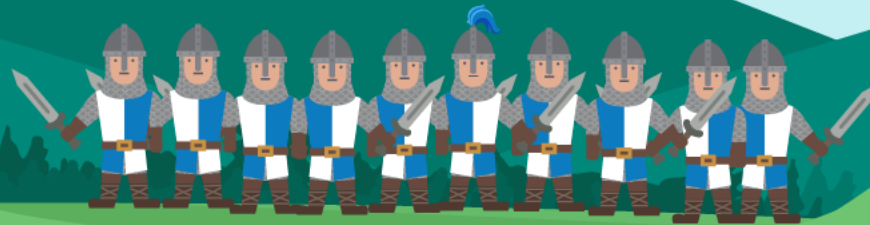


AULA 16

Características do TCP

Desafio dos dois generais



Transporte orientado para conexão: TCP

- ▶ Resumo de mecanismos de transferência confiável de dados e sua utilização:
 - ▶ Soma de verificação - Usada para detectar erros de bits em um pacote transmitido
 - ▶ Temporizador - Usado para controlar a temporização/retransmissão de um pacote, possivelmente porque o pacote (ou seu ACK) foi perdido dentro do canal
 - ▶ Número de sequência - Usado para numeração sequencial de pacotes de dados que transitam do remetente ao destinatário



Transporte orientado para conexão: TCP

► Reconhecimento

- Usado pelo destinatário para avisar o remetente que um pacote ou conjunto de pacotes foi recebido corretamente.

► Reconhecimento negativo

- Usado pelo destinatário para avisar o remetente que um pacote não foi recebido corretamente

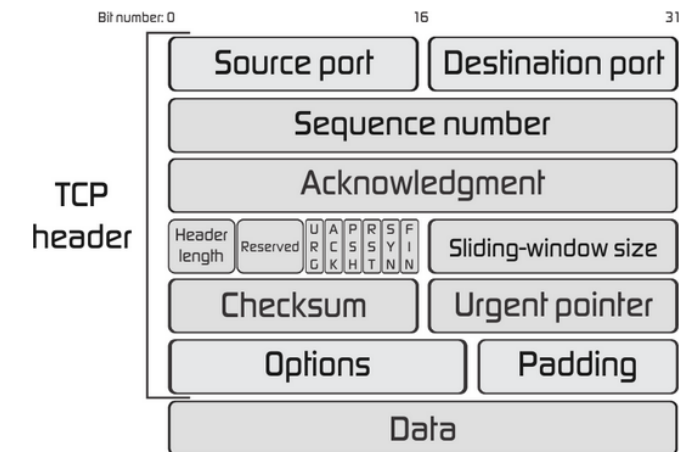
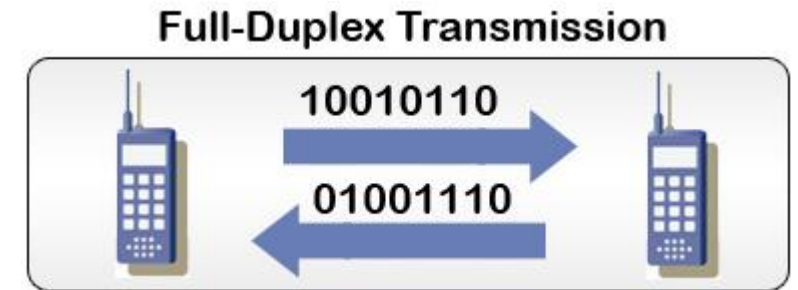
► Janela, paralelismo

- O remetente pode ficar restrito a enviar somente pacotes com números de sequência que caiam dentro de uma determinada faixa

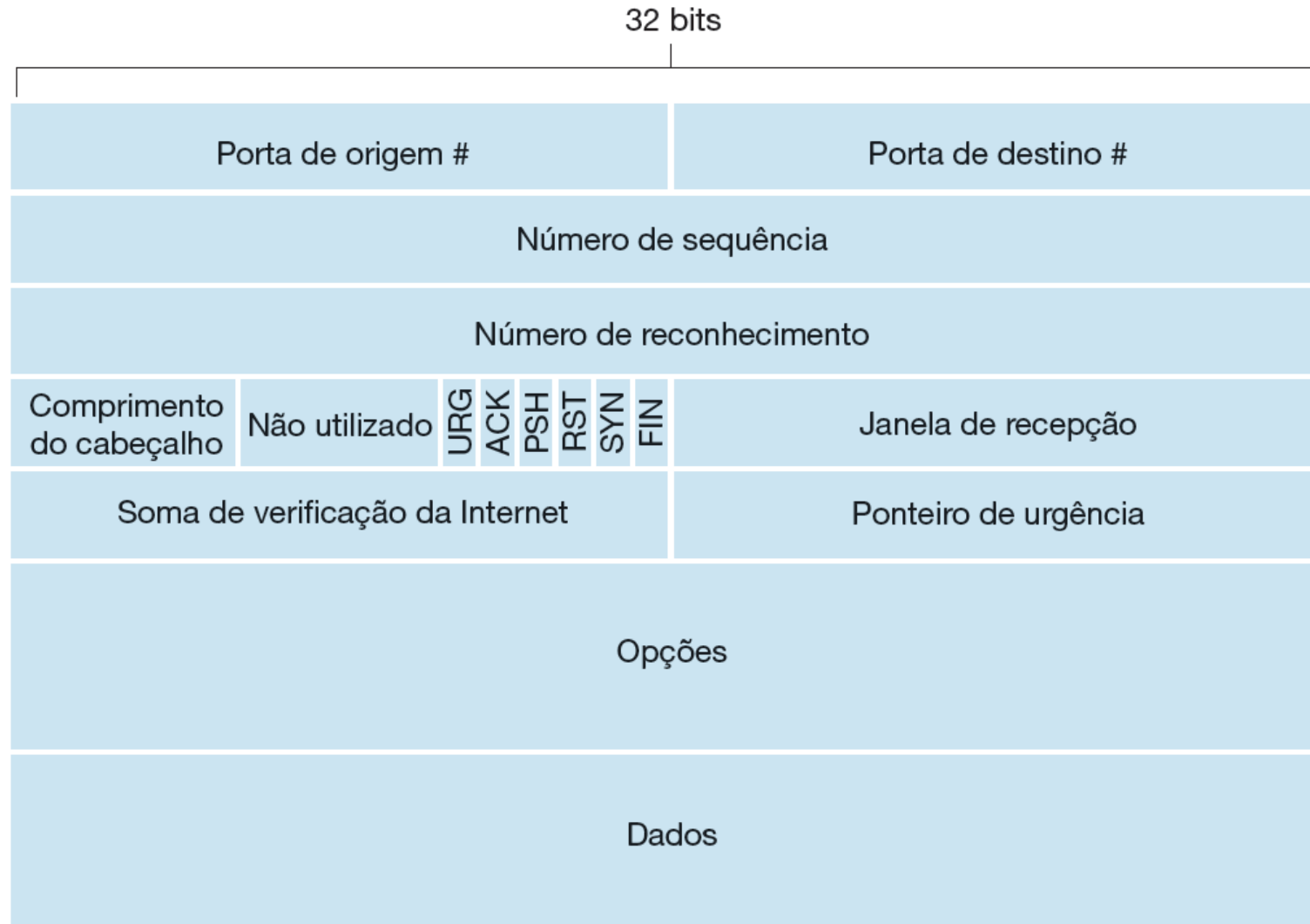


A conexão TCP

- ▶ Uma conexão TCP provê um serviço full-duplex
 - ▶ A conexão TCP é sempre ponto a ponto
 - ▶ Uma vez estabelecida uma conexão TCP, dois processos de aplicação podem enviar dados um para o outro
- ▶ O TCP combina cada porção de dados do cliente com um cabeçalho TCP,
 - ▶ segmentos TCP

