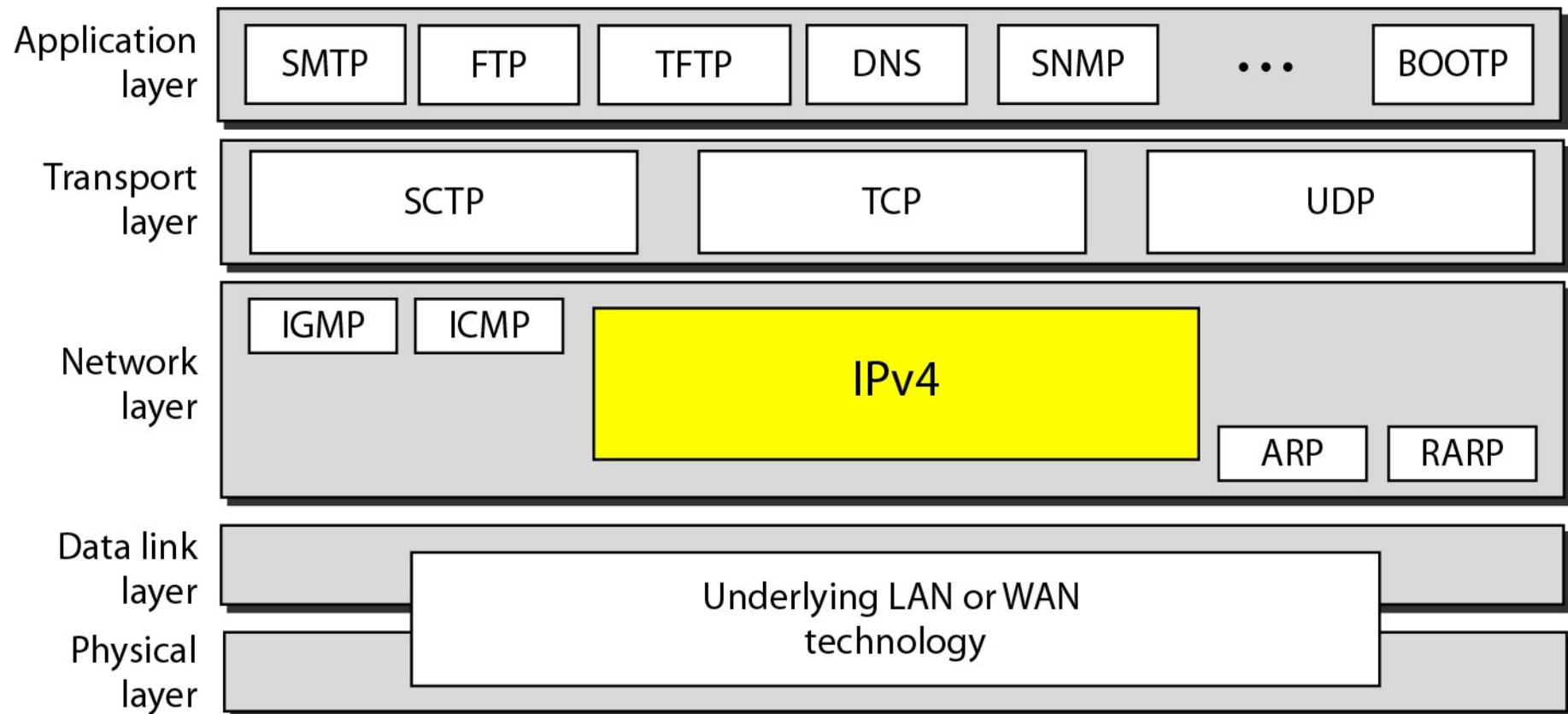


# Camada de Transporte



# Transport Control Protocol (TCP)

---

- ❑ Um protocolo extremamente confiável projetado para operar sobre redes do tipo *ConnectionLess* (sem conexão).
- ❑ Oferece suas funcionalidades através de serviços orientados a conexão
- ❑ Garante que os dados serão entregues confiavelmente, em sequência e sem confusão ou erro

