CSE320 System Fundamentals II Networking

TONY MIONE

Topics

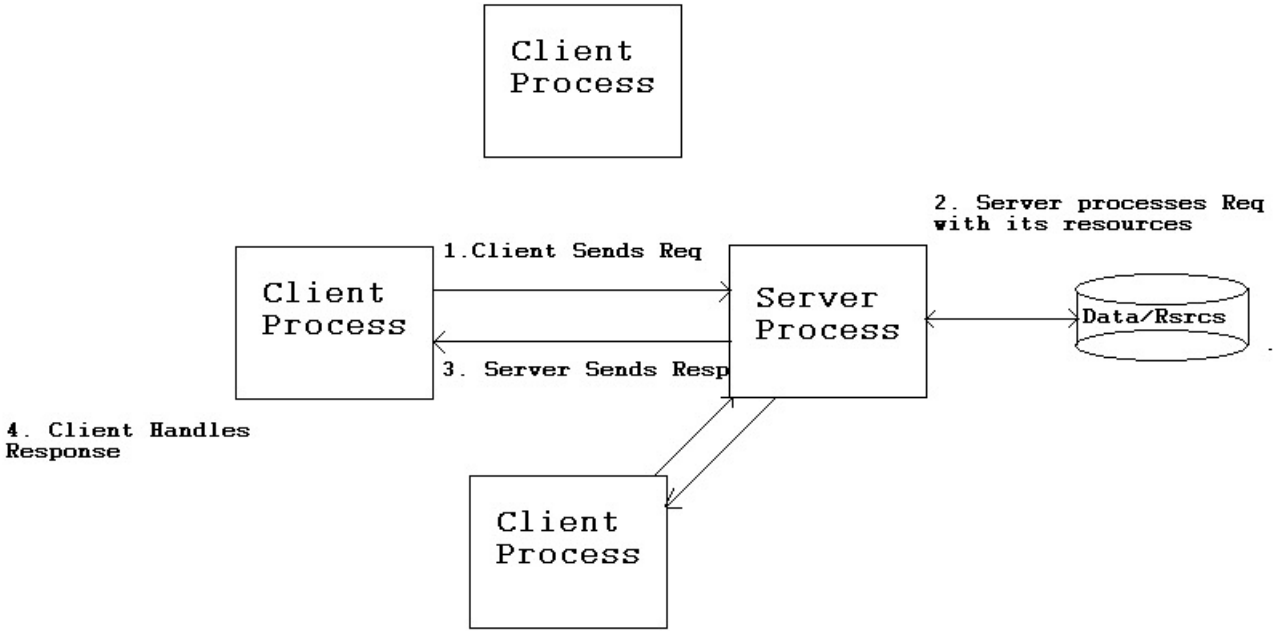
- Client/Server Transactions
- Computer Networks/Network Architecture
- Ethernet/Bridged Ethernet
- LANs/WANs
- Protocols
- Internet – connected networks
- Programmer's view of the internet
  - DNS (Domain Name Service)
  - IP addresses
- Structure of Internet Connections

Client-Server Model

Client-Server model

- Application: a server process and one or more client processes
  - Clients and servers are not machines (hosts) but processes
- Server process
  - Manages resources and provide services to clients
- Example
  - Web server, FTP server, email server

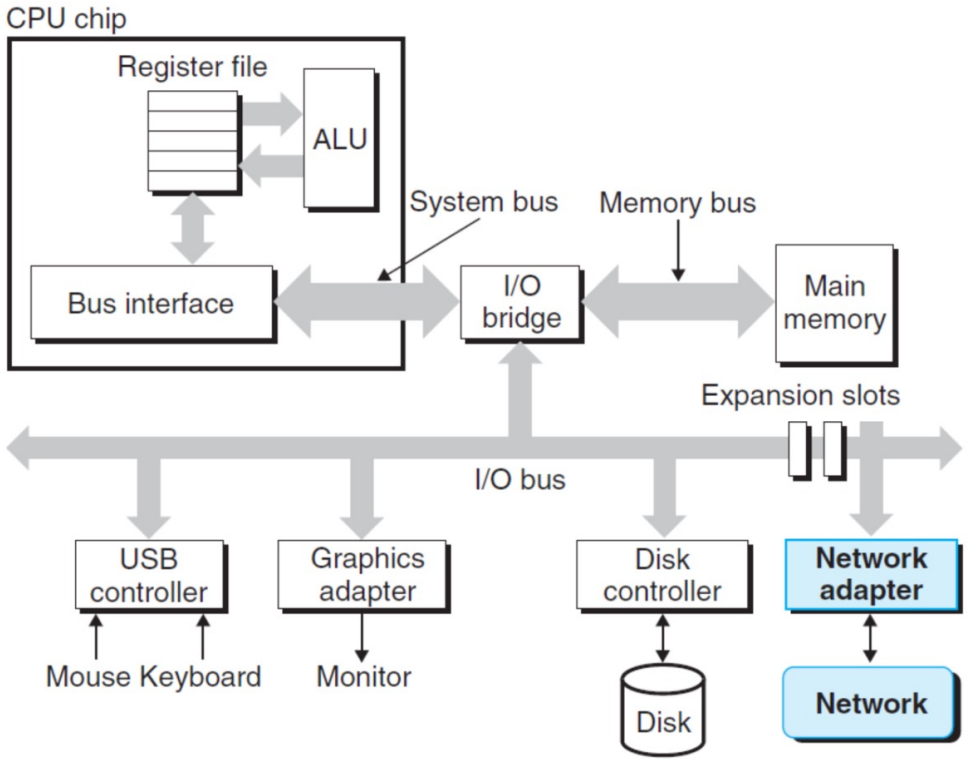
Client/Server Transactions



Servers handle requests from one or more clients
Server manages data and other resources
Server responds to requests from clients
Client and server are processes running on hosts (same or different)

Networks

- Network
  - To a host, a network is just another I/O device that serves as a source and sink for data



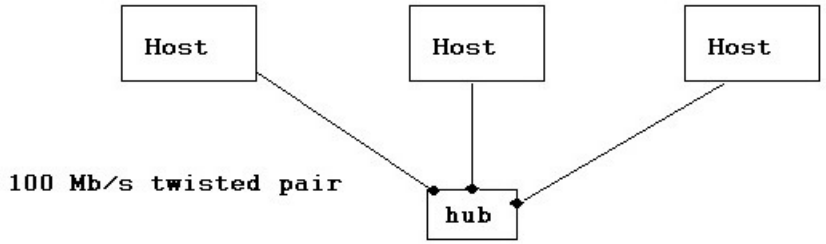
Networks

- Adapter
  - Physical interface to the network
  - Copy data between host and network

- Local Area Network (LAN)
  - Spans a building or a campus
  - Ethernet is the most popular technology used for a LAN

Ethernet

SAN – System Area Network – Spans room or floor
LAN – Local Area Network – Spans Building or campus
WAN – Spans country/world

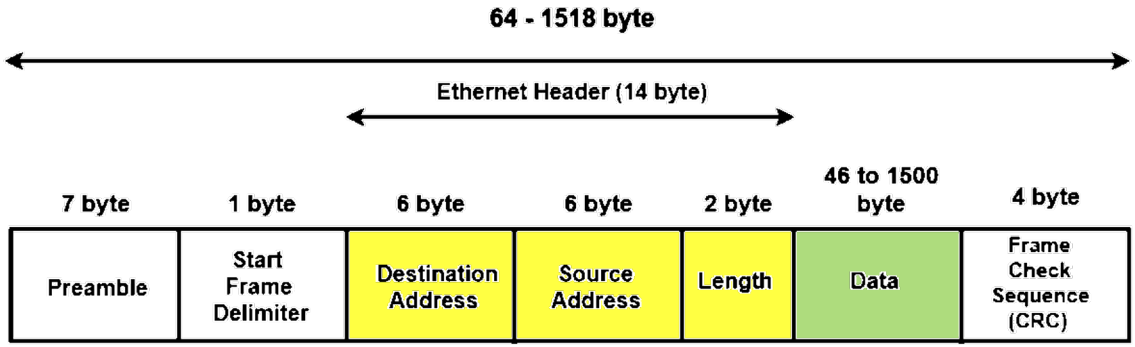


Ethernet segment consists of wires (twisted pairs of wires) and a hub
Hub – contains ports. Hosts connected to ports.
Hub blindly copies data received onto all ports
Hardware uses Ethernet protocol
Wires - Have the same maximum bit bandwidth 100 Mb/s, 1 Gb/s
Connects an adapter and a port on the hub

Networks (LAN: Ethernet)

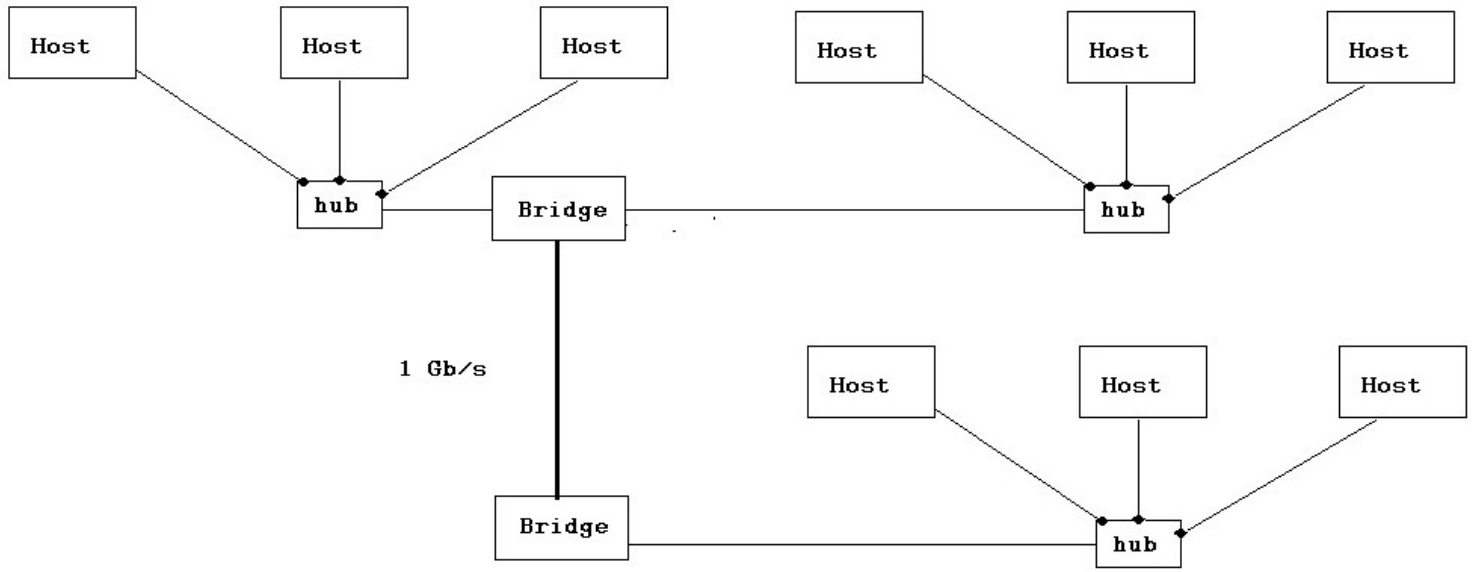
Ethernet segment

- Each adapter (hardware interface) has a unique 48 bit hardware (MAC or Media Access Control) Address: 00:20:be:e3:45:06
- A host can send a frame to any other host on the same segment
- Frame
  - Header: source/destination, frame length
  - Payload: data bits



IEEE 802.3 Ethernet Frame Format

Bridged Ethernet

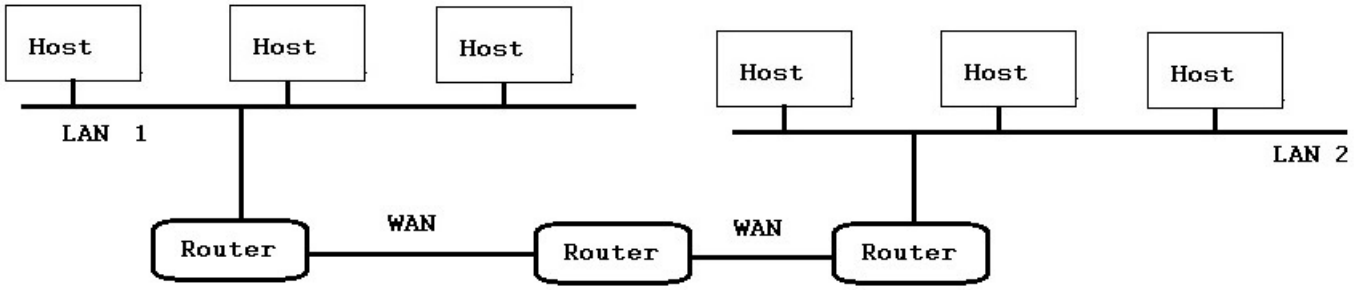


SANs/LANs connected together with Bridges.

Bridges add intelligence. Discover where hosts are (which port on the bridge leads to the host).

