Redes de computadores

Camada de transporte

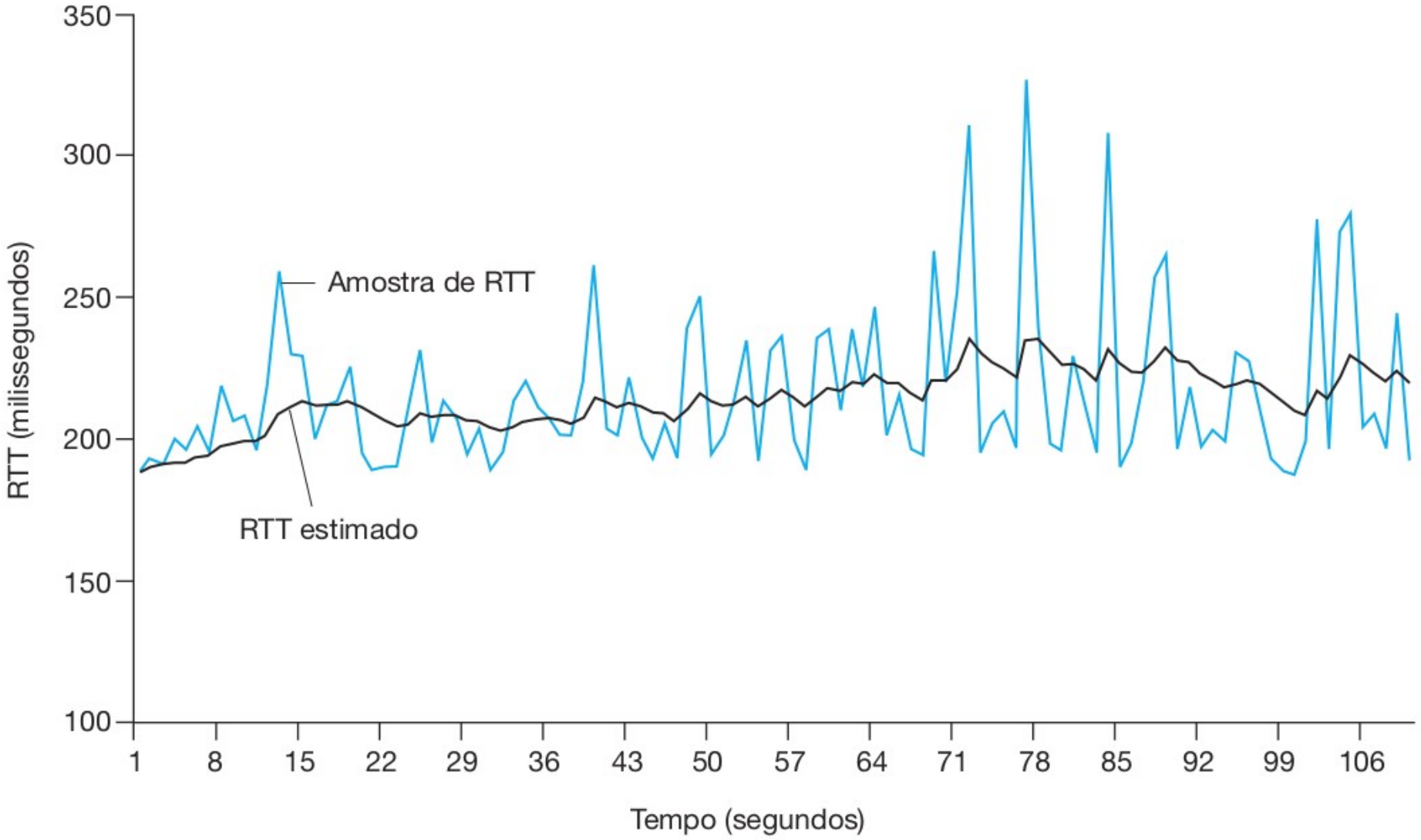
Prof. Luís Eduardo Tenório Silva
luis.silva@garanhuns.ifpe.edu.br

INSTITUTO FEDERAL
Pernambuco
Campus Garanhuns

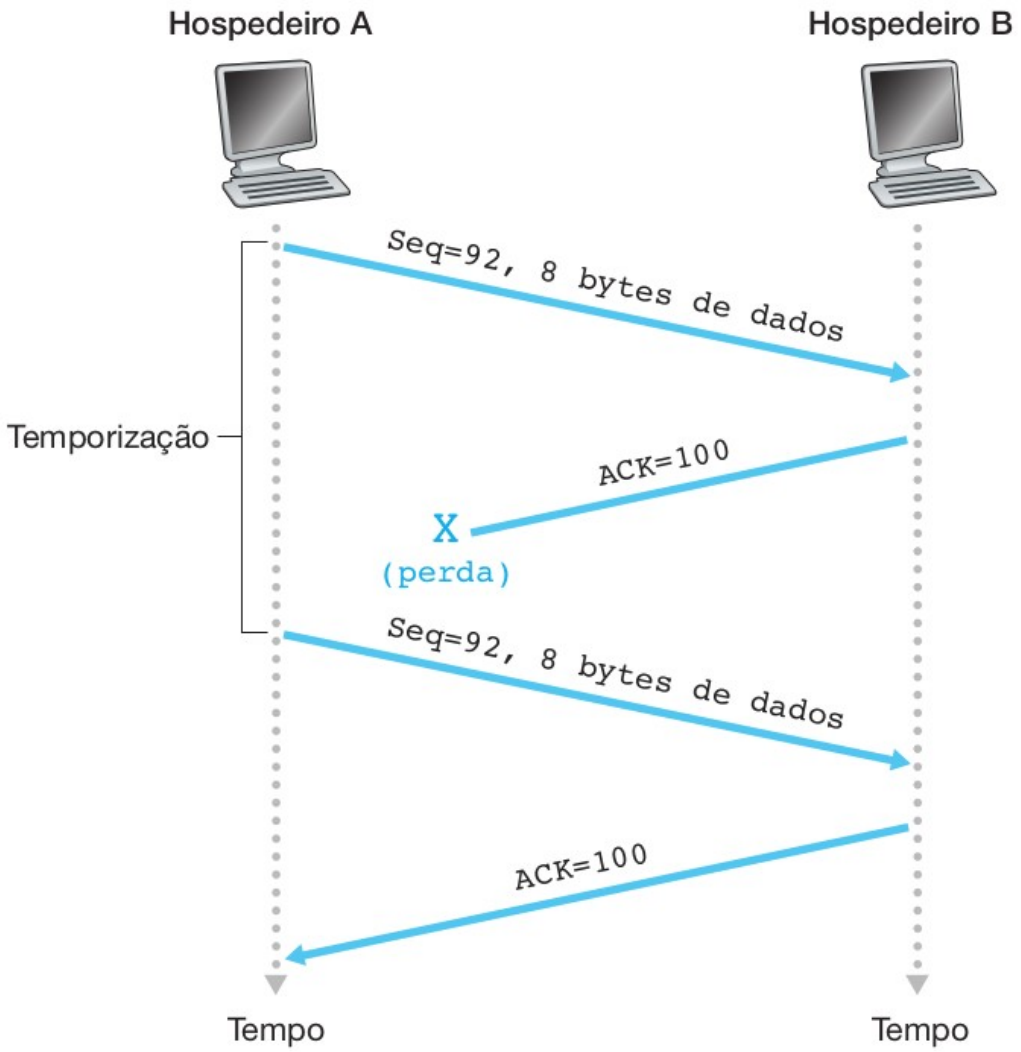
Estimativa de RTT

- RTT (*Round trip time*): Tempo de ida e volta de um pacote;
- TCP utiliza um mecanismo de **controle de temporização para retransmissão de dados**;
  - Tempo de retransmissão deverá ser **superior ao RTT**;
- Tempo de retransmissão de dados é estimado baseado na **média** de RTTs de cada segmento não repetido (SampleRTT);
- A média é utilizada para remover **variações** decorrente de atrasos nos sistemas intermediários
- Fórmula para estimar RTT (*RFC 6298*):
  - $EstimatedRTT = (1 - \alpha) \bullet EstimatedRTT + \alpha \bullet SampleRTT$
  - Onde:
    - $\alpha = 0{,}125$ (ou 1/8)

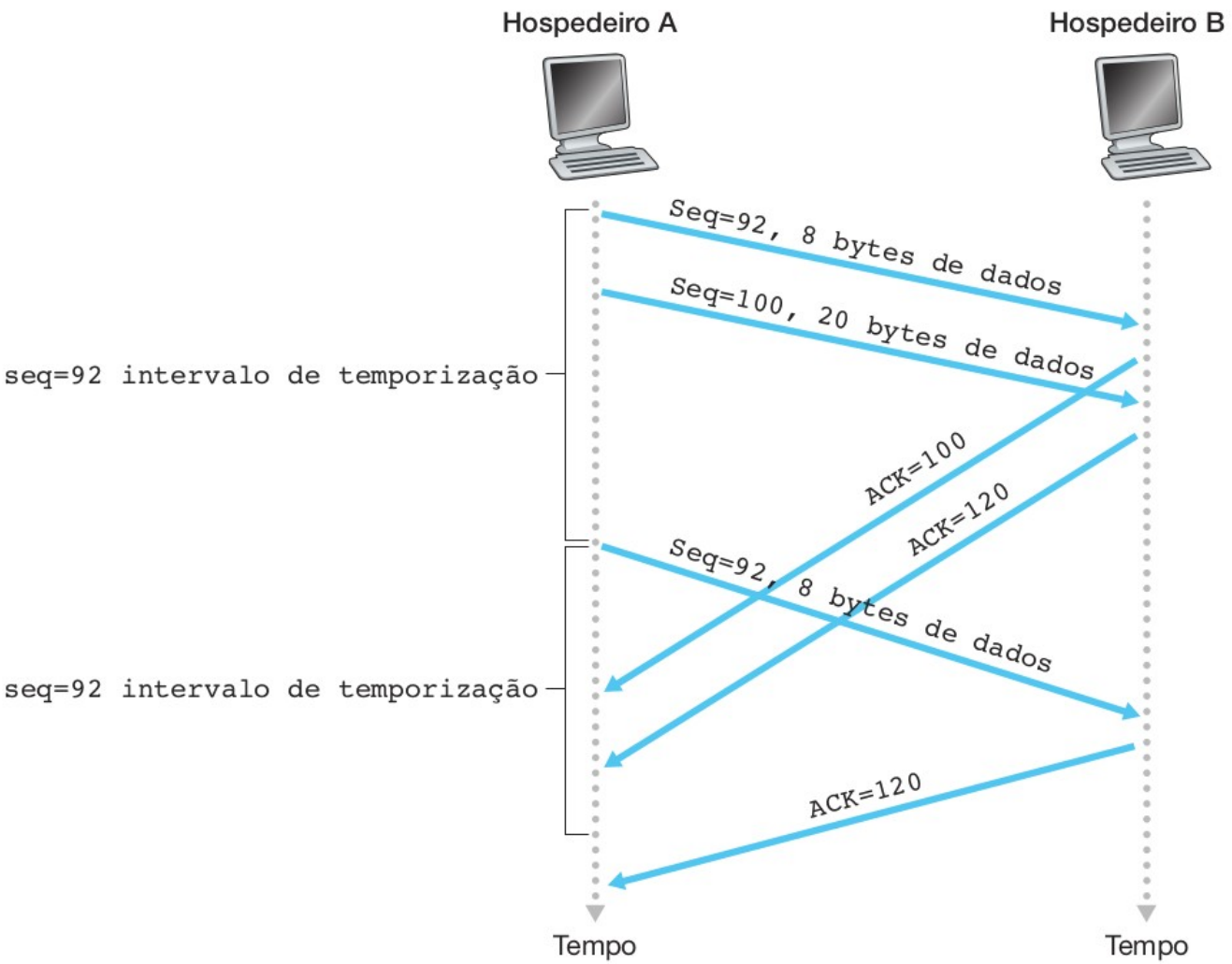
Estimativa de RTT



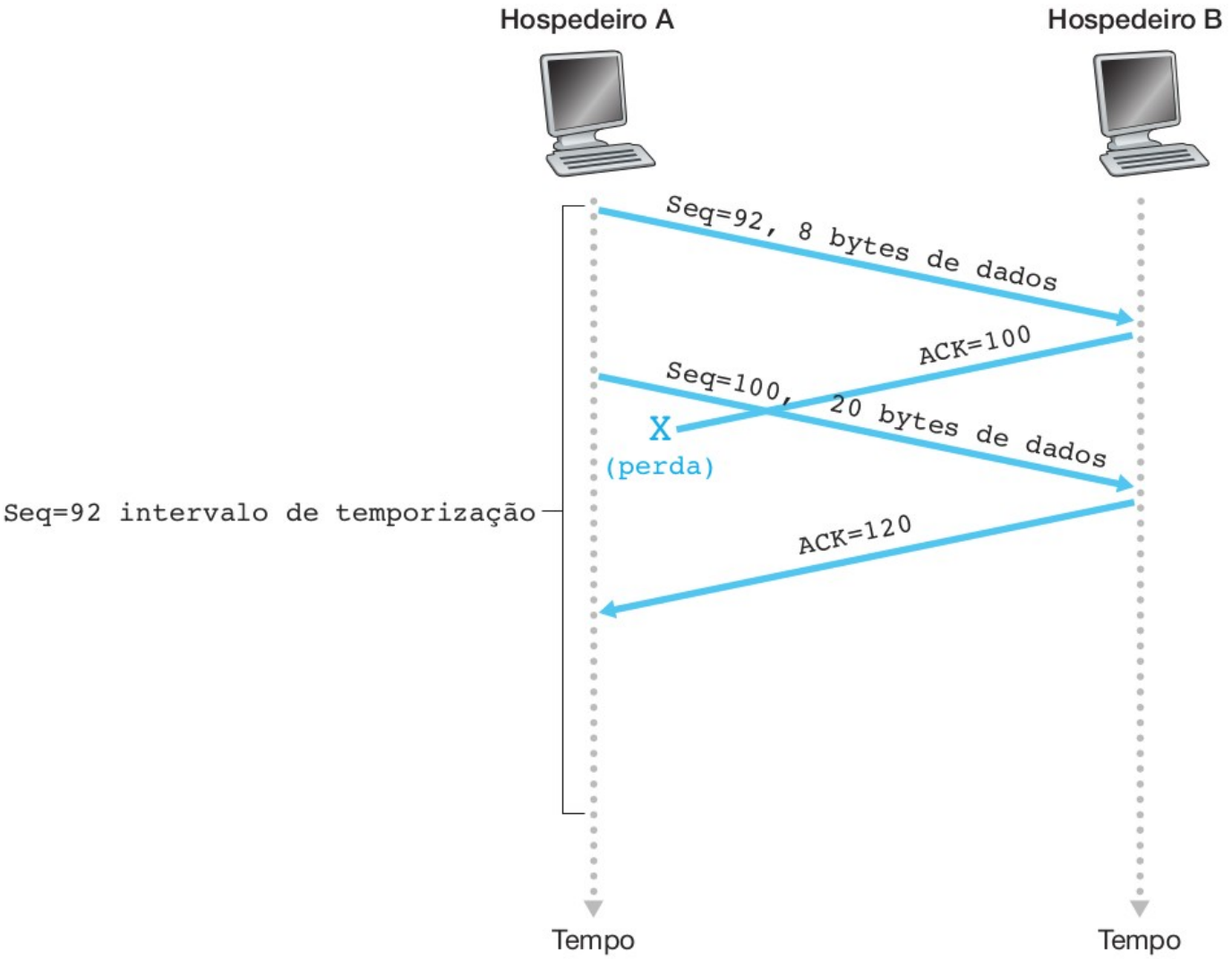
Retransmissão de reconhecimento perdido



Atraso no reconhecimento

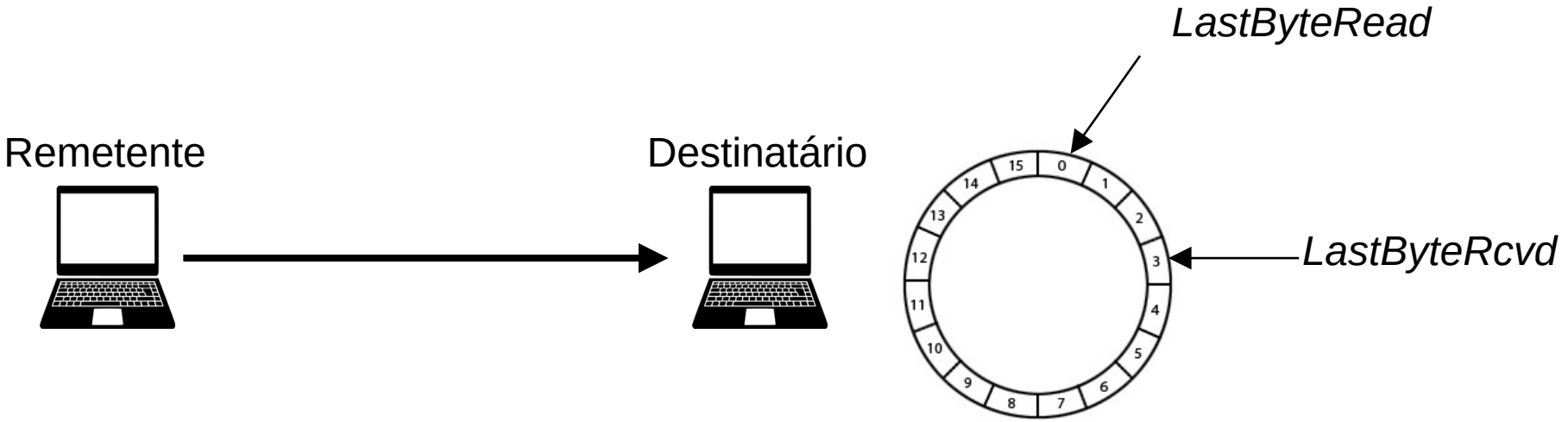


Reconhecimento cumulativo

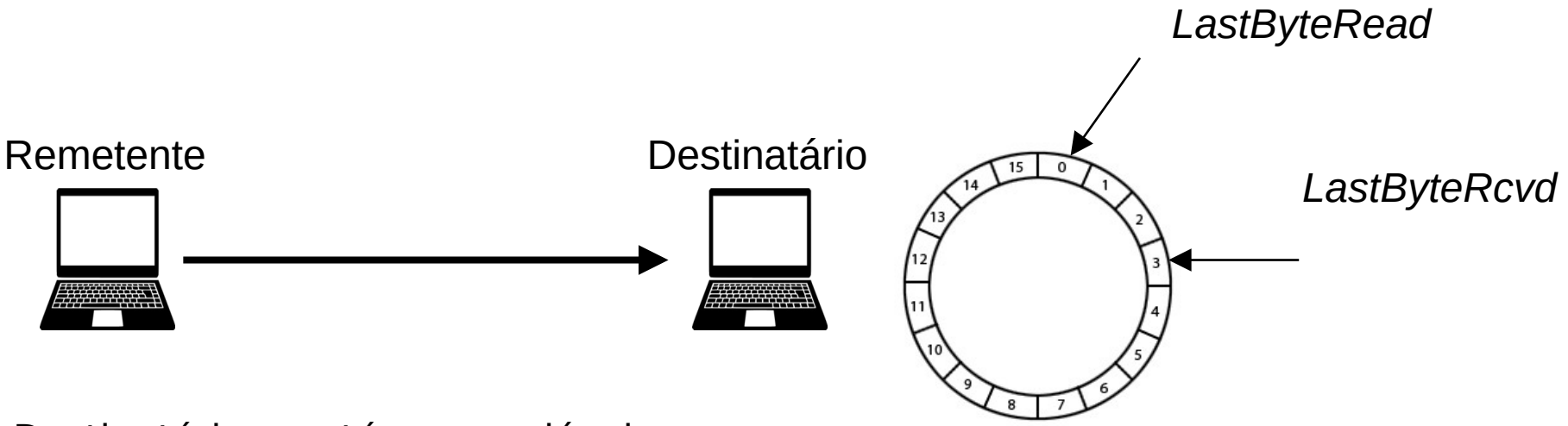


Controle de fluxo

- Evita estouro de ***buffer* do destinatário** pelo grande fluxo de dados que chega em uma conexão TCP;
  - Compatibiliza velocidade de envio do remetente com a capacidade de leitura do destinatário (velocidade de consumo);
- Utiliza o campo **janela de recepção** do cabeçalho TCP;
  - Dar ao remetente a ideia da quantidade de *buffer* livre no destinatário;

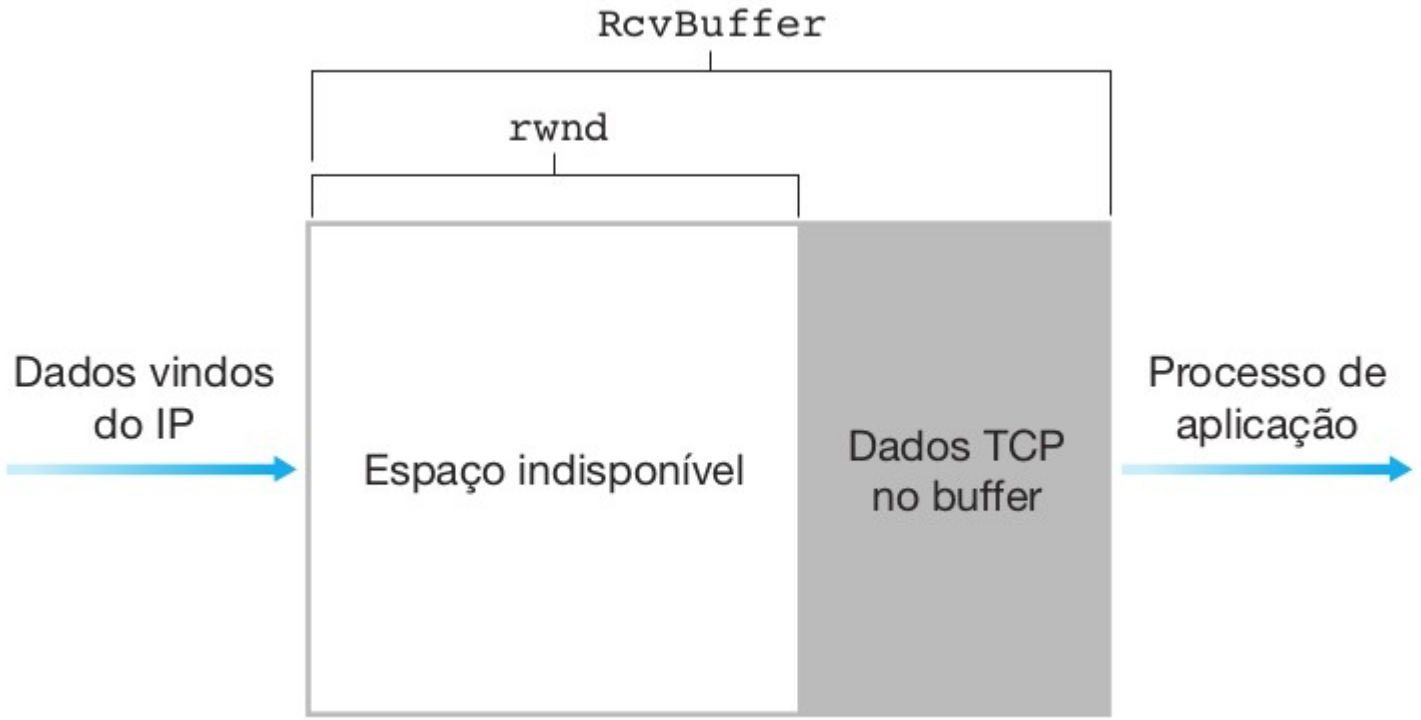


Controle de fluxo



- Destinatário mantém as variáveis:
  - **LastByteRead**: Último byte lido pela aplicação;
  - **LastByteRcvd**: Último byte recebido pela aplicação;
  - **RcvBuffer**: Tamanho do buffer de recepção;
  - *RcvBuffer >= LastByteRcvd – LastByteRead*
    - Não sobrecarregar o buffer
- O **tamanho da janela** (RcvWindow) é definido por:
  - RcvWindow = *RcvBuffer - (LastByteRcvd – LastByteRead)*
    - Próximo de 0, pouco espaço no buffer do destinatário.

Controle de fluxo

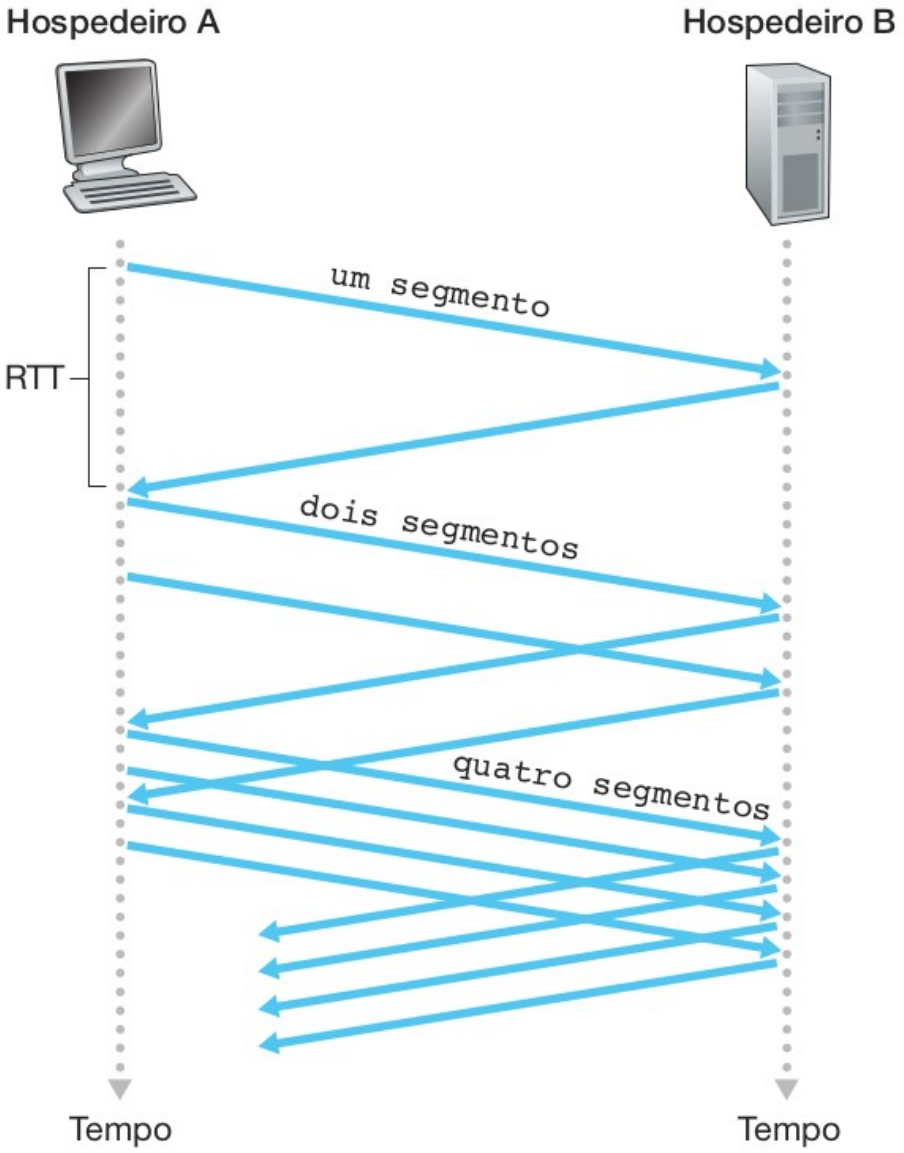


Controle de congestionamento

- Evita o congestionamento da rede percebido, limitando a taxa à qual os sistemas finais encaminham os segmentos;
  - Modificação do valor de janela de congestionamento.
- Como **identificar** o congestionamento?
  - **Perda de segmentos**: Esgotamento do temporizador (timeout);
  - **Atrasos**: Recebimento de 3 ACKs duplicados.
- Como **determinar** a taxa de envido do TCP?
  - Algoritmo de controle de congestionamento TCP
    - **Partida lenta;**
    - **Prevenção de congestionamento;**
    - **Recuperação rápida;**