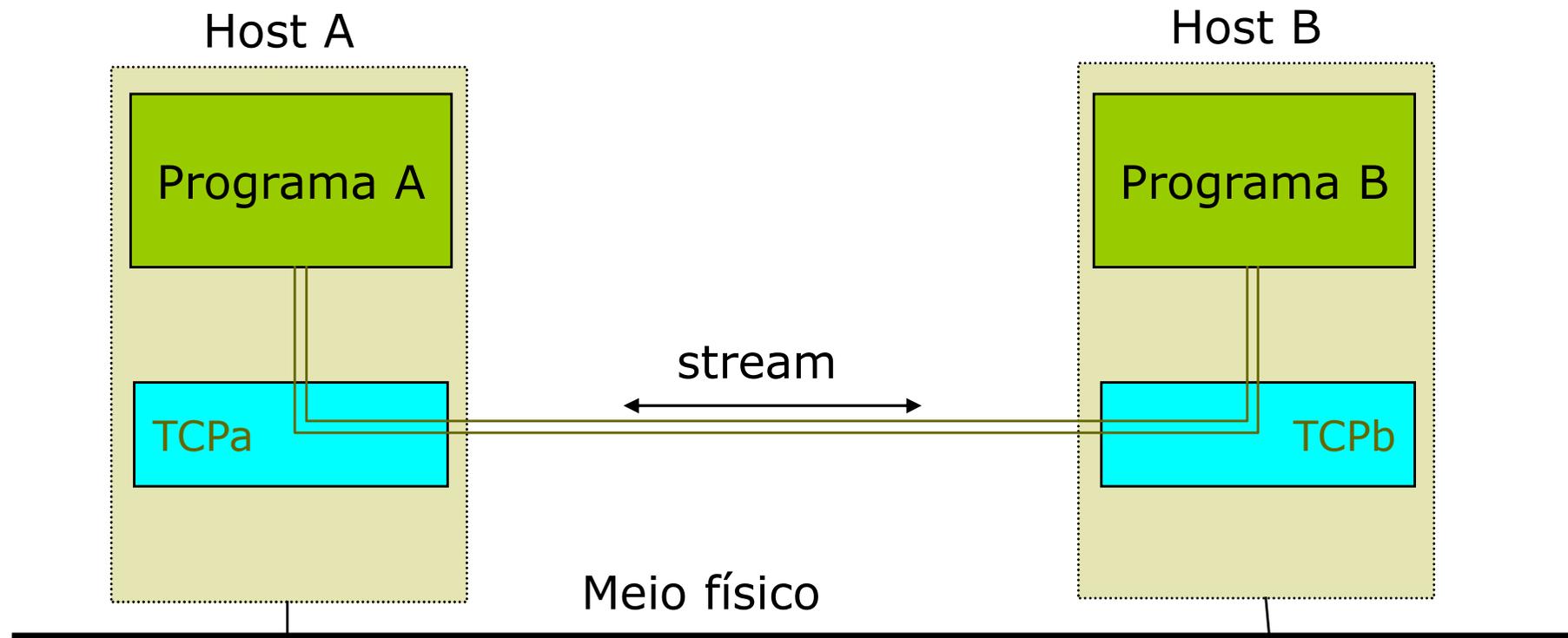
# Streams

---

## □ Conceitos básicos - Streams

- *Outgoing and Incoming Data streams*: durante uma conexão TCP, uma aplicação envia uma sequência de dados para uma aplicação parceira. Ao mesmo tempo, ela pode receber uma sequência vinda da aplicação parceira
- TCP oferece um serviço *full-duplex*, que simultaneamente gerencia duas *streams*.

# Representação de Streams



# Primitiva da Camada Transporte

---

## □ Segmento

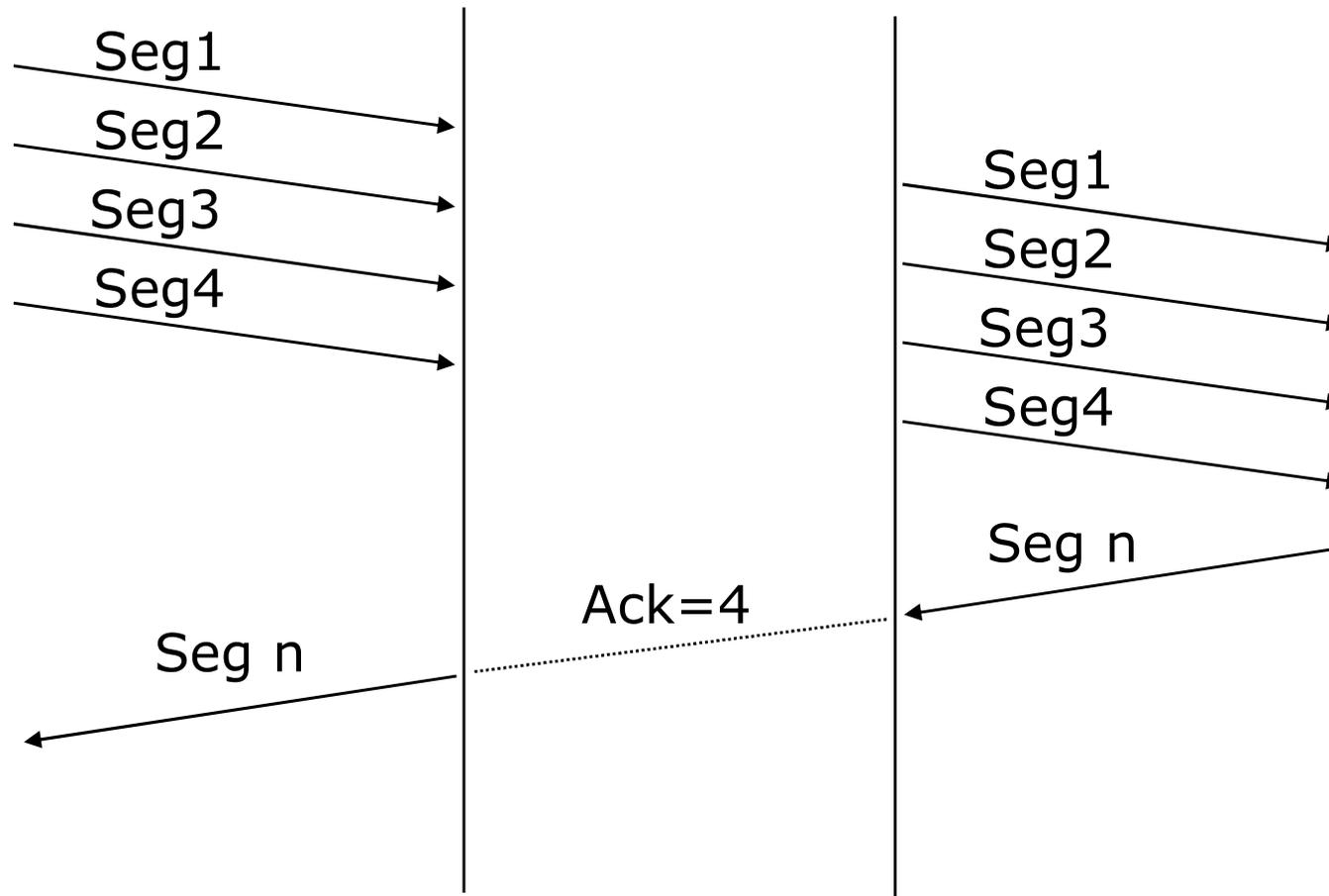
- Uma primitiva do protocolo TCP deve ser convertida em uma sequência de dados de tal modo que ela seja enviada em datagramas
- A troca de serviços entre o TCP e o IP é feita através de segmentos (TPDU do TCP)
- O tamanho máximo de um segmento é configurado pela variável MSS (*maximum segment size*)

