Servicios definidos en la capa de transporte.

Hay 2 tipos de servicio, orientado y no orientado a conexion.

. El orientado a conexión consta de 3 partes.

-Establecimiento

-Transferencia de datos -Liberación

En el servicio no orientado a la conexión se tratan los paquetes de manera individual.

Pila de protocolos TCP/IP

capa de red Fisica. especifica los coracteristicas dei hardware que se usara para la red

Capa de vinculo de datos. dentifica el tipo de protocolo de red del paquete, en este caso TCP/IP, proporciona tambien control de errores y estructuras

Capa de Internet. acepta y transfiere paquetes para la red, incluye el protocolo IP, el protocolo de solución de direcciones (ARP) y el protocolo de mensajes de control de Internet (ICMP)

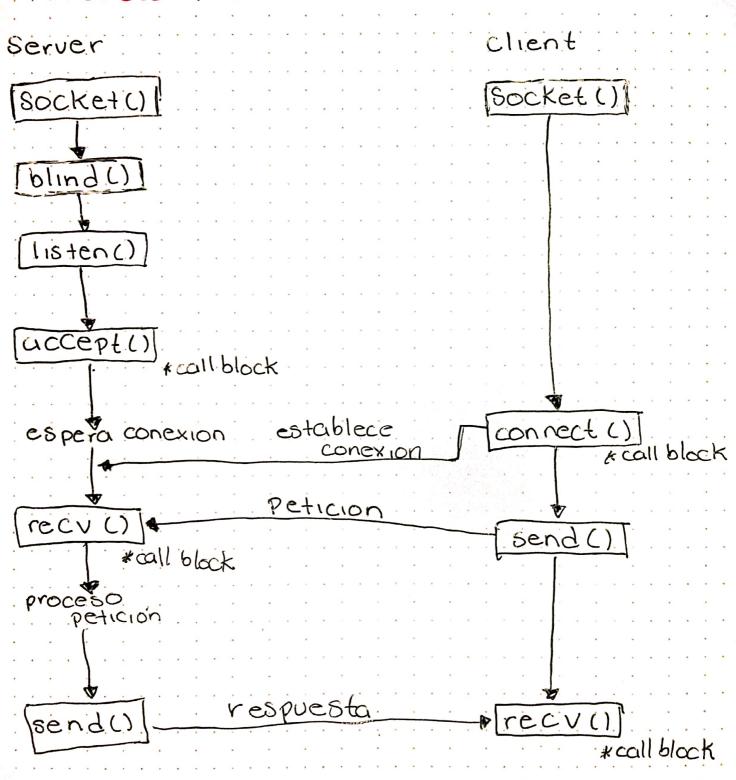
Capa de transporte. garantiza que los paquetes lleguen en secuen cia y sin errores

Capa de aplicación define las aplicaciones de red y los servicios de internet estandar que puede utilizar Un USUATIO.

Modela Cliente - Servidor

el cliente hace una petición o solicita un servicio de red y el servidor responde a las peticiones de los clientes, proporcionando el servicio requendo

Sockets TCP.



Socket() crea un socket, sus parametros son; Dominio, tipo y protocolo. Devuelve un entero.

>=0 5, se crea correctamente 20 5, se produce un error en la creacion

blind () asocia un socket a un puerto. Uso:

int bind Lint socked, struct sockaddr *addr, int addr len);

sockEd: Socket creado con anterioridad

addr: Puntero a la estructura sockaddr-in

addien: Tamaño de la estructura apuntada por el puntero addr

devuelve O si funciona o Ko si hay un error

listen () Escucha de un puerto TCP, sus parametros:
-descriptor del socket a poner ala escucha
- Tamaño de cola de aceptacion de peticiones

Deuvelve O si funciona o <0 si hay un error

Parametros

- Socket asociado a la dirección y puerto puesto a la escucha
- estructura que contendra la dirección y puerto de l cliente del que se acepta la conexión
- Puntero al tamaño de la estuctura

Alaceptor la conexion se crea un nuevo socket que es el que atiende la conexion

· El socket original que escuchaba la dirección y puerto no se altera despues de alterar la conexión

KO error en la aceptación.

Connect() solicita conexión sus parametros son:

Descriptor del socket que se usa para la conexión

Estructura con dirección y puerto al que desea conectarse

Longitud de la estrucctura anterior

develve O si es correcto o TO si hay un error

reculo recepción de datos en UDP, parametros:

Descriptor de l socket

- buffer donde almacena datos a leer

- Tamaño maximo de los datos a leer

- Opciones de envio, en general O(ninguna opción)

devuelve < O si hay error o >=0 con el numero de datos

leidos o escritos

Send to () Enviode datos en UPP, parametros:

-Descriptor del socket - buffer con los datos a enviar

- Tamaño de los datos a enviar - Opciones de envio, en general O (ninguna opcion) - puntero a la estructura de datos que contiene la dirección y puerto donde enviar los datos - Longitud de la estructura anterior

devuelve <0 si hay error >=0 si nomero de bytes escritos.

Mora Gozman Jose Antonio

TAREA 3.