

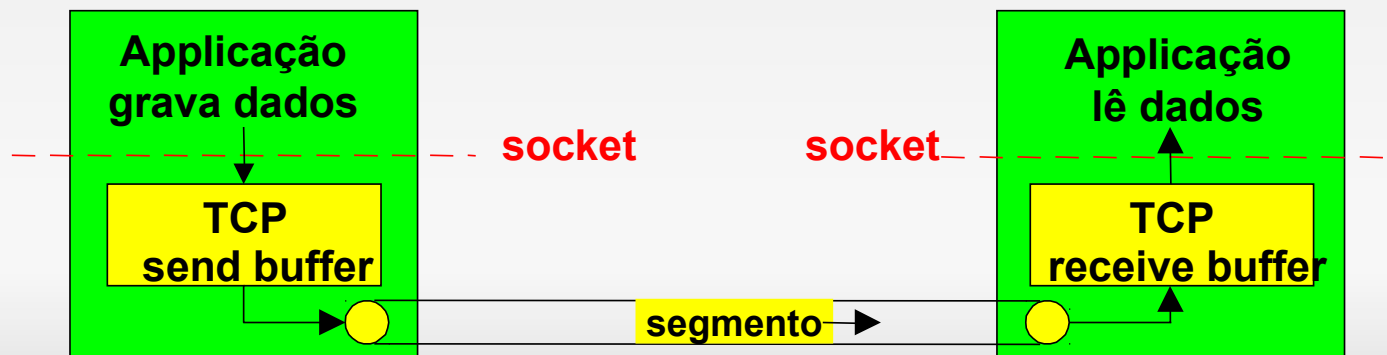
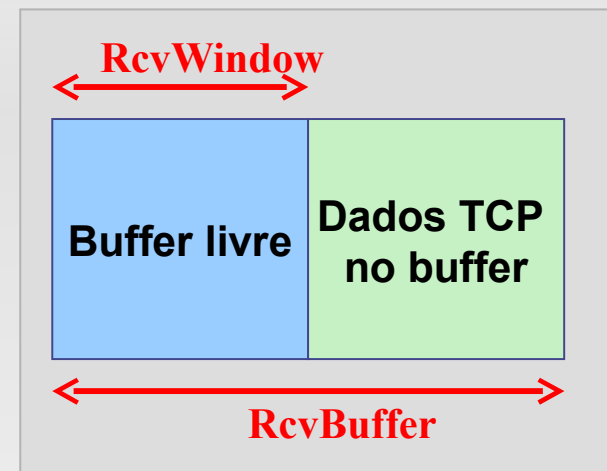
# Controle de fluxo e Controle de congestionamento no TCP

Líliam B. Leal

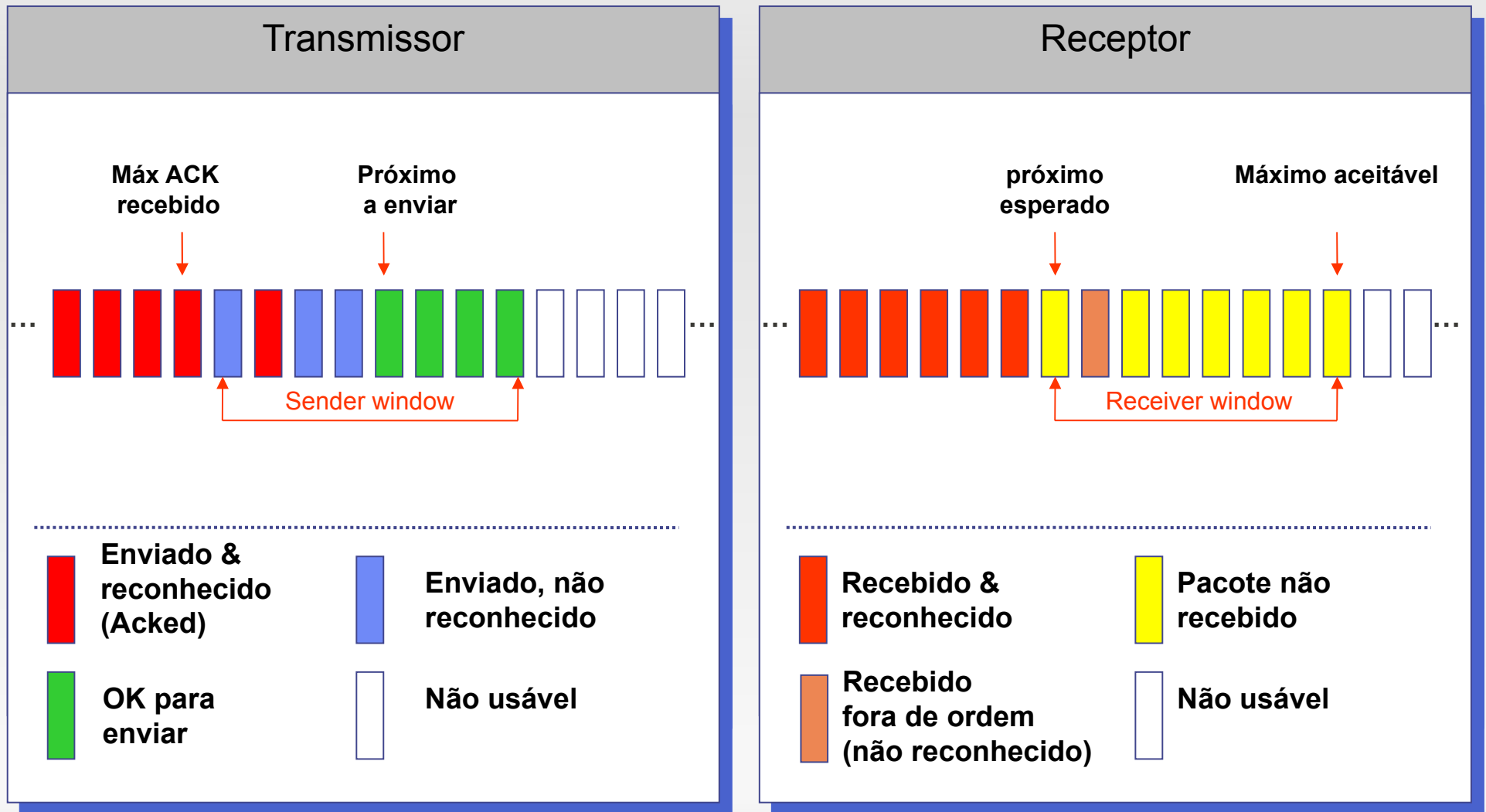
# Controle de Fluxo TCP

- Emissor não deve enviar mais dados do que o receptor é capaz de receber;
- Receptor avisa ao emissor o tamanho da janela (espaço livre no buffer)
  - Campo **RcvWindow** no cabeçalho TCP