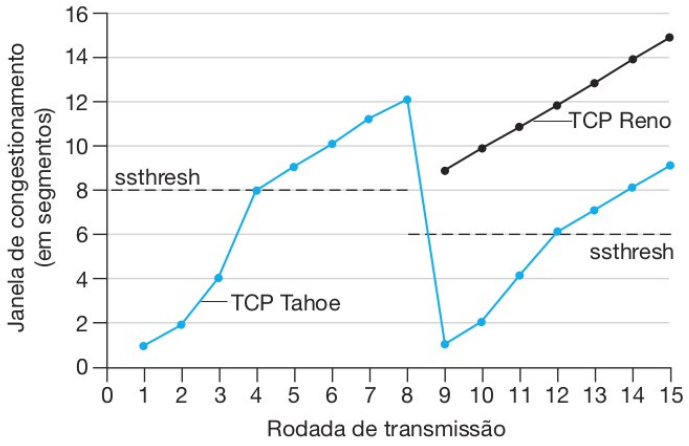
Partida lenta

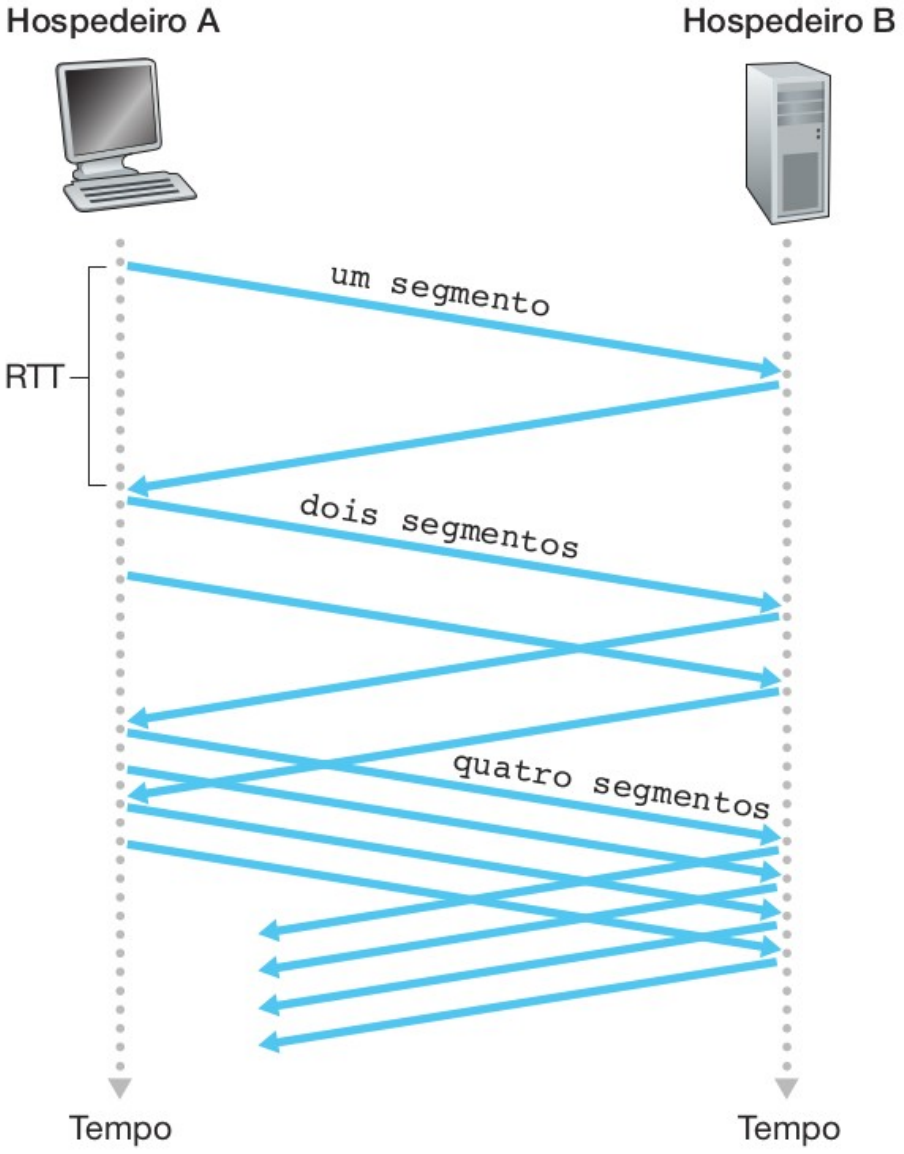
- O valor da janela(***cwnd***) começa em 1 MSS;
- Aumenta em 1 MSS sempre que recebe um reconhecimento;
- A taxa de envio **cresce exponencialmente**;



Partida lenta

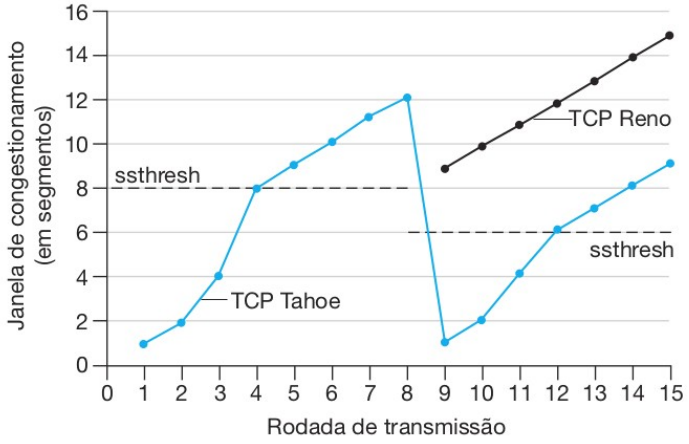
- Perda de segmento:
  - define um limiar (*ssthresh*)com a metade do *cwnd*
  - *cwnd* == 1MSS;
- Quando o *cwnd* == *ssthresh*
  - Finaliza o modo de partida lenta
  - Inicia o modo de **prevenção de congestionamento**;

