NEED NANO,MEGA,GIGA,MICRO,ETC

Study guide for exam 2, scheduled for Wednesday November 11, 2009

This exam covers most topics discussed in class from Wednesday October

7 up to and including Wednesday November 4. The format of the test is

the same as all others: definitions, problems, compare/contrast, etc.

No multiple choice or true/false.

Chapter 6:

---------

What is the primary function of the transport layer? (Note: I gave one

definition in my notes; Tanenbaum says something somewhat different

in the book. I consider either to be correct.)

What is QoS? Be able to describe this and/or some of the aspects of

QoS, such as jitter, etc.

What are the similarities/differences between DLL and transport layers?

What is an initial connection protocol, and what is it used for?

- How does using a name server handle this same problem? (Not to be

confused with Domain Name Service [DNS] which is another definition

of "name server").

What is a three-way handshake? Why is a three-way handshake used in

connection management? That is, how are delayed duplicates handled in

connection management, with respect to a three-way handshake?

What is the two-army problem, and what does it relate to, networking-wise?

What is the difference between upward and downward multiplexing? Why might

either be used in a network connection?

What, in general terms, are TCP and UDP?

TCP/UDP:

\* What is a segment? What are the size limitations on payloads?

\* How are fragmentation, sequencing, delay, congestion dealt with?

\* Describe the TCP three-way handshake for connection management

\* Describe the window management policy (in the sliding window protocol)

\* How can low-bandwidth bursty traffic be handled? (Delayed ACKs and Nagle's

algorithm...)

\* What is silly window syndrome?

\* What is the slow start algorithm used for?

\* Timers: in general, what is the solution that TCP uses to deal with

the potential high variance in RTTs?

UDP-Related:

\* Remote Procedure Call - client stub, server stub, marshaling, general

operation thereof

\* Real-Time Transport Protocol (RTP) - general operation, features (or lack

thereof), why exactly it's usually used with UDP; why is it not exactly

application layer and not exactly transport layer?

Performance issues:

\* What can cause performance to degrade?

\* What are the rules of thumb in performance tuning?

\* What is the general approach to use in fast TPDU processing?

Performance issues:

\* Gigabit network protocol problems: what does being 'delay limited'

mean? (As opposed to bandwidth limited)

\* What is the bandwidth delay product?

\* In performance measurement, why should you ...

- make sure the sample size is large enough?

- make sure the samples are representative?

- (continue with the list)

\* What are the general rules for protocol design for high-bandwidth lines?

\* What is header prediction?

Socket programming:

\* Understand the various primitives in terms of what they do.

\* Specific function prototypes will not be asked.

\* Which ones are specