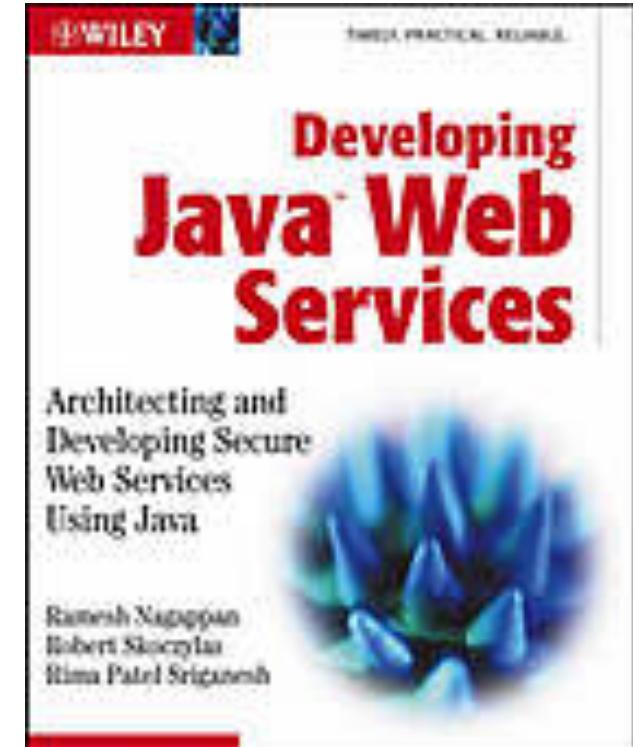
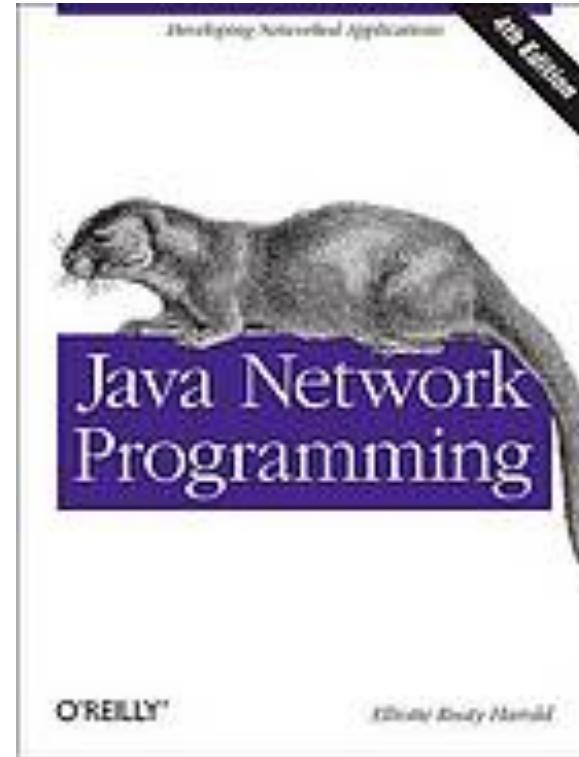
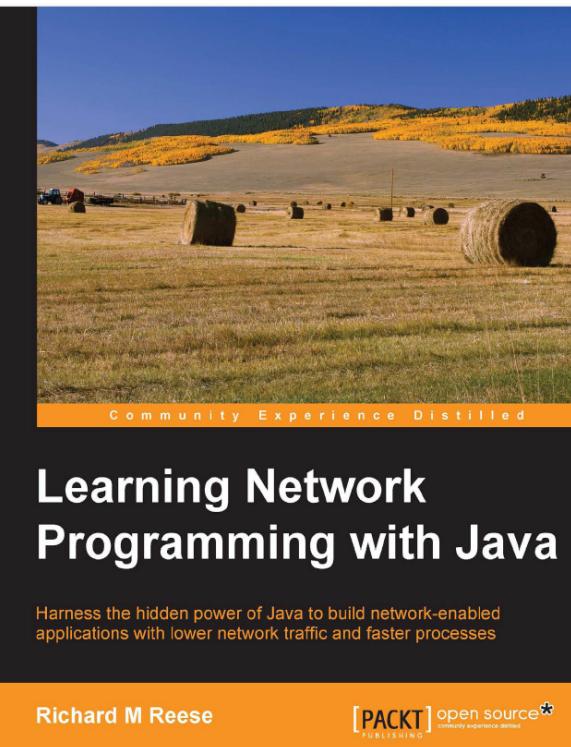
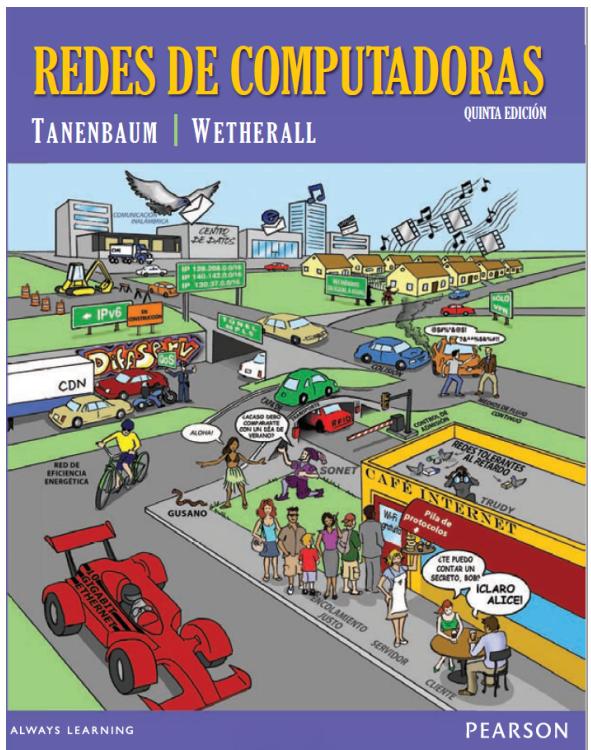


REDES DE COMPUTADORES Y LABORATORIO

Yor Castaño, MSc



BIBLIOGRAFÍA



DE APLICACIÓN A TRANSPORTE:

- ¿Que sabemos del modelo OSI y de la pila TCP/IP?
- ¿Que sabemos de la capa de aplicación?
- Que sabemos sobre los servicios y protocolos de la capa de aplicación?
- ¿Esta claro el concepto de alta disponibilidad (HA: High Availability)?
- ¿Por que es importante HA?
- ¿Que no entendemos?

