

Camada de Transporte



Roteiro



- Introdução / Visão geral
- Transporte de dados
 - Protocolo UDP
- Transporte confiável de dados
 - Protocolo TCP

Introdução / Visão Geral



- Objetivo: oferecer transporte de dados entre processos do usuário (camada de aplicação) residentes em *hosts* distintos
 - Extensão dos serviços da camada de rede
 - Comunicação fim-a-fim
- Modalidades de transporte:
 - Transporte não confiável
 - ✦ Projeto mais simples
 - ✦ Exemplo: Protocolo UDP
 - Transporte confiável
 - ✦ Mais complexo
 - ✦ Exemplo: Protocolo TCP

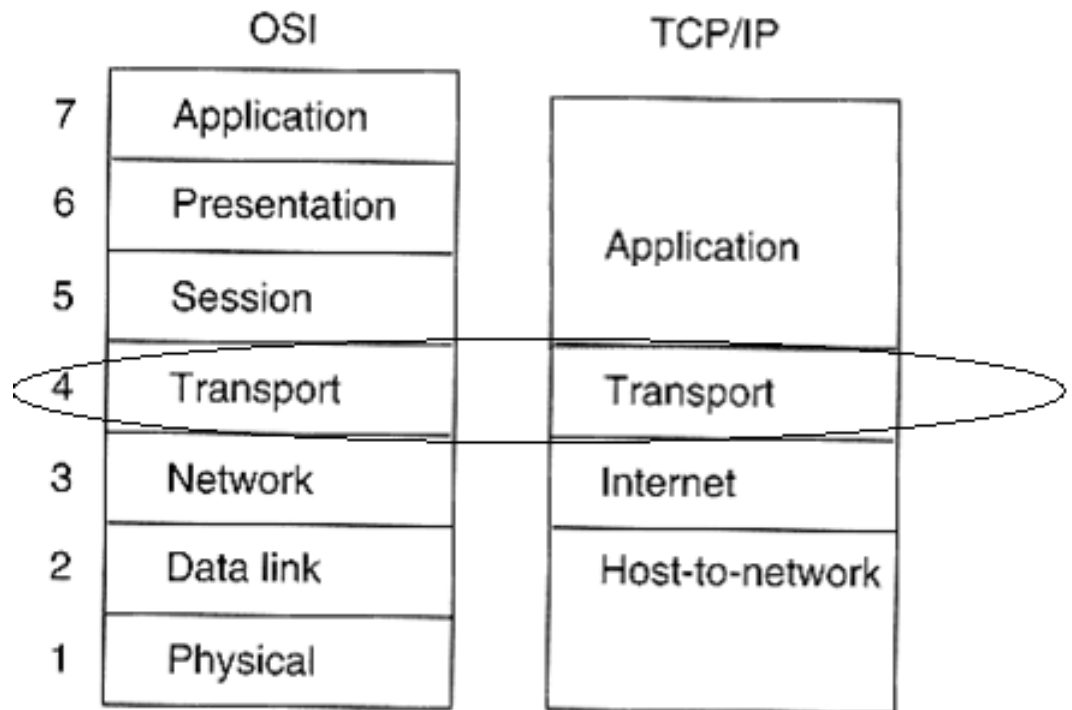
Camada de Transporte

Presente nos
Modelos

OSI/ISO

e

TCP/IP



Camada de Transporte



1.COMUNICAÇÃO LÓGICA ENTRE PROCESSOS DE APLICAÇÃO

Foco: Transporte de dados

- **Multiplexação/demultiplexação**
- **Endereçamento**
- **Entidades-pares**
- **Protocolo UDP**
- **Protocolo TCP**

Por que camada de transporte?

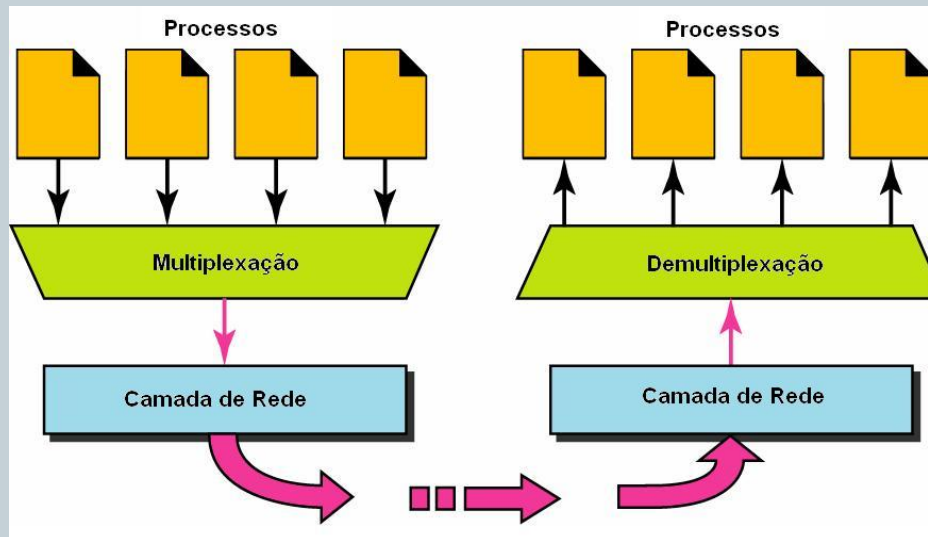


- A camada de rede resolveria?
 - Contexto:
 - ✦ Oferece um serviço de comunicação entre hosts (host-to-host)
 - Algoritmos que interagem com roteadores vizinhos para localizar a melhor rota para o destino
 - Como prover paralelismo no host?
 - ✦ Para tratar um processo, tudo bem... mas e para **N** processos usuários?
- Um serviço de transporte – intermediário entre as camadas de aplicação e de rede – é mais lógico e prudente!
 - Nesse caso, pode-se ver transporte como um mecanismo para IPC (*Interprocess Communication*).

