**网络应用程序的核心：**

run on different end system.

communicate over network

e.g., web server software communicates with browser software

**两种应用程序体系结构**

client-server

peer-to-peer

**client-server**

server:

主机总是打开（服务器）

固定的IP地址

data centers for scaling（扩展）

clients:

与服务器通信

may be intermittently connected

IP地址动态分配

互相之间不能直接通信

**peer-to-peer**

没有一直打开的主机

任意端系统之间可以直接通信

每个peer互相提供和请求服务

self-scalability（自扩展性）:新的peer带来新的service capacity，也带来新的服务需求

peers are intermittently connected and change IP addresses

管理复杂

**进程通信**

Who send/recv msg to/from network? Processes

Where does process send/recv msg to/from? socket

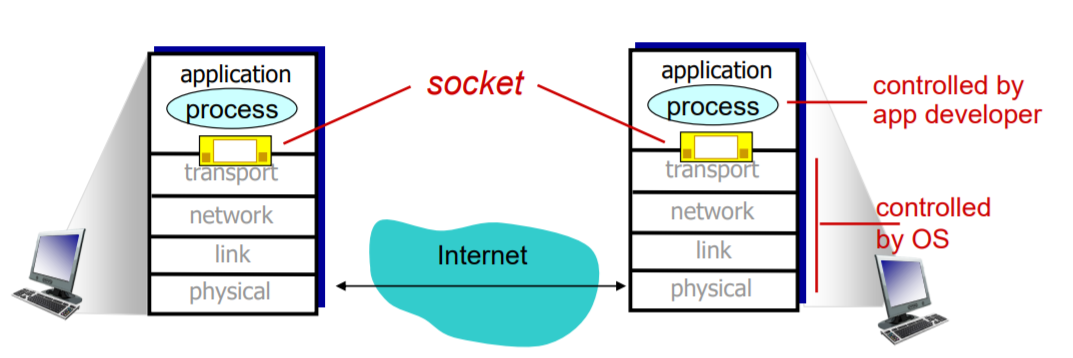
客户进程和服务器进程：

发起会话的是客户进程

在会话开始前等待连接的是服务器进程

**Socket 套接字**

进程通过套接字发送、接受messages



**IP和端口号**

IP是用于标识主机的32比特的唯一的量。

除了知道报文发送的目的地的主机地址外，发送进程还必须指定运行在接受主机上的接受进程（接受套接字），因为一台主机可以运行多个网络应用。

端口号（port number）被用于这个目的。

**app应用需要的运输服务**

可靠的数据传输 data integrity

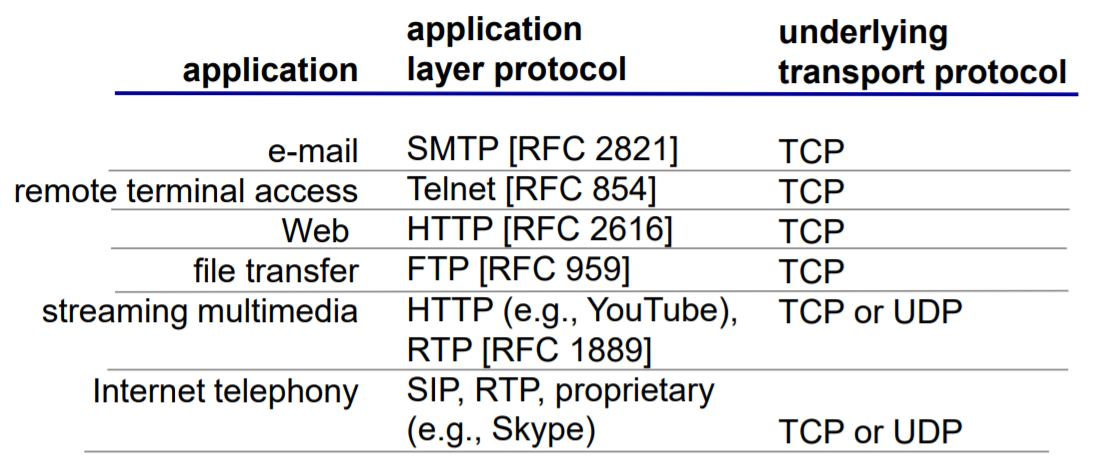
吞吐量 throughput

定时 timing

安全性 security

**因特网提供的服务**

TCP, UDP



**应用层协议**