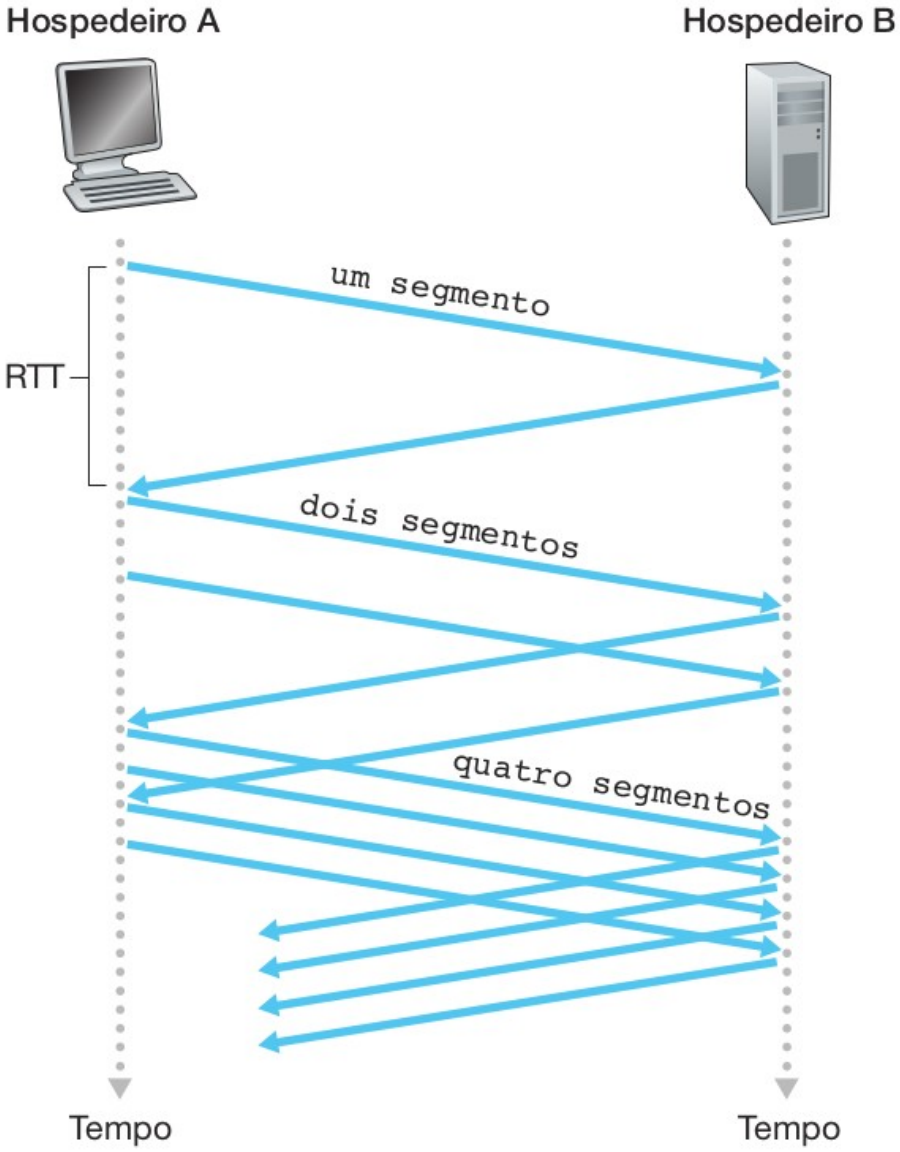




Prevenção de congestionamento

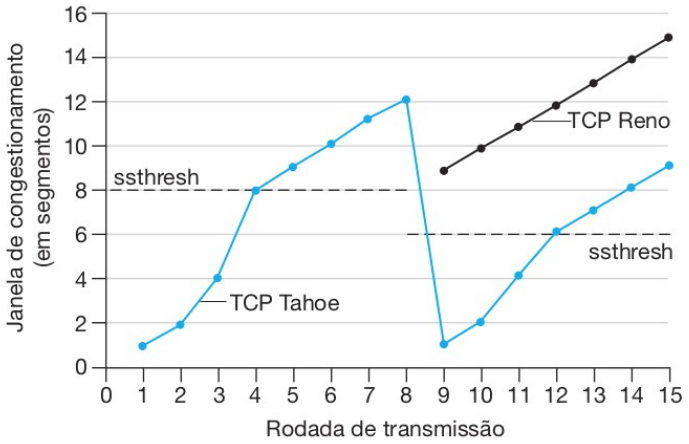
- **Aumento linear** do *cwnd*;
- ***Timeout***
  - *ssthresh == cwnd / 2*
  - *cwnd* == 1MSS;
  - Volta ao modo de **partida lenta**.
- **3 ACK duplicados**
  - *cwnd == cwnd / 2*
  - Inicia o modo de **recuperação rápida**;



