

# Redes de computadores

---

Protocolo TCP

# Dilema dos generais

---

- Enviar um mensageiro e definir um tempo para que este volte com a confirmação
  - Quantas confirmações são necessárias?
- Definir um número de mensagens máximo a enviar antes do ataque.
  - Ainda correm o risco de atacar sozinhos

# O estabelecimento de conexão

---

- O protocolo TCP enfrenta o mesmo problema dos generais
- Como estabelecer a conexão de maneira confiável?
- Handshake de 3 vias
  - SYN
  - SYN ACK
  - ACK

# Transferência Confiável de Dados

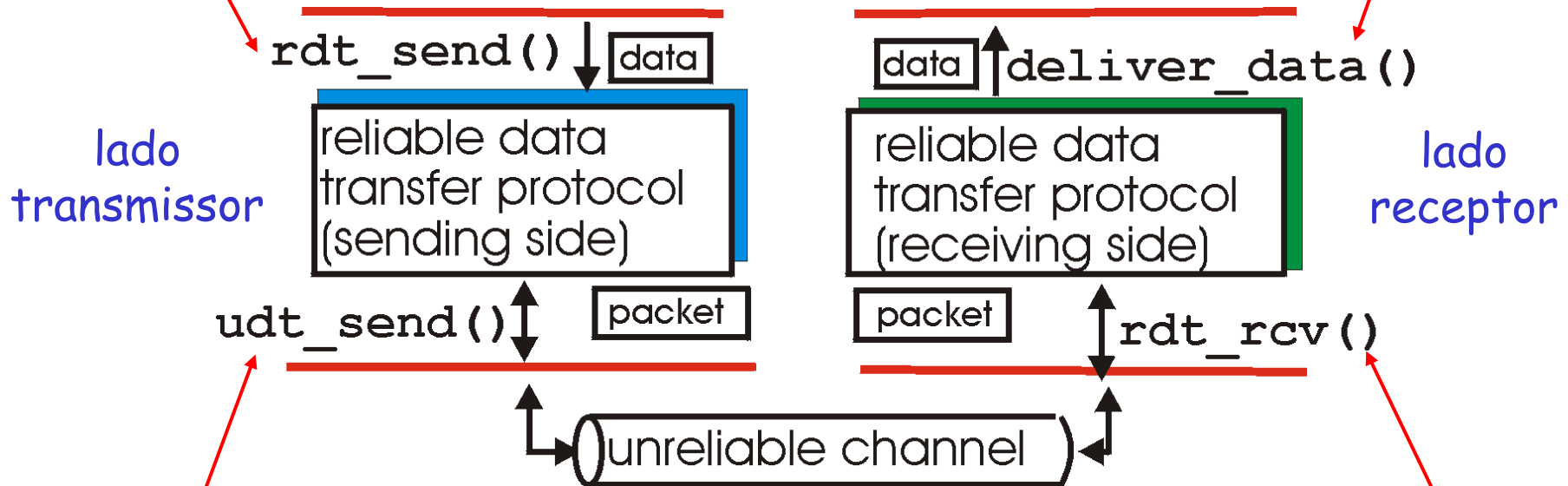
---

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

# Transferência confiável: o ponto de partida

`rdt_send()`: chamada da camada superior, (ex., pela aplicação). Passa dados para entregar à camada superior receptora

`deliver_data()`: chamada pela entidade de transporte para entregar dados para cima



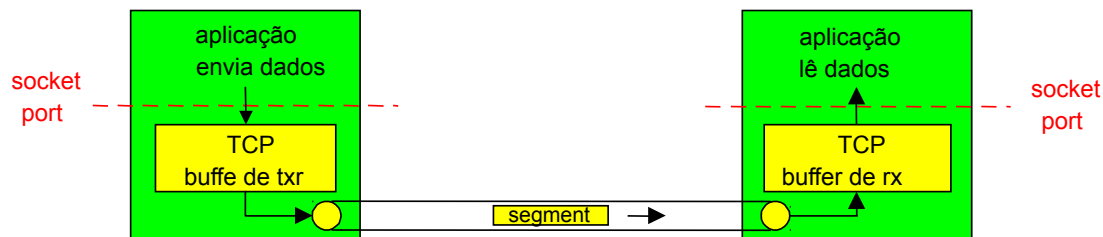
`udt_send()`: chamada pela entidade de transporte, para transferir pacotes para o receptor sobre o canal não confiável

