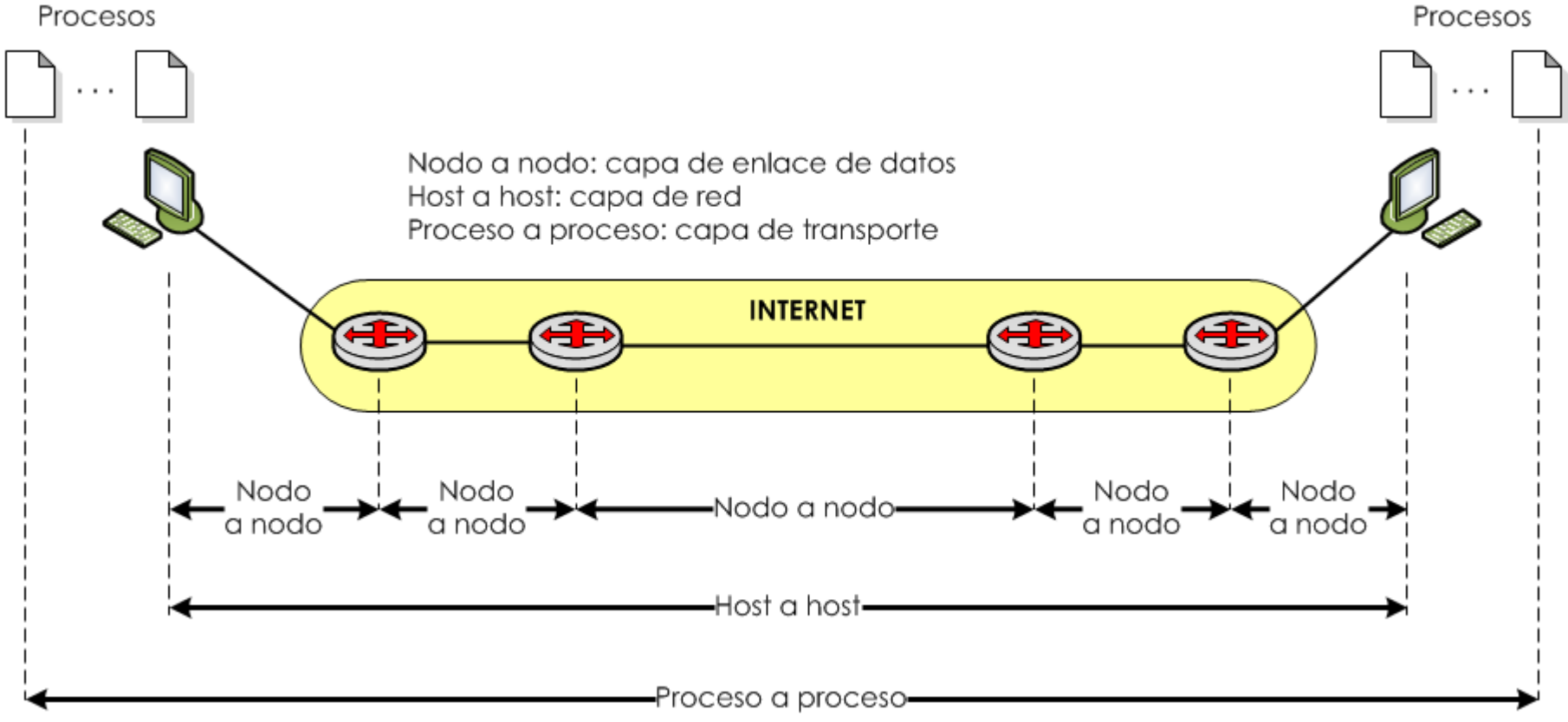
LA CAPA DE TRANSPORTE

Basado en Kurose & Ross – *"Computer Networking. A Top-Down Approach"*

El contexto de la capa de transporte

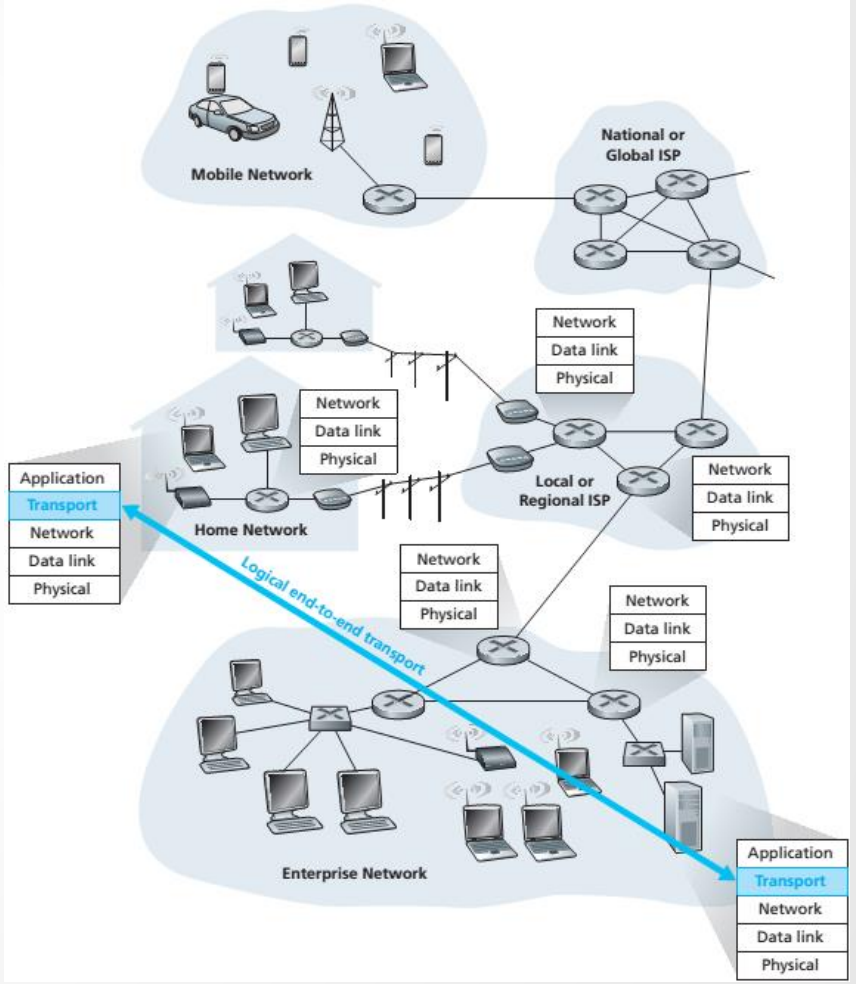


La capa de transporte

- Principios detrás de los servicios de la capa de transporte:
  - Multiplexado/Demultiplexado
  - Transferencia de datos fiable
  - Control de flujo
  - Control de congestionamiento
- Los protocolos de transporte en Internet:
  - UDP: Transporte no orientado a la conexión
  - TCP: Transporte orientado a la conexión
  - SCTP: Stream Control Transmission Protocol