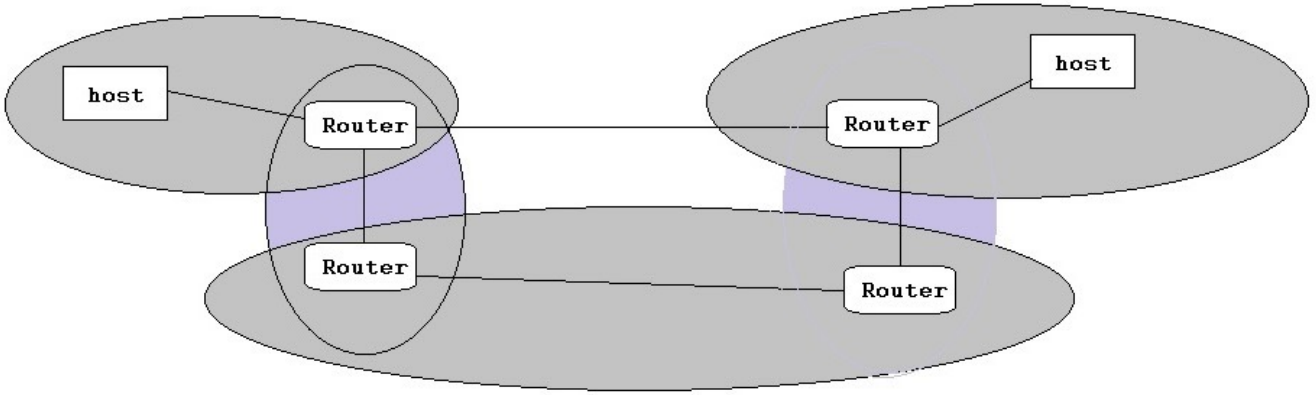
Wires on various ethernet segments may have differing bandwidths

internets



Multiple incompatible LANs can be connected via Routers
Connect WANs (Wide Area Networks, e.g. point-to-point phone connections)
internet (with small I - not The Internet)

Network Architecture



- Networks are connected in an Ad-hoc fashion
- No particular hierarchy or topology
- Different router and link capabilities
- Packets travel from a source to destination host by hopping through networks from router to router
- Different packets may use different routes

Network (internet)

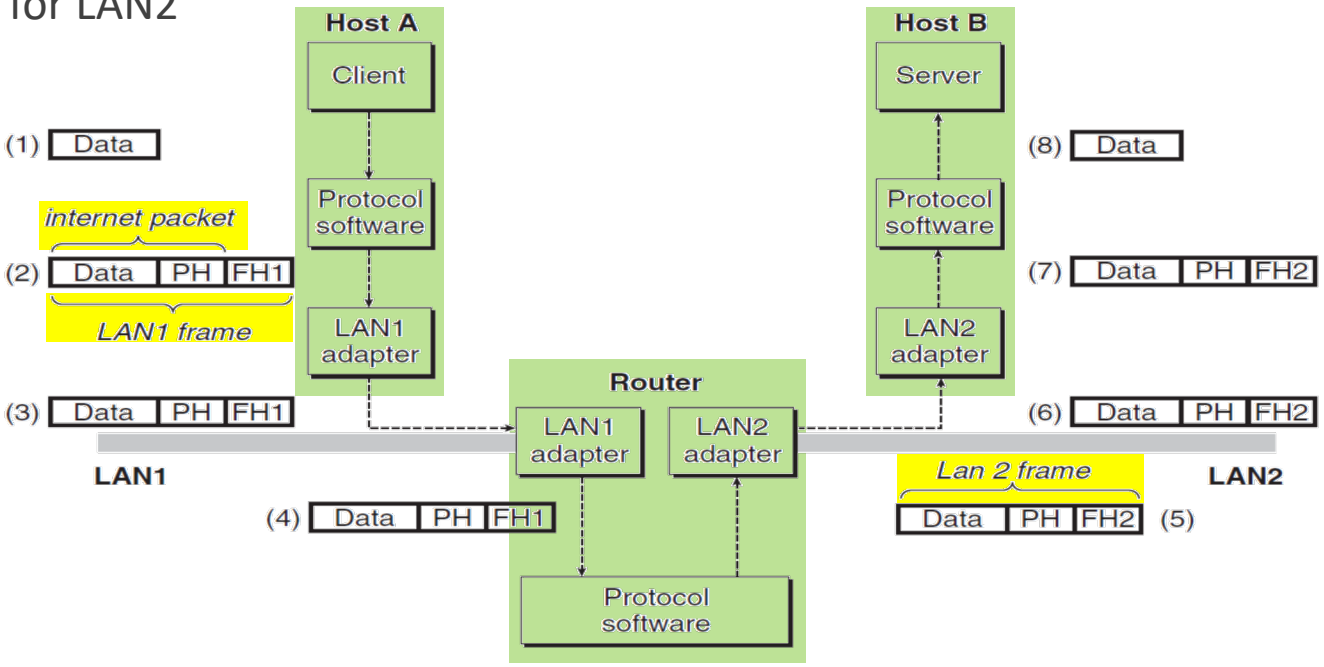
- Protocol software (runs on hosts and routers)
  - How a source host can send data to a destination host across incompatible networks?

- Naming scheme
  - Different LAN technologies have different ways of assigning addresses to hosts
  - internet protocol defines a unique format for host addresses

- Delivery mechanism
  - Packet: uniform way to bundle up data bits into discrete chunks
    - Header: source/destination address, packet size
    - Payload: data

Network (internet)

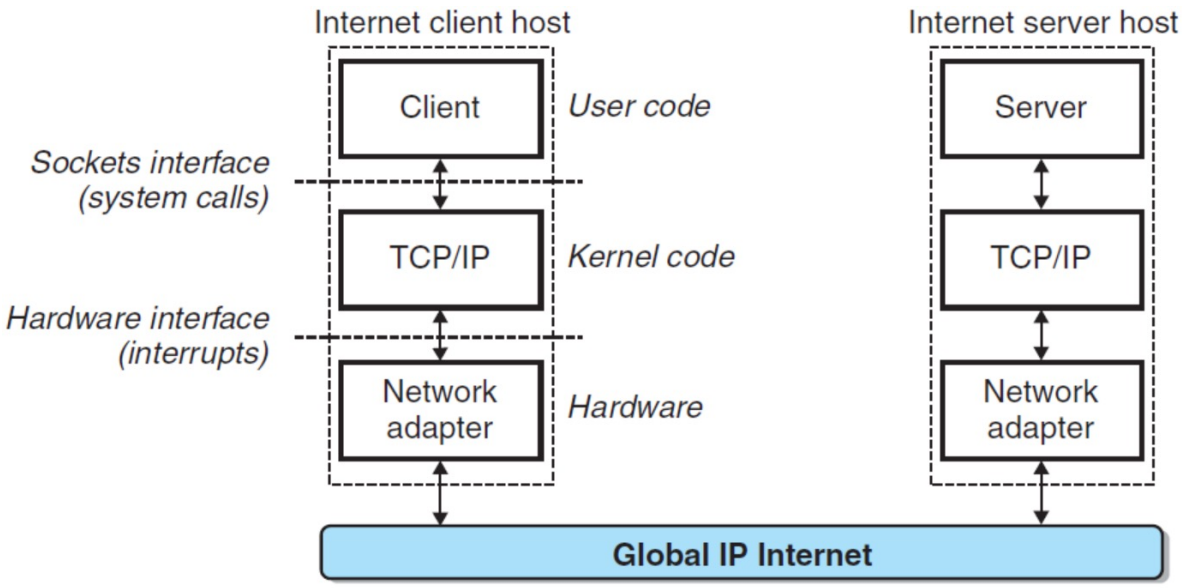
- How data travel from one host to another on an internet
  - PH: internet packet header
  - FH1: frame header for LAN1
  - FH2: frame header for LAN2



