Camada de Transporte

Protocolos TCP e UDP

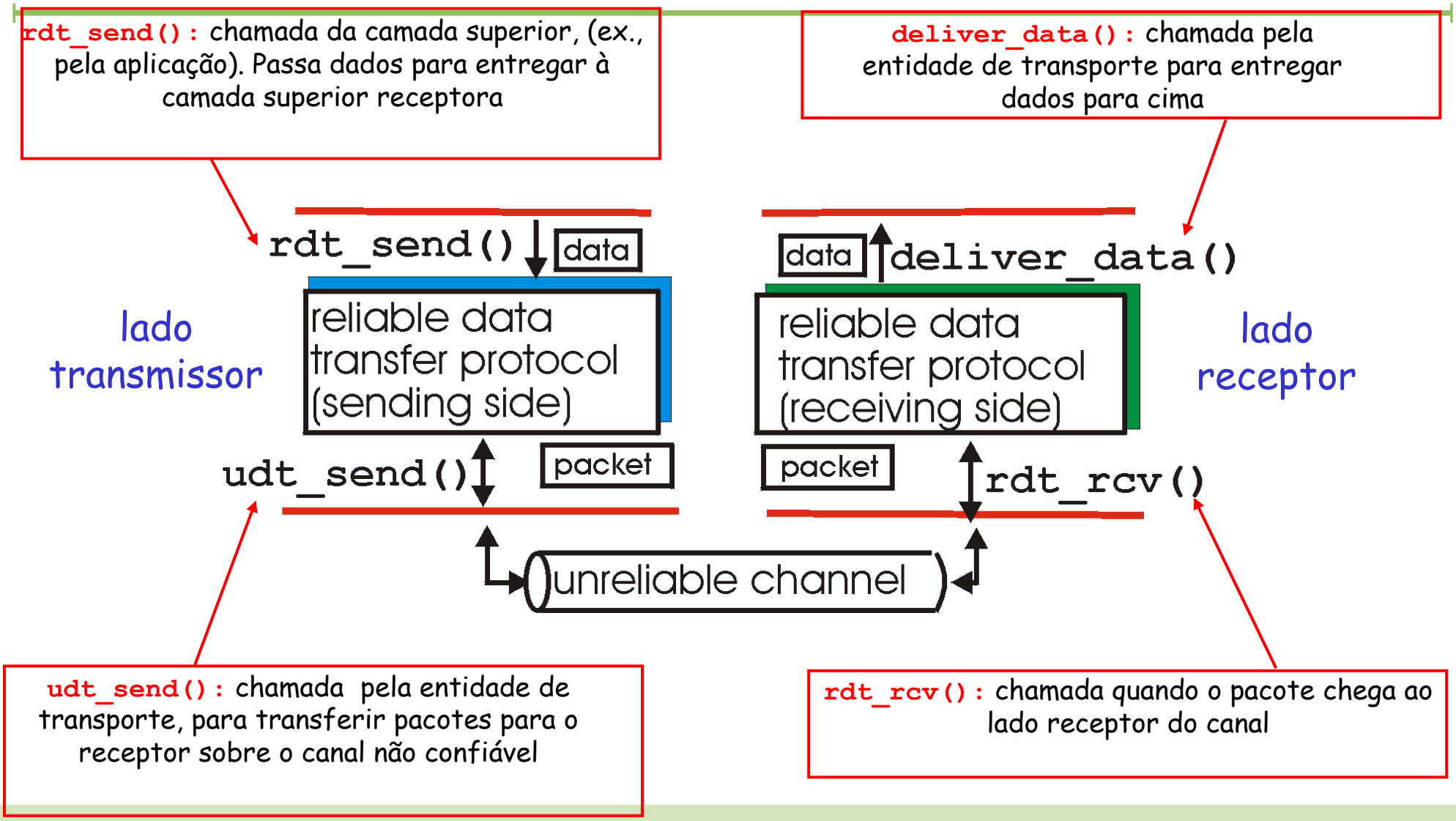
O estabelecimento de conexão

-
- Como estabelecer a conexão de maneira confiável?
- Handshake de 3 vias
  - SYN
  - SYN ACK
  - ACK

