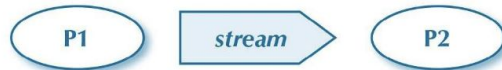


UT3. Programación de Comunicaciones en red

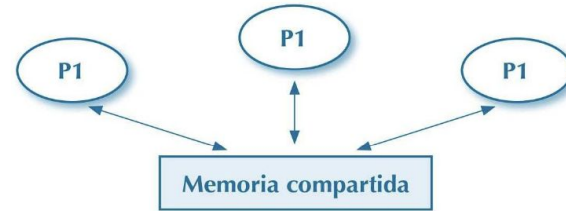
PSP - DAM
Francisco Gallego Perona

Modelos de comunicaciones entre procesos

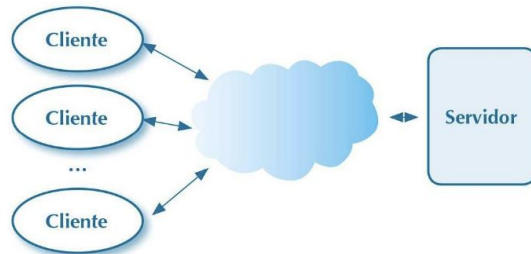
Mediante E/S



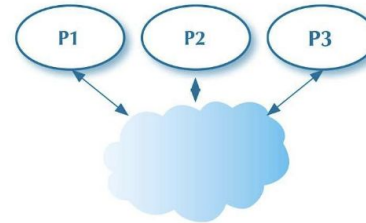
Mediante memoria compartida



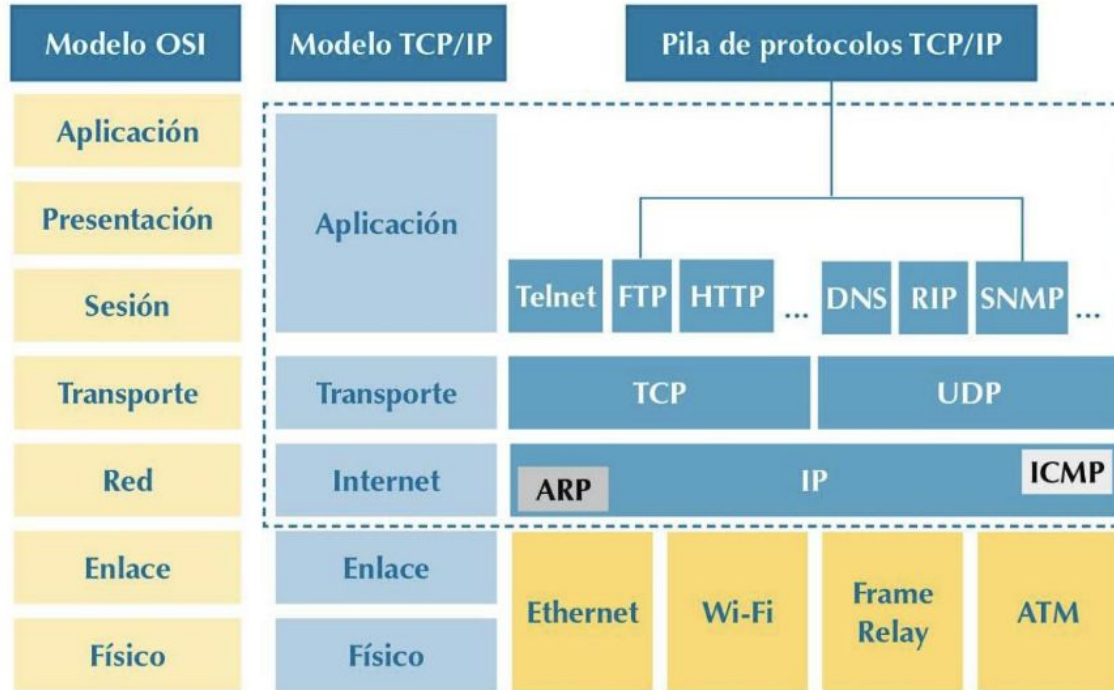
Distribuido C/S
(Cliente/Servidor)



