Arquitecturas cliente/servidor

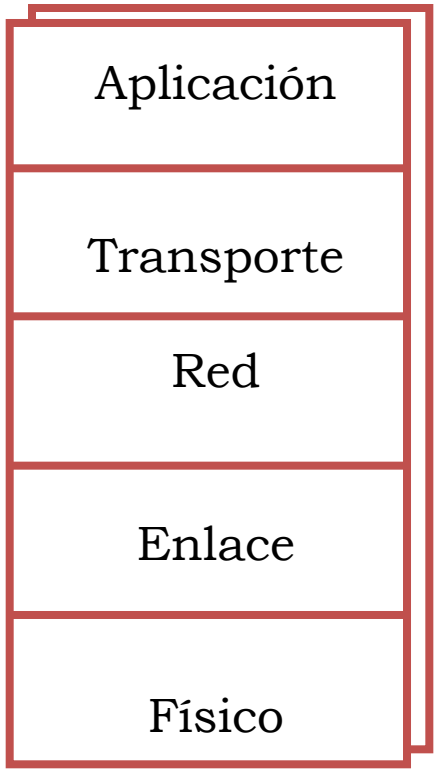
Conceptos básicos

Conceptos básicos

1. Definición de puerto
2. Sockets
3. Conceptos cliente/servidor
4. Definición de Stream
5. Concurrencia, multiprogramación y multitarea
6. Servidores iterativos y concurrentes
7. Estándares

Introducción
Pila de protocolos en Internet

- aplicación: compuesto por las aplicaciones de red
  - FTP, SMTP, STTP
- transporte: transferencia de datos host-host
  - TCP, UDP
- red: ruteo de datagramas desde fuente a destino
  - IP, protocolos de ruteo
- enlace: transferencia de datos entre elementos vecinos en la red
  - P2P, Ethernet
- físico: transporte de bits p.e. “RJ-45”



¿QUÉ ES UN PUERTO?

Puerto

- Un puerto en la capa de transporte esta representado por un número de 16 bits, que es utilizado para identificar los puntos finales de la conexión, en las cabeceras UDP o TCP.
- Los números de puerto oscilan entre 0 y 65,535.
- La pila de protocolos de red añade los puertos como una abstracción para la red.
- Son canales que utiliza el subsistema de red para redireccionar la información al programa apropiado.

Clasificación de puertos

- Puertos bien conocidos [0, 1023]
  - Se utilizan para servicios de red bien conocidos (FTP, HTTP, Telnet, DNS, …)

- Puertos registrados
  - Oscilan entre 1024 a 49151 y pueden ser usados de manera temporal por los clientes, pero también pueden representar servicios registrados por un tercero.

- Puertos dinámicos o privados
  - Oscilan entre 49152 y 65535, pueden también ser usados por el cliente, pero se utilizan menos frecuentemente.

*** Controlados por la IANA (Internet Assigned Numbers Authority) http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.txt**

Socket

- Un número de puerto y una dirección de red.
- Un par de sockets, uno en cada host, forma una conexión única.
- Es un punto final de un enlace de comunicación de dos vías entre dos programas que se ejecutan a través de la red.
- El cliente y el servidor deben ponerse de acuerdo sobre el protocolo que utilizarán.