Transferência Confiável de Dados

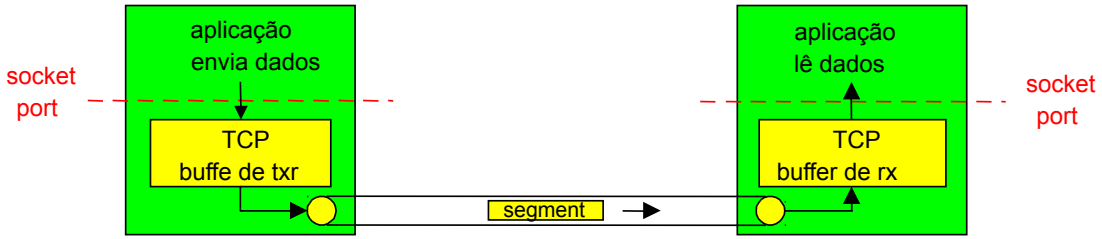
- Importante nas camadas de aplicação, transporte e enlace
- Top-10 na lista dos tópicos mais importantes de redes!
- Caracteristicas dos canais não confiáveis determinarão a complexidade dos protocolos confiáveis de transferência de dados (rdt)

Transferência confiável: o ponto de partida



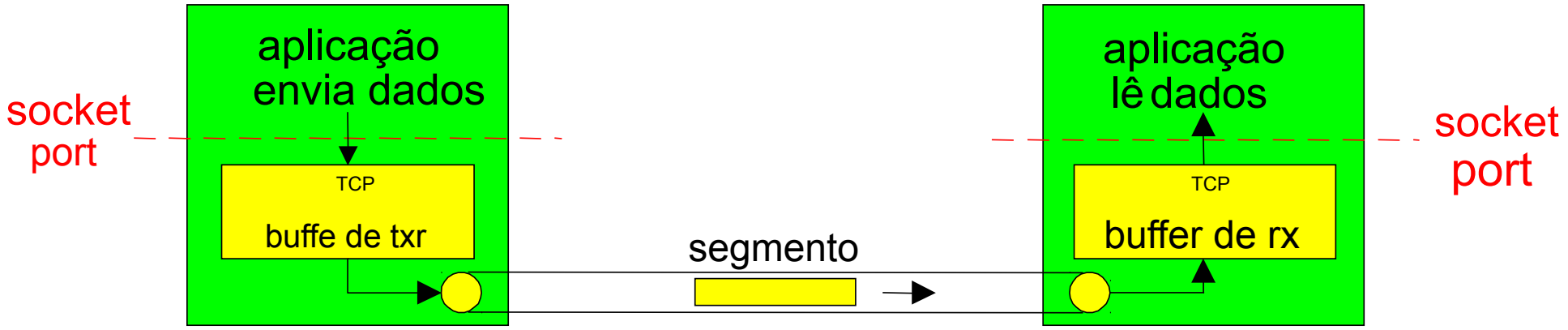
TCP: Overview RFCs: 793, 1122, 1323, 2018, 2581

- ponto-a-ponto:
  - um transmissor, um receptor
- confiável, seqüêncial *byte stream:*
  - não há contornos de mensagens
- pipelined: (transmissão de vários pacotes em confirmação)
  - Controle de congestão e de fluxo definem tamanho da janela
- *buffers de transmissão e de recepção*