## Bufferização do receptor

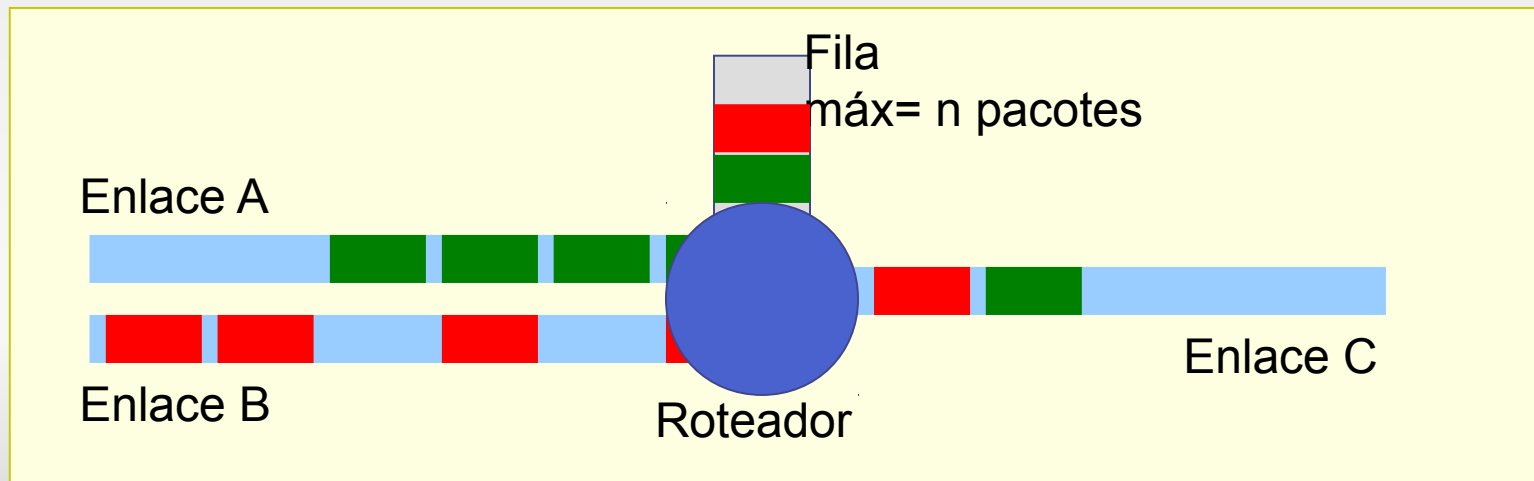


# Janela Deslizante



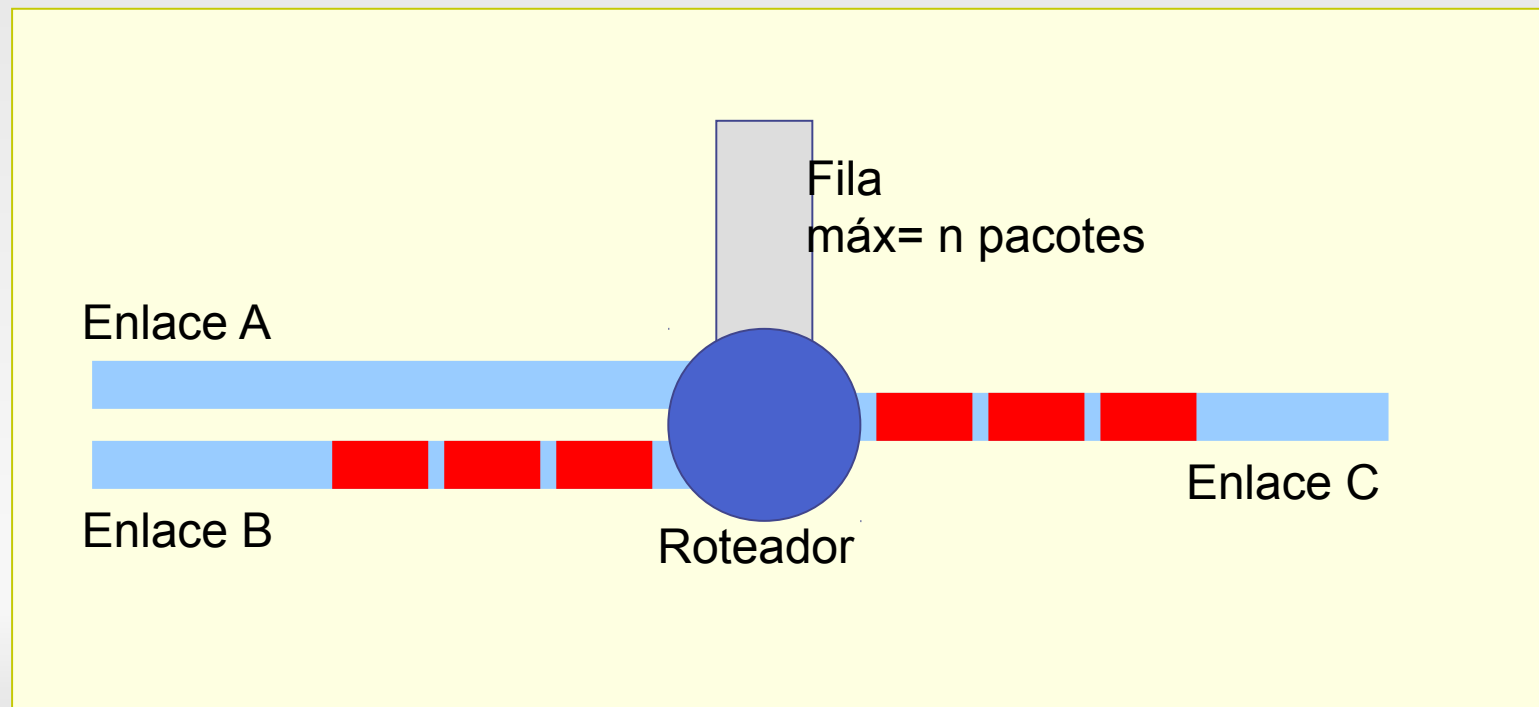
# Congestionamento

- Porque ocorre congestionamento?
  - Congestionamento ocorre quando há muito tráfego na rede.
  - Roteadores tem capacidade de enfileiramento
    - Quando pacotes não podem ser transmitidos naquele momento, eles aguardam na fila até que o roteador possa enviá-los.
    - Filas tem tamanho limitado;
    - Se a fila está cheia, pacotes que chegam são descartados;



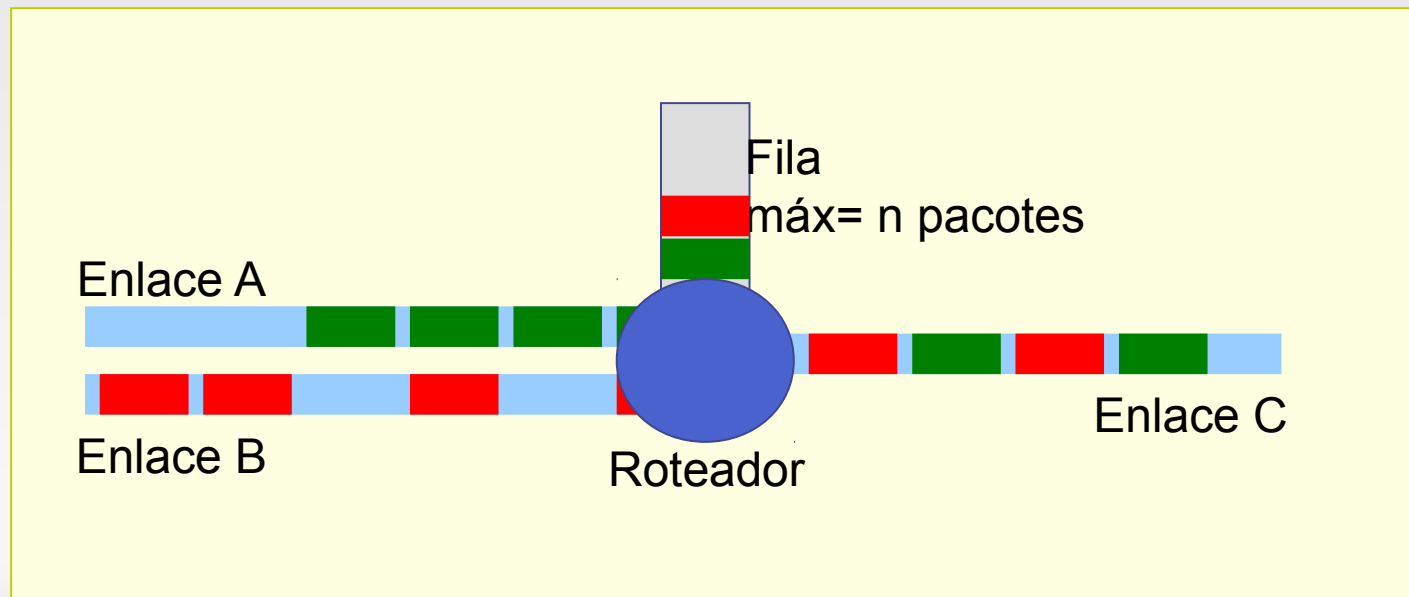
# Congestionamento

- Fluxo de entrada no roteador é  $\leq$  fluxo de saída
  - Fila vazia (buffer do roteador)



