

Aula 14

Professores:

Anna Dolejsi Santos (UFF)

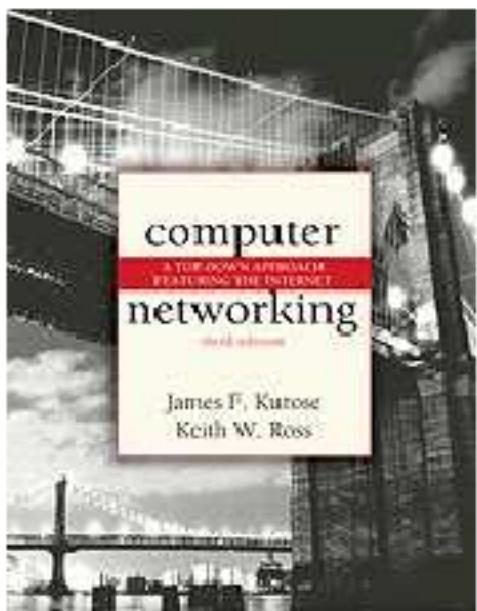
Célio Vinicius Neves de Albuquerque (UFF)

Controle de congestionamento

Conteúdo:

- 3.6 Princípios de controle de congestionamento
- 3.7 Controle de congestionamento do TCP

Livro Texto



REDES DE COMPUTADORES E A INTERNET UMA ABORDAGEM TOP-DOWN

(www.aw.com/kurose_br)

James F. Kurose e Keith W. Ross

Copyright: 2006 - 3a. Edição

ISBN: 8588639181

<http://www.pearson.com.br/>

Referência Adicional:

Computer Networks, Andrew Tanenbaum -
Capítulos 1-6.

Obs: As figuras que não têm referências pertencem ao material disponilizado pelo autor do livro texto ou foram produzidas pelo professor desta disciplina.

Conteúdo do Capítulo 3

- 3.1 Serviços da camada de transporte
- 3.2 Multiplexação e demultiplexação
- 3.3 UDP: Transporte não orientado a conexão
- 3.4 Princípios da transferência confiável de dados
- 3.5 Transporte orientado a conexão: TCP
 - transferência confiável**
 - controle de fluxo**
 - gerenciamento de conexões**
- 3.6 Princípios de controle de congestionamento**
- 3.7 Controle de congestionamento do TCP

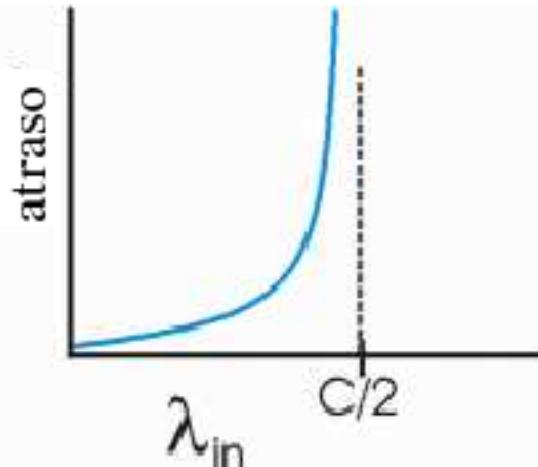
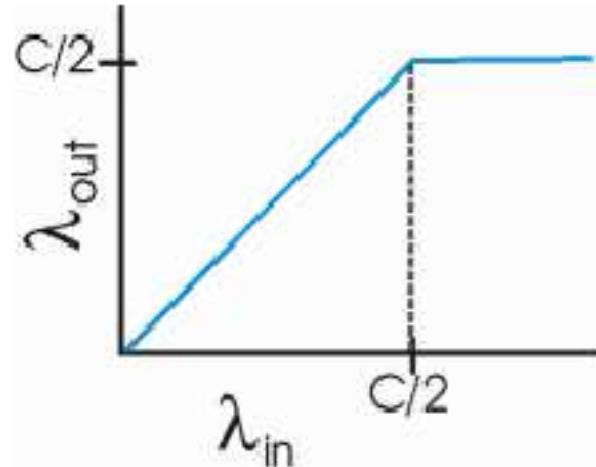
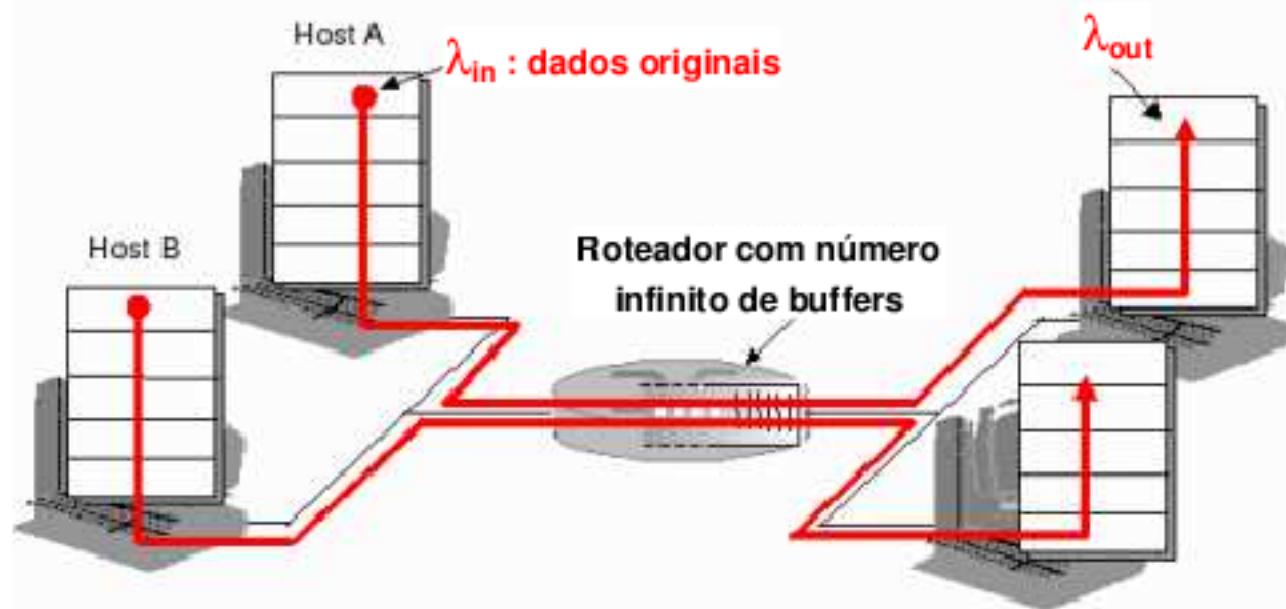
Princípios de Controle de Congestionamento