Multiplexação / demultiplexação



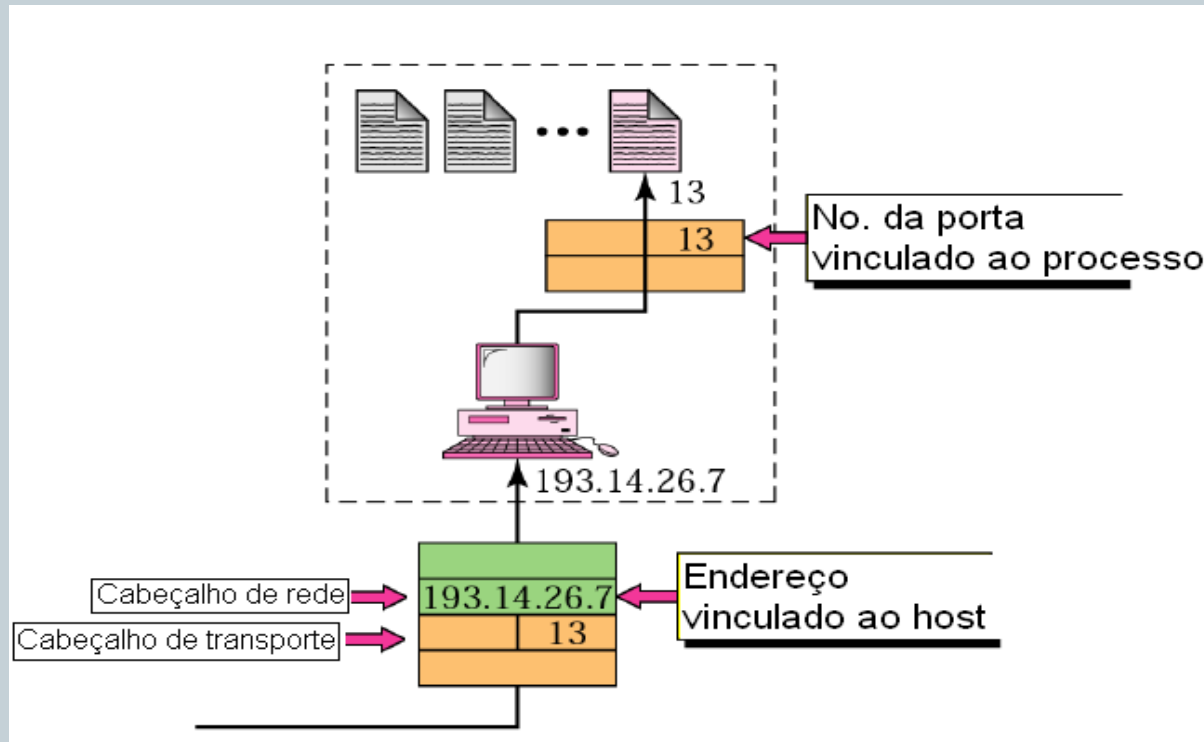
- Multiplexação
 - O protocolo de transporte aceita mensagens de vários processos
- Demultiplexação
 - Transporte entrega mensagens para vários processos



Endereçamento



- Processos (aplicações do usuário) são vinculados a endereços de transporte (portas)