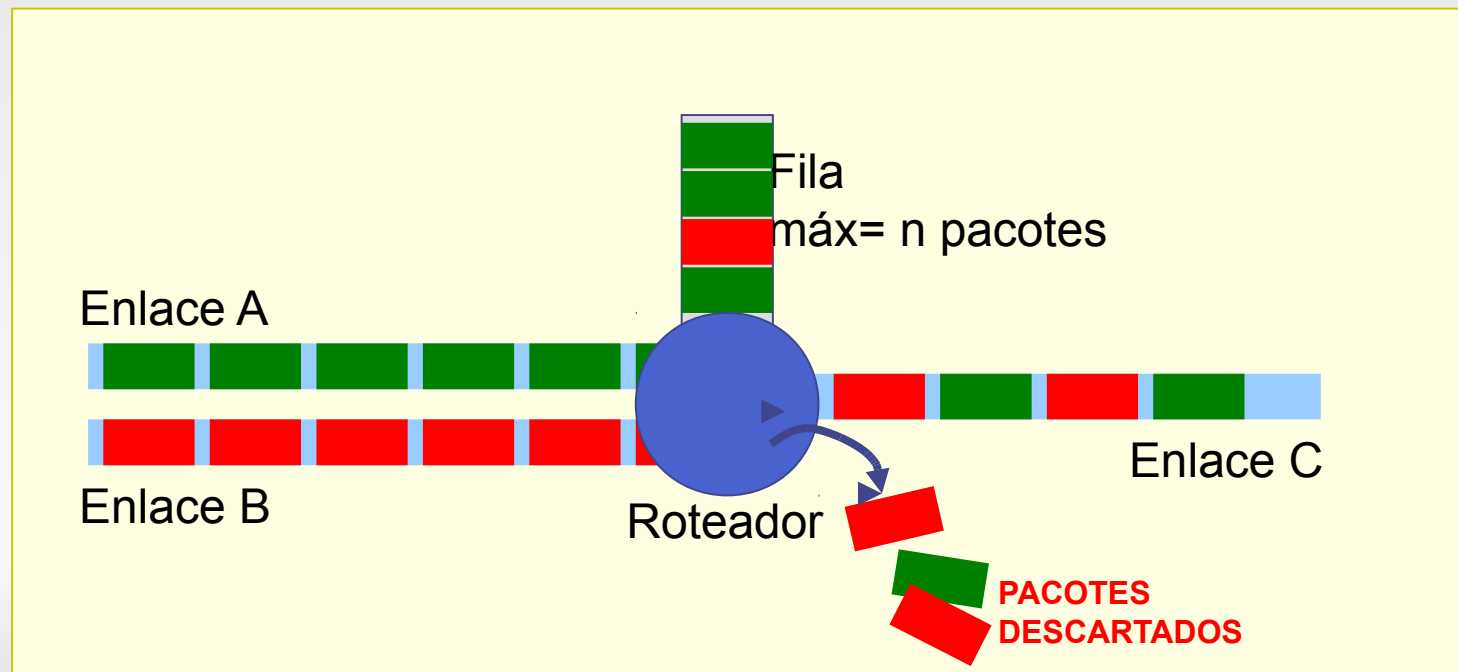
# Congestionamento

- Fluxo de entrada no roteador  $>$  fluxo de saída
  - Fila entra em ação
  - Pacotes sofrerão atrasos
  - Crescimento da fila depende do comportamento da chegada dos pacotes



# Congestionamento

- Fluxo de entrada do roteador é  $>$  fluxo de saída
  - Descartes ocorrem quando a fila está cheia
  - Descartes são geralmente aleatórios
  - Mas existem políticas de prioridade de pacotes nas filas.



# Filas

- Filas com alta ocupação implica:
  - Aumento de atraso;
  - Aumento de variação de atraso;
  - Aumento da probabilidade de perda de pacotes.



# Colapso por Congestionamento

- Congestionamento tende a ficar pior se não controlado
- Transferência de dados sofrem atrasos
- Pacotes são perdidos
- Protocolos tentam retransmitir pacotes
- Usuários tentam retransmitir ou pedir a informação novamente
- Enviar dados para uma rede que enfrenta congestionamento aumenta o congestionamento
- Dados úteis na rede são cada vez menores

# Colapso

- É quando o aumento de carga de tráfego da rede resulta em redução do tráfego útil executado
- Possíveis causas
  - Retransmissão desnecessária de pacotes ainda a caminho
  - Pacotes não entregues
    - Pacotes consomem recursos da rede e depois são descartados;
  - Tráfego de controle
    - Alto percentual de tráfego de controle
  - Pacotes atrasados ainda em fila

# Lidando com o Congestionamento

- Para lidar com o congestionamento precisamos:
  - Uma forma de determinar se a rede está ficando congestionada ou já está congestionada.
  - Um algoritmo para reduzir a taxa de transmissão em momentos de congestionamento
  - Um mecanismo para aumentar a taxa de transmissão quando o congestionamento terminar
- É necessário ter algum feedback da rede
- É muito difícil saber onde é exatamente o congestionamento.
  - Não precisamos saber onde está, basta saber que está havendo congestionamento.