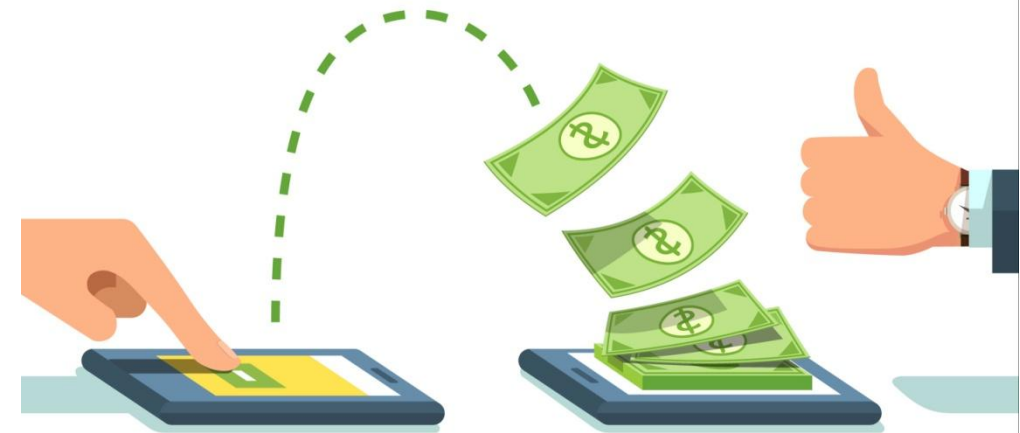


Segmento TCP



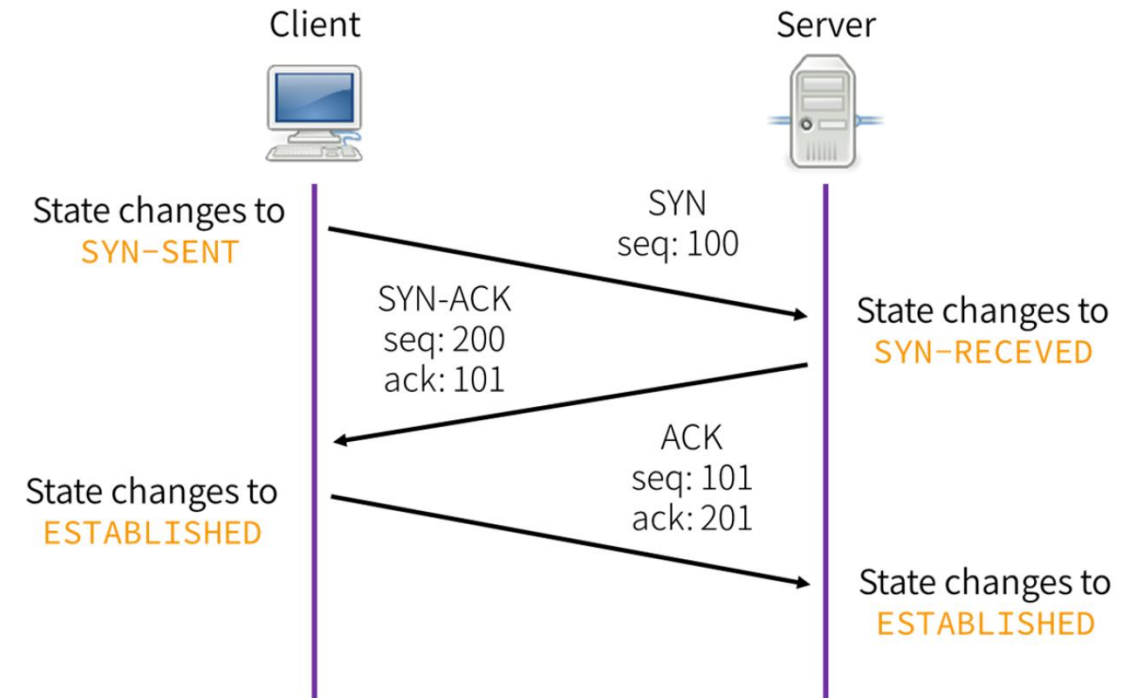
Transferência confiável de dados

- ▶ O TCP cria um serviço de transferência confiável de dados sobre o serviço de melhor esforço do IP
- ▶ O serviço de transferência garante que a cadeia de bytes é idêntica à cadeia de bytes enviada pelo sistema final que está do outro lado da conexão.
- ▶ Os procedimentos recomendados no [RFC 6298] para gerenciamento de temporizadores TCP utilizam apenas um único temporizador de retransmissão