# Confiabilidade do TCP

---

- ❑ Os segmentos do protocolo TCP serão enviados através de TCB (*transport control block*)
- ❑ Embora sejam enviados TCBS, o reconhecimento dos dados transmitidos sobre a stream é feito para cada Byte
- ❑ O TCP utiliza o mecanismo de reconhecimento em *Piggybacking*

# Piggybacking



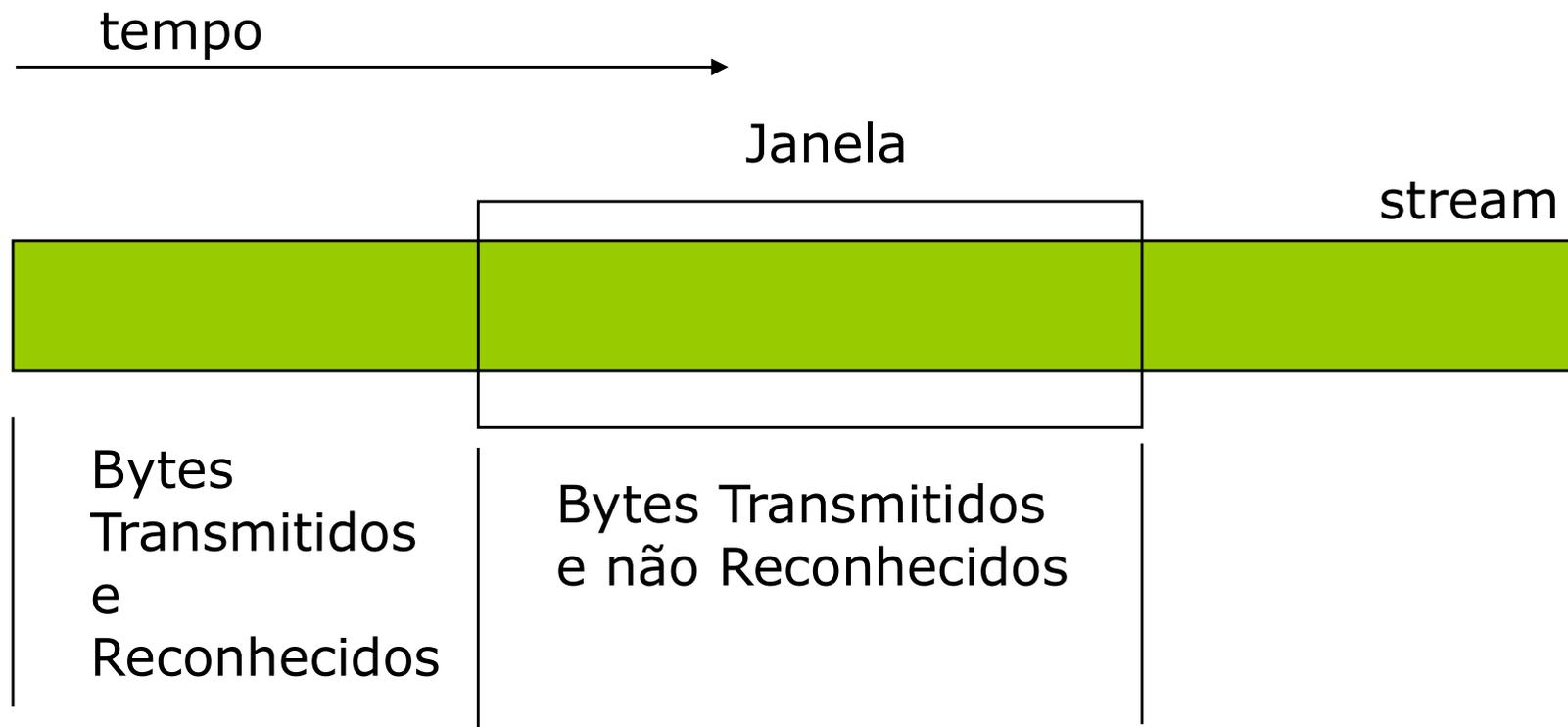
# *Sliding Window*

---

- ❑ Visando desempenho, o TCP não pode reconhecer a todo instante bytes que chegam (além de haver necessidade de ter dados para transmitir)
- ❑ Para tanto ele armazena as primitivas em uma janela, e enquanto houver “espaço” o TCP pode transmitir sem ter a preocupação de reconhecer

# *Sliding Window*

---



# Considerações

---

- ❑ Janela é um dos grandes problemas em aplicações desbalanceadas tais como transferência de arquivos
- ❑ Normalmente o cliente envia uma primitiva de requisição e pode receber vários segmentos como resposta
- ❑ A Janela deve ser ajustada para atender à necessidade do sistema

# Considerações (1)

---

- “*Silly Window Syndrome*” culmina em sério problema de desempenho quando as aplicações parceiras (enviadora e receptora) trabalham em velocidades diferentes

# Protocolo TCP

---

Source Port				Destination Port				
Sequence Number								
Acknowledgment Number								
Data Off	Reserved	U	A	P	R	S	F	Window
Checksum				Urgent Pointer				
Options & Padding								
