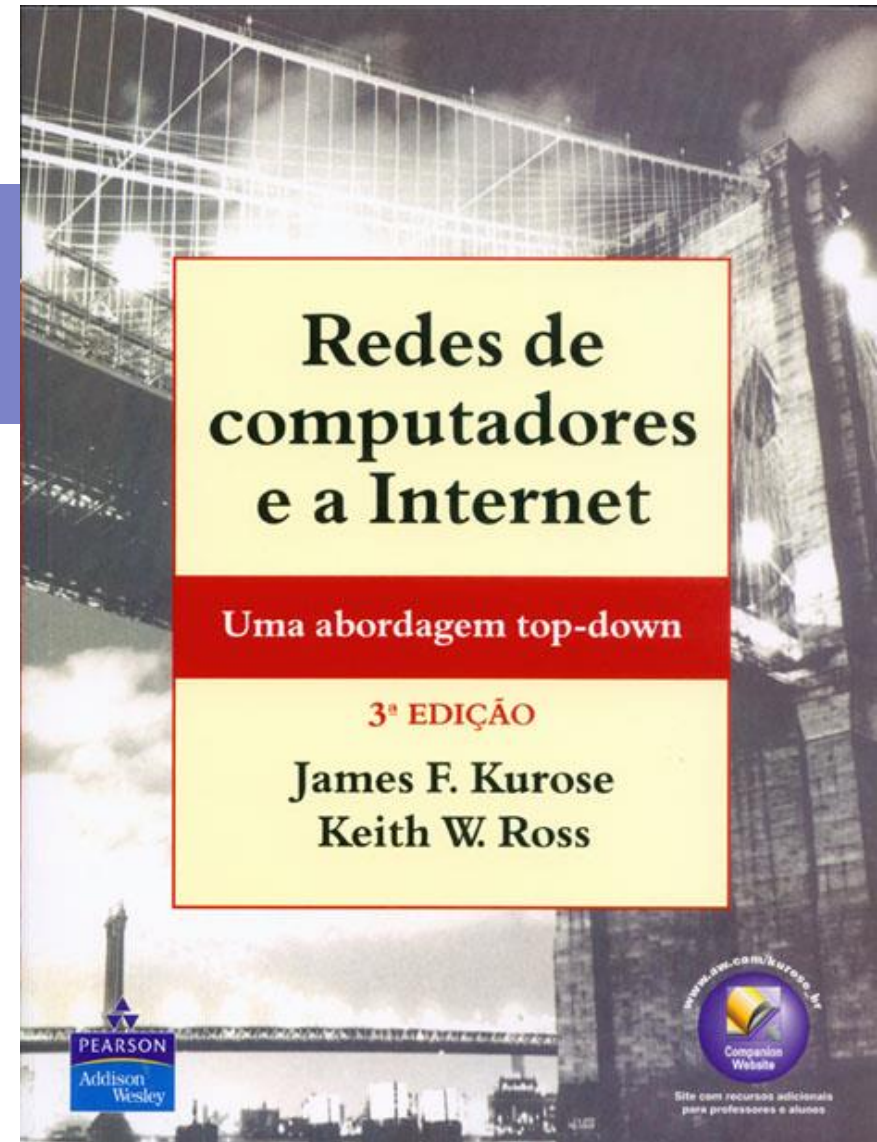


Redes de computadores e a Internet

Capítulo 3

Camada de transporte

(Alterado por Atslands Rocha
e Arthur Callado)

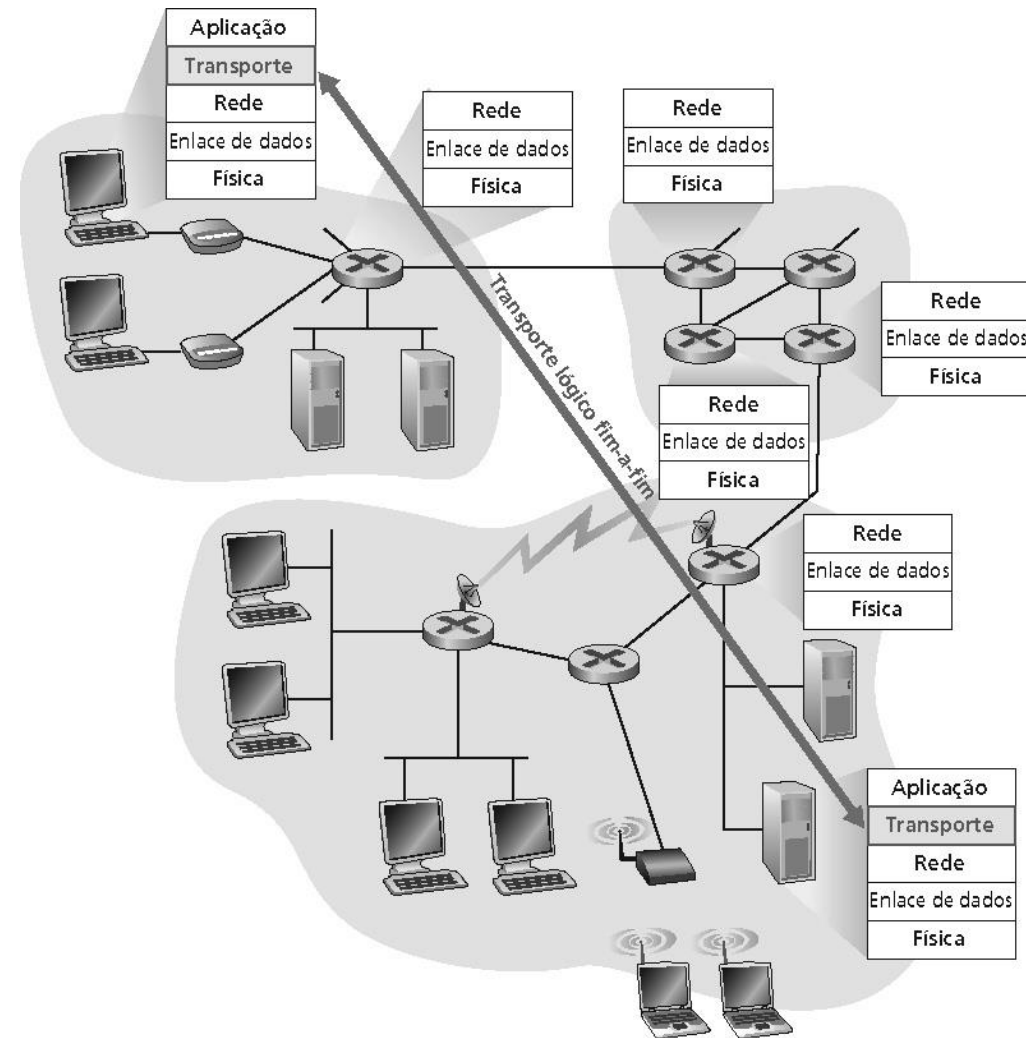


3 Camada de transporte

- 3.1 Serviços da camada de transporte
- 3.2 Multiplexação e demultiplexação
- 3.3 Transporte não orientado à conexão: UDP
- 3.4 Transporte orientado à conexão: TCP
 - Estrutura do segmento
 - Transferência confiável de dados
 - Controle de fluxo
 - Princípios de Controle de Congestionamento