Congestionamento:

- informalmente: "muitas fontes enviando muitos dados muito rapidamente para a *rede* poder tratar"
- diferente de controle de fluxo!
- manifestações:
 - perda de pacotes (esgotamento de buffers em roteadores)
 - longos atrasos (enfileiramento nos buffers dos roteadores)
- um dos 10 problemas mais importantes em redes!

Causas/custos de congestionamento: cenário 1

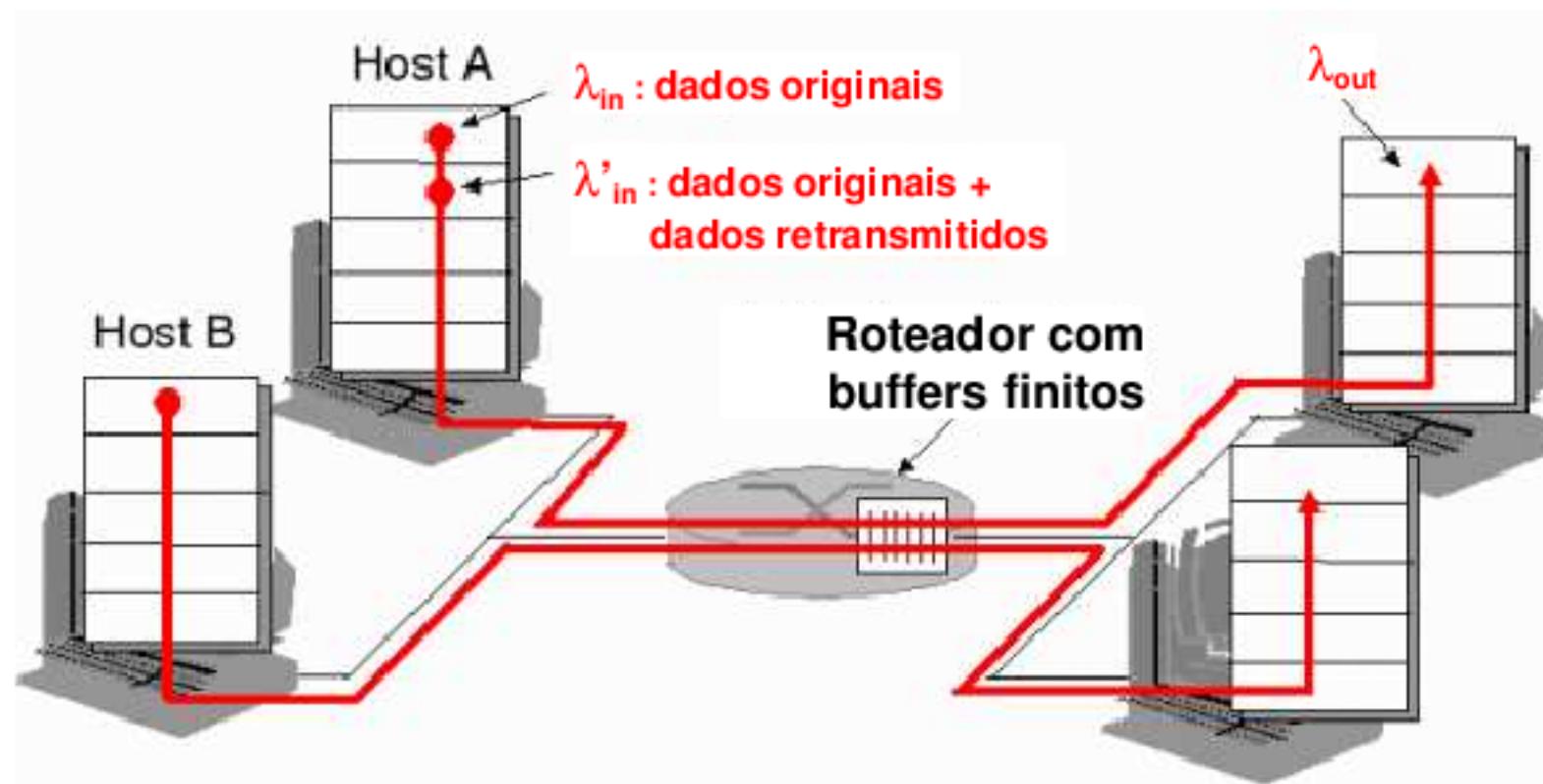
- dois remetentes,
dois receptores
- um roteador,
buffers infinitos
- sem retransmissão



