

# Conteúdo do Capítulo 3

- ❑ 3.1 Serviços da camada de transporte
- ❑ 3.2 Multiplexação e demultiplexação
- ❑ 3.3 UDP: Transporte não orientado a conexão
- ❑ 3.4 Princípios da transferência confiável de dados
- ❑ 3.5 Transporte orientado a conexão: TCP
  - transferência confiável
  - controle de fluxo
  - gerenciamento de conexões
- ❑ 3.6 Princípios de controle de congestionamento
- ❑ 3.7 Controle de congestionamento do TCP

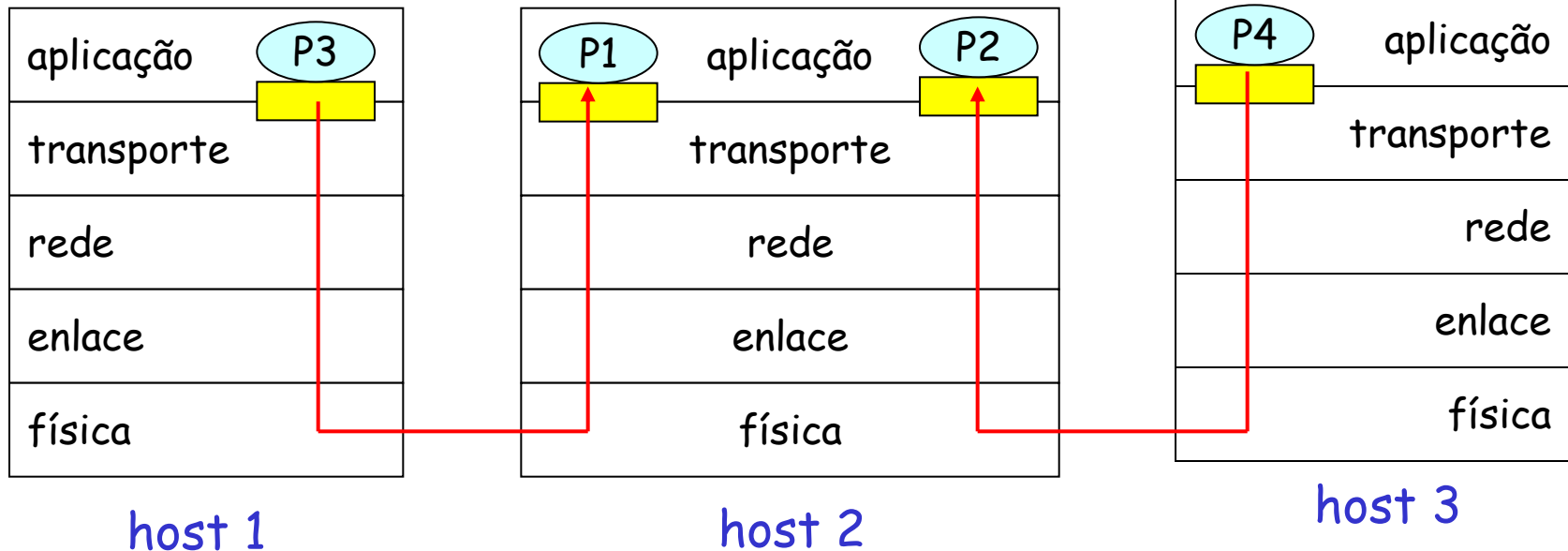
# Multiplexação/demultiplexação

## Processo:

Pode ter um ou mais **sockets**, portas pelas quais dados passam da rede para o processo e do processo para rede.

 = socket       = processo

A camada de transporte do host destinatário não entrega dados diretamente a um processo, mas a um **socket** intermediário.



