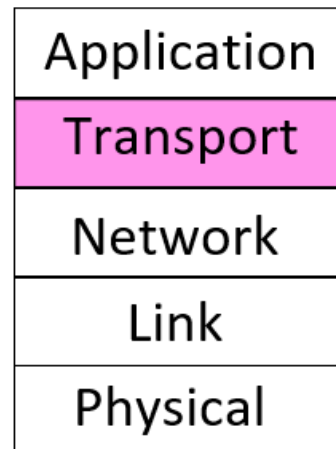


Capítulo 3

Capa de Transporte Direccionamiento



Direccionamiento

- **Problema:** El **direccionamiento** explícito de los destinos.
 - ¿Cómo hacer para que un proceso servidor *adecuado atiende* a las necesidades de una máquina cliente?
 - Consideraremos los siguientes casos:
 - El cliente podría no saber cuál proceso servidor es el adecuado para un servicio particular.
 - El cliente podría saber cuál proceso servidor es adecuado, pero dicho proceso servidor no atiende porque está inactivo.

Direccionamiento

- **Problema:** El cliente necesita un tipo de servicio.
 - Sin embargo no sabe cuál proceso servidor es adecuado para contactar.
 - Asumir que los procesos servidores están activos.
 - **¿Cómo hacer que el cliente se entere de un proceso servidor para un tipo de servicio?**

