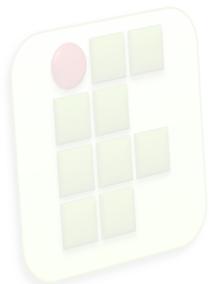


**Instituto Federal de Educação,
Ciência e Tecnologia do Rio Grande do Sul
Campus Currais Novos**

Redes de Computadores e Aplicações

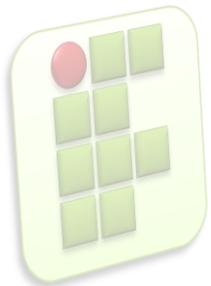


Aula 43 - Camada de Transporte
TCP (*Transmission Control Protocol*)

Prof. Diego Pereira <diego.pereira@ifrn.edu.br>

Objetivo

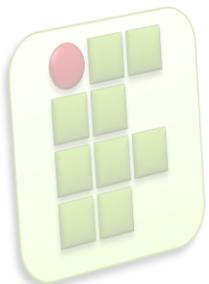
- Conhecer o funcionamento do protocolo TCP;
- Aprender as principais características do protocolo e em que situações ele é recomendado;



Protocolo TCP

■ Fundamentos

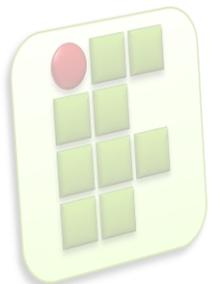
- Define a unidade de dados do serviço de circuito virtual, denominada seguimento TCP
 - Especifica o formato e a função dos campos
- Multiplexa mensagens geradas pelos processos no serviço da camada de rede
 - Encapsula segmentos em datagramas IP
- Demultiplexa segmentos para os respectivos processos destino
 - Extrai mensagens dos segmentos



Protocolo TCP

■ Fundamentos

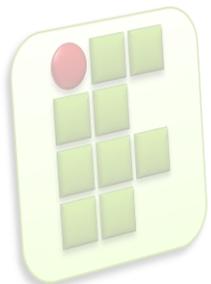
- Adota uma abordagem baseada em fluxo de dados (*data stream*)
 - Trata o fluxo de dados como uma cadeia contínua de bytes
 - Decide como agrupar bytes em segmentos
- Adota uma abordagem orientada à conexão *full-duplex*
 - Estabelecimento da conexão
 - Transferência de dados
 - Fechamento da conexão



Protocolo TCP

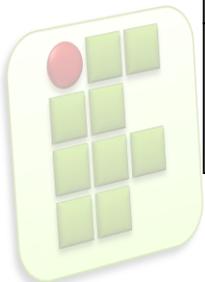
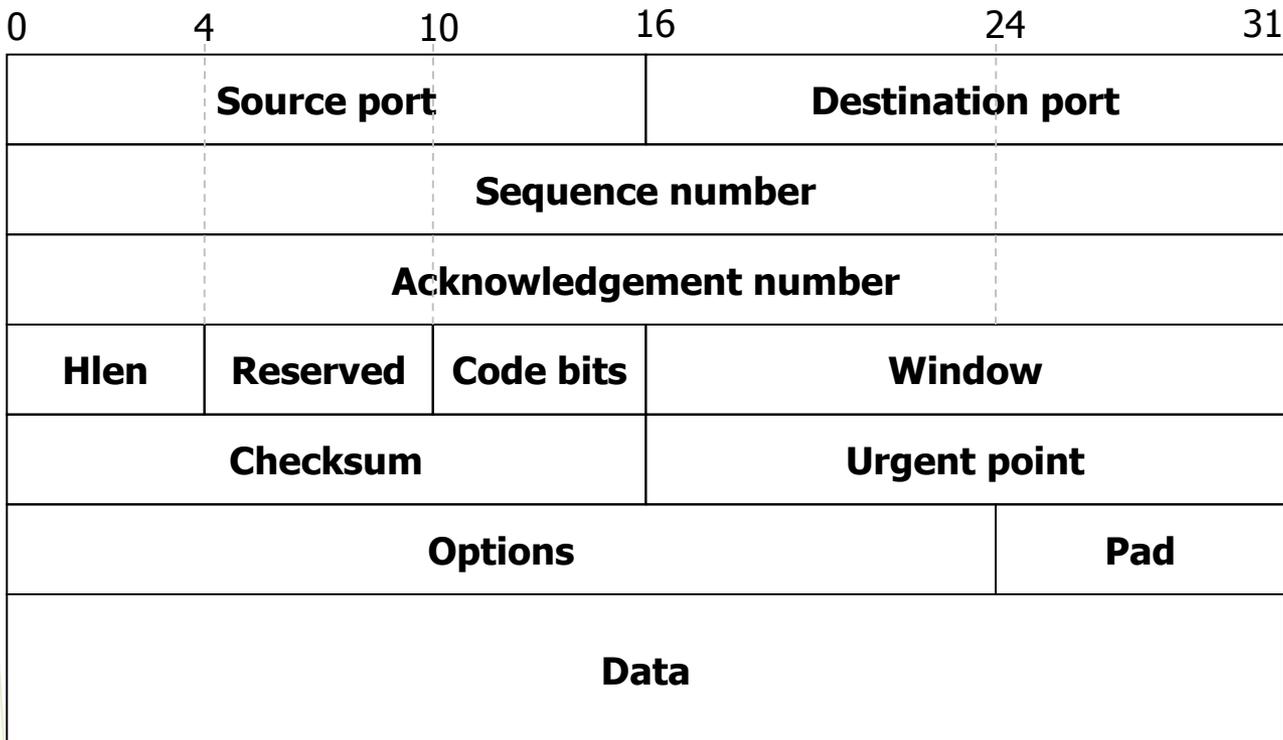
■ Fundamentos

- Define mecanismos integrados de controle de erro e seqüência
 - Asseguram a entrega do fluxo de dados na seqüência correta e sem erros
- Define mecanismo de controle de fluxo
 - Regula e compatibiliza a taxa de transmissão das unidades envolvidas
 - Evita descarte de segmentos por falta de recursos da estação destino



Protocolo TCP

- Formato do segmento TCP



Protocolo TCP

■ Campos do segmento

■ Hlen

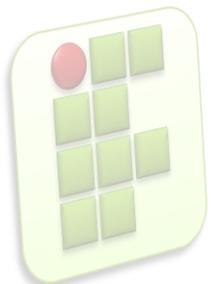
- Tamanho do cabeçalho em unidades de 4 bytes;

■ Reserved

- Reservado para uso futuro (Não utilizado);

■ Checksum

- Assegura a integridade do segmento;



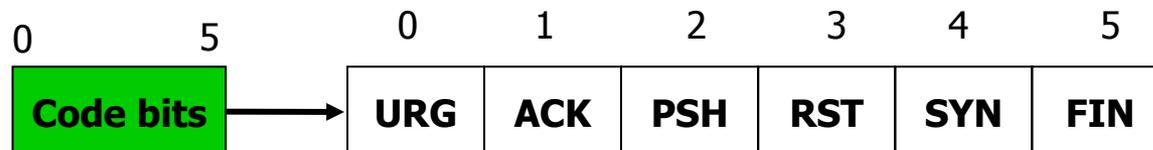
Protocolo TCP

■ Campos do segmento

■ Code bits

■ Indica propósito e conteúdo do segmento

- URG: Dados urgentes
- ACK: reconhecimento
- PSH: mecanismo de push(encaminhar segmento)
- RST: abordo de conexão (reset)
- SYN: Abertura de conexão
- FIN: fechamento de conexão



Protocolo TCP

■ Campos do segmento

■ Options

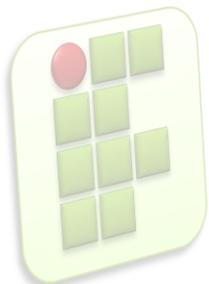
- Lista variável de informações opcionais
 - MSS – Maximum Segment Size;
 - Opção sinalizada pelo segmento SYN;
- Torna o tamanho do cabeçalho variável

■ Padding

- Bits 0 que tornam o segmento múltiplo de 32 bits

■ Data

- Dados do segmento



Protocolo TCP

■ Portas

■ *Source port*

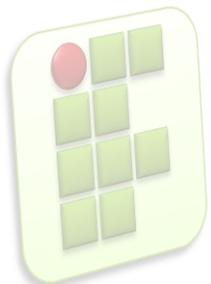
- Porta associada ao processo de origem

■ *Destination port*

- Porta associada ao processo de destino

■ *Endpoint(Socket)*

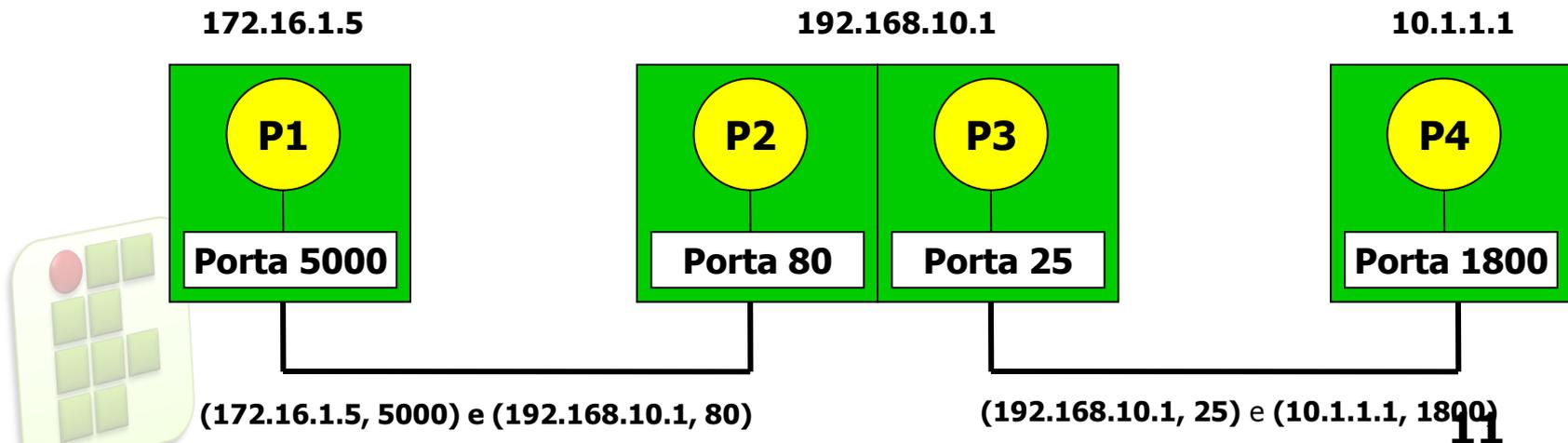
- Definido pelo par (Endereço IP, porta)
- Identifica de forma única cada porta ou ponto de comunicação na inter-rede
- Também conhecido como Socket



Protocolo TCP

■ Conexão

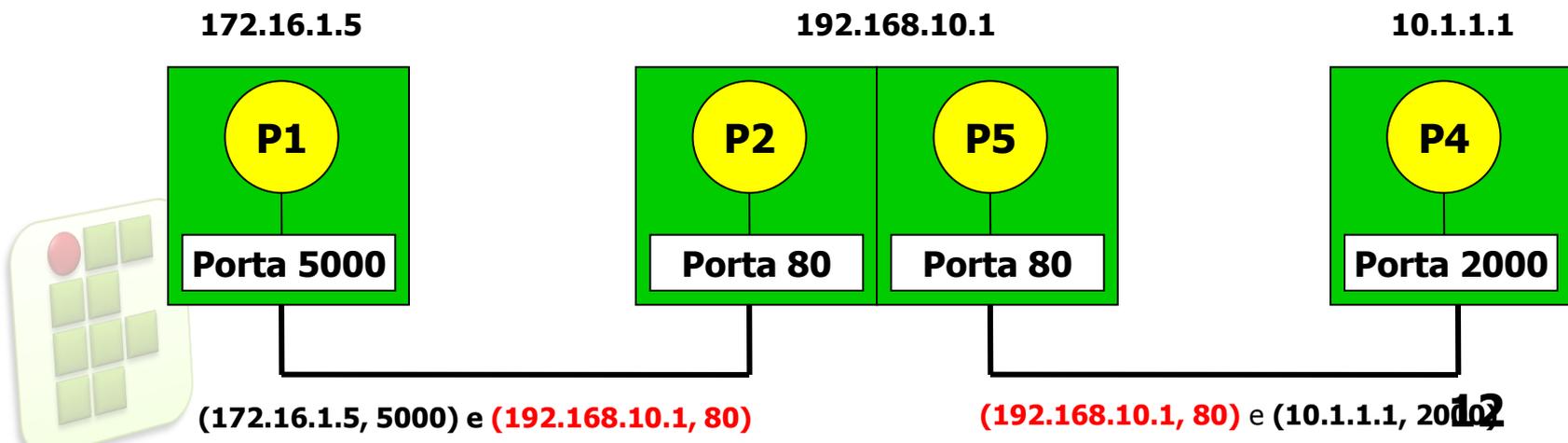
- Cada conexão é identificada por um par de endpoints
- Também conhecida como *Socket pair*
- Várias conexões por estação



Protocolo TCP

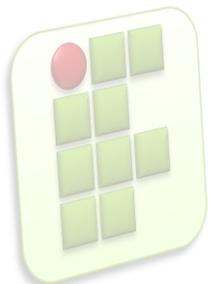
■ Conexão

- Cada *endpoint* local **pode** participar de diversas conexões com *endpoints* remotos
 - Compartilhamento de *endpoints*
 - O Sistema Operacional deve garantir que o par de *endpoint* da conexão é único



Protocolo TCP

- Demultiplexação de mensagens
 - Segmentos recebidos são associados às **conexões**, não apenas as portas
 - Avalia o par de *endpoints* da conexão
 - Portas origem e destino são obtidas do segmento recebido
 - Endereço IP origem e destino são obtidos do datagrama IP
 - Cada conexão possui um *buffer* de transmissão e um *Buffer* de recepção em cada extremidade