Recordando… Un proceso…

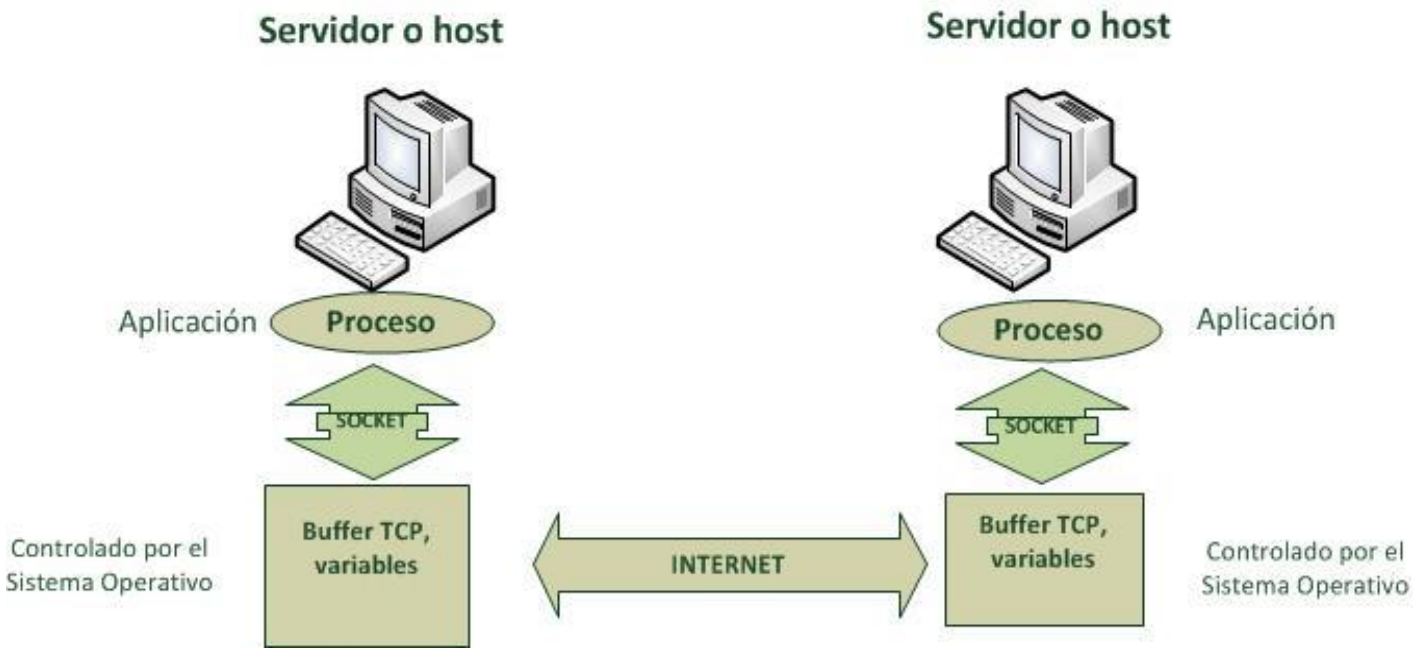
… Es un programa en ejecución en una computadora.

- Dentro de la misma computadora dos procesos se comunican generalmente usando comunicación entre procesos (definida por OS).

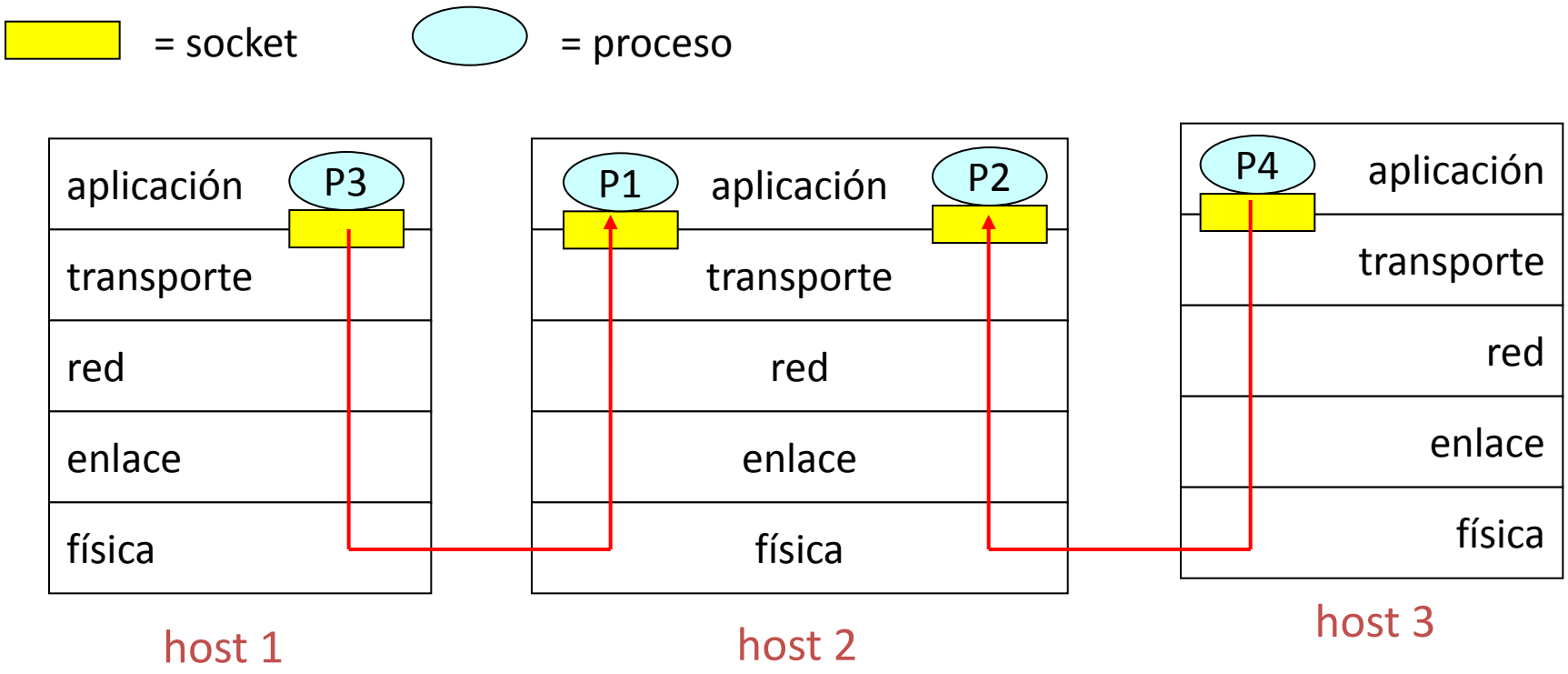
- Procesos en diferentes hosts se comunican vía intercambio de mensajes.