# Multiplexação/demultiplexação

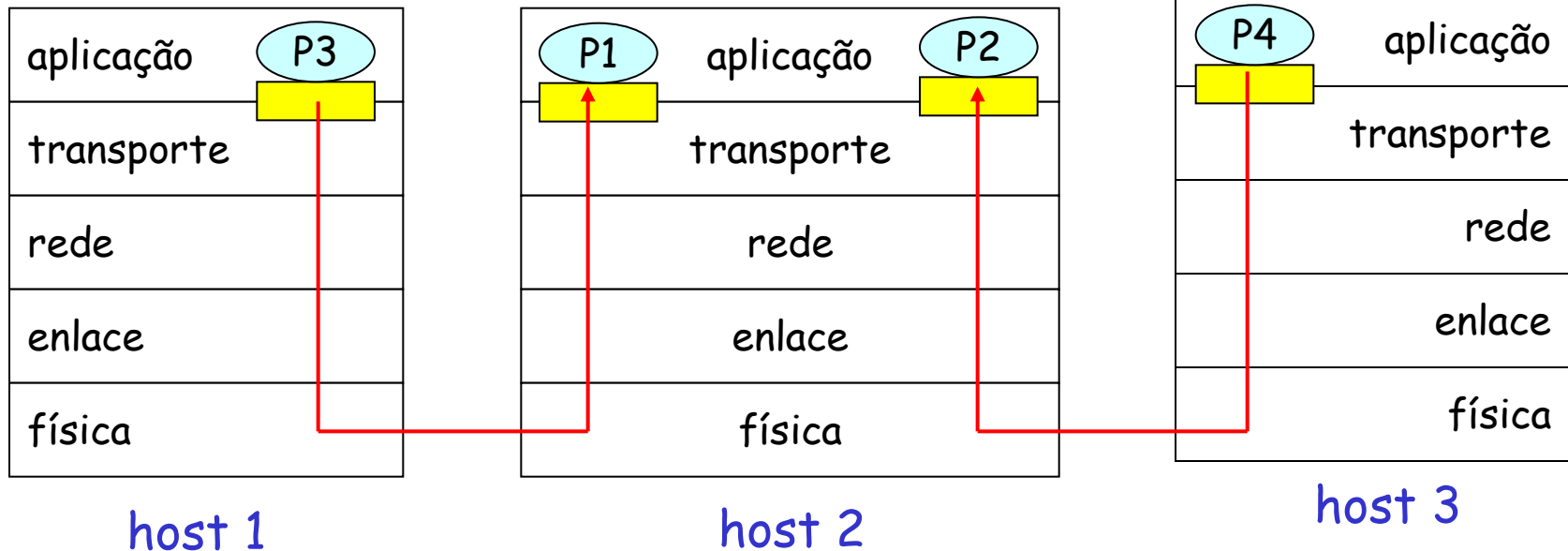
## Demultiplexação no destino:

A tarefa de entregar os dados contidos em um segmento à porta correta é chamada de...

■ = socket      ○ = processo

## Multiplexação na origem.:

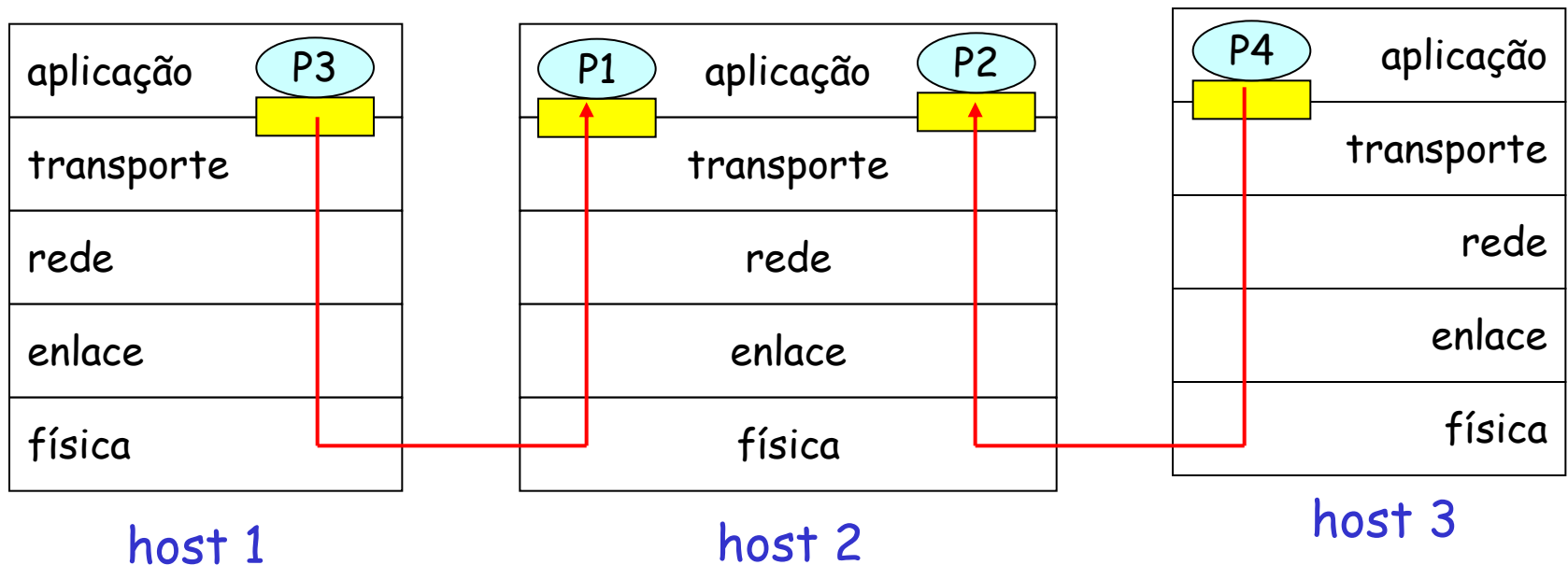
O trabalho de reunir dados de muitas portas, encapsular os dados com o cabeçalho (usado posteriormente para a demultiplexação) para criar segmentos e passá-los para camada de rede é chamada de .....



# Multiplexação/demultiplexação

Note que a camada de transporte do hospedeiro que está no meio tem de demultiplexar segmentos que chegam da camada de rede abaixo para os processos P1 e P2 acima; isso é feito direcionando à porta correspondente do processo os dados contidos no segmento que está chegando. A camada de transporte desse hospedeiro também tem de juntar dados de saída dessas portas, formar segmentos da camada de transporte e passá-los à camada de rede.

 = socket       = processo

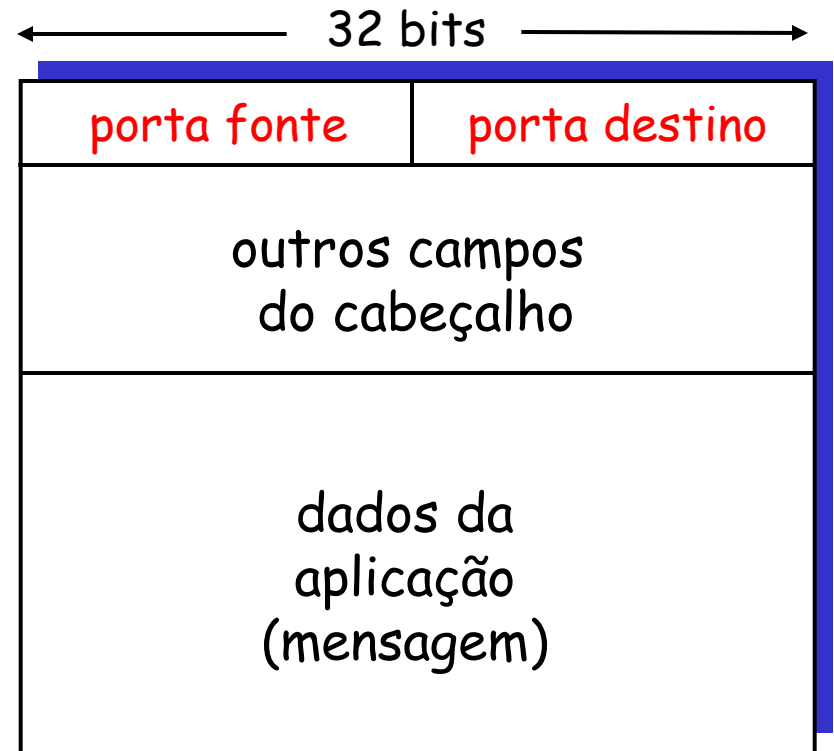


# Multiplexação/demultiplexação

- ❑ Lembra da metáfora das duas casas de crianças?
- ❑ Cada uma das crianças é identificada por seu nome próprio.
- ❑ Quando Bill recebe uma grande quantidade de correspondência do carteiro, realiza uma operação de demultiplexação ao examinar para quem as cartas estão endereçadas e, em seguida, entregar a correspondência a seus irmãos.
- ❑ Ann realiza uma operação de multiplexação quando coleta as cartas de seus irmãos e entrega a correspondência na agência do correio.