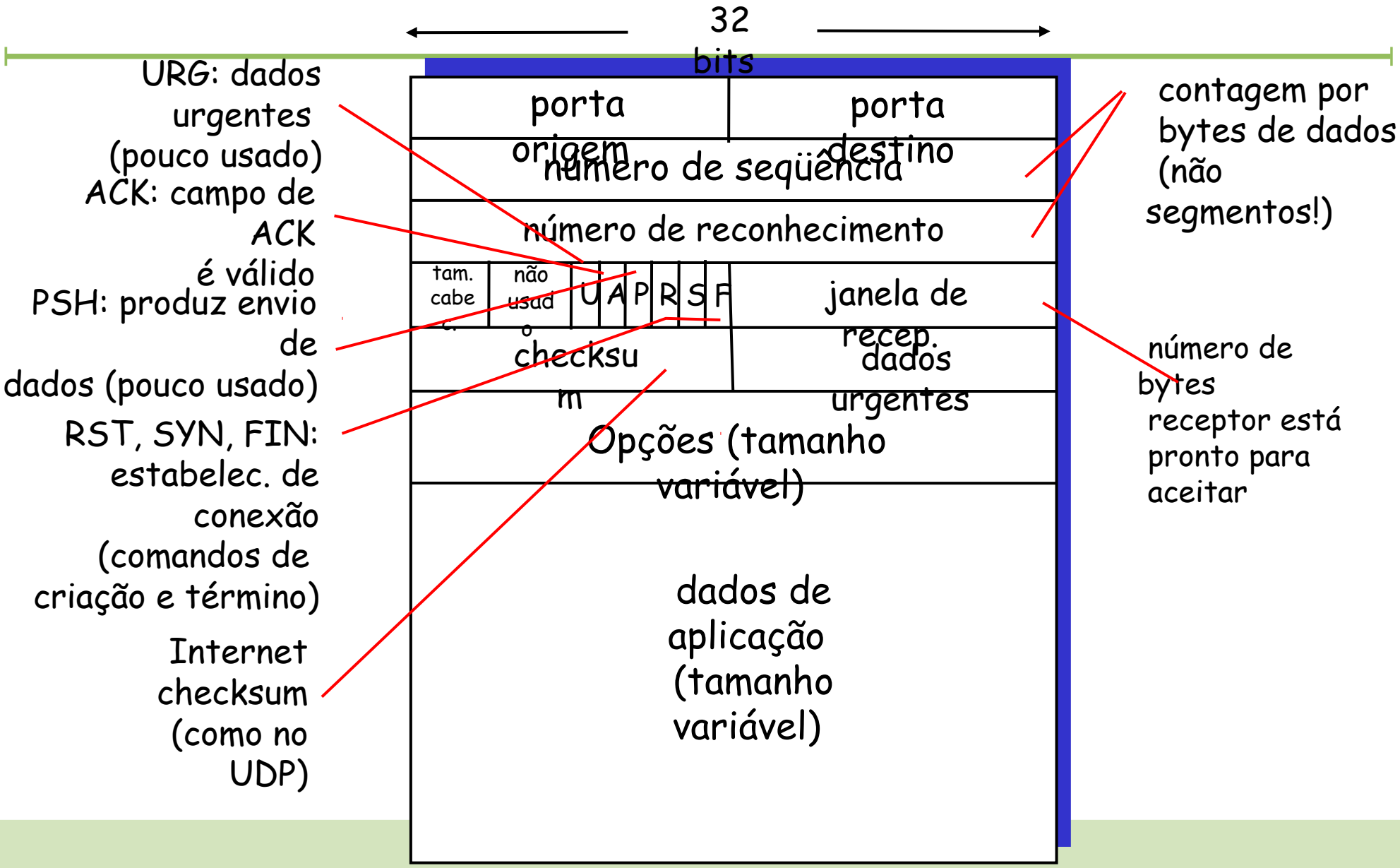


- dados full-duplex:
  - transmissão bi-direcional na mesma conexão
  - MSS: maximum segment size
- orientado à conexão:
  - handshaking (troca de mensagens de controle) inicia o estado do transmissor e do receptor antes da troca de dados
- controle de fluxo:
  - transmissor não esgota a capacidade do receptor

Sockets


socket
port
aplicação
envia dados
TCP
buffe de txr
segmento
aplicação
lê dados
TCP
buffer de rx
socket
port

Estrutura do Segmento TCP



Campos do Segmento TCP

- ***Source/Destination Ports*** (16 bits cada): Indica os números dos ports associados aos processos.
- ***Sequence Number*** (32): Permite reordenação dos segmentos e detecção de perda.
- ***Ack Number*** (32): Confirma recebimento dos segmentos com Sequence Number inferior ao indicado.
- ***Hlen*** (4): Tamanho do header TCP em unidades de 32 bits. Default = 5 => 20 bytes

Campos do Segmento TCP

Code (6): Bits que, quando ativos, informam a validade de alguns campos do header e indicam funções adicionais do segmento.

- ***URG***: Indica que o segmento carrega dados urgentes.
- ***ACK***: Indica validade do campo Acknowledgement.
- ***PSH***: Indica que os dados desse segmento e os presentes no buffe de recepção do end-pointer destino devem ser enviados imediatamente à aplicação associada.
- ***RST***: Indica que a conexão deve ser cancelada imediat.
- ***SYN***:Indica início da sequência de transmissão(Seq. nr.).
- ***FIN***: Indica que o end-pointer origem deseja encerrar a transmissão.

Campos do Segmento TCP

Window (16): Usado no controle de fluxo no sentido destino => fonte. Informa a disponibilidade de bytes no buffer de recepção do end-point local.

Urgent Pointer (16): Usado para envio de dados com prioridade acima do normal - dados assíncronos (caracteres de controle ou comandos de interrupção). Informa a posição, relativa ao início do campo de dados do segmento, em que encerra-se uma sequência de bytes a ser entregue com urgência pelo end-point remoto à aplicação correspondente.