- grandes retardos
qdo. congestionada
- vazão máxima
alcançável

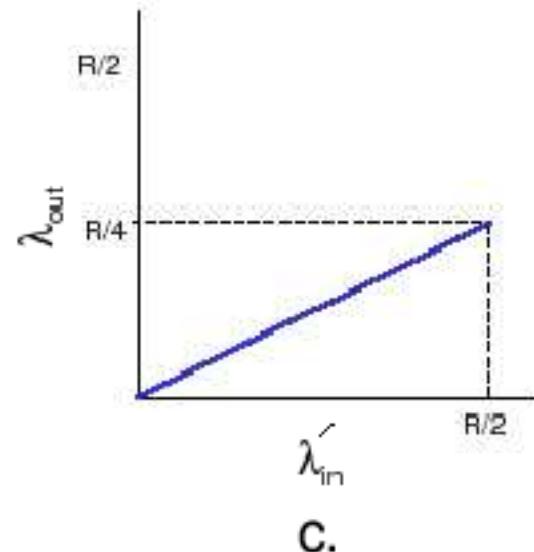
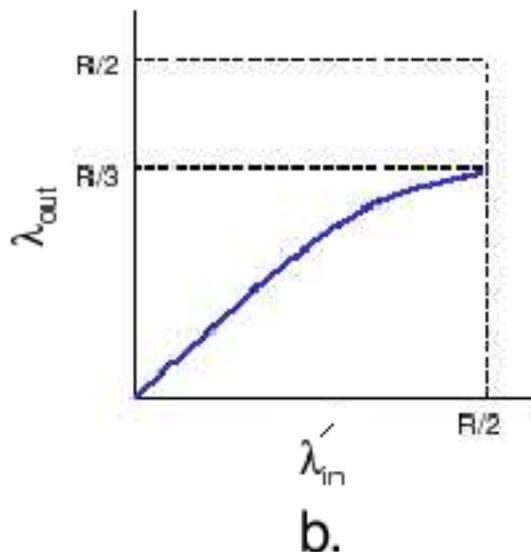
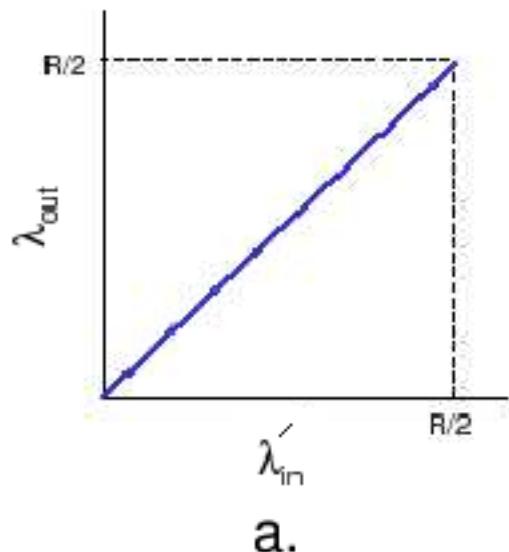
Causas/custos de congestionamento: cenário 2

- Um roteador, buffers *finitos*
- retransmissão pelo remetente de pacote perdido



Causas/custos de congestionamento: cenário 2

- ❑ sempre: $\lambda_{in} = \lambda_{out}$ (goodput)
- ❑ retransmissão "perfeita" apenas com perdas: $\lambda'_{in} > \lambda_{out}$
- ❑ retransmissão de pacotes atrasados (não perdidos) torna λ'_{in} maior (do que o caso perfeito) para o mesmo λ_{out}