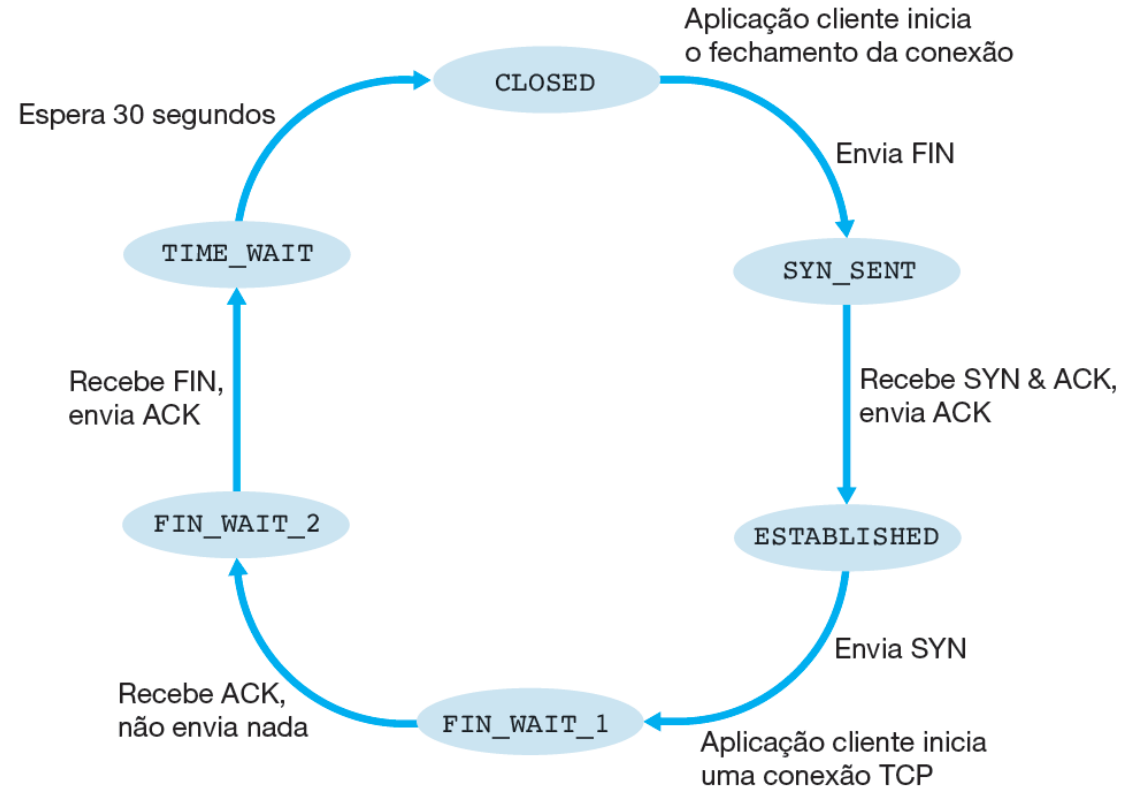


Three-way handshake

- ▶ Estabelecer uma conexão TCP normal requer três etapas separadas
 - ▶ O primeiro host (Client) envia ao segundo host (Server) uma mensagem "sincronizar" (SYN) com seu próprio número de sequência (e.g., 100), que Bob recebe
 - ▶ Server responde com uma mensagem de reconhecimento de sincronização (SYN-ACK) com seu próprio número de sequência (e.g., 200) e o número de confirmação $x + 1$ (e.g., 101), que o Client recebe
 - ▶ O Client responde com uma mensagem de confirmação (ACK) com o número de confirmação $y + 1$ (e.g., 201)



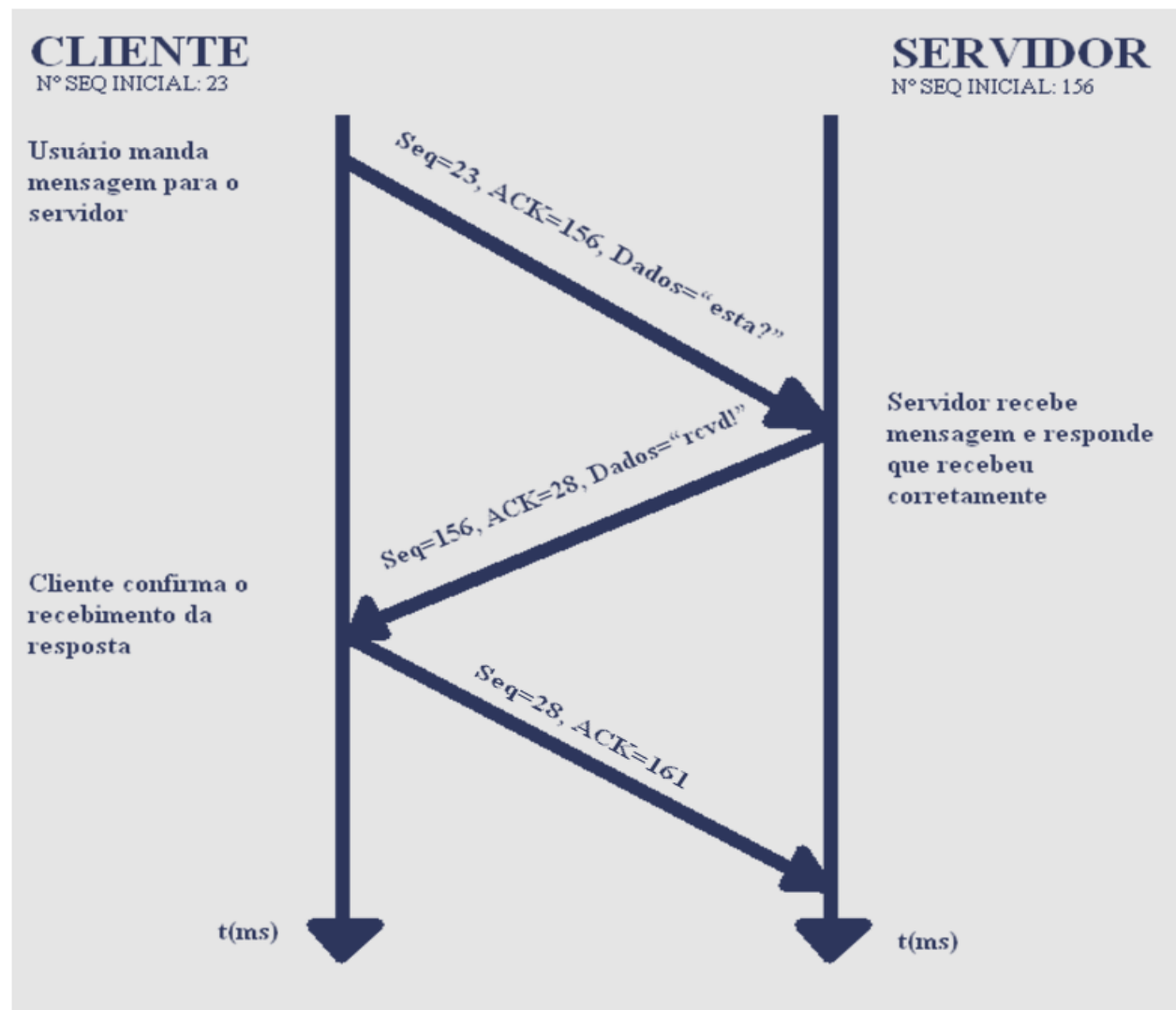
Three-way handshake



As mensagens de sincronização agem como solicitações de serviço de um servidor para o outro, enquanto as mensagens de confirmação retornam ao servidor solicitante para informar que a mensagem foi recebida

Números de sequência e números de reconhecimento

- ▶ O número de sequência
 - ▶ o número do primeiro byte do segmento
- ▶ O número de reconhecimento
 - ▶ o hospedeiro A atribui a seu segmento o número de sequência do próximo byte que ele estiver aguardando do hospedeiro B
- ▶ Como o TCP somente reconhece bytes até o primeiro byte que estiver faltando na cadeia, dizemos que o TCP provê reconhecimentos cumulativos





Controle de Fluxo

Controle de Fluxo

- ▶ O TCP provê um serviço de controle de fluxo às suas aplicações, para eliminar a possibilidade de o remetente estourar o buffer do destinatário
- ▶ Controle de fluxo é um serviço de compatibilização de velocidades
- ▶ O TCP oferece serviço de controle de fluxo fazendo que o remetente mantenha uma variável denominada janela de recepção



Controle de Fluxo

- ▶ A janela de recepção (rwnd) e o buffer de recepção (RcvBuffer)