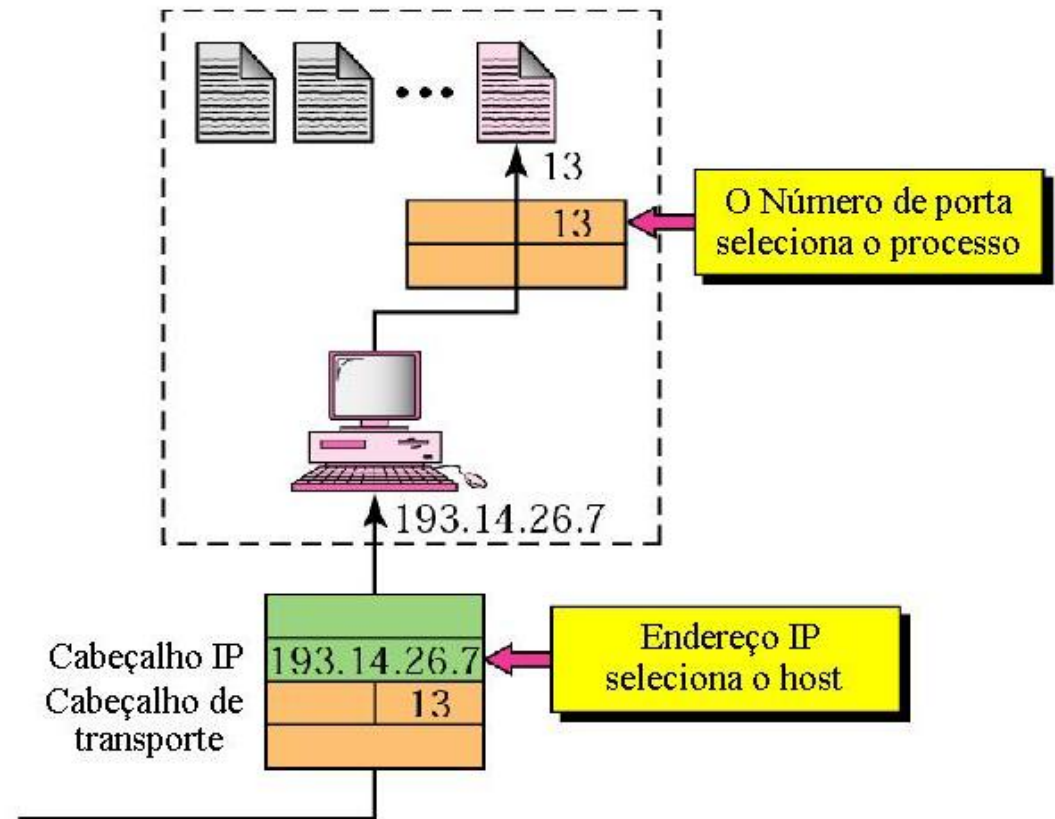
3 Protocolos e serviços de transporte

- São executados nos sistemas finais
- Lado emissor: Msgs da aplicação
-> segmentos da camada de rede
- Lado receptor: segmentos -> msgs para a camada de aplicação
- Internet: TCP e UDP



3 Protocolos e serviços de transporte

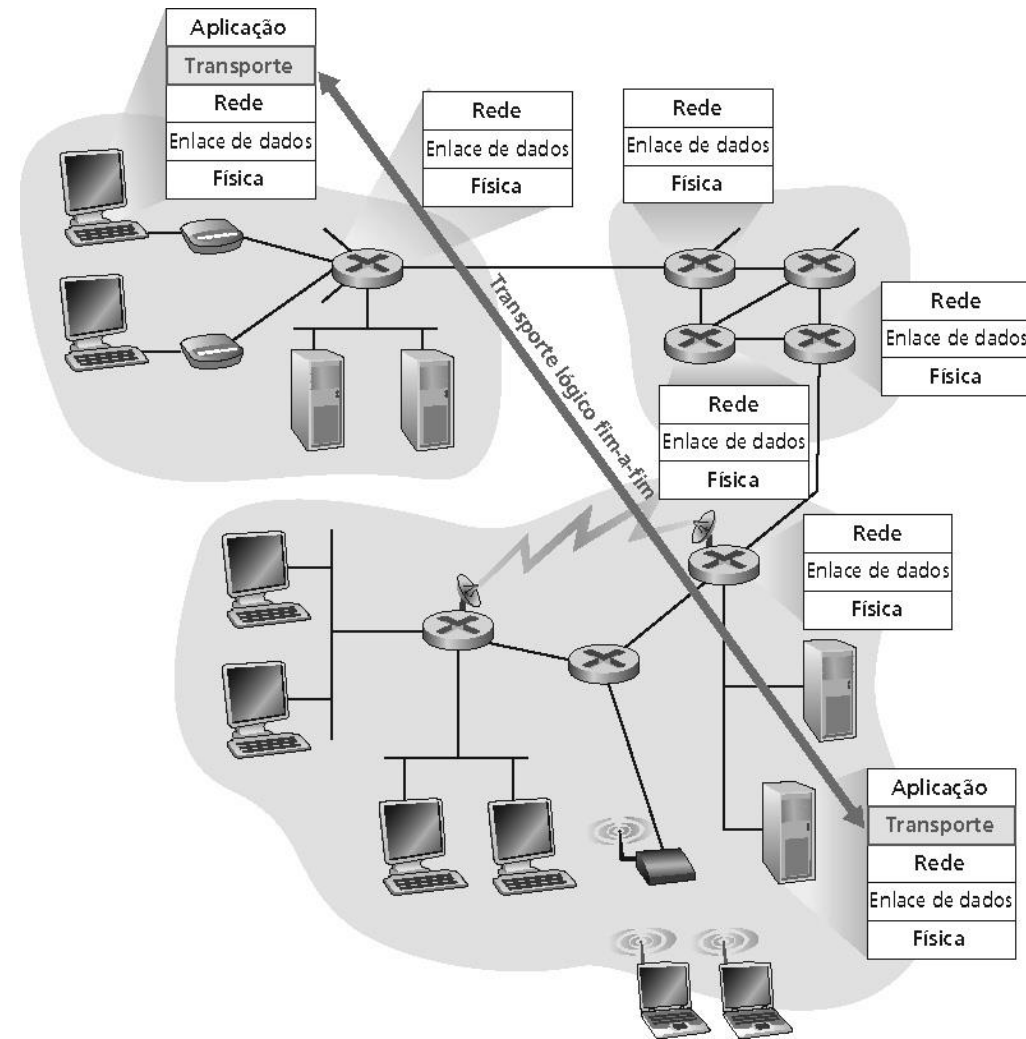
- **Camada de rede:** comunicação lógica entre os hosts.
- **Camada de transporte:** comunicação lógica entre os processos.



3

Protocolos da camada de transporte da Internet

- Serviços não disponíveis:
 - Garantia a atrasos
 - Garantia de banda
- TCP:
 - Confiável, detecção de erros, garante ordem de entrega;
 - Controle de congestionamento
 - Controle de fluxo
 - Orientado à conexão
- UDP:
 - Não confiável, detecção de erros, sem ordem de entrega;
 - Extensão do “melhor esforço” do IP



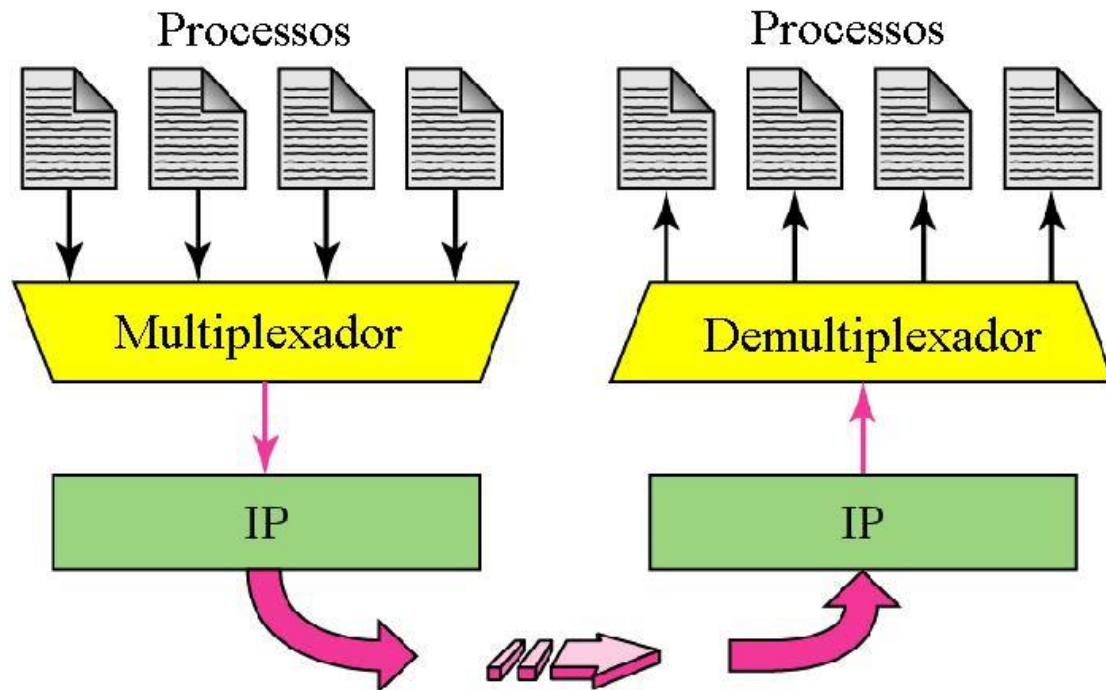
3 Multiplexação/demultiplexação

Demultiplexação no host receptor:

entrega os segmentos recebidos ao socket correto

Multiplexação no host emissor:

coleta dados de múltiplos sockets, envelope os dados com cabeçalho



- Números de portas: 16 bits (0-65535)
- Portas Conhecidas: 0-1023 (www.iana.org)