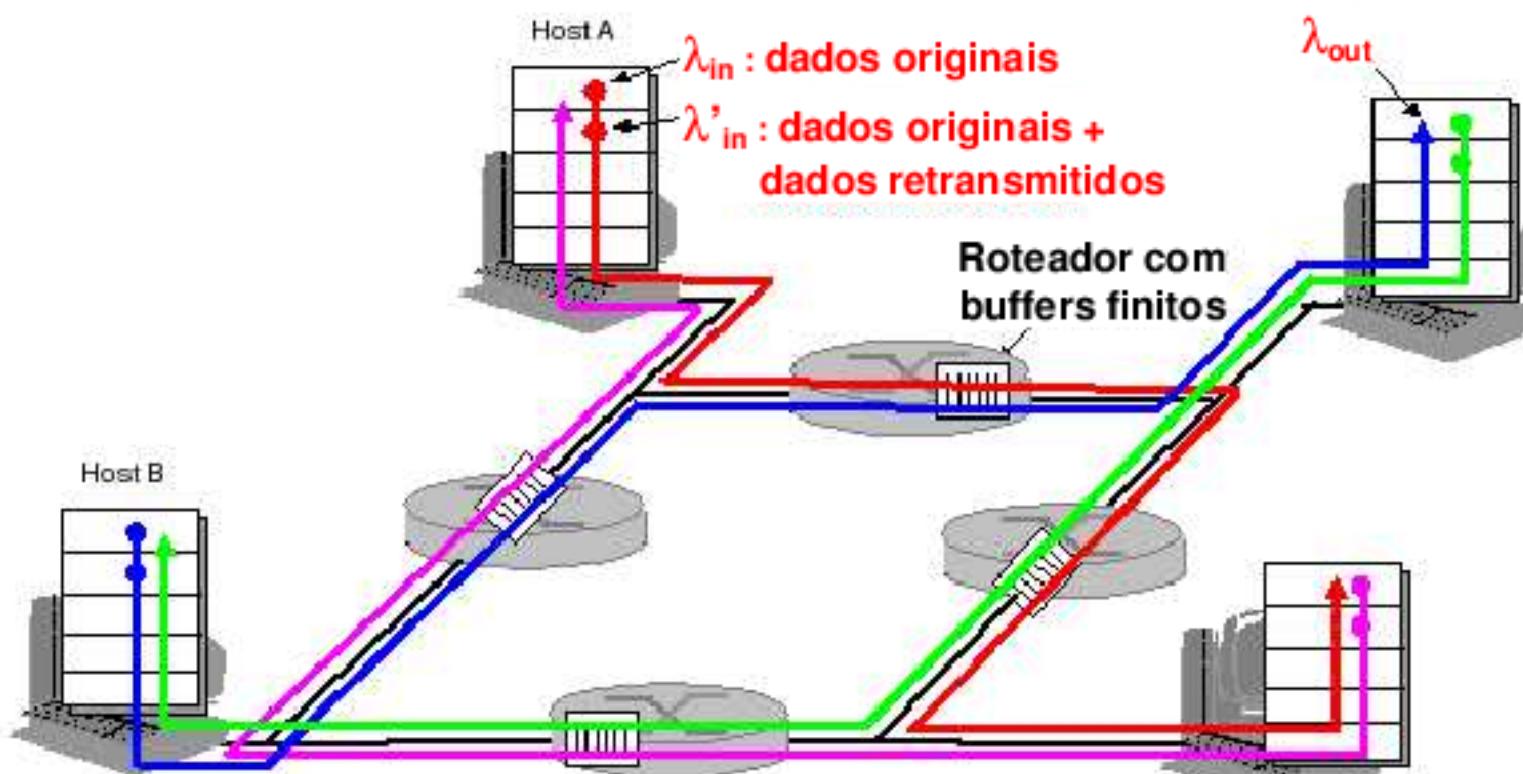
"custos" de congestionamento:

- ❑ mais trabalho (retransmissão) para dado "goodput"
- ❑ retransmissões desnecessárias: enviadas múltiplas cópias do pacote

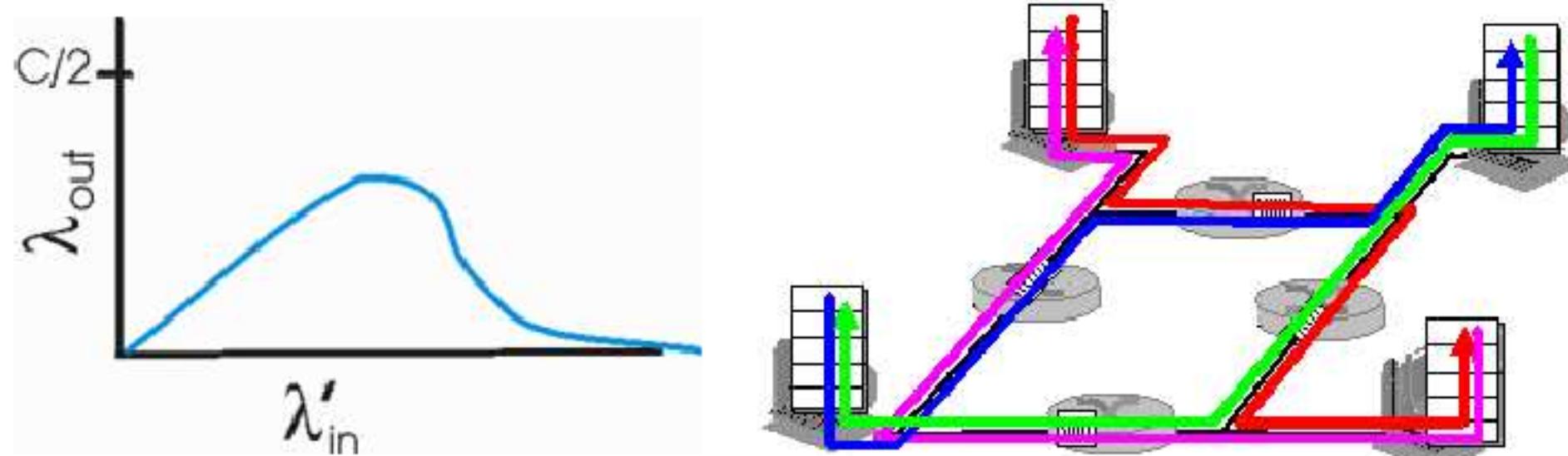
Causas/custos de congestionamento: cenário 3

- quatro remetentes
- caminhos com múltiplos enlaces
- temporização/retransmissão

P: o que acontece à medida que λ_{in} e λ'_{in} crescem ?