CONCEPTOS IMPORTANTES:

- Modelo OSI
- TCP/IP
- Capa presentación y sesión en OSI
- Capa Aplicación en OSI y TCP/IP
- DNS
- EMAIL
- POP
- IMAP
- SMTP (MUA, MTA)

CONCEPTOS IMPORTANTES:

- Modelo OSI
- TCP/IP
- Capa presentación y sesión en OSI
- Capa Aplicación en OSI y TCP/IP
- DNS
- EMAIL
- POP
- IMAP
- SMTP (MUA, MTA)
- WWW
- HTTP
- ARQUITECTURA CLIENTE/SERVIDOR
- CLIENTE
- SERVIDOR
- PUERTOS (Well Known)
- SOCKETS
- IETF
- IANA
- RFC

TIPOS DE ARQUITECTURA CLIENTE/SERVIDOR

Arquitectura de dos niveles

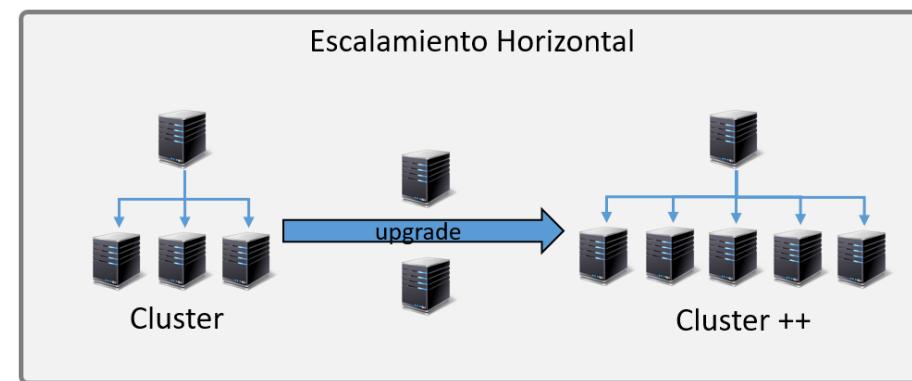
- Código cliente de la aplicación
- Servidor de base de datos

Arquitectura de tres niveles

- La capa de presentación
- La capa de funcionalidad
- La capa de lógica de los datos