Protocolo TCP

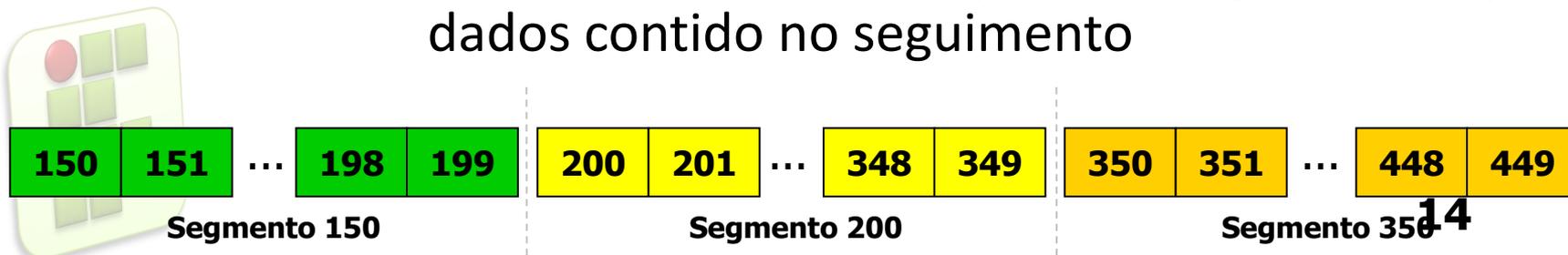
■ Controle de seqüência

■ Fluxo de dados é tratado como uma seqüência de bytes

- Cada byte possui um número de seqüência
- Numeração nem sempre começa em 0 (zero)
- Negociado no estabelecimento da conexão

■ Campo *Sequence number*

- Indica o número de seqüência do primeiro byte de dados contido no seguimento



Protocolo TCP

■ Controle de seqüência

Números de seqüência:

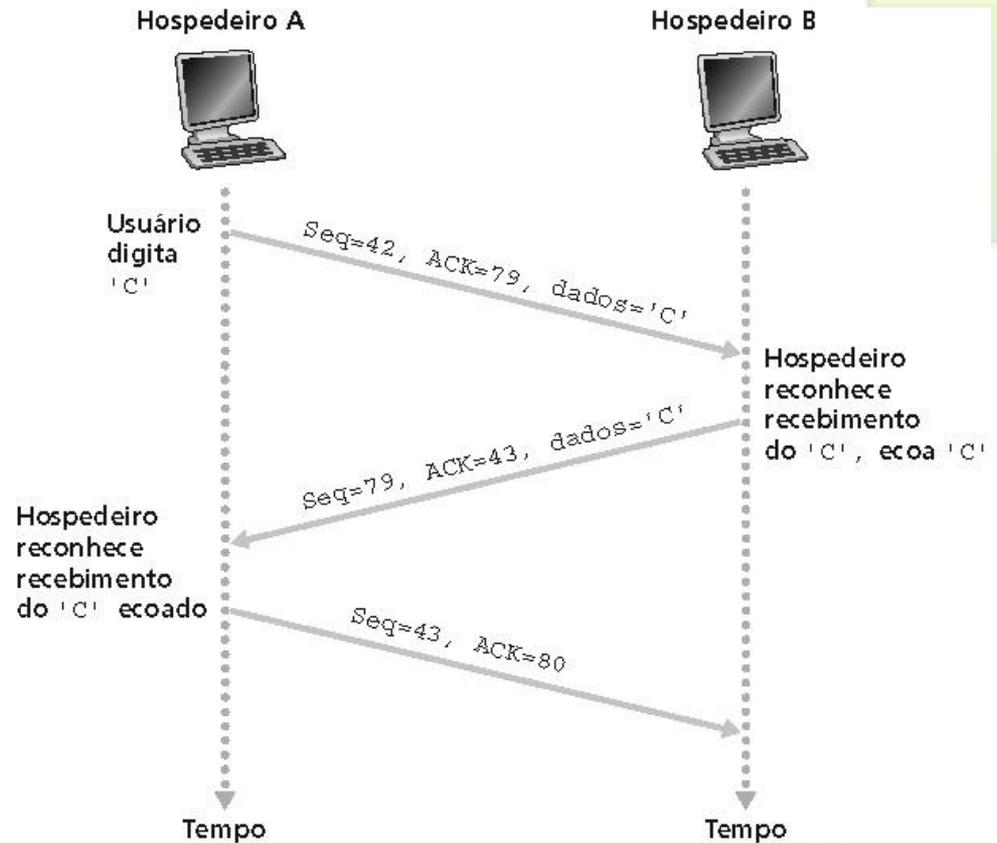
- Número do primeiro byte nos segmentos de dados

ACKs:

- Número do próximo byte esperado do outro lado
- ACK cumulativo

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

- A especificação do TCP não define, fica a critério do implementador



Protocolo TCP

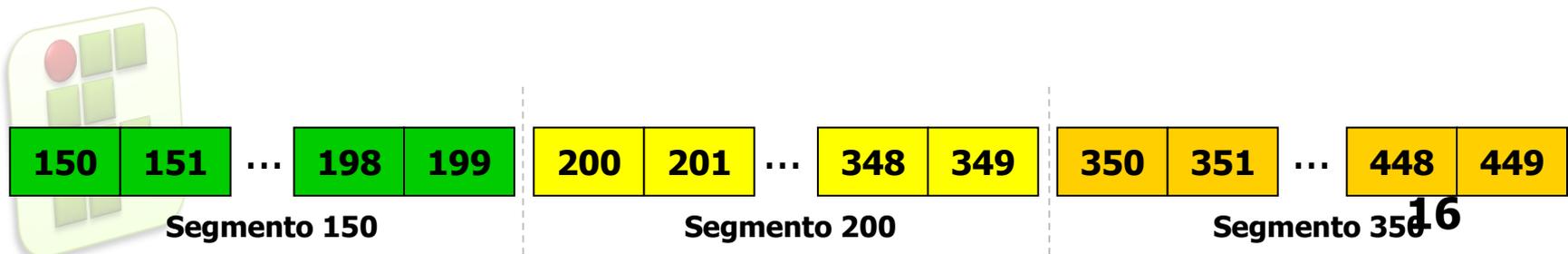
■ Controle de erros

■ Reconhecimento positivo

- Destino retorna uma mensagem indicando o correto recebimento do segmento
- Reconhecimento pode pegar carona no segmento de dados do fluxo inverso