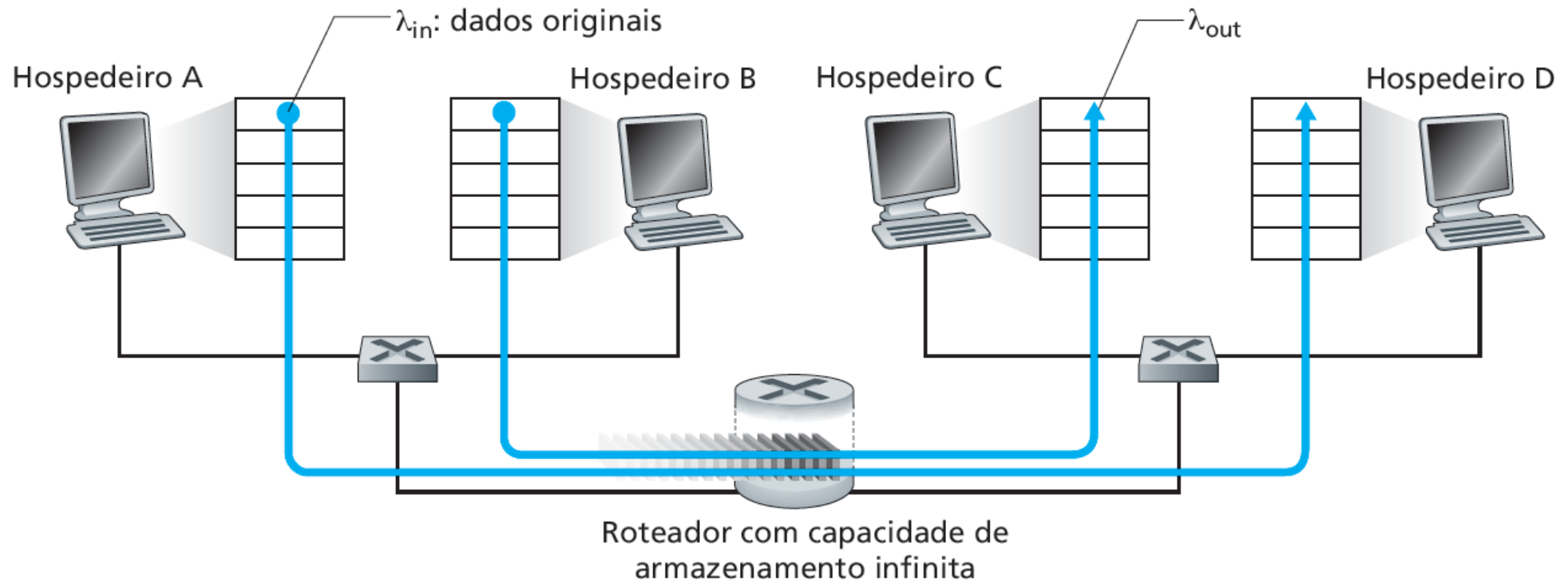


Controle de Congestionamento

Princípios de controle de congestionamento

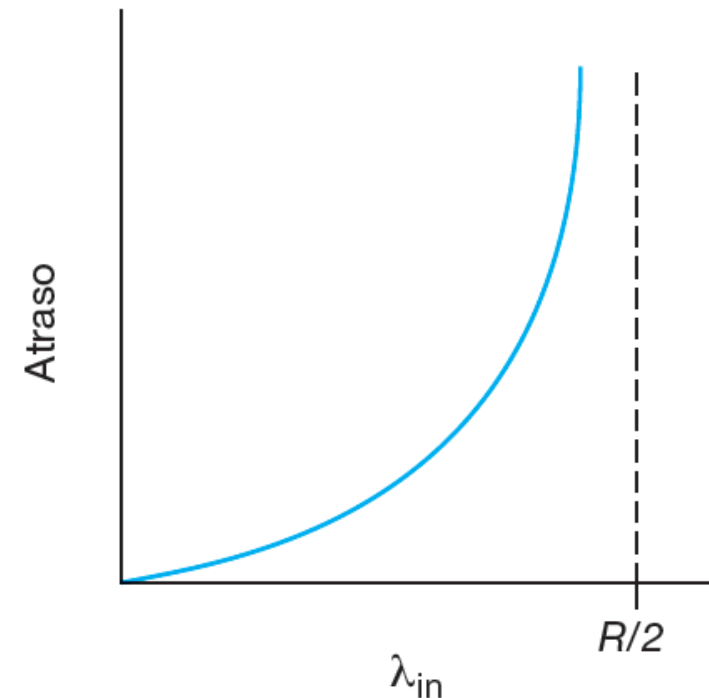
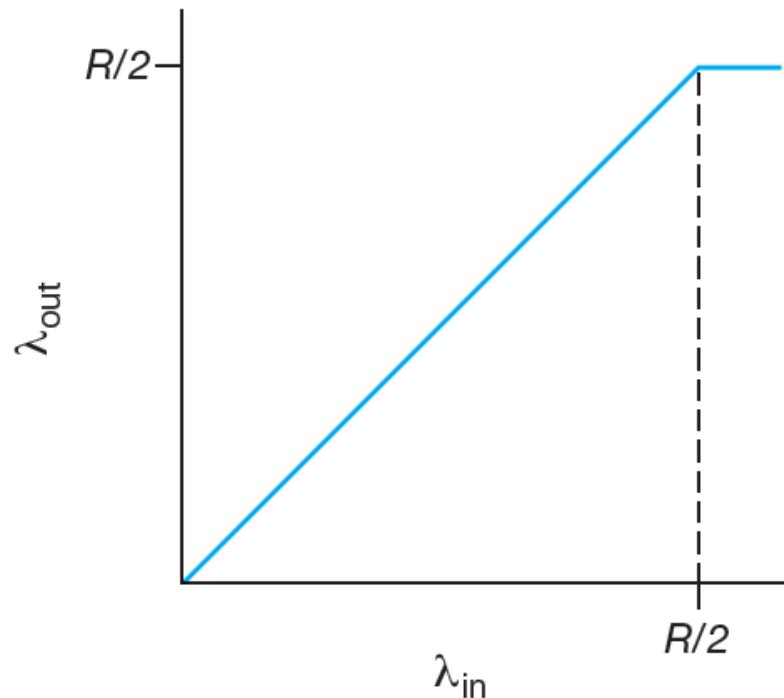
► As causas e os custos do congestionamento

- Cenário de congestionamento I: duas conexões compartilhando um único roteador com número infinito de buffers.