3 Demultiplexação não orientada à conexão

- Socket UDP identificado por 2 valores:
(IP de destino, porta destino)
- No receptor (segmento UDP):
 - Verifica a porta de destino no segmento
 - Direciona o segmento UDP para o socket com este número de porta
- **Datagramas com IP de origem diferentes e/ou portas de origem diferentes são direcionados para o mesmo socket!**

3 Demultiplexação orientada à conexão

- Socket TCP identificado por 4 valores:
 - IP de origem, porta de origem, IP de destino, porta de destino
- Receptor usa os valores para direcionar o segmento ao socket apropriado
- Datagramas com IP de origem diferentes e/ou portas de origem diferentes são direcionados para sockets diferentes (exceto pacote com a requisição de estabelecimento da conexão).
- Servidor pode suportar vários sockets TCP simultâneos:
 - Cada socket é identificado pelos seus próprios 4 valores.
 - Servidores Web possuem sockets diferentes para cada cliente conectado.
 - HTTP não persistente terá um socket diferente para cada requisição.

3 UDP: User Datagram Protocol

- Serviço “melhor esforço”, segmentos UDP podem ser:
 - Perdidos
 - Entregues fora de ordem para a aplicação
- **Sem conexão:**
 - Não há apresentação entre o UDP transmissor e o receptor
 - Cada segmento UDP é tratado de forma independente dos outros

Por que existe um UDP?

- Não há estabelecimento de conexão (que redundaria em atrasos)
- Não há estado de conexão (nem no transmissor, nem no receptor)
- Cabeçalho de segmento reduzido (8 bytes)
- Não há controle de congestionamento: UDP pode enviar segmentos **tão rápido quanto desejado** (e possível)

3 Mais sobre UDP

- Utilização:
 - Multimídia contínua (Tolerantes à perda, Sensíveis à taxa), DNS, SNMP.

Checksum também é realizado para alguns campos do cabeçalho IP!

Objetivo: detectar “erros” no segmento transmitido

Transmissor coloca o valor do checksum no campo da msg UDP.

Receptor computa o checksum do segmento recebido.

- Verifica se o checksum calculado é igual ao valor do campo checksum:
 - NÃO - erro detectado
 - SIM - não há erros.

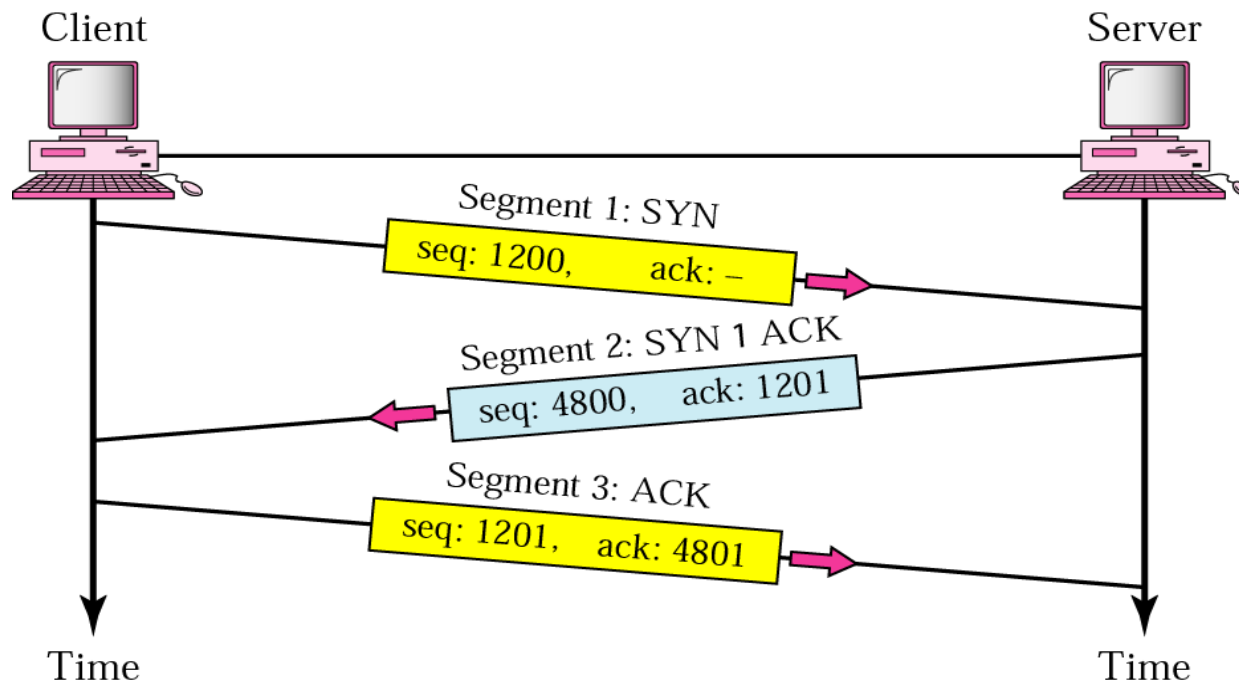
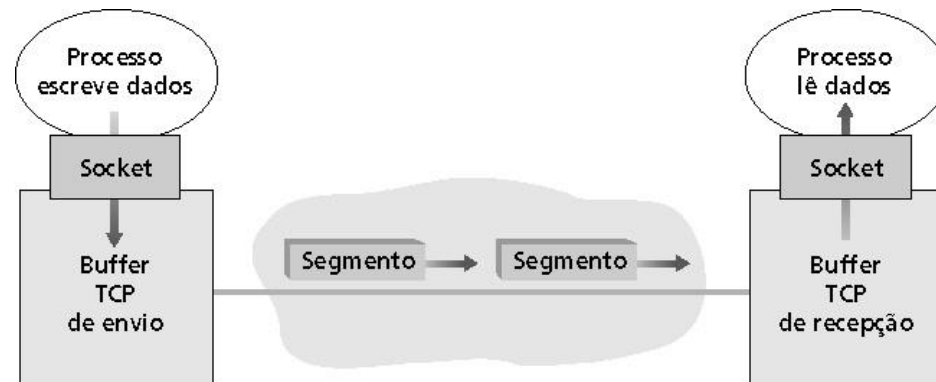
UDP não fornece correção de erros! Apenas descarta o segmento ou repassa o segmento à aplicação com um aviso!

3

TCP: overview

RFCs: 793, 1122, 1323, 2018, 2581

- **Ponto-a-ponto, confiável, fluxo sequencial de bytes.**
- **Serviço full-duplex, orientado à conexão:**
 - Apresentação de 3 vias antes da troca de dados