Data								

# Protocolo TCP

---

- ❑ *Source/Destination Port*: são pontos nos quais os programas da aplicação podem acessar e ser acessados pela rede.
- ❑ *Sequence number*: o número de sequência associado às streams de bytes de tal modo que eles sejam reconhecidos assincronamente

# Protocolo TCP

---

- ❑ *Acknowledge number*: reconhece o último byte recebido pelo parceiro. Este campo somente tem sentido quando a *Flag 'A'* está com seu valor igual a um.
- ❑ *Data Offset*: indica o comprimento do cabeçalho (incluindo as opções e for o caso) em palavras de 32 bits.
- ❑ *Reserved*: Campo reservado com valor zero

# Protocolo TCP

---

## □ Flags

- Urgent (1): indica existência de dados urgentes (*Urgent Pointer* deve ser considerado)
- Ack (1): indica presença de reconhecimento
- Push (1): indica presença de '*data to be pushed*'
- Restart (1): indica restabelecimento de conexão
- Syn (1): indica pedido estabelecimento de conexão
- Fin (1): indica finalização de conexão

# Protocolo TCP

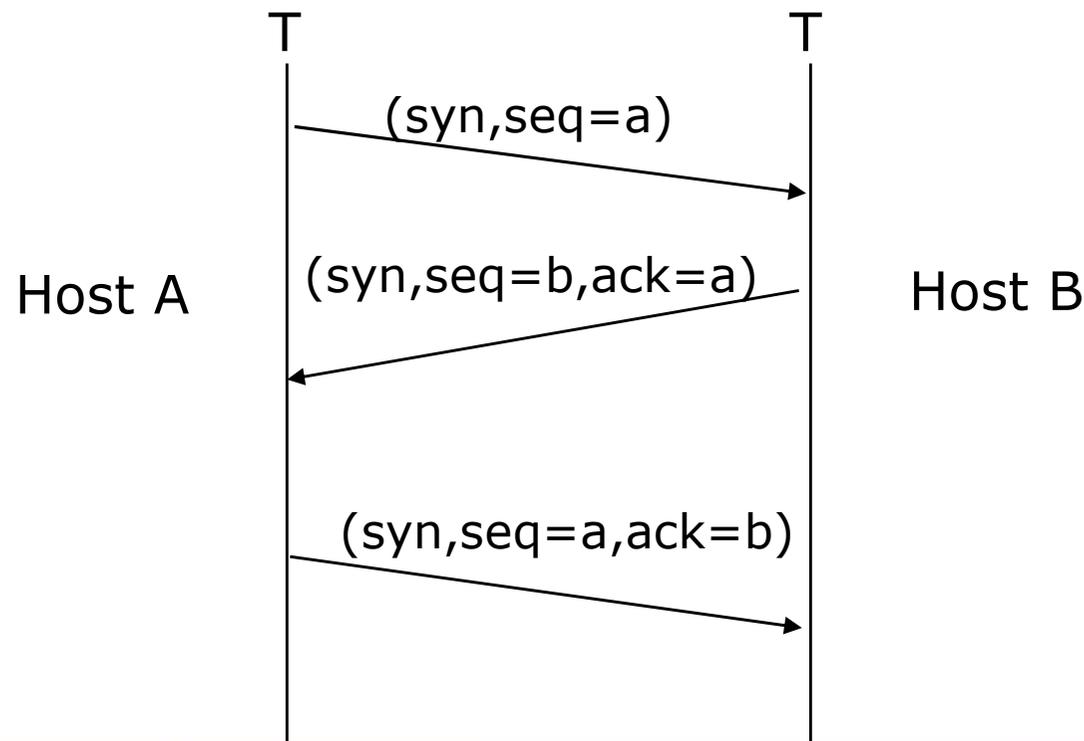
---

- ❑ *Window*: espaço disponível no transmissor disponível para receber dados do parceiro
- ❑ *Checksum*: resultado do complemento de 1 da soma de parcelas de 16 bits em complemento de um, incluindo o pseudo-header
- ❑ *Urgent pointer*: aponta para o byte dentro da stream a partir do qual há dados urgentes

