**Network Programming**

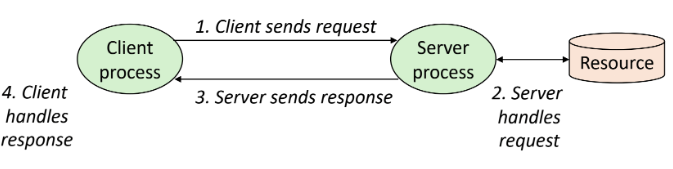
- Network applications are based on the client-server model

- Server process and one or more client processes

- Server manages some resource; provides service by manipulating resource for clients

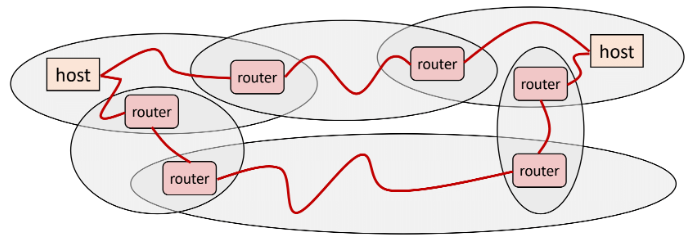
- Server activated by request from client

- Clients and servers are processes running on hosts, can be the same or different hosts



- In hardware, network host treated as another kind of I/O device

- Network is a system of boxes and wires organized by geographical proximity



- No particular topology

- Send packets from source to destination by hopping through networks

- Router forms bridge from one network to another; different packets may take different routes

- How is it possible to send bits across incompatible LANs and WANs?

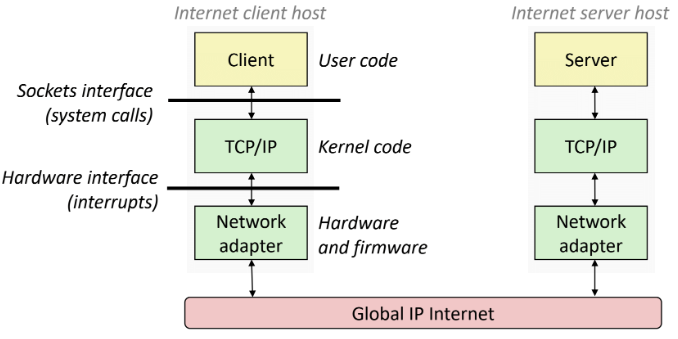
- Protocol is a set of rules that governs how hosts and routers should cooperate when they transfer data from network to network, smoothing out the difference between different networks

- Protocol provides a naming scheme: defines a uniform format for host addresses; each host (and router) is assigned at least one of these addresses

- Protocol provides a delivery mechanism: defines standard transfer unit (packet); packet consists of header and payload

- Global Internet based on the TCP/IP protocol family

- Accessed via a mix of Unix file I/O and functions from the sockets interface



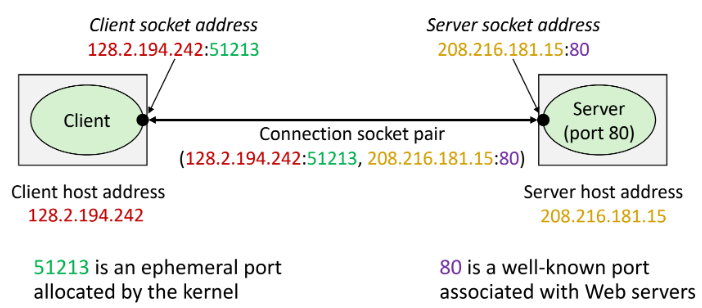
- Programmer’s view:

- Hosts are mapped to set of 32-bit IP addresses

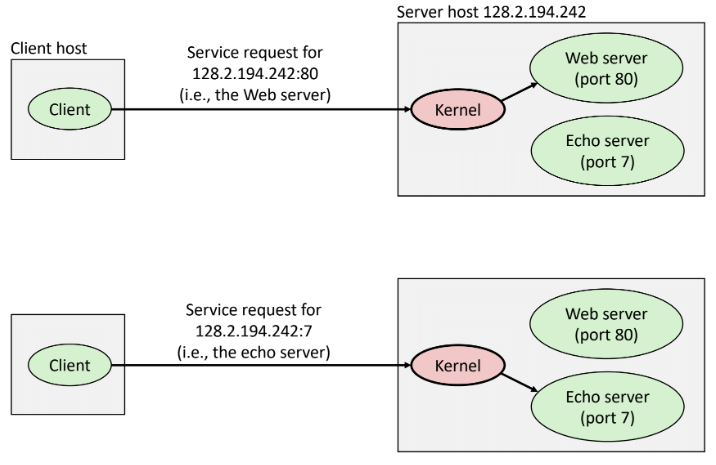
- Set of IP addresses is mapped to set of identifiers called Internet domain names