Campos do Segmento TCP

Options (variável): Informações adicionais do segmento TCP. Ex.: Tamanho máximo do segmento a ser negociado na fase de estabelecimento.

Pad (var): Enchimento. Arredonda o tamanho do header para múltiplo de 4 bytes, quando do uso de ***Options.***

Data (var): Contém dados da Camada de Aplicação

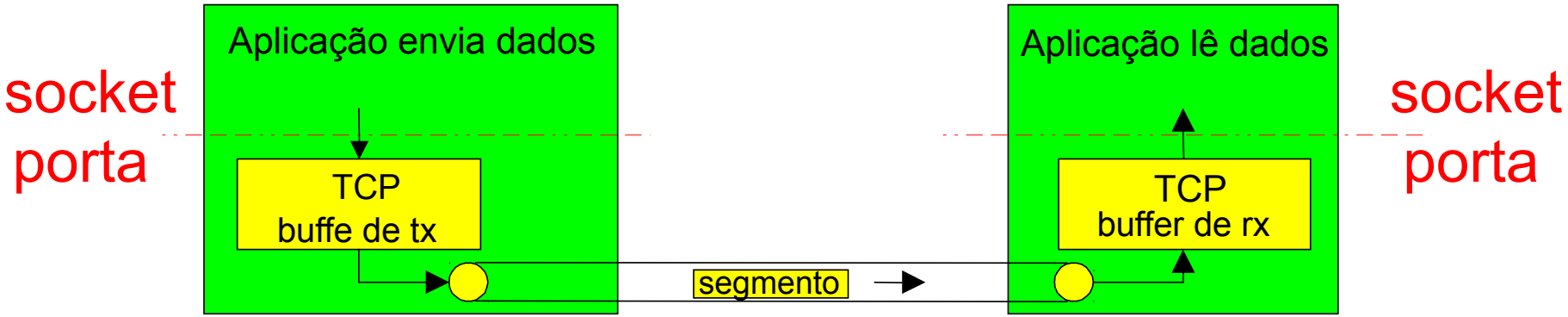
Protocolo TCP

- Ponto-a-ponto:
  - um transmissor, um receptor
- Confiável, seqüêncial 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

