

# Redes de Computadores

## Controle de fluxo

### Aula 07

## Controle de fluxo

- ❑ Mecanismo de retroalimentação que informa a fonte (emissor) a capacidade de recepção do destino (receptor)
  - Objetivo é evitar perda de dados por estouro em buffers de recepção
- ❑ Mecanismos básicos: *stop-and-wait* e janela deslizante
- ❑ Análise considera transmissão sem erros
  - Todos os quadros são transmitidos com sucesso
  - Nenhum quadro é perdido
  - Nenhum quadro possui erros
  - Ordem de chegada é igual a ordem de emissão



Próxima aula será analisado o que ocorre se houver erros

## Introdução

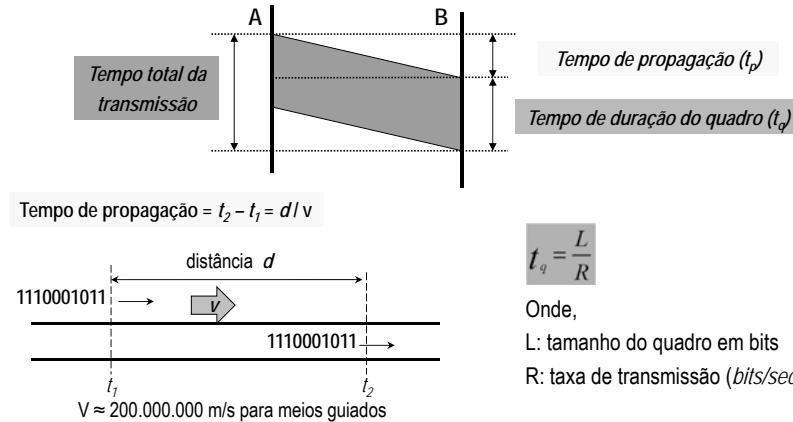
- ❑ Comunicação em um enlace envolve a coordenação entre dois dispositivos, o emissor e o receptor
  - Serviço de entrega confiável ou não
- ❑ Serviço confiável
  - Controle de fluxo:
    - Define quantos dados um emissor pode enviar antes de esperar uma confirmação (*ack*) do receptor
    - Está relacionado com capacidade de *buffer* do 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?

## Questões importantes

- ❑ Característica do enlace:
  - *half-duplex* versus *full-duplex*
- ❑ Duração da transmissão de um quadro depende:
  - Tempo de propagação do sinal no meio
    - Função da distância a ser percorrida e da velocidade de propagação
  - Tempo de duração do quadro
    - Função do tamanho do quadro e do tempo de propagação
- ❑ Quantidade de dados (quadros) que o emissor pode enviar antes de ser obrigado a parar de transmitir

## Duração da transmissão

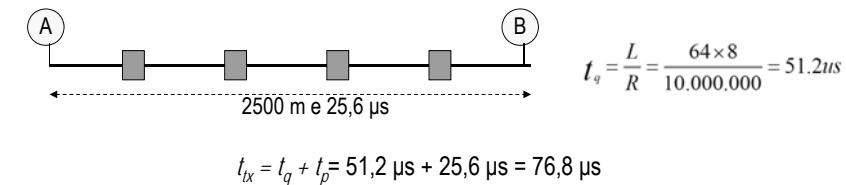
- Soma do tempo de propagação ( $t_p$ ) com a duração do quadro ( $t_q$ )



## Exemplo: IEEE 802.3 (padrão original)

- Padrão define:

- Taxa de transmissão de 10 Mbps  $\rightarrow 0,1\mu\text{s}$  por bit
- Diâmetro da rede: 2500 m, composto por 5 segmentos de até 500 m, interligados por quatro repetidores
  - Maior distância entre dois dispositivos
- Cabo coaxial (tempo de propagação = 25,6  $\mu\text{s}$ , incluindo os 4 repetidores )
- Quadro mínimo igual a 64 bytes (512 bits)