Causas/custos de congestionamento: cenário 3



Outro "custo" de congestionamento:

- quando pacote é descartado, qq. capacidade de transmissão já usada (antes do descarte) para esse pacote foi desperdiçada!

Abordagens de controle de congestionamento

Duas abordagens amplas para controle de congestionamento:

Controle de congestionamento fim a fim :

- não tem realimentação explícita pela rede
- congestionamento inferido a partir das perdas, retardo observados pelo sistema terminal
- abordagem usada pelo TCP

Controle de congestionamento com apoio da rede:

- roteadores realimentam os sistemas terminais
 - bit indicando congestionamento (SNA, DECbit, TCP/IP ECN, ATM)
 - taxa explícita p/ envio pelo remetente

Estudo de caso: controle de congestionamento no ABR da ATM

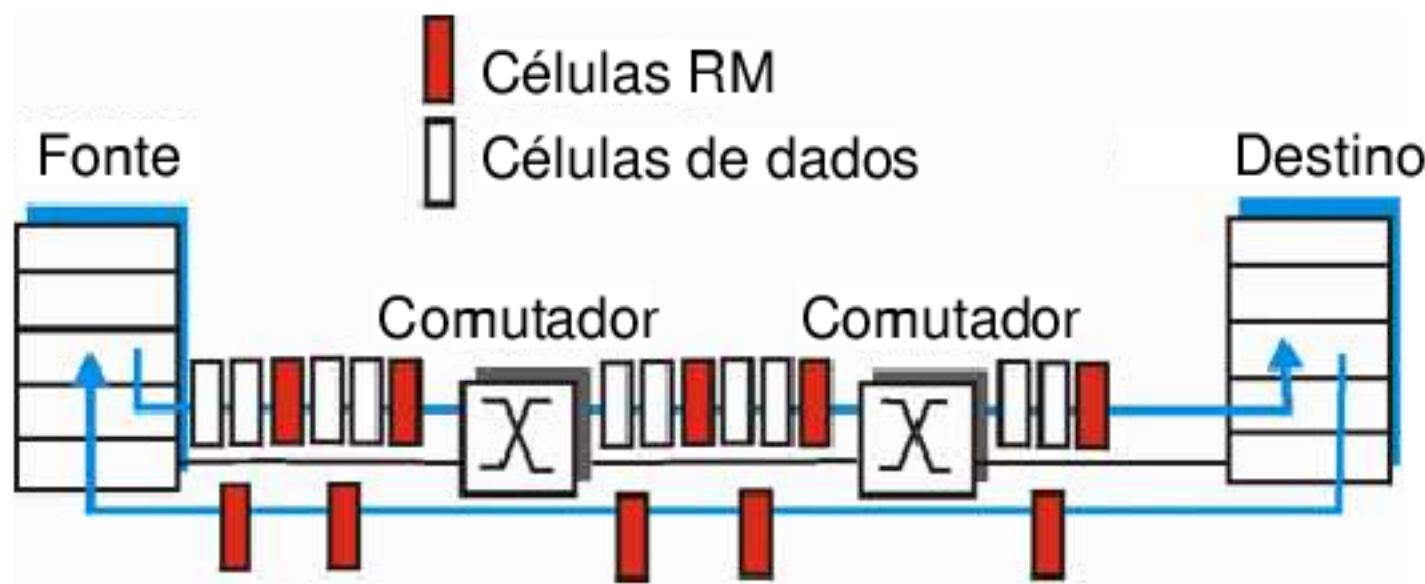
ABR (*available bit rate*):

- "serviço elástico"
- se caminho do remetente "subcarregado":
 - remetente deveria usar banda disponível
- se caminho do remetente congestionado:
 - remetente reduzido à taxa mínima garantida

células RM (*resource management*):

- enviadas pelo remetente, intercaladas com células de dados
- bits na célula RM iniciados por comutadores ("apoio da rede")
 - bit NI: não aumente a taxa (congestionamento moderado)
 - bit CI: indicação de congestionamento
- células RM devolvidas ao remetente pelo receptor, sem alteração dos bits

Estudo de caso: controle de congestionamento em ABR da ATM



- Campo ER (explicit rate) de 2 bytes na célula RM
 - comutador congestionado pode diminuir valor ER na célula
 - taxa do remetente assim ajustada p/ menor valor possível entre os comutadores do caminho
- bit EFCI em células de dados ligado por comutador congestionado
 - se EFCI ligado na célula de dados antes da célula RM, receptor liga bit CI na célula RM devolvida

Conteúdo do Capítulo 3

- 3.1 Serviços da camada de transporte
- 3.2 Multiplexação e demultiplexação
- 3.3 Transporte não orientado a conexão: UDP
- 3.4 Princípios da transferência confiável de dados
- 3.5 Transporte orientado a conexão: TCP
 - transferência confiável**
 - controle de fluxo**
 - gerenciamento de conexões**
- 3.6 Princípios de controle de congestionamento
- 3.7 Controle de congestionamento do TCP

Controle de Congestionamento do TCP

- ❑ controle fim-a-fim (sem assistência da rede)
- ❑ transmissor limita a transmissão:

$$\text{LastByteSent} - \text{LastByteAcked} \leq \text{CongWin}$$

- ❑ Praticamente,

$$\text{taxa} = \frac{\text{CongWin}}{\text{RTT}} \text{ Bytes/seg}$$

- ❑ **CongWin** é dinâmica, em função do congestionamento percebido da rede

Como o transmissor percebe o congestionamento?

- ❑ evento de perda = estouro do temporizador *ou* 3 acks duplicados
- ❑ transmissor TCP reduz a taxa (**CongWin**) após evento de perda

três mecanismos:

- AIMD
- partida lenta
- conservador após eventos de estouro de temporização

Controle de Congestionamento do TCP

- controle fim-a-fim (sem assistência da rede)
- transmissor limita a transmissão:

$$\text{LastByteSent} - \text{LastByteAcked} \leq \text{CongWin}$$

- Praticamente,

$$\text{taxa} = \frac{\text{CongWin}}{\text{RTT}} \text{ Bytes/seg}$$

- CongWin** é dinâmica, em função do congestionamento percebido da rede

OBS.: Na verdade, a taxa de transmissão que o TRANSMISSOR vai poder transmitir é dada pela expressão acima.

Como o transmissor percebe o congestionamento?

- evento de perda = estouro do temporizador ou 3 acks duplicados
- transmissor TCP reduz a taxa (**CongWin**) após evento de perda

três mecanismos:

- AIMD
- partida lenta
- conservador após eventos de estouro de temporização

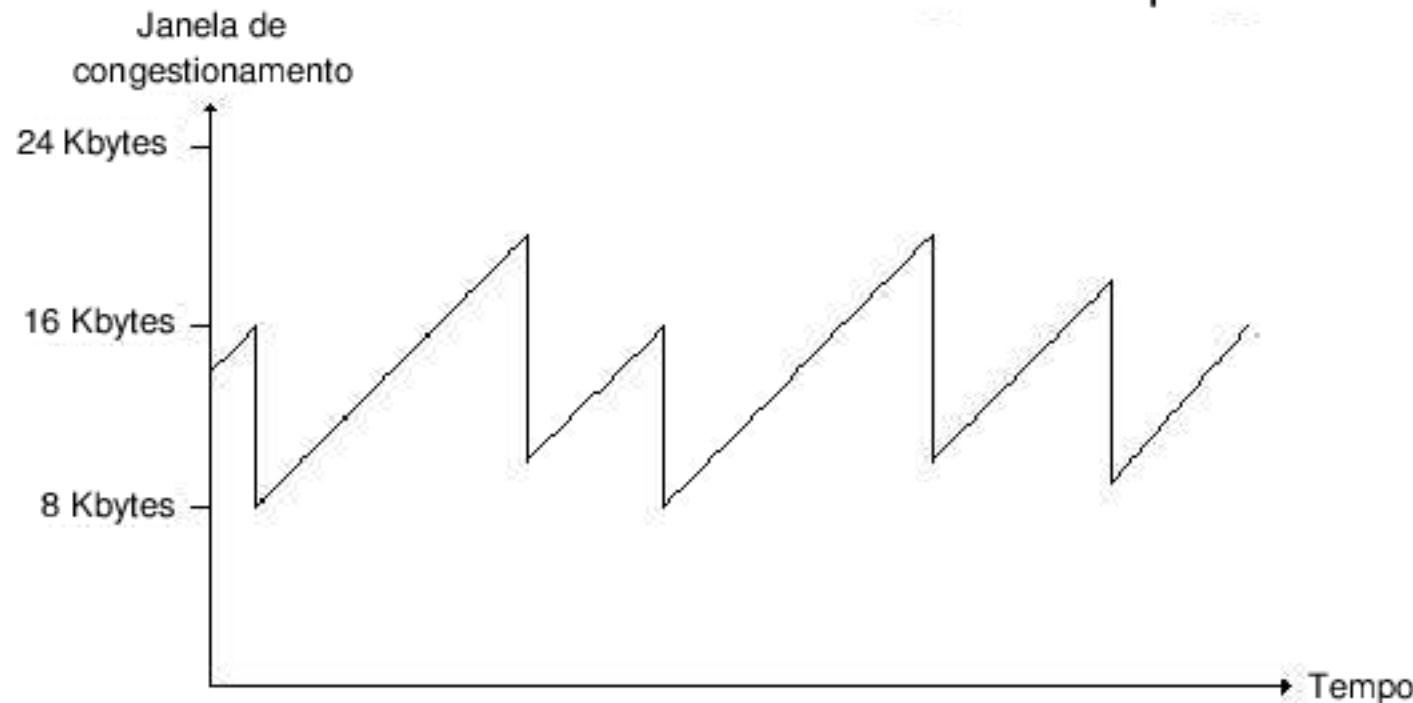
AIMD do TCP

decrescimento multiplicativo:

corta **CongWin** pela metade
após evento de perda

crescimento aditivo:

incrementa **CongWin** de 1 MSS
a cada RTT na ausência de
eventos de perda: *sondagem*



