

Camada de Transporte

Protocolos TCP e UDP

Protocolo UDP

- Protocolo de transporte da Internet “sem gorduras”
- Serviço “best effort” , 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



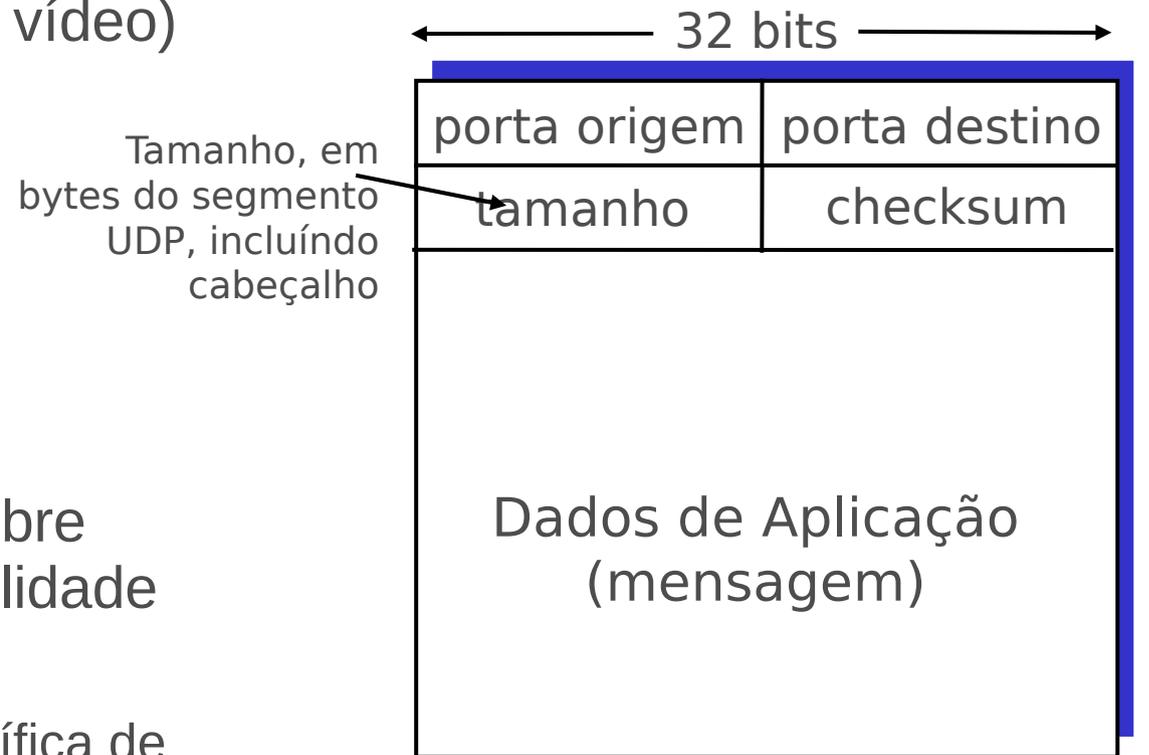
Vantagens do UDP

- Não há estabelecimento de conexão (que pode redundar em atrasos)
- Simples: não há estado de conexão nem no transmissor, nem no receptor
- Cabeçalho de segmento reduzido
- Não há controle de congestionamento: UDP pode enviar segmentos tão rápido quanto possível



UDP

- Muito usado por aplicações de multimídia contínua (Voz e vídeo)
 - tolerantes à perda
 - sensíveis à taxa
- Outros usos do UDP
 - DNS
 - SNMP
- Transferência confiável sobre UDP: acrescentar confiabilidade na camada de aplicação
 - recuperação de erro específica de cada aplicação



formato do segmento UDP

UDP Checksum

- Objetivo: detectar “erros” (ex., bits trocados) no segmento transmitido
- Transmissor:
 - computa o checksum do segmento a enviar
 - coloca o valor do checksum no campo de checksum do UDP
- Receptor:
 - computa o checksum do segmento recebido
 - verifica se o checksum calculado é igual ao valor do campo checksum:
 - Checksum diferente - erro detectado
 - Checksum igual - não há erros



Protocolo TCP

- Ponto-a-ponto:
 - um transmissor, um receptor
- Confiável, seqüencial de bytes stream.
- Pipelined: transmissão de vários pacotes em confirmação
- Controle de congestão e de fluxo definem tamanho da janela
- Usa buffers de transmissão e de recepção

