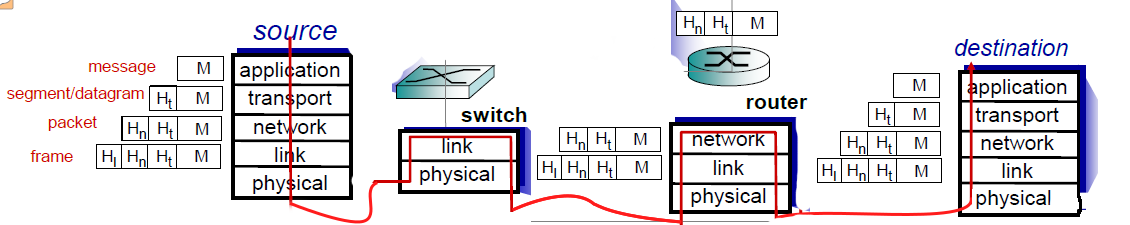
***Chapter 1*  
What is the internet?** The internet is a system of connected devices called hosts or endsystems. The endsystems use hooks that allow sending and receiving data to the internet.  
**What is a protocol?**  
 Protocols are rules that define format, and order of msgs sent and received among network entities. As well as actions taken on msg xmission and receipt.  
**Network Edge** The network edge consists of the clients and servers.   
**Network Core** The system of interconnected routers and the network of networks.  
**Packet Switching vs Circuit Switching** In packet switching hosts break application-layer messages into packets. These packets are forwarded from one router to the next. Each packet is transmitted at full link capacity. Packet switching: store and forward. Entire packet must arrive at router before it can be transmitted. 2L/R assuming zero propagation delay. Packets form a queue. Packets can be dropped. Routing is the determination of the source destination. While forwarding moves packets input to appropriate router output.   
 In Circuit switching end-end resource allocated to, reserved for “call” between source and destination.   
 Packet switching allows more users to use network. Packet switching is good for burst data. Congestion is possible in packet switching. Circuit Switching has FDM and TDM.   
**Loss, delay, throughput** Packets are lost if the queue/buffer reach max capacity. The delay is generated from the queuing, and transmission of these packets. Packet delay = processing + queue + transmission + propagation (wire travel). Throughput is the rate at which bits are transferred between sender/receiver. Instantaneous vs average. Throughput is as fast as the slowest/narrowest bottleneck  
**OSI Model** Application→Presentation→Session→Transport→Network→Data-Link→Physical. (Presentation and Session) can be left out. Why layers? Complex systems, and pieces modularize and easy the maintenance and updates to the system.  
**Encapsulation** The process of attaching and removing headers for the message as it is passed thru routers and switches from source to destination.   
 ****

***Chapter 2*Client vs P2P** Servers are always on hosts. And all clients communicate with the server. In P2P Architecture, there is no always on host. Clients connect directly to one another.   
**Sockets** processes send and receive message to/from its socket. A socket is between the application layer and the transport layer.  
**TCP/UDP** TCP offers reliable transport as well as flow control, congestion control but does not provide timing, and thru put guarantee, nor security. UDP is unreliable, and does not offer anything. It is barebones shoot and forget transport.  
**Persistent vs Non-persistent HTTP Connections** Non at most one object can be sent over TCP connection. The connection is then closed. Multi Objects means multi connections. Persistent is polar opposite. Multiple objects can be sent. The Overhead for connections is lost in persistent. One connection. Many objects.  
**Web Caches** In web caches, the ISP or local server contains the web cache. This reduces response time on the end user. And reduces traffic on the institution’s access link.  
**FTP** Port 21. USER,PASS,LIST,RETR,STOR {331 OK, 125 connect already open,425 can’t open connect. 452 error writing}  
**SMTP** TCP. Port 25. Handshake, transfer,close. ASCII Text. 7-bit. Mail Serv to Mail Serv. HELO,MAIL FROM, RCPT TO, DATA, QUIT single period CRLF.CRLF ends the message. SMTP Pushes.  **POP** Post Office Protocol. Download and delete. No Stateless, and no organization of the message. List,retr,dele,quit **IMAP** Internet Mail Access Protocol. Everything POP doesn’t have.   
**DNS** Domain Name System. Human Readable to IP. .com/.org/.net all have a “DB” that is used for lookup. A = hostname,value=IP;CNAME=name is canonical; NS=name is the domain, value is hostname; MX=mailerserver associated with name  
**P2P** D\_c-s >= max {NF/u\_s,F/d\_min} where N is file copies, d\_min is min client d/l rate and F/d\_min is max client download time. With P2P, we are fast. We are only as slow if little clients are connected.   
**Socket Programming** A socket is the “door” or connector from the application level, process, to the transport layer. UDP uses datagrams, and is a fire and forget style message sending. TCP is reliable, stream open connection messaging system. UDP : sendto(), SOCK\_DGRAM, recvfrom(); TCP : SOCK\_STREAM, send(), recv(),listen()