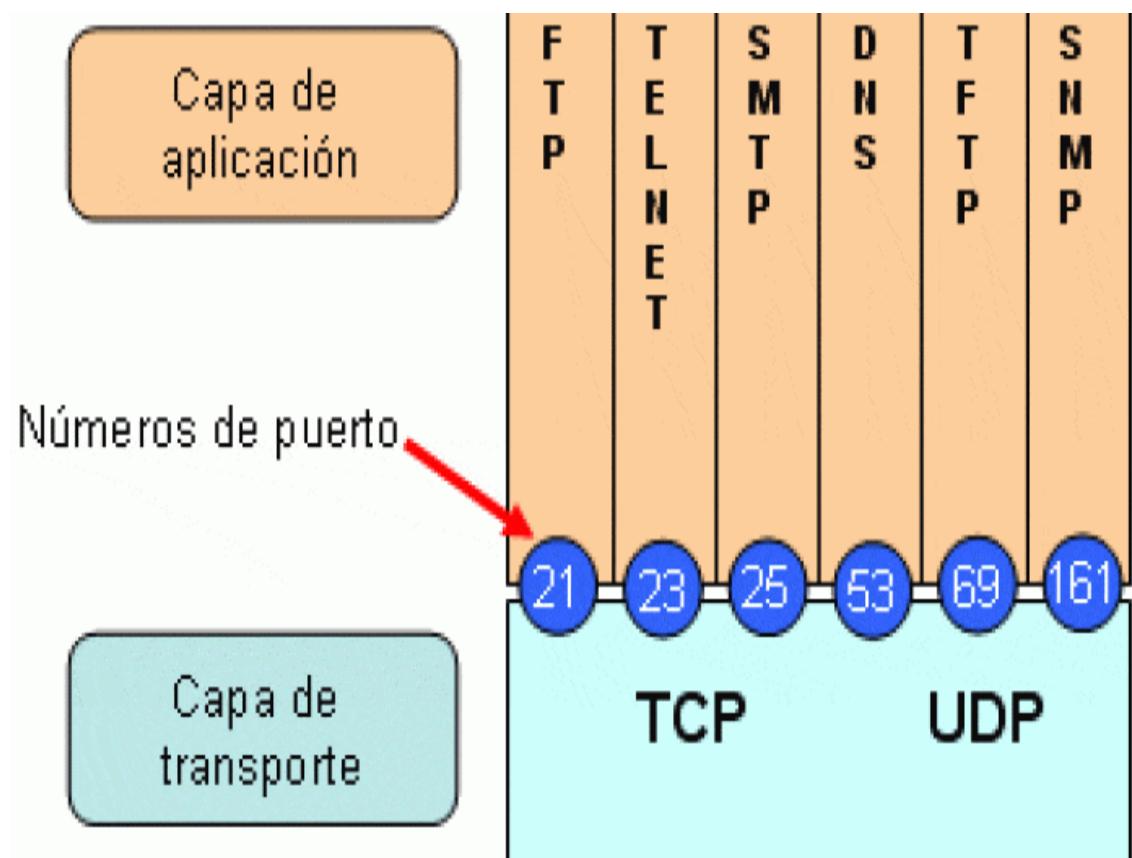
Arquitectura de N-Niveles

- Escalar vertical-horizontal



PUERTO

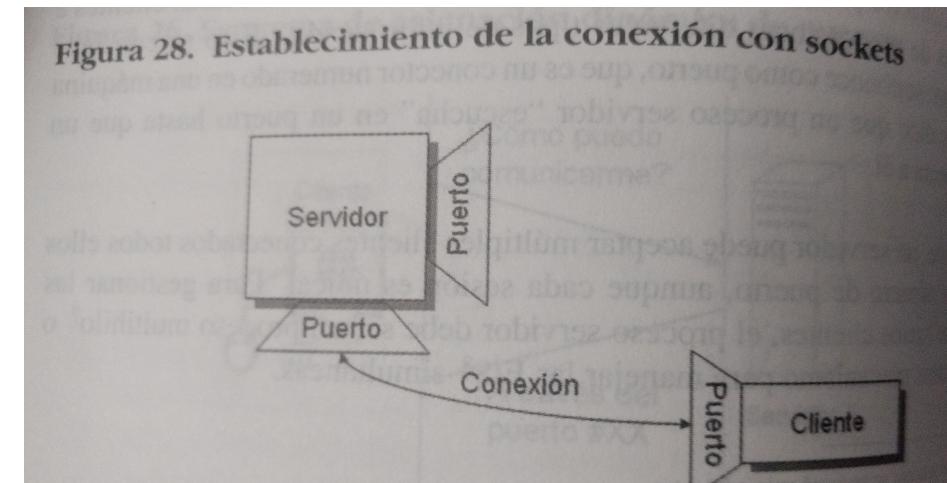
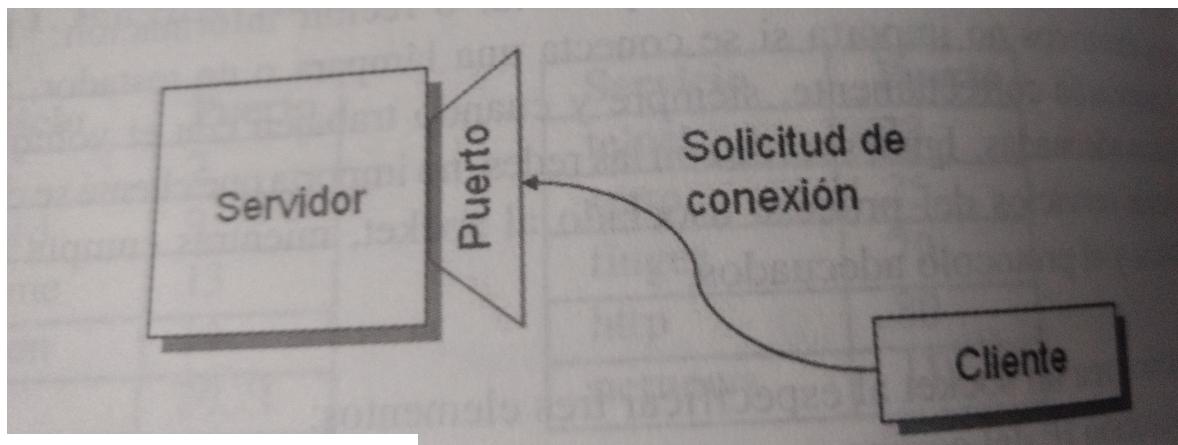
- Número asignado a un proceso, siempre y cuando el proceso solicite escuchar algún tipo de tráfico mediante **Sockets**
- Cantidad de puertos 65.536, de este total hay 1024 reservados



SOCKET

Una aplicación crea un socket al especificar tres elementos:

- Dirección IP del equipo. (Ej: 192.168.1.10)
- El puerto que la aplicación esté utilizando. (Ej: 21)
- El tipo de servicio (UDP o TCP) utilizado por la aplicación. (TCP)*

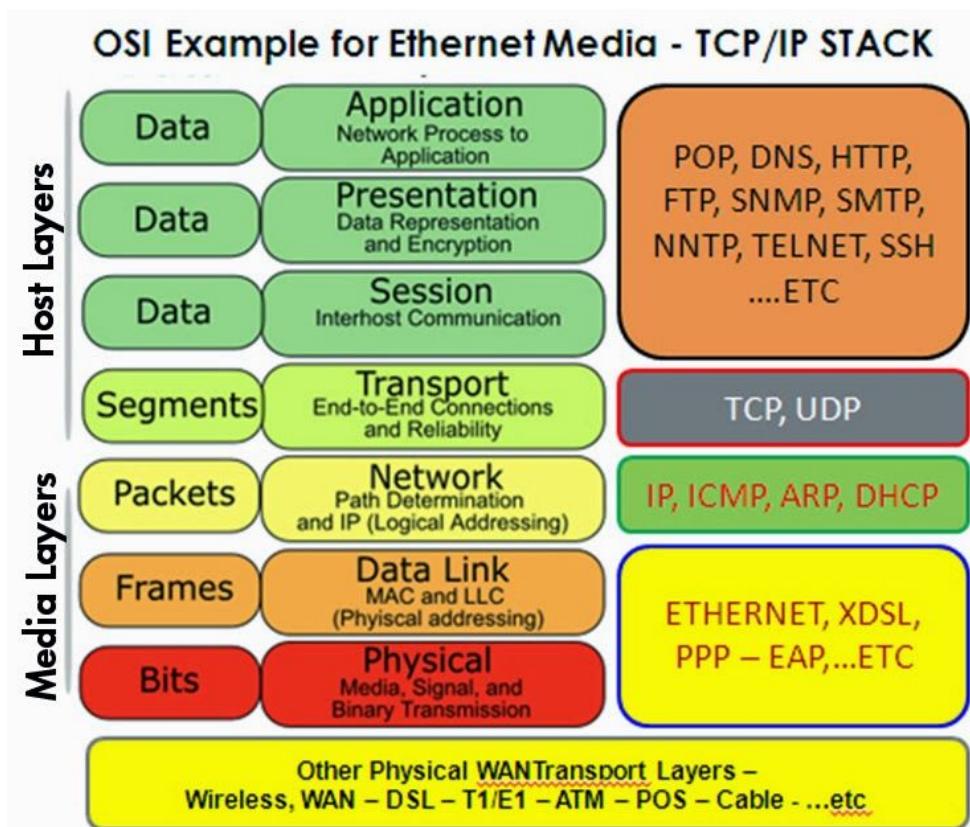


OTROS CONCEPTOS:

- nslookup
- Ping
- traceroute
- Webhook Relay (tunneling)
- Docker
- Jmeter (Testing tools)
- Postman/Postwoman (HTTP Request)
- Whois
- Namechecker
- microservicios
- Python y Flask
- Balanceadores de carga (nginx)
- Proxy y proxy reverso
- DynDNS
- ORM en bases de datos
- Cloud Providers
- PaaS, IaaS, CaaS
- Packet Tracer

COMPETENCIAS

- Describir el uso de las primitivas.
- Describir UDP.
- Describir TCP.



- *De ahora en adelante entendamos la **capa de transporte** como los vehiculos de comunicación*
- *Mientras que la **capa de red** es la autopista en la que transitan esos vehiculos.*

LA CAPA DE TRANSPORTE

- Recordemos.... La capa de red provee entrega de paquetes punto a punto mediante el uso de datagramas o circuitos virtuales.
- El objetivo de la capa de transporte es proporcionar un servicio de transmisión de datos eficiente, confiable y económico a sus usuarios, procesos que normalmente son de la capa de aplicación.
- Gracias a esta capa, los programadores pueden escribir código de acuerdo con un conjunto estándar de **primitivas**; estos programas pueden funcionar en una amplia variedad de redes sin necesidad de preocuparse por lidiar con diferentes interfaces de red y distintos niveles de confiabilidad.

PRIMITIVAS DEL SERVICIO DE TRANSPORTE

- La capa de transporte debe proporcionar algunas operaciones a los programas de aplicación (software de alto nivel) a través de una interfaz de servicios.
- Cada servicio tiene su propia interfaz.
- Interfaz orientada a la conexión. (Ejemplo Hipotético)

Primitiva	Paquete enviado	Significado
LISTEN	(ninguno)	Se bloquea hasta que algún proceso intenta conectarse.
CONNECT	CONNECTION REQ.	Intenta activamente establecer una conexión.
SEND	DATA	Envía información.
RECEIVE	(ninguno)	Se bloquea hasta que llegue un paquete DATA.
DISCONNECT	DISCONNECTION REQ.	Solicita que se libere la conexión

- Los mensajes enviados a través de una entidad de transporte a otra se conocen como **segmentos**.

PRIMITIVAS DEL SERVICIO DE TRANSPORTE

- Interfaz **orientada a la conexión**.

Primitiva	Paquete enviado	Significado
LISTEN	(ninguno)	Se bloquea hasta que algún proceso intenta conectarse.
CONNECT	CONNECTION REQ.	Intenta activamente establecer una conexión.
SEND	DATA	Envía información.
RECEIVE	(ninguno)	Se bloquea hasta que llegue un paquete DATA.
DISCONNECT	DISCONNECTION REQ.	Solicita que se libere la conexión

- Los mensajes enviados a través de una entidad de transporte a otra se conocen como **segmentos**.

PRIMITIVAS DEL SERVICIO DE TRANSPORTE

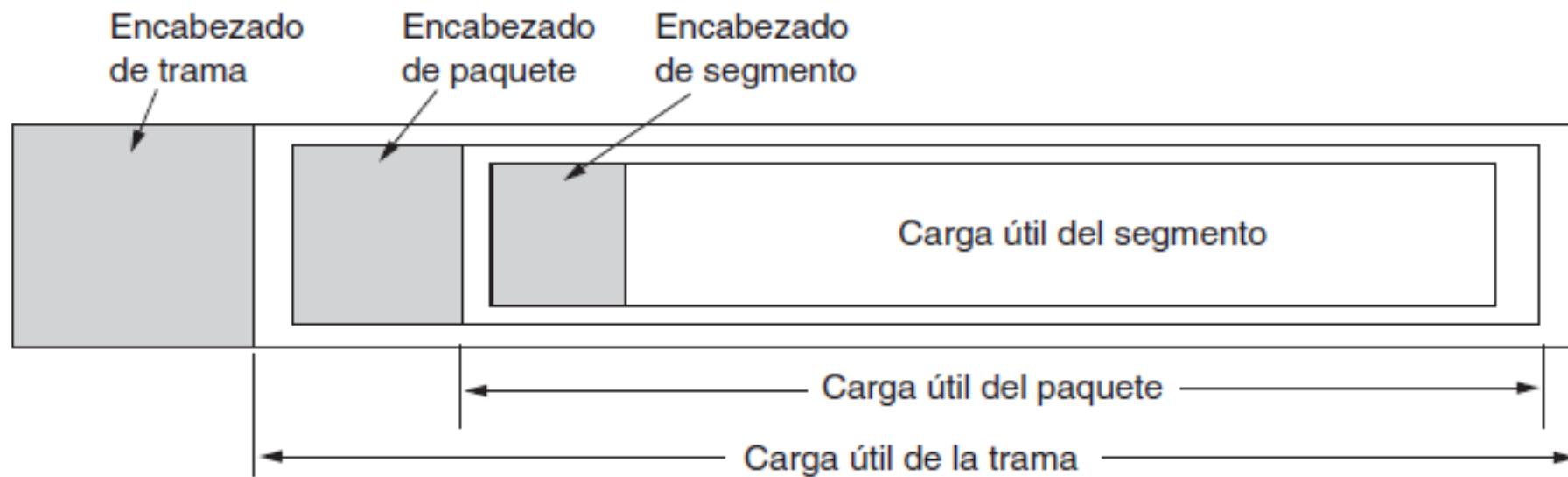


Figura 6-3. Anidamiento de segmentos, paquetes y tramas.

