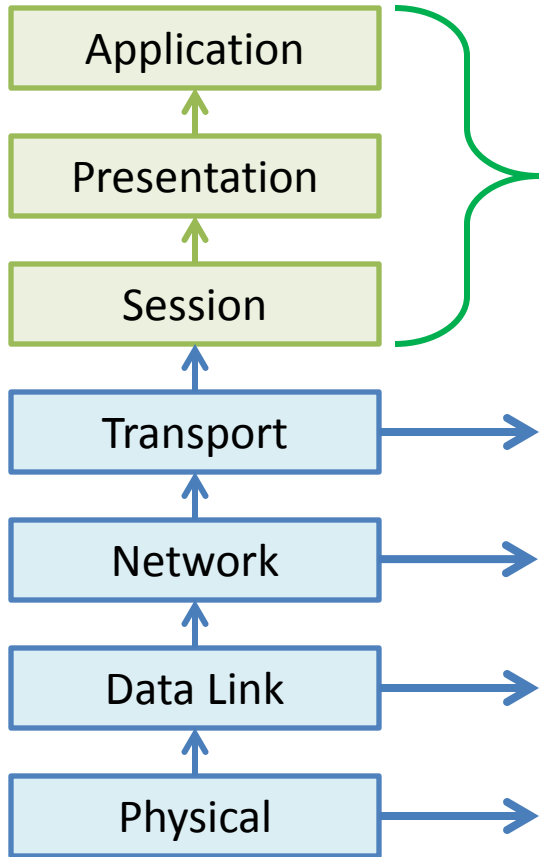


Networking II

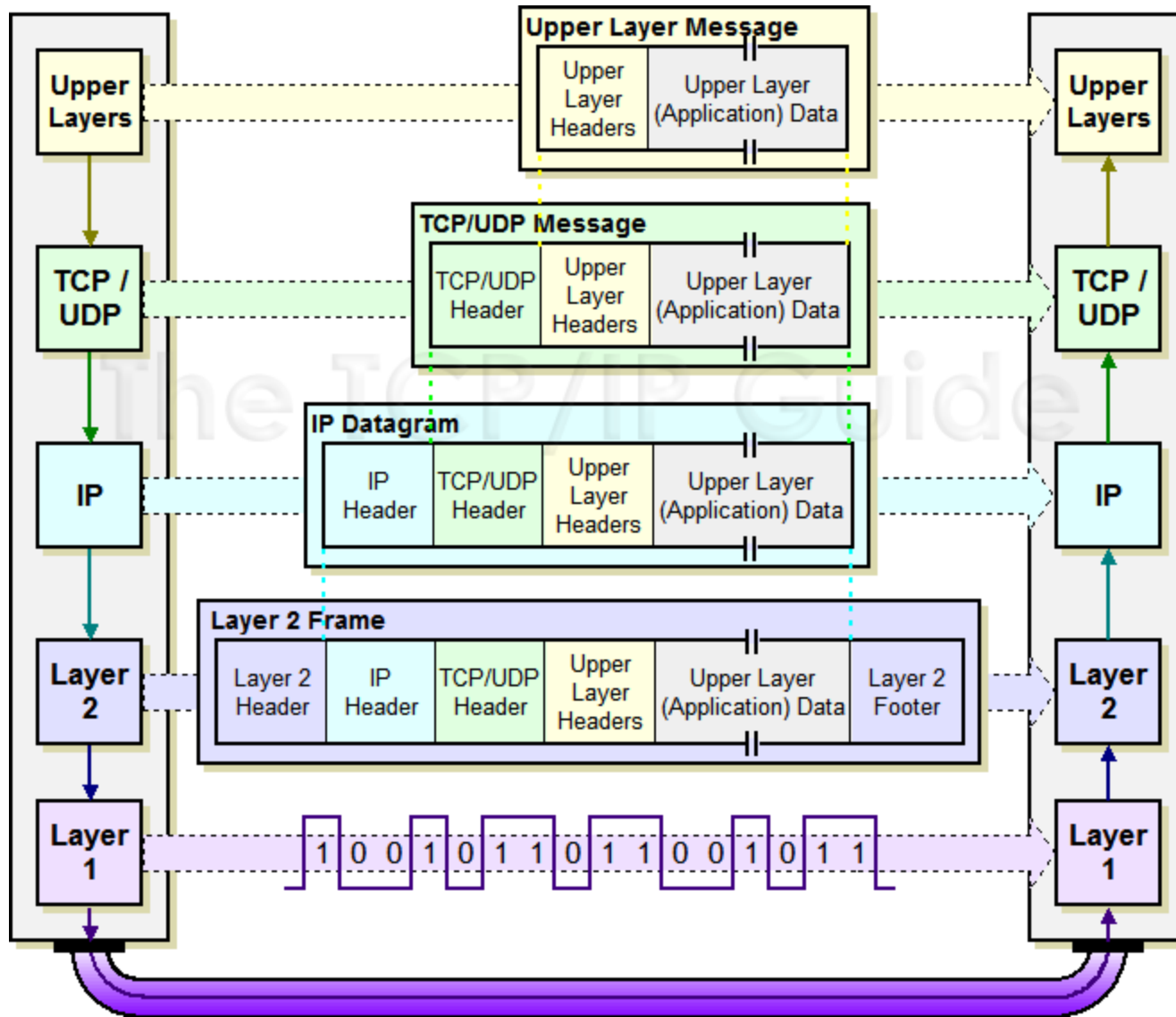
CS 241

Nov. 11, 2013

OSI Protocol Stack / “OSI Model”



Network Packet Encapsulation



Understanding IP

- The network layer provides “host-to-host” connectivity.
 - In IP, done via **IP Addresses**
 - Globally unique 32-bit numbers
 - Usually written as four 8-bit integers: **127.0.0.1**
 - **IPv6**: 128-bits, written as eight sets of 16-bit hexadecimal numbers
(ex: 2001:0DBB:AC10:FE01:0000:0000:0000:C3D4
== 2001:0DBB:AC10:FE01::C3D4)
 - IP addresses are hard to remember!
 - **Domain names** associate easy-to-remember names that can be translated to IP addresses via the DNS protocol.