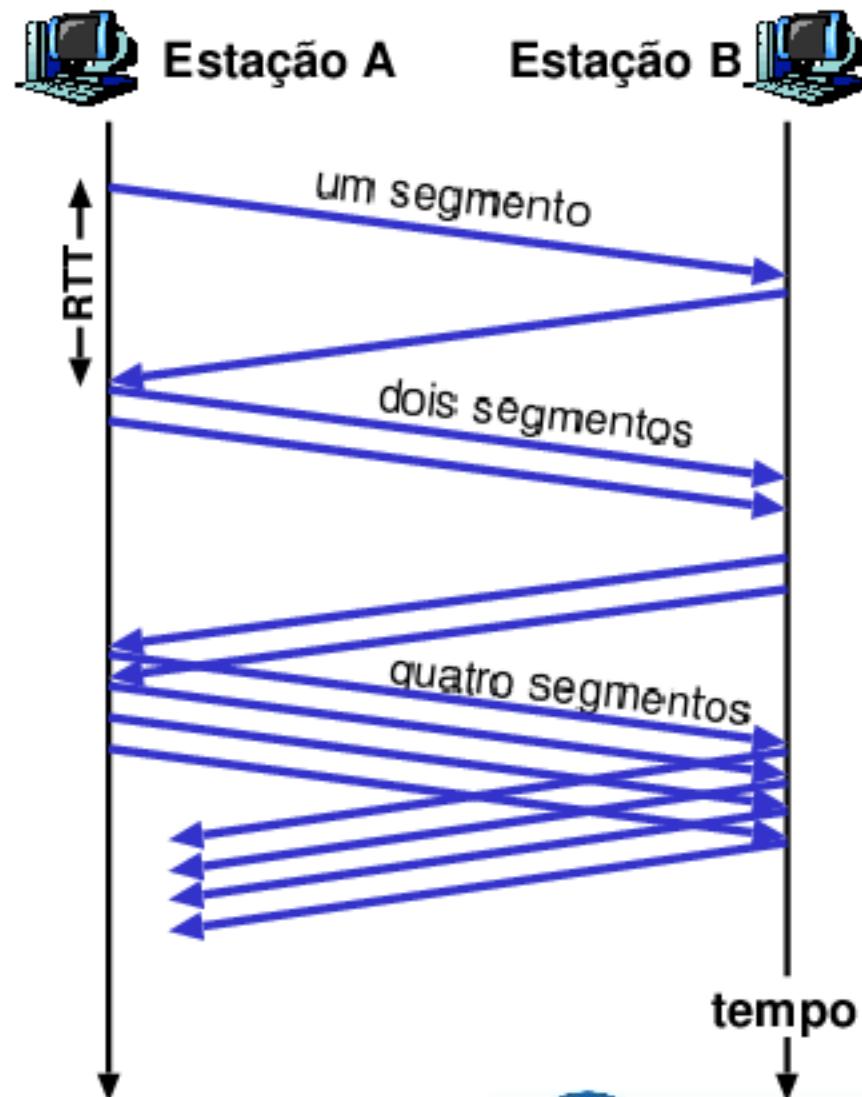
Conexão TCP de longa duração

Partida Lenta do TCP

- No início da conexão,
 $CongWin = 1 \text{ MSS}$
 - Exemplo: $\text{MSS} = 500 \text{ bytes}$
 $\& \text{RTT} = 200 \text{ mseg}$
 - taxa inicial = 20 kbps
- largura de banda disponível
pode ser $>> \text{MSS}/\text{RTT}$
 - é desejável um crescimento
rápido até uma taxa
considerável
- No início da conexão, aumenta
a taxa exponencialmente até o
primeiro evento de perda

TCP: Partida lenta (mais)

- ❑ No início da conexão, aumenta a taxa exponencialmente até o primeiro evento de perda:
 - duplica CongWin a cada RTT
 - através do incremento da CongWin para cada ACK recebido
- ❑ Resumo: taxa inicial é baixa mas cresce rapidamente de forma exponencial



Refinamento

- ❑ Após 3 ACKs duplicados:
 - corta CongWin pela metade
 - a janela depois cresce linearmente

- ❑ Mas após estouro de temporizador:
 - CongWin é reduzida a 1 MSS;
 - janela cresce exponencialmente
 - até um limiar, depois cresce linearmente

Filosofia:

- 3 ACKs duplicados indica que a rede é capaz de entregar alguns segmentos

- estouro de temporizador antes de 3 ACKs duplicados é mais "alarmante".