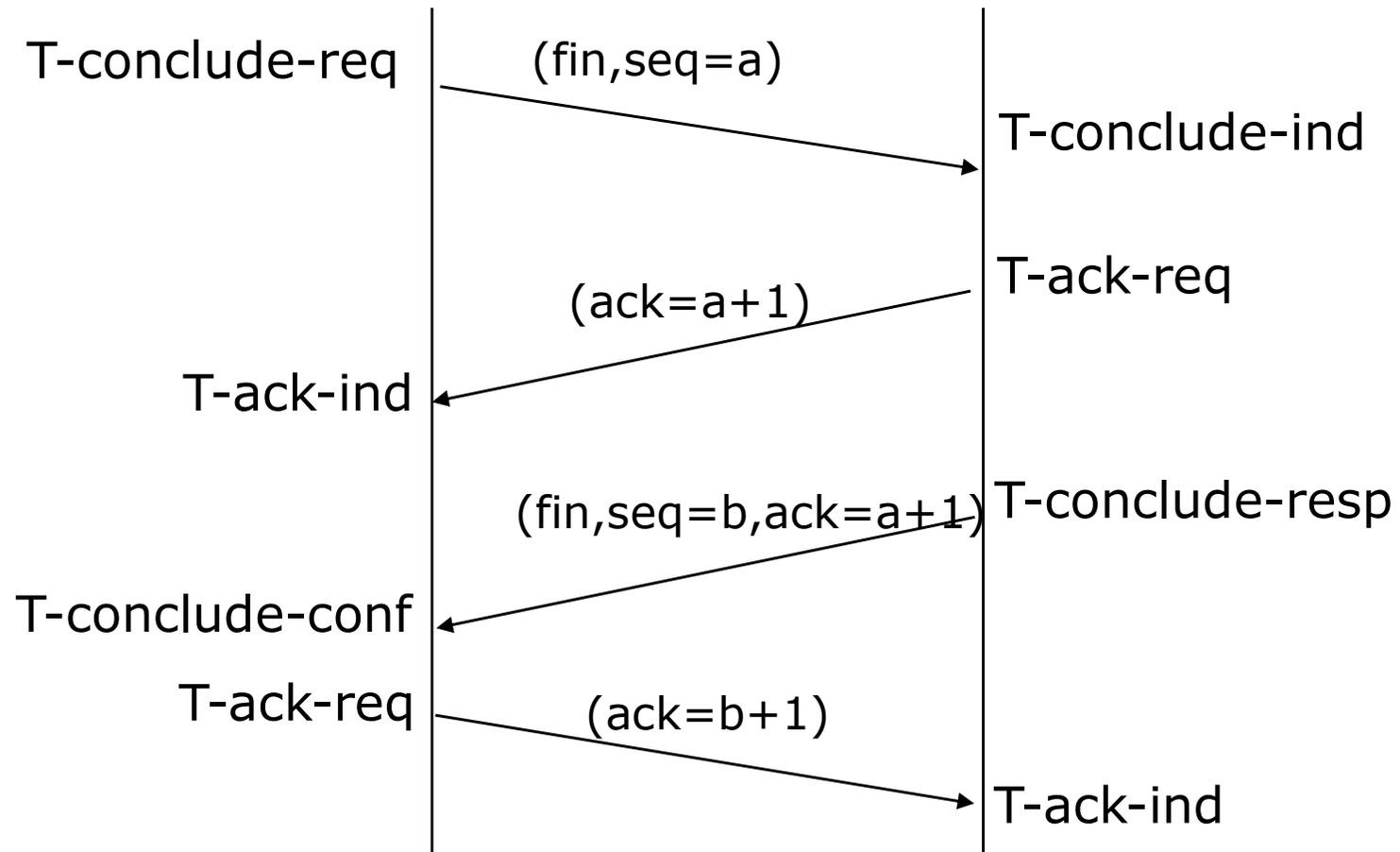
# Three-way-handshake

---

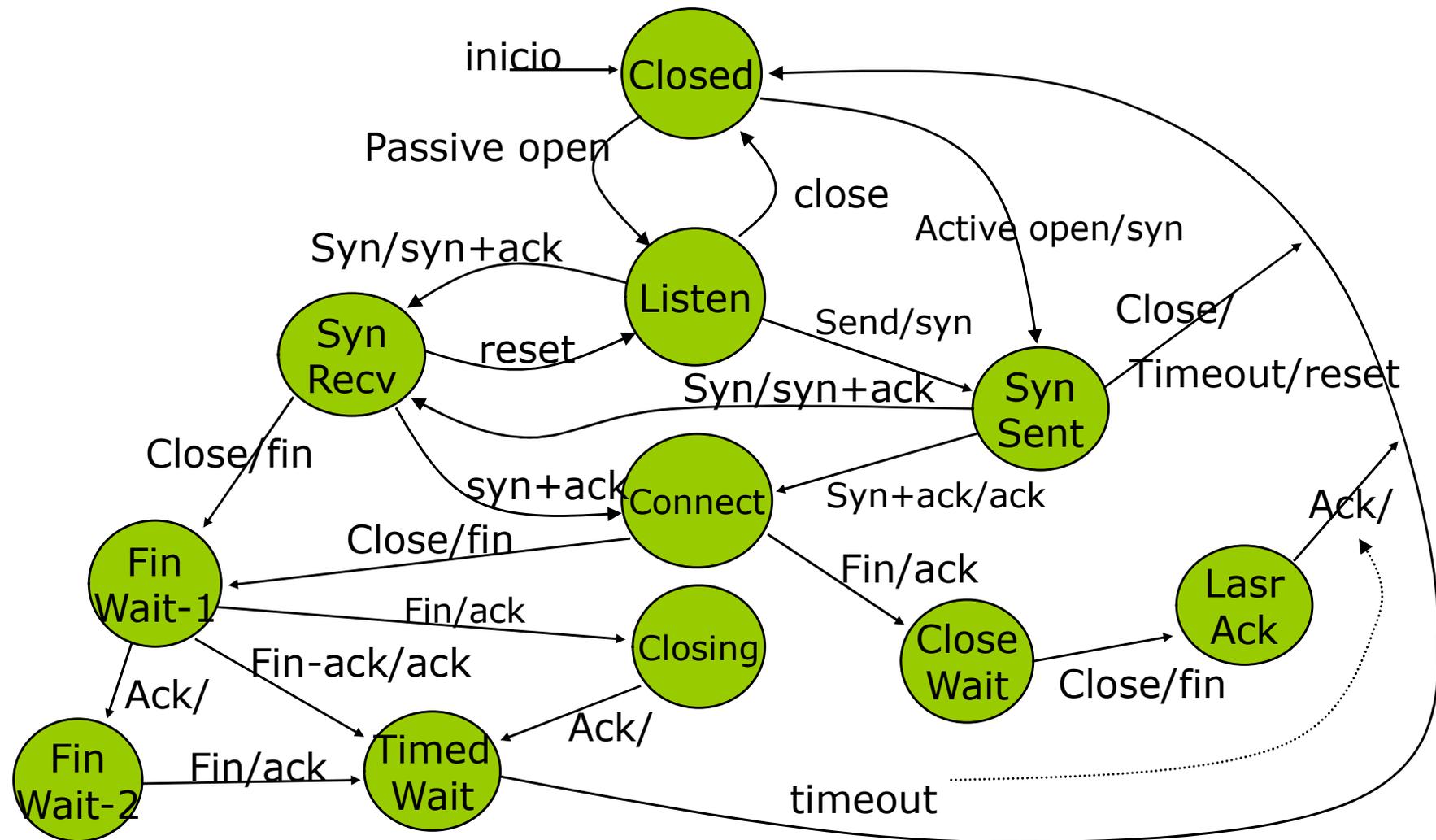
- Mecanismo para estabelecimento de conexão no qual são trocadas três primitivas



