



VNiVERSiDAD  
D SALAMANCA

<http://informatica.usal.es/gii>

## Extracto de los temas transporte y aplicación

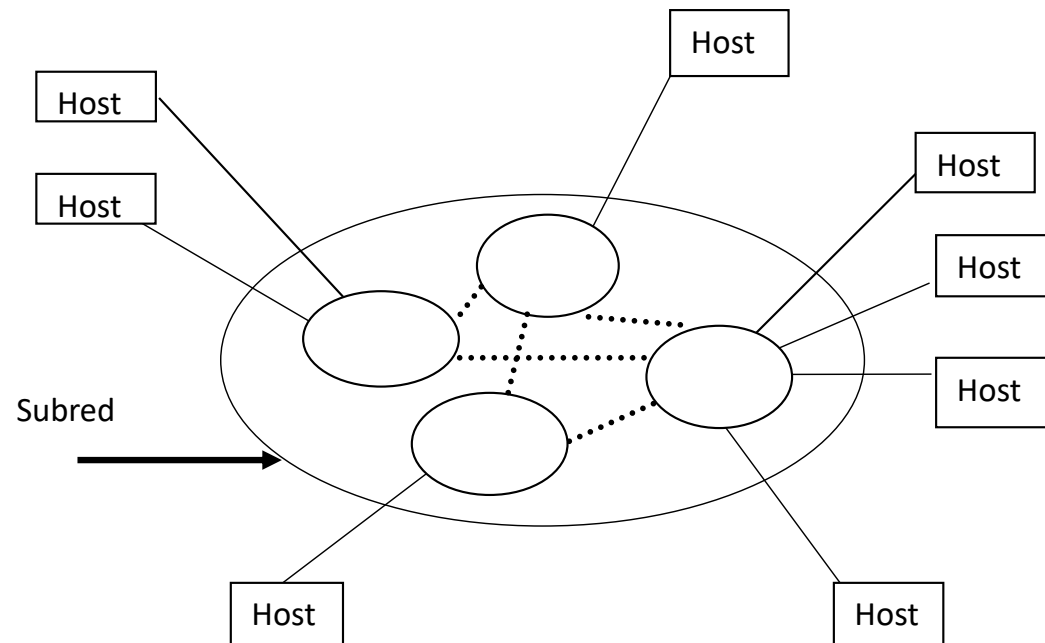
Introducción al nivel de transporte  
Nivel de transporte en Internet: TCP y UDP  
El modelo cliente-servidor  
Los servicios de Internet  
Programación con sockets



VNiVERSiDAD  
D SALAMANCA

# Introducción

- **Función:** proporcionar un transporte de los datos extremo a extremo (host a host), independientemente de la red o redes en uso (optimizar los servicios del nivel de red)
- El Nivel de transporte busca proporcionar un servicio eficiente y fiable a sus usuarios, los procesos de la capa de aplicación





# El nivel de Transporte en Internet I

- Internet ofrece los dos tipos de servicios en el nivel de transporte
  - Con conexión (TCP) y sin conexión (UDP)
- Direccionamiento - **Números de puerto**
  - Los puntos finales de una comunicación son dos procesos de usuario, uno en cada sistema
  - Los dos sistemas pueden estar en redes diferentes, conectadas por uno o más routers
  - Esto requiere tres niveles de direccionabilidad:
    - Red y host (Dirección IP)
    - Identificador único para cada comunicación en un host
      - A este identificador se le denomina **Número de Puerto**
      - Tanto TCP como UDP usan un entero de 16 bits para esta identificación





# El nivel de Transporte en Internet II

- Los numero de puerto en el rango 1 a 255 están reservados para los **puertos bien conocidos** ([RFC 7100](#))
  - Cada servicio de Internet tiene reservado un número de puerto
  - Son asignados por la organización llamada *Internet Assigned Number Authority*
  - Algunos sistema operativos reservan puertos adicionales para programas privilegiados (4.3 BSD reserva los puertos 1-1023 para procesos de superusuario)
  - El resto son para puertos del usuario que se denominan **puertos efímeros**
  - Por ejemplo
    - FTP es un servicio bien conocido que se proporciona en TCP en el puerto 21