■ Reconhecimento cumulativo

- Diversos segmentos consecutivos podem ser reconhecidos em uma única mensagem

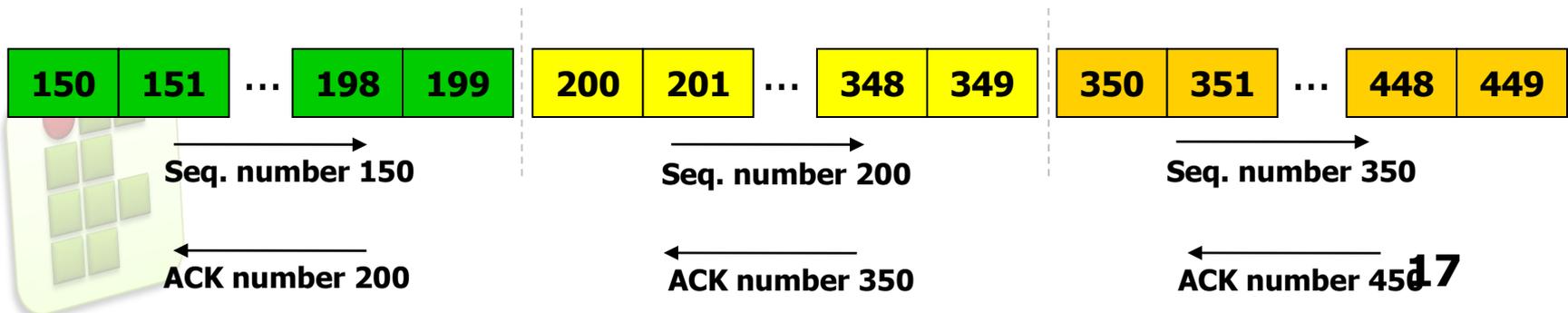


Protocolo TCP

■ Controle de erros

■ *Acknowledgment number*

- Indica o número de sequência do próximo byte que espera receber
- Indica o correto recebimento dos bytes com número de sequência anterior
- Bit ACK do **Code Bits** deve ser ativado

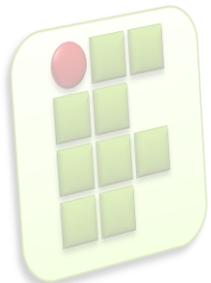


Protocolo TCP

■ Controle de erros

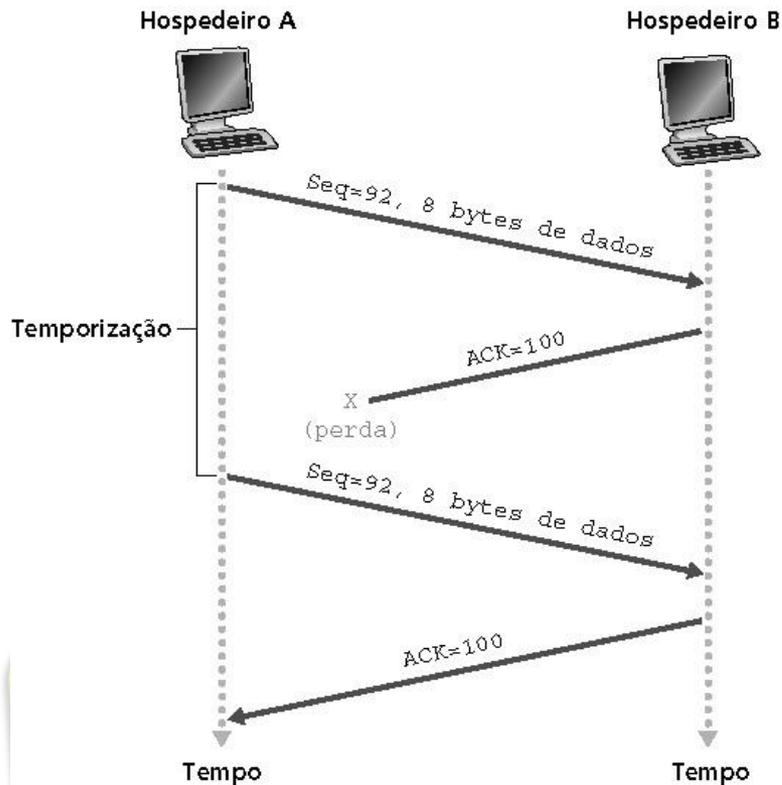
■ Realizado através de Retransmissão

- Origem adota um temporizador para cada segmento enviado
- Segmento é retransmitido quando a origem não recebe o **reconhecimento** (*ack*) antes de expirar o temporizador
- Temporizador é reativado em cada retransmissão