`rdt_rcv()`: chamada quando o pacote chega ao lado receptor do canal

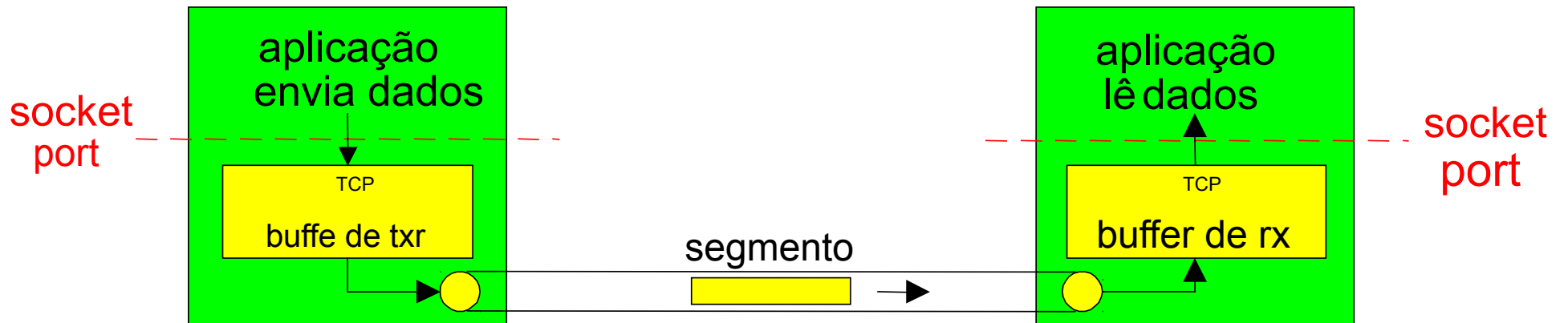
# TCP: Overview

RFCs: 793, 1122, 1323, 2018,  
2581

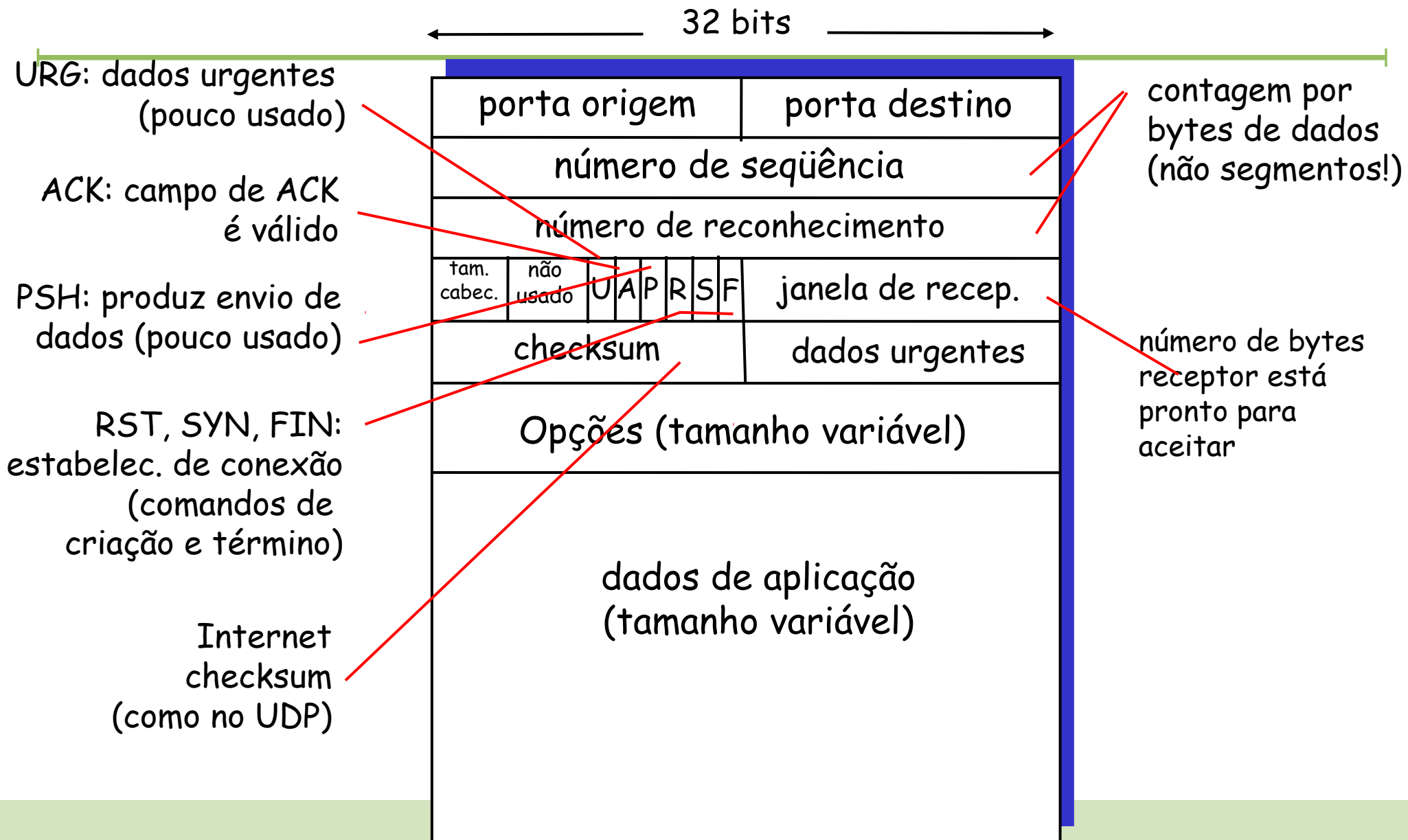
- **ponto-a-ponto:**
  - um transmissor, um receptor
- **confiável, seqüencial *byte stream*:**
  - não há contornos de mensagens
- **pipelined:** (transmissão de vários pacotes em confirmação)
  - Controle de congestão e de fluxo definem tamanho da janela
- **buffers de transmissão e de recepção**
-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



# 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 buffer 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