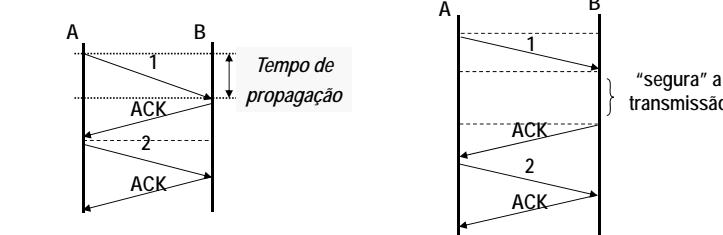


## Quantidade de dados

- Objetivo é evitar estouro de buffer no receptor
  - Controle de fluxo
- Protocolos de controle de fluxo
  - Controla a quantidade de dados que pode ser enviada antes do transmissor ser obrigado a parar de transmitir
  - Duas estratégias
    - Stop and wait
    - Janela deslizante

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

- Emissor só pode enviar um quadro / 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)



As setas representam quadros e não bits, portanto, os gráficos acima não mostram a duração dos quadros.

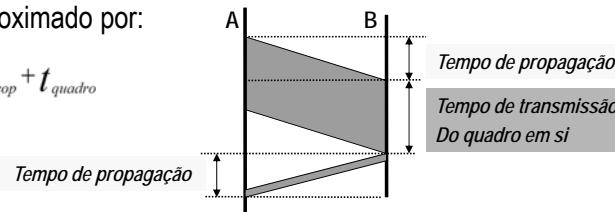
## Análise de desempenho: *stop-and-wait*

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

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

- Pode ser aproximado por:

$$T = 2 \times t_{prop} + t_{quadro}$$



- Eficiência da utilização do meio é:

$$U = \frac{t_{quadro}}{2t_{prop} + t_{quadro}}$$

fazendo  $a = \frac{t_{prop}}{t_{quadro}}$   $\Rightarrow U = \frac{1}{2a+1}$

Redes de Computadores

9

## Análise do método *stop-and-wait*

- Exemplo: Canal de 1 Mbps, tempo de propagação = 1 ms, e quadro de 1000 bits

- $t_q = 1000 \text{ bits} * 1 \mu\text{s} = 1 \text{ ms}$ ;  $t_p = 1 \text{ ms}$ ;  $t_{total} = 2 \text{ ms}$
  - Considerando o  $t_q$  do *ack* desprezível, se tem uma transmissão a cada 3 ms ( $2 \cdot t_p + t_q$ )
  - Então, o canal de 1 Mbps apresenta um desempenho de 333,33.. Kbps!!! ☺

- Inconvenientes do *Stop-and-wait*

- Subutiliza o meio de transmissão
  - Comunicação é sempre *half-duplex* mesmo quando o meio permite *full-duplex*

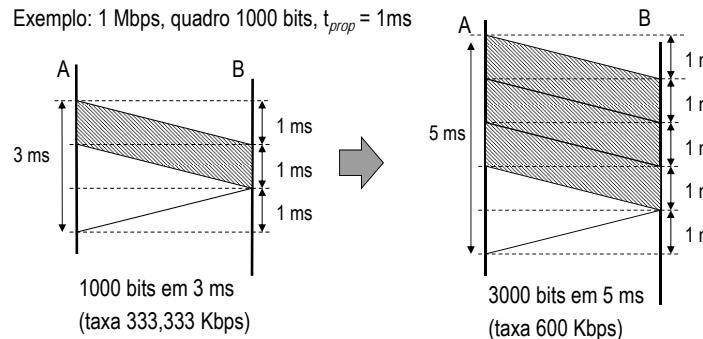
Problema do *stop-and-wait* é ter apenas um quadro em trânsito, por vez.

Redes de Computadores

10

## Envio de $n$ quadros

Exemplo: 1 Mbps, quadro 1000 bits,  $t_{prop} = 1 \text{ ms}$



- Questão:

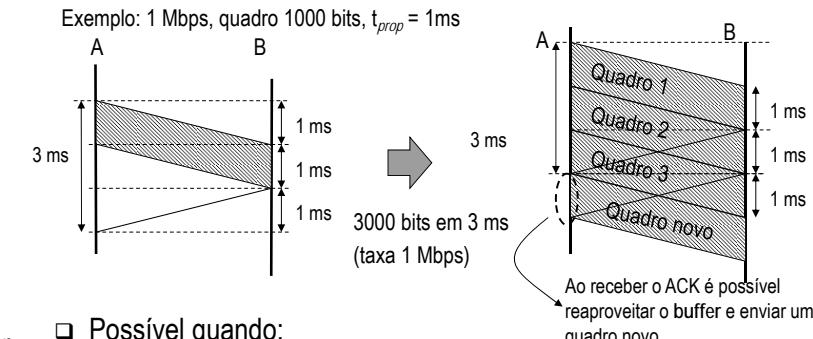
- Porque simplesmente não aumentar o tamanho do quadro?