# Closing Connection



# TCP: Máquina de Estados Finita



# Portas

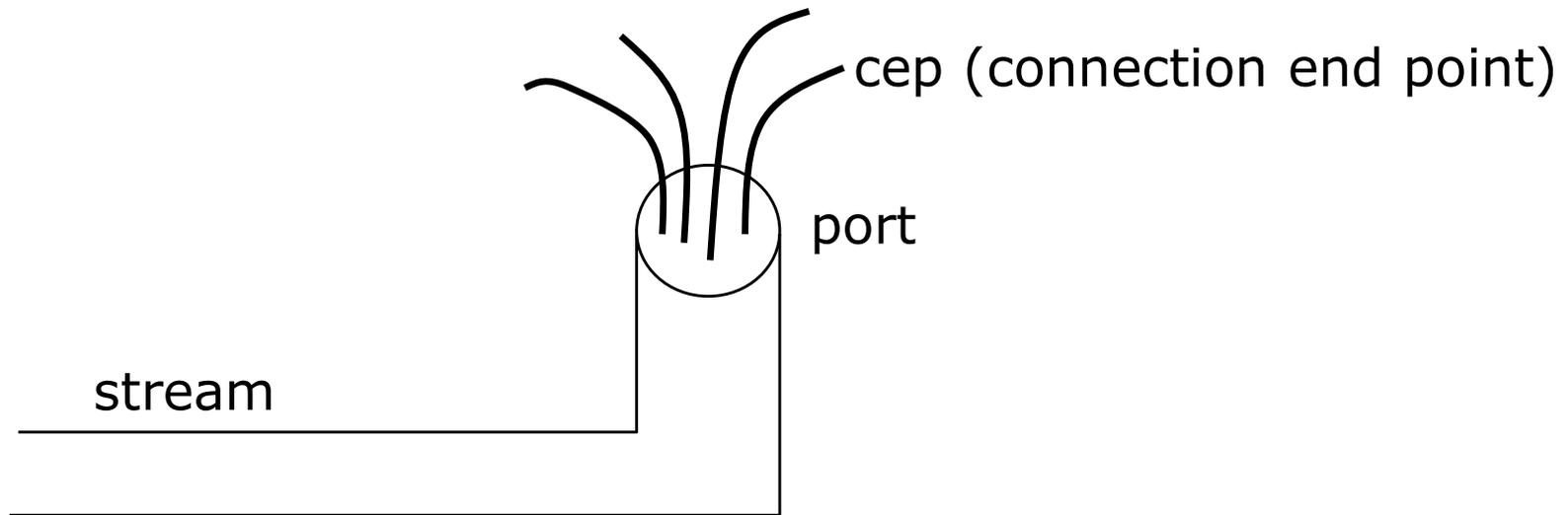
---

- ❑ As portas são os pontos de acesso à camada de transporte, e, portanto, aos protocolos TCP/UDP
- ❑ A manipulação das portas é similar ao descritor de arquivos, sendo comumente referenciado como *socket descriptor*
- ❑ *Socket* é um objeto que armazena IPAddr:port

# Portas

---

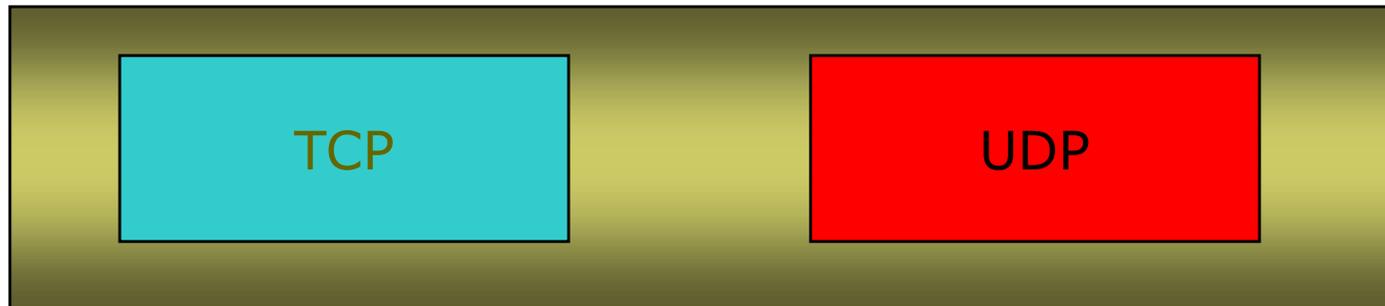
- Na realidade, várias conexões podem tomar lugar sobre um mesmo *socket* sendo que cada instância pode ser representada como a seguir:



# Camada de Transporte

## User Datagram Protocol

---



# User Datagram Protocol (UDP)

---

- Um protocolo de transporte não orientado a conexão, projetado para utilizar meios do tipo A

Source Port	Destination Port
Segment Size	Checksum
UDP Data	

Segment size inclui o cabeçalho e portanto deve ser maior ou igual a 8

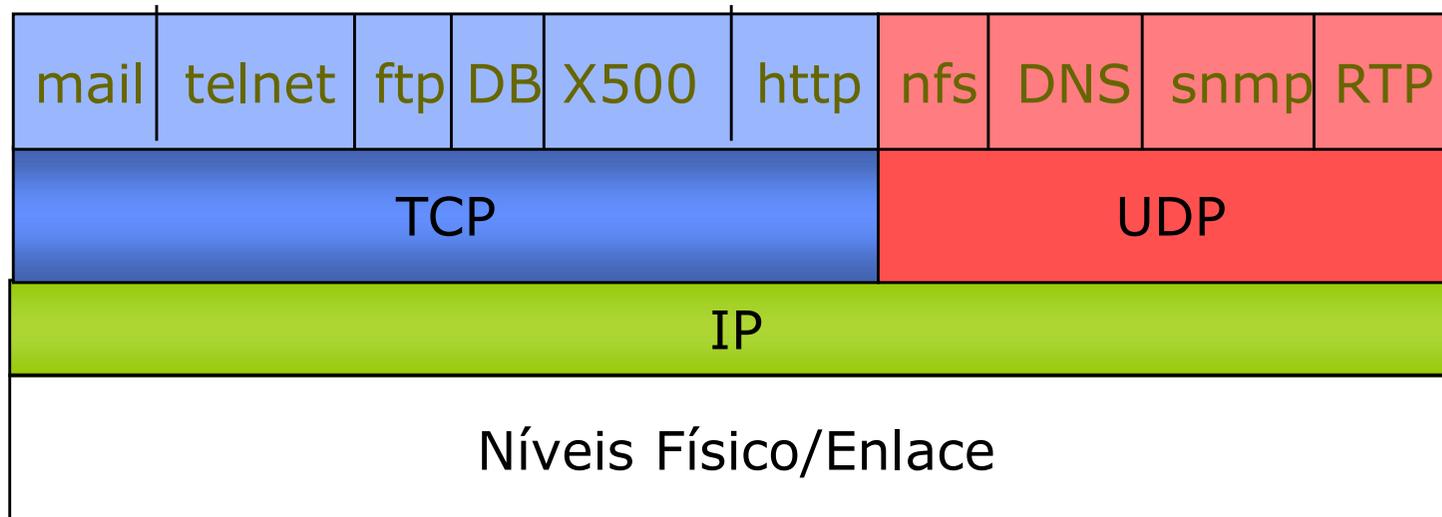
# User Datagram Protocol (UDP)