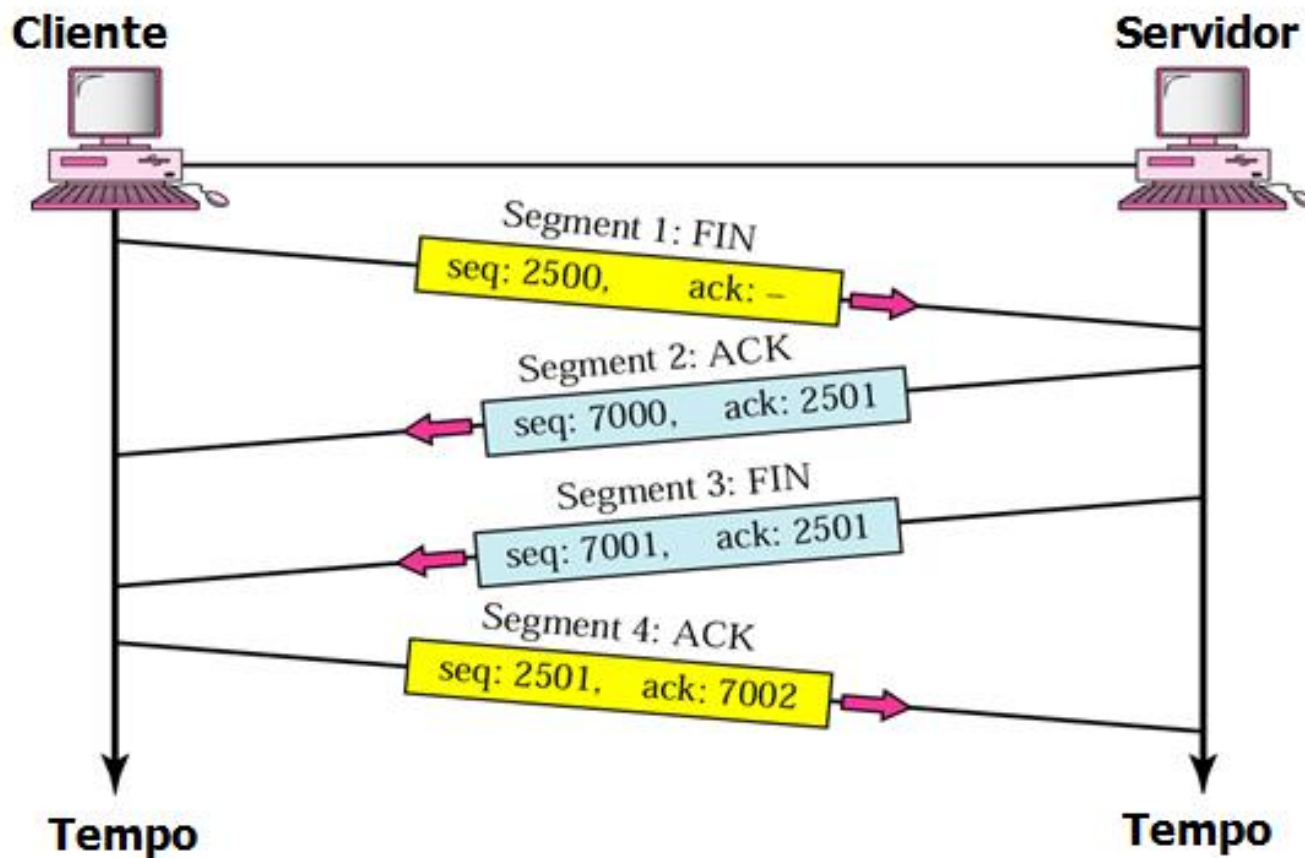


3

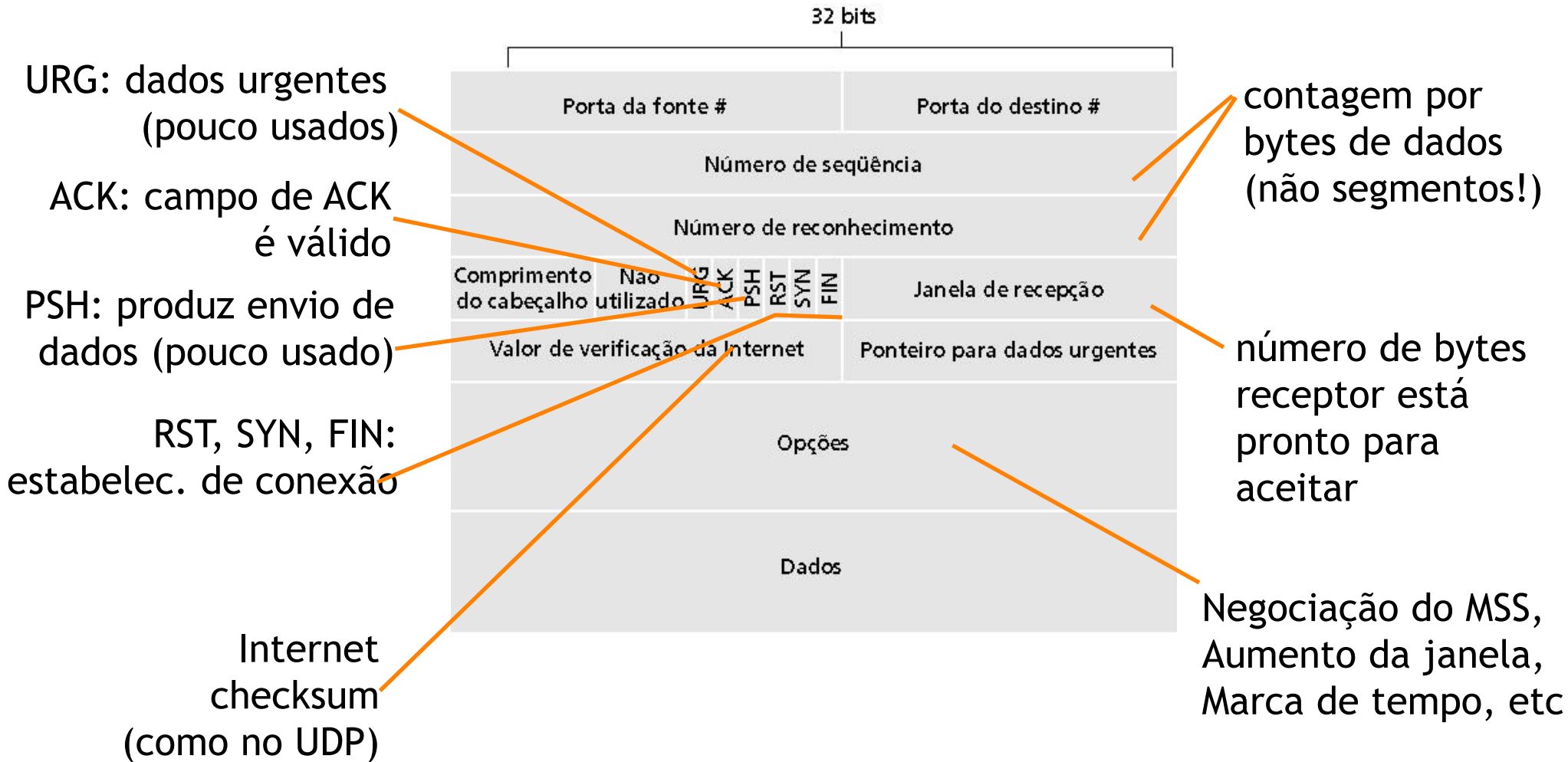
TCP: overview

RFCs: 793, 1122, 1323, 2018, 2581

• Finalização da conexão TCP.