# Detectando Congestionamento

- Duas Abordagens
  - Explícita: a rede diz a v você
  - Implícito: o transmissor infere usando estatísticas de tráfego.
- Na Internet:
  - TCP usa perda de pacotes como indicio de congestionamento (abordagem implícita)
  - TCP ECN (Explicit Congestion Notification) descarte aleatório de pacotes em um roteador.
- Pacotes perdidos sempre indicam congestionamento??

# Controle de Congestionamento no TCP

- Idéia Básica:
  - Não há maneira do TCP determinar exatamente as condições da rede.
  - TCP considera pacotes perdidos como congestionamento.
- Controla a transmissão com algoritmos simples
- Se pacotes não são perdidos:
  - TCP assume que a rede não está congestionada
  - Aumenta a taxa de transmissão
- Se pacotes são perdidos:
  - TCP assume que a rede está congestionada
  - Reduz a taxa de transmissão
- Reduzir ou aumentar = controle de transmissão

# Como o TCP controla a transmissão?

- Introduz uma variável “congestion window” no esquema de janela deslizante;
- Ajusta o total de dados sendo injetado na rede
- Como determinar o tamanho da janela?
  - $\text{Window Size} = \min(\text{advertised window}, \text{congestion window})$
  - Advertised Window: é usada para controle de fluxo
    - Enviada ao TCP transmissor pelo TCP receptor
  - Congestion Window: É usada para controle de congestionamento
    - Valor determinado pelo transmissor em função da vazão da rede.

# Controle de Congestionamento TCP

- Objetivo

- Fazer com que a rede esteja sempre em plena carga, porém não congestionada.
- Fazer uma aproximação do ideal: introduzir um novo pacote na rede assim que outro saiu
- O TCP testa continuamente a capacidade da rede.

- Método

- Laço fechado com notificação implícita (perda de pacotes indicam situações de congestionamento)

# Janela de Congestionamento

- Uma conexão TCP controla sua taxa de transmissão limitando o número de segmentos que podem ser transmitidos sem que uma confirmação seja recebida.
- Esse número é chamado tamanho da janela TCP ( $w$ )
- Uma janela TCP começa com um pequeno número de  $w$  e então o incrementa arriscando que exista mais largura de banda.
- Isso ocorre até que algum segmento seja perdido.
- Então, a conexão TCP reduz  $w$  para um valor seguro, e continua a arriscar o crescimento.



# Controle de Congestionamento

- O controle é feito através de duas variáveis em cada lado da conexão
  - Janela de Congestionamento
  - Limiar (thershold)
    - Controla o crescimento da janela de congestionamento