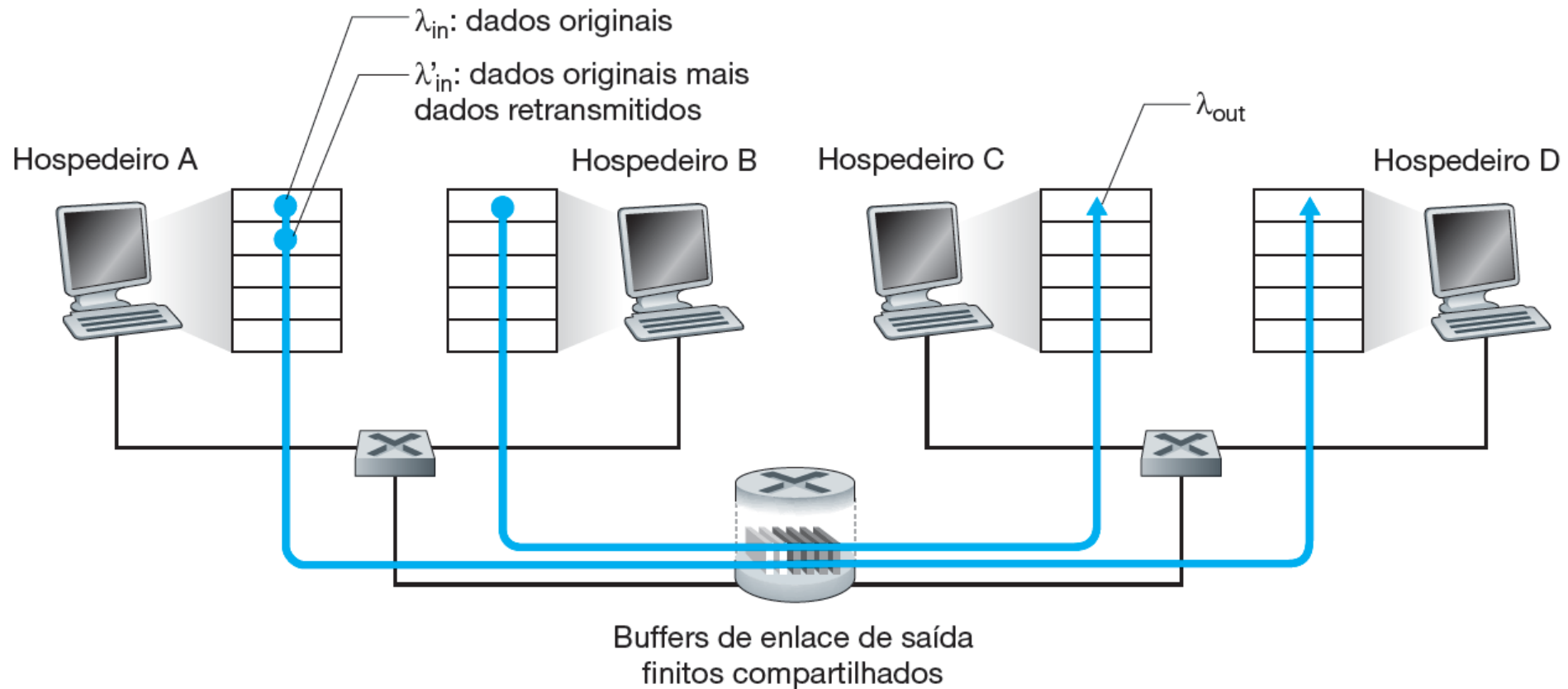
Princípios de controle de congestionamento

- ▶ As causas e os custos do congestionamento
 - ▶ Cenário de congestionamento I: vazão e atraso em função da taxa de envio do hospedeiro.



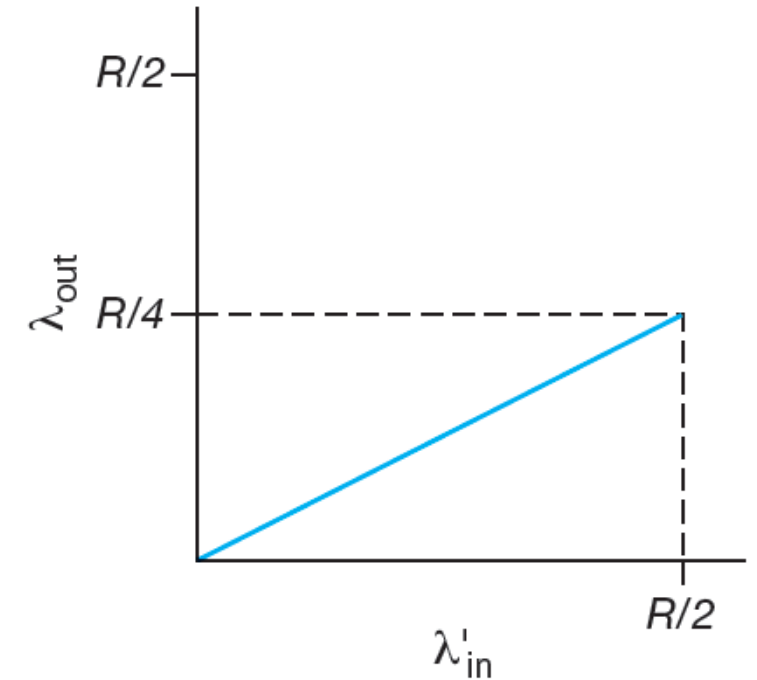
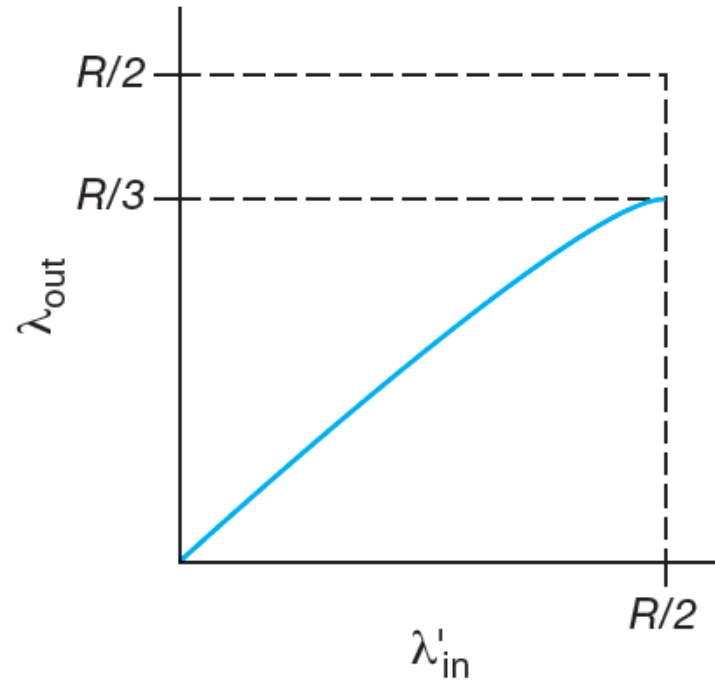
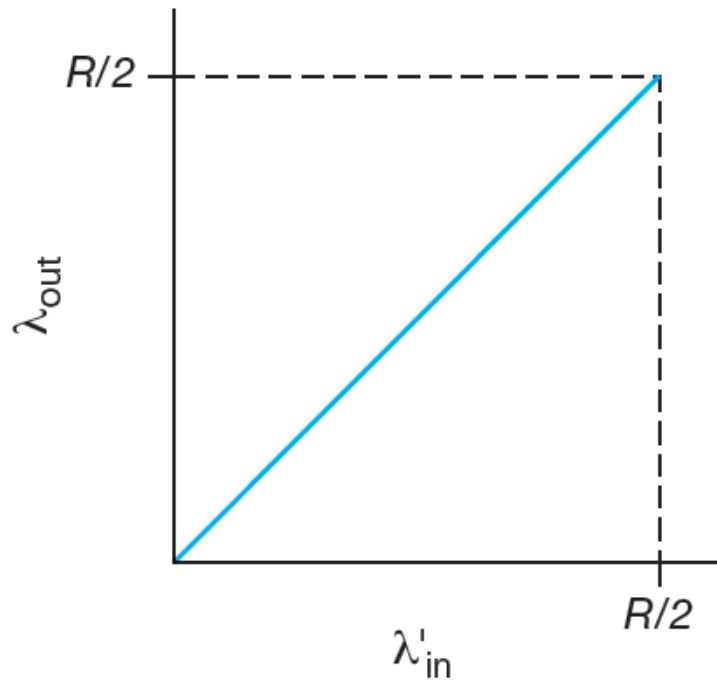
Princípios de controle de congestionamento