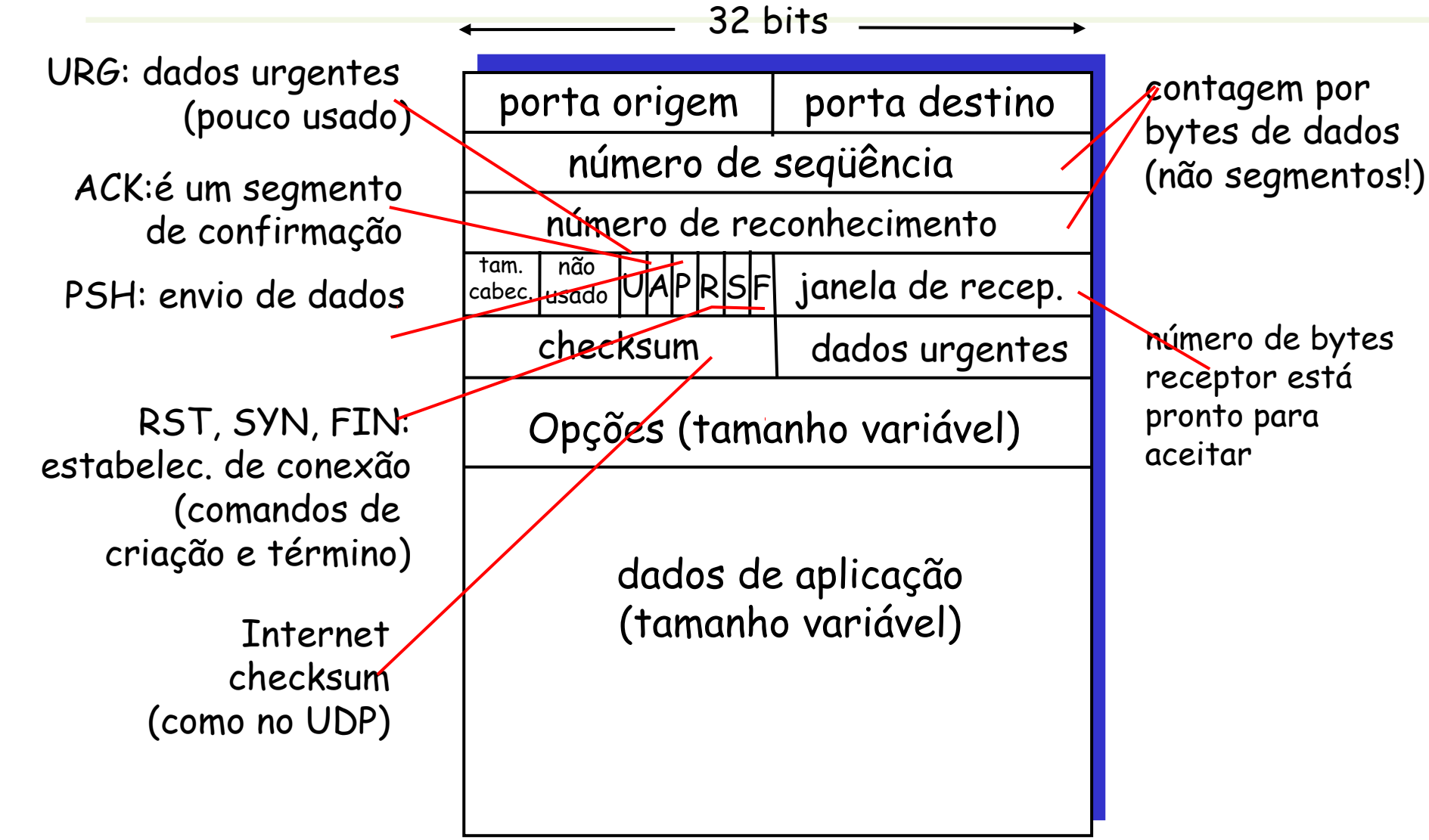
Protocolo TCP

- 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



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

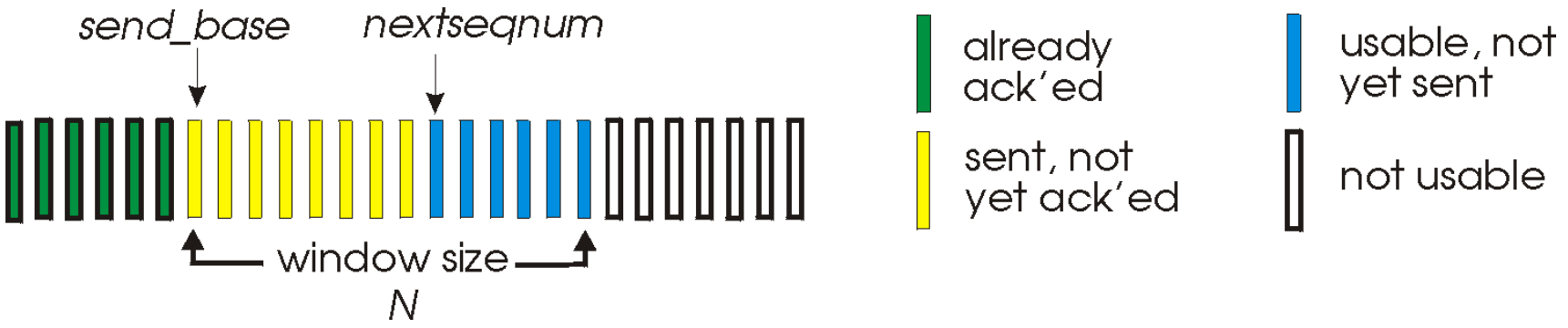
- Flags
  - URG – Há dados urgentes a serem entregues neste segmento
  - ACK – Este é um segmento de confirmação (acknowledge)
  - PSH – Push indica que este é um segmento que contém dados
  - RST – Reset a conexão precisa ser reiniciada (fechada imediatamente)
  - SYN – Estabelece uma nova conexão
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