- Process on one Internet host can communicate with a process on another Internet host over a connection

- Connection is uniquely identified by the socket addresses of its endpoints (socket pair)



- Ports used to identify services



- Socket interface is a set of system-level functions used in conjunction with Unix I/O to build network applications

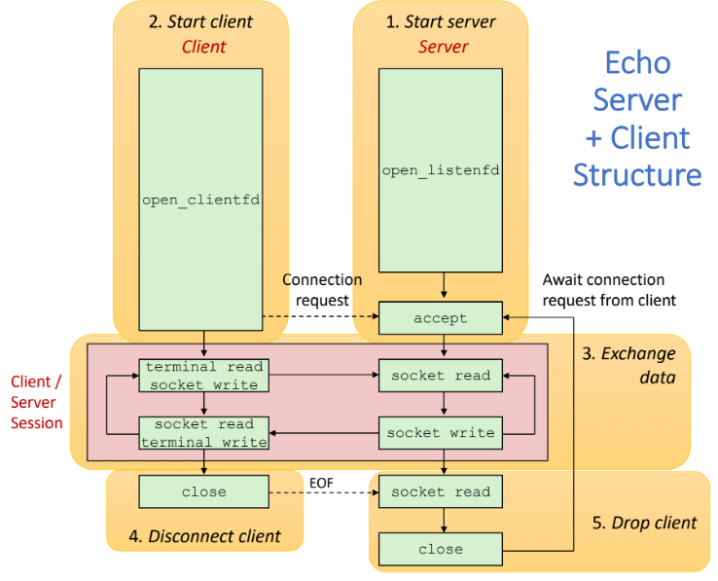
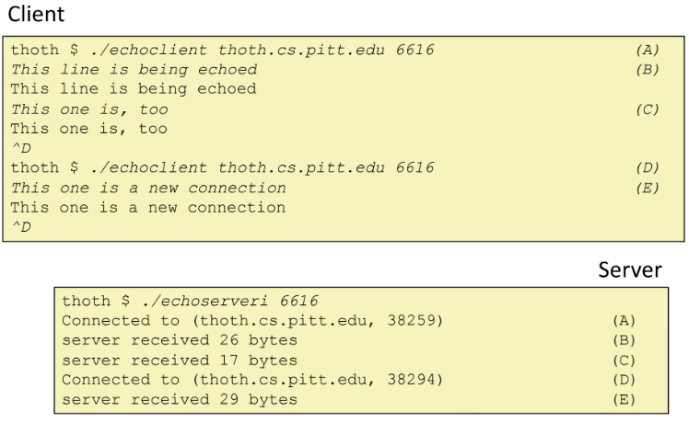
- To the kernel, a socket is an endpoint of communication

- To an application, a socket is a file descriptor that lets the application read/write from/to the network

- All Unix I/O devices, including networks, are modeled as files

- Clients and servers communicate with each other by reading from and writing to socket descriptors

- Main distinction between regular file I/O and socket I/O is how the application “opens” the socket descriptors



- Clients and servers communicate using the HyperText Transfer Protocol (HTTP)

- Client and server establish TCP connection

- Client requests content

- Server responds with requested content

- Client and server close connection eventually

- Content is a sequence of bytes with an associated MIME (Multipurpose Internet Mail Extensions) type

- Content returned can be either static or dynamic

- Static content: stored in files and retrieved in response to an HTTP request

- Dynamic content: content produced on-the-fly in response to an HTTP request

- Web content is associated with a file that is managed by the server

- HTTP request is a request line, followed by zero or more request headers

- HTTP response is a response line followed by zero or more response headers, possibly followed by content

**PROXIES**