


**ARQUITECTURA DE REDES, SISTEMAS Y SERVICIOS**  
 Área de Ingeniería Telemática

---

# Servicios de Internet

---

Área de Ingeniería Telemática  
<http://www.tlm.unavarra.es>

Arquitectura de Redes, Sistemas y Servicios  
 3º Ingeniería de Telecomunicación

---

---

---

---

---

---

---

---


**ARQUITECTURA DE REDES, SISTEMAS Y SERVICIOS**  
 Área de Ingeniería Telemática

# Temario

1. Introducción
2. Arquitecturas, protocolos y estándares
3. Conmutación de paquetes
4. Conmutación de circuitos
5. Tecnologías
6. Control de acceso al medio en redes de área local
7. Servicios de Internet

1

---

---

---

---

---

---

---

---


**ARQUITECTURA DE REDES, SISTEMAS Y SERVICIOS**  
 Área de Ingeniería Telemática

# Temario

1. Introducción
2. Arquitecturas, protocolos y estándares
3. Conmutación de paquetes
4. Conmutación de circuitos
5. Tecnologías
6. Control de acceso al medio en redes de área local
7. **Servicios de Internet**
  - La Web
  - E-Mail.
  - FTP, Telnet
  - Otros
  - **Desarrollo de clientes y servidores**

2

---

---

---

---

---

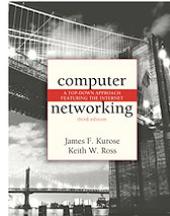
---

---

---

## Material

Del Capítulo 2 de  
Kurose & Ross,  
"Computer Networking a top-down approach  
featuring the Internet"  
Addison Wesley




---

---

---

---

---

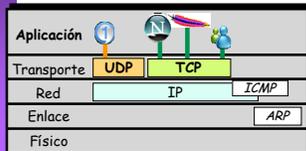
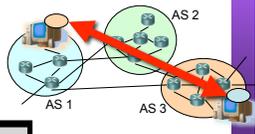
---

---

---

## Las aplicaciones

- Son software
- Diferentes máquinas y Sistemas Operativos
- Quienes se comunican son **procesos**



- Intercambian mensajes
- Emplean **Protocolos de nivel de aplicación**

---

---

---

---

---

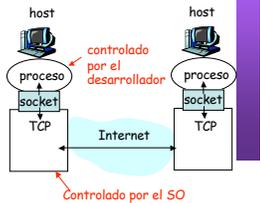
---

---

---

## Sockets

- Los procesos envían y reciben mensajes a través de un **socket**
- Delega en el nivel de transporte para que haga llegar los mensajes al otro socket
- Acceso a través de un **API**
- Puede escoger el protocolo de transporte
- Puede configurar algunos parámetros del mismo
- No controla cómo se comporta




---

---

---

---

---

---

---

---

**upna**  
UNIVERSIDAD POLITÉCNICA DE NAVARRA

ARQUITECTURA DE REDES,  
 SISTEMAS Y SERVICIOS  
Área de Ingeniería Telemática

## Programación con Sockets

**API de Sockets**

- Introducida en el UNIX BSD4.2 en 1983
- Centrada en el paradigma cliente/servidor
- Ofrece dos tipos de servicios de transporte:
  - STREAM: flujo de datos fiable orientado a conexión
  - DGRAM: datagramas

**Socket**

- Creado por la aplicación
- Controlado por el S.O.
- A través suya la aplicación envía y recibe mensajes

6

---

---

---

---

---

---

---

---

**upna**  
UNIVERSIDAD POLITÉCNICA DE NAVARRA

ARQUITECTURA DE REDES,  
 SISTEMAS Y SERVICIOS  
Área de Ingeniería Telemática

## Sockets y UDP

**UDP: no hay "conexión" entre cliente y servidor**

- No hay handshaking
- El emisor debe indicar explícitamente la dirección IP y el puerto del destino para cada paquete
- El servidor debe extraer la dirección IP y el puerto del emisor del paquete

- UDP ofrece transferencia no fiable de grupos de bytes ("datagramas") entre el cliente y el servidor

7

---

---

---

---

---

---

---

---

**upna**  
UNIVERSIDAD POLITÉCNICA DE NAVARRA

ARQUITECTURA DE REDES,  
 SISTEMAS Y SERVICIOS  
Área de Ingeniería Telemática

## Ejemplo en pseudo-código

**Cliente**

- Crear el socket UDP (Dgram)
- Solicitar al S.O. que se envíen ciertos datos a un destino (IP+puerto) concreto (...)