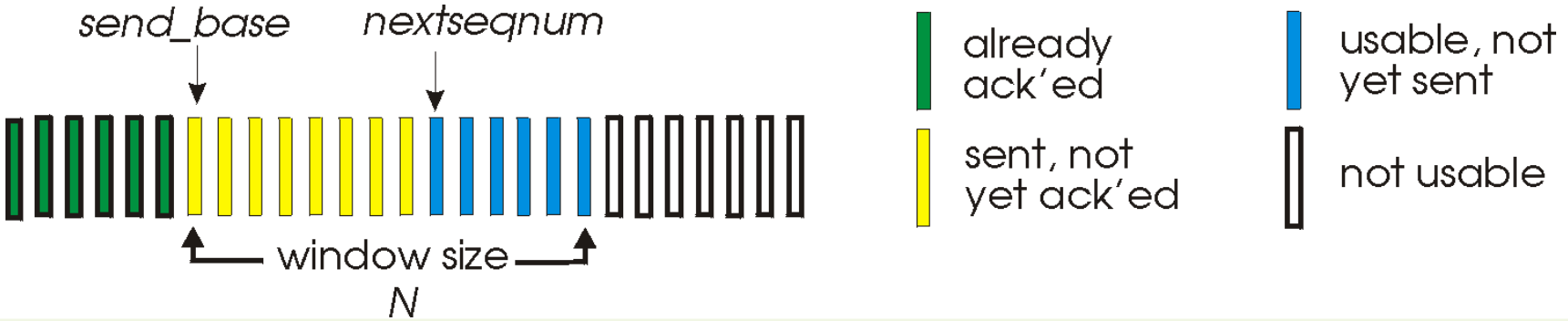
Janelamento

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

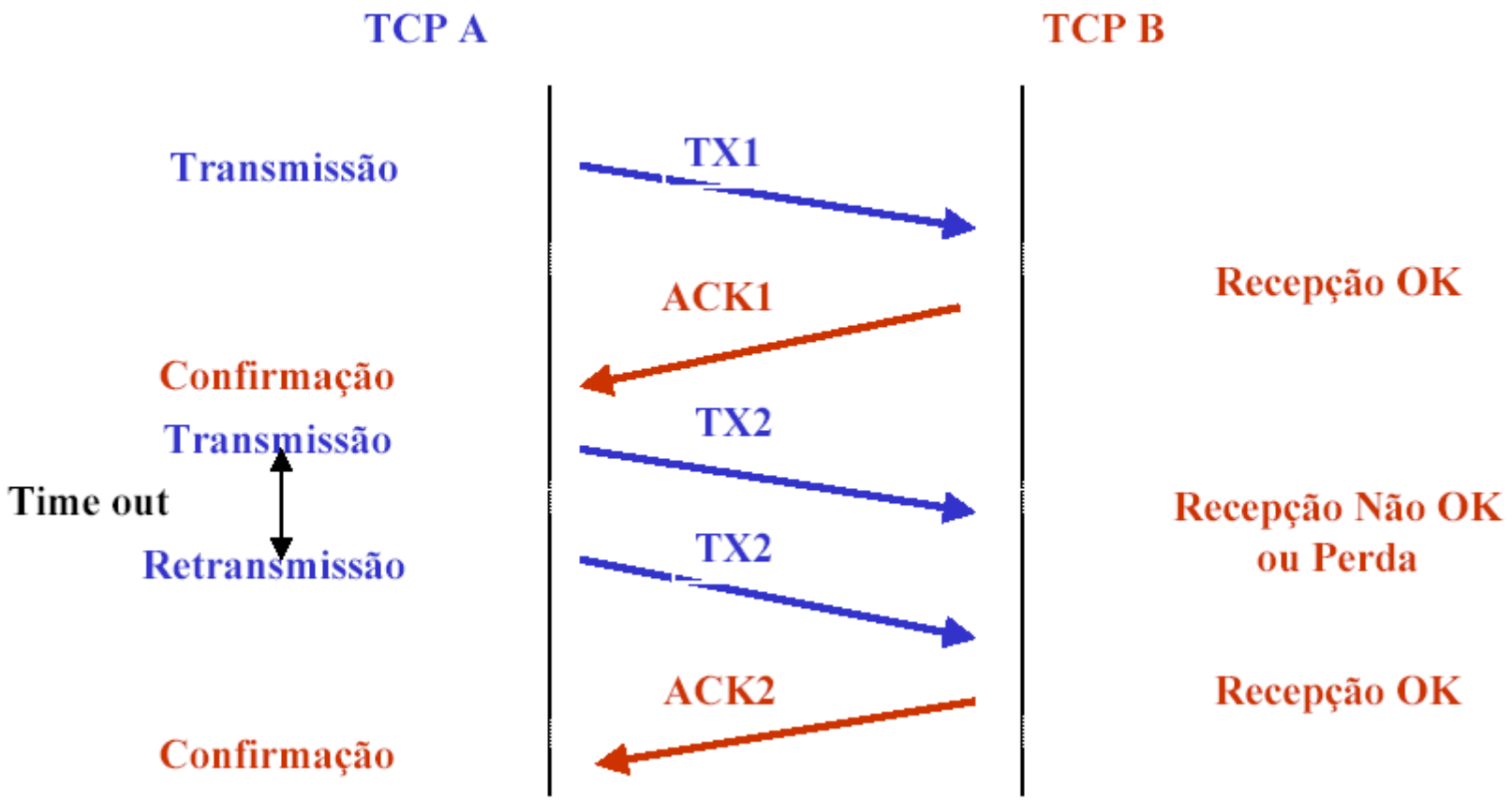


Ack

- 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

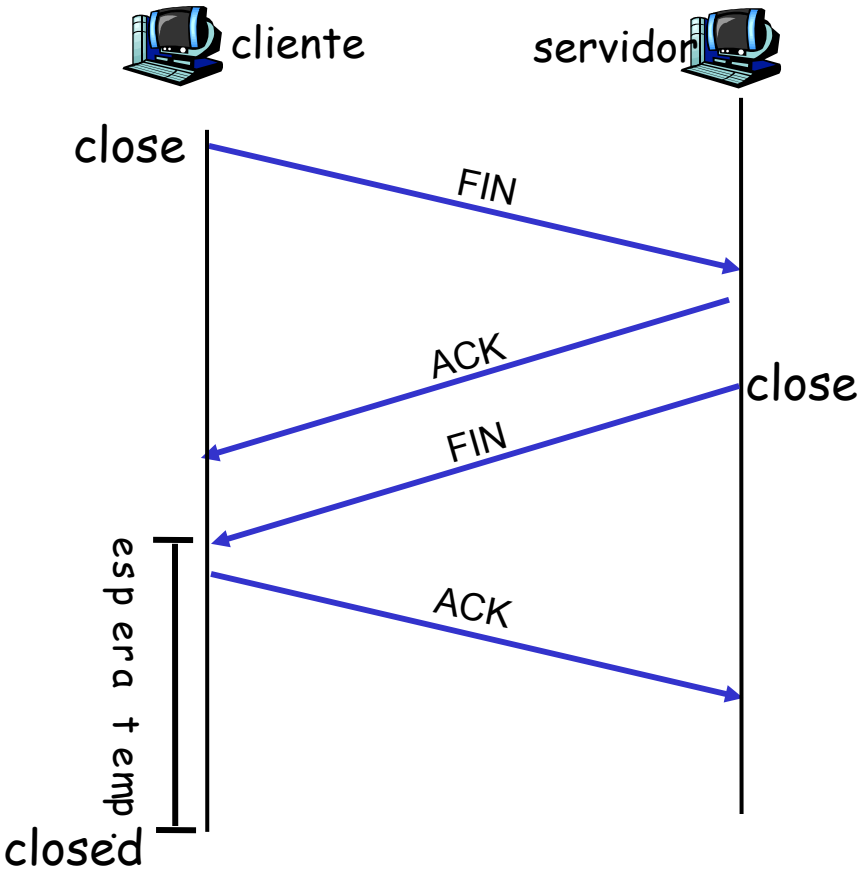


Retransmissão de segmentos



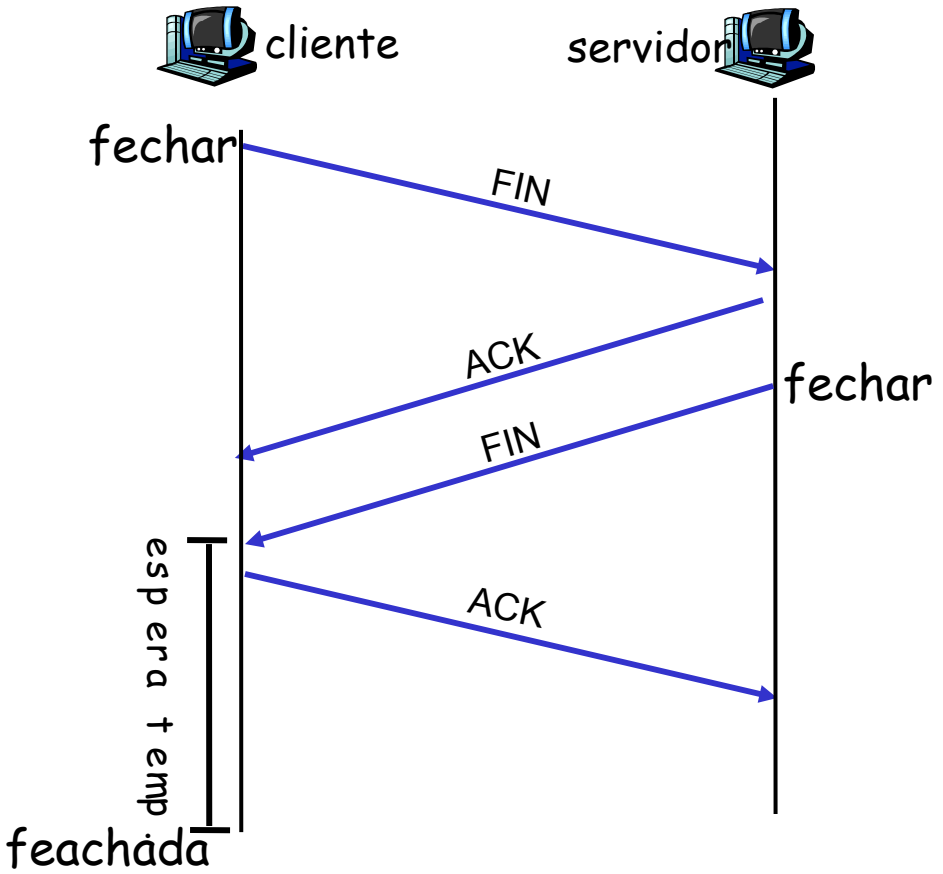
Fechando a conexão