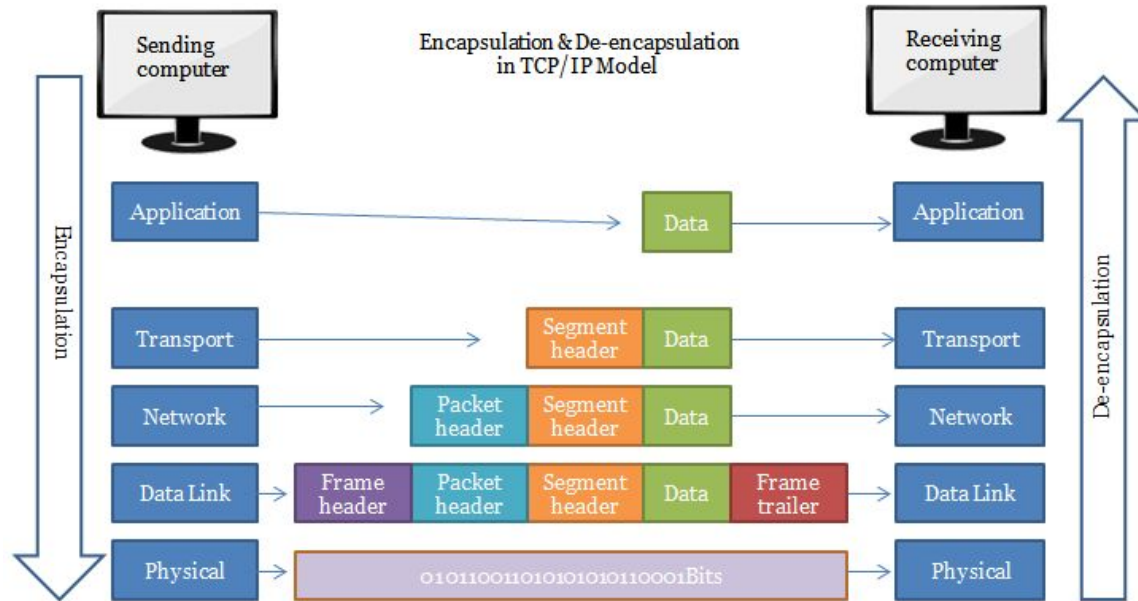
Distribuido P2P
(entre iguales)



Modelo de niveles TCP/IP



Encapsulación de datos en TCP/IP



Header en TCP vs UDP

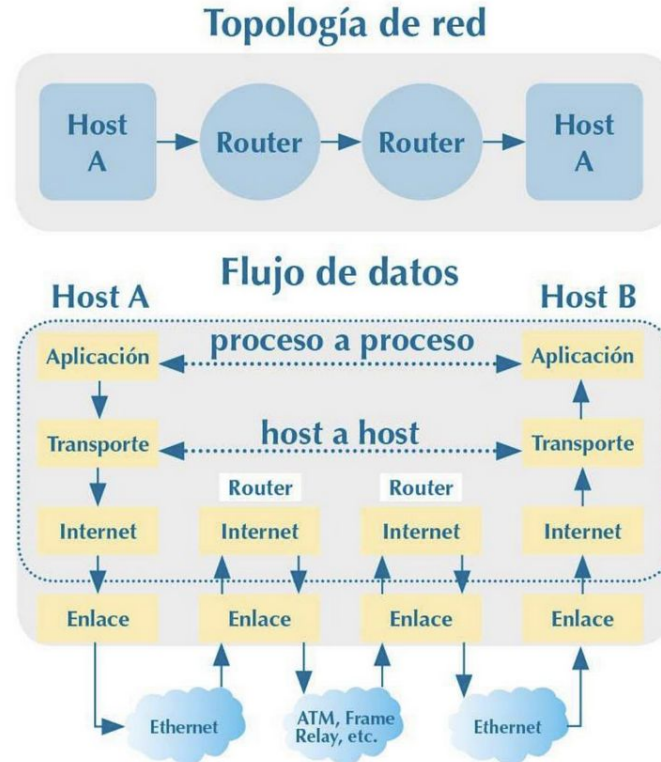
TCP Segment Header Format

Bit #	0	7	8	15	16	23	24	31
0	Source Port				Destination Port			
32	Sequence Number							
64	Acknowledgment Number							
96	Data Offset	Res	Flags		Window Size			
128	Header and Data Checksum				Urgent Pointer			
160...	Options							