Servicios y protocolos de transporte

- Proporcionar *comunicación lógica* entre procesos de aplicaciones ejecutándose en hosts diferentes
- Los protocolos de transporte se ejecutan en sistemas finales
  - Lado emisor: dividir los mensajes de las aplicaciones en segmentos, pasarlos a la capa de red
  - Lado receptor: re ensamblar segmentos en mensajes, pasar a la capa de aplicación
- Existe más de un protocolo de transporte disponible para las aplicaciones
  - TCP
  - UDP
  - SCTP



Capa de transporte vs. capa de red

- *Capa de red:*
  - Comunicación lógica entre hosts
- *Capa de transporte:*
  - Comunicación lógica entre procesos
  - Opera sobre la capa de red y mejora los servicios de la capa de red

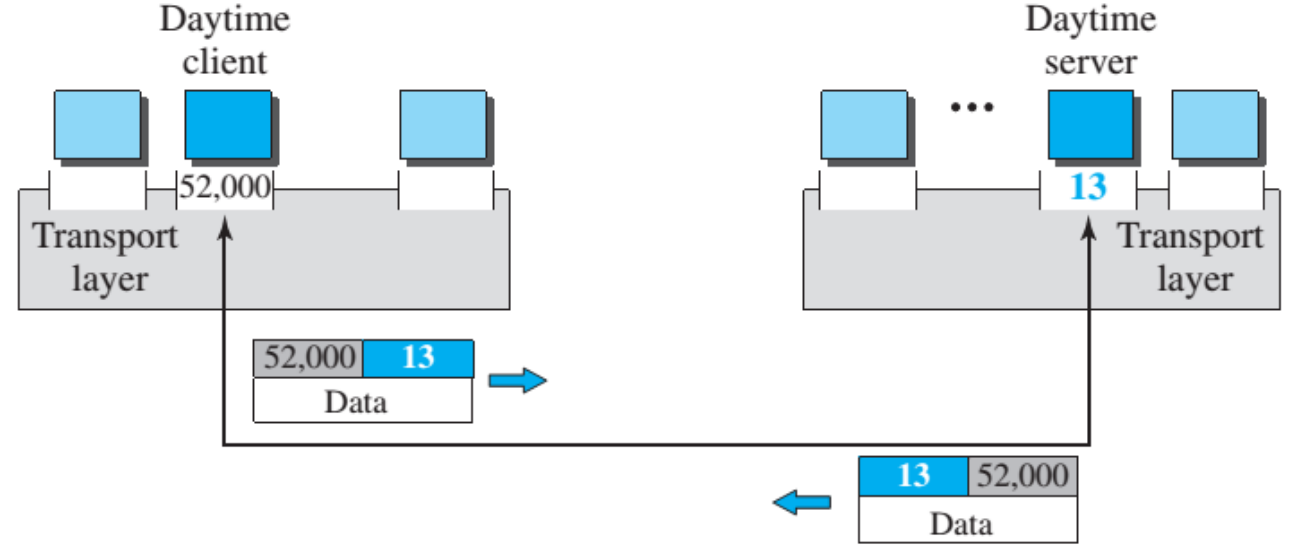
Analogía con una familia:

12 niños envían cartas a otros 12 niños

- Procesos = niños
- Mensajes de aplicación = cartas en sobres
- Hosts = casas
- Protocolo de transporte = Ana y Bill
- Protocolo de capa de red = servicio postal

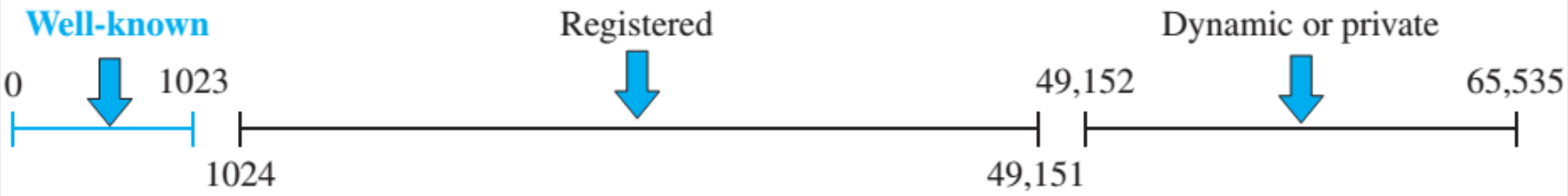
Número de puerto

- Un puerto es una interfaz lógica asociada a una aplicación de red.
- Se define mediante un entero de 16 bits.