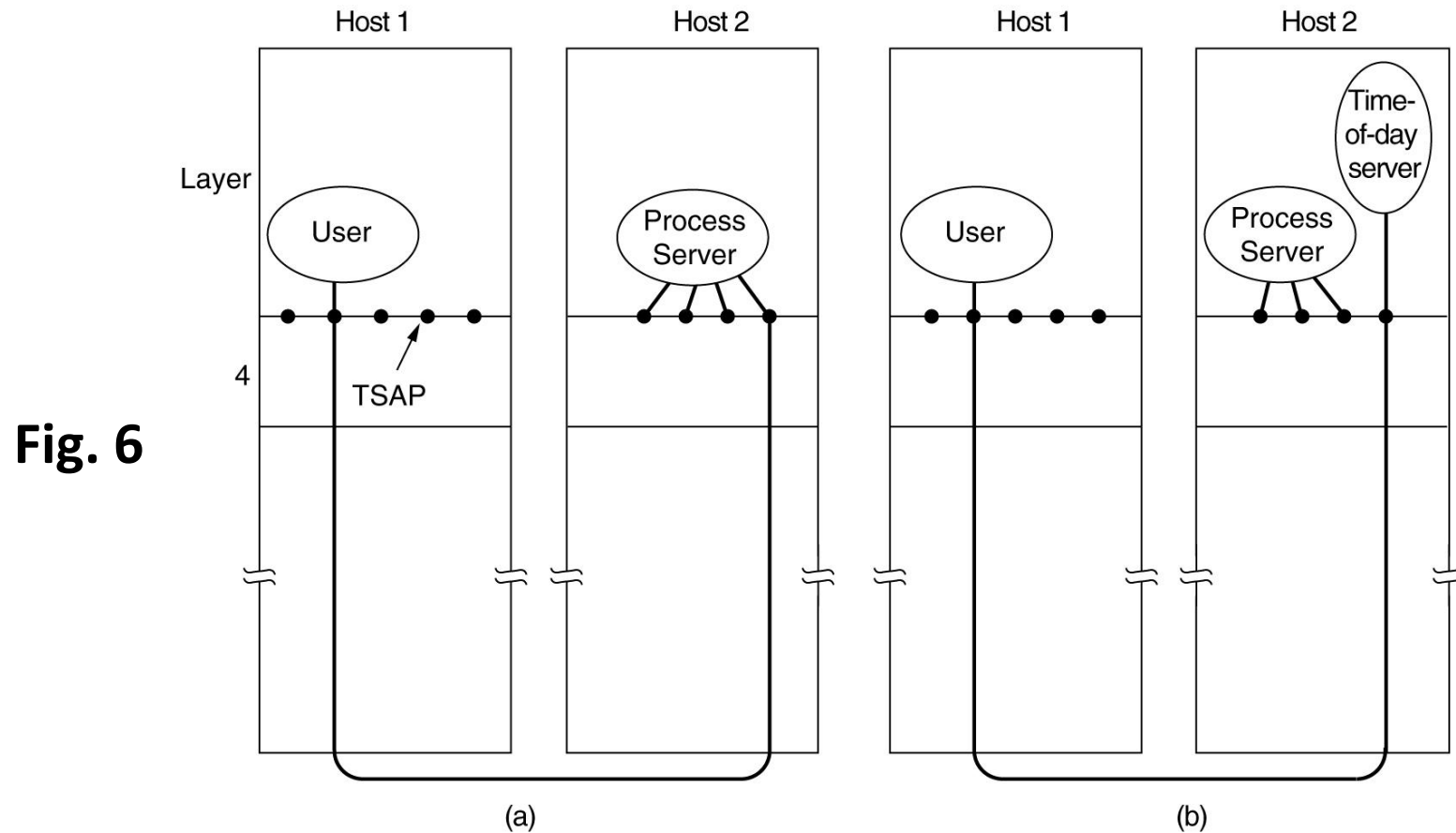
Direccionamiento

- **Solución:** Existe un proceso especial llamado **servidor de directorio** que para cada tipo de servicio sabe cuáles son los puertos de los servidores que prestan ese tipo de servicio.
 - Pasos seguidos:
 1. El usuario establece una conexión con el servidor de directorio (que escucha en un puerto bien conocido).
 2. El usuario envía un mensaje especificando el nombre del servicio.
 3. El servidor de directorio le devuelve la dirección puerto.
 4. El usuario libera la conexión con el servidor de directorio y establece una nueva con el servicio deseado.
- **¿Cómo se hace cuando se crea un servicio nuevo?**
 - El servicio nuevo debe registrarse en el servidor de directorio, dando su nombre de servicio como la dirección de su puerto.
 - El servidor de directorio registra esta información en su base de datos.

Direccionamiento

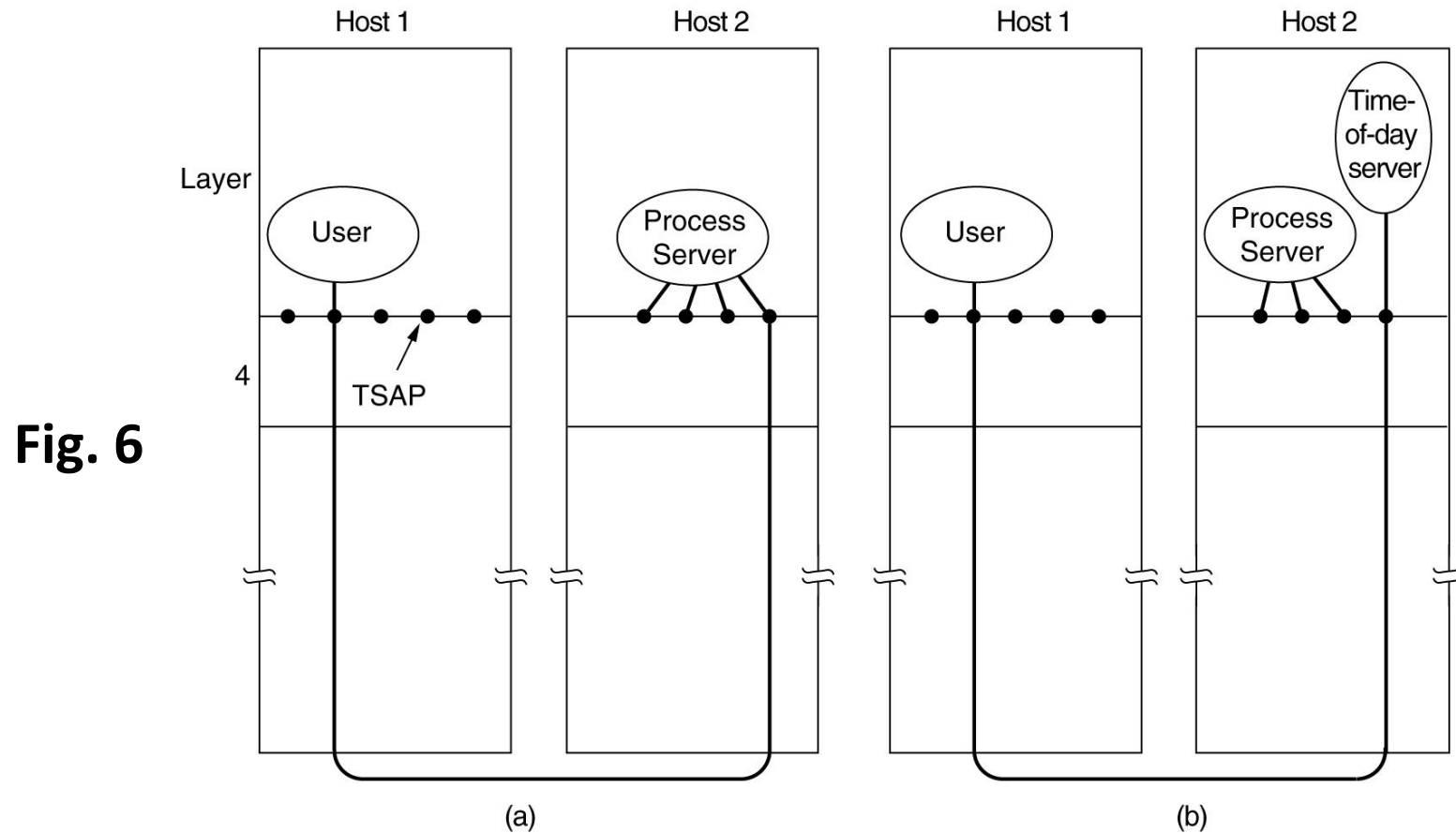
- **Problema:** ¿Cómo hacer para que un proceso servidor atienda a las necesidades de una máquina cliente?
 - El cliente sabe cuál es el proceso de servidor adecuado.
 - Pero ese proceso servidor **no** está activo.
- **Solución:** Usar servidor que ejecuta los servidores inactivos (**protocolo inicial de conexión**)
 - **Servidor de procesos** = intermediario de los servidores de menor uso.
 - En las figuras siguientes están los pasos de la solución.