Socket

- Los procesos envían/reciben mensajes a/desde sus socket
  - Proceso transmisor transmite mensajes por el puerto
  - Proceso transmisor confía en la infraestructura de transporte al otro lado del puerto la cual lleva los mensajes al socket en el proceso receptor



Relación proceso /socket



Clasificación de Sockets

- Orientado a Conexión

  – Establece un camino virtual entre servidor y cliente, fiable, sin pérdidas de información ni duplicados, la información llega en el mismo orden que se envía.

  – El cliente abre una sesión en el servidor y este guarda un estado del cliente.

- No orientado a conexión
  - Envío de datagramas de tamaño fijo. No es fiable, puede haber pérdidas de información y duplicados, y la información puede llegar en distinto orden del que se envía.
  - No se guarda ningún estado del cliente en el servidor, por ello, es más tolerante a fallos del sistema.

Concepto Cliente / Servidor

- La comunicación de dos hosts se realiza generalmente mediante la filosofía Cliente/Servidor.
- El usuario cliente obtiene servicios de la máquina remota proveedora de un servicio (servidor).
- El servidor proporciona un puerto de comunicación por el cuál se deben de conectar todos los clientes que deseen obtener dicho servicio.
- Se establece un socket en la máquina local (cliente) y otro en la máquina remota (servidor) , y se comunican entre sí por el puerto proporcionado, así se establece la vía de comunicación entre dos hosts interconectados por una red.

Proceso Cliente

1. Abre el canal de comunicaciones para conectarse a la dirección de red atendida por el servidor

2. Enviar al servidor un mensaje de petición de servicio y esperar hasta recibir respuesta

3. Cerrar el canal de comunicación y terminar la ejecución del proceso.

Proceso Servidor

1. Abre el canal de comunicación e informa a la red de la dirección a la que responderá como de la disposición para aceptar peticiones de servicio.