# El nivel de Transporte: TCP

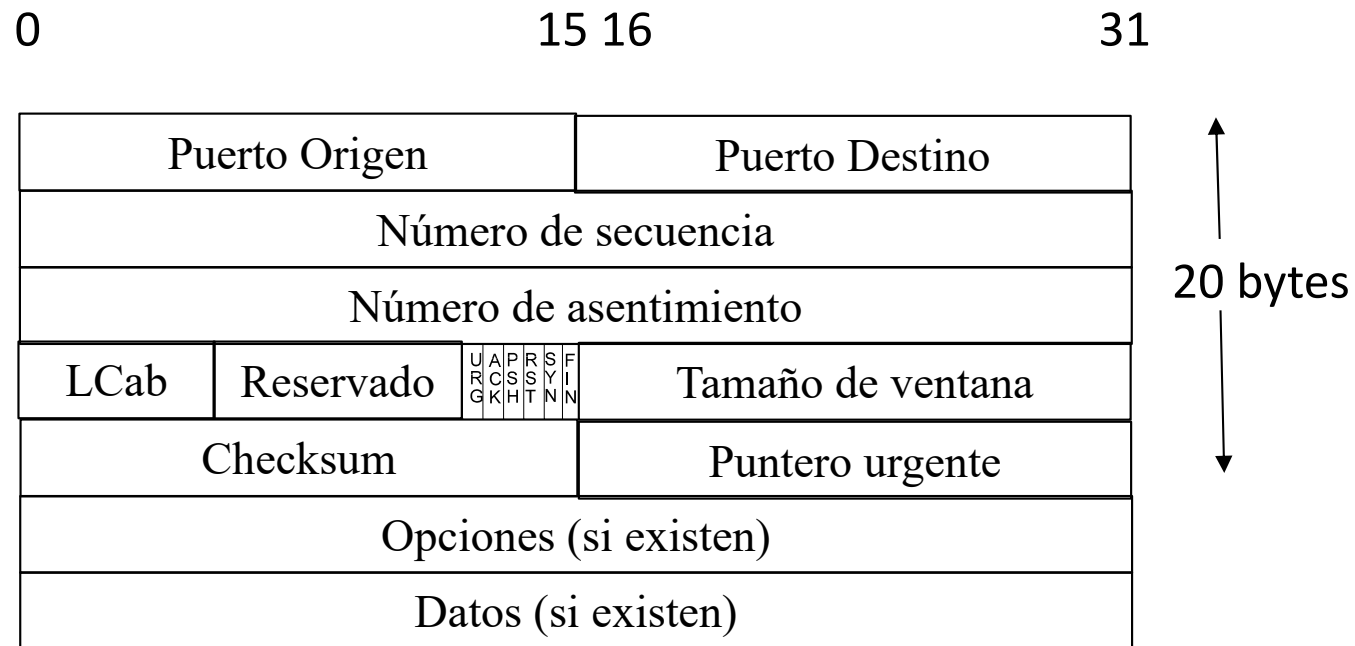
- **TCP - *Transmisión control protocol***

- Es un protocolo orientado a conexión y fiable
- TSAP = Conexión (Dirección IP origen, nº puerto), (Dirección IP destino, nº puerto)
- Un puerto TCP se pueden compartir entre varias conexiones
- Es responsable de
  - Establecer, mantener y terminar una conexión entre procesos
  - Permitir múltiples conexiones entre procesos distintos en un mismo *host*
  - Fraccionar el mensaje en datagramas, reensamblarlo en destino (ordenando los datagramas si no han llegado en orden)
  - Garantizar que la transmisión sea segura extremo a extremo (*checksum*, asentimientos, retransmisiones, *timeouts*, ...)
  - Proporcionar un control de flujo extremo a extremo utilizando la técnica de ventana deslizante
- En la [RFC 793](#) se describe el estándar TCP





# La cabecera TCP





# El nivel de Transporte: UDP

- UDP: "*User Datagram Protocol*"
  - Proporciona un servicio de transporte no fiable, sin conexión
  - TSAP = (Dirección IP, nº Puerto)
  - No proporciona asentimiento de los datagramas recibidos al receptor. Al igual que IP los paquetes pueden llegar desordenados, perderse o duplicarse
  - La única facilidad que incorpora UDP es la asignación y gestión de los números de puerto para identificar aplicaciones individuales que se ejecutan en un host
  - No fragmenta
  - UDP es más rápido que TCP debido a la sobrecarga en las funciones que realiza TCP
  - Se describe en la [RFC 768](#)

|               |                |
|---------------|----------------|
| Puerto origen | Puerto destino |
| Longitud      | Checksum       |
| Datos         |                |
|               |                |



# El modelo Cliente-Servidor I

- Modelo estándar para desarrollo de aplicaciones en red
  - Un servidor es un proceso que esta esperando peticiones de los procesos clientes para los que tiene que hacer algo. El esquema típicamente es de la siguiente forma:
    - El proceso servidor se arranca en un determinado sistema. El proceso se inicializa y queda en espera de que contacten con él los procesos clientes
    - Comienza un proceso cliente, en la misma computadora o en otra distinta que está conectada con el sistema servidor mediante una red
    - El proceso cliente envía una petición a través de la red al servidor
      - Algunos ejemplos son:
        - » Devuelve la fecha y hora al cliente
        - » Imprime un fichero por el cliente
        - » Lee o escribe un fichero en el sistema del servidor por el cliente
        - » Permite al cliente entrar en el sistema del servidor
        - » Ejecuta un comando por el cliente en el sistema del servidor
    - Cuando el proceso servidor termina de proporcionar el servicio el servidor vuelve a la espera de nuevos clientes
  - Ventajas:
    - Esquema válido de direccionamiento
    - Minimiza el tráfico en la red







## El modelo Cliente-Servidor II

- Cuando un proceso cliente desea contactar con un servidor, el cliente debe saber la forma de identificar al servidor que desea
  - Dirección IP (red y host) donde se ejecuta el servidor
  - Identificar a ese servidor en concreto
    - Para resolver esto, tanto TCP y UDP utilizan un grupo de **puertos bien conocidos**. Por ejemplo FTP esta en el puerto 21
  - El proceso cliente también debe obtener un número de puerto local que irá en la cabecera del protocolo de transporte para que el proceso servidor pueda enviarle las respuestas
    - Este número de puerto debe ser único en el sistema y recibe el nombre de **puerto efímero** (de corta vida)
- En toda comunicación es necesaria la 5-tupla que definimos como **Asociación**. Es la 5-tupla que especifica completamente dos procesos que tienen una conexión:
  - {protocolo,dirección-local,puerto-local,dirección-remota,puerto-remoto}
    - Protocolo (TCP o UDP)
    - La dirección IP del host local (32 bits)
    - El número de puerto local (16 bits)
    - La dirección IP del host remoto (32 bits)
    - El número de puerto remoto (16 bits)





# El nivel de aplicación

- Se construye directamente sobre el nivel de transporte
  - En UNIX: /etc/services
  - <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>

| Servicio                          | Protocolo | Protocolo transporte | Nº puerto |
|-----------------------------------|-----------|----------------------|-----------|
| Transmisión de ficheros           | FTP       | TCP                  | 21        |
| Login remoto cifrado              | SSH       | TCP                  | 22        |
| Login remoto                      | TELNET    | TCP                  | 23        |
| Correo electrónico                | SMTP      | TCP                  | 25        |
| Web                               | HTTP      | TCP                  | 80        |
| Transmisión de ficheros sencilla  | TFTP      | UDP                  | 69        |
| Servidor de nombres de dominio    | DNS       | UDP                  | 53        |
| Configuración dinámica de equipos | DHCP      | UDP                  | 67/68     |





# Los sockets de Berkeley I

- Uno de los API (*Application Program Interface*) de comunicación para sistemas Unix
  - Es un interfaz muy usado por programadores para acceder a los servicios que proporcionan los protocolos de comunicación de la red
  - Los sockets proporcionan la interfaz entre el nivel de transporte y los niveles superiores
- Se suele decir que un socket es un mecanismo que proporciona un punto final para establecer una comunicación a través de la red
- Podríamos definir un **socket** como un punto de acceso a un servicio perfectamente identificado en la red, por el se mandan y aceptan paquetes de información