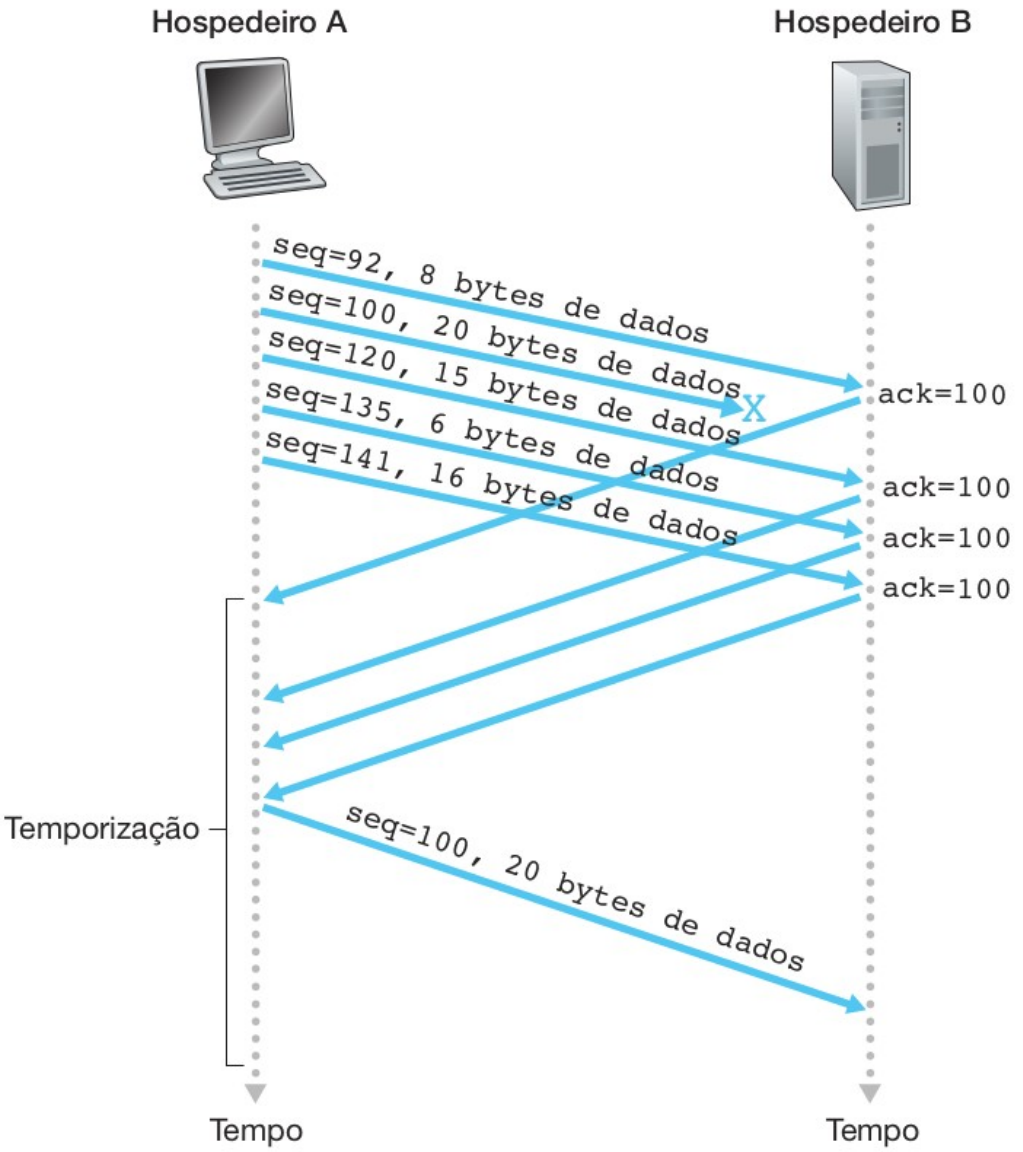


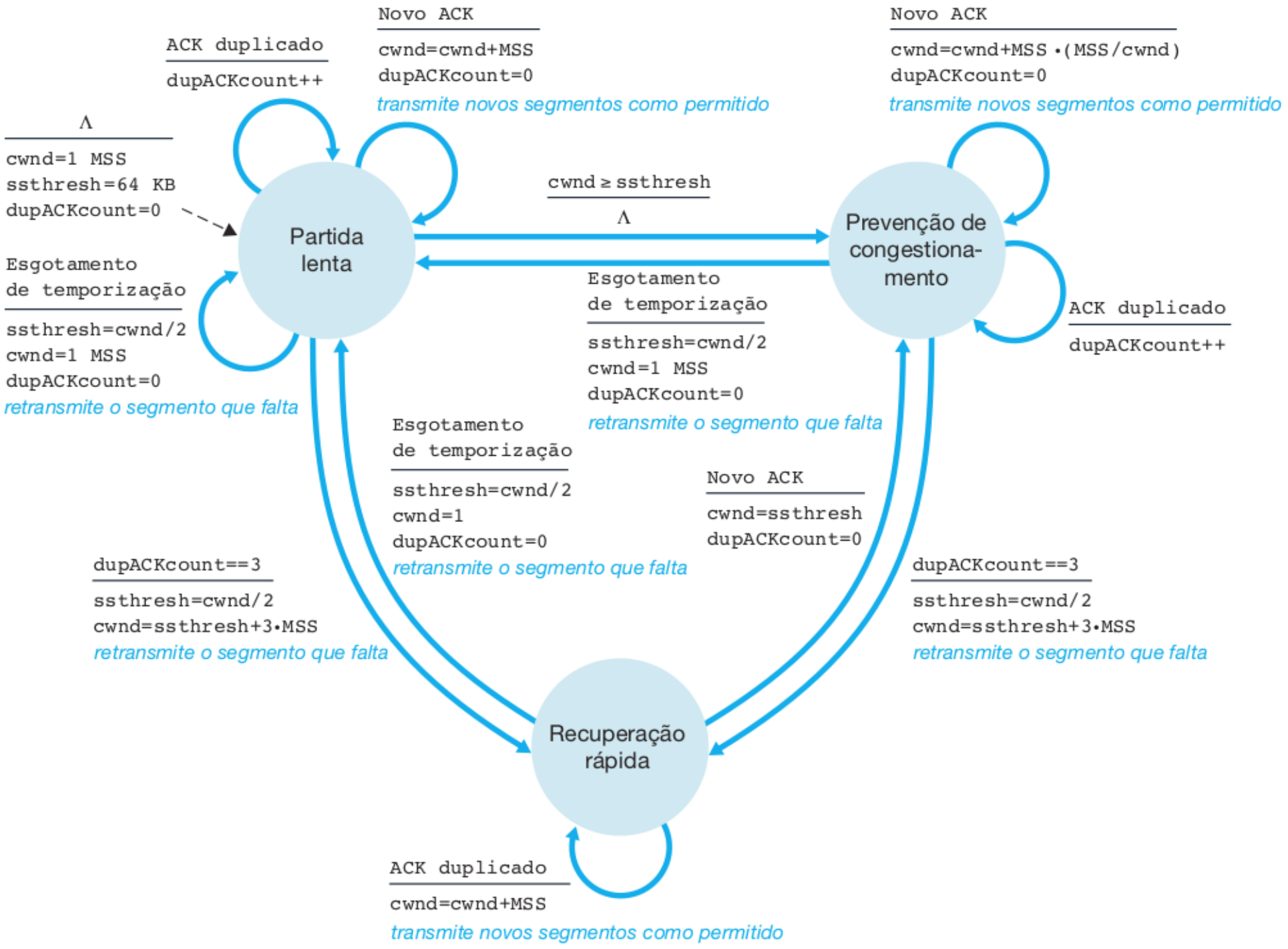
Recuperação rápida

- 3 ACKs duplicados determina uma perda (e não atraso nos pacotes).
- **Ação**:
  - Não aguarda o timeout e já reenvia o segmento perdido;
  - *cwnd == cwnd / 2;*
  - Aumento linear do *cwnd* (prevenção de congestionamento);
- ***Timeout***
  - *ssthresh == cwnd / 2;*
  - *cwnd* == 1MSS;
  - Volta ao modo de **partida lenta**.
- **Sem ACK duplicado**
  - *cwnd == ssthresh;*
  - Inicia o modo de **prevenção de congestionamento**;