Redes de Computadores

11

## Envio de $n$ quadros continuamente

Exemplo: 1 Mbps, quadro 1000 bits,  $t_{prop} = 1 \text{ ms}$



- Possível quando:

- houver dados suficientes para enviar se obtém um efeito *pipeline*
    - Transmite quadro de dados  $i+1$  antes de receber ack do  $i$
  - Enlace for *full duplex*

Redes de Computadores

12

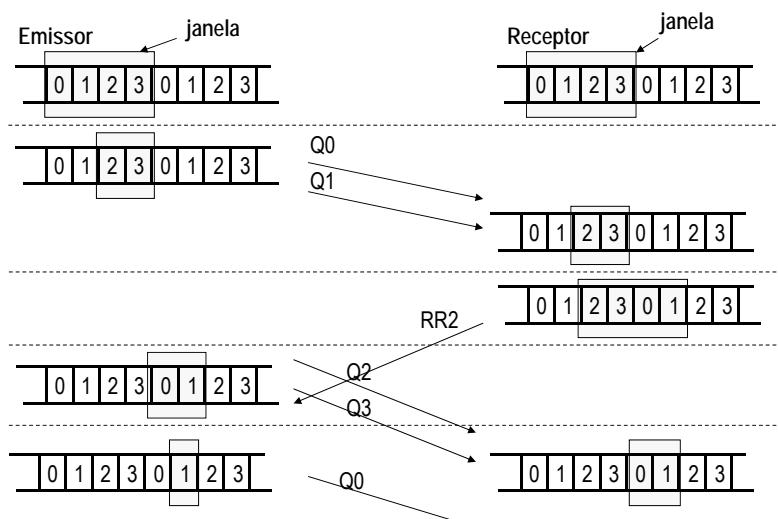
## Protocolo de controle de fluxo por janela deslizante

- ❑ Permite a origem enviar  $q$  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 buffer do destino
- ❑ Princípio básico:
  - Numerar sequencialmente (módulo N) os quadros a serem transmitidos
  - A capacidade inicial da janela (*buffer*) é  $2^k$ , depois vai sendo ajustado conforme quadros são enviados e confirmados pelo receptor
    - Informar continuamente a capacidade de “absorção” de quadros do destino
    - Controlar os quadros enviados e recebidos

## Implementação da janela deslizante

- ❑ Dois tipos de quadros
  - Quadro de dados: informação e número de seqüência
  - 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

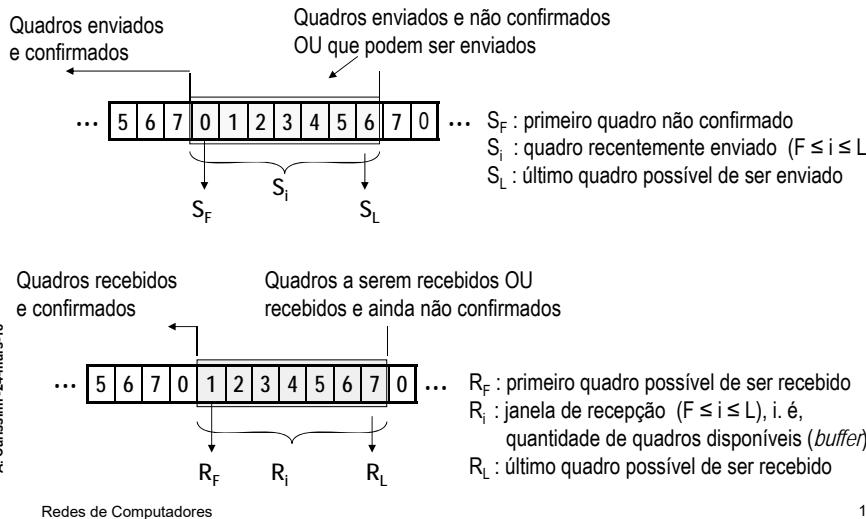
## Exemplo funcionamento janela deslizante (sem erros)



