CSCI 4210 OPERATING SYSTEMS

Network Programming
David Goldschmidt (goldsd3@rpi.edu)
Summer 2023 (U23)

OSI REFERENCE MODEL

Open Systems Interconnection (OSI) Reference Model (~1984)

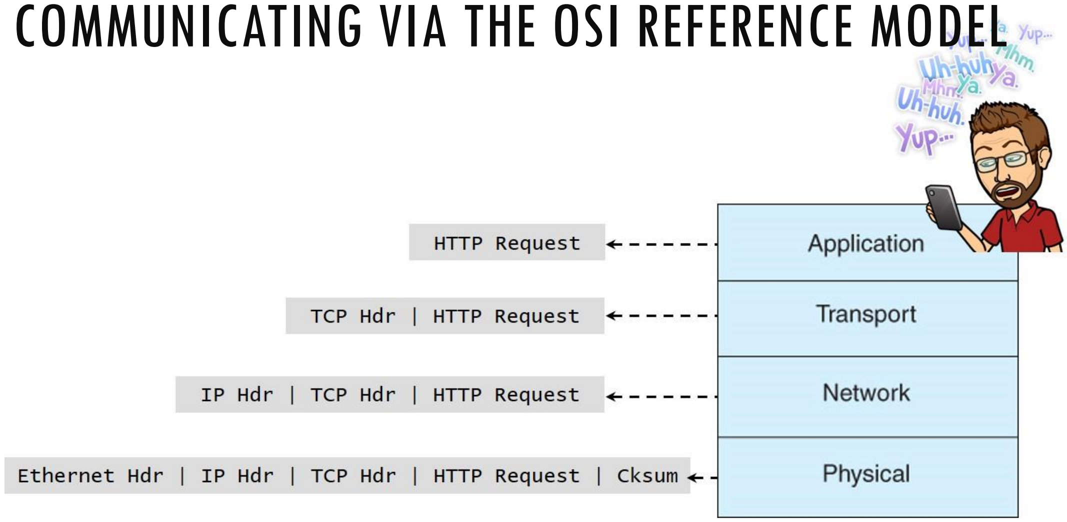
Standardization of how communication should occur across a network, describing where and how network protocols fit together with one another

A seven-layer protocol stack that supports interoperability of networking components:

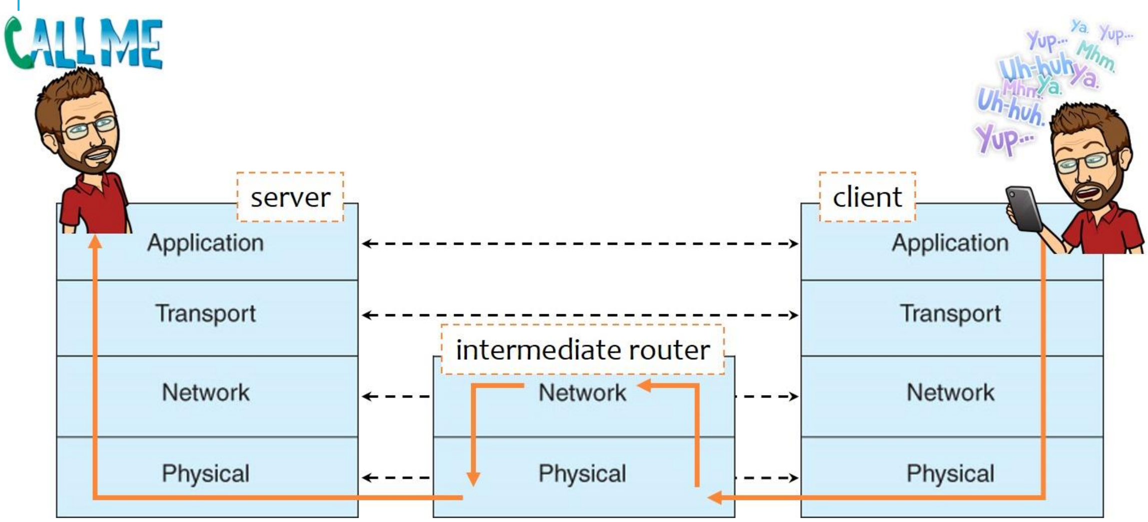
- Layer 7: Application (e.g., HTTP, HTTPS, NFS, SMTP, SNMP, TELNET)
- Layer 6: Presentation (e.g., SSL, FTP, SSH)
- Layer 5: Session (e.g., RPC)
- Layer 4: Transport (e.g., TCP, UDP)
- Layer 3: Network (e.g., IP, ICMP, ARP, OSPF)
- Layer 2: Data Link (e.g., MAC)
- Layer 1: Physical (e.g., Ethernet, Frame Relay, IEEE 802.11)

