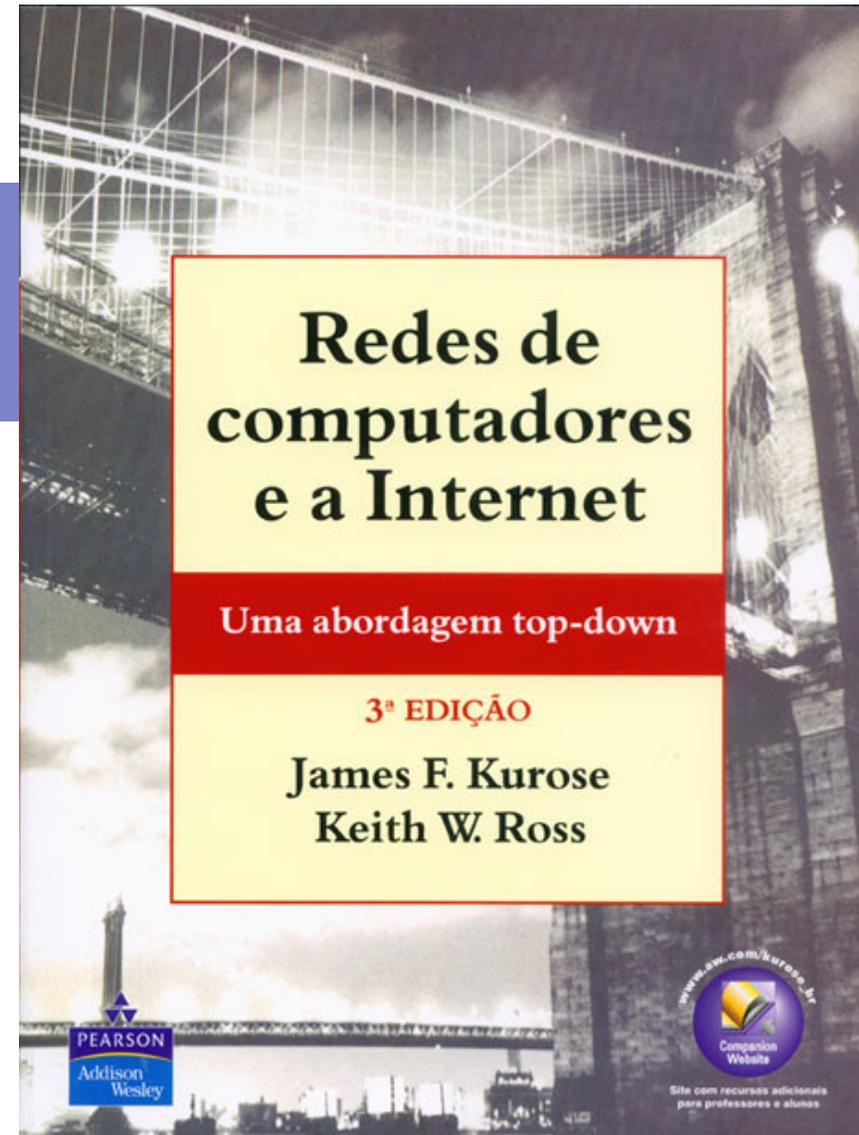


# Redes de computadores e a Internet

Prof. Gustavo Wagner

## Capítulo 3

Camada  
de  
transporte



# 3 Camada de transporte

## Objetivos do capítulo:

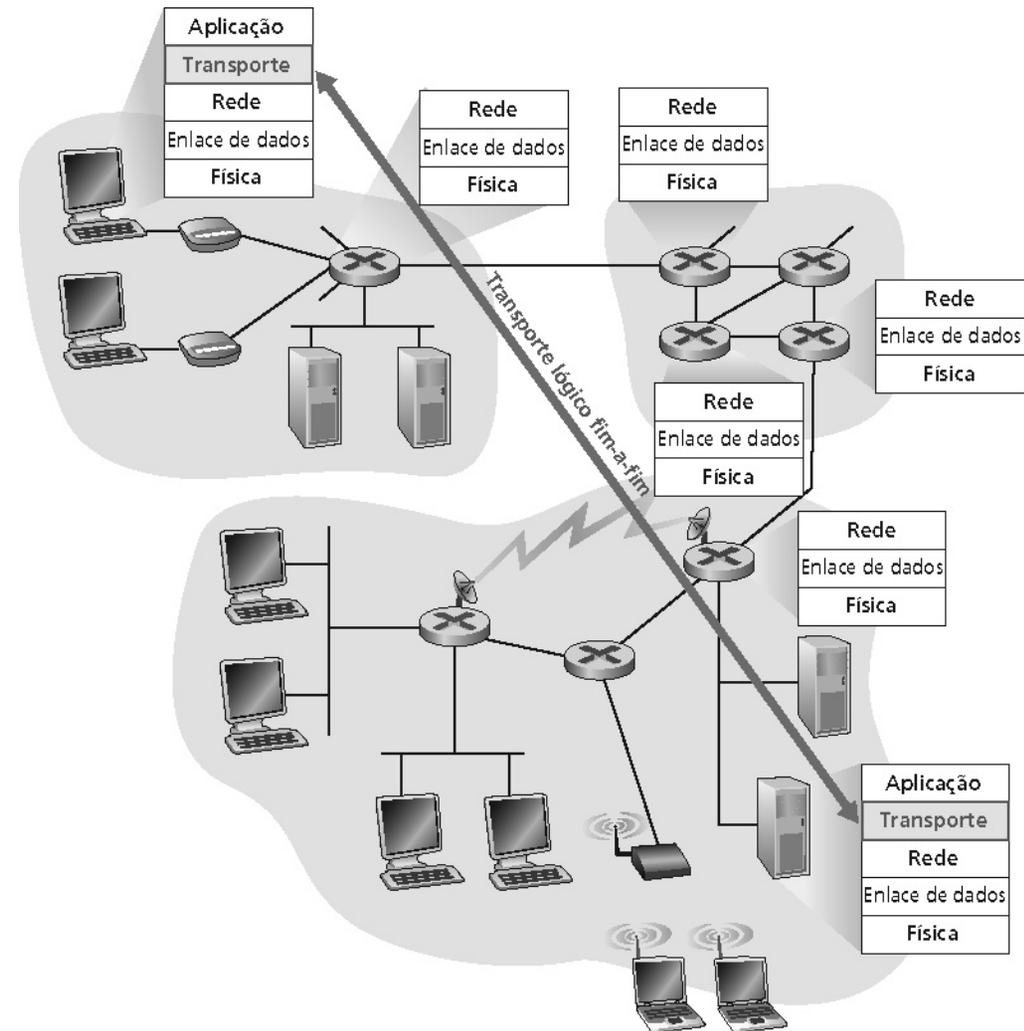
- Entender os princípios por trás dos serviços da camada de transporte:
  - Multiplexação/demultiplexação
  - Transferência de dados confiável
  - Controle de fluxo
  - Controle de congestionamento
- Aprender sobre os protocolos de transporte na Internet:
  - UDP: transporte não orientado à conexão
  - TCP: transporte orientado à conexão
  - Controle de congestionamento do TCP

# 3 Camada de transporte

- 3.1 Serviços da camada de transporte
- 3.2 Multiplexação e demultiplexação
- 3.3 Transporte não orientado à conexão: UDP
- 3.4 Princípios de transferência confiável de dados
- 3.5 Transporte orientado à conexão: TCP
  - Estrutura do segmento
  - Transferência confiável de dados
  - Controle de fluxo
  - Gerenciamento de conexão
- 3.6 Princípios de controle de congestionamento
- 3.7 Controle de congestionamento do TCP

# 3 Protocolos e serviços de transporte

- Fornecem **comunicação lógica** entre processos de aplicação em diferentes hospedeiros
- Os protocolos de transporte são executados nos sistemas finais
  - Lado emissor: quebra as mensagens da aplicação em segmentos e envia para a camada de rede
  - Lado receptor: remonta os segmentos em mensagens e passa para a camada de aplicação
- Há mais de um protocolo de transporte disponível para as aplicações
  - Internet: TCP e UDP