Endereçamento

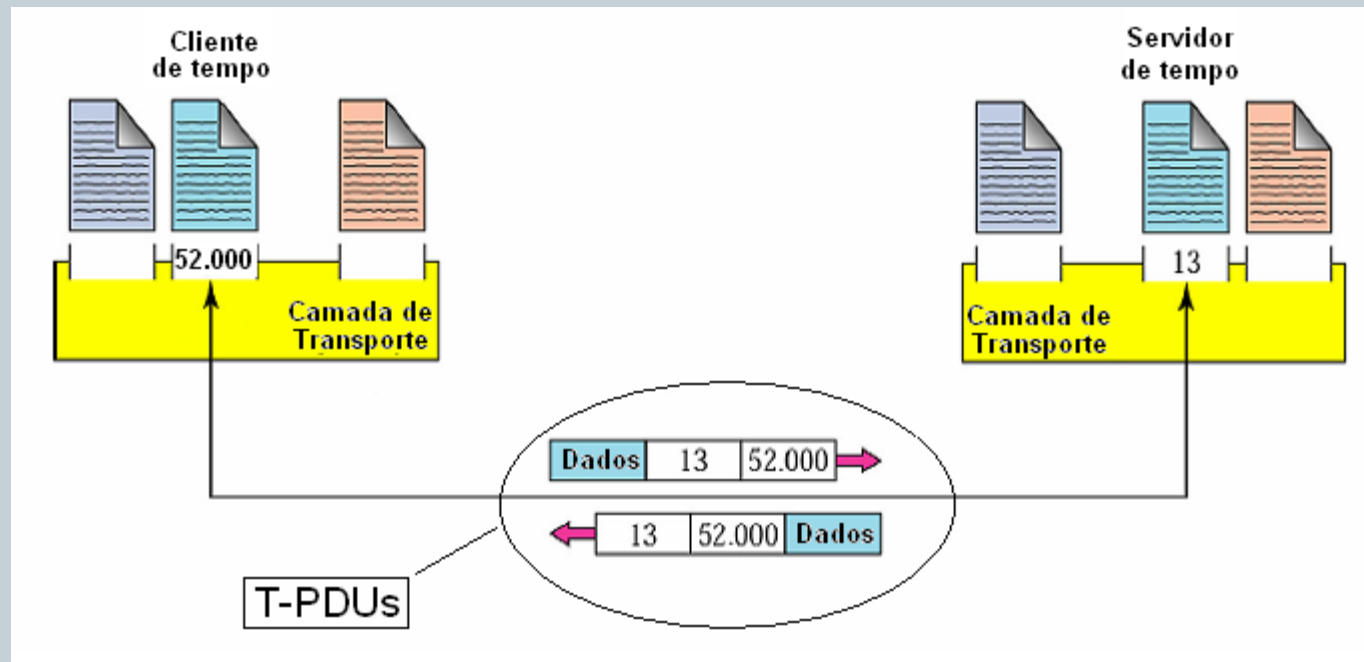


- Portas são números inteiros de 16 bits

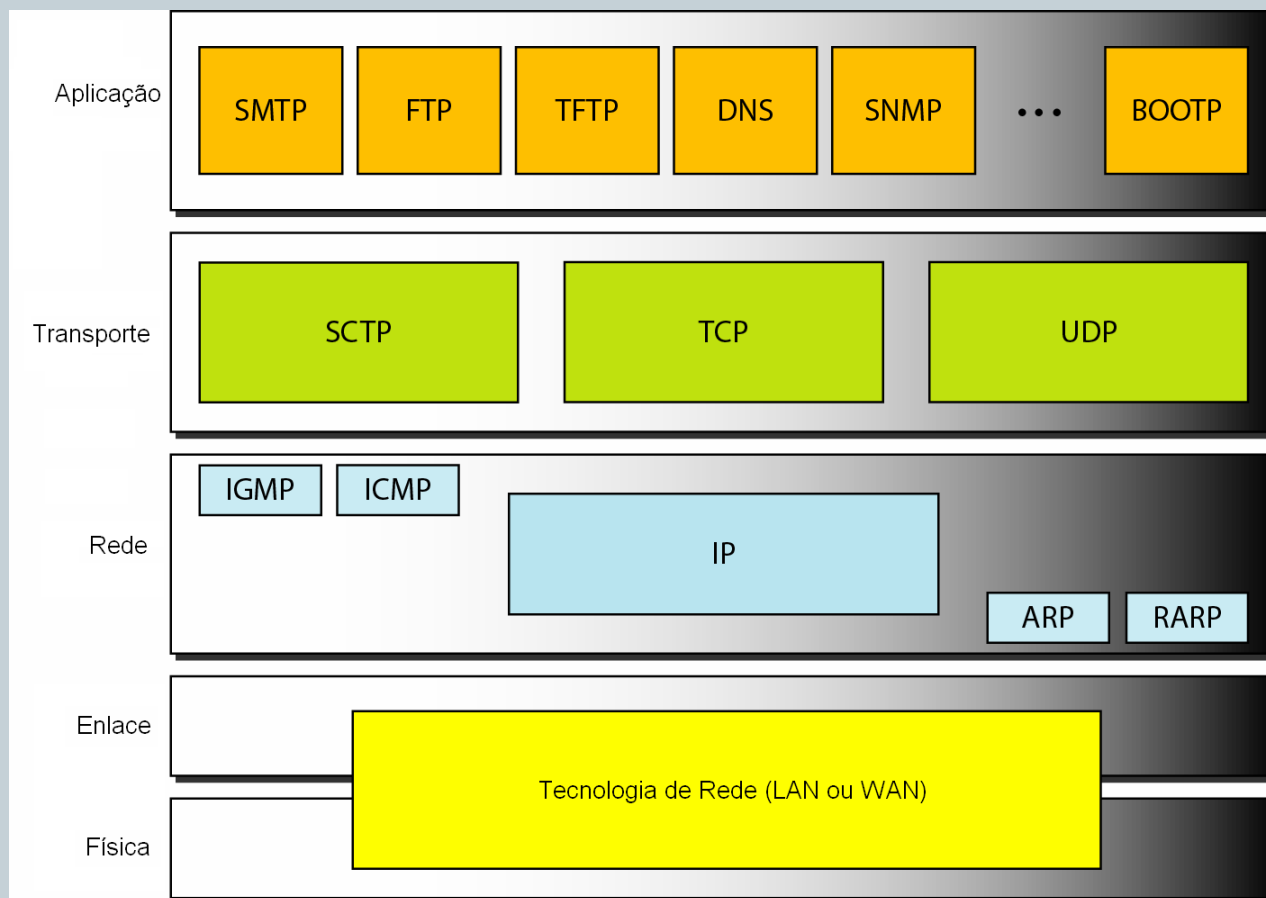


Entidades-pares de transporte

- Paradigma cliente-servidor
 - Processo no host local (cliente) solicita serviços a outro processo, normalmente localizado no host remoto (servidor)



Protocolos de Transporte



Protocollo UDP

User Datagram Protocol

Protocolo UDP

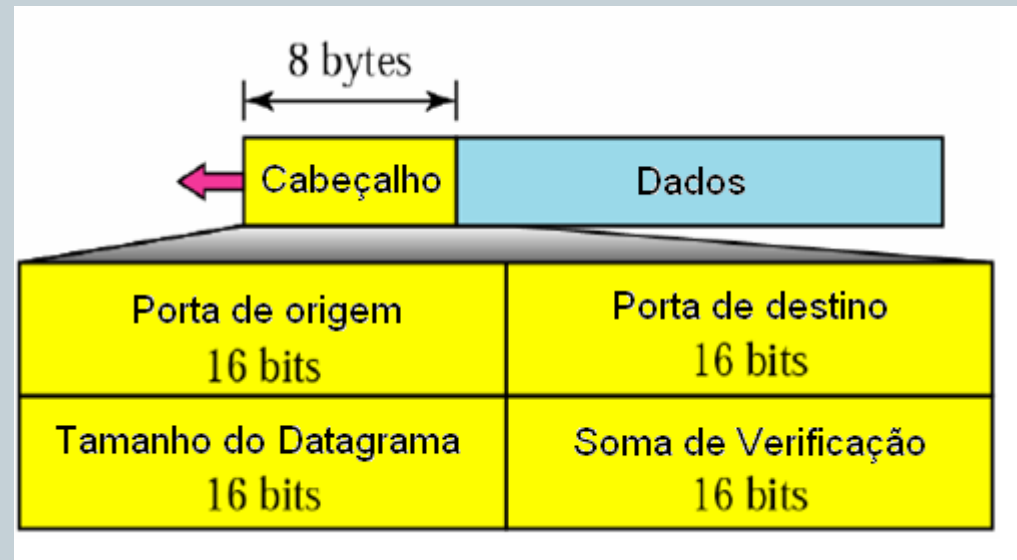


- Datagramas de tamanho variável
- Não há estabelecimento de conexão
- Pequeno *overhead* no cabeçalho do pacote
 - Favorece aplicações multimídia
- Adequado para processos que transmitem mensagens curtas
- Taxa de envio não regulada
 - Não tem controle de fluxo
 - Receptor pode ser inundado se chegar um número excessivo de mensagens

Cabeçalho de datagramas UDP



- Portas de origem e destino
- *Total length* habilita datagramas UDP de até 64Kbytes
- Checksum (16 bits)
 - Detecção de erros
 - **Erro => datagrama descartado**



Quem usaria o UDP como camada de transporte?



- Qualquer aplicação que queira mais controle sobre o fluxo de transmissão de dados
- Aplicações que suportem perdas de dados pequenas
- Alguns exemplos:
 - Aplicações multimidia
 - VoIP
 - Gerenciamento de rede (SNMP)
 - Protocolo de roteamento RIP
 - Serviço de nomes DNS