Primitiva	Paquete enviado	Significado
LISTEN	(ninguno)	Se bloquea hasta que algún proceso intenta conectarse.
CONNECT	CONNECTION REQ.	Intenta activamente establecer una conexión.
SEND	DATA	Envía información.
RECEIVE	(ninguno)	Se bloquea hasta que llegue un paquete DATA.
DISCONNECT	DISCONNECTION REQ.	Solicita que se libere la conexión

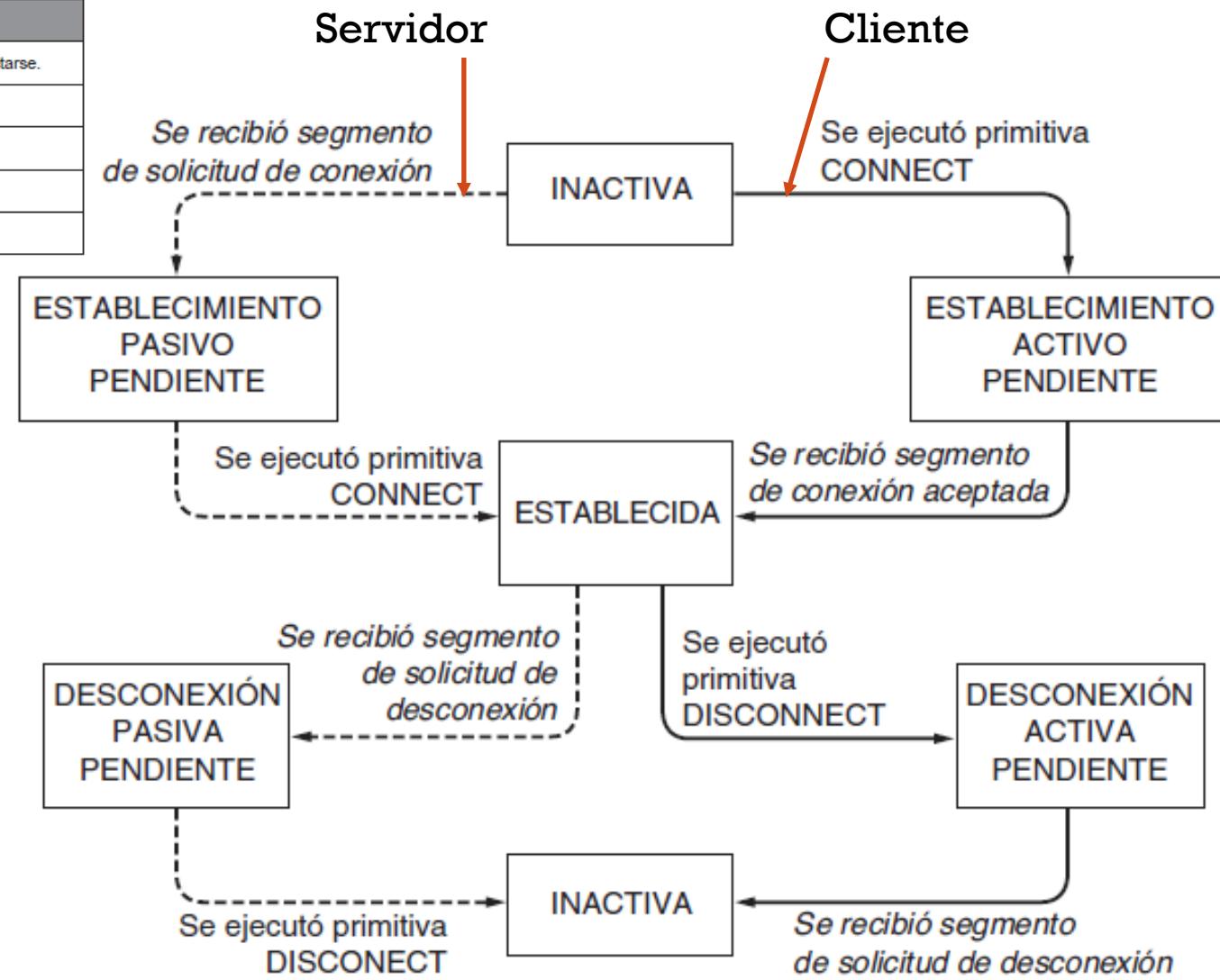


Figura 6-4. Un diagrama de estado para un esquema simple de manejo de conexiones. Las transiciones etiquetadas en cursiva se producen debido a la llegada de paquetes. Las líneas continuas muestran la secuencia de estados del cliente. Las líneas punteadas muestran la secuencia de estados del servidor.

SOCKETS DE BERKLEY

- Son un conjunto de primitivas de transporte: las primitivas de socket que se utilizan para TCP.