Direccionamiento



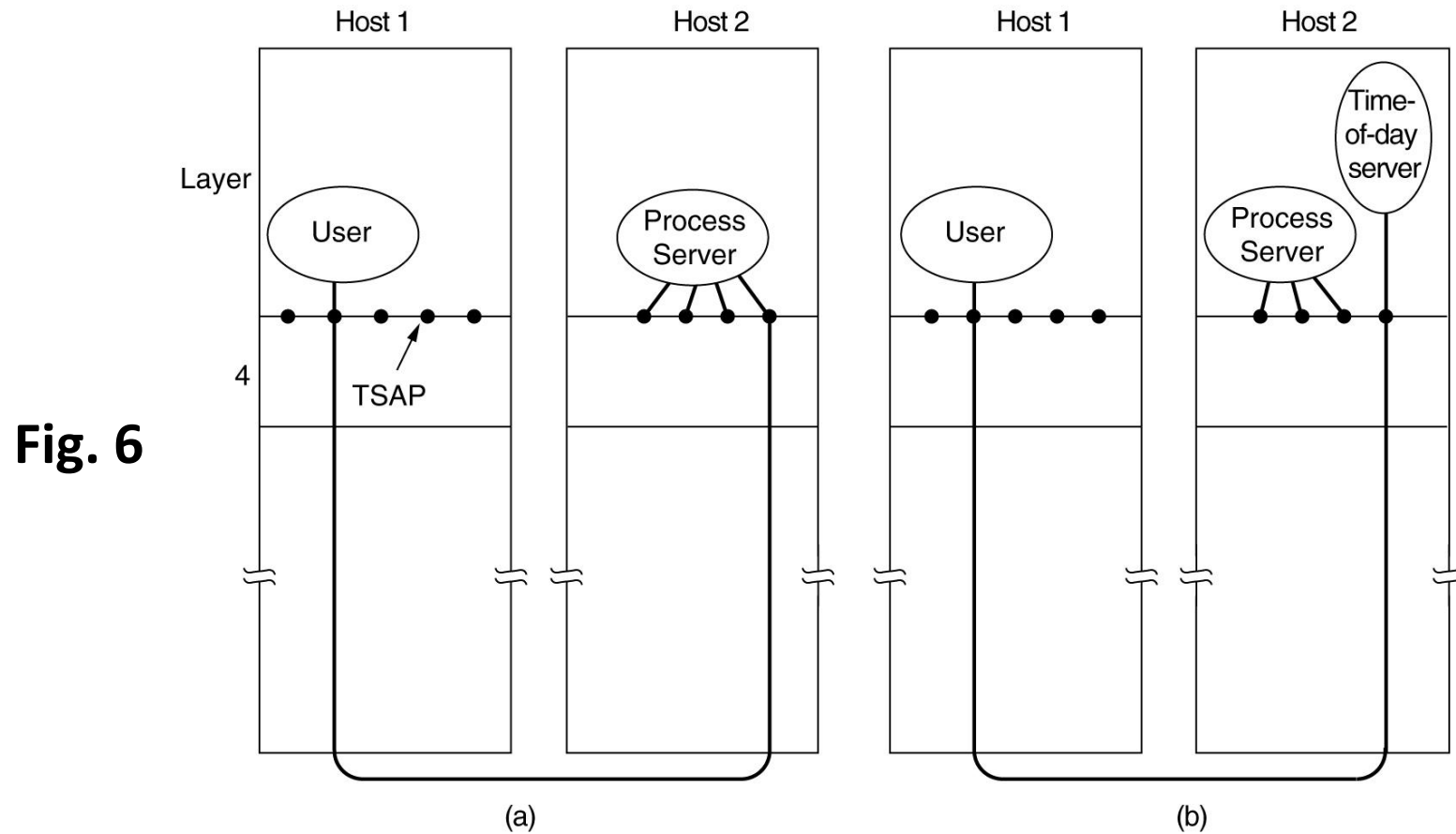
1. Escucha en un grupo de puertos al mismo tiempo esperando una solicitud de conexión
2. Un usuario emite una solicitud CONNECT, especificando el puerto del servicio que desea.