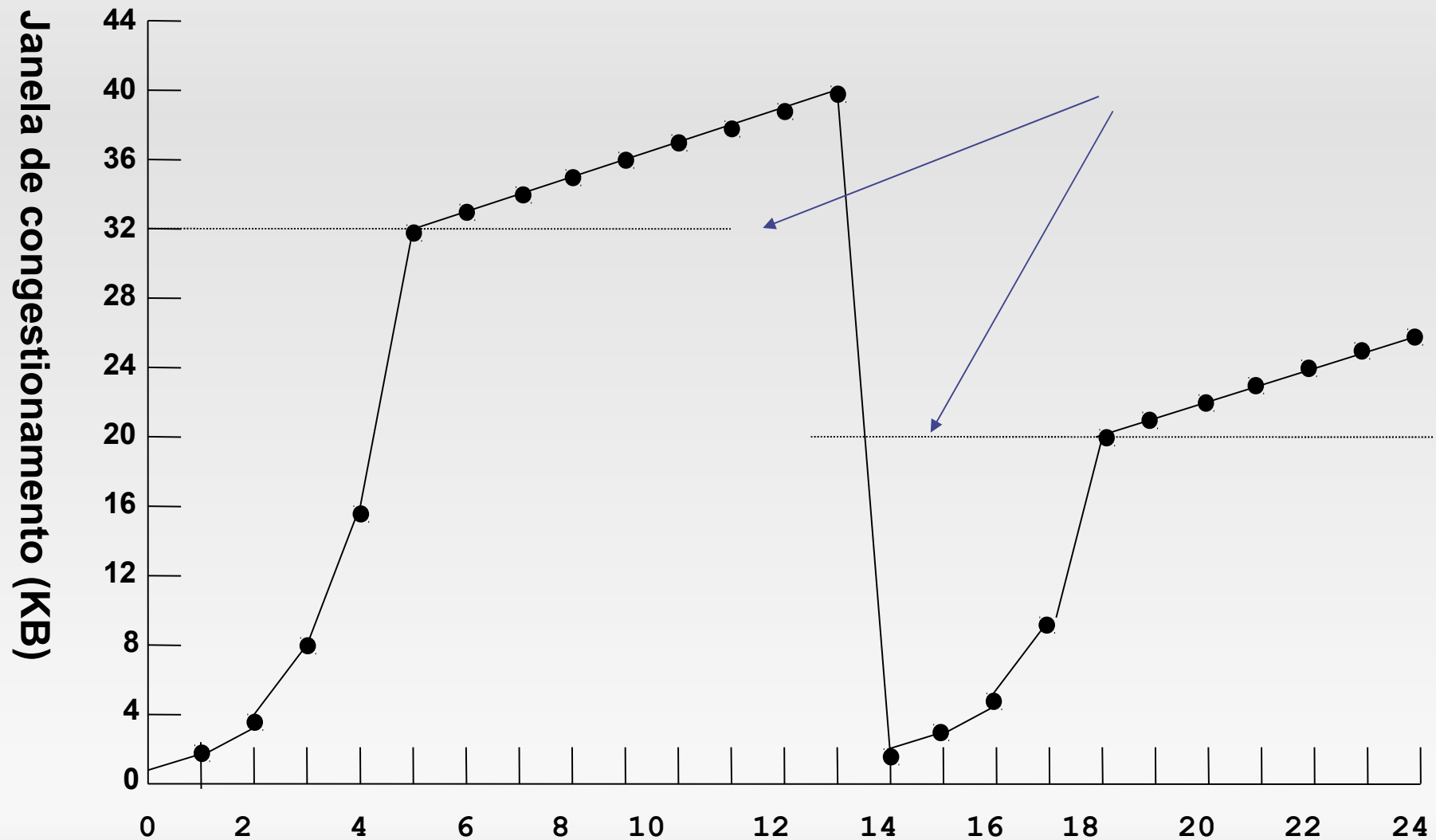
# Controle de Congestionamento TCP

- Algoritmos
  - Slow start (partida lenta) e congestion avoidance (evitar congestionamento)
  - Fast retransmit (retransmissão rápida) e fast recovery (recuperação rápida)
- Detecção da perda de pacotes
  - TCP Tahoe interpreta estouro do temporizador (timeout) e acks duplicados do mesmo modo.
    - Ativa o slow start
  - TCP Reno, trata de forma diferente o recebimento de acks duplicados e o estouro do temporizador.
    - Ativa o fast recovery (acks duplicados)
    - Ativa o slow start (estouro do temporizador)

# Evolução do Fluxo de dados TCP

- No início, a janela de congestionamento tem o tamanho de 1 (um) segmento.
- Tal segmento tem o tamanho do maior segmento suportado (MSS).
- O primeiro segmento é enviado e então é esperado seu reconhecimento (ACK).
- Se o ACK chegar antes que ocorra o timeout, o transmissor duplica o tamanho da janela de congestionamento e envia dois segmentos.
- Se esses dois segmentos também forem reconhecidos antes de seus timeouts, o transmissor duplica novamente sua janela, enviando agora quatro segmentos.