- ▶ As causas e os custos do congestionamento
 - ▶ Cenário 2: dois hospedeiros (com retransmissões) e um roteador com buffers finitos



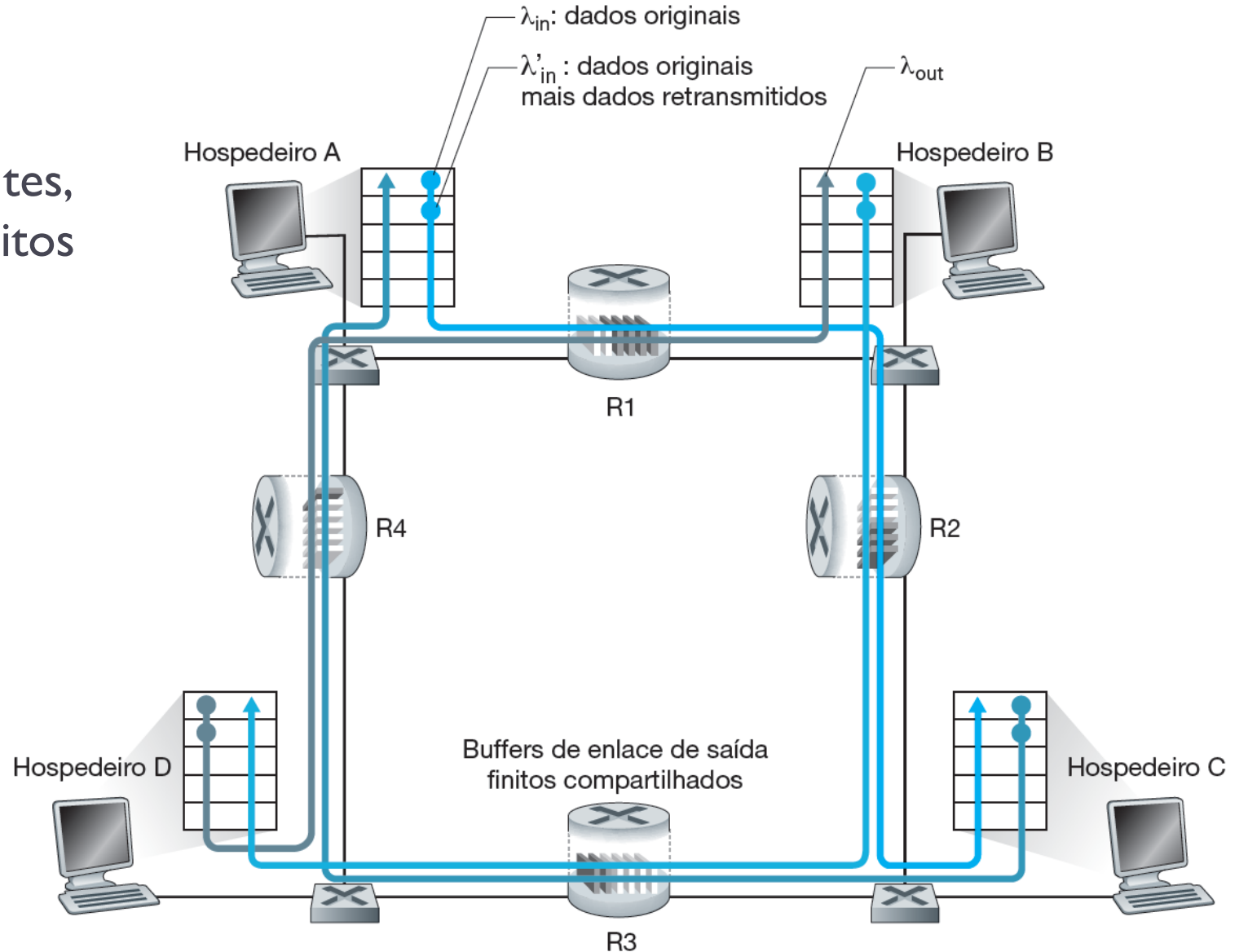
Princípios de controle de congestionamento

- ▶ As causas e os custos do congestionamento
 - ▶ Cenário de congestionamento 2: Desempenho com buffers finitos



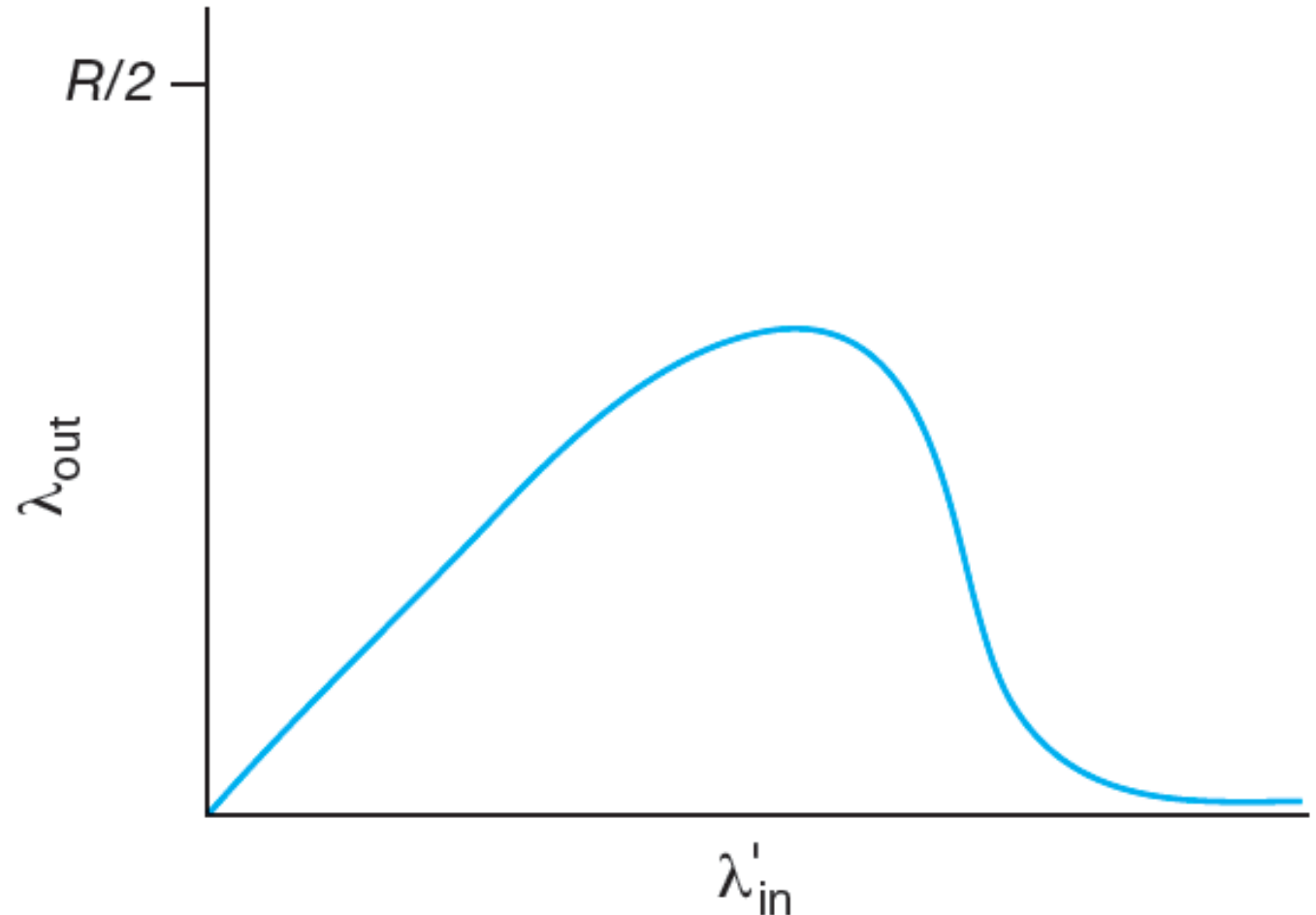
Princípios de controle de congestionamento