- **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.



Protocolo UDP

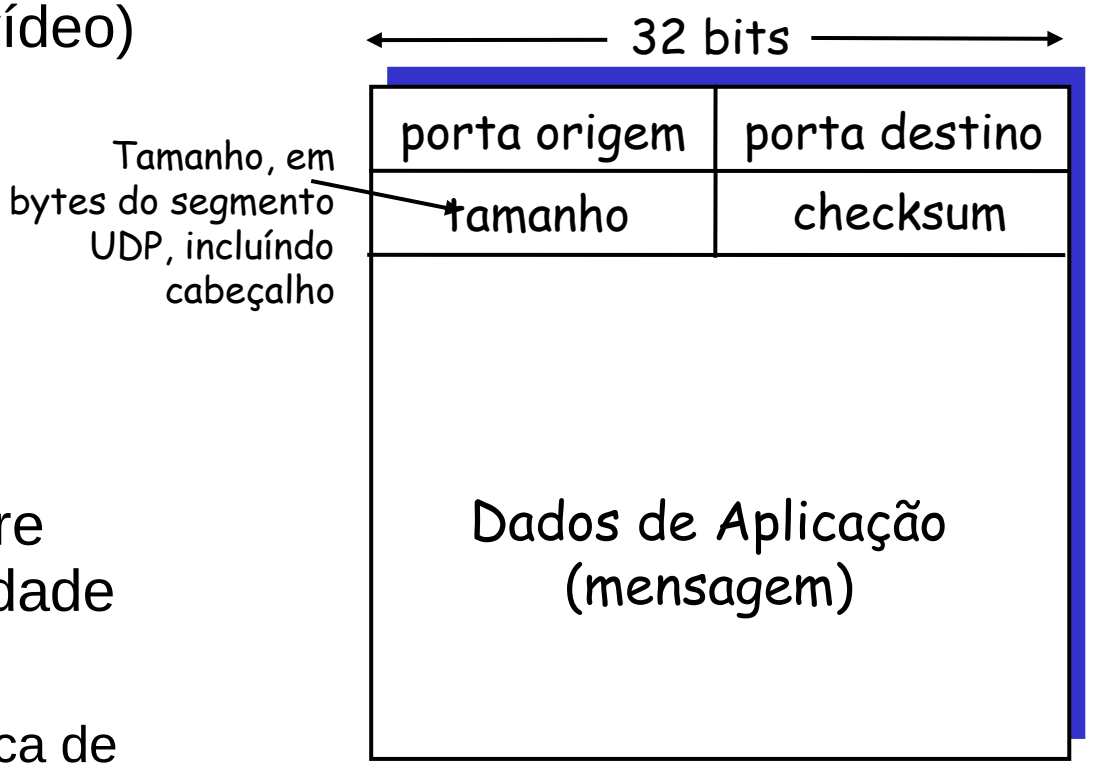
- Protocolo de transporte da Internet “sem gorduras”
- Serviço “best effort” , segmentos UDP podem ser:
  - perdidos
  - entregues fora de ordem para a aplicação
- Sem conexão:
  - não há apresentação entre o UDP transmissor e o receptor
  - cada segmento UDP é tratado de forma independente dos outros

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 mutimídia 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

Atividade

- Descreva as características do protocolo TCP.
- Em que situação é recomendado o uso do UDP
- Descreva a funcão do campo window
- Qual a função do campo número de sequência
- Descreva o handshake em 3 vias