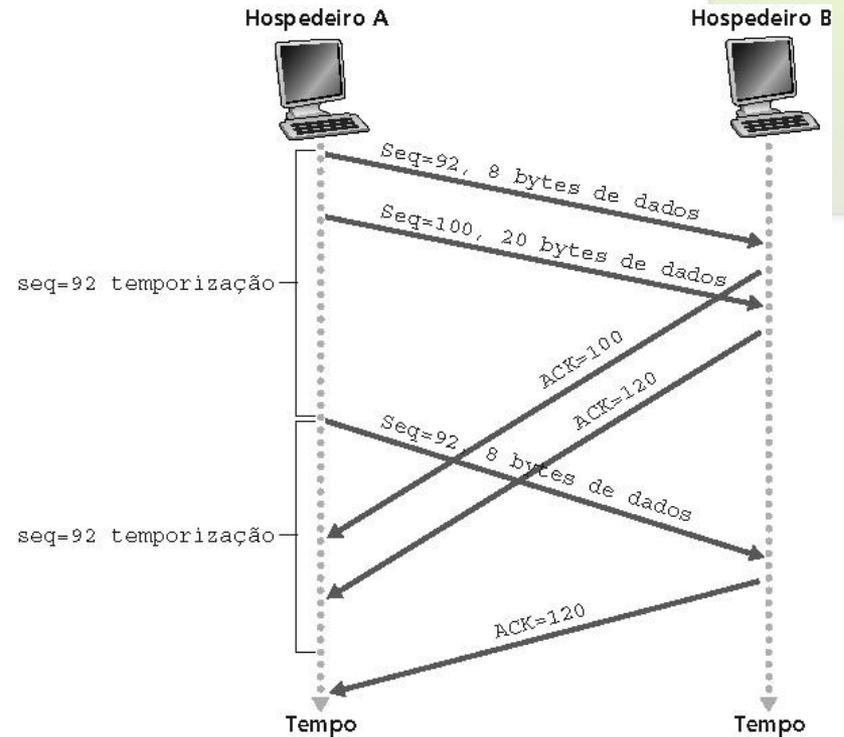


Protocolo TCP

■ Controle de erros - Cenários



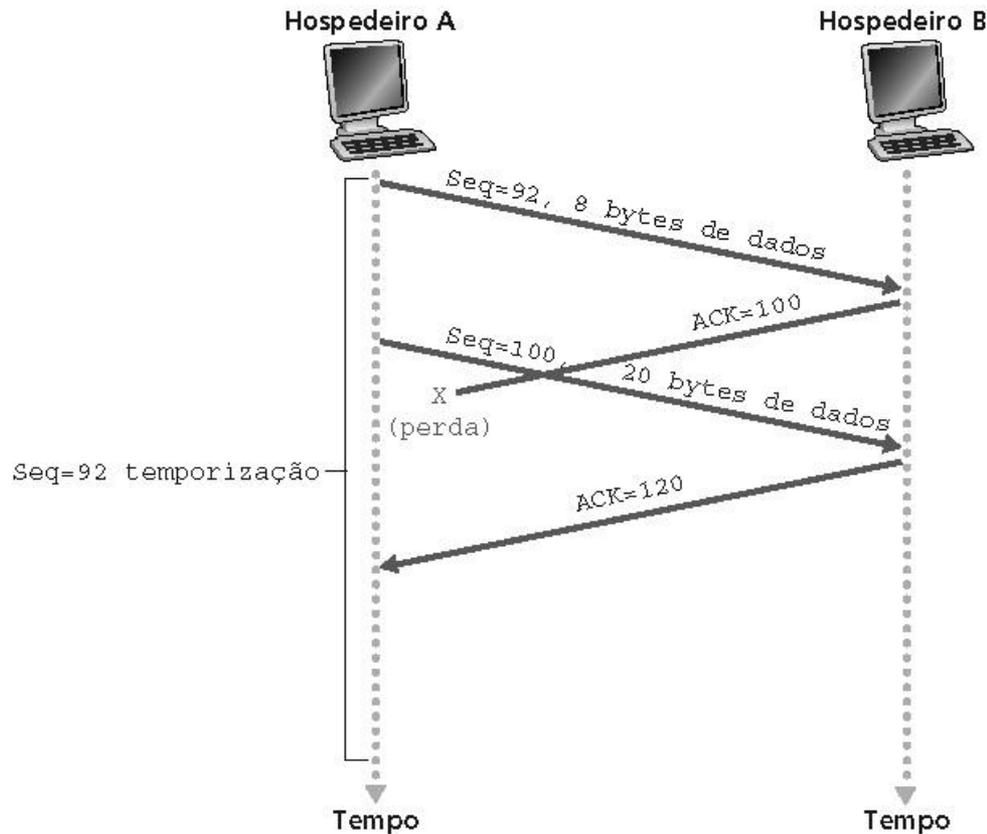
Cenário com perda do ACK



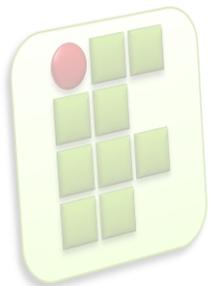
Temporização prematura, ACKs cumulativos 19

Protocolo TCP

■ Controle de erros - Cenários



Cenário de ACK cumulativo



Protocolo TCP

■ Controle de fluxo

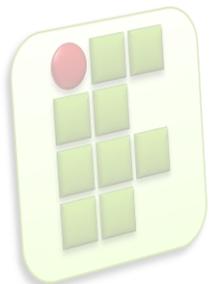
■ Objetivo

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

■ Implementação

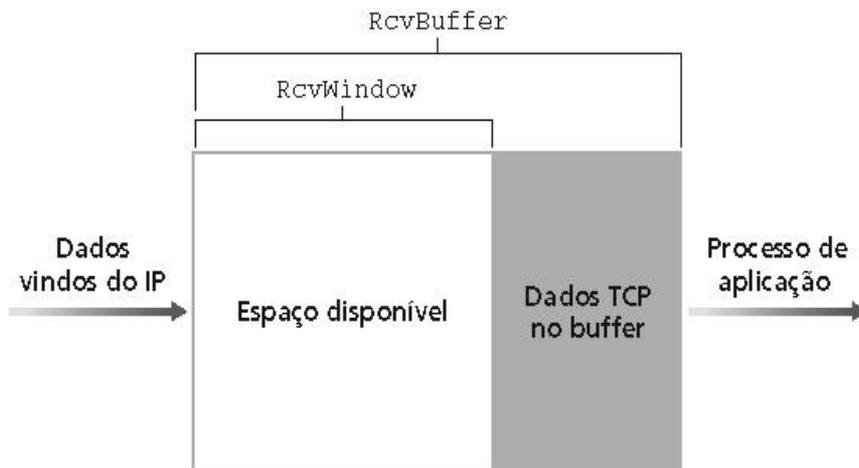
■ Janela deslizante

- Entidades negociam o número de bytes adicionais que podem ser recebidos a partir do último reconhecimento
 - Destino define o tamanho de sua janela de recepção em cada segmento
 - Origem atualiza o tamanho de sua janela de transmissão a cada reconhecimento
 - Reconhecimento deslocam a janela de transmissão da origem para o primeiro *byte* sem reconhecimento



Protocolo TCP - Controle de fluxo

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

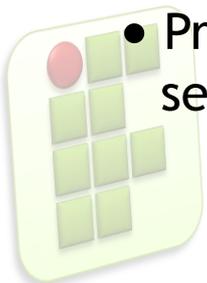


Controle de fluxo

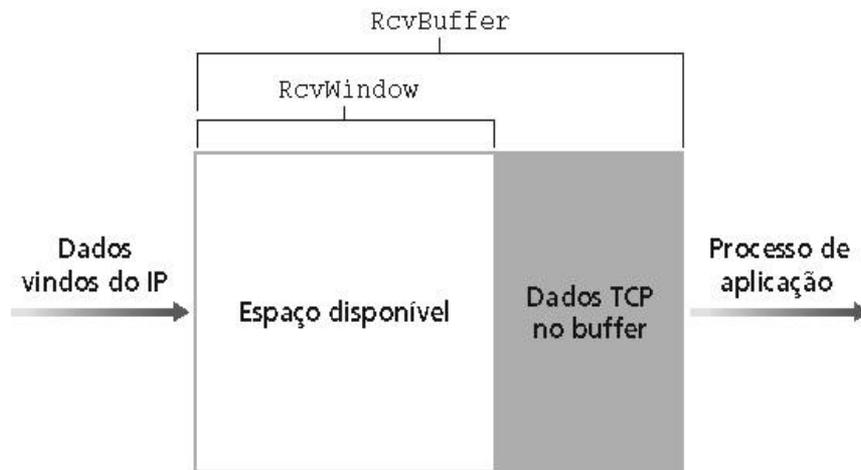
Transmissor não deve esgotar os buffers de recepção enviando dados rápido demais

- Serviço de **speed-matching**: encontra a taxa de envio adequada à taxa de vazão da aplicação receptora

- Processos de aplicação podem ser lentos para ler o buffer



Protocolo TCP - Controle de fluxo



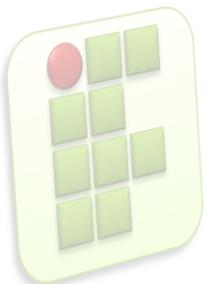
- Receptor informa a área disponível incluindo valor **RcvWindow** nos segmentos
- Transmissor limita os dados não confirmados ao **RcvWindow**
- Garantia contra overflow no buffer do receptor

(suponha que o receptor TCP descarte segmentos fora de ordem)