- ▶ As causas e os custos do congestionamento
 - ▶ Cenário 3: quatro remetentes, roteadores com buffers finitos e trajetos com múltiplos roteadores



Princípios de controle de congestionamento

- ▶ As causas e os custos do congestionamento
 - ▶ Cenário de congestionamento 3: desempenho obtido com buffers finitos e trajetos com múltiplos roteadores.



Tipos de controle de congestionamento

- ▶ Controle de congestionamento fim a fim
 - ▶ A camada de rede não fornece nenhum suporte explícito à camada de transporte com a finalidade de controle de congestionamento.
- ▶ Controle de congestionamento assistido pela rede
 - ▶ Os componentes da camada de rede (isto é, roteadores) fornecem retroalimentação específica de informações ao remetente a respeito do estado de congestionamento na rede.



Controle de congestionamento do TCP

- ▶ A abordagem adotada pelo TCP é obrigar cada remetente a limitar a taxa à qual enviam tráfego para sua conexão como uma função do congestionamento de rede percebido
- ▶ Se um remetente TCP perceber que há pouco congestionamento no caminho entre ele e o destinatário, aumentará sua taxa de envio
- ▶ Se perceber que há congestionamento, reduzirá sua taxa de envio



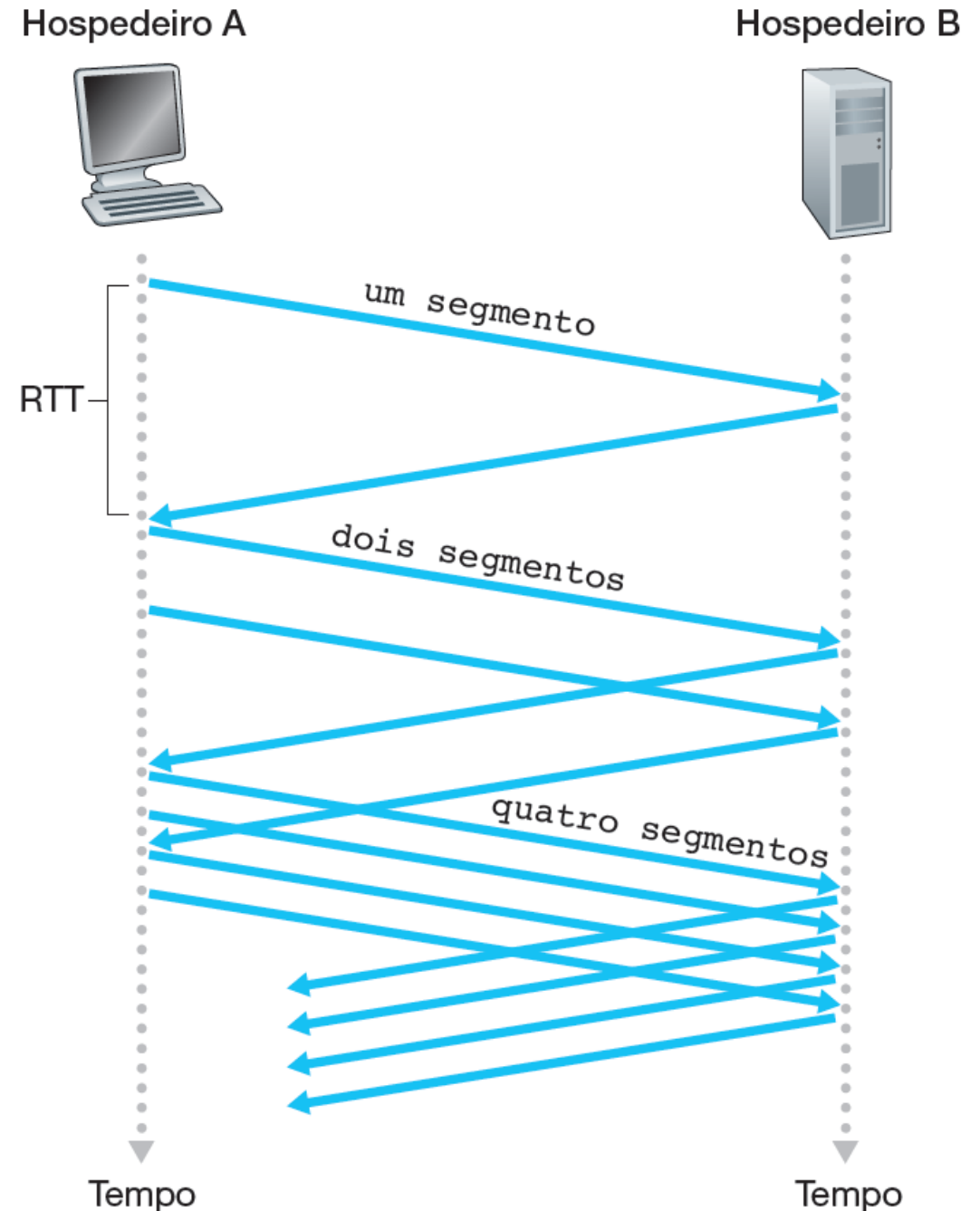
Controle de congestionamento do TCP

- ▶ Mas essa abordagem levanta três questões:
 - ▶ Como um remetente TCP limita a taxa pela qual envia tráfego para sua conexão?
 - ▶ Como um remetente TCP percebe que há congestionamento entre ele e o destinatário?
 - ▶ Que algoritmo o remetente deve utilizar para modificar sua taxa de envio como uma função do congestionamento fim a fim percebido?