# Como funciona a demultiplexação

- ❑ **host recebe os datagramas IP**
  - cada datagrama possui os endereços IP da origem e do destino
  - cada datagrama transporta 1 segmento da camada de transporte
  - cada segmento possui números das portas origem e destino (lembre: números de portas bem conhecidas para aplicações específicas)
- ❑ **host usa os endereços IP e os números das portas para direcionar o segmento ao socket apropriado**



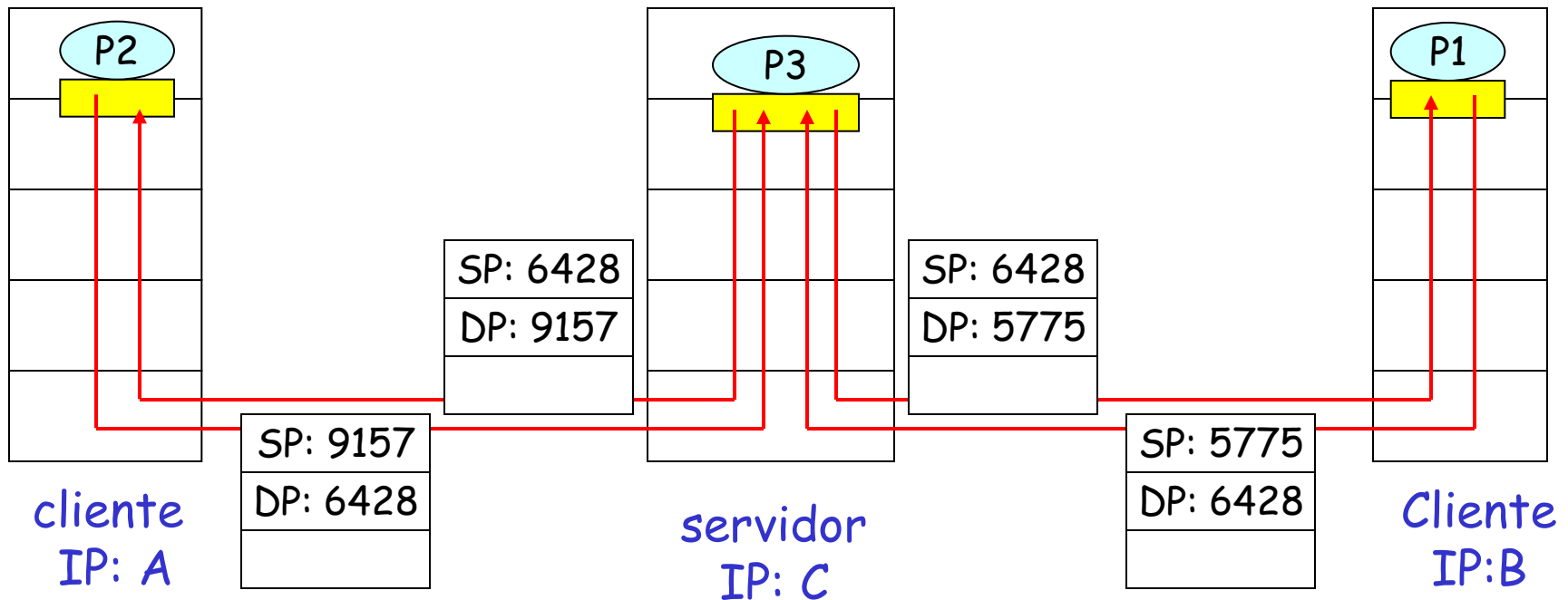
formato de segmento  
TCP/UDP

# Multiplexação/demultiplexação

- ❑ Cada número de porta é um número de 16 bits na faixa de 0 a 65535.
- ❑ Os números de porta 0 e 1023 são denominados **números de porta bem conhecidos**; eles são restritos!
- ❑ Lista dos números de porta em <http://www.iana.org>
- ❑ Estão reservados para utilização por protocolos de aplicação bem conhecidos como:
  - HTTP (PORTA 80)
  - FTP (PORTA 21)
- ❑ Sempre que for desenvolvida uma nova aplicação, deve-se atribuir a ela um número de porta.