Protocollo TCP

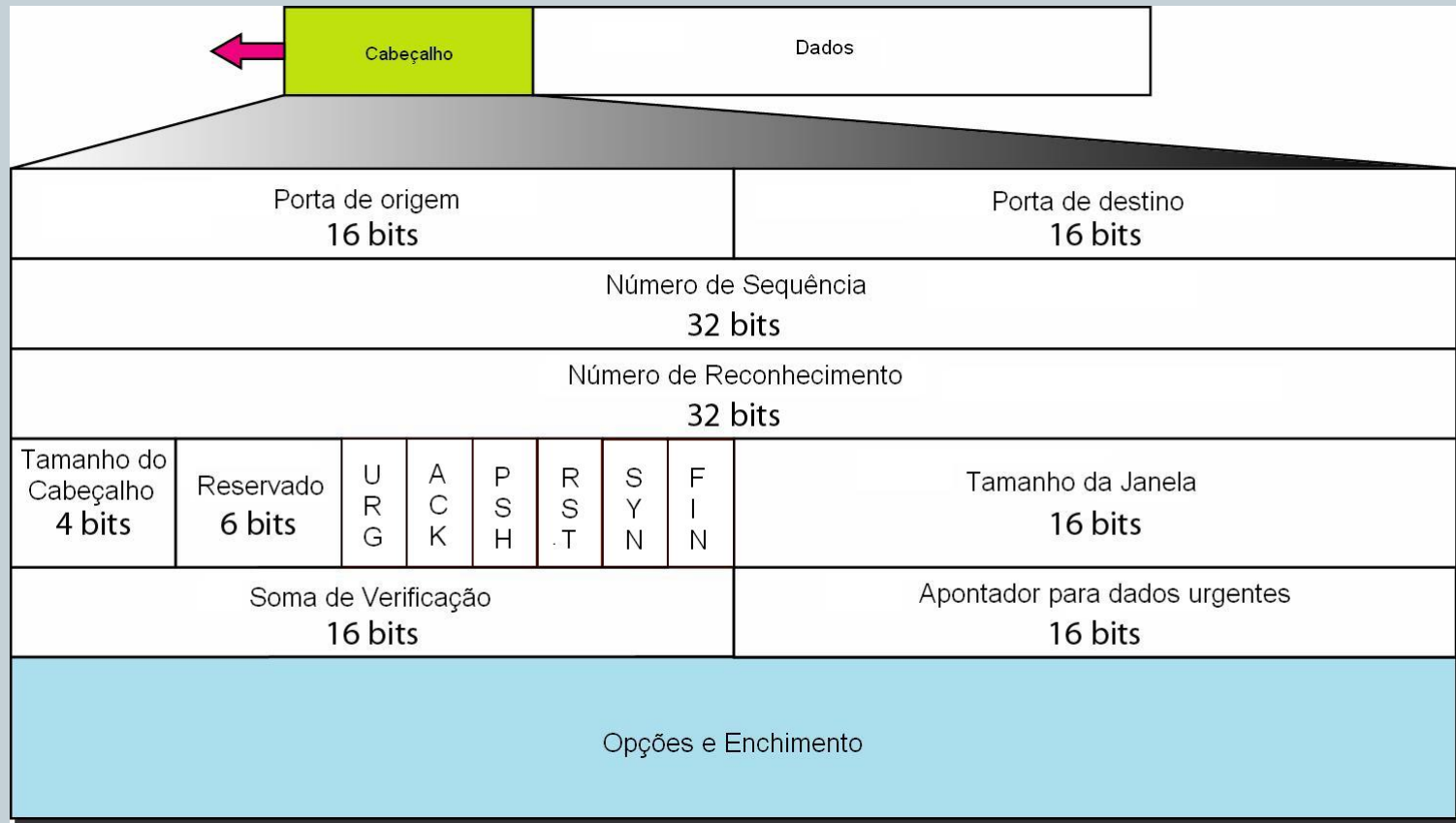
Transmission Control Protocol

Características do TCP

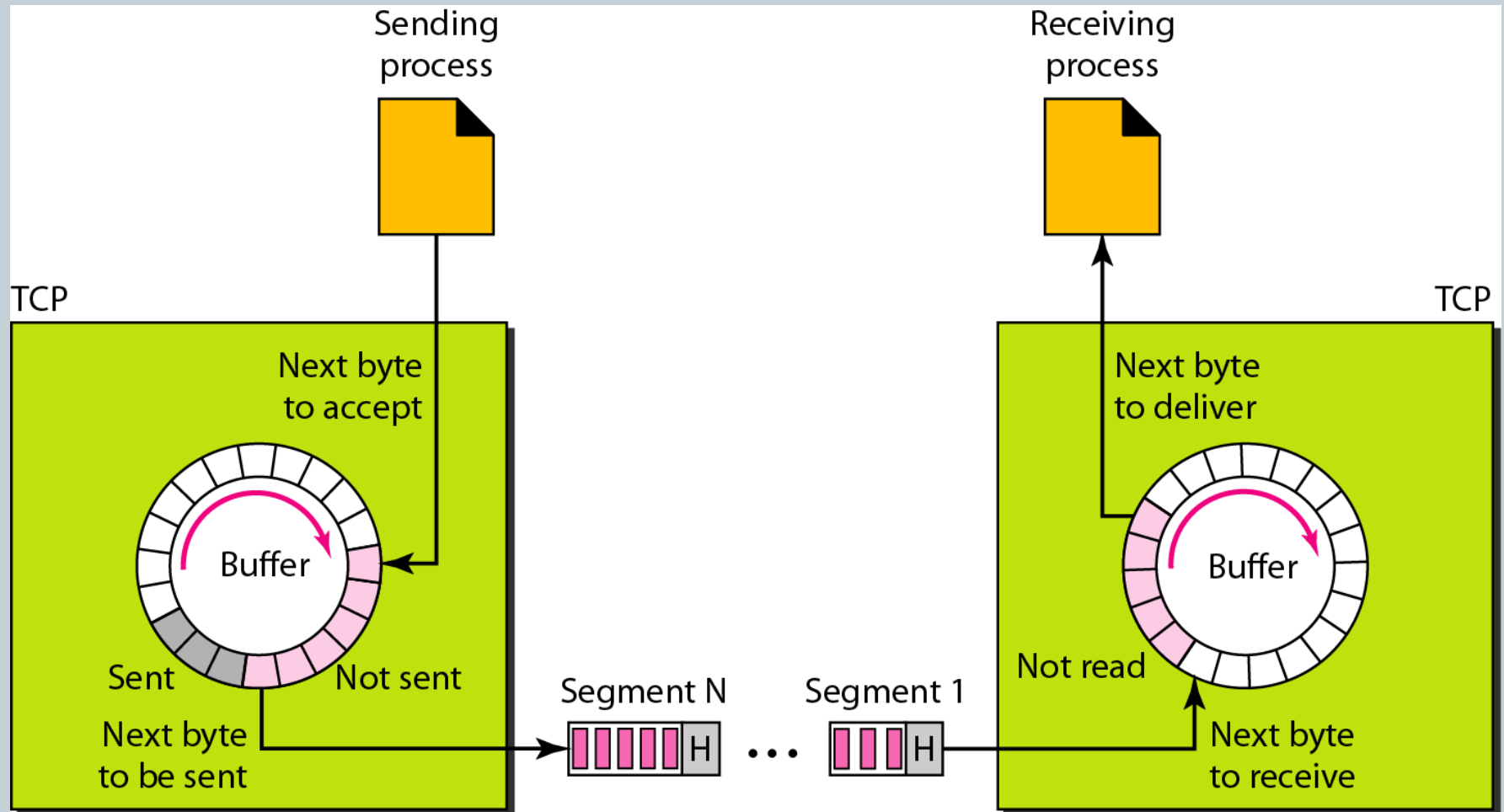


- Conexão fim-a-fim de um para um
 - Não faz multicast!
- Unidade de informação: segmento
 - Os dados do usuário são divididos em segmentos, de acordo com o MSS (Maximum Segment Size)
- Orientado a conexão
 - Porta de origem, porta de destino

Cabeçalho TCP



Segmentos TCP



Camada de Transporte



2.COMUNICAÇÃO LÓGICA ENTRE PROCESSOS DE APLICAÇÃO

Foco: Transporte CONFIÁVEL de dados

- **Estabelecimento de conexão**
- **Sequenciamento**
- **Controle de Fluxo**
- **Controle de Erros**
- **Encerramento de conexão**

O que é transporte confiável de dados?