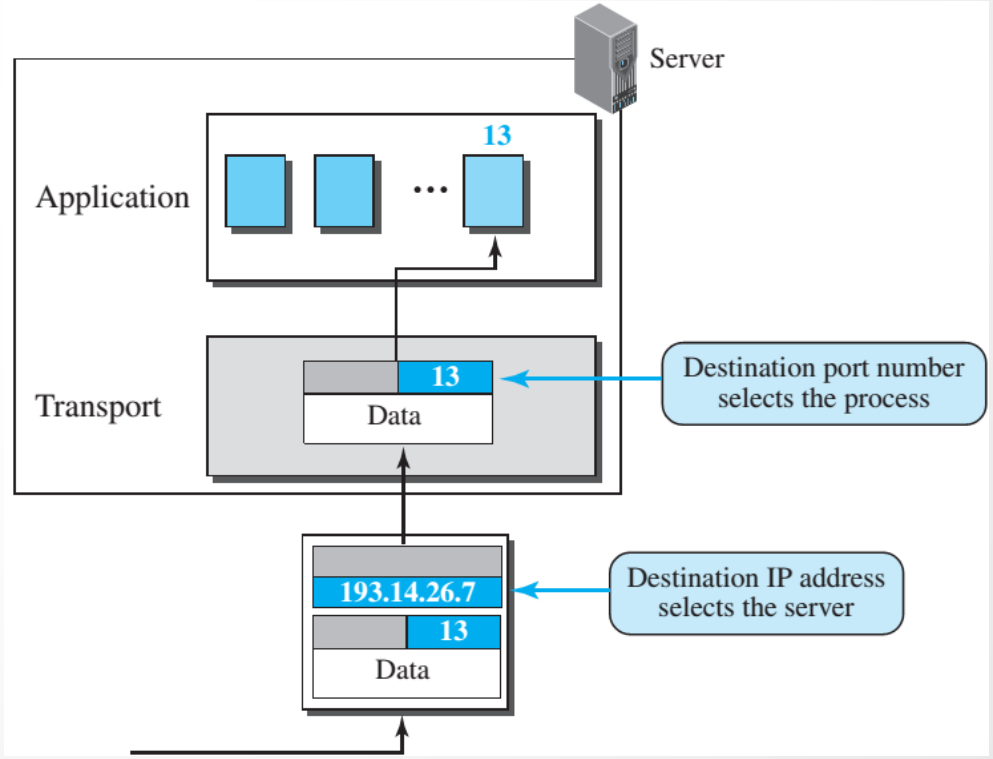
Número de puerto

- Los números de puerto son administrados por ICANN (Internet Corporation for Assigned Names and numbers) a traves de IANA (Internet Assigned numbers Authority).
- Se han definido los siguientes rangos:
  - ***Puertos bien conocidos***: de 0 a 1023 – reservado para su uso por parte de aplicaciones “bien conocidas”. Son asignados y controlados por ICANN
  - ***Puertos registrados***: de 1024 a 49151 – pueden ser usados por cualquier aplicación. No son ni controlados ni asignados por ICANN, pero pueden registrarse con ICANN para evitar duplicidad.
  - ***Puertos dinámicos o privados***: de 49152 a 65535 – pueden ser usados como puertos efímeros o privados.



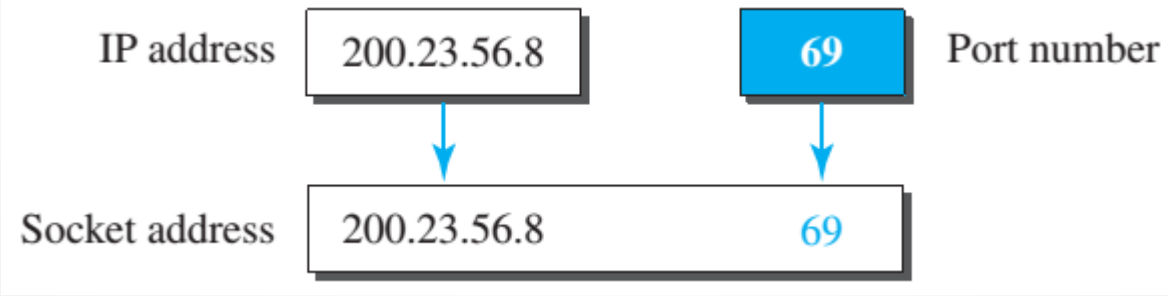
Dirección IP y número de puerto

- Dirección IP – Identifica al host
- Número de puerto – identifica al servicio

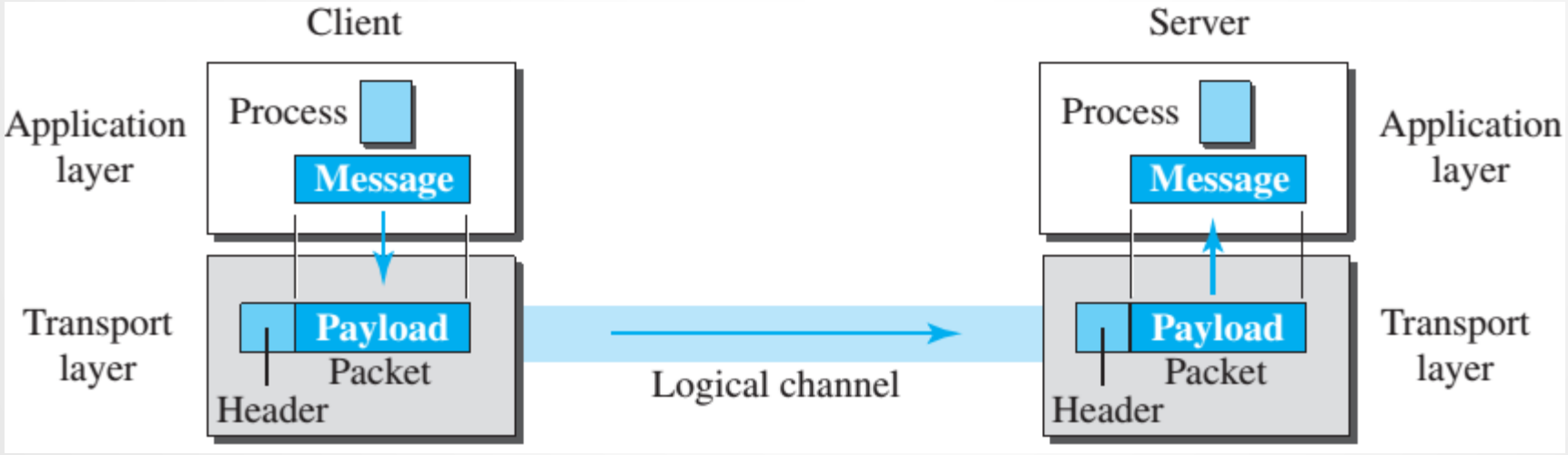


SOCKET

- Es un extremo en una comunicación interprocesos a través de una red de computadoras.
- La dirección de un socket se determina por la dirección IP del host y el número de puerto del servicio