# Demultiplexação sem Conexões

O processo P2 cuja porta UDP é 9157 envia dados de aplicação ao processo P3 cuja porta é 6428. A camada de transporte do host A cria um segmento (dados da aplicação mais número das portas fonte e destino) e passa pra camada de rede.

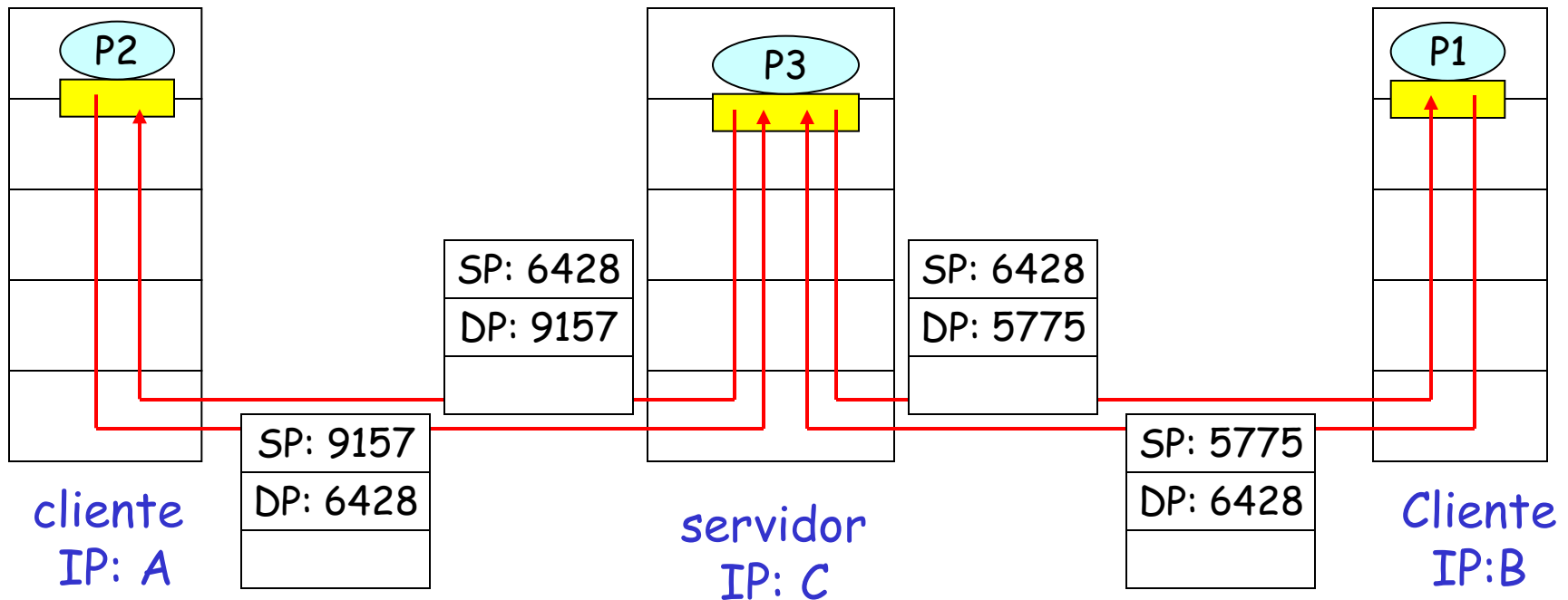


SP (*source port*) provê "endereço de retorno"



# Demultiplexação sem Conexões

A camada de rede encapsula o segmento em um datagrama IP e faz uma tentativa de melhor esforço para entregar o segmento ao hospedeiro destinatário. Se o segmento chegar a máquina de destino, esta examinará o número da porta de destino 6428 e o entregará a sua porta identificada por este numero.