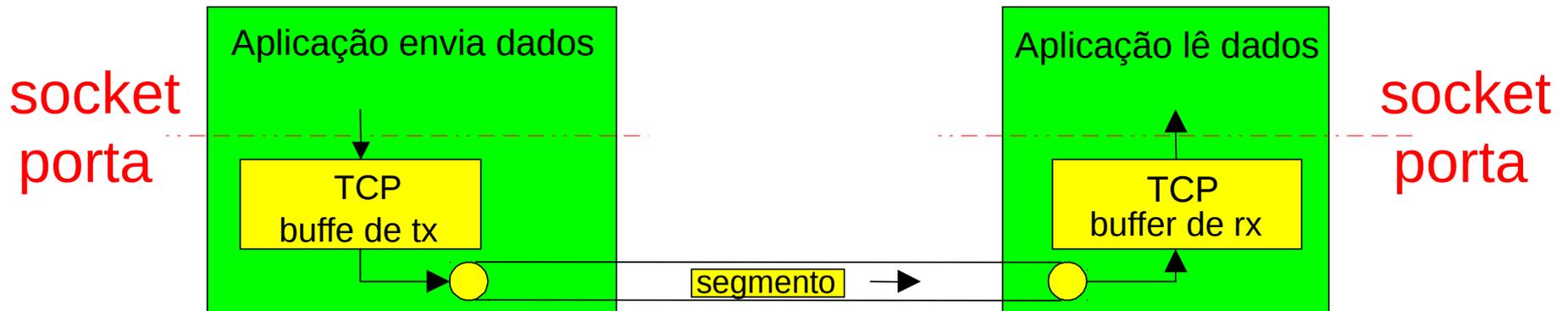


Protocolo TCP

- Dados full-duplex:
- Transmissão bi-direcional na mesma conexão
- Orientado à conexão:
- Handshaking: troca de mensagens de controle, inicia o estado do transmissor e do receptor antes da troca de dados
- Controle de fluxo:
 - transmissor não esgota a capacidade do receptor