3 Estrutura do segmento TCP

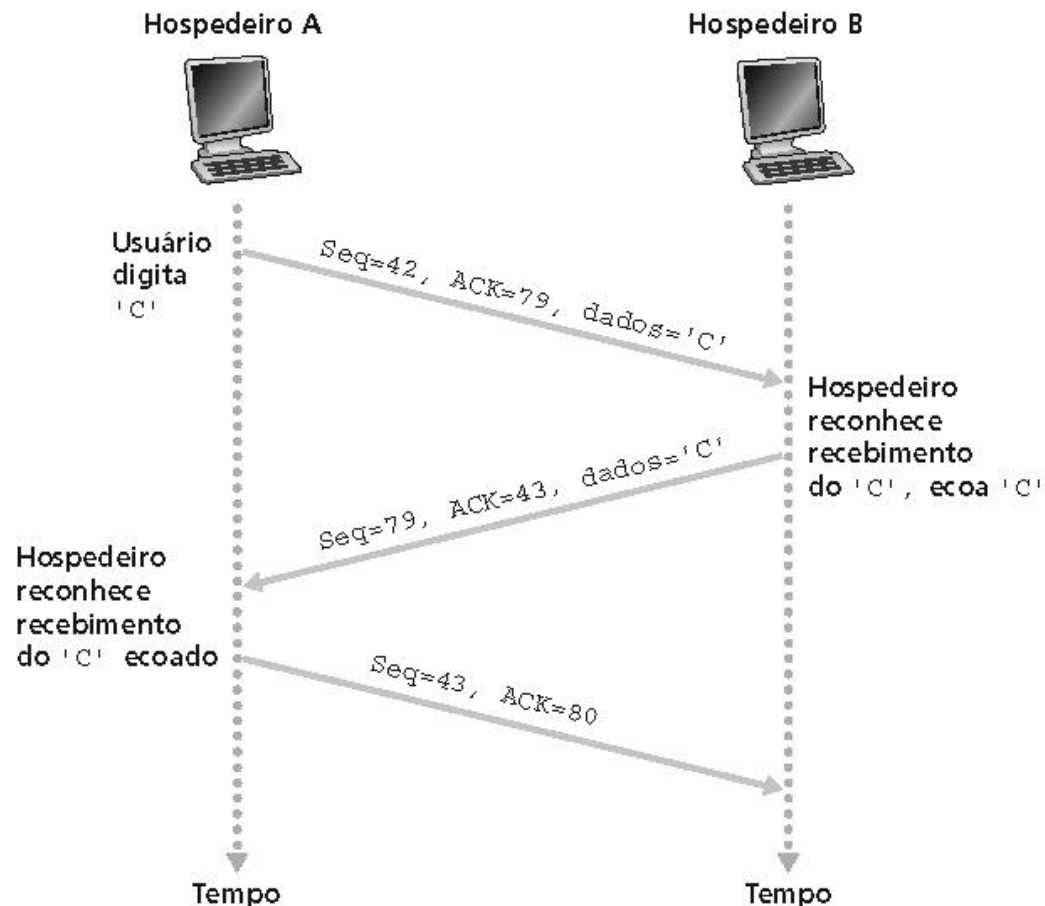


3 Número de seqüência e ACKs do TCP

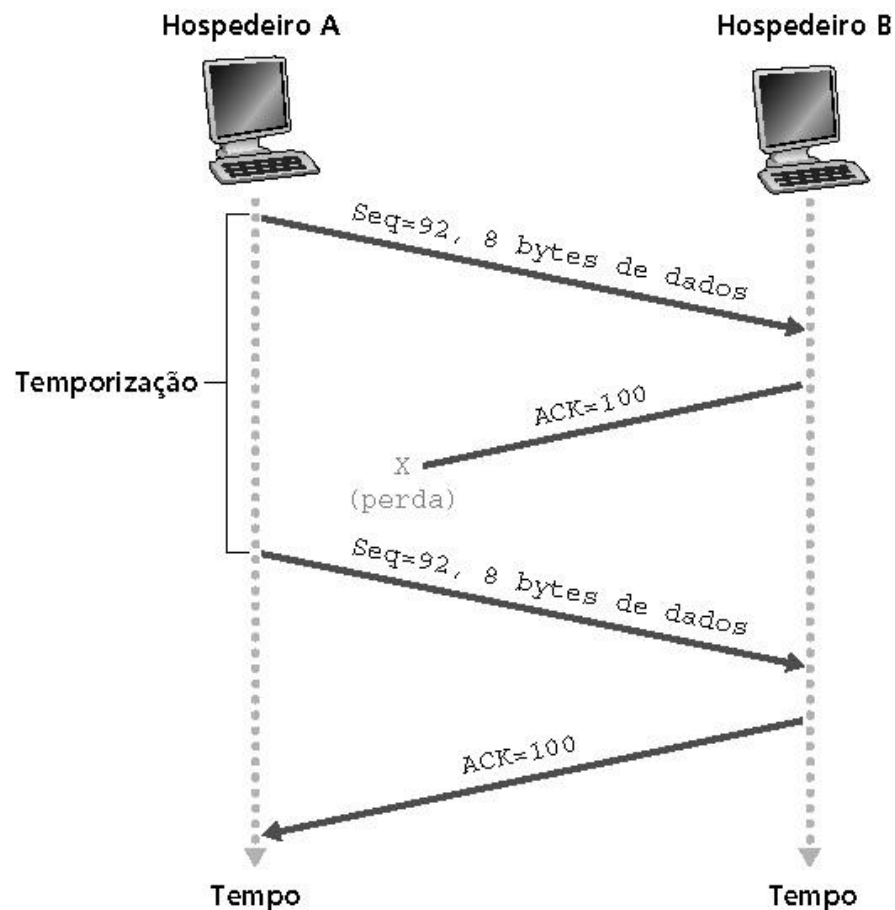
- Ex: Aplicação Telnet
- Nota: Piggyback sendo usado!

P.: Como o receptor trata segmentos fora de ordem?

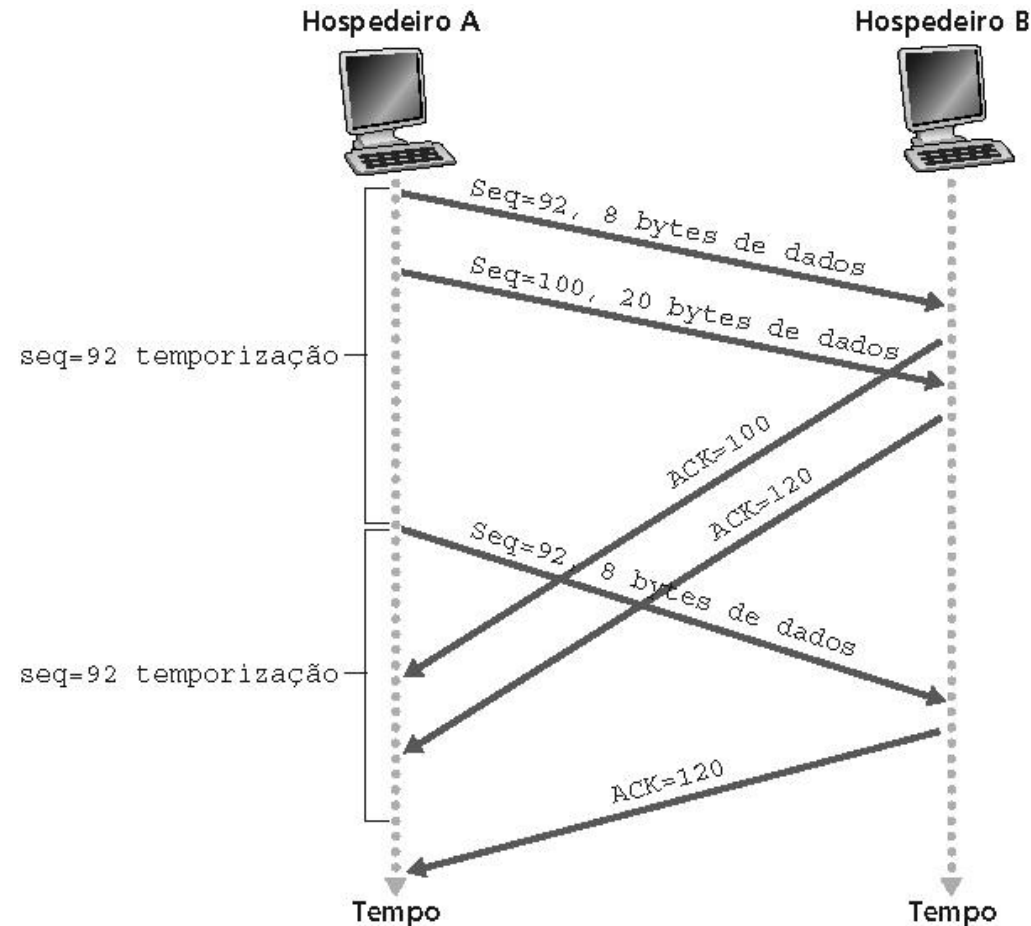
- Armazena segmentos fora de ordem e espera pelo segmento que falta (Usado na prática!)