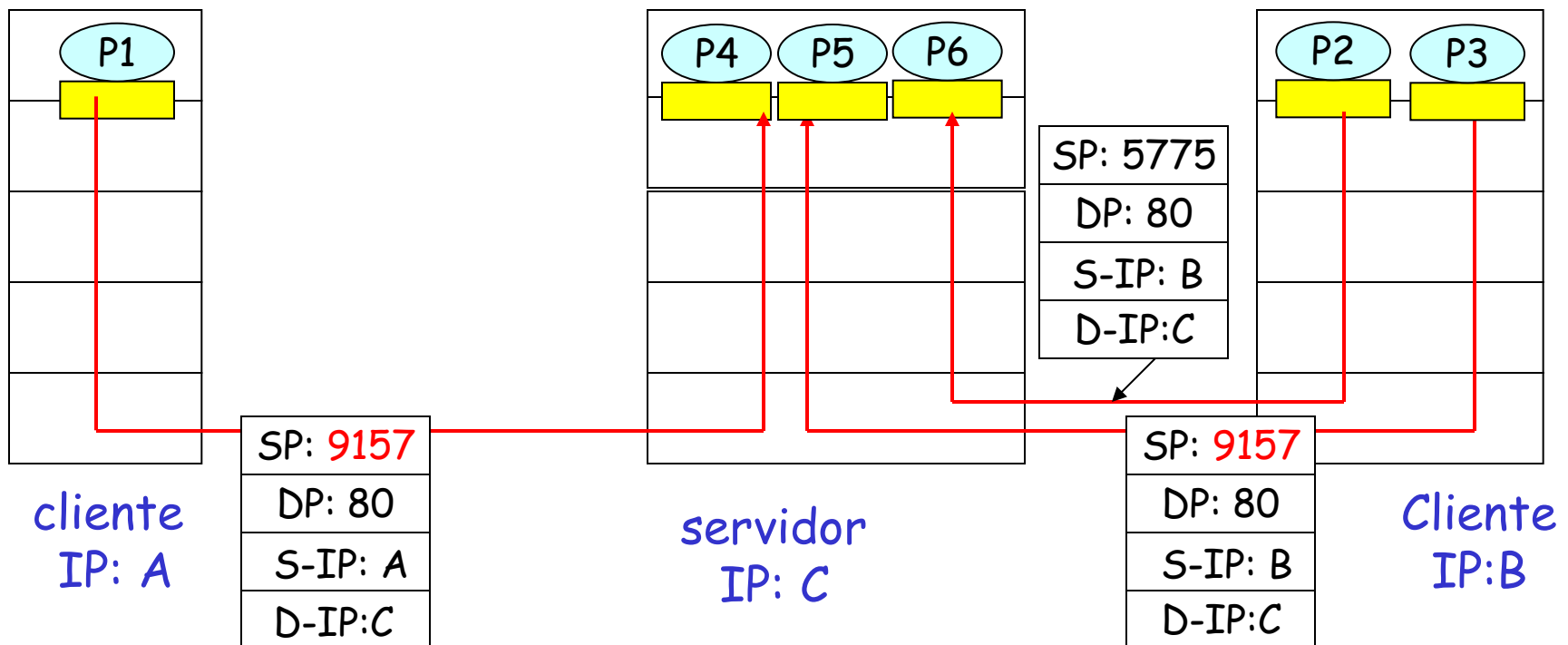


SP (*source port*) provê "endereço de retorno"