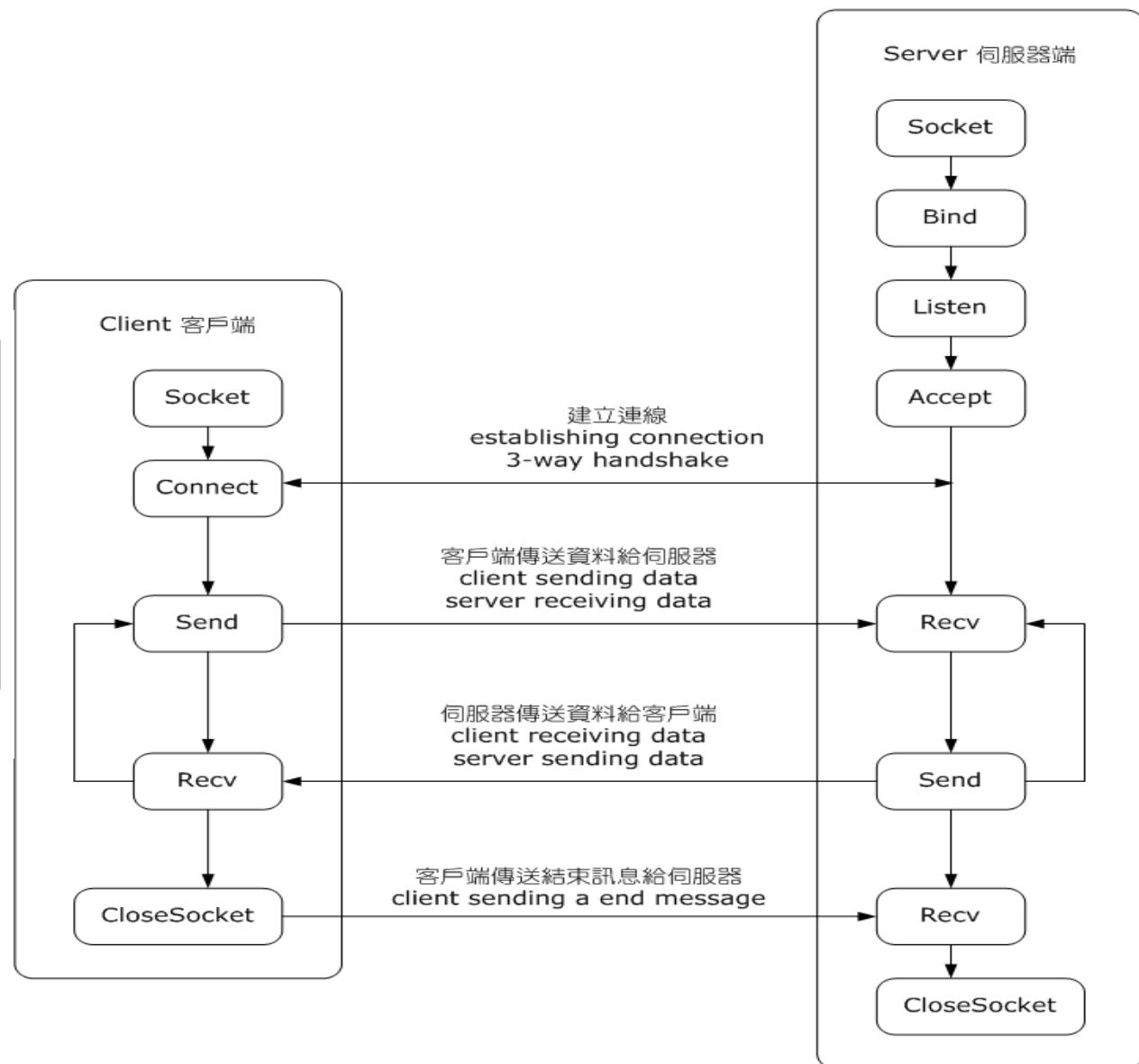
Primitiva	Significado
SOCKET	Crea un nuevo punto terminal de comunicación.
BIND	Asocia una dirección local con un socket.
LISTEN	Anuncia la disposición de aceptar conexiones; indica el tamaño de la cola.
ACCEPT	Establece en forma pasiva una conexión entrante.
CONNECT	Intenta establecer activamente una conexión.
SEND	Envía datos a través de la conexión.
RECEIVE	Recibe datos de la conexión.
CLOSE	Libera la conexión.

Figura 6-5. Las primitivas de socket para TCP.

TCP Socket 基本流程圖
TCP Socket flow diagram

Primitiva	Significado
SOCKET	Crea un nuevo punto terminal de comunicación.
BIND	Asocia una dirección local con un socket.
LISTEN	Anuncia la disposición de aceptar conexiones; indica el tamaño de la cola.
ACCEPT	Establece en forma pasiva una conexión entrante.
CONNECT	Intenta establecer activamente una conexión.
SEND	Envía datos a través de la conexión.
RECEIVE	Recibe datos de la conexión.
CLOSE	Libera la conexión.

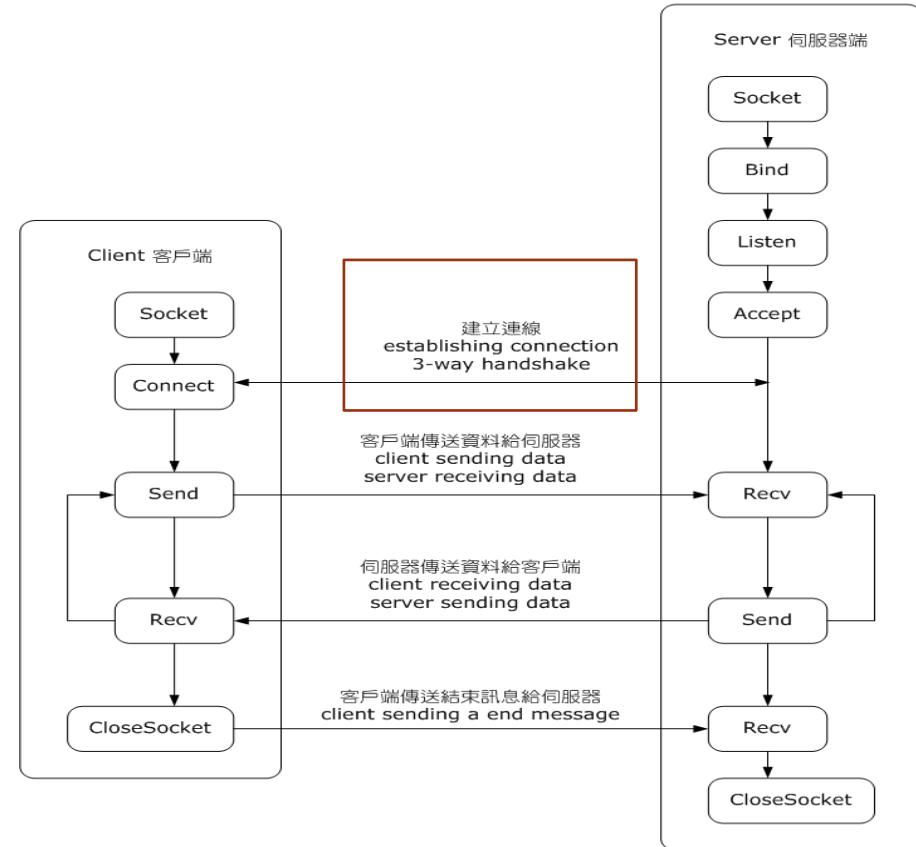
Figura 6-5. Las primitivas de socket para TCP.