3 TCP: cenários de retransmissão

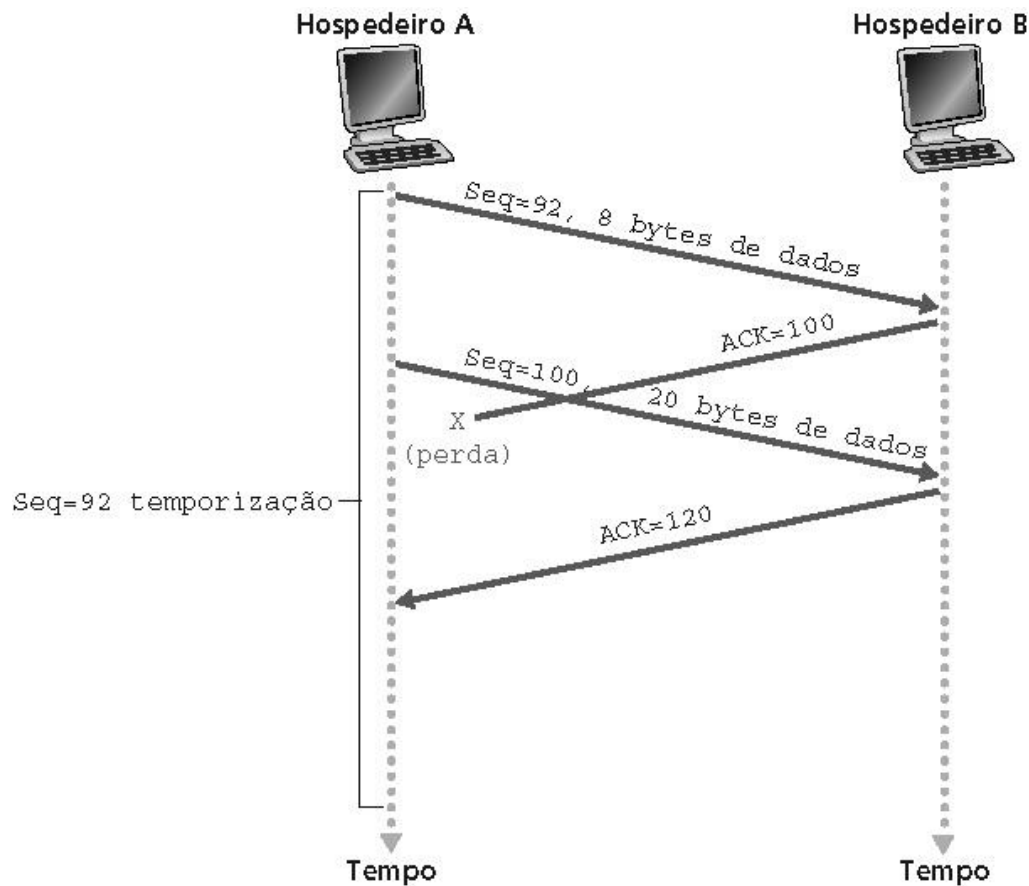


Cenário com perda do ACK



Temporização prematura, ACKs cumulativos

3 TCP: cenários de retransmissão



Cenário de ACK cumulativo

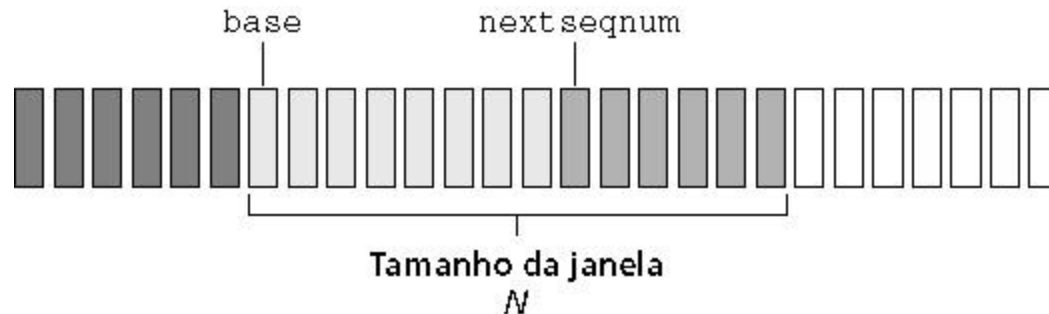
3 Retransmissão rápida

- Com frequência, o tempo de expiração é relativamente longo:
 - Longo atraso antes de reenviar um pacote perdido.
- Detecta segmentos perdidos por meio de ACKs duplicados
 - Se o segmento é perdido, haverá muitos ACKs duplicados.
- Se o transmissor recebe 3 ACKs para o mesmo dado, ele supõe que o segmento após o dado confirmado foi perdido:
 - **Retransmissão rápida:** reenvia o segmento antes de o temporizador expirar.
- **P: TCP é Go-Back-N ou repetição seletiva?**

3 Go-Back-N

Transmissor:

- “janela” de até N pacotes não reconhecidos, consecutivos, são permitidos.



Legenda:

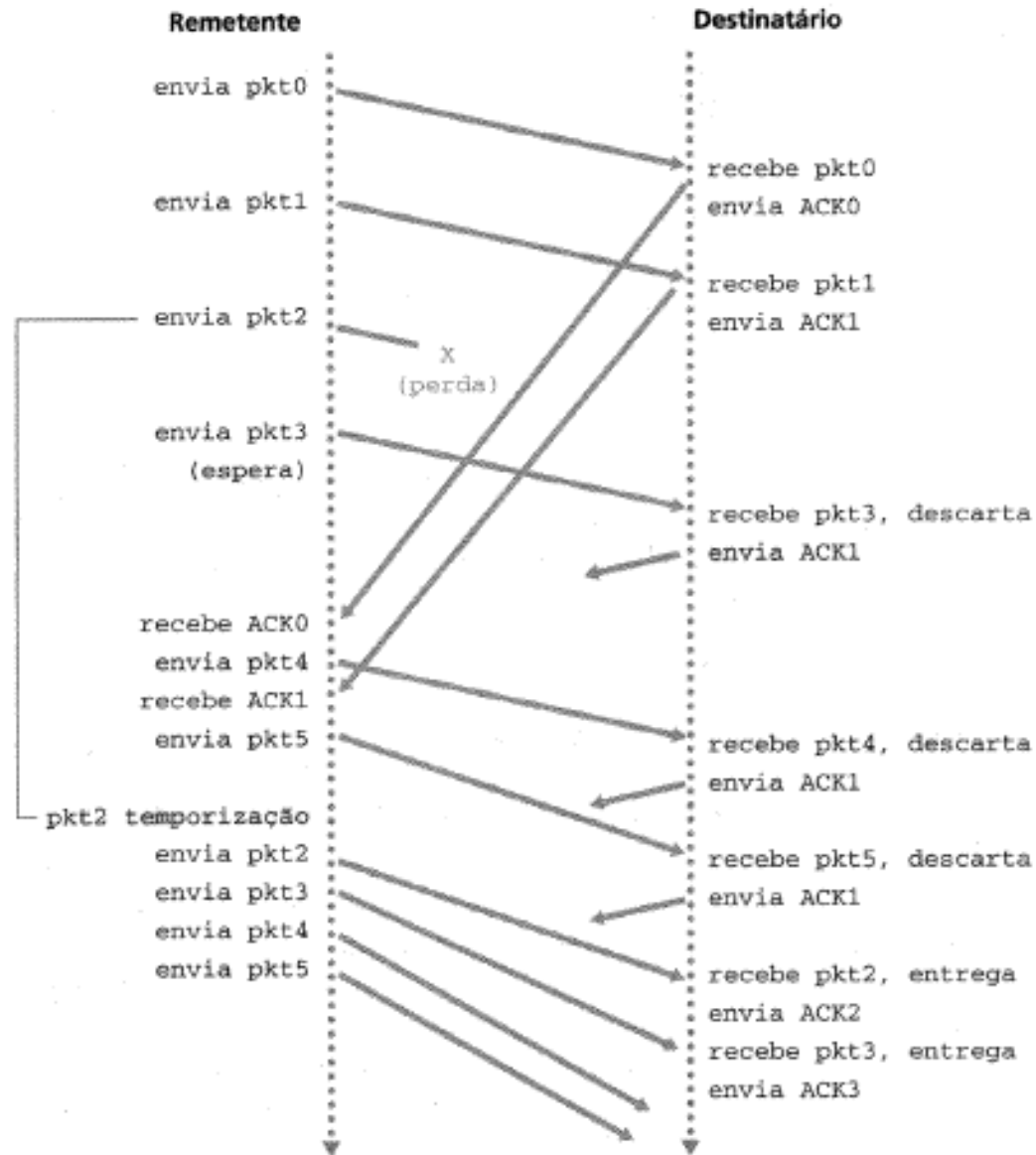
| | | | |
|--|------------------------------------|--|-----------------------------------|
| | Já reconhecido | | Autorizado, mas ainda não enviado |
| | Enviado, mas ainda não reconhecido | | Não autorizado |

- ACK cumulativo
 - Pode receber ACKs duplicados.
- Temporizador para cada pacote enviado e não confirmado.