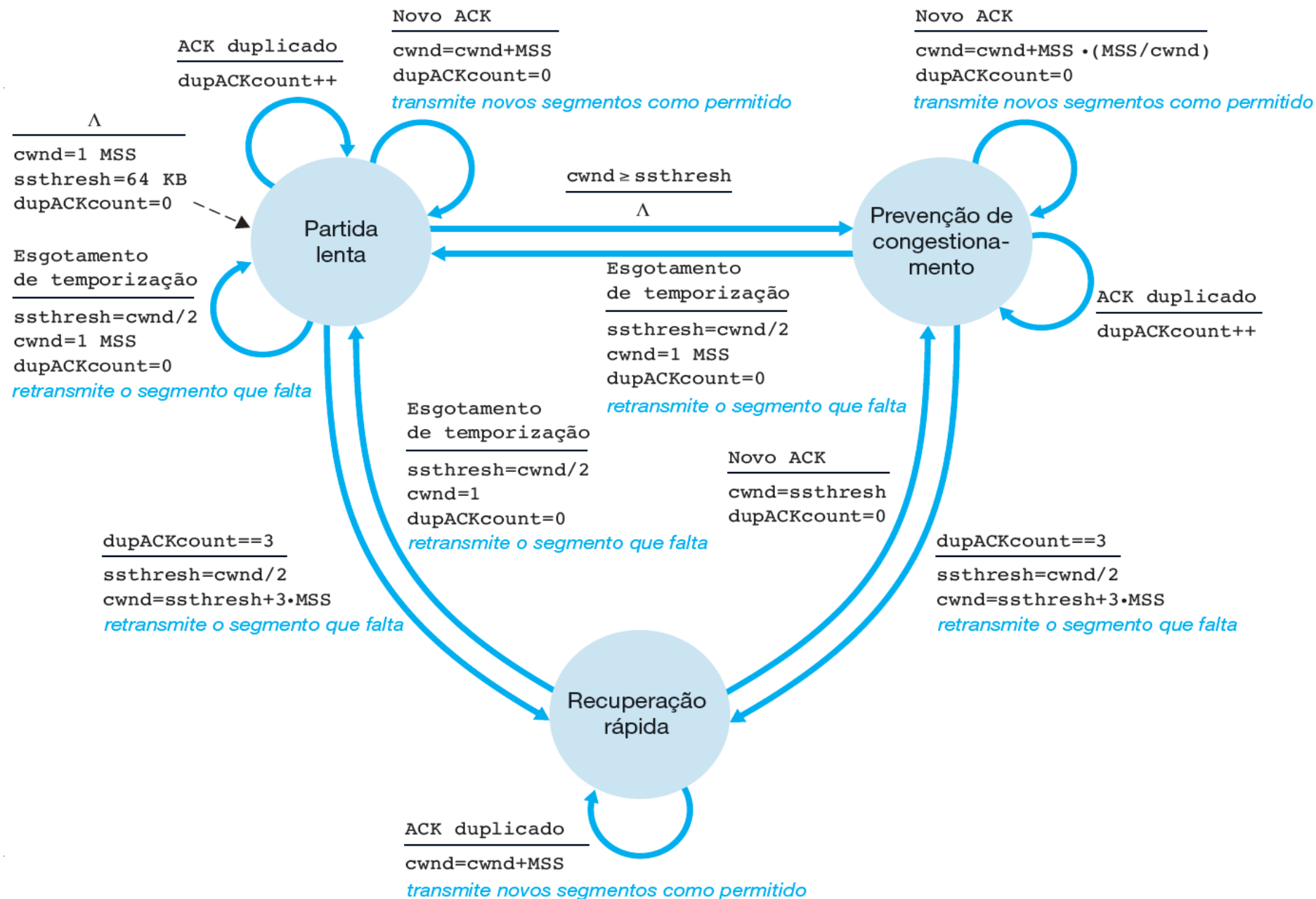


Partida Lenta

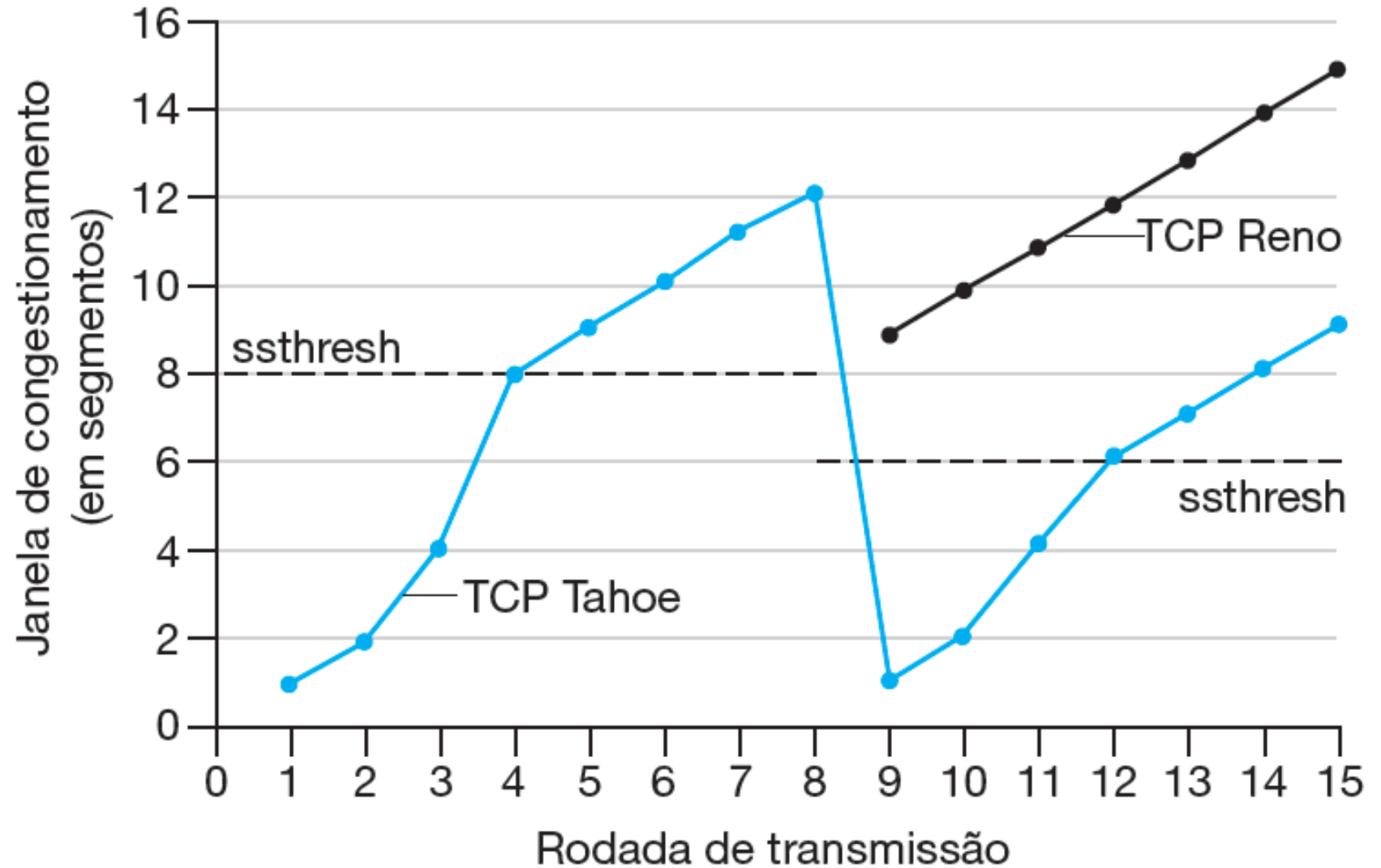
- ▶ Aumenta a velocidade em progressão geométrica
 - ▶ Inicia com um único segmento
- ▶ Deve reduzir a janela caso seja detectado um timeout ou acks duplicados



Diagrama



Recuperação rápida



Comportamento do TCP : Dentes de Serra