# 3 Camada de transporte vs. camada de rede

- **Camada de rede:** comunicação lógica entre os hospedeiros
- **Camada de transporte:** comunicação lógica entre os processos
  - Depende dos serviços da camada de rede

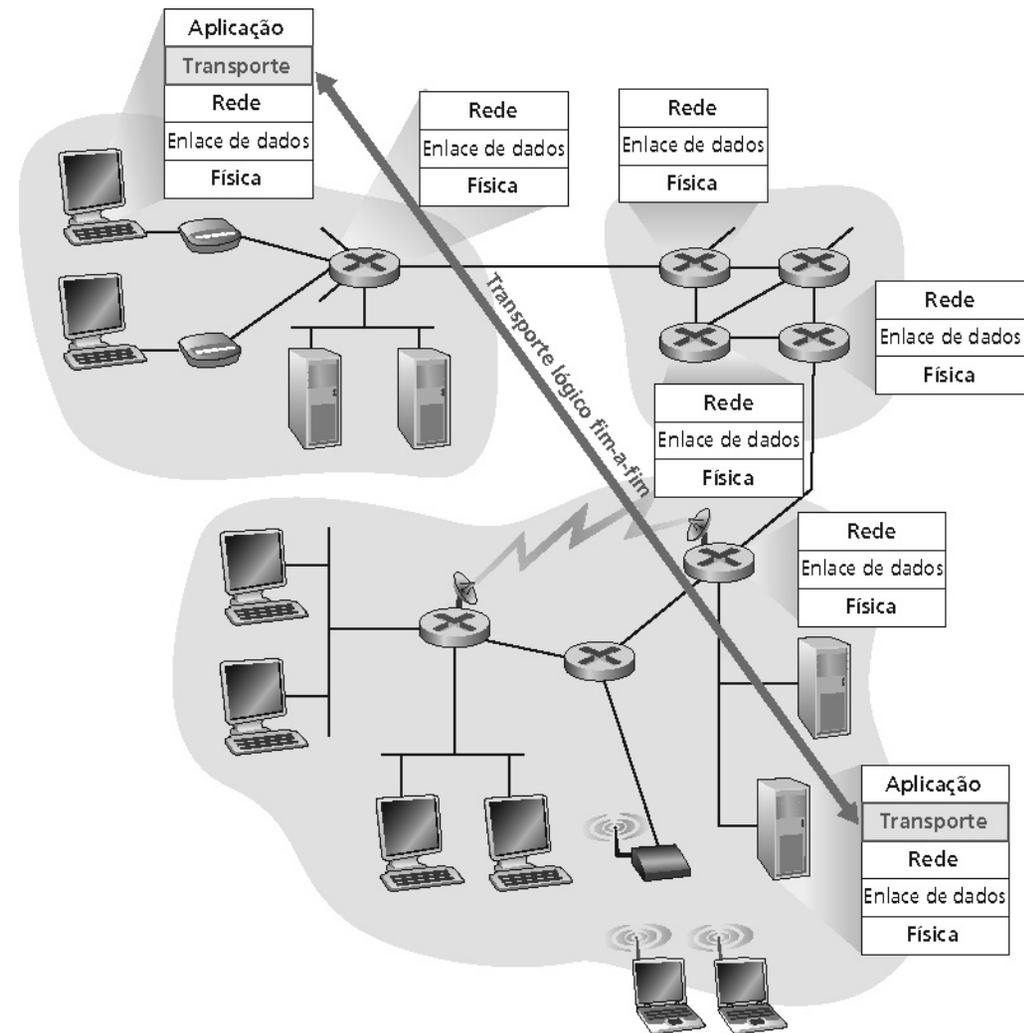
## **Analogia com uma casa familiar:**

12 crianças enviam cartas para 12 crianças

- Processos = crianças
- Mensagens da aplicação = cartas nos envelopes
- Hospedeiros = casas
- Protocolo de transporte = Anna e Bill
- Protocolo da camada de rede = serviço postal