# Evolução do Fluxo de dados TCP



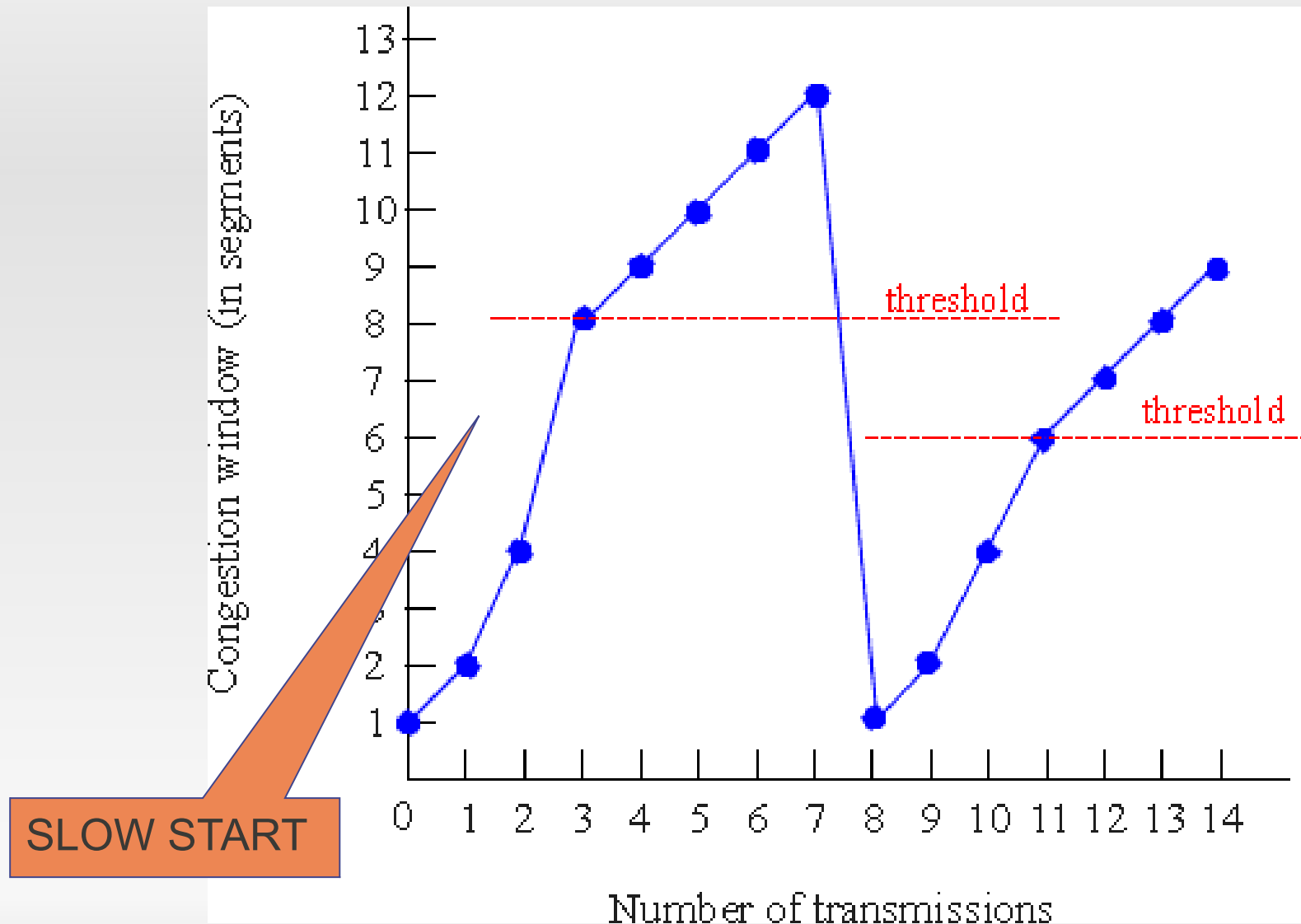
# Evolução de uma conexão TCP

- Esse processo continua até que:
  - O tamanho da janela de congestionamento seja maior que o limiar, ou maior que o tamanho da janela do receptor;
  - Ocorra timeout antes do ACK

# Duas Fases dessa Evolução

- A primeira fase, em que a janela de congestionamento cresce exponencialmente é chamada de inicialização lenta (slow start), pelo fato de começar com um segmento.
- A taxa de transmissão começa pequena porém cresce muito rapidamente (duplicando de tamanho a cada confirmação).

# Graficamente

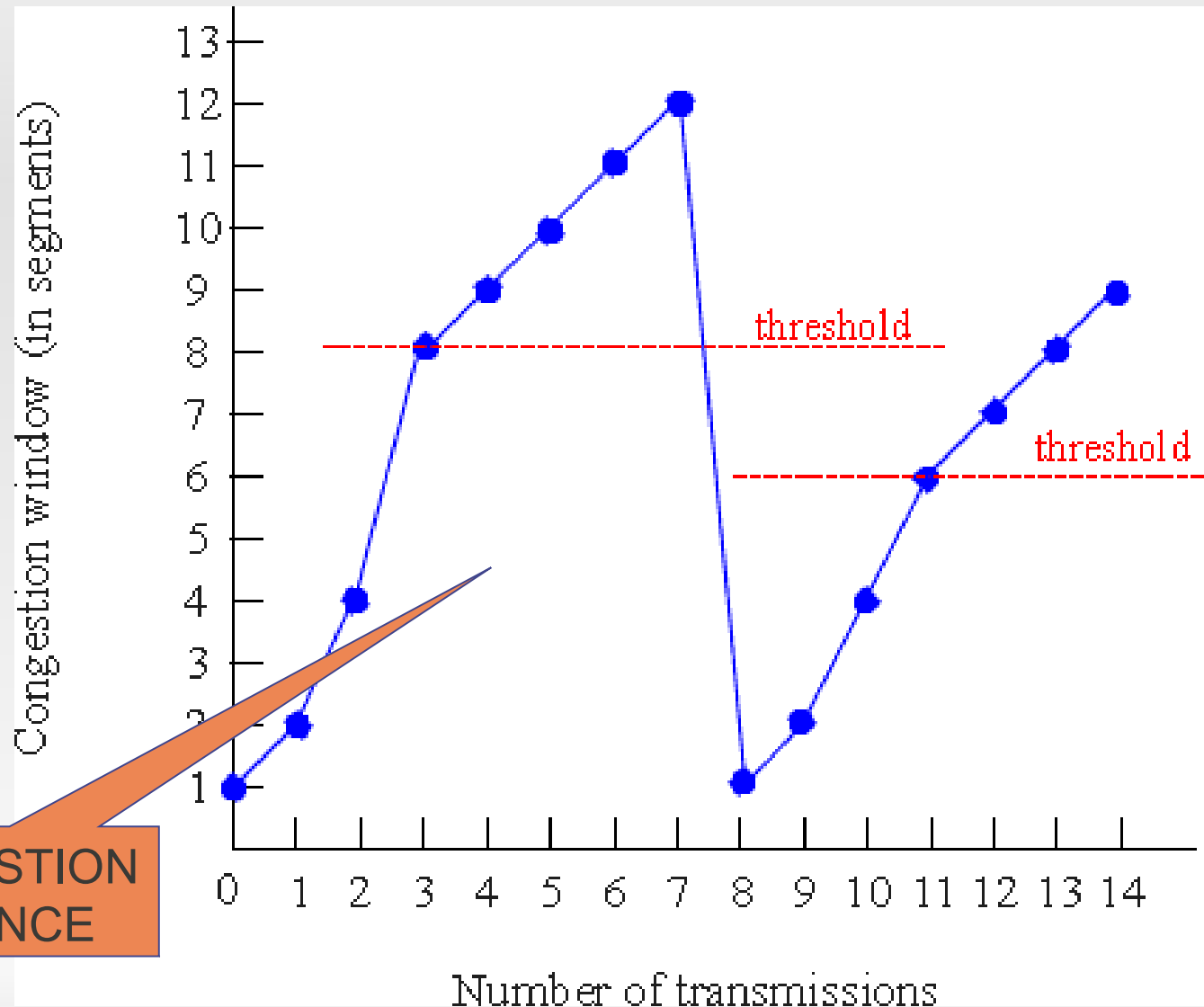


# Duas Fases desta Evolução

- Uma vez ultrapassado o limiar, e a janela do receptor ainda não seja um limitante, o crescimento da janela passa a ser linear.
- Esta fase é chamada de prevenção de congestionamento (congestion avoidance).
- Sua duração também depende da não ocorrência timeouts, e da aceitação do fluxo por parte do receptor.