IP Internet

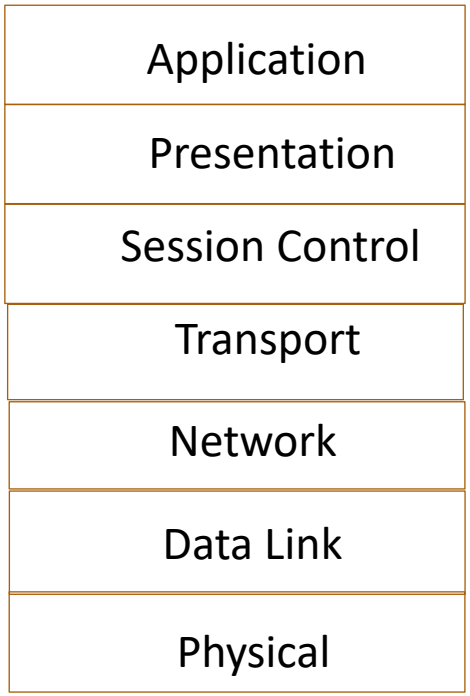
- Global IP Internet (with the capital I)
  - Most famous implementation of an internet
  - Protocols
    - IP (Internet Protocol): naming scheme and delivery mechanism for packets (called datagram)
    - UDP (User Datagram Protocol): packets can be lost or duplicated
    - TCP (Transmission Control Protocol): reliable bidirectional connection
  - Internet clients and servers communicate using
    - Socket interface functions
    - Unix I/O functions

IP Internet



- Hardware and software organization of an Internet application

ISO 7-Layer Model



User application services (email, file, web, etc.)

Machine dependent translation, Encrypt/decrypt, etc

(Depends on Application)

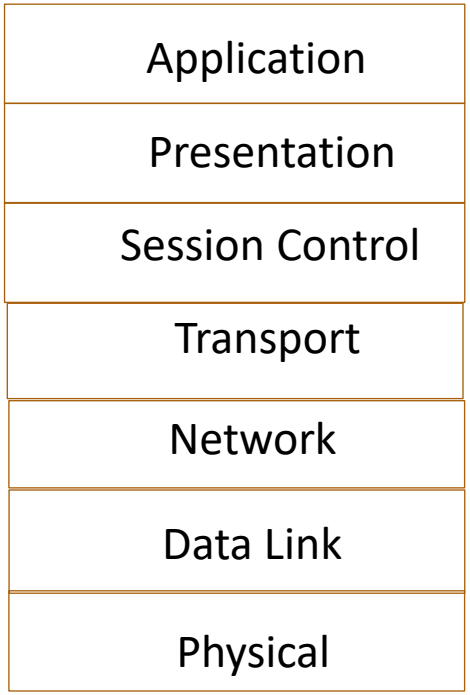
Process-to-Process Transfer

End-to-End Transfer (machine to machine)
Path determination (logical address/packet routing)

Data transfer to adjacent node (i.e. Ethernet, wifi)

Electrical/hardware (bit transmission)

ISO 7-Layer Model - Protocols



SMTP, NFS, HTTP, POP3, IMAP, etc.

[Part of Application layer]

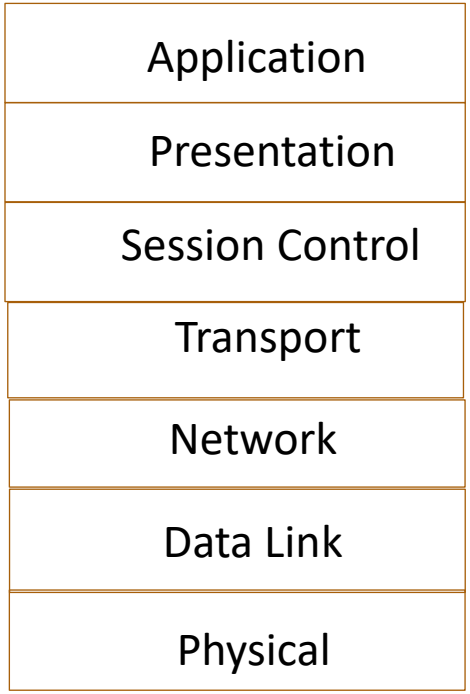
[Part of Application layer]

TCP, UDP

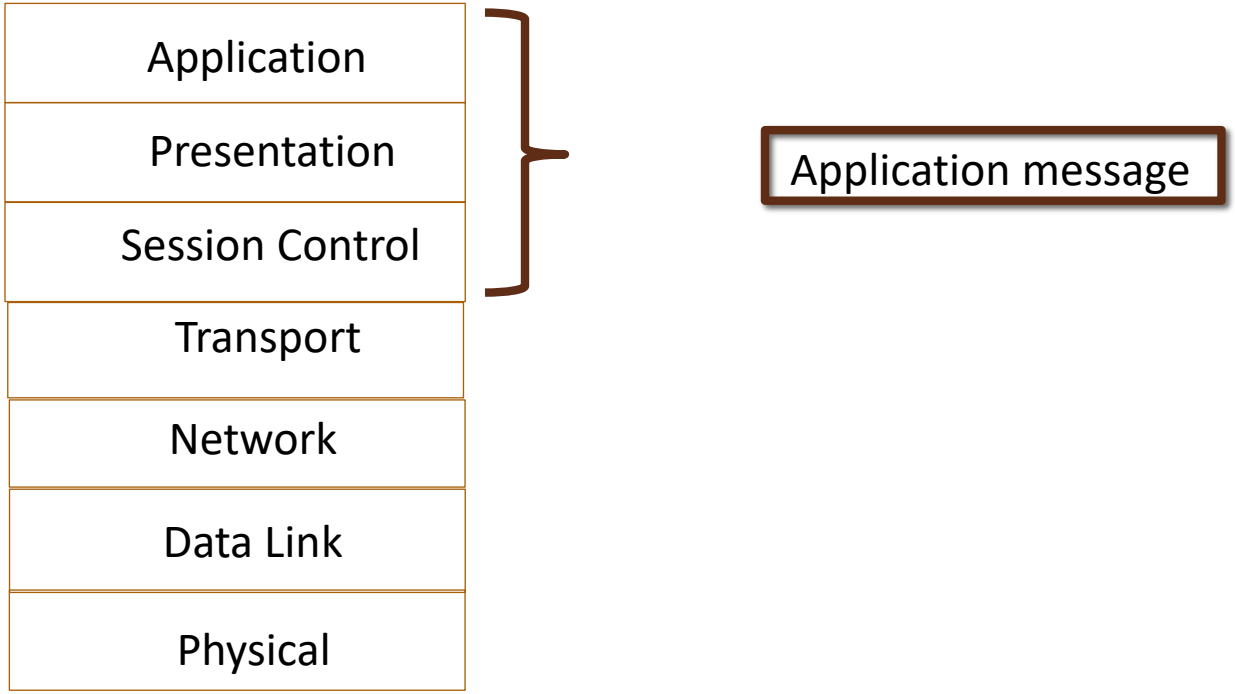
IP [Also: ICMP]