- Espaço disponível no buffer

= **RcvWindow**

= **RcvBuffer - [LastByteRcvd - LastByteRead]**



Protocolo TCP

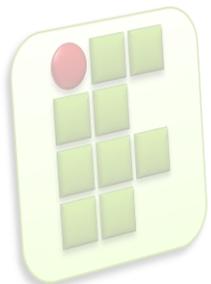
- Controle de fluxo

- Campo *Window*

- Sinaliza o tamanho da janela de recepção da entidade em cada segmento enviado

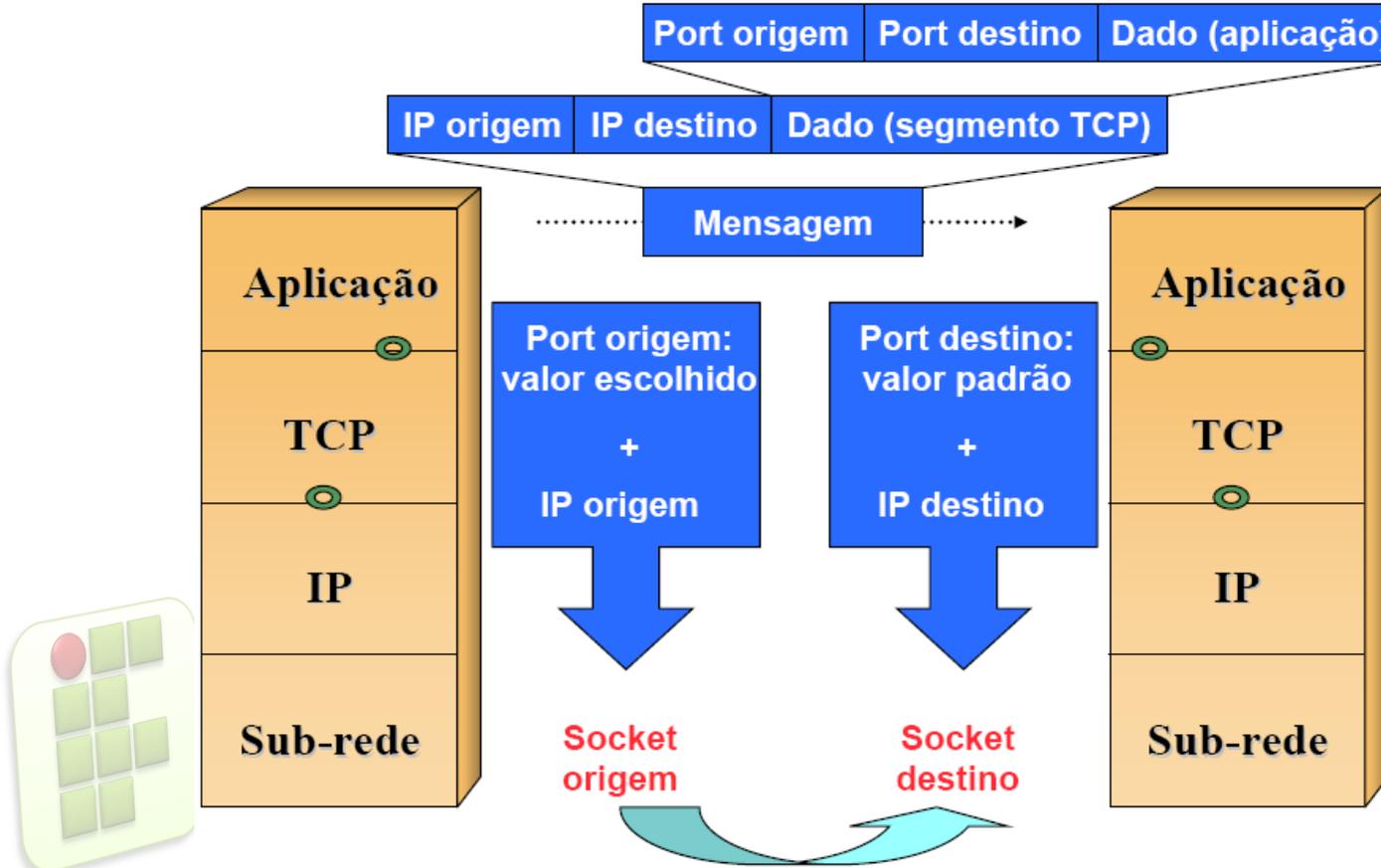
- Applet on-line

- http://wps.aw.com/br_kurose_redes_3/40/10271/2629597.cw/index.html



Protocolo TCP

- Processo de estabelecimento de conexões

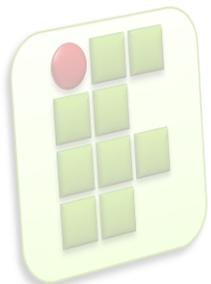


Protocolo TCP

■ Estabelecimento de conexões

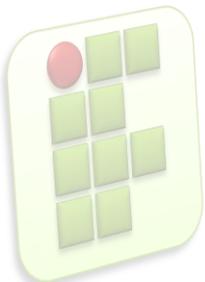
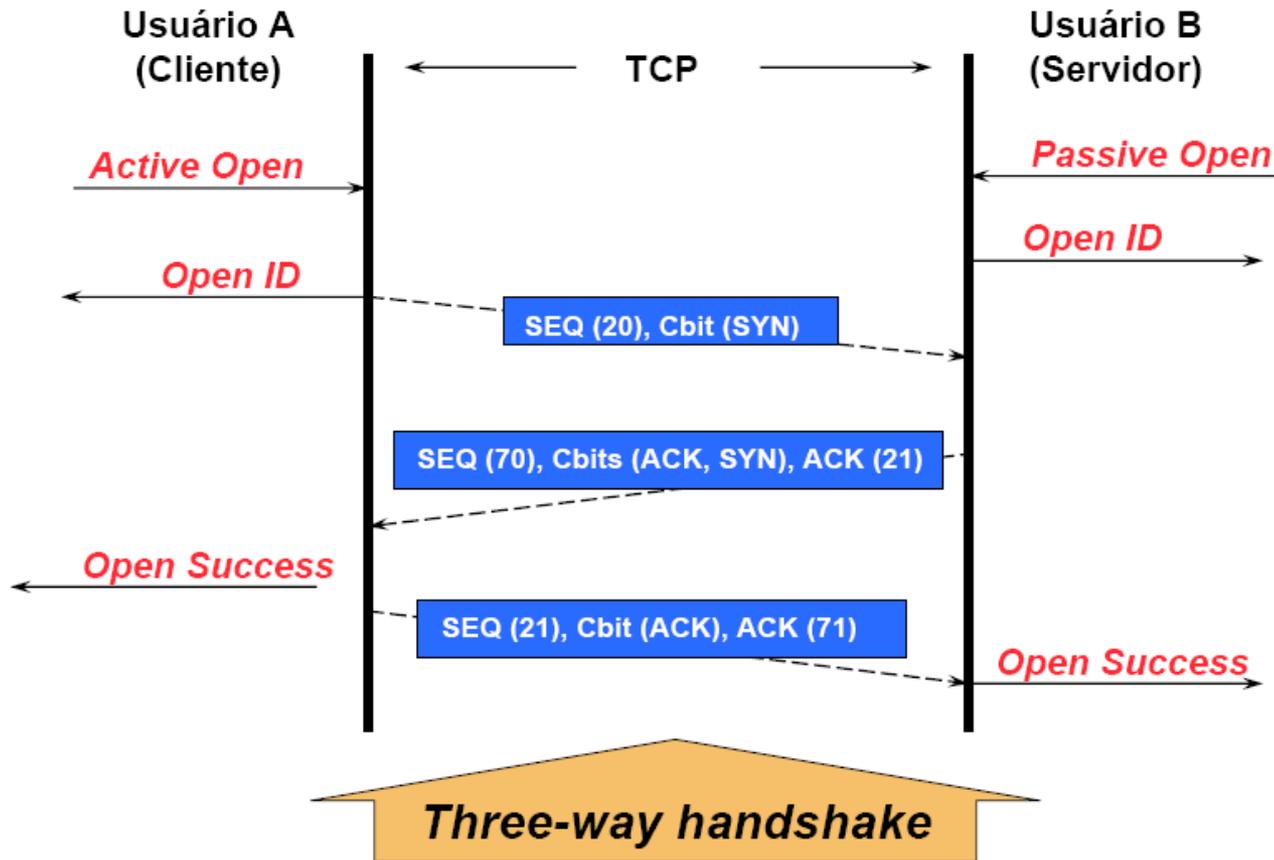
■ *Three way handshake*