- Dados chegando em ordem e sem erros no receptor
 - Como garantir isso?
- Exige:
 - Sequenciamento dos segmentos
 - ✦ Número de sequência e de reconhecimento
 - Estabelecimento e liberação de conexão
 - Controle de fluxo
 - Controle de erros
 - ✦ Retransmissão de segmentos

Sequenciamento de segmentos

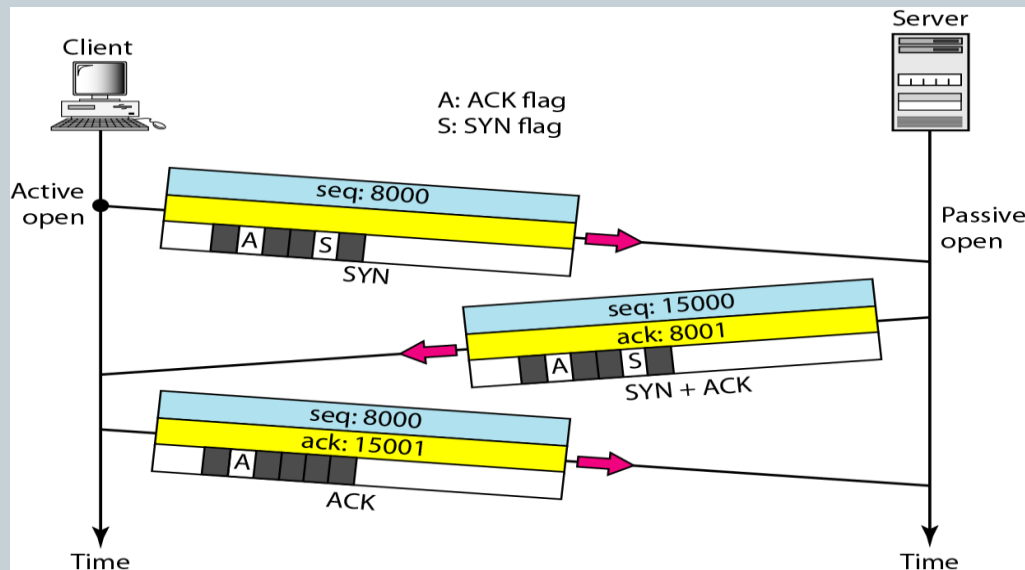


- Bytes transmitidos em cada conexão são numerados pelo TCP
 - Número inicial é escolhido aleatoriamente
 - O campo de sequência no cabeçalho define o número do primeiro byte contido no segmento
- O campo de reconhecimento indica o número do próximo byte esperado pelo receptor
 - Este número é cumulativo

Estabelecimento de conexão



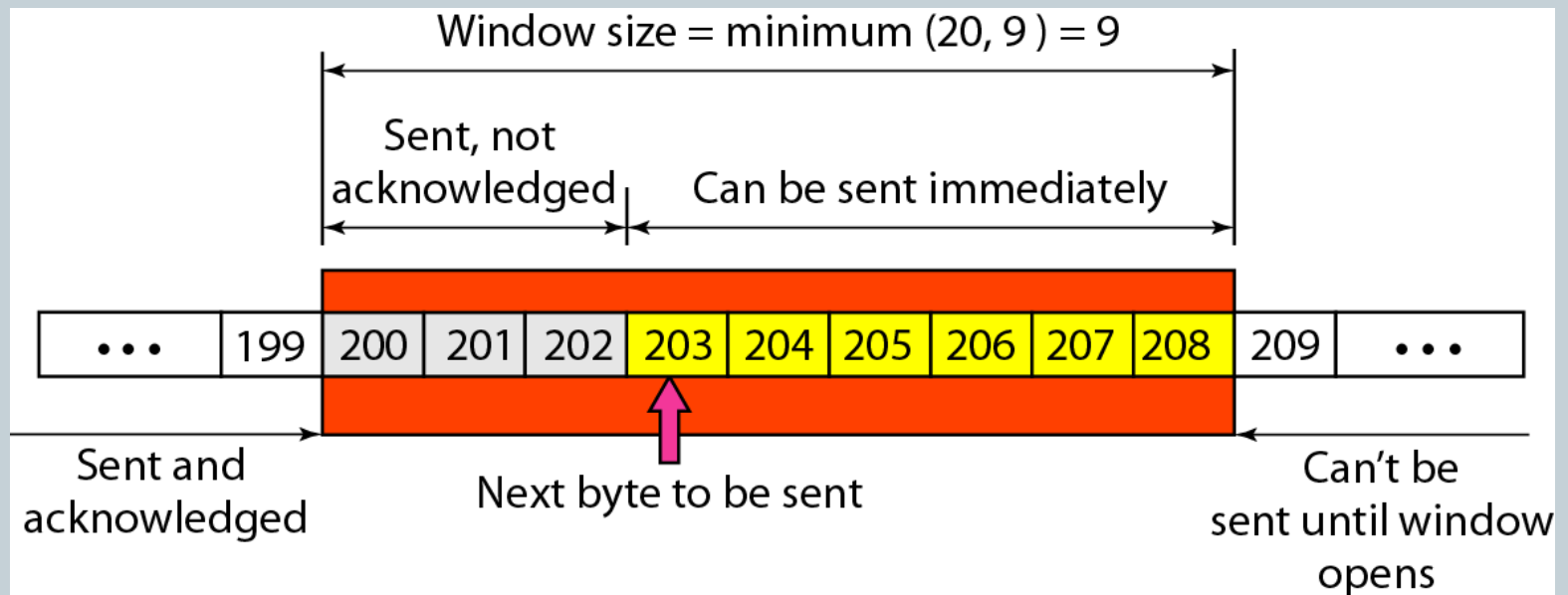
- Útil para transmissão confiável de dados
 - Acordos entre as partes sobre o tipo de serviço prestado
 - Define valores iniciais para as variáveis de controle que serão utilizadas na fase de transferência de dados



Controle de Fluxo



- Uso de Janela Deslizante
 - ✦ Receptor indica quantos bytes pode receber no máximo