Direccionamiento



3. Si no hay ningún servidor esperándolo, consigue una conexión al servidor de procesos.
4. El servidor de procesos genera el servidor solicitado, permitiéndole **heredar la conexión** con el usuario existente.

Direccionamiento



5. El nuevo servidor hace el trabajo requerido y el servidor de procesos retorna a escuchar solicitudes nuevas.

Direccionamiento

- **Ejercicio:** Responder:
 - ¿Cuál es la diferencia entre la solución del *servidor de procesos* y la solución del *servidor de directorio*?
 - ¿Cuándo haría falta combinarlas?

Direccionamiento en TCP

- **Puertos bien conocidos**
 - N° puertos bien conocidos < 1024
 - Tabla de puertos bien conocidos.
 - **Demonios** = procesos servidores que atienden en un puerto
 - P. ej. que el *demonio FTP* se conecte a sí mismo al puerto 21 en el tiempo de arranque.

Port	Protocol	Use
20, 21	FTP	File transfer
22	SSH	Remote login, replacement for Telnet
25	SMTP	Email
80	HTTP	World Wide Web
110	POP-3	Remote email access
143	IMAP	Remote email access
443	HTTPS	Secure Web (HTTP over SSL/TLS)
543	RTSP	Media player control
631	IPP	Printer sharing

Direccionamiento en TCP

- **Problema:** Se podría llenar la memoria con demonios que están inactivos la mayor parte del tiempo.
- **Solución:** Un solo demonio llamado **inetd** (**demonio de internet**), escucha un conjunto de puertos al mismo tiempo y espera por un pedido de conexión.
 - Usuarios potenciales de un servicio comienzan a hacer **pedido CONNECT** especificando el puerto del servicio que quieren.
 - Si no hay ningún servidor esperando por ellos, *inetd* bifurca un nuevo proceso y ejecuta el demonio apropiado en él, y ese demonio maneja la solicitud.

Direccionamiento en TCP

- Inetd aprende qué puertos va a usar de un **archivo de configuración**.
 - Los demonios asociados a los puertos de este archivo **solo** están activos si hay trabajo para hacer.
- Se puede tener **demonios permanentes** en los puertos más ocupados e inetd en los demás.
 - Esto lo fija el administrador de sistema.