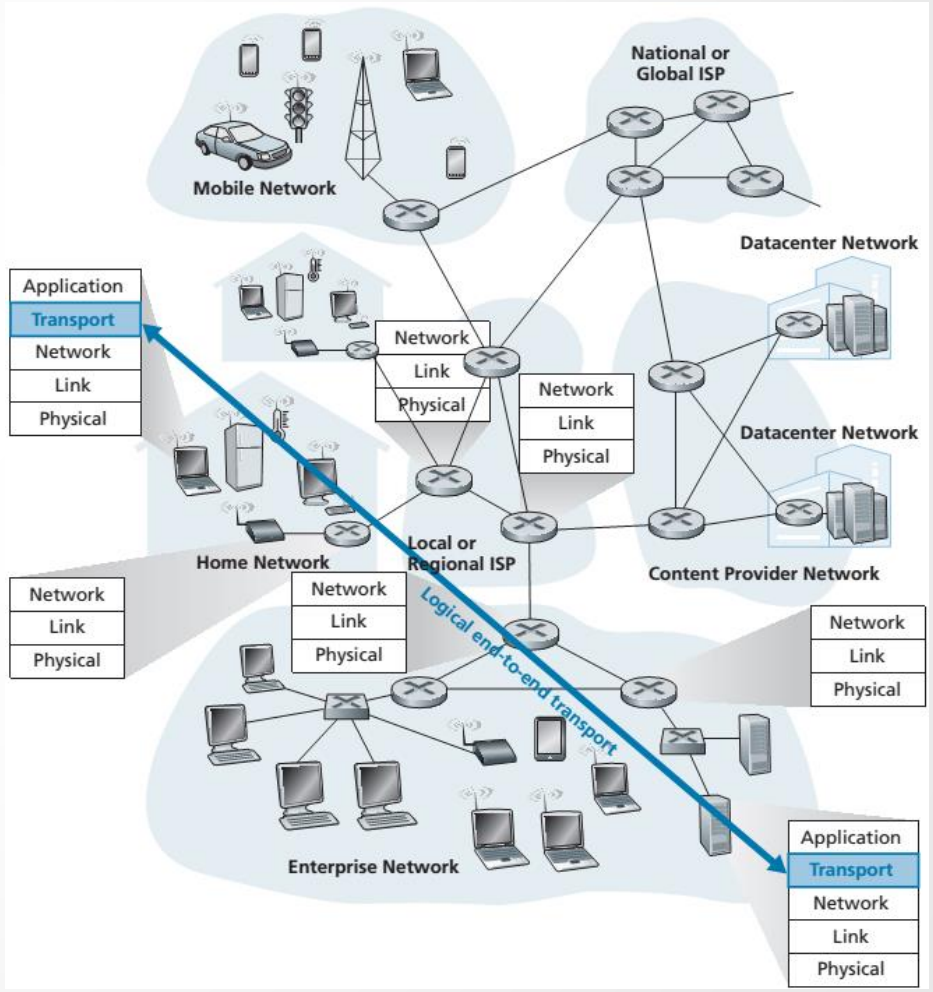


Encapsulamiento y desencapsulamiento

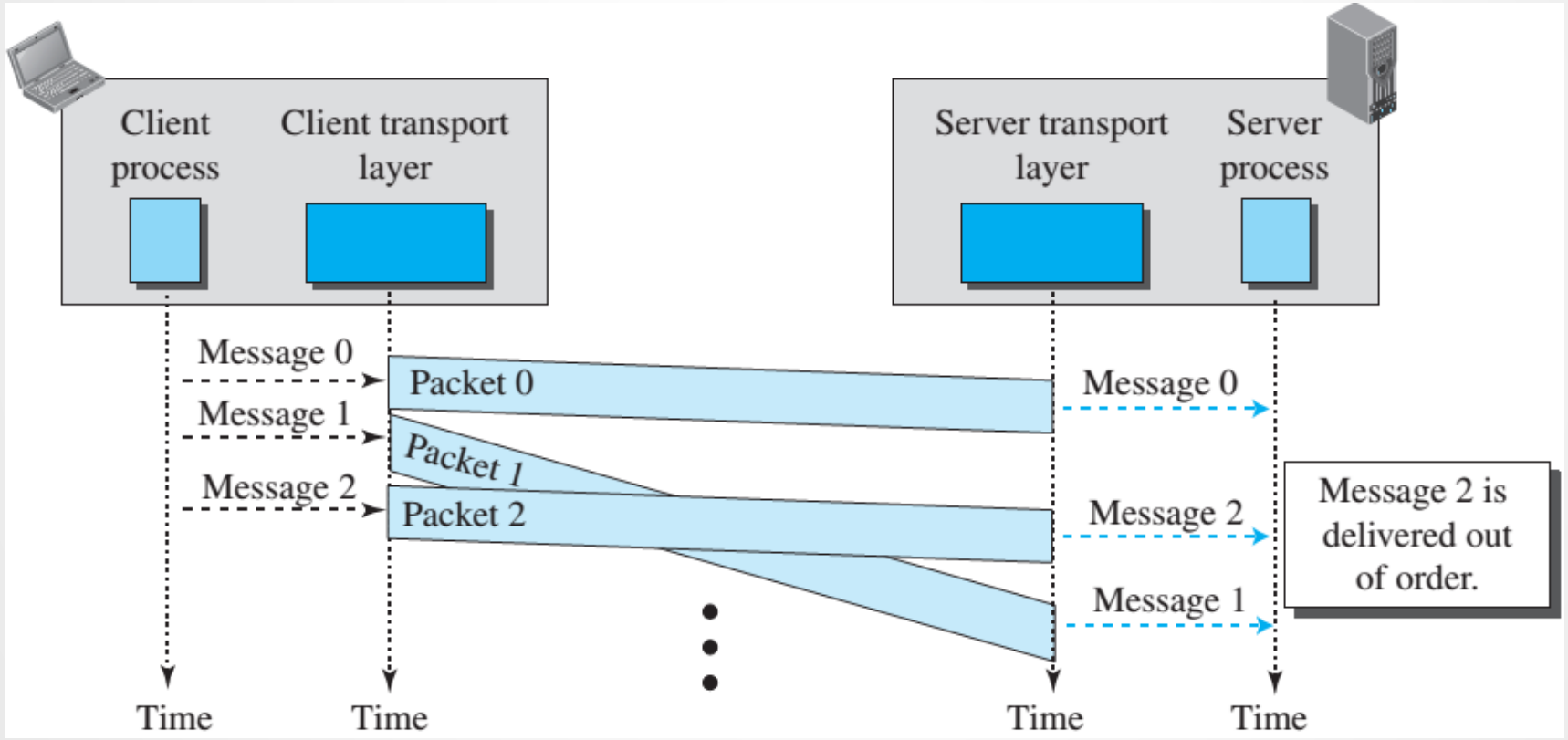


Protocolos de la capa de transporte

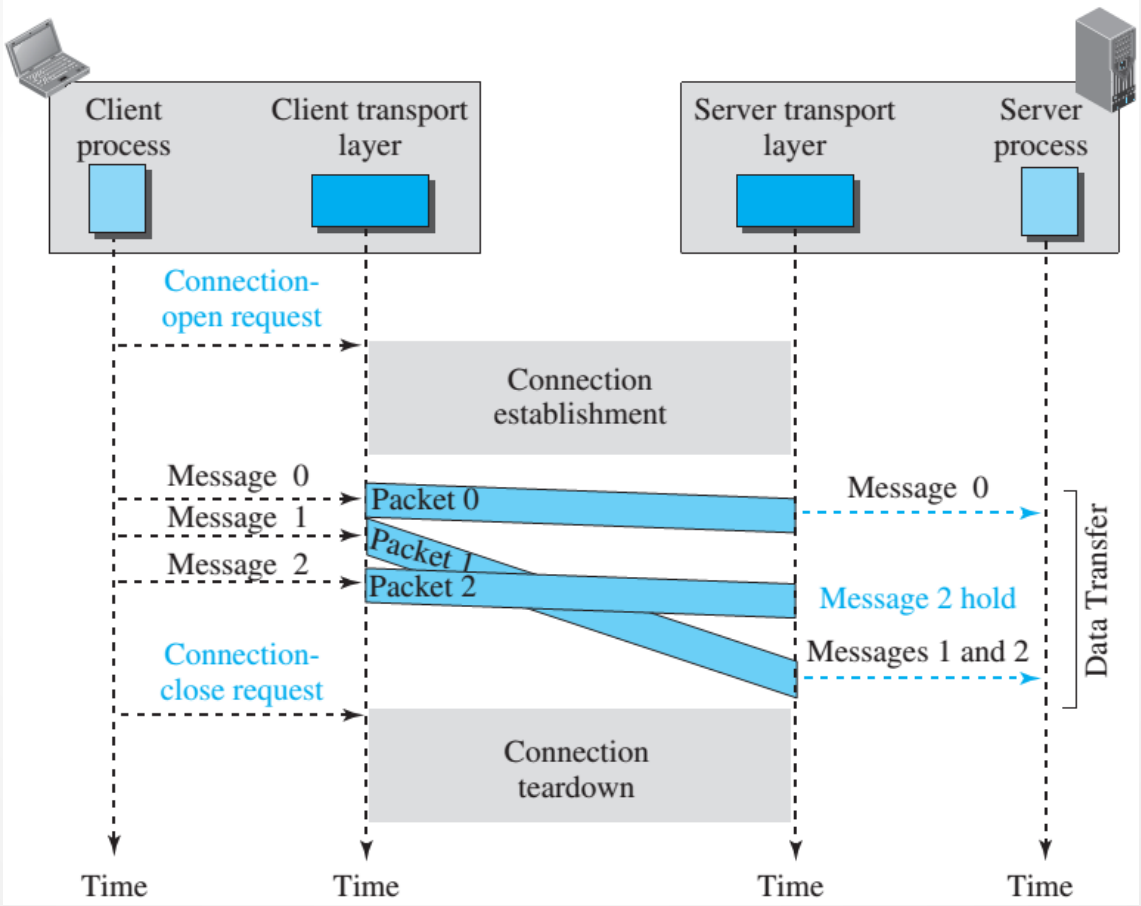
- Entrega fiable y ordenada: TCP
  - Control de congestionamiento
  - Control de flujo
  - Configuración de conexión
- Entrega no fiable y fuera de orden: UDP
  - Una extensión sin mas del protocolo IP de “mejor esfuerzo”
- Servicios no disponibles:
  - Garantías de retardo
  - Garantías de ancho de banda