Controle de Fluxo



- TCP possui funcionalidades similares ao Go-Back-N e à Repetição Seletiva
 - O emissor pode ter bytes transmitidos mas ainda não reconhecidos na janela
 - O receptor recebe segmentos ordenados
 - ✦ **Ao receber, devolve um ACK para o emissor e entrega os bytes para a camada de aplicação**
 - O receptor envia apenas reconhecimentos cumulativos
 - ✦ **Não reconhece segmentos que chegarem fora de ordem**
 - ✦ **Armazena segmentos fora de ordem em buffers de recepção (não os descarta)**

Controle de Erros

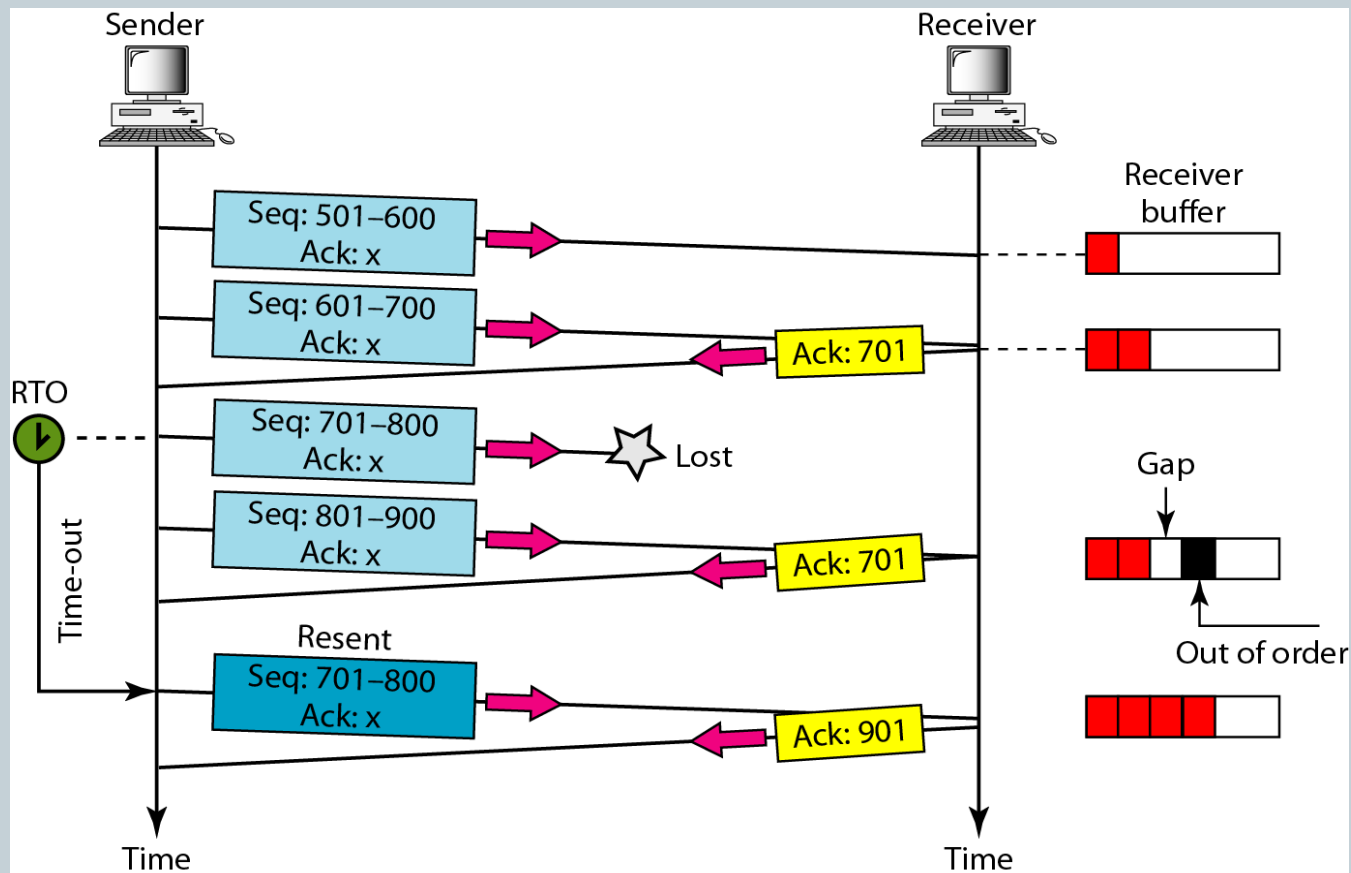


- Checksum
- Confirmação
- Retransmissão
 - ✦ Temporização (RTO – Retransmission Time-Out)
 - ✦ Após três segmentos com ACK duplicados (retransmissão rápida)