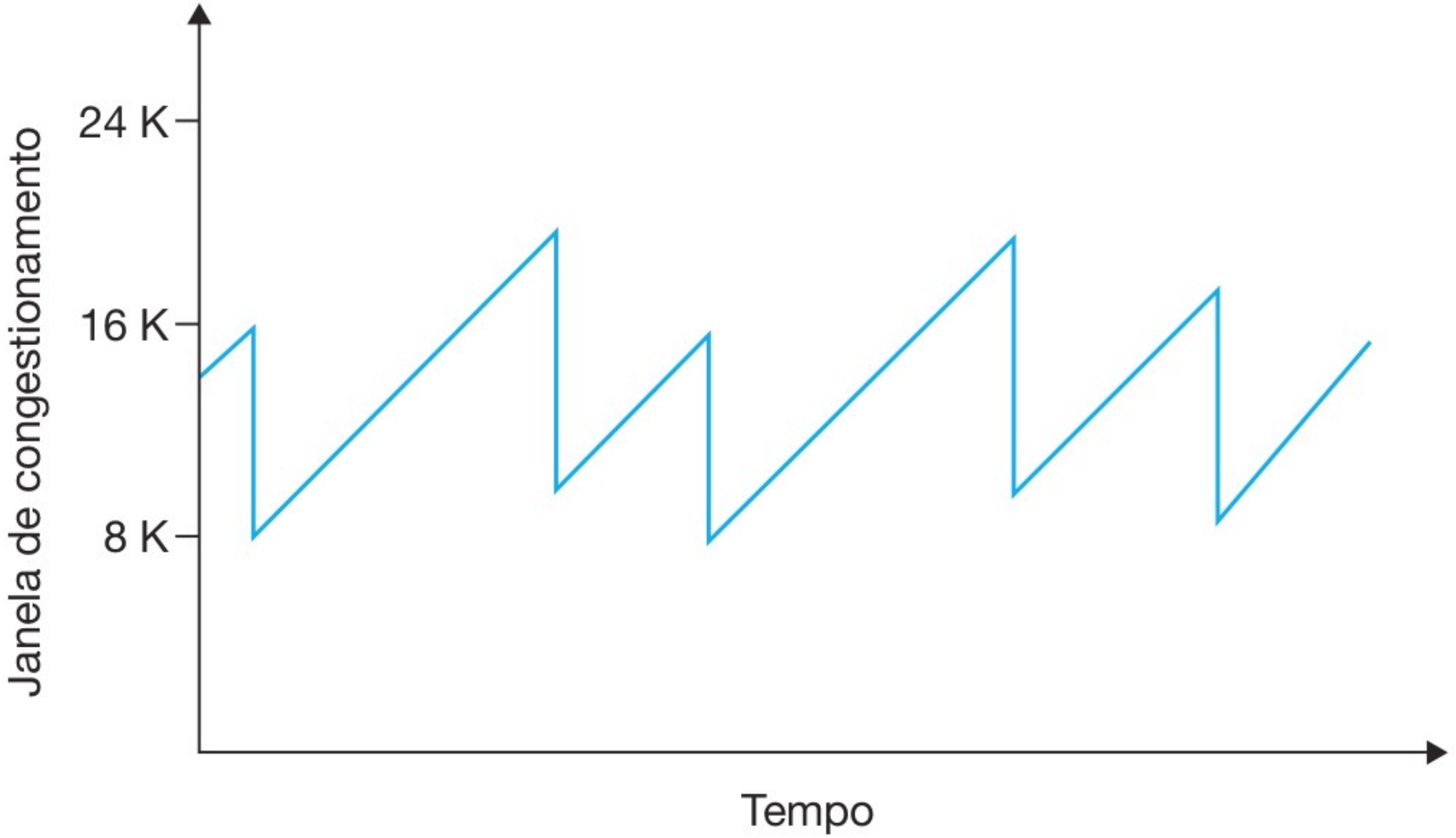


Recuperação rápida




Partida lenta
Prevenção de congestiona-mento
Recuperação rápida
Λ
cwnd=1 MSS
ssthresh=64 KB
dupACKcount=0
ACK duplicado
dupACKcount++
Novo ACK
cwnd=cwnd+MSS
dupACKcount=0
transmite novos segmentos como permitido
cwnd ≥ ssthresh
Λ
Novo ACK
cwnd=cwnd+MSS •(MSS/cwnd)
dupACKcount=0
transmite novos segmentos como permitido
ACK duplicado
dupACKcount++
Esgotamento de temporização
ssthresh=cwnd/2
cwnd=1 MSS
dupACKcount=0
retransmite o segmento que falta
Esgotamento de temporização
ssthresh=cwnd/2
cwnd=1 MSS
dupACKcount=0
retransmite o segmento que falta
Esgotamento de temporização
ssthresh=cwnd/2
cwnd=1
dupACKcount=0
retransmite o segmento que falta
Novo ACK
cwnd=ssthresh
dupACKcount=0
dupACKcount==3
ssthresh=cwnd/2
cwnd=ssthresh+3•MSS
retransmite o segmento que falta
dupACKcount==3
ssthresh=cwnd/2
cwnd=ssthresh+3•MSS
retransmite o segmento que falta
ACK duplicado
cwnd=cwnd+MSS
transmite novos segmentos como permitido

Visão macro da vazão do TCP



Variantes do TCP

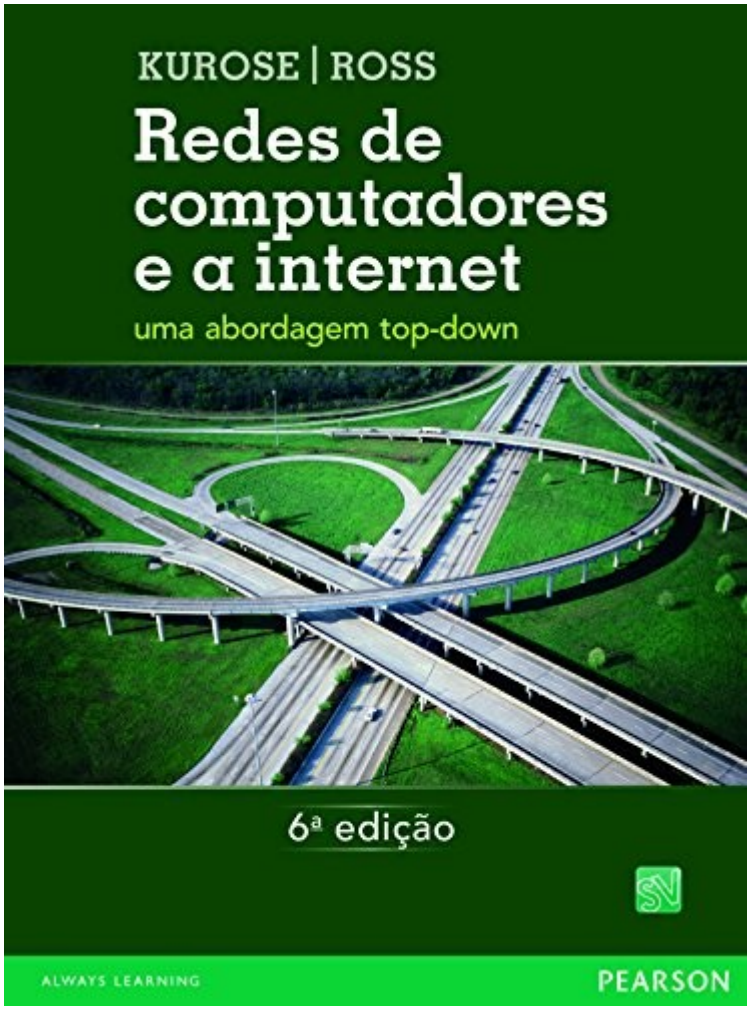
- **Tahoe**
  - Base para outras variantes;
  - Qualquer tipo de perda volta à partida lenta.
- **Reno**
  - Usa partida lenta, prevenção de congestionamento e recuperação rápida.
- **Vegas**
  - Detecta congestionamento através da variação de atraso em vez de perdas de pacotes.
- **New Reno**
  - Evolução do RENO;
  - Introduz o recurso de Reconhecimento seletivo para recuperação mais precisa.

Variantes do TCP

- **Westwood**
  - Melhora desempenho em redes sem fio, onde há mais perdas de pacotes.
- **BBR (Bottleneck Bandwidth and RTT)**
  - Desenvolvido pelo Google;
  - Baseado em estimativas de largura de banda e RTT.

Referências

- Capítulo 3 do Livro do Kurose.



Dúvidas?