Servicio no orientado a la conexión

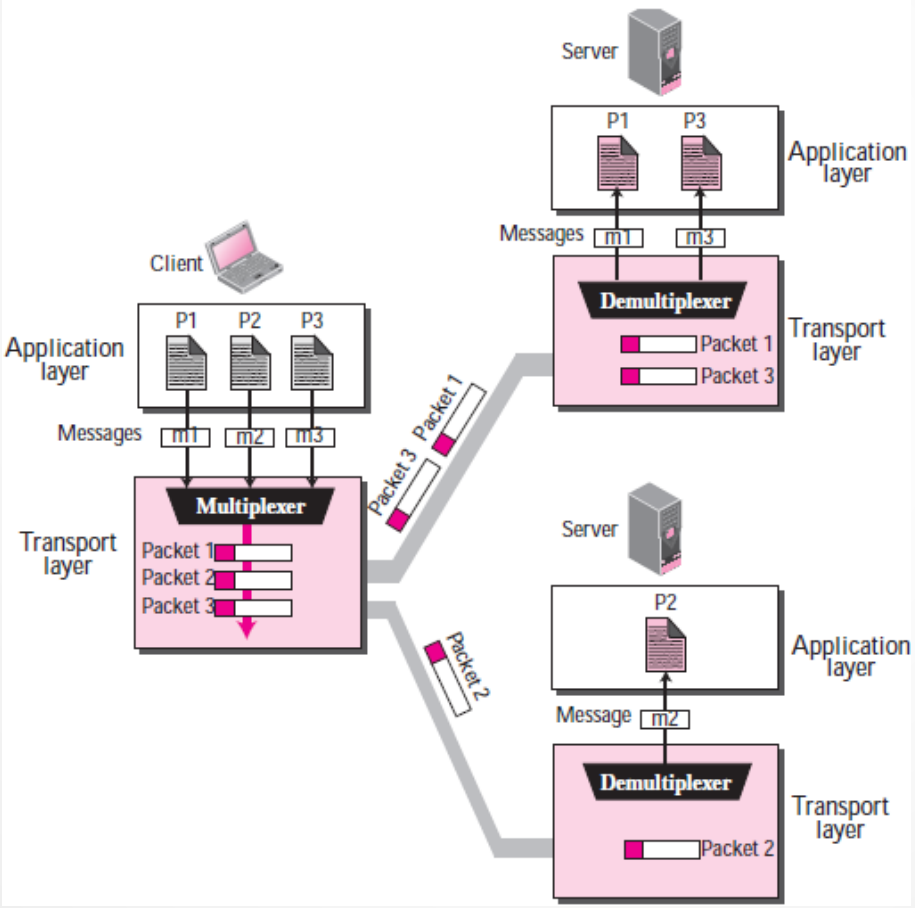


Servicio orientado a la conexión



MULTIPLEXADO Y DEMULTIPLEXADO

Multiplexado y demultiplexado



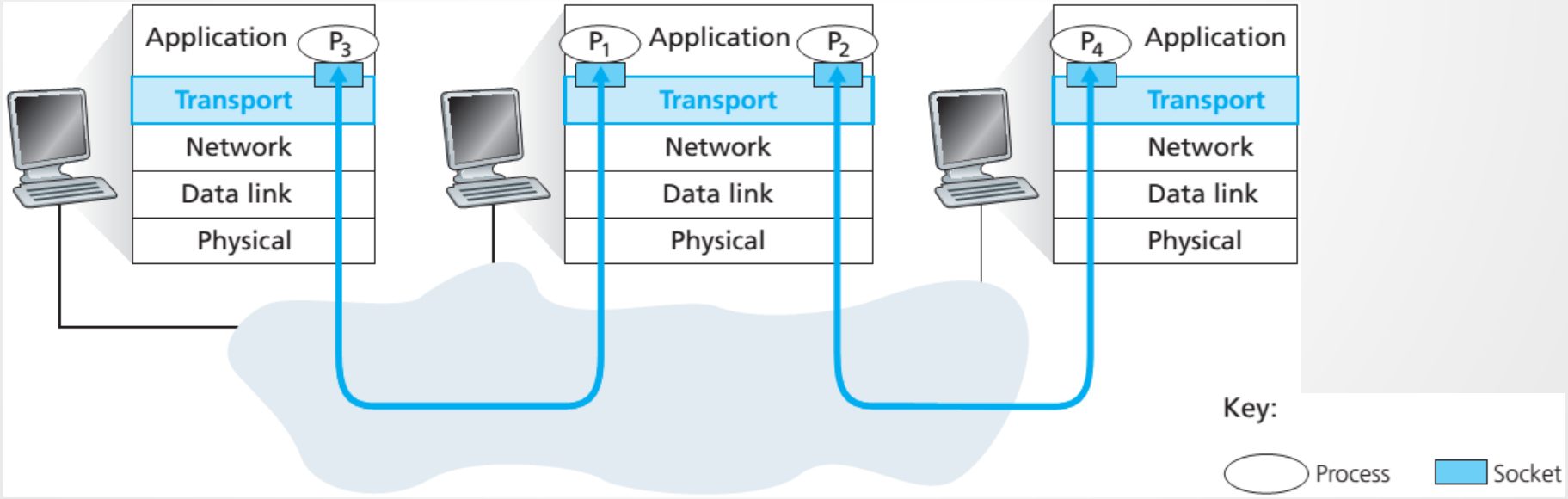
Multiplexado/demultiplexado

Demultiplexado en el host receptor:

Entregar los segmentos recibidos al socket correcto

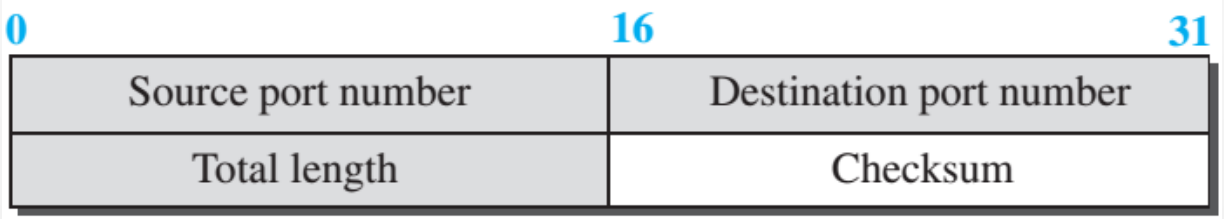
Multiplexado en el host emisor:

Reunir datos de múltiples sockets, etiquetando los datos con un encabezado (usado luego para el demultiplexado)



Como funciona el demultiplexado

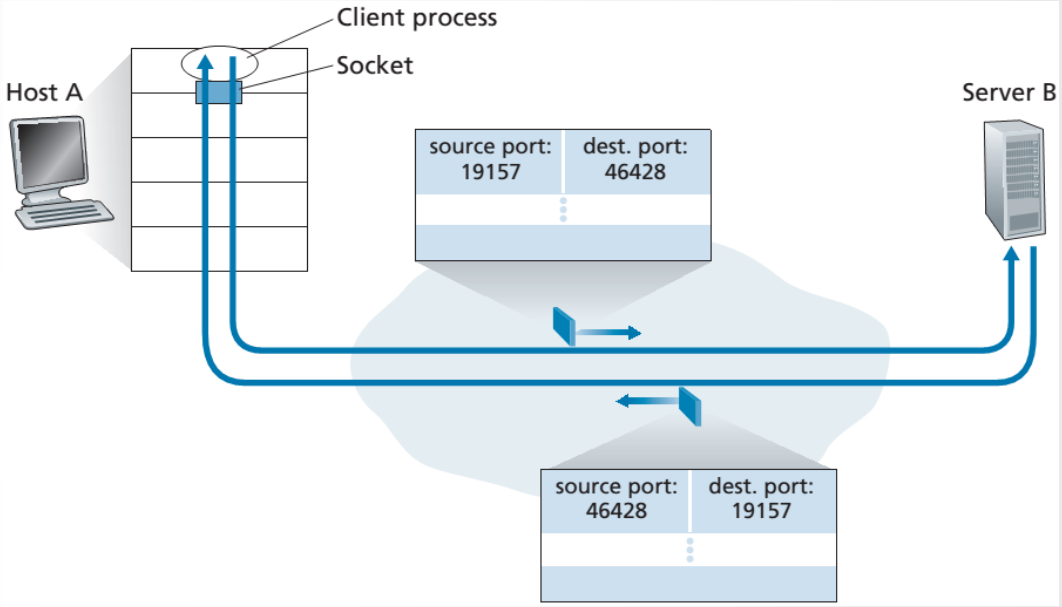
- Un host recibe datagramas IP
  - Cada datagrama tiene una dirección IP de origen y una dirección IP de destino
  - Cada datagrama transporta 1 segmento de la capa de transporte
  - Cada segmento tiene un número de puerto de origen y de destino
- El host usa las direcciones IP y los números de puerto para dirigir los segmentos a los sockets apropiados