2. Espera a que el Cliente realice una petición de servicio en la dirección que el tiene declarada.

3. Cuando recibe una petición de servicio, atiende al Cliente.

4. La conexión es cerrada.

Procesos que se comunican

Proceso Cliente: proceso que inicia la comunicación

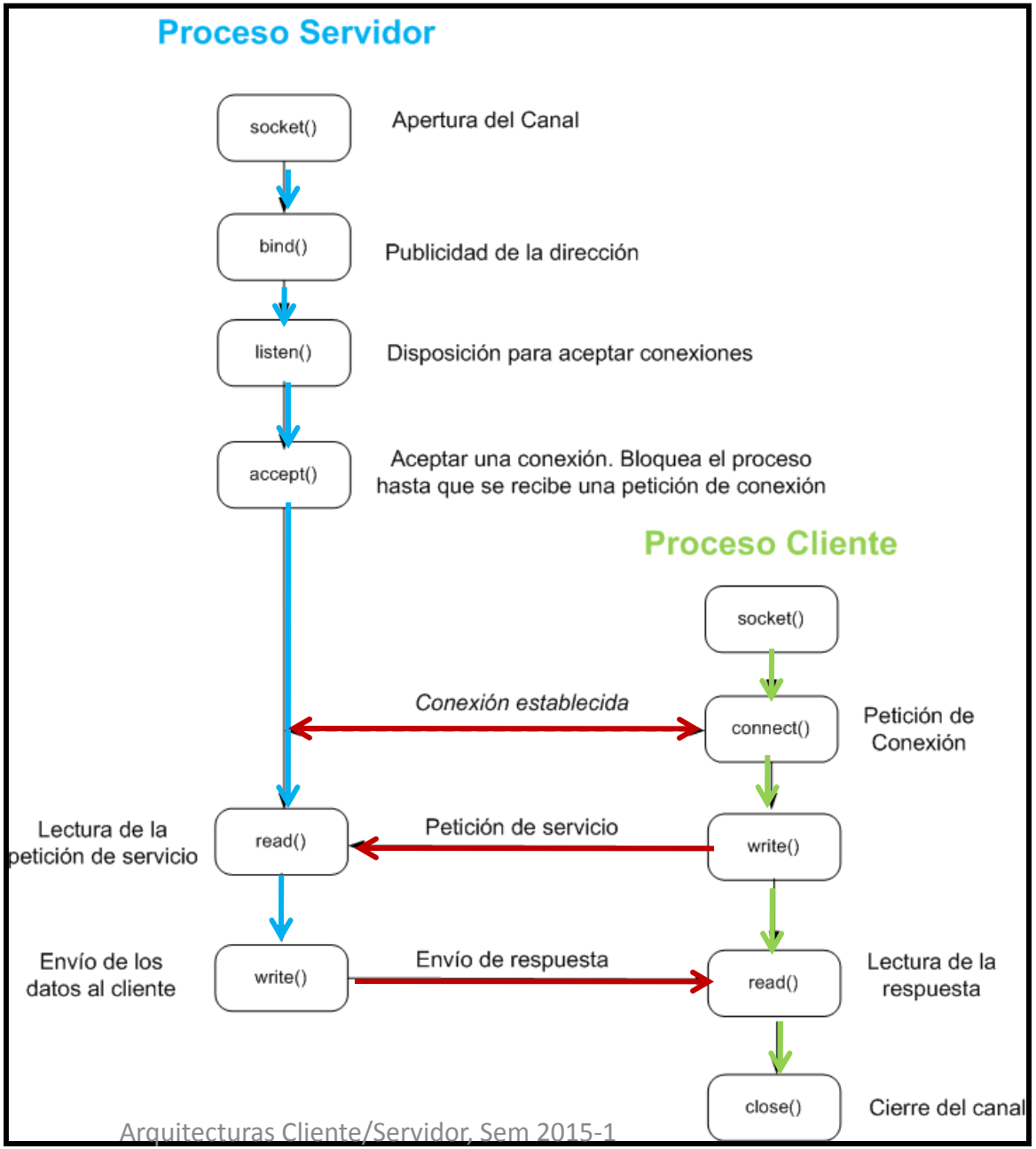
Proceso servidor: proceso que espera por ser contactado