Ethernet, 802.3, 802.11(b/g/n/ac)

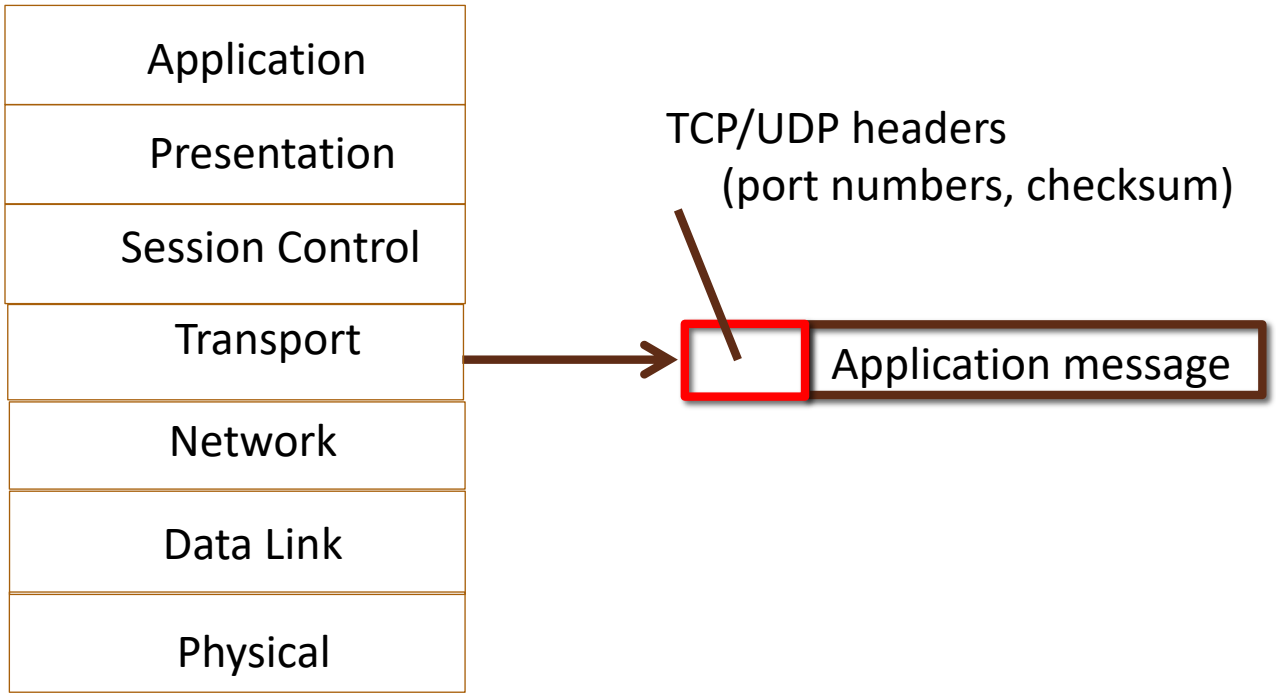
ISO 7-Layer Model – Headers



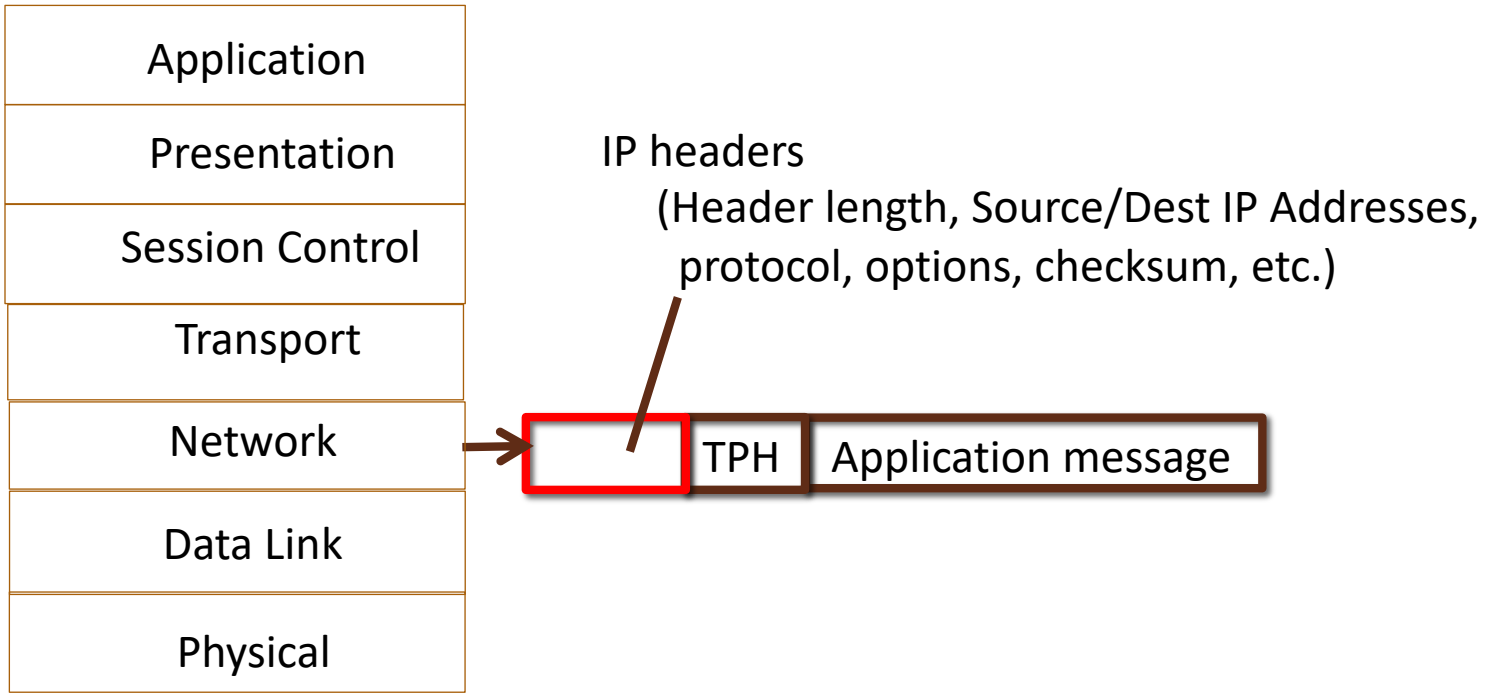
ISO 7-Layer Model – Headers



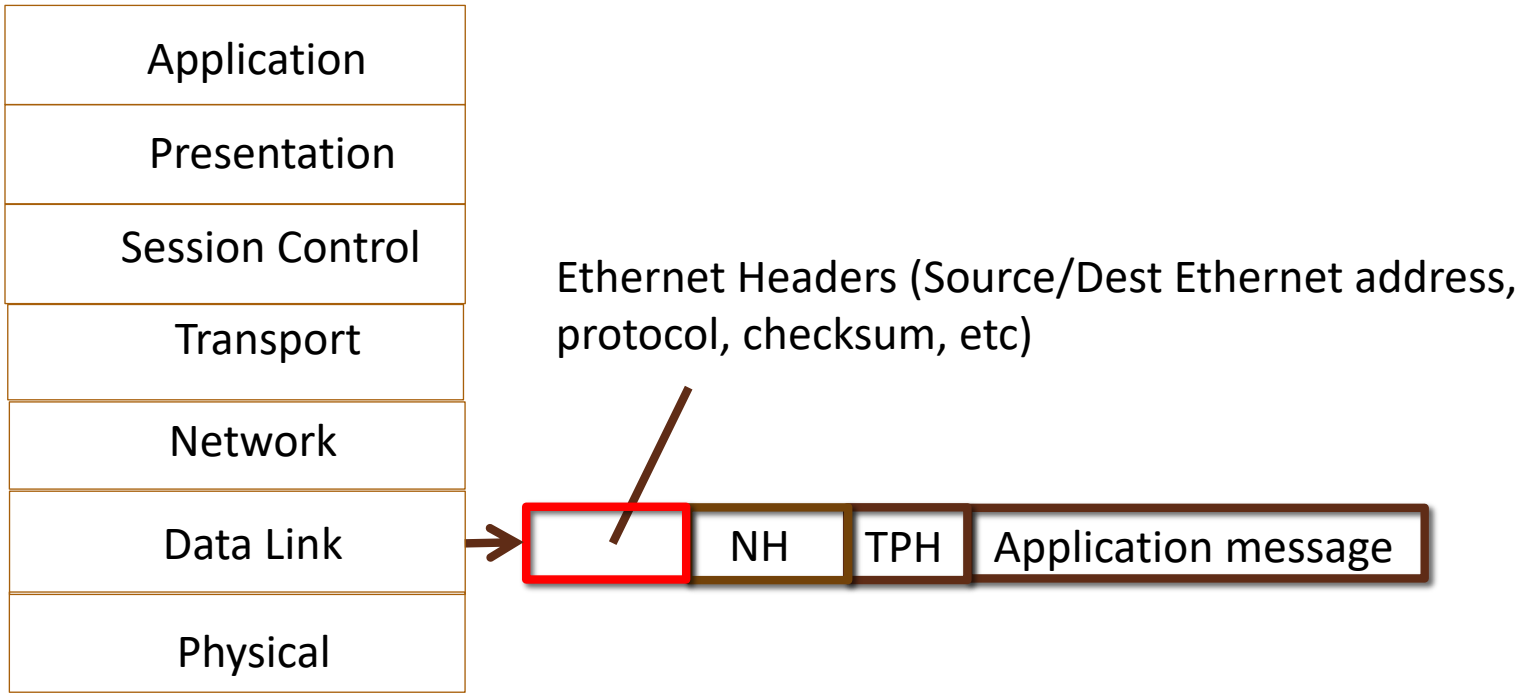
ISO 7-Layer Model – Headers



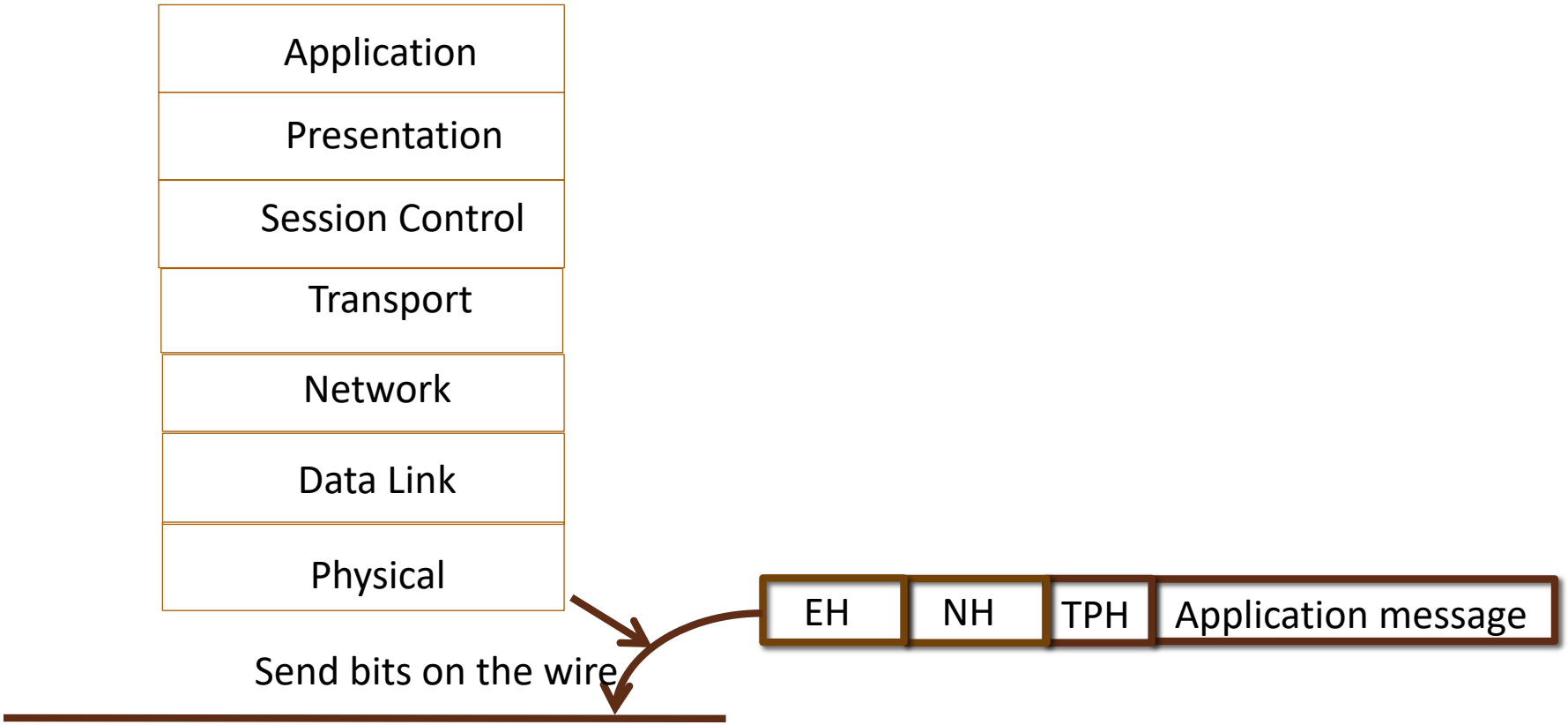
ISO 7-Layer Model – Headers



ISO 7-Layer Model – Headers



ISO 7-Layer Model – Headers



Programmer's View

- Programmers use host names and ports to contact services
- Name is easy to deal with, but machines prefer numbers
- DNS (Domain Name Service)
  - ~~Large~~ **Huge** distributed database of hosts
  - Provides translation of names to IP (Internet protocol) Addresses
- IP Addresses – 'Logical' address associated with a host
  - IPv4 – 32 bits – Usually written in 'dotted decimal form'
  - IPv6 – 128 bits – IPv6 is still growing but is not as widely adopted as v4
- Servers are assigned a port number ('Well known ports')
- Given an IP Address and Port, a program can create a connection to a service on the same or another host

IP Addresses

- IP address is an unsigned 32-bit integer
- Presented in *dotted-decimal* notation
- Stored in an *IP Address struct*

```
struct in_addr{
