Other notes/definitions:

* packet-switching: hosts break application-layer messages into packets; allows more users to use network
* store and forward: entire packet must arrive at router before it can be transmitted on next link
* routing: determines source-destination route taken by packets
* forwarding: move packets from router’s input to appropriate router output

Exam #1 will cover all of Chapters 1 and 2 in our textbook. The exam is closed book, no notes.  
You should understand:

**1) The four sources of packet delay and the key factor(s) that affect each.**(Section 1.4.1) Slides 1-42 to 51

**dnodal = dproc + dqueue + dtrans + dprop**

Nodal Processing The time taken to process headers, check for bit errors, and determine destination

**dproc: nodal processing**

+ check bit errors

+ determine output link

+ typically < msec

Queueing Delay Packet waits at link for transmission Affected by level of congestion

**dqueue: queueing delay (aka buffer)**

+ time waiting at output linkfor transmission

+ depends on congestionlevel of router

+ Units:

+ R: link bandwidth (bps)

+ L: packet length (bits)

+ a: average packet arrivalrate

+ traffic intensity = La/R

+ La/R ~ 0: avg. queueing delay small

+ La/R -> 1: avg. queueing delay large

+ La/R > 1: more “work” arriving than can be serviced, average delay infinite!

Transmission Delay The time it takes to physically transmit the bits to the link Affected by bandwidth

**dtrans: transmission delay:**

+ L: packet length (bits)

+ R: link bandwidth (bps)

+ dtrans = L/R

Propagation Delay The time it takes for the packet to travel from sender to receiver Affected by length of link}

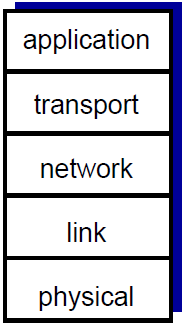
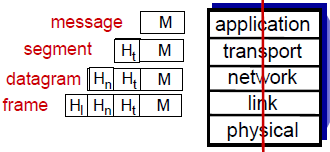
**dprop: propagation delay:**

+ d: length of physical link

+ s: propagation speed (~2x108 m/sec)

+ dprop = d/s

**2) The name and function of each layer in the IP stack.   
(Section 1.5.1) Slides 1-60 to 62**



+ application:

* supporting network applications
* e.g. FTP, SMTP, HTTP
* messages
* packet-switching: hosts break application-layer messages into packets
  + forward packets from one router to the next, across links on path from source to destination
  + each packet transmitted at full link capacity

+ transport:

* process-process data transfer; Performs host-to-host communications
* e.g. TCP, UDP
* TCP provides flow-control, connection establishment, and reliable transmission of data
* Segments

+ network:

* routing of datagrams from source to destination
* e.g. IP, routing protocols
* Exchanges datagrams across network boundaries.
* Internet Protocol (IP) routes datagrams to the next IP router that is closer to the destintation
* Datagrams

+ link:

* data transfer between neighboring network elements
* e.g. Ethernet, 802.111 (WiFi), PPP
* Transfers data between neighboring network elements (no need to go through routers)
* Frames

+ physical:

* bits “on the wire”
* The actual electrical bits on a wire
* Bits

**3) Define the following and describe their role in process communications:  
IP address (IPv4 & IPv6), Socket, and Protocol  
(Sections 1.1.3, 1.5.2, 1.7.3, 2.1.2)**

**IP Address**

* v4 is a 32-bit unique identifier (IP address) for a computer using the Internet Protocol for communications
* v6 is 128-bit
* Servers have a permanent IP address; Clients may have dynamic IP addresses

**Socket**

* Connects Application Layer to Transport Layer
* It sends and receives messages for a process
* A socket has an IP address, a port number, and a transport protocal
* door between application process and end-end transport protocol
* Two socket types for two transport services:
  + UDP: unreliable datagram (push but don’t check if client received)
  + TCP: reliable, byte stream-oriented (push and check if client received)

**Protocol**

* control sending, receiving of messages (e.g., TCP, IP, HTTP, Skype, 802.11)
* protocols define format, order of messages sent and received among network entities, and actions taken on message transmission, receipt
* all communication activity in Internet governed by protocols
* layers: each layer implements a service via its own internal-layer actions and relying on services provided by layer below
  + modularization eases maintenance, updating of system
  + explicit structure allows identification, relationship of complex system’s pieces

**4) The three primary protocols used for email. (Section 2.3)**

**SMTP:**

* Simple Mail Transfer Protocol
* used to send email out to email client
* uses TCP to reliably transfer email message from client to server
* direct transfer: sending server to receiving serverl
* delivery/storage to receiver’s server
* three phases of transfer: handshaking (greeting); transfer of messages; closure
* command/response interaction (like HTTP)
* uses persistent connections
* “Push” protocol: multiple objects sent in multipart message

**POP / POP3**

* Post Office Protocol
* authorization, download
* stateless across sessions
* “download and delete” (Deletes message after it has been downloaded) or “download-and-keep” (copies of messages on different clients)

**IMAP**

* Internet Mail Access Protocol
* keeps all messages in one place: at server
* keeps user state across sessions
* Messages aren’t deleted on server
* More features, including manipulation of stored messages on server
* allows user to organize messages in folders

**5) Dedicated and shared access networks as discussed. (Section 1.2.1)**

**Shared access networks:**

* cable network, i.e. broadband: homes share access network to cable headend; every packet sent by a home travels on the upstream channel to the head end;
* requires a distributed multiple access protocol to coordinate transmissions and avoid collisions
* a number of consumers access an internet connection and the cost of that connection is shared between them
* available bandwidth is split between all concurrent users
* all data is transmitted over one network, consuming away more and more pieces of bandwidth as more devices are connected and types of data are transmitted