Demultiplexado no orientado a la conexión

- Cuando un host recibe un segmento UDP:
  - Verifica el número de puerto destino en el segmento
  - Dirige el segmento UDP al socket con dicho número de puerto
- Datagramas IP con direcciones IP de origen y/o números de puerto origen diferentes se direccionan al mismo puerto, si tienen la dirección IP y puerto destino iguales
- Un socket UDP se identifica por la tupla :

  (Dirección IP destino, Número de puerto destino)

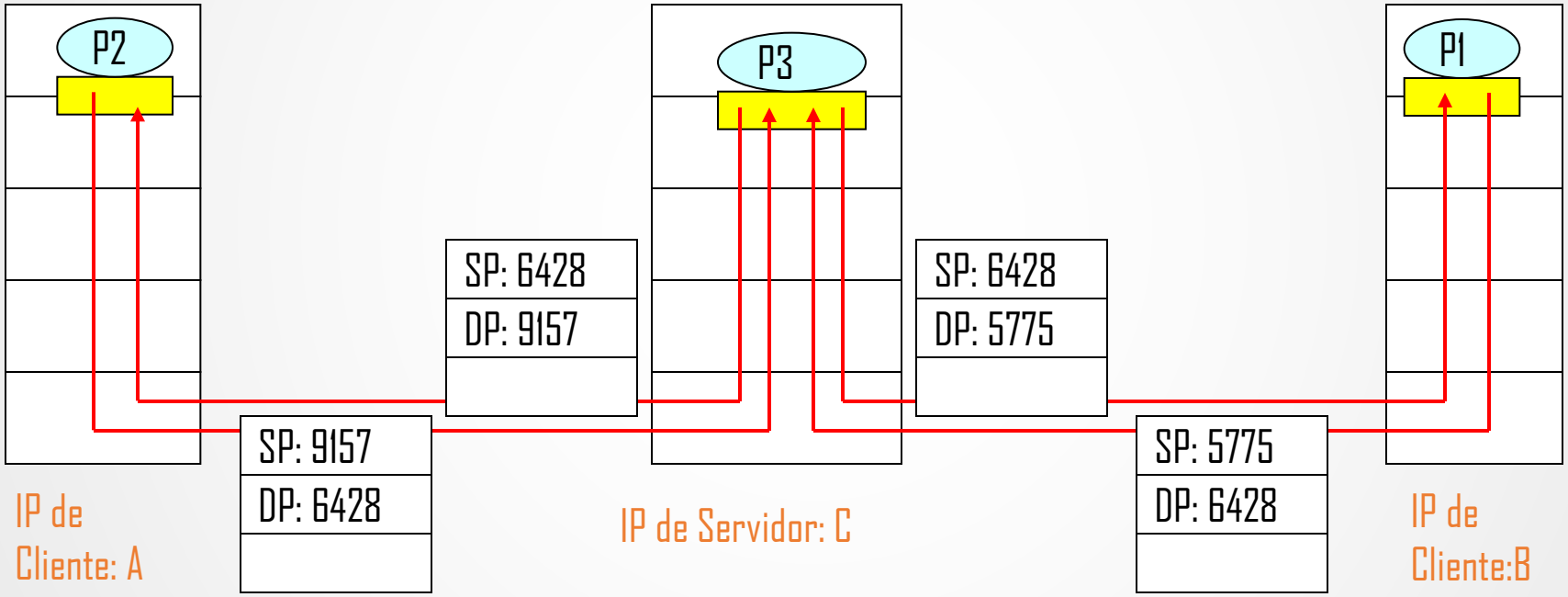


- Crear sockets con números de puerto:

```
DatagramSocket mySocket1 = new DatagramSocket(12534);
```