UDP Datagram Header Format

Bit #	0	7	8	15	16	23	24	31
0	Source Port				Destination Port			
32	Length				Header and Data Checksum			

Implementación de niveles de TCP/IP en hosts y routers



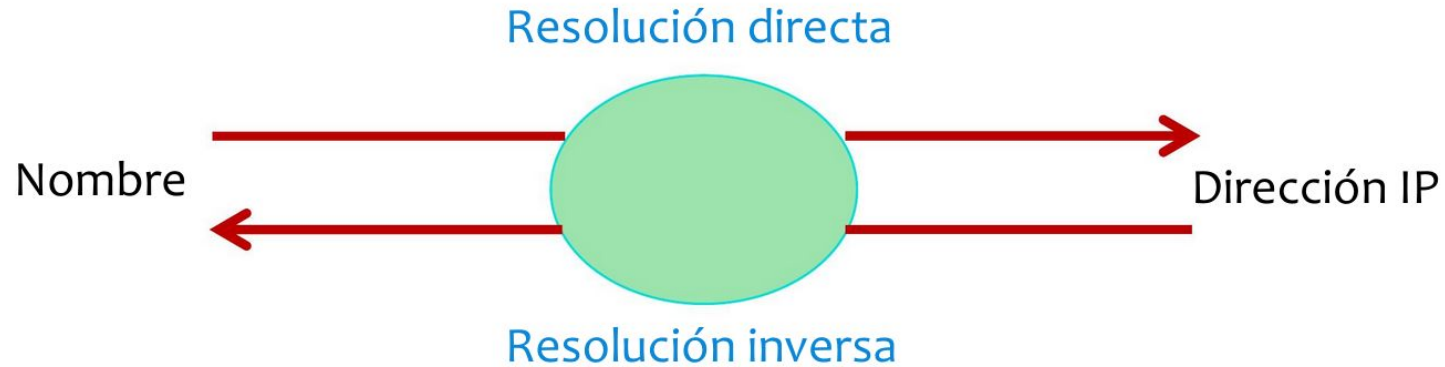
Nivel de transporte: TCP vs UDP

➤ Nivel de transporte:

- Comunicación host a host, entre sockets o puntos de conexión.
 - Socket: dirección IP + número de puerto.
- Protocolo UDP.
 - No orientado a conexión, entrega de paquetes de datos individuales (datagramas).
 - No garantiza la entrega de datagramas.
 - No garantiza la entrega en orden de datagramas.
 - Tamaño de datagrama limitado por MTU, que suele ser 1500 bytes para Ethernet. Datagramas mayores se fragmentan.
- Protocolo TCP.
 - Orientado a conexión, flujo de datos entre extremos de una conexión.
 - Garantiza la entrega de datos.
 - Garantiza la entrega de datos en orden.

Resolución de nombres

Basada en el protocolo de nivel de aplicación DNS, que funciona sobre UDP.



Resolución de nombres

Basada en el protocolo de nivel de aplicación DNS, que funciona sobre UDP.

```
pacotoh@pop-os:~$ nslookup www.google.es
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
Name:   www.google.es
Address: 142.250.178.163
Name:   www.google.es
Address: 2a00:1450:4003:811::2003
```

Clases de Java para comunicaciones en red

El paquete `java.net` proporciona clases que permiten llevar a cabo comunicaciones entre procesos utilizando los protocolos estándares de la familia de TCP/IP.