# 3 Protocolos da camada de transporte da Internet

- Confiável, garante ordem de entrega (TCP)
  - Controle de congestionamento
  - Controle de fluxo
  - Orientado à conexão
- Não confiável, sem ordem de entrega: UDP
  - Extensão do “melhor esforço” do IP
- Serviços não disponíveis:
  - Garantia a atrasos
  - Garantia de banda



# 3 Camada de transporte

- 3.1 Serviços da camada de transporte
- 3.2 Multiplexação e demultiplexação
- 3.3 Transporte não orientado à conexão: UDP
- 3.4 Princípios de transferência confiável de dados
- 3.5 Transporte orientado à conexão: TCP
  - Estrutura do segmento
  - Transferência confiável de dados
  - Controle de fluxo
  - Gerenciamento de conexão
- 3.6 Princípios de controle de congestionamento
- 3.7 Controle de congestionamento do TCP

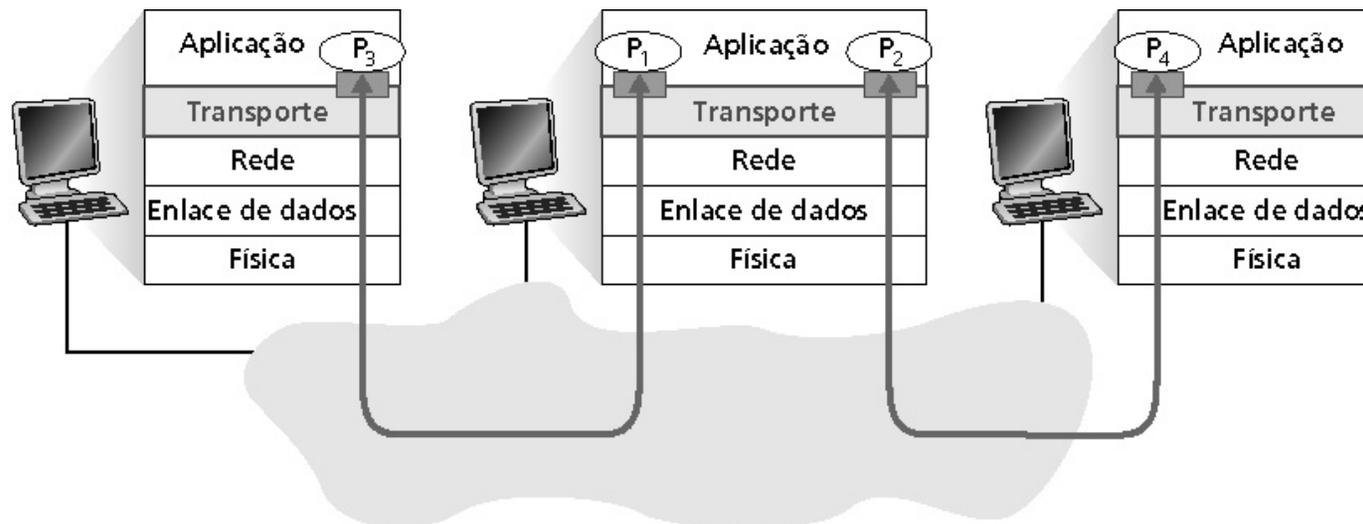
# 3 Multiplexação/demultiplexação

Demultiplexação no hospedeiro receptor:

entrega os segmentos recebidos ao socket correto

Multiplexação no hospedeiro emissor:

coleta dados de múltiplos sockets, envelope os dados com cabeçalho (usado depois para demultiplexação)

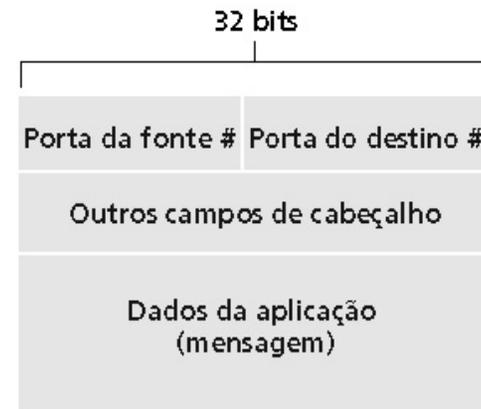


Legenda:

○ Processo ■ Socket

# 3 Como funciona a demultiplexação

- **Computador recebe datagramas IP**
  - Cada datagrama possui endereço IP de origem e IP de destino
  - Cada datagrama carrega 1 segmento da camada de transporte
  - Cada segmento possui números de porta de origem e destino (lembre-se: números de porta bem conhecidos para aplicações específicas)
- **O hospedeiro usa endereços IP e números de porta para direcionar o segmento ao socket apropriado**



# 3 Demultiplexação não orientada à conexão

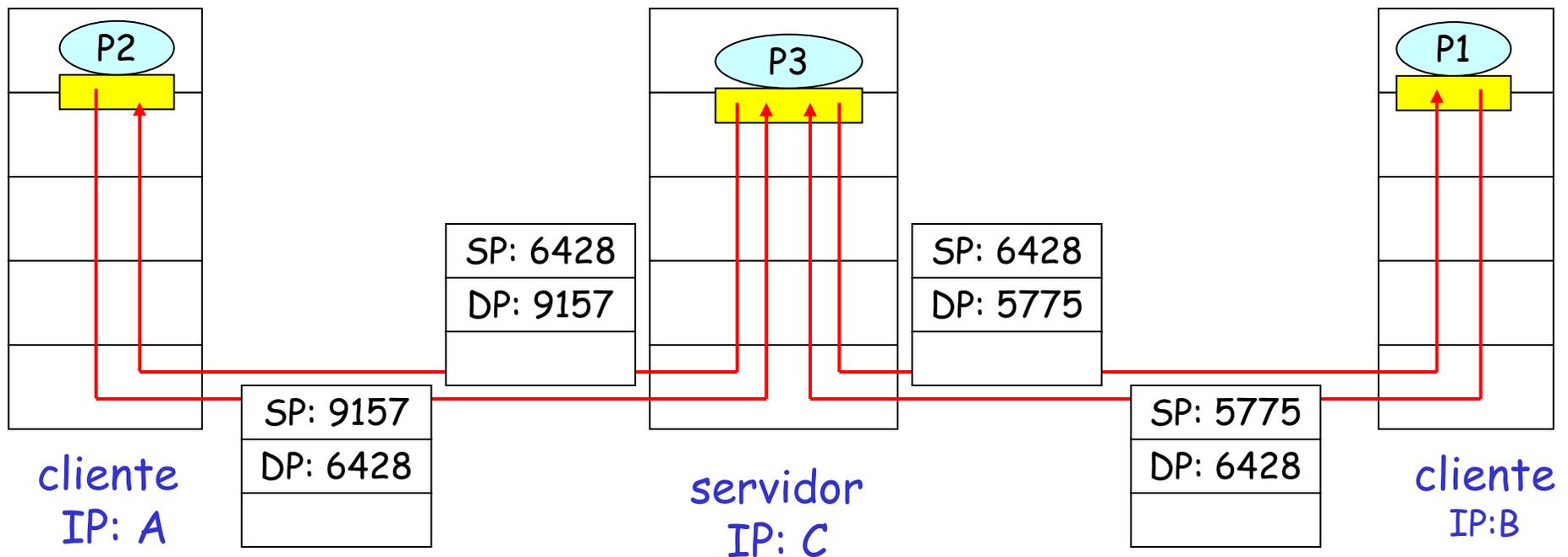
- Cria sockets com números de porta:

```
DatagramSocket mySocket1 = new DatagramSocket(99111);  
DatagramSocket mySocket2 = new DatagramSocket(99222);
```

- Socket UDP identificado por 2 valores:  
(endereço IP de destino, número da porta de destino)
- Quando o hospedeiro recebe o segmento UDP:
  - Verifica o número da porta de destino no segmento
  - Direciona o segmento UDP para o socket com este número de porta
- Datagramas com IP de origem diferentes e/ou portas de origem diferentes são direcionados para o mesmo socket