Sockets en UNIX

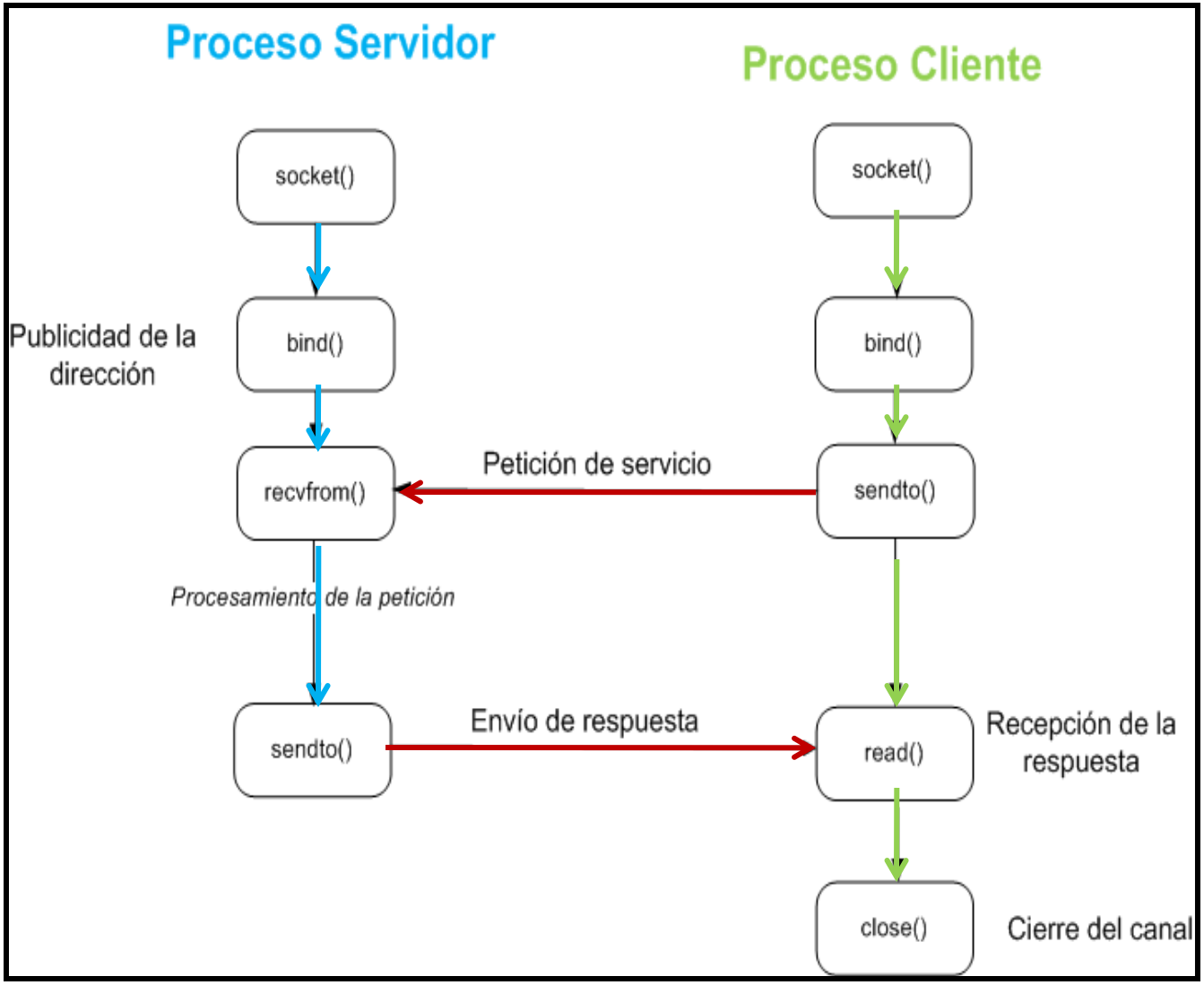
API para sockets

- Fue introducida en BSD4.1 UNIX, 1981
- El socket es explícitamente creado, usado, y liberado por las aplicaciones
- Sigue el modelo cliente/servidor
- También conocida como API de Berkeley, definida en lenguaje C

Secuencia de llamadas para una comunicación cliente/servidor Orientada a Conexión



Secuencia de llamadas para una comunicación cliente/servidor No orientada a Conexión



Otras API's

- Sockets en DOS
  - Una de ellas es TCP/IP for DOS Toolkit :
    - http://www.drdos.com/dosdoc/Tcpip/dos_api/

- Sockets en MAC
  - Este módulo proporciona una interfaz con el gestor TCP/IP de Macintosh, MacTCP. Hay un módulo de acompañamiento, que proporciona una interfaz con el servidor de nombres (permitiendo la traducción nombres de nodo a direcciones IP)
    - http://pyspanishdoc.sourceforge.net/mac/module-mactcp.html