   uint32_t   s_addr; /* network byte order (big endian) */
};
```

- Addresses are stored in ‘network byte’ order (big endian)
- Other numbers placed into a protocol header are also stored in this order

Changing Byte Order

```
//IP address structure
struct in_addr {
    uint32_t s_addr; //Address in network byte order (big-endian)
};

#include <arpa/inet.h>

//Return values in network byte order
uint32_t htonl(uint32_t hostlong);
uint16_t htons(uint16_t hostshort);

//Return values in host byte order
uint32_t ntohl(uint32_t netlong);
uint16_t ntohs(uint16_t netshort);
```

Changing Byte Order (cont)

```
#include <stdio.h>
#include <arpa/inet.h>
int main() {
   uint32_t hl = 0x12345678;
   uint16_t hs = 0x1234;
   uint32_t nl = htonl(hl);
   uint32_t ns = htons(hs);
   unsigned char *p;

   p = (unsigned char*) &hl;
   printf("hl: %x %x %x %x\n", p[0], p[1], p[2], p[3]);
   p = (unsigned char*) &nl;
   printf("nl: %x %x %x %x\n", p[0], p[1], p[2], p[3]);
   p = (unsigned char*) &hs;
   printf("hs: %x %x\n",      p[0], p[1]);
   p = (unsigned char*) &ns;
   printf("ns: %x %x\n",      p[0], p[1]);
}
```

Output:

hl: 78 56 34 12

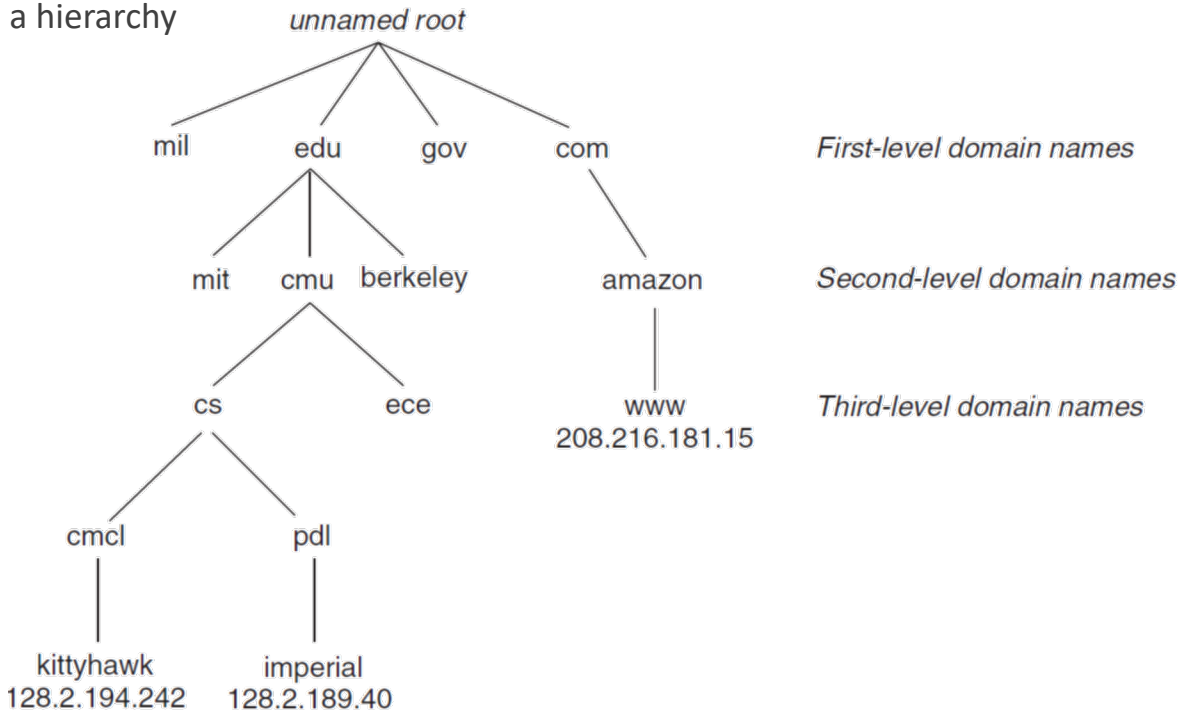
nl: 12 34 56 78

hs: 34 12

ns: 12 34

IP Internet

- Internet Domain Names
  - Human friendly name instead of large integers
  - Mechanism to map domain names to IP addresses
    - Set of domain names forms a hierarchy



DNS

- DNS maps hostnames to ip address
- Also maps names and addresses of special service machines
  - Name servers
  - Mail exchange hosts

The ***nslookup*** application makes queries via DNS and reports results

DNS - Examples

```
cse320@cse320-VirtualBox: nslookup
> localhost
Server:         127.0.1.1
Address:  127.0.1.1#53

** server can't find localhost: NXDOMAIN
> www3.cs.stonybrook.edu
Server:         127.0.1.1
Address:  127.0.1.1#53

Non-authoritative answer:
Name:   www3.cs.stonybrook.edu
Address: 130.245.27.3
> exit
```

Structure of Internet Connections

- A Connection is a relationship between two processes on the same or different hosts. Properties include:
  - ***Point-to-point:*** The connection connects a pair of processes
  - ***Full-duplex:*** Data flows in both directions simultaneously
  - ***Reliable:*** Data sent by the source host is guaranteed to reach the destination host (eventually) without errors and in the correct order.