[HTTP: http://www.ietf.org/rfc/rfc2616.txt]

COMMUNICATING VIA THE OSI REFERENCE MODEL



COMMUNICATING VIA THE OSI REFERENCE MODEL



IP DATAGRAM STRUCTURE

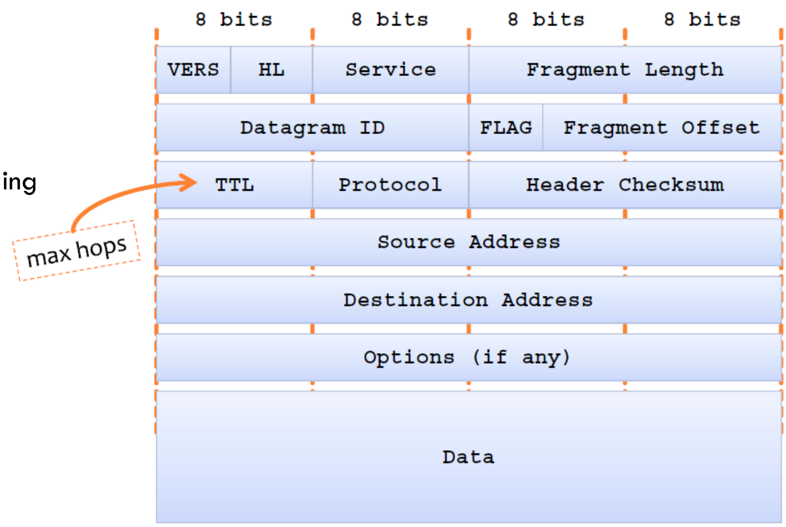
https://tools.ietf.org/html/rfc1122

Internet Protocol (IP):

Connection-less

Unreliable (i.e., no re-sending of dropped datagrams)

Provides host-to-host delivery of datagrams

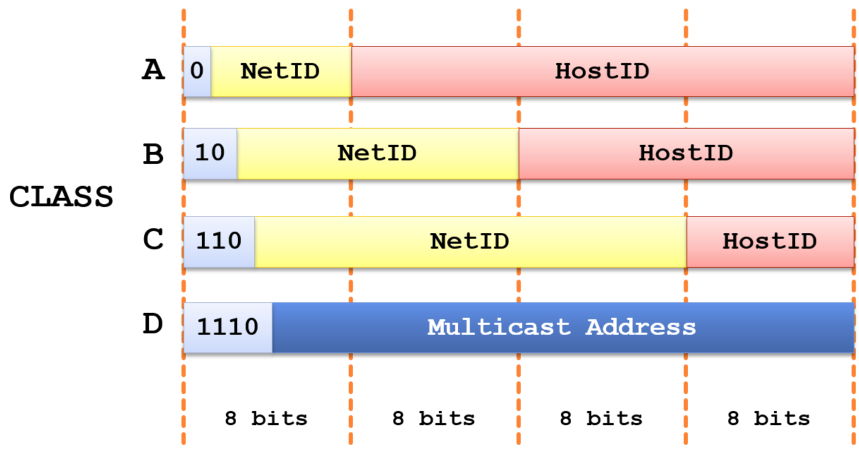


IP ADDRESSES

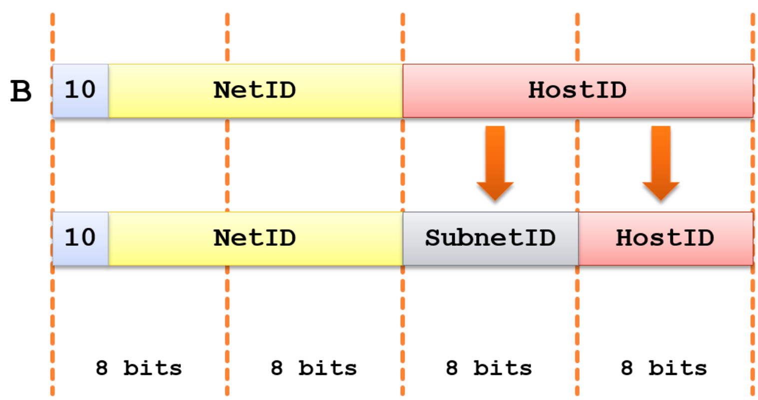
Each IP address contains information about what network the destination host is on, which enables routing to occur at intermediate "hops" (i.e., routers) along the path from a source endpoint to the destination endpoint