Controle de Erros



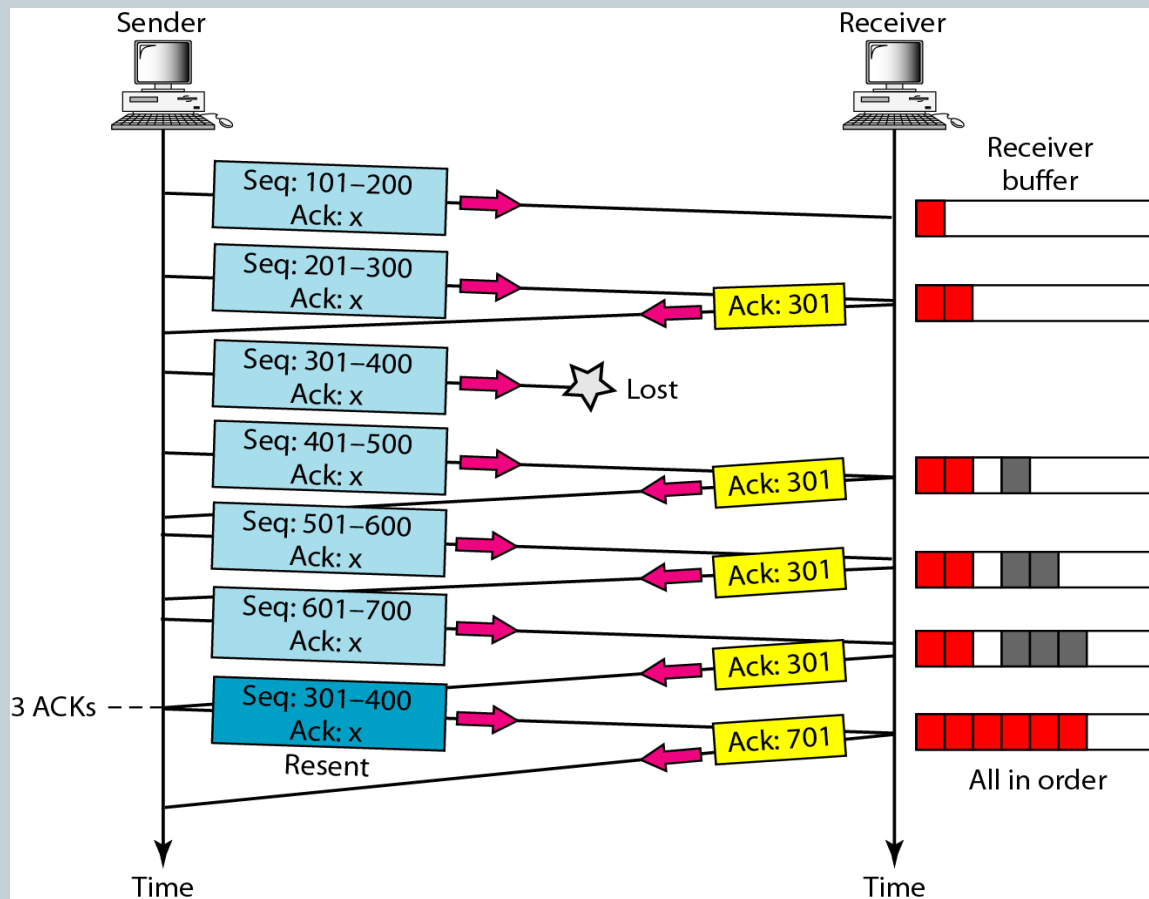
- RTO



Controle de Erros



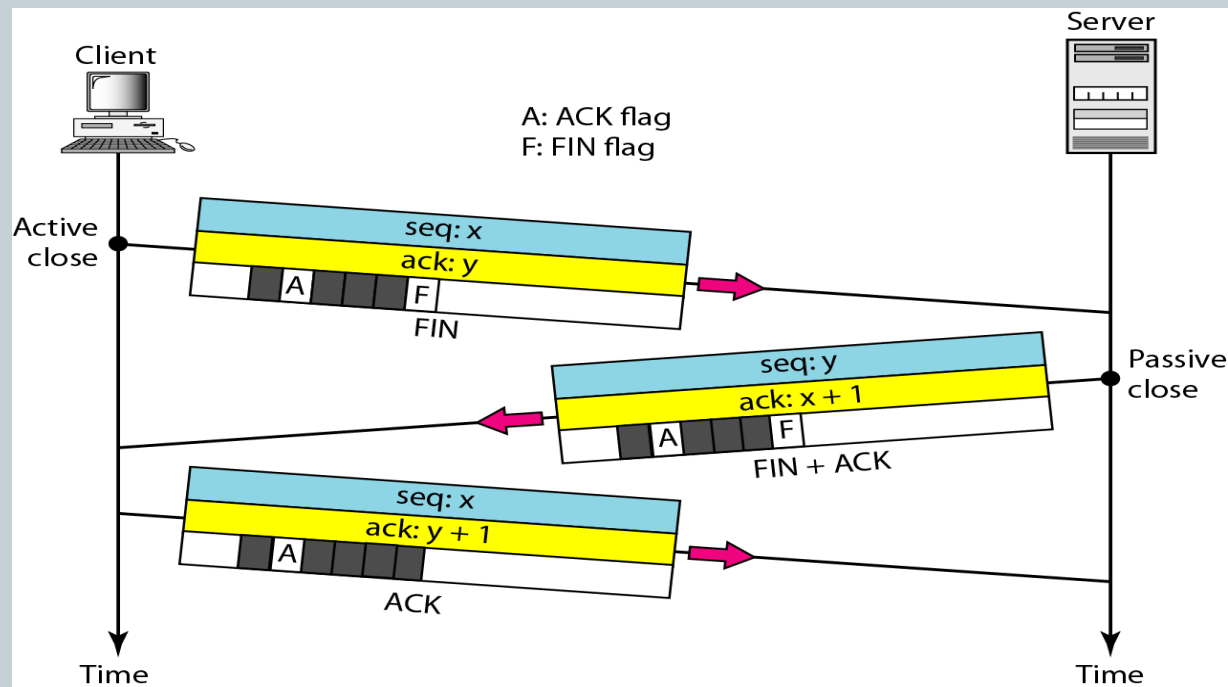
- Retransmissão rápida



Encerramento de conexão



- Finaliza a transmissão confiável de dados
 - Encerra os acordos feitos entre as partes sobre o tipo de serviço prestado



Referência Bibliográfica



- Fourozan, B. A. *Comunicação de Dados e Redes de Computadores*. 4. edição. São Paulo: McGraw-Hill, 2008.