# Demultiplexação Orientada a Conexões

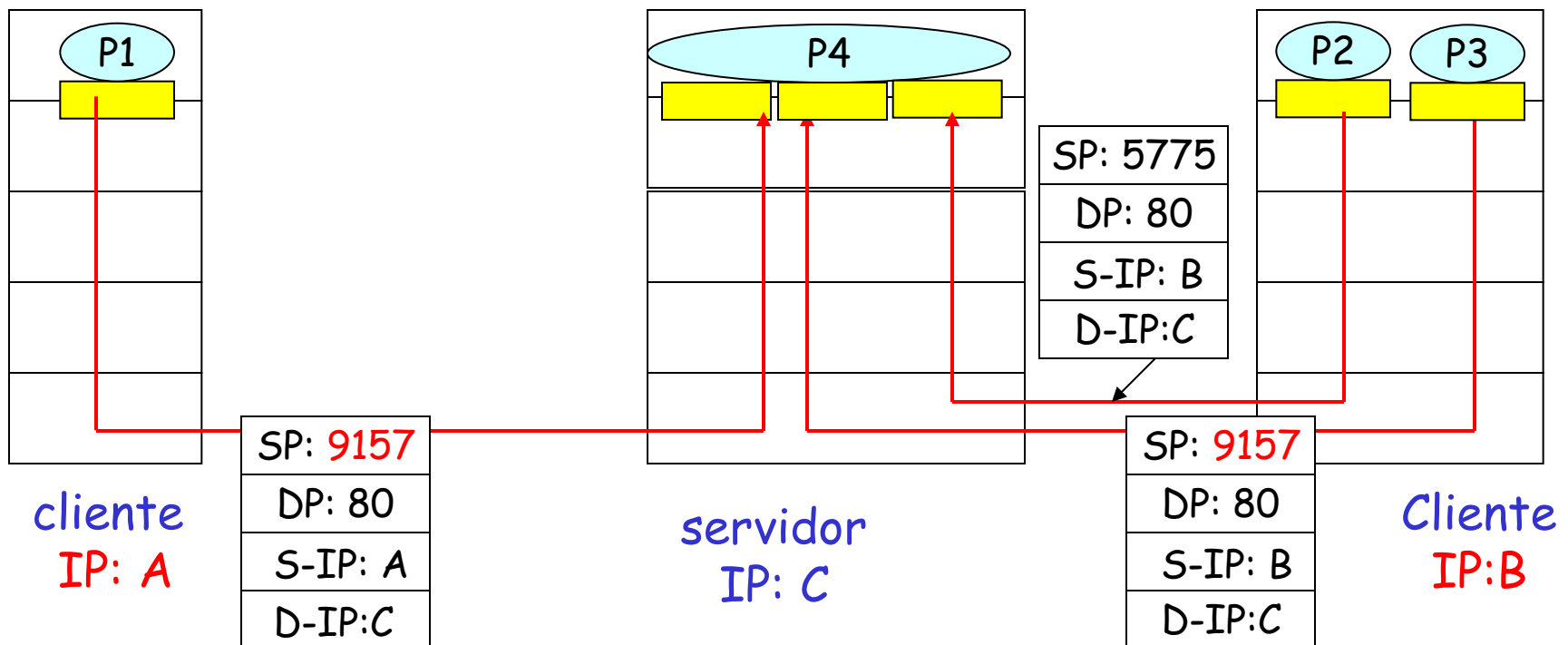
- ❑ Socket TCP identificado pela tupla-4 elementos:
  - endereço IP origem
  - número da porta origem
  - endereço IP destino
  - número da porta destino
- ❑ receptor usa todos os quatro valores para direcionar (demultiplexar) o segmento para o socket apropriado
- ❑ O host Servidor pode suportar vários sockets TCP simultâneos:
  - cada socket é ligado a um processo e identificado pela sua tupla de 4 elementos.
- ❑ Servidores Web geram um novo processo para cada nova (têm sockets diferentes) conexão cliente.

# Demultiplexação Orientada a Conexões (cont)



# Demultiplexação Orientada a Conexões

O servidor é capaz de demultiplexar corretamente as duas conexões que têm o mesmo número de porta de fonte, já que elas têm endereço IP de fontes diferentes.



# exercício

1. O que é Multiplexação?
2. Sabendo que um socket UDP é totalmente identificado por uma tupla com dois elementos consistindo de um numero de porta de origem e um numero de porta de destino. Desenhe um hospedeiro A enviando um segmento para um servidor B e depois o retorno desse segmento para o hospedeiro A.
3. Agora, sabendo que o socket TCP é identificado por uma tupla de quatro elementos, que elementos são estes? Explique a figura abaixo. Dois clientes que usam o mesmo numero de porta de destino (80) para se comunicar com a mesma aplicação de servidor Web