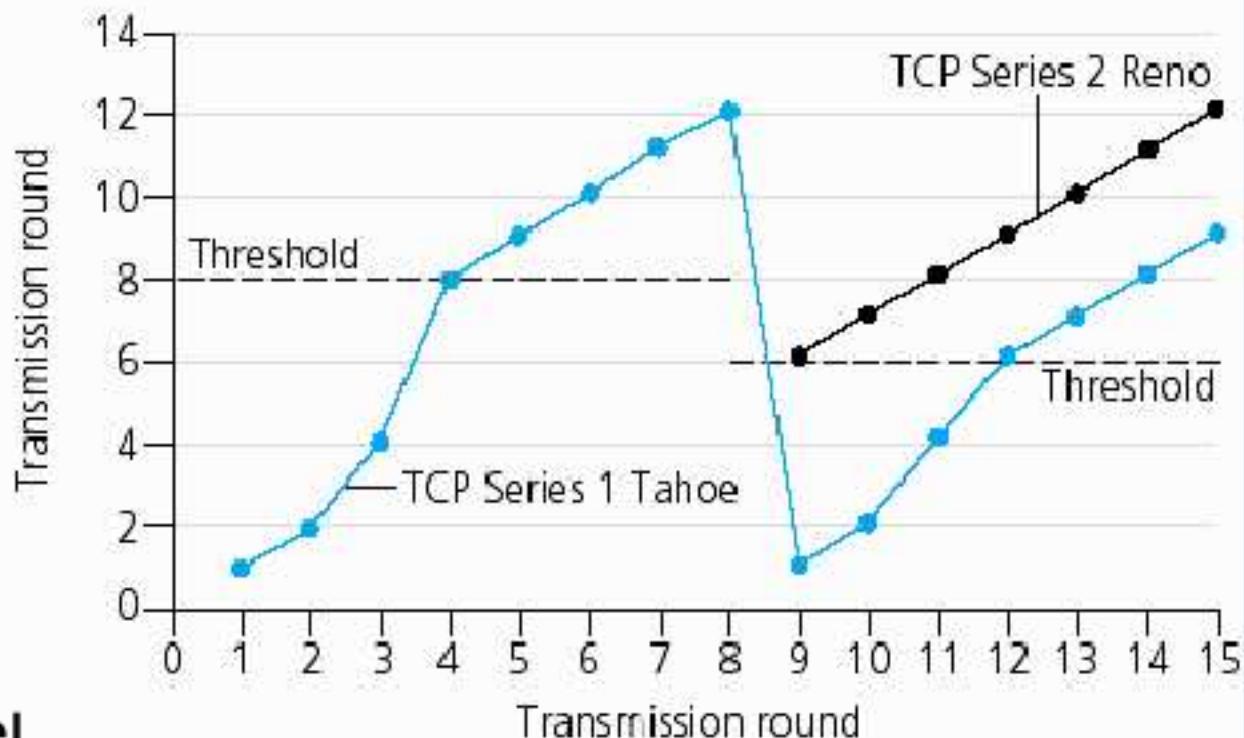
Refinamento (mais)

P: Quando o crescimento exponencial deve mudar para linear?

R: Quando CongWin atinge 1/2 do seu valor antes do estouro do temporizador.

Implementação:

- Limiar (*Threshold*) variável
- Com uma perda o limiar passa a ser 1/2 da CongWin imediatamente anterior à perda.



Resumo: Controle de Congestionamento do TCP

- ❑ Quando a **CongWin** está abaixo do **limiar**, transmissor está na fase de **início lento**, janela cresce exponencialmente.
- ❑ Quando a **CongWin** está acima do **limiar**, transmissor está na fase de **evitar congestionamento**, janela cresce linearmente.
- ❑ Quando chegam **ACKs triplicados**, **limiar** passa a ser **CongWin/2** e **CongWin** passa ao valor do **Limiar**.
- ❑ Quando **estoura o temporizador**, **Limiar** passa a ser **CongWin/2** e **CongWin** passa a ser 1 MSS.

Controle de congestionamento do transmissor TCP

Evento	Estado	Ação do Transmissor TCP	Comentário
ACK recebido para dados ainda não reconhecidos	Partida lenta	$CongWin = CongWin + MSS,$ $If (CongWin > Limiar)$ seta estado para "Evitar congestionamento"	Resulta na duplicação da CongWin a cada RTT
ACK recebido para dados ainda não reconhecidos	Evitar congestiona- mento	$CongWin = CongWin + MSS * (MSS/CongWin)$	Incremento aditivo, resultando no incremento da CongWin de 1 MSS a cada RTT
Perda detectada por ACKs triplicados	qualquer	$Limiar = CongWin/2,$ $CongWin = Limiar,$ Seta estado para "Evitar Congestionamento"	Recuperação rápida, implementa decrescimento multiplicativo. CongWin não cai abaixo de 1 MSS.
Estouro de temporizador	qualquer	$Limiar = CongWin/2,$ $CongWin = 1 MSS,$ Seta estado para "Partida lenta"	Entra estado de "partida lenta"
ACK duplicado	qualquer	Incrementa contador de ACKs duplicados para o segmento que está sendo reconhecido	CongWin e Threshold não se alteram