Datos

**Servidor**

- Crear el socket UDP (Dgram)
- Asignarle el puerto en el que esperar
- Esperar un datagrama
- Datagrama recibido (o no)

8

---

---

---

---

---

---

---

---

upna

ARQUITECTURA DE REDES, SISTEMAS Y SERVICIOS  
 Área de Ingeniería Informática

## Ejemplo en C (I)

**Cliente**

**Servidor**

```

struct sockaddr_in dirsock, emisor;
int sockservidor, ret,
    frlen;
char *buf[2000];

sockservidor=socket(PF_INET,SOCK_DGRAM,0);
if (sockservidor==-1) ERROR();
dirsock.sin_family=AF_INET;
          
```

Crear el socket UDP (...)

Cliente

Servidor

9

---

---

---

---

---

---

---

---

---

---

upna

ARQUITECTURA DE REDES, SISTEMAS Y SERVICIOS  
 Área de Ingeniería Informática

## Ejemplo en C (II)

**Cliente**

**Servidor**

```

dirsock.sin_addr.s_addr=INADDR_ANY;
dirsock.sin_port=htons(99);
ret= bind(sockservidor, (struct
sockaddr*)&dirsock, sizeof(dirsock));
if (ret==-1) ERROR();

ret=recvfrom(sockservidor, buf, 2000, 0,
(struct sockaddr*)&emisor, &rlen);
          
```

Asignar Puerto (...)  
Esperar a recibir

Cliente

Servidor

Recibe datagramas al puerto 99

10

---

---

---

---

---

---

---

---

---

---

upna

ARQUITECTURA DE REDES, SISTEMAS Y SERVICIOS  
 Área de Ingeniería Informática

## Ejemplo en C (III)

**Cliente**

```

int sockcliente, ret;
struct sockaddr_in dirsock;
struct hostent *resolvhst;

sockcliente=socket(PF_INET,SOCK_DGRAM,0);
if (sockcliente==-1) ERROR();
          
```

**Servidor**

- Esperando a recibir

Crear el socket UDP (...)

Cliente

Servidor

Recibe datagramas al puerto 99

11

---

---

---

---

---

---

---

---

---

---





upna

ARQUITECTURA DE REDES, SISTEMAS Y SERVICIOS  
 Anal. de redes y sistemas de redes

## Cliente en C (y IV)

### Cliente

```

write(sockcliente, ...);
read(sockcliente, ...);
...
close(sockcliente);

```

### Servidor

- Escribir/Leer del socket
- Cierre de la conexión

Acepta conexiones al puerto 80

18

---

---

---

---

---

---

---

---

---

---

upna

ARQUITECTURA DE REDES, SISTEMAS Y SERVICIOS  
 Anal. de redes y sistemas de redes

## Servidor en C (I)

### Cliente

- Crear el socket TCP (Stream)

### Servidor

```

int sockservidor, sockconectado;
int ret, dirllen=sizeof(dirsock);
struct sockaddr_in dirsock;

sockservidor=socket(PF_INET, SOCK_STREAM, 0);
if (sockservidor==-1) ERROR();

```

19

---

---

---

---

---

---

---

---

---

---

upna

ARQUITECTURA DE REDES, SISTEMAS Y SERVICIOS  
 Anal. de redes y sistemas de redes

## Servidor en C (II)

### Cliente

- Crear el socket TCP (Stream)

### Servidor

```

dirsock.sin_family=AF_INET;
dirsock.sin_addr.s_addr=INADDR_ANY;
dirsock.sin_port=htons(80);

ret= bind(sockservidor, (struct
sockaddr*)&dirsock, sizeof(dirsock));
if (ret==-1) ERROR();

```

Puerto 80

20

---

---

---

---

---

---

---

---

---

---

upna

ARQUITECTURA DE REDES, SISTEMAS Y SERVICIOS  
Andrés Fajardo Hernández

## Servidor en C (III)

**Ciente**

- Solicitar al S.O. que lo conecte con un destino (IP+puerto) concreto (...)

**Servidor**

```
ret=listen(sockservidor,5);
if (ret==-1) ERROR();

sockconectado=accept(sockservidor, (struct
sockaddr*)&dirsock, &dirrlen);
```

Entregar conexión

Completar conexiones al Puerto 80

21

---

---

---

---

---

---

---

---

---

---

upna

ARQUITECTURA DE REDES, SISTEMAS Y SERVICIOS  
Andrés Fajardo Hernández

## Servidor en C (y IV)

**Ciente**

Escribir/Leer del socket

- Cierre de la conexión

**Servidor**

```
write(sockconectado,...);
read(sockconectado,...);
.
.
.
.
close(sockconectado);
```

Enviar/Recibir  
Cerrar Conexión (...)

Completar conexiones al Puerto 80

22

---

---

---

---

---

---

---

---

---

---