3 Go-Back-N

- Sempre envia ACK para pacotes corretamente recebidos com o mais alto número de sequência **em ordem**.
 - Pode gerar ACKs duplicados.
- Pacotes fora de ordem:
 - Descarta (não armazena) -> **não há buffer de recepção!**
 - Reconhece pacote com o mais alto número de sequência em ordem.

3 GBN em ação

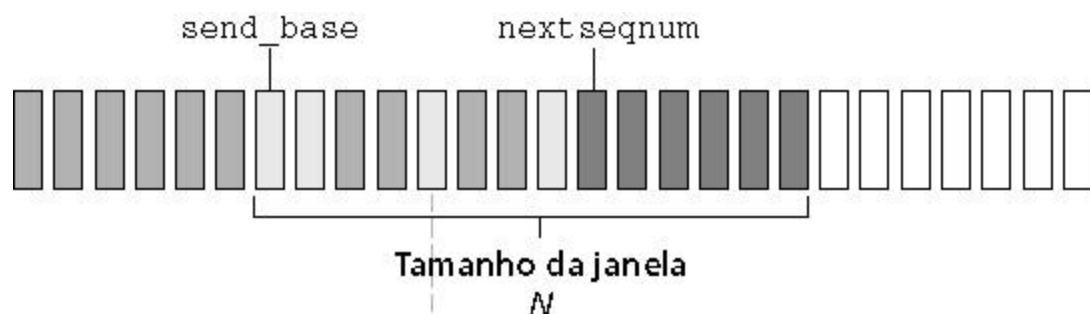


3 Retransmissão seletiva

- Armazena pacotes, quando necessário, para eventual entrega em ordem para a camada superior.
- Transmissor somente reenvia os pacotes para os quais um ACK não foi recebido.
- Janela de transmissão
 - N números de seqüência consecutivos.
 - Novamente limita a quantidade de pacotes enviados, mas não reconhecidos.

3

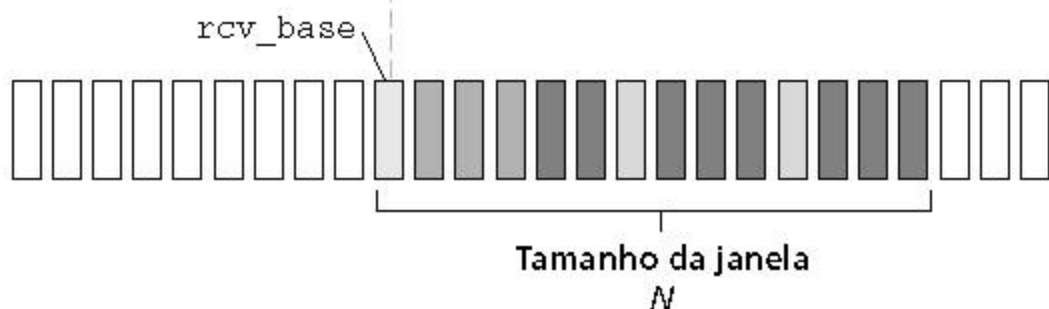
Retransmissão seletiva: janelas do transmissor e do receptor



a. Visão que o remetente tem dos números de sequência

Legenda:

| | | | |
|--|-----------------------------|--|-----------------------------------|
| | Já reconhecido | | Autorizado, mas ainda não enviado |
| | Enviado, mas não autorizado | | Não autorizado |



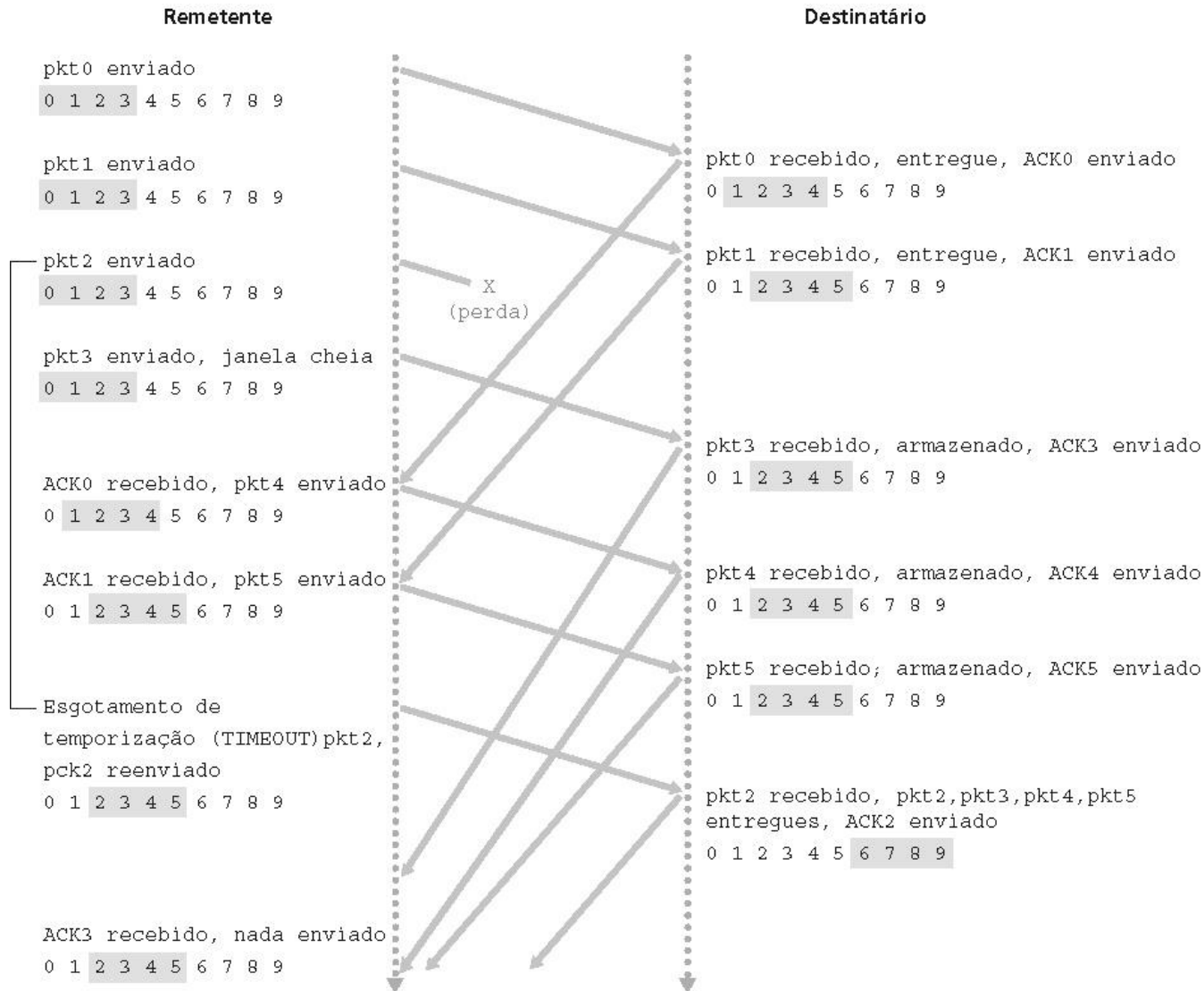
b. Visão que o destinatário tem dos números de sequência

Legenda

| | | | |
|--|---|--|------------------------------|
| | Fora de ordem (no buffer), mas já reconhecido (ACK) | | Aceitável (dentro da janela) |
| | Aguardado, mas ainda não recebido | | Não autorizado |



3 Retransmissão seletiva em ação



3 TCP é Híbrido: Go-Back-N e Seletivo

- Resumo:
- TCP é Go-Back-N porque:
 - ACKS acumulativos;
 - Segmentos corretamente recebidos em fora de ordem não são reconhecidos individualmente.
- TCP é seletivo porque:
 - Armazena segmentos recebidos corretamente e fora de ordem.
 - Retransmite somente segmento não recebido.
- Ou seja, TCP é híbrido!

3 TCP: controle de fluxo

- lado receptor da conexão TCP possui um buffer de recepção:



Controle de fluxo

Transmissor não deve esgotar os buffers de recepção enviando dados rápido demais.

- TCP encontra a taxa de envio adequada à taxa de vazão do receptor.