Structure of Internet Connections (cont)

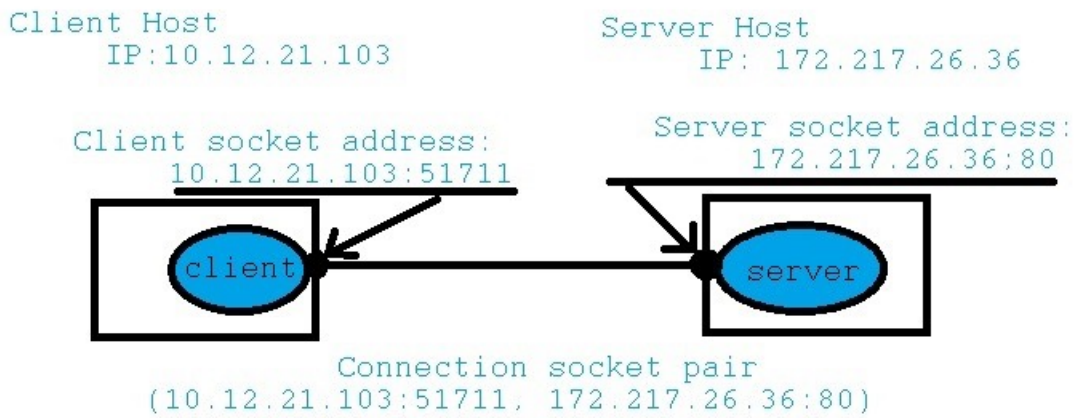
- A ***socket*** is the endpoint of a connection
  - Integer value similar to a file descriptor in unix
  - Internally is made up of IP address:port
    - **IP Address** is the address of the host on which the socket is open
    - **Port** is a 16 bit number related to the process using the socket
      - **Ephemeral port** (temporary, assigned number) – Assigned by kernel where client runs
      - **Well-known port** (Fixed, associated with a specific service) – Well-known ports are standardized and are kept in a configuration file on the server host (80-web server, 25-mail transfer server)

Well known ports are listed in /etc/services (on Unix systems)

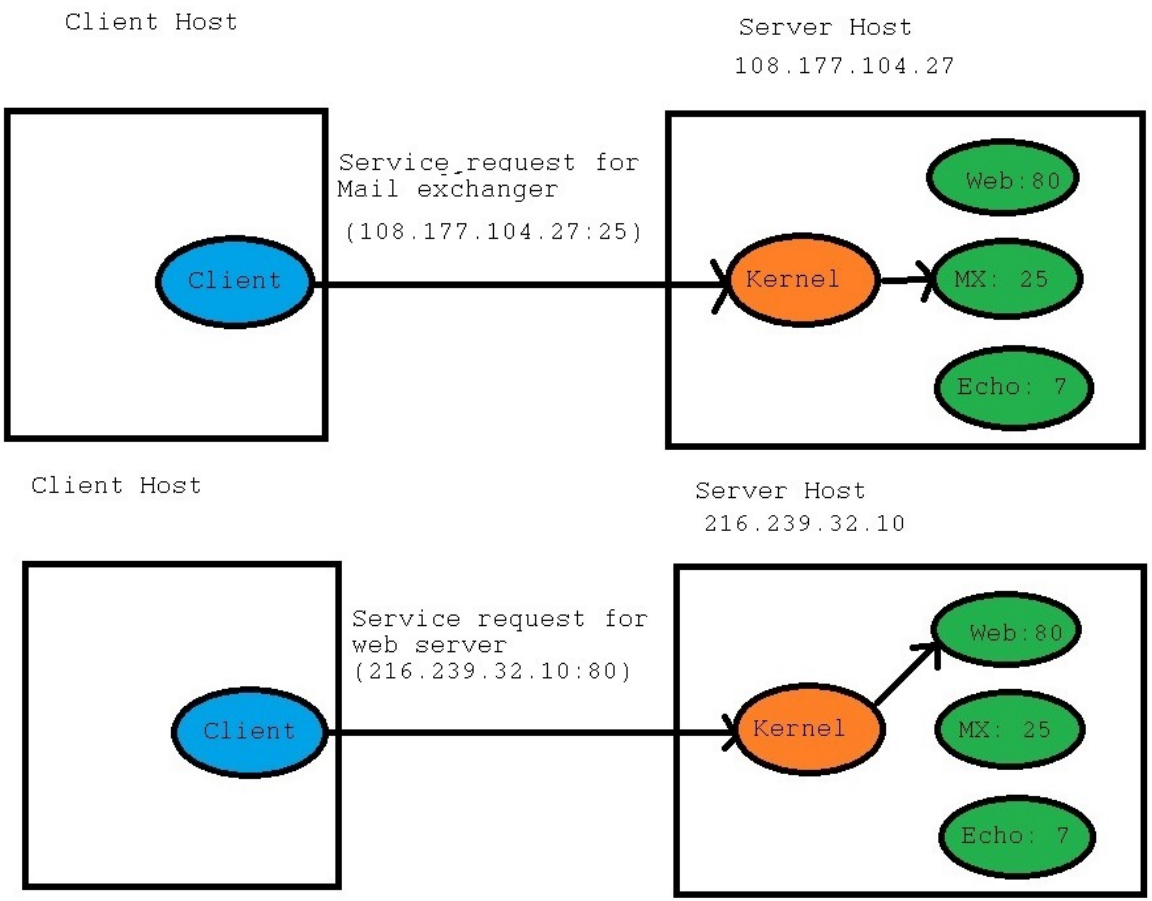
```
$ cat /etc/services
...
ftp             21/tcp
ssh             22/tcp                          # SSH Remote Login Protocol
ssh             22/udp
telnet          23/tcp
smtp            25/tcp          mail
...
http            80/tcp          www             # WorldWideWeb HTTP
```

Connection socket pair

A Connection is uniquely identified by its endpoint socket addresses (i.e. **socket pair**)



Ports identify services



Sockets

- Kernel perspective => Endpoint of a communication
- Program perspective => file descriptor to read from/write to network
- Socket interface is a set of system level functions combined with Unix I/O that allow the construction of network based applications.
- Created in early 80's and distributed with Berkeley Unix

Questions?