- Negocia e sincroniza o valor inicial dos números de seqüência em ambas as direções
- Baseado na arquitetura cliente-servidor
- O servidor deve está com a porta aberta em estado de escuta (*Listening*)



Protocolo TCP

- Estabelecimento de conexões



Protocolo TCP

■ Transmissão de dados

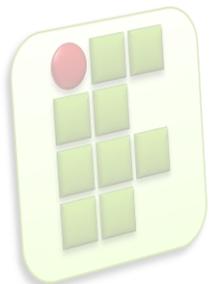
■ Entrega de dados “*fora-de-banda*”

■ Campo *Urgent Point*

- o transmissor transmite o dado urgente na área de dados e seta o bit URG (campo *Codebits*), indicando a posição no segmento onde o dado urgente terminou
- O receptor deve notificar a aplicação sobre a chegada do dado urgente tão logo quanto possível

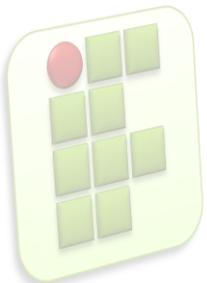
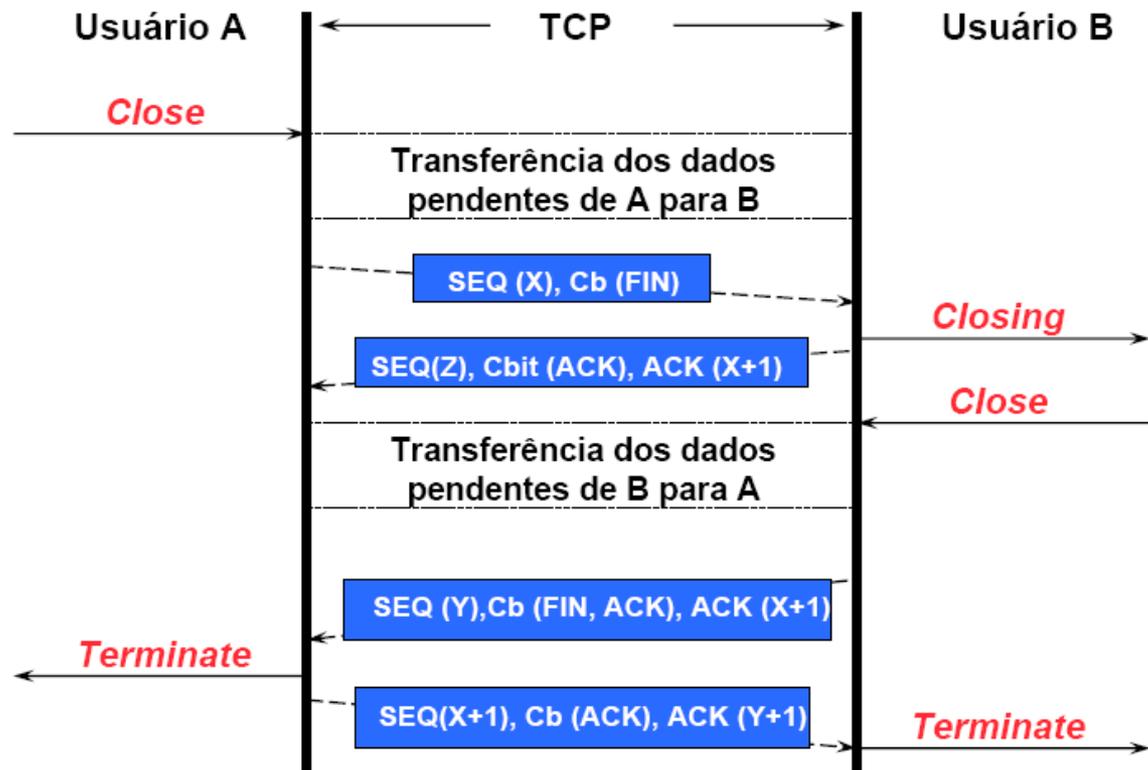
■ Mecanismo de *Push*

- Aplicação avisa ao TCP para enviar o dado imediatamente
- Força a geração de um segmento com os dados já presentes no *Buffer*
- Não aguarda o preenchimento do *Buffer*
- Segmentos gerados pelo mecanismo de *PUSH* são marcados com o flag PSH no campo *codebits*