# Los sockets de Berkeley II

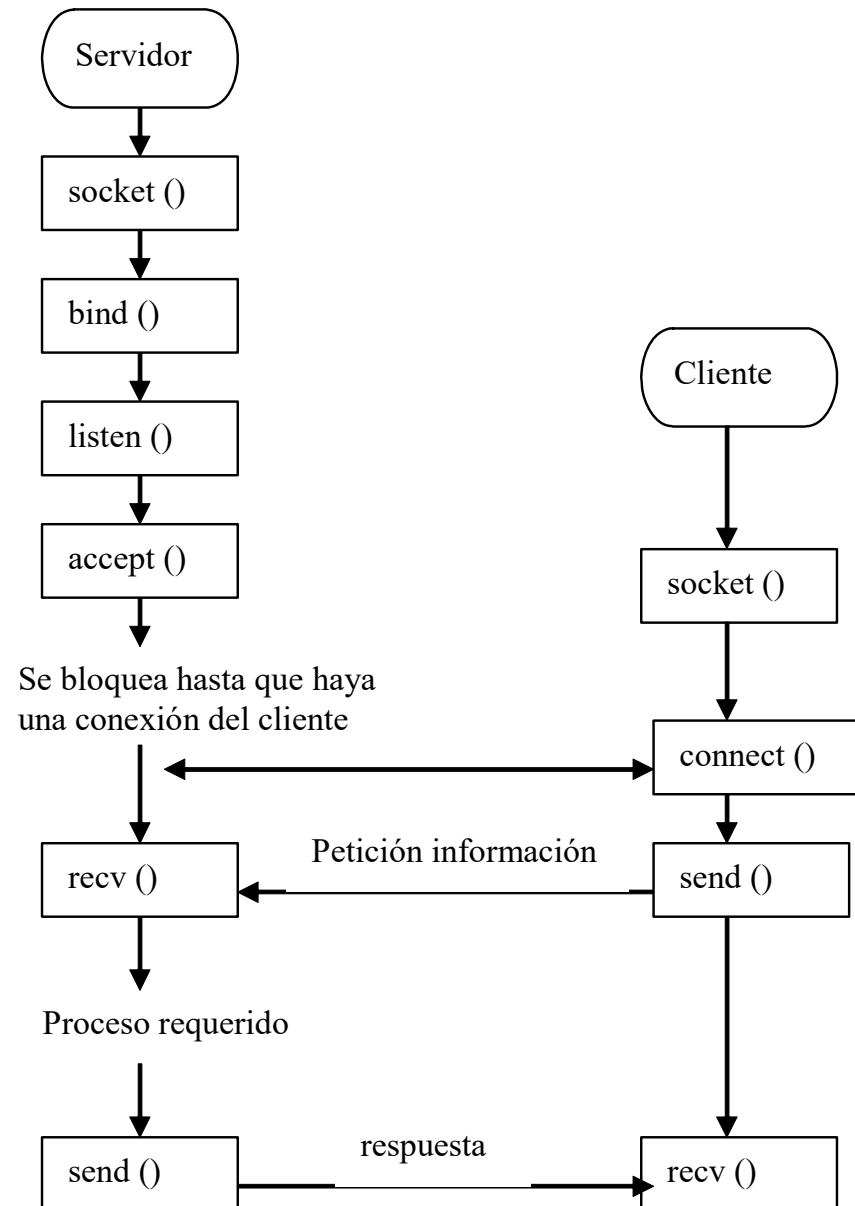
| Operación                           | Cliente                     | Servidor      |
|-------------------------------------|-----------------------------|---------------|
| Crear un socket                     | <i>socket</i>               | <i>socket</i> |
| Asociar dirección                   | <i>bind</i>                 | <i>bind</i>   |
| Especificar nº conexiones en espera |                             | <i>listen</i> |
| Solicitar conexión                  | <i>connect</i>              |               |
| Esperar y aceptar conexión          |                             | <i>accept</i> |
| Enviar información                  | <i>write, send, sendto</i>  | idem          |
| Recibir información                 | <i>read, recv, recvfrom</i> | idem          |
| Desconectar socket                  | <i>shutdown, close</i>      | idem          |