## Janela deslizante no youtube

- ❑ Simulação da janela deslizante
  - <http://www.youtube.com/watch?v=EHaSQBOrYDI>

## Implementação janela deslizante



## Análise de desempenho: janela deslizante

- O uso do canal depende do tamanho da janela e do parâmetro  $a$ 
  - Relação entre o tempo de propagação e o tempo de duração do quadro

$$a = \frac{t_{prop}}{t_{quadro}}$$

- Também pode ser visto como a capacidade em bits para “preencher” todo o meio em relação ao tamanho do quadro ( $L$ ), dado por:

$$a = \frac{d/v}{L/R} = R \times \frac{d}{vL}$$

← Produto largura de banda-atraso

- Desempenho se traduz pela quantidade de bits no meio

- Tempo de transmissão é  $W = n * t_{quadro}$
- Se preencher meio obtém eficiência de 100%

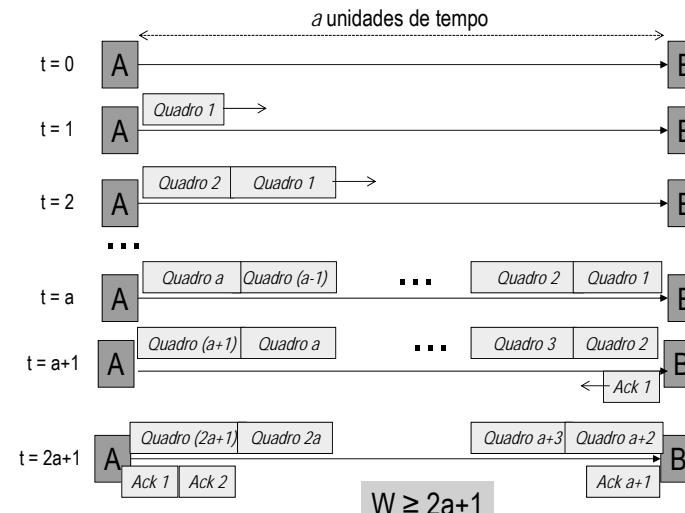
## Análise de desempenho: janela deslizante (cont.)

- Se:
  - Normalizando tempo de duração  $t_{quadro} = 1$  então  $a$  = 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*

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

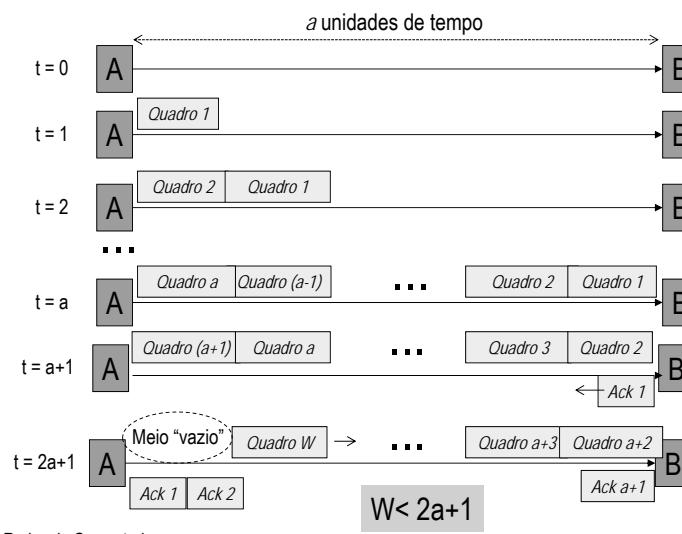
## Outra forma de ver... (eficiência = 1)

Supondo  $t_{quadro}=1$ ,  
tem-se  $a = t_{propagação}$   
pois  $a = t_{propagação}/t_{quadro}$



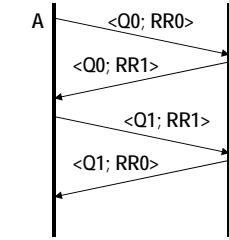
## Outra forma de ver... (eficiência < 1)

Supondo  $t_{quadro}=1$ ,  
tem-se  $a = t_{propagação}$   
pois  $a = t_{propagação}/t_{quadro}$



## 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  $RNR\ n$ .



## 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