Demultiplexado no orientado a la conexión

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



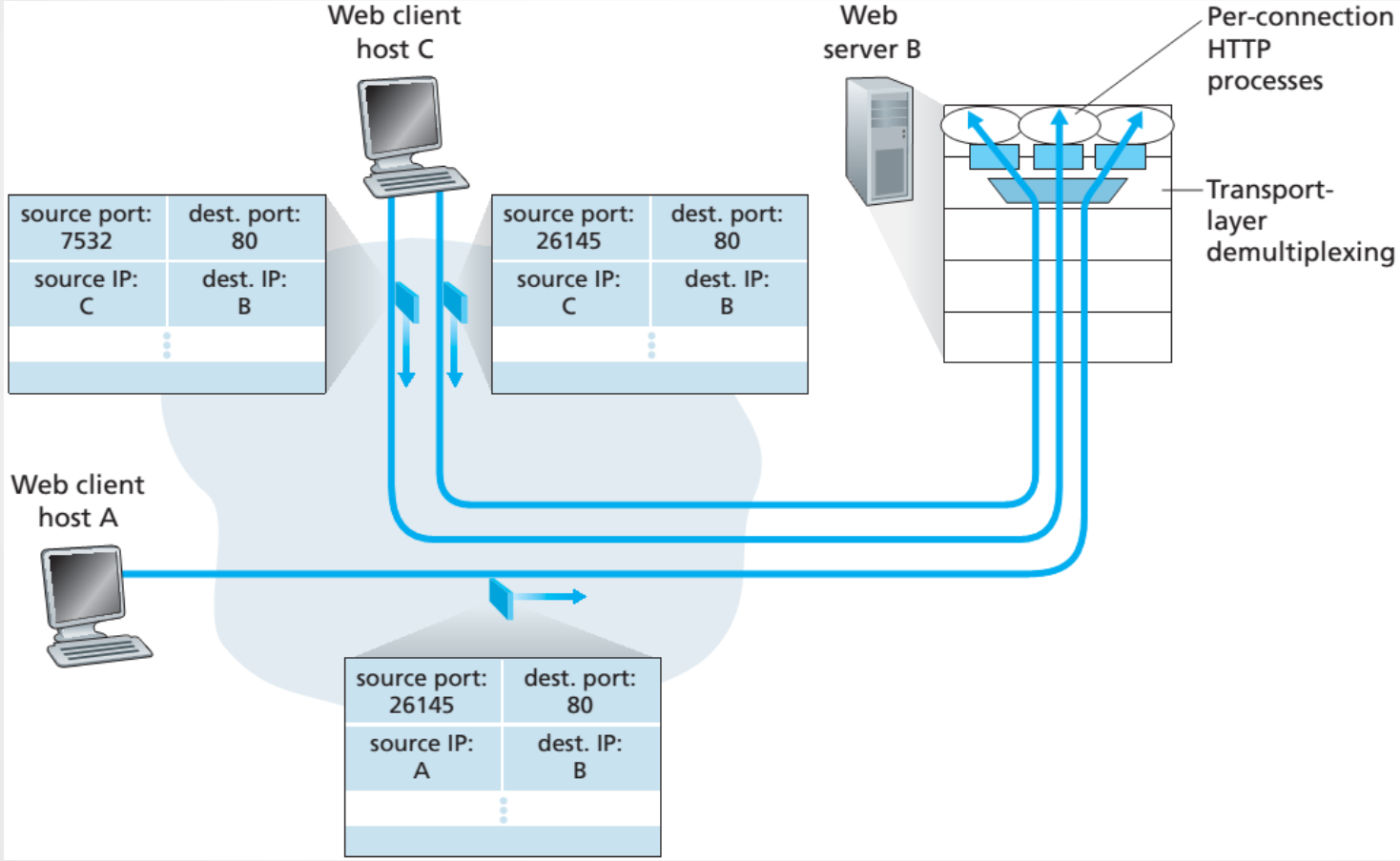
SP proporciona la “dirección de retorno”

Demultiplexado orientado a la conexión

- Un socket TCP se identifica por la tupla:
  - Dirección IP de origen
  - Número de puerto origen
  - Dirección IP de destino
  - Número de puerto destino
- El host receptor usa los 4 valores para dirigir el segmento al socket apropiado

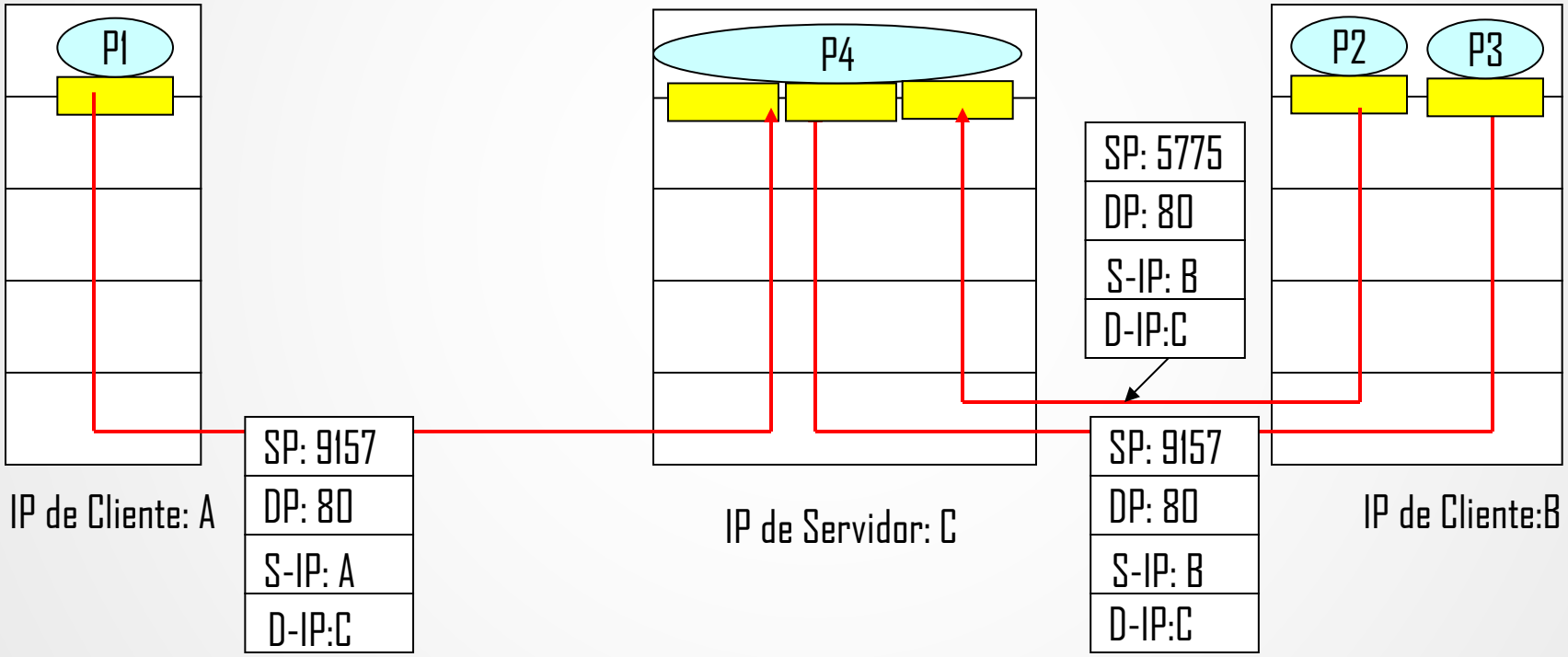
- El host servidor puede admitir muchos sockets TCP simultáneos:
  - Cada socket se identifica mediante su propia tupla
- Los servidores Web tiene **diferentes sockets** para cada conexión de cliente
  - HTTP no persistente tendrá sockets diferentes para cada consulta

Demultiplexado orientado a la conexión



Demultiplexado orientado a la conexión:

- Ejemplo de servidor web multihilo



Multiplexado y demultiplexado TPC/IP