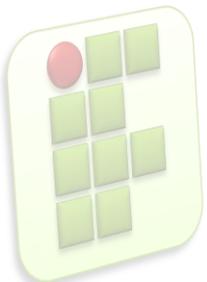
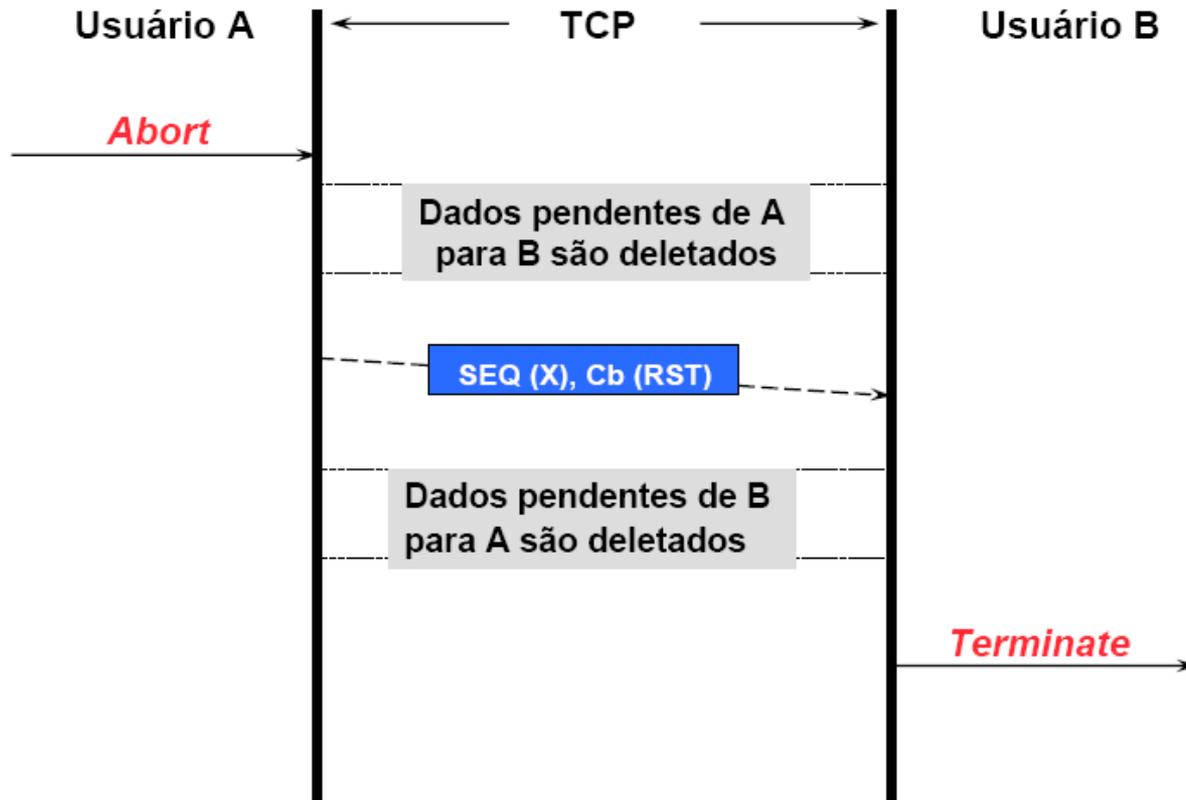
Protocolo TCP

- Fechamento de conexão (Liberação ordenada)
 - Ocorre separadamente em cada direção da conexão



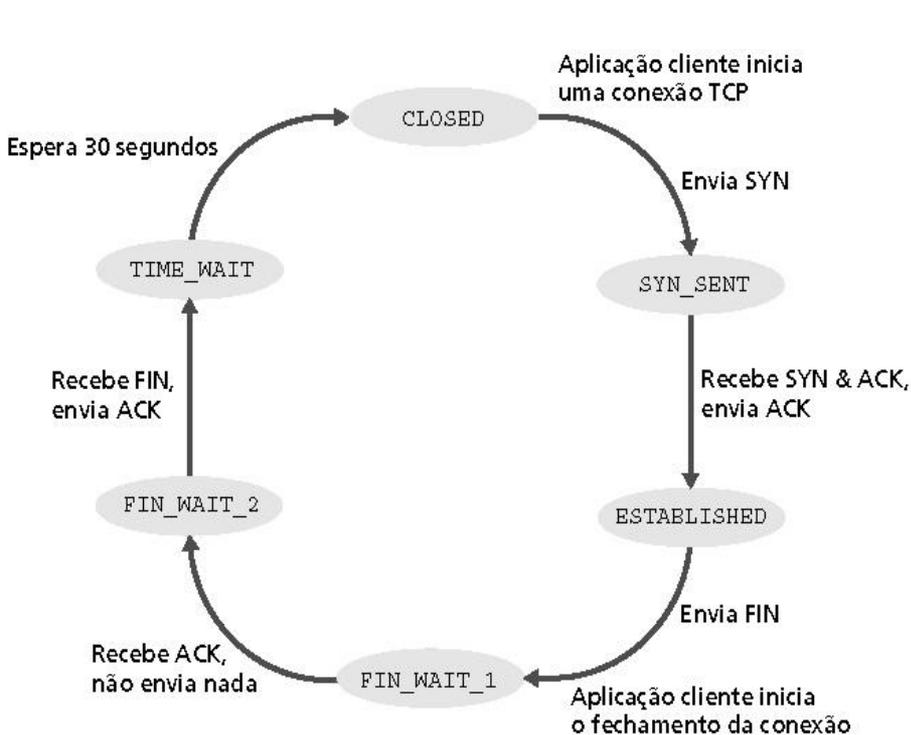
Protocolo TCP

- Fechamento de conexão (Término abrupto)

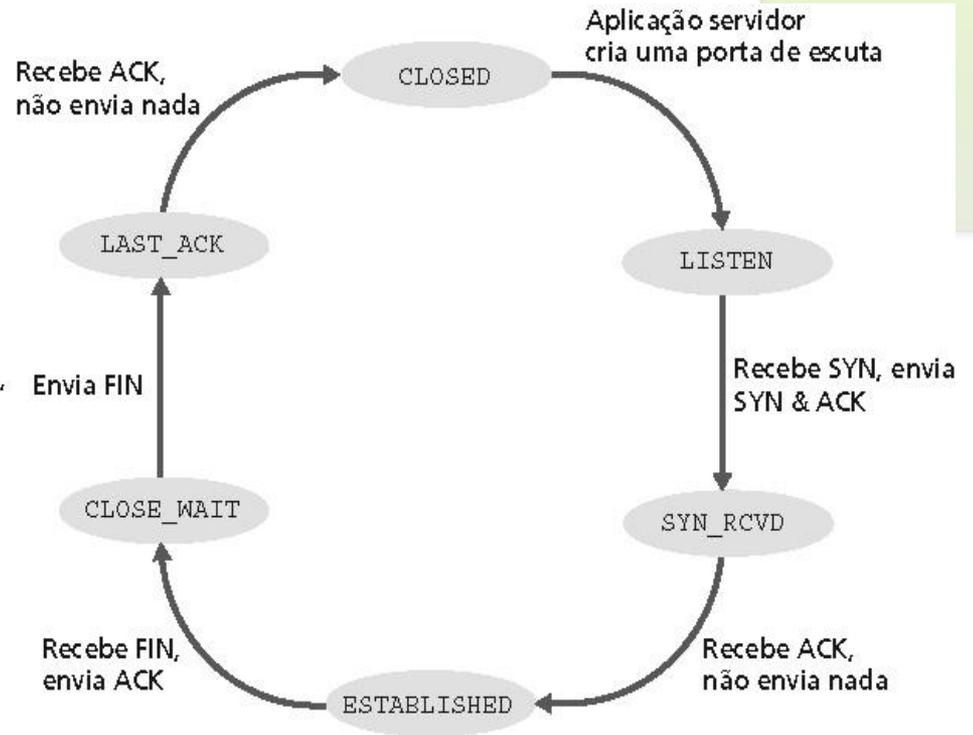


Protocolo TCP

- Estados das conexões



Estados do cliente



Estados do servidor

Referências

- Comer, Douglas E., Interligação de Redes Com Tcp/ip
- James F. Kurose, Redes de Computadores e a Internet
- Escola Superior de Redes, Arquitetura e Protocolos de Redes TCP/IP
- Escola Superior de Redes, Roteamento avançado