DIRECCIONAMIENTO Y ESTABLECIMIENTO DE CONEXIÓN

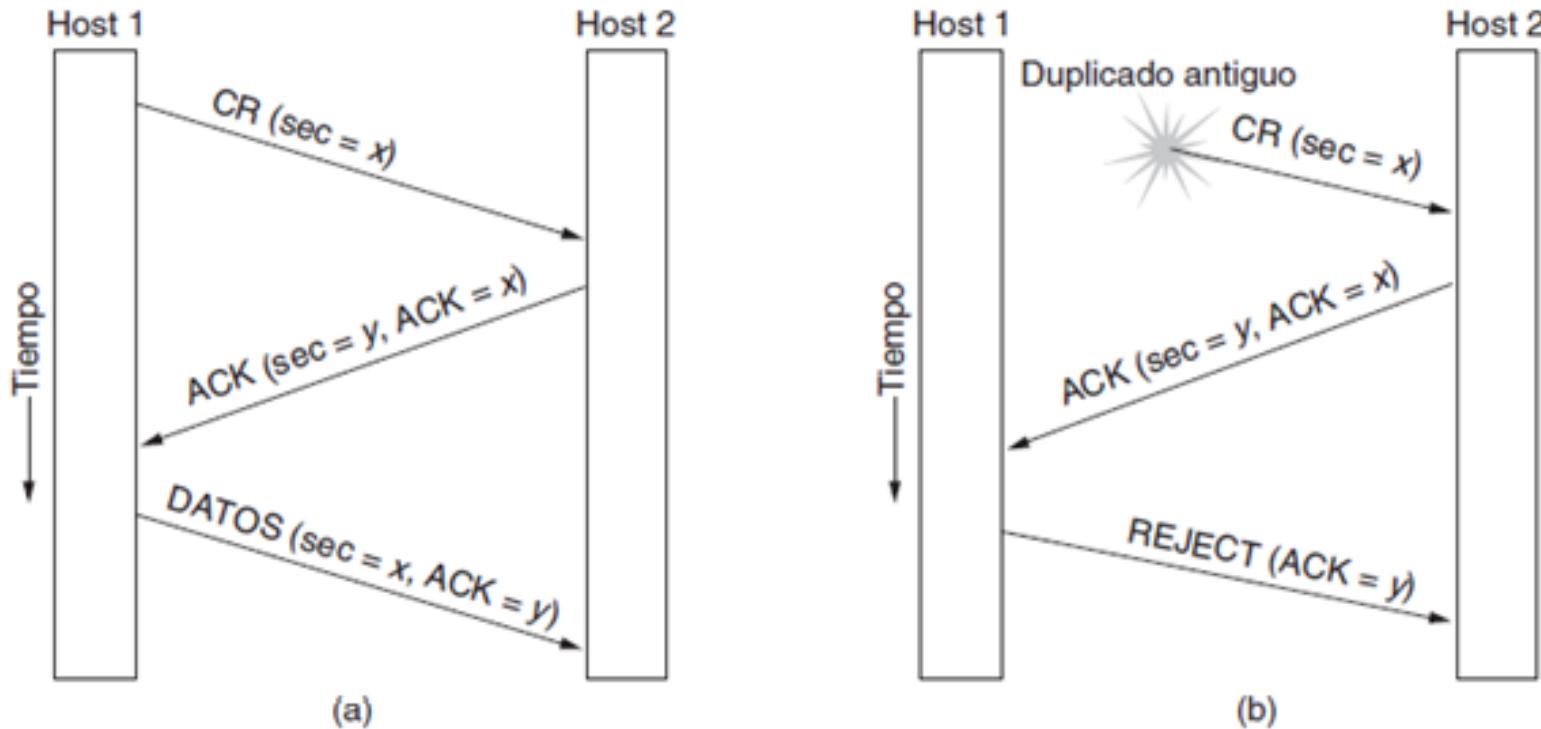
- El método que permite definir las direcciones de transporte en las que los procesos pueden escuchar solicitudes de conexión es a través de puntos terminales conocidos como **puertos**.
- **Acuerdo de tres vías (three-way handshake)**. Es un protocolo que implica que un igual (receptor) verifique que la conexión sea realmente la actual.