---

- O UDP é utilizado para vários fins nos dias atuais:
  - Gerenciamento de rede
  - Network File System
  - Domain Name System
  - E outras aplicações de caráter esporádico

# Visão da Arquitetura Internet

---



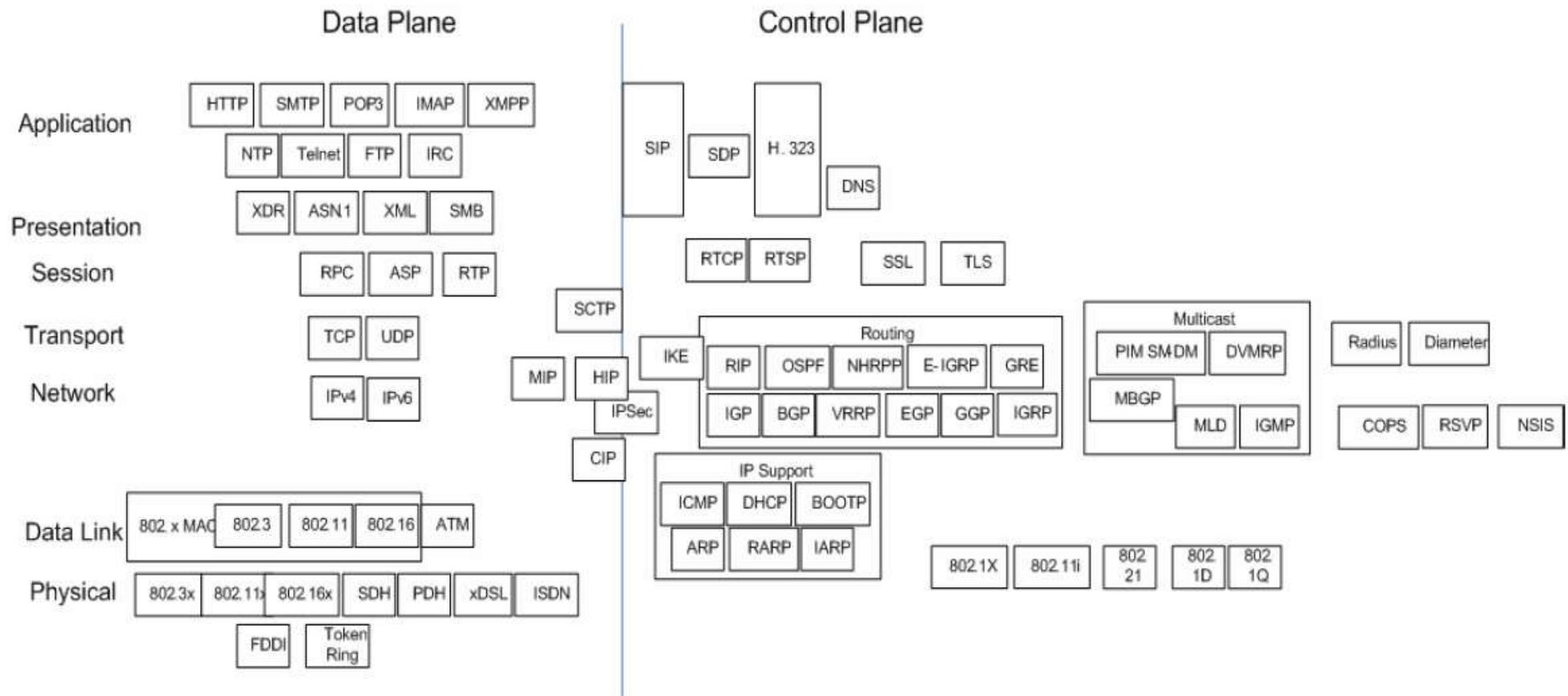
# Aplicações básicas

---

- ❑ nslookup: verifica estado e configurações do domínio (DNS)
- ❑ traceroute: relata os hops (e tempos) existentes entre o host d origem e o host destino
- ❑ netstat: verifica o estado da rede, incluindo a tabela de rotas
- ❑ route: manipula tabela de rotas

# Arquitetura TCP/IP

## Plano de Dados x Plano Controle



Visão Dos Planos de Dados e Controle Aguiar (2008)

AGUIAR, R. L. Some comments on hourglasses. **SIGCOMM Comput. Commun. Rev.**, v. 38, n. 5, p. 69–72, 2008. Disponível em: <<http://doi.acm.org/10.1145/1452335.1452346>>. Acesso em: 31/8/2013.