# Graficamente



CONGESTION  
AVOIDANCE

# Evolução de uma conexão TCP

- Na ocorrência de um timeout, as seguintes atitudes são tomadas:
  - O valor do limiar passa a ser a metade do atual tamanho da janela de congestionamento.
  - O tamanho da janela de congestionamento volta ser do tamanho de um segmento (ativa-se o slow start).
  - O tamanho da janela de congestionamento volta a crescer exponencialmente.

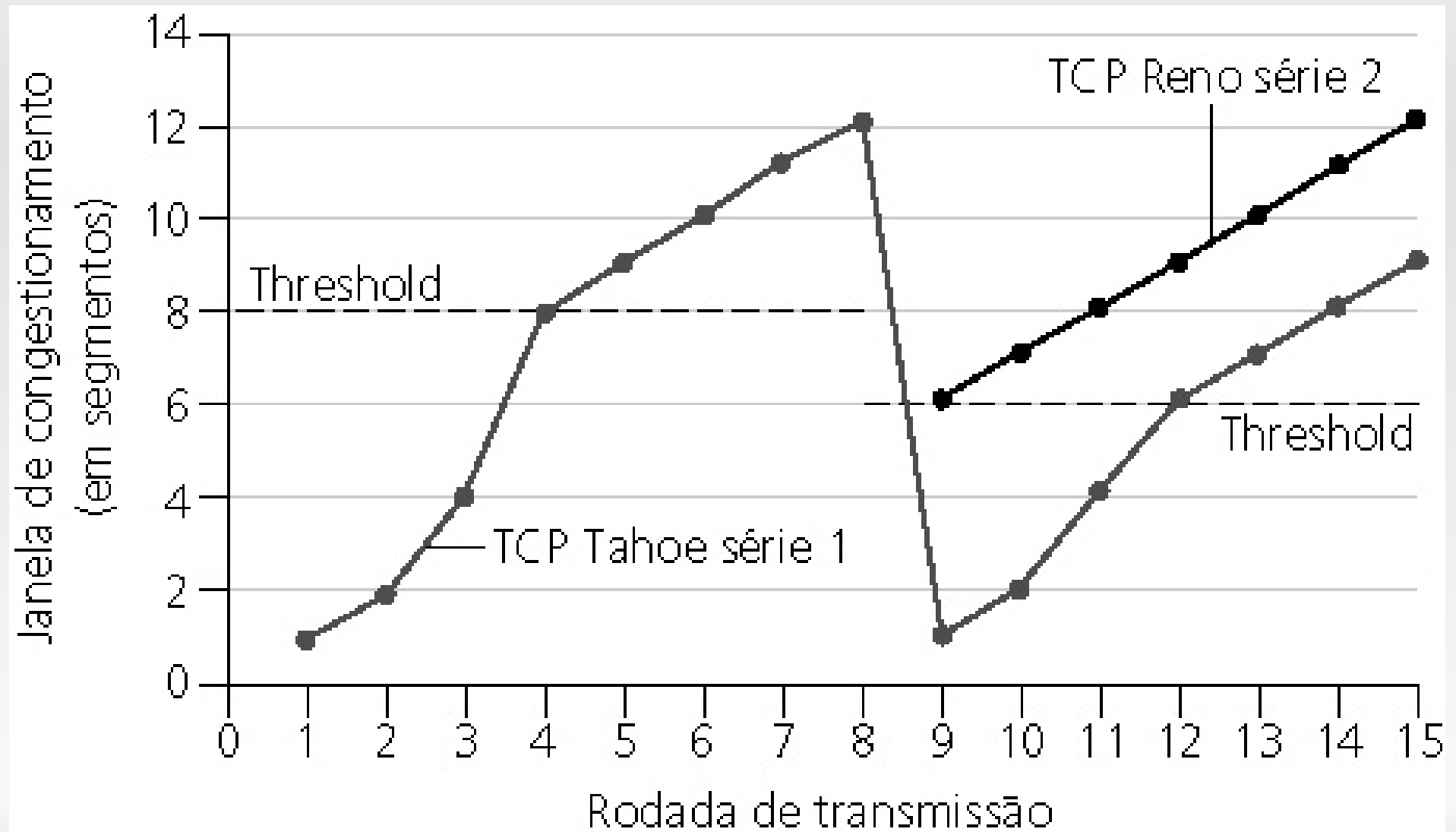
# Resumo

- Quando o tamanho da janela de congestionamento está abaixo do limiar, seu crescimento é exponencial.
- Quando este tamanho está acima do limiar, o crescimento é linear.
- Todas as vezes que ocorrer um timeout, o limiar é modificado para a metade do tamanho da janela e o tamanho da janela passa a ser 1.

# Reação a eventos de ACKs duplicados

- O TCP Reno reage de forma diferente a retransmissão rápida (ACKs duplicados).
- Na ocorrência de retransmissão rápida o TCP Reno reage do seguinte modo (utiliza o algoritmo additive-increase, multiplicative-decrease - AIMD):
  - Diminui a janela de congestionamento pela metade;
  - E então aumenta sua taxa de envio linearmente.
- Mas porque fazer distinção no tratamento de sintomas? O que difere a perda de pacotes por timeout da chegada de ACKs duplicados?

# Gráficamente



# Algoritmo AIMD gráficamente