NETWORK CLASS	LEADING BITS	# of NETWORKS	# of HOSTS	NETWORK/HOST BIT FIELDS
CLASS A	0...	128	16,777,214	8 / 24 bits
CLASS B	10...	16,384	65,534	16 / 16 bits
CLASS C	110...	2,097,152	254	24 / 8 bits
CLASS D MULTICAST	1110...	n/a	n/a	n/a

DECODING IP ADDRESSES



SUBDIVIDING INTO SUBNETS



UDP DATAGRAM STRUCTURE

https://tools.ietf.org/html/rfc768

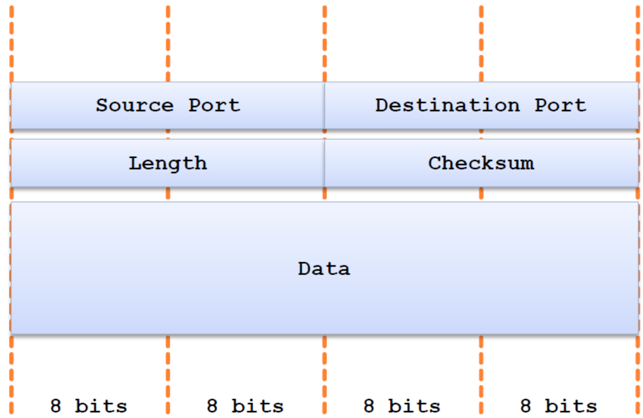
User Datagram Protocol (UDP):

Connection-less

Unreliable (i.e., no re-sending of dropped datagram)

Simple

Low overhead



TCP SEGMENT STRUCTURE

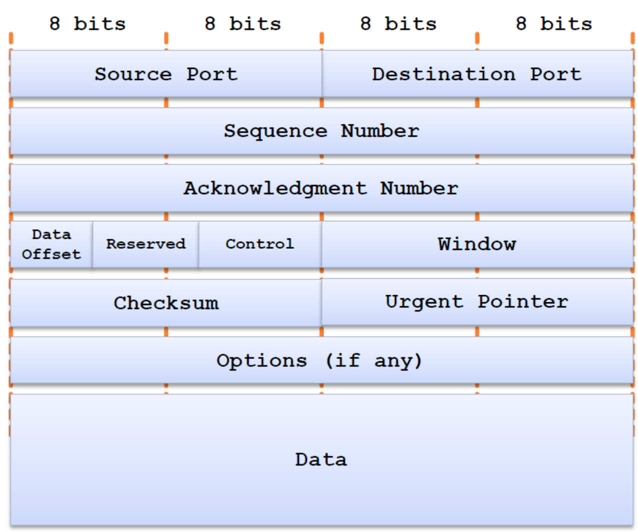
https://tools.ietf.org/html/rfc793

Transmission Control Protocol (TCP):

Connection-oriented

Reliable (i.e., re-sending of dropped packets, sequencing and reordering of packets, general error checking)

Overhead



INTER-PROCESS COMMUNICATION (IPC)

Inter-process communication (IPC) requires the following:

1. Synchronization
2. Protocol (i.e., how is communication to occur between the endpoints?)
3. Precision
4. Data marshalling (i.e., translating from “host format” to “network format”)

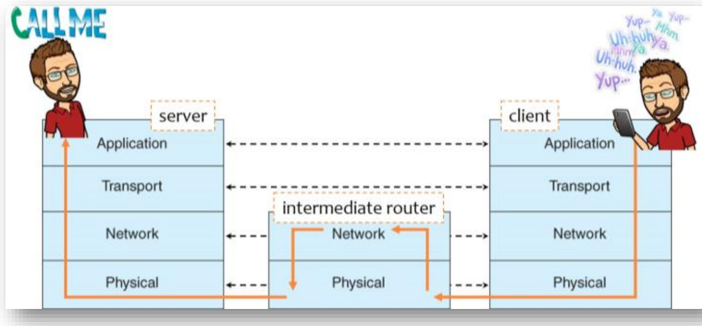
SOCKETS-BASED COMMUNICATION

A socket is an endpoint for communication, so (at least) two endpoints are required

A server process creates one or more sockets that it then listens on for incoming connection requests or incoming datagrams

Server processes listen on specific port numbers, which are 2-byte values

Sockets-based communication can be connection-oriented or connection-less



DATA MARSHALLING

How can we make sure that the remote recipient endpoint correctly interprets data that we send?

For example, what date does 04/01/2010 represent?

And with multi-byte data types (e.g., int, double, etc.), in which order do we send the bytes?

Big endian stores the most significant byte (MSB) first, i.e., in the lowest memory address

Little endian stores the least significant byte (LSB) first, i.e., in the lowest memory address

Network format (i.e., network byte order) is standardized to use big endian

- see htons(), ntohs(), htonl(), ntohl(), etc. (also try "man endian")