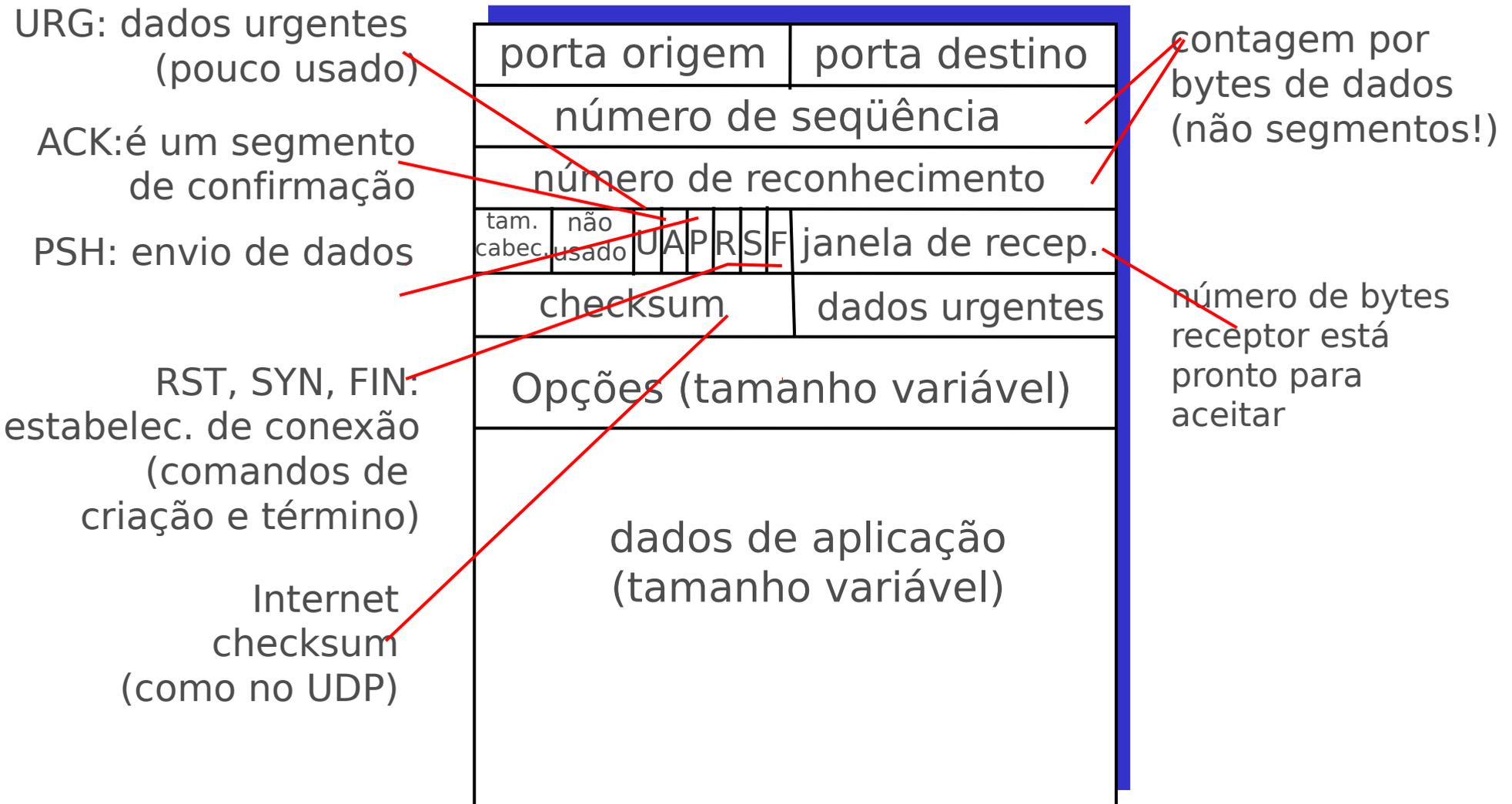


Protocolo TCP



Protocolo TCP

32 bits



Campos do TCP

- Portas de Origem e Destino (16 bits cada): Números das porta que indicam os programas da camada de aplicação
- Número de Sequência (32 bits): Usado para manter a ordem dos segmentos
- Número de Ack: se é um pacote de confirmação indica qual o segmento que está sendo confirmado
- Hlen: Header Length tamanho do cabeçalho TCP em unidades de 4 bytes (32 bits). Padrão 5.



Campos do TCP

- Flags
 - URG – Há dados urgentes a serem entregues neste segmento
 - ACK – Este é um segmento de confirmação (acknowledge)
 - PSH – Push indica que este é um segmento que contém dados
 - RST – Reset a conexão precisa ser reiniciada (fechada imediatamente)
 - SYN – Estabelece uma nova conexão
 - FIN – Finaliza uma conexão



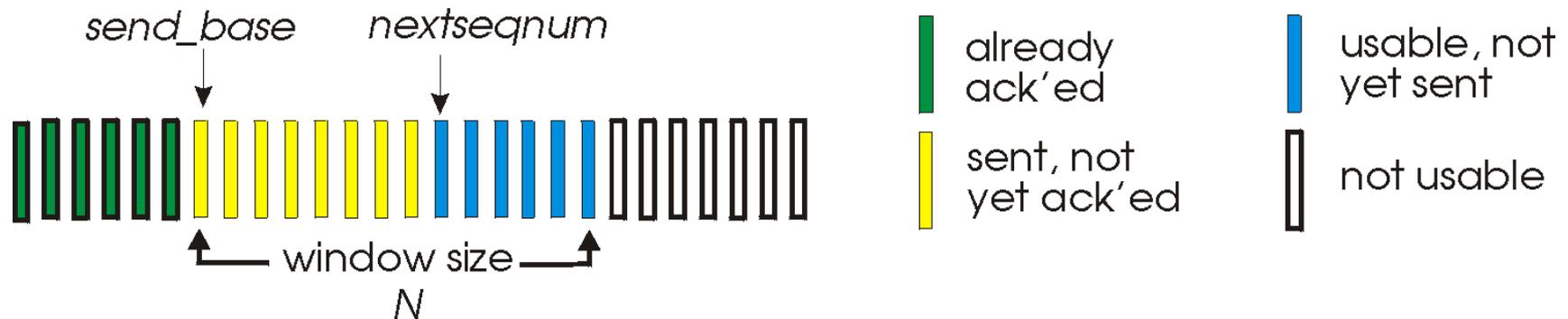
Campos do TCP

- Window – indica qual o espaço disponível na memória local para recepção de dados
- Urgent Pointer – Indica que parte dos dados sendo transmitidos é urgente
- Options – É opcional não necessariamente presente, pode trazer informações como tamanho máximo do segmento etc.
- Pad – Enchimento dados aleatórios adicionados para fechar os 4 bytes se estiver usando *options*
- Data – os dados efetivamente transmitidos



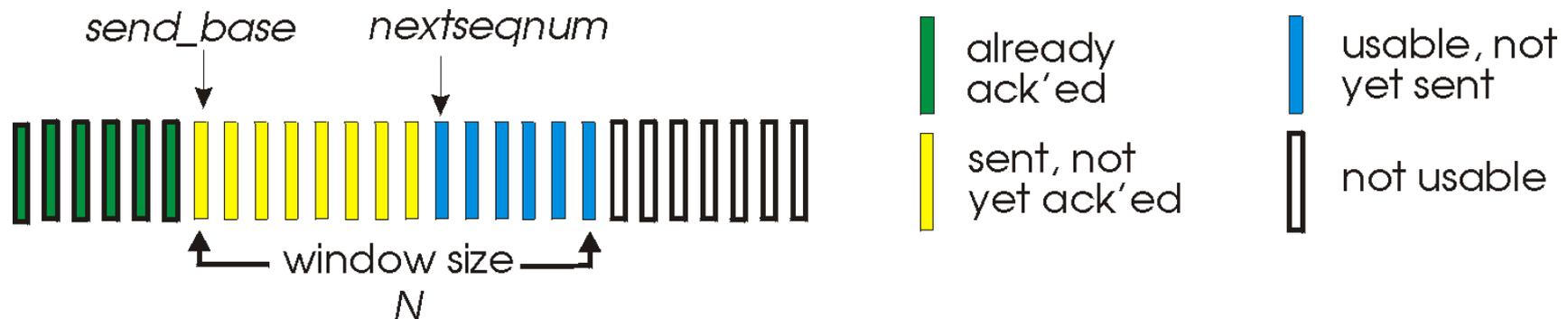
Janelamento

- Transmissor:
 - Número de seqüência com k bits no cabeçalho do pacote
 - “janela” de até N pacotes não reconhecidos, consecutivos, são permitidos