TCP Socket 基本流程圖
TCP Socket flow diagram



SEGMENTOS

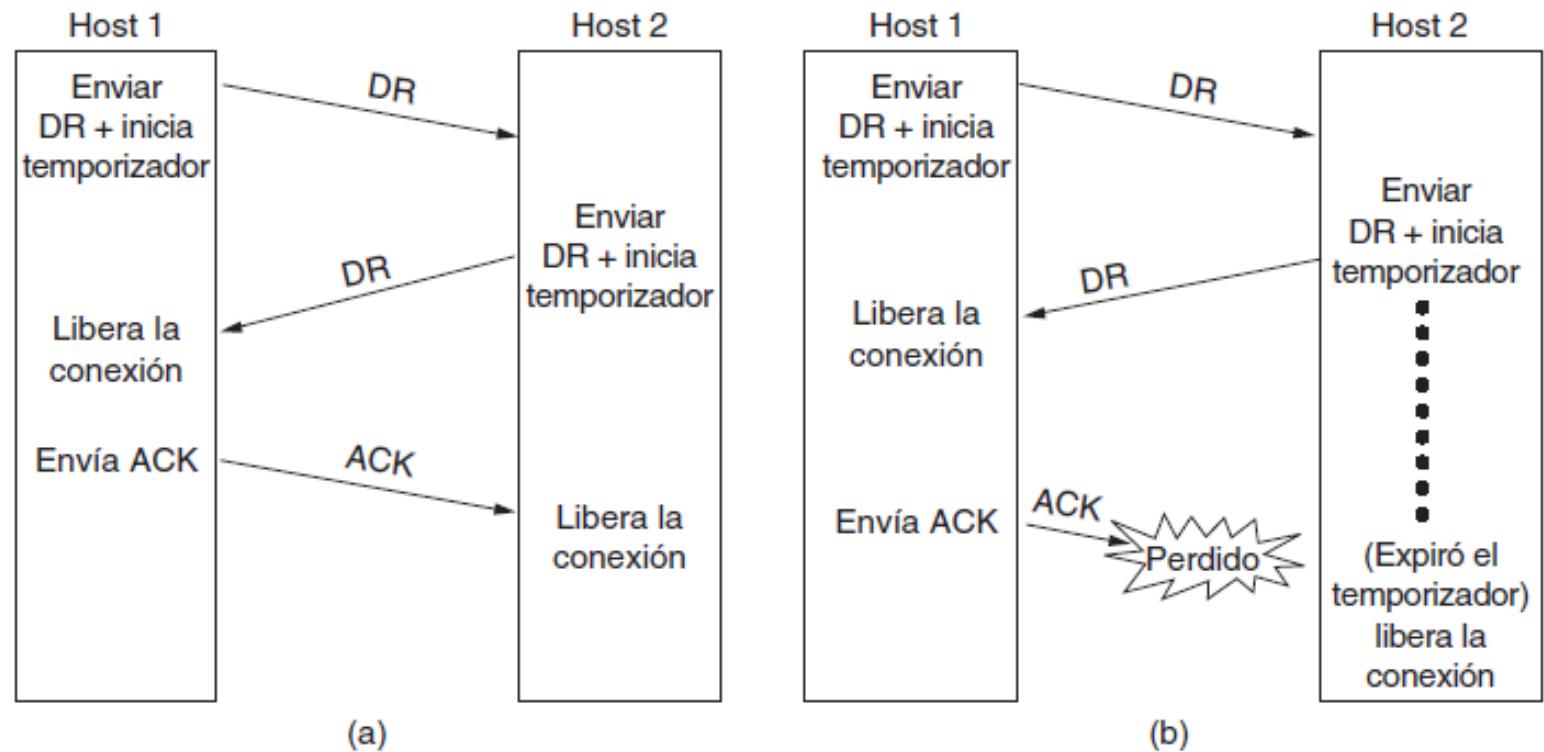
- CR (CONNECTION REQUEST)
- DR (DISCONNECTION REQUEST)
- ACK (acknowledgement), es un mensaje que el destino de la comunicación envía al origen de esta para confirmar la recepción del mensaje.



Establecimiento de una conexión

Algunos segmentos:

- ACK (acknowledgement), es un mensaje que el destino de la comunicación envía al origen de esta para confirmar la recepción del mensaje.
- CR (CONNECTION REQUEST)
- DR (DISCONNECTION REQUEST)



Liberación de una conexión

LOS PROTOCOLOS DE TRANSPORTE DE INTERNET: UDP

UDP (USER DATAGRAM PROTOCOL)

- Proporciona una forma para que las aplicaciones envíen datagramas IP encapsulados sin tener que establecer una conexión. [RFC 768](#).
- UDP es un protocolo simple para interacciones cliente/servidor y multimedia.
- UDP trasmite segmentos que consisten en un encabezado de 8 bytes seguido de la carga útil.

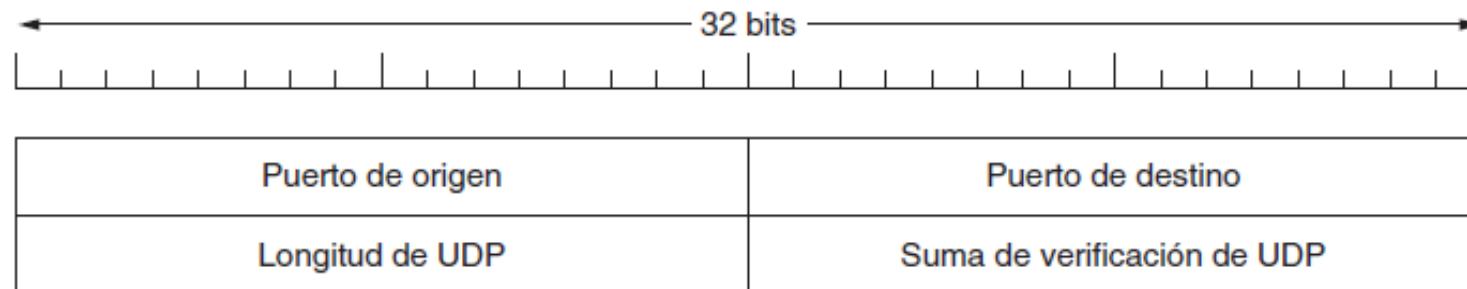


Figura 6-27. El encabezado UDP.

- La longitud incluye el encabezado de 8 bytes y los datos. La longitud mínima es de 8 bytes y la máxima es de 655515 bytes.