**Dedicated:**

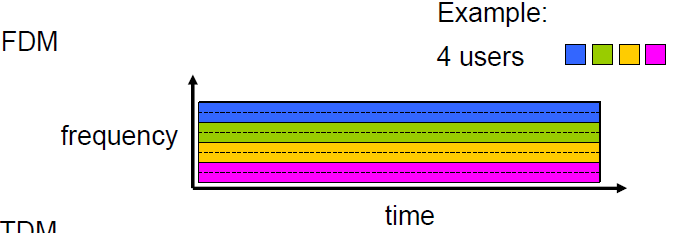
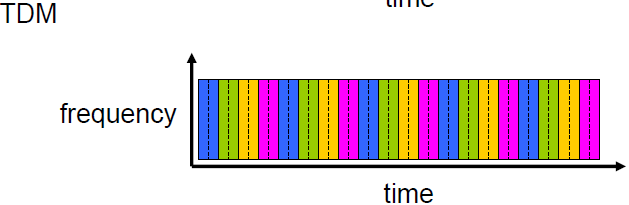
* Internet access and bandwidth dedicated to one user and one user alone
* dedicated Internet refers to bandwidth division among a network’s users - all users have a portion of bandwidth dedicated to them solely
* DSL (digital subscriber line): voice, data transmitted at different frequencies over dedicated line to central office
* establish a private connection between an ISP and the customer’s premises
* This makes dedicated Internet very valuable for large organizations with many users, cloud-based phones or web-enabled devices.

**Wired, wireless communication links:** home network, Ethernet (Enterprise access network), wireless access networks (wireless LANs, wide area wireless access)

**6) FDM and TDM. (Section 1.3.2)**

circuit switching(commonly used in traditional telephone networks):

* circuit-like (guaranteed) performance
* end-end resources allocated to, reserved for “call” between source & dest
* dedicated resources: no sharing; circuit segment idle if not used by call (no sharing)
* network establishes dedicated and end-to-end connection between the two hosts, e.g. PC and workstation

Frequency Division Multiplexing:

* Every user broadcasts at same time using different frequencies
* Spectrum of a link is divided up among the connections established across the link
* Each connection has dedicated frequency band for duration of connection

Time Division Multiplexing:

* Users take turns broadcasting
* Time is divided into frames of fixed duration, and each frame is divided into a fixed number of time slots
* Network dedicates one time slot in every frame of established connection across a link

**Packet-switching:** hosts break application-layer messages into packets; allows more users to use network

Packet switching versus circuit switching

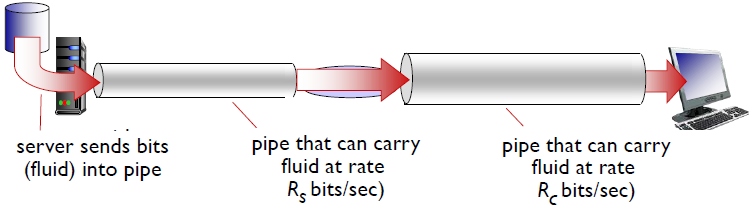
* packet switching allows more users to use network!
  + circuit-switching: 10 users
  + packet switching: with 35 users, probability > 10 active at same time is less than .0004 (Ex. Slide 1-30)
* packet switching
  + great for bursty data
  + excessive congestion possible (packet delay and loss) so protocols are required

**7) Throughput vs. bandwidth. (Section 1.4.4)**

Bandwidth is the *maximum number* of bits that *can travel* over a link. (transmission rate)

Throughput:

* is *actual amount* of data *that travels* through a link.
* rate (bits/time unit) at which bits transferred between sender/receiver
  + instantaneous: rate at given point in time
  + average: rate over longer period of time
* bottleneck link: link on end-end path that constrains end-end throughput; typically client or server (ends) are where the bottleneck is



**8) HTTP (non-persistent and persistent). (Section 2.2.2)**

HTTP: hypertext transfer protocol

Web’s application layer protocol

client/server model

uses TCP: client initiates connection, server accepts connection, HTTP messages are exchanged between broswer and web server, TCP connection is closed

HTTP is “stateless”: server maintains no information about past client requests

RTT (RoundTrip Time) definition: time for a small packet to travel from client to server and back

**non-persistent HTTP:**

brand-new connection must be established and maintained *for each requested object*

at most one object sent over TCP connection (connection then closed)

downloading multiple objects requires multiple connections

non-persistent HTTP response time = 2RTT+ file transmission time

Issues:

* “three-way handshake”: client sends small TCP segment to server, server acknowledges and responds with a small TCP segment, and, finally, the client acknowledges back to server.
* requires 2 RTTs per object: (From handshake: First two parts take one RTT and the last takes one additional RTT)
* OS overhead for each TCP connection; significant server burden
* browsers often open parallel TCP connections to fetch referenced objects

**persistent HTTP:**

+ multiple objects can be sent over single “persistent” TCP connection between client, server

+ Default mode uses persistent connections with pipelining (requests for objects can be made back-to-back without waiting for replies to pending requests)

+ server leaves connection open after sending response; closes when no request received for a while, i.e. timeout

+ subsequent HTTP messages between same client/server sent over open connection

+ client sends requests as soon as it encounters a referenced object

+ as little as one RTT for all the referenced objects

**9) Web caching and its benefits. (Section 2.2.5)**

Also known as: proxy server. (Web browsers can cache too.)

A cached is both server and client at same time because receives and sends from/to browser (server) and sends and receives requests to/from origin web server (client).

Caches an object originally received from web server and sends it to client when requested.

Web caches are meant to satisfy a client request withtout involving origin server

Reduces response time for client and traffic on an access link

Can have old/bad/stale pages because of caching locally versus what is on origin web server.