaum, A. *Redes de Computadores* (4<sup>a</sup> edição), Editora Campus, 2003.
  - ♦ Capítulo 3, seções 3.3 e 3.4

Redes de Computadores

20