- Processos de aplicação podem ser lentos para ler o buffer

3 Controle de fluxo TCP: como funciona



- Receptor informa a área disponível nos segmentos.
- Transmissor limita os dados não confirmados à área livre.
 - Garantia contra “estouro” no buffer do receptor.

3 Princípios de controle de congestionamento

Congestionamento:

- Informalmente: “muitas fontes enviando dados acima da capacidade da **rede** de tratá-los”.
- Diferente de controle de fluxo!
- Sintomas:
 - Perda de pacotes (saturação de buffer nos roteadores).
 - Atrasos grandes (filas nos buffers dos roteadores).

Duas abordagens para o problema de controle de congestionamento:

Controle de congestionamento fim-a-fim:

- Congestionamento é inferido a partir das perdas e atrasos observados nos sistemas finais.
- Abordagem usada pelo TCP (IP não fornece informações sobre congestionamento!).
- TCP reduz as janelas em congestionamento!

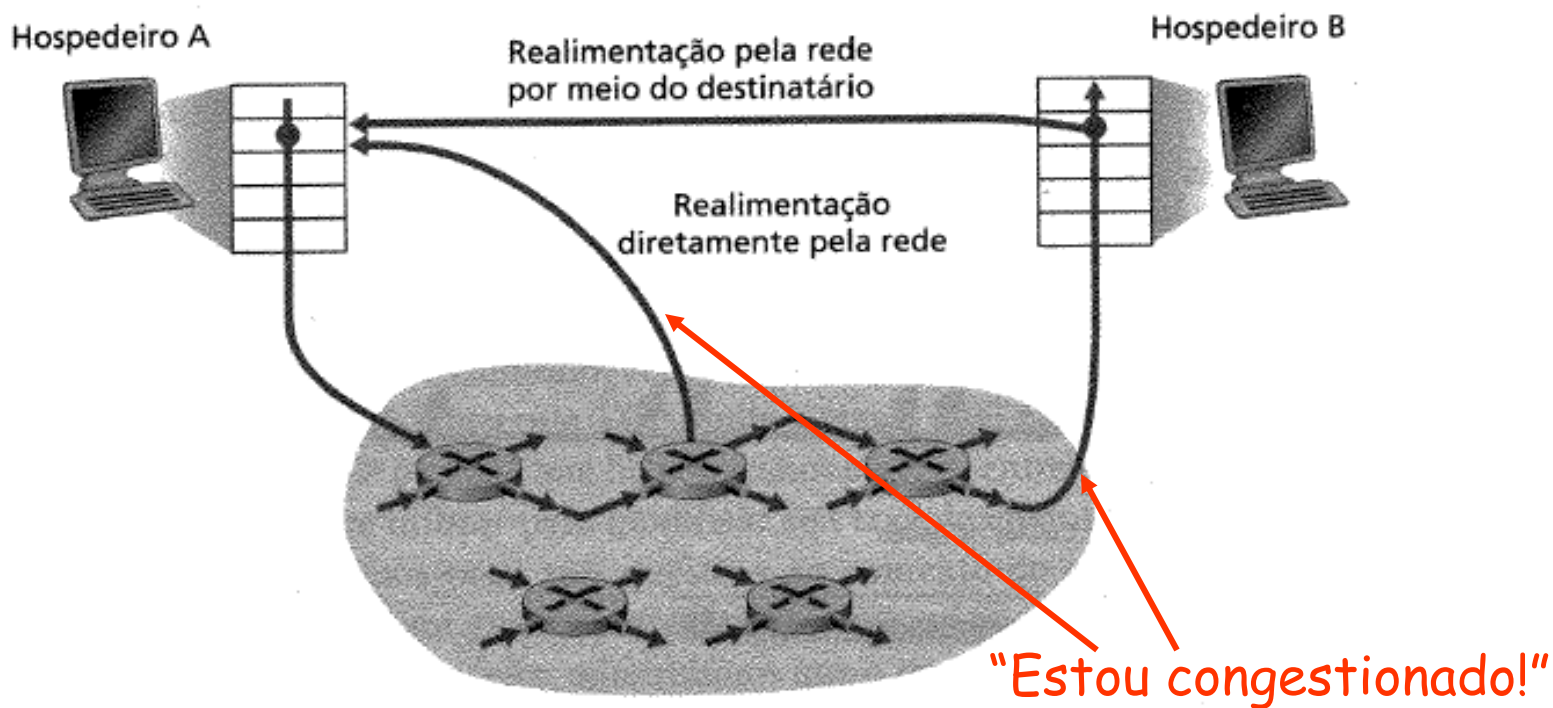
Controle de congestionamento assistido pela rede:

- Roteadores enviam informações para os sistemas finais.
 - Ex: Um Bit indicando congestionamento. (Ex: ATM)
 - Proposto recentemente para o TCP.



3

Controle de congestionamento: Realimentação da Rede



3 Janela de Congestionamento

Modos de operação:

Início Lento: A janela começa com o tamanho de 1 **MSS** (máximo de bytes de dados que cabem em 1 pacote naquela conexão) e dobra de tamanho a cada ACK recebido

Evitar Colisão: aumenta o tamanho da janela em 1 MSS a cada ACK recebido

Critérios de mudança:

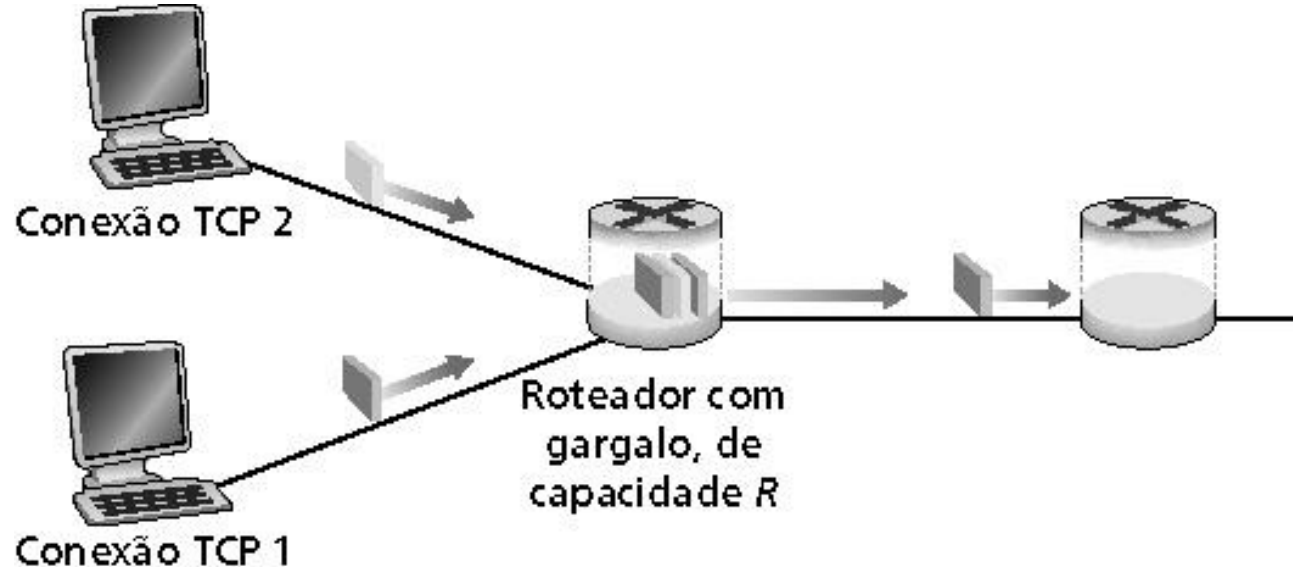
Tamanho da janela passou do limite de início lento: mudar para *evitar colisão*

1 ACK duplicado: indicativo de perda pequena, retransmitir pacote e diminuir o tamanho da janela e passar ao modo de *evitar colisão*

3 ACKs duplicados ou **estouro do temporizador:** indicativo de grave congestionamento, retransmitir pacote e passar ao modo de *início lento*

3 Equidade do TCP

Objetivo da equidade: se K sessões TCP compartilham o mesmo enlace do gargalo com largura de banda R , cada uma deve ter taxa média de R/K





Fontes: 1. Redes de Computadores e a Internet. James Kurose;
2. Comunicação de Dados e Redes de Computadores. Forouzan.