Clases de Java para interfaces de red

Para acceder a la información de las interfaces de red utilizamos los comandos **ifconfig** (Linux) e **ipconfig** (Windows). En Java tenemos la clase **NetworkInterface**, con métodos estáticos que permiten obtener la información de todas las interfaces de red del sistema.

Resolución de nombres con Java

La clase **InetAddress** tiene métodos estáticos que realizan resolución de nombres. Es decir, dado un nombre, permiten obtener la dirección IP asociada. También tiene métodos que realizan la resolución inversa de nombres.

Método	Funcionalidad
<code>static InetAddress getByName(String host)</code>	Obtiene una dirección IP a partir de un nombre de host, utilizando servicios de resolución de nombres, o bien a partir de una representación textual de una dirección IP.
<code>static InetAddress[] getAllByName(String host)</code>	Similar al método anterior, pero más completo, porque devuelve un array con todas las direcciones IP que se pueden obtener utilizando servicios de resolución de nombres. También se puede especificar el host mediante la representación textual de una dirección IPv4 o IPv6.

Resolución de nombres con Java

```
public String  
getHostName()
```

Este método permite realizar una resolución inversa de nombre, es decir, obtener un nombre de host a partir de su dirección IP. Si la `InetAddress` se creó sin especificar un nombre para el host, entonces se realiza una resolución inversa de nombres para obtener el nombre del host a partir de su dirección IP.

También se puede crear una `InetAddress` indicando una dirección IP y un nombre de host, con `static InetAddress getByAddress(String host, byte[] addr)`. En ese caso, este método devolvería el nombre proporcionado al crear la `InetAddress`, sin realizar resolución inversa de nombres.

```
String  
getCanonicalHostName()
```

Devuelve el nombre de host canónico o FQDN (*fully qualified domain name*), utilizando el protocolo DNS para obtener esta información.

```
static InetAddress  
getLocalHost()
```

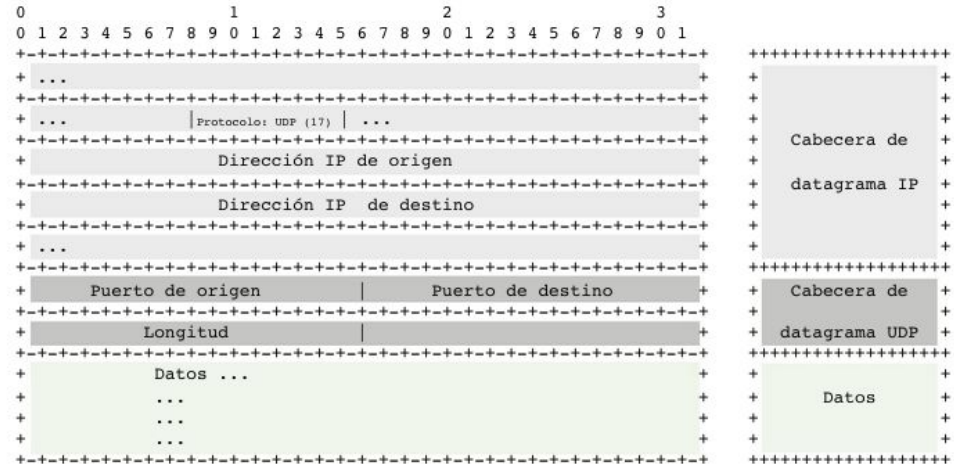
Devuelve la dirección IP del propio host. En Linux, normalmente se obtiene su nombre de `/etc/hostname`, y se resuelve el nombre para obtener una dirección IP. Dependiendo del sistema, se podría obtener la dirección de la interfaz de *loopback*, una dirección IP privada perteneciente a la red local, o incluso una IP externa o pública.

Si no se puede resolver este nombre, se lanza una excepción de tipo `UnknownHostException`.

Clases de Java para sockets UDP

El protocolo de transporte UDP no es orientado a la conexión. Solo permite el envío de paquetes de datos individuales, llamados datagramas.

Estos datagramas pueden llegar en orden distinto al orden en el que se enviaron, e incluso no llegar.



Clases de Java para sockets UDP