# 3 Demultiplexação não orientada à conexão

```
DatagramSocket serverSocket = new DatagramSocket(6428);
```

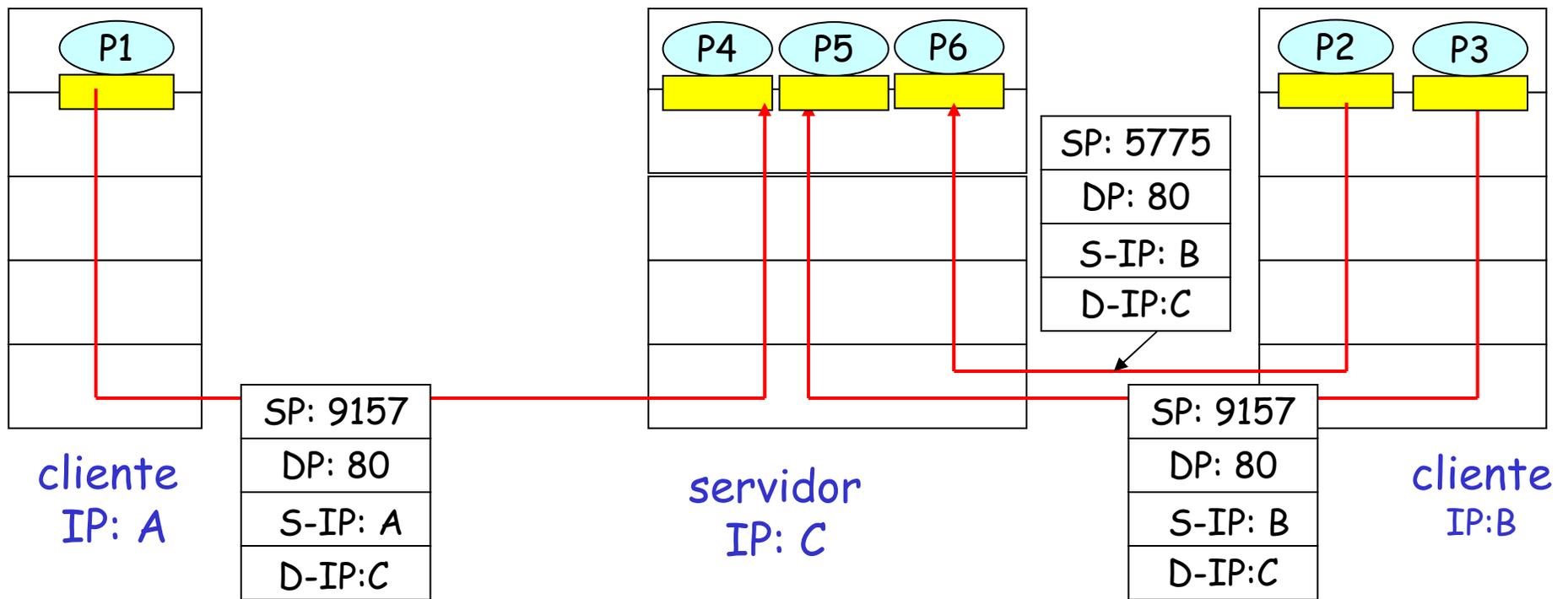


SP fornece o “endereço retorno”0

# 3 Demux orientada à conexão

- Socket TCP identificado por 4 valores:
  - Endereço IP de origem
  - End. porta de origem
  - Endereço IP de destino
  - End. porta de destino
- Hospedeiro receptor usa os quatro valores para direcionar o segmento ao socket apropriado
- Hospedeiro servidor pode suportar vários sockets TCP simultâneos:
  - Cada socket é identificado pelos seus próprios 4 valores
  - Servidores Web possuem sockets diferentes para cada cliente conectado
  - HTTP não persistente terá um socket diferente para cada requisição
  - HTTP persistente terá um único socket no lado servidor

# 3 Demux orientada à conexão



# 3 Demux orientada à conexão servidor Web “threaded”