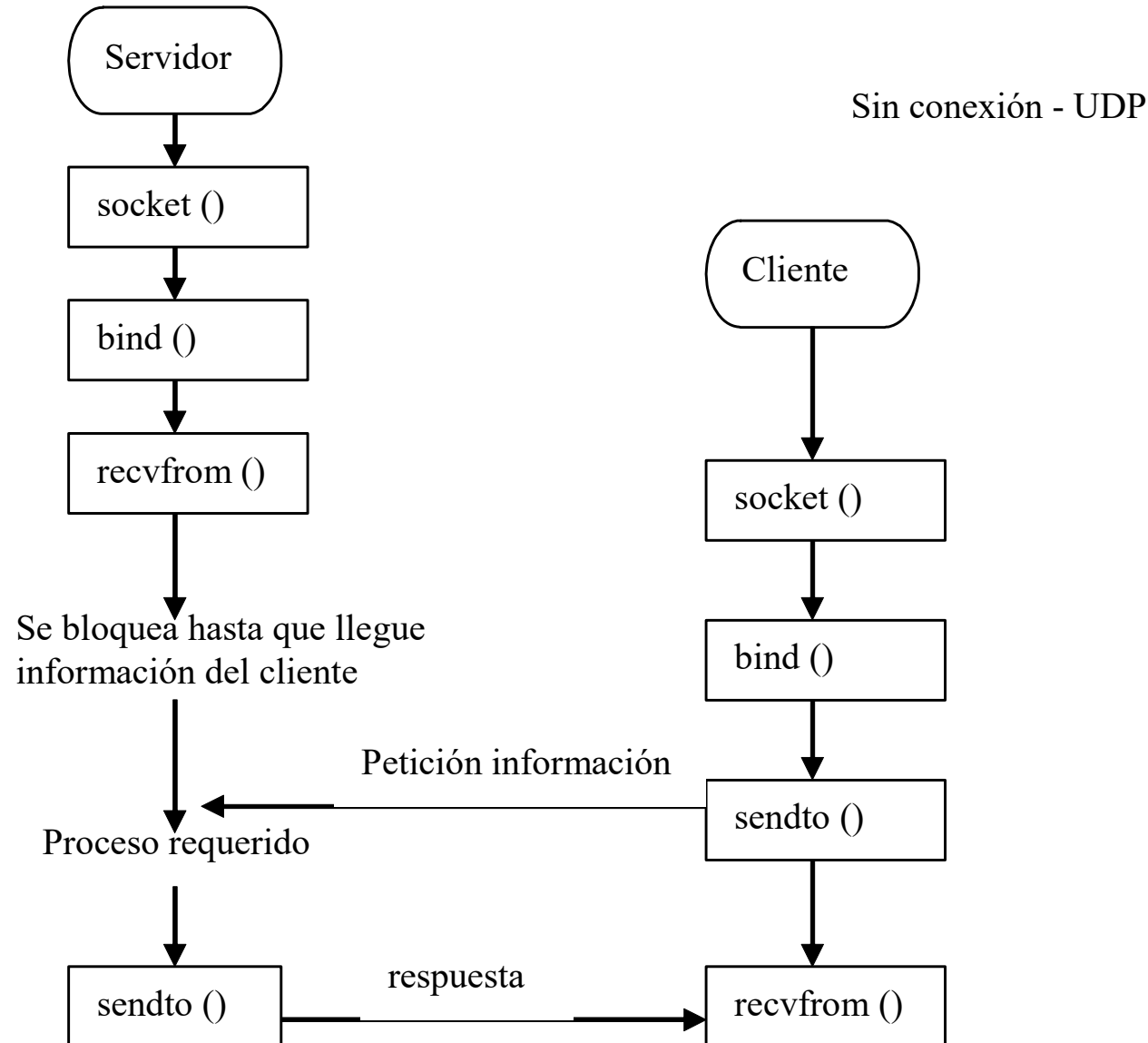
# Los sockets de Berkeley III

Con conexión - TCP





# Los sockets de Berkeley IV





# Ejemplo de la orden netstat -a

**A) Estamos en el host TEJO y la salida de la orden es la siguiente:**

Active Internet connections (including servers)

| Proto | Recv-Q | Send-Q | Local Address | Foreign Address | (state) |
|-------|--------|--------|---------------|-----------------|---------|
| tcp   | 0      | 0      | *.23456       | *.*             | LISTEN  |

**B) Instantes después en TEJO la salida de netstat es la siguiente:**

Active Internet connections (including servers)

| Proto | Recv-Q | Send-Q | Local Address      | Foreign Address    | (state)     |
|-------|--------|--------|--------------------|--------------------|-------------|
| tcp   | 0      | 0      | tejo.usal.es.23456 | tejo.usal.es.4854  | ESTABLISHED |
| tcp   | 0      | 0      | tejo.usal.es.4854  | tejo.usal.es.23456 | ESTABLISHED |
| tcp   | 0      | 0      | *.23456            | *.*                | LISTEN      |

**C) Momentos después en TEJO la salida es:**

Active Internet connections (including servers)

| Proto | Recv-Q | Send-Q | Local Address      | Foreign Address     | (state)     |
|-------|--------|--------|--------------------|---------------------|-------------|
| tcp   | 0      | 0      | tejo.usal.es.23456 | encina.usal.es.1184 | ESTABLISHED |
| tcp   | 0      | 0      | *.23456            | *.*                 | LISTEN      |

**y en ENCINA**

Active Internet connections (including servers)

| Proto | Recv-Q | Send-Q | Local Address       | Foreign Address    | (state)     |
|-------|--------|--------|---------------------|--------------------|-------------|
| tcp   | 0      | 0      | encina.usal.es.1184 | tejo.usal.es.23456 | ESTABLISHED |