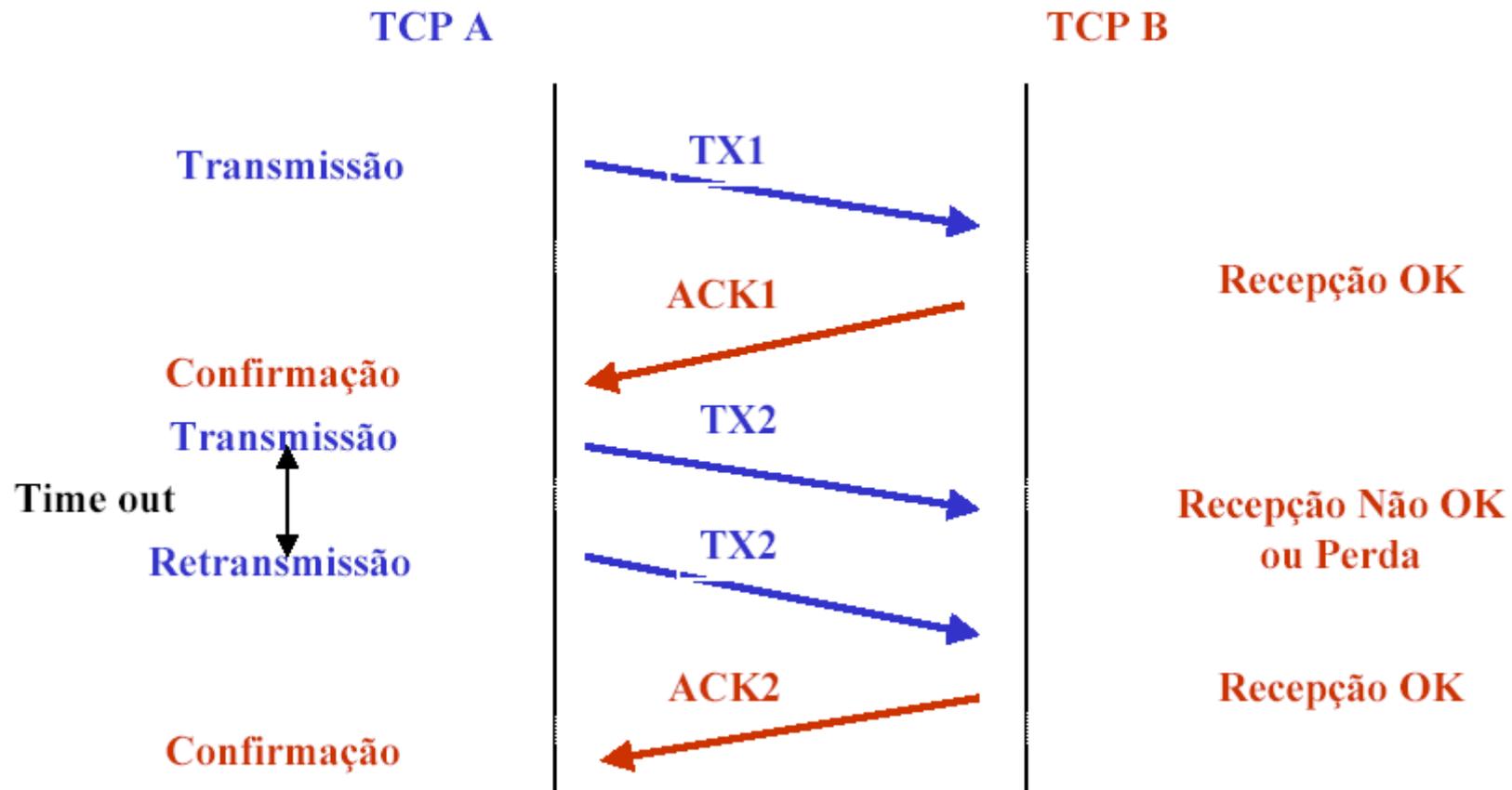


Ack

- Confirmação de recebimento
- ACK(n): reconhece todos os pacotes até o número de sequência N (incluindo este limite). “ACK cumulativo”
- Há um temporizador para cada pacote enviado e não confirmado
- Timeout(n): retransmite pacote n e todos os pacotes com número de sequência maior que estejam dentro da janela

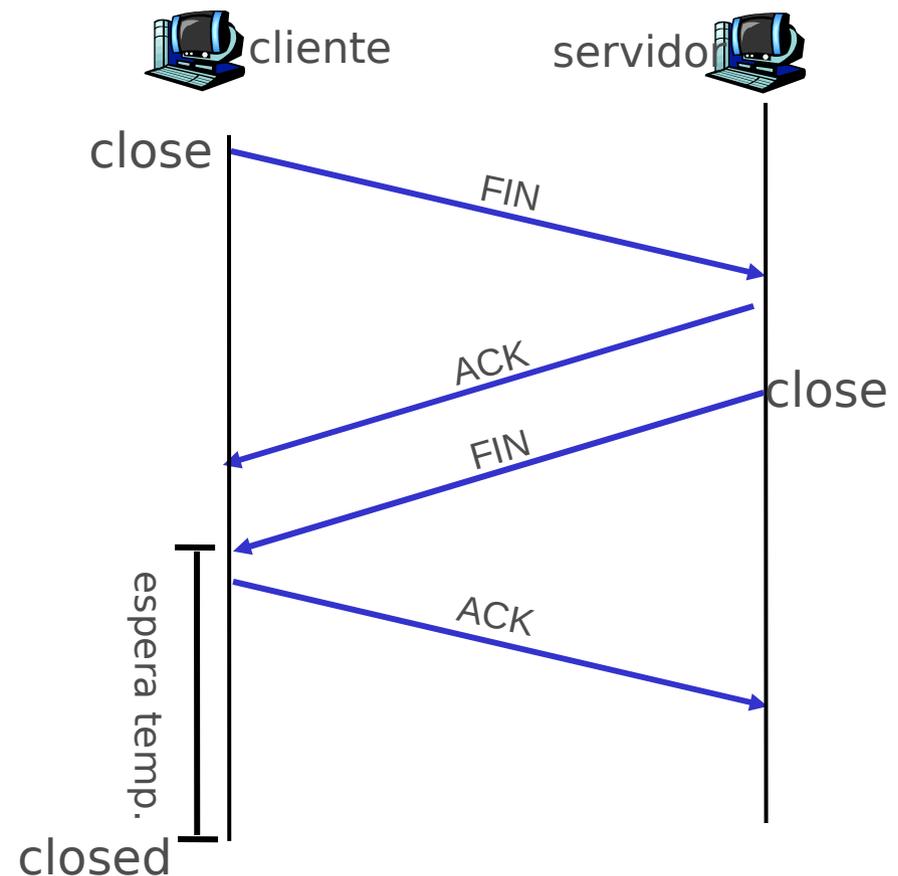


Retransmissão de segmentos



Fechando a conexão

- **Passo 1:** o cliente envia o segmento TCP FIN ao servidor
- **Passo 2:** servidor recebe FIN, responde com ACK. Fecha a conexão, envia FIN.



Fechando a conexão

- **Passo 3:** cliente recebe FIN, responde com ACK.
 - Entra “espera temporizada” - vai responder com ACK a FINs recebidos
- **Passo 4:** servidor, recebe ACK. Conexão fechada.