Understanding TCP

- TCP Provides

-

-

-

-

-

-

-

- TCP Doesn't Provide

-

-

-

-

-

*...while the **session** is active.*

Understanding UDP

- UDP Provides

-

- UDP Doesn't Provide

-

-

-

-

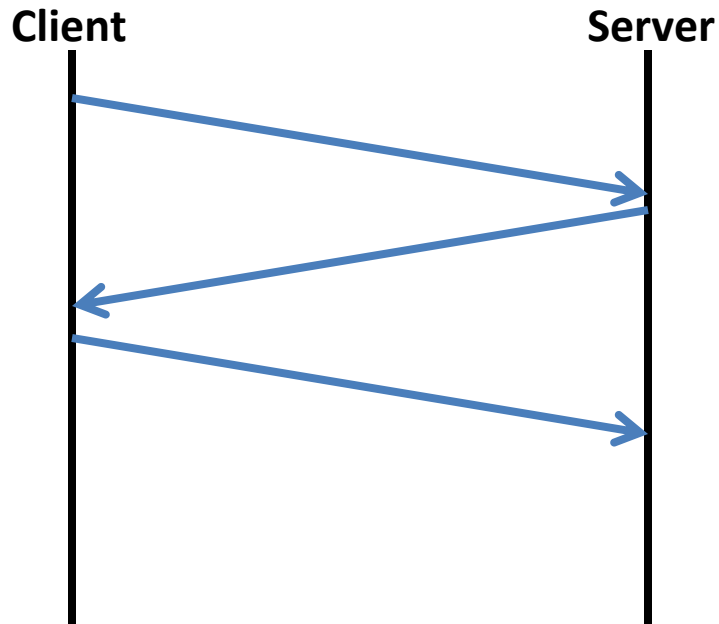
-

-

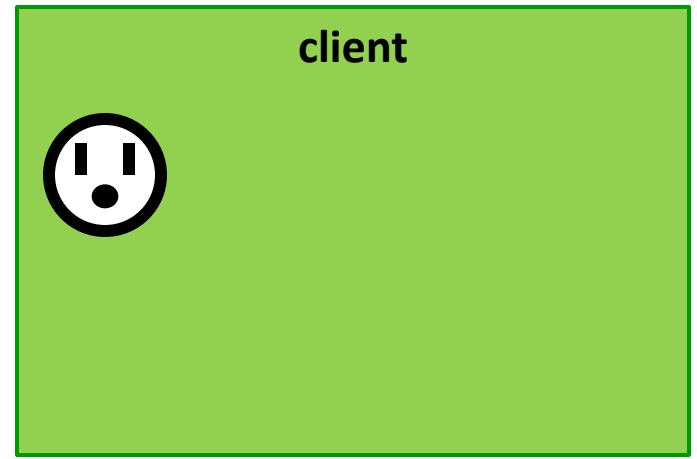
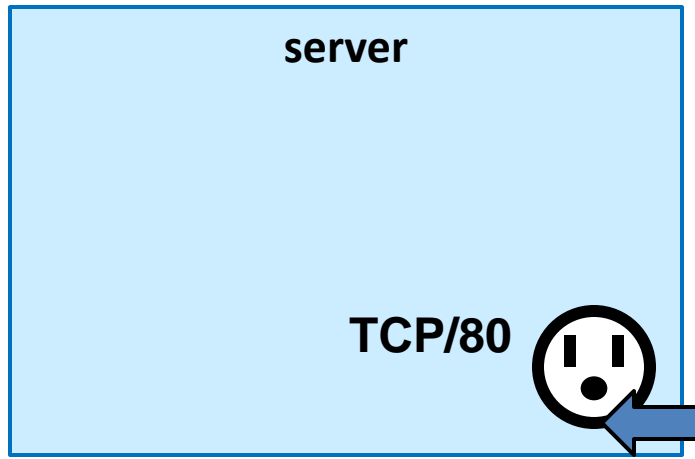
-

TCP Session

- When a client connects to a host on TCP, a “**TCP session**” is initiated.
 - Requires a **three-way handshake** before any data can be sent on the TCP socket.



TCP Sessions



Network Vocabulary

- **Socket Address**

- Complete identification of the socket you are connecting to. Made up of three pieces:

- Protocol (ex: TCP)
 - Network Address (ex: 127.0.0.1)
 - Port Number (ex: 80)

- **Port Number**

- Globally shared system resource, 16-bit integer
 - A port number can only be used by one process at a time on the entire system

Network socket

- A network socket is stream-based IPC.
 - Similar to a pipe:
 - Uses the file descriptor interface
 - Is stream-based, not segment- or message-based
 - Different from a pipe:
 - The file descriptor is bi-directional (read and write)
 - Reliability based on the transport protocol used
 - Special type of “server socket” that listens for incoming connections from remote hosts and **does not** transmit any application data!

Creating a network socket

socket () : Create an endpoint for communication

```
int socket(int network_protocol,  
          int transport_protocol,  
          int sub_protocol)
```

IP: AF_INET

IPv6: AF_INET6

TCP: SOCK_STREAM

UDP: SOCK_DGRAM

Setting up a server socket

getaddrinfo () : network address translation

- Translates a hostname (IP address or domain name), port, and protocol into a **socket address** struct.

bind () : binds an socket address to a socket

- Required in order to know what port number your socket will be listening for new connections

listen () : places the socket in a listening state

Using Sockets

accept () : accept a communication on a socket

```
int accept(int sockfd,  
          struct sockaddr *addr,  
          socklen_t *addrlen);
```

connect () : initiate a connection on a socket

```
int connect(int sockfd,  
           struct sockaddr *addr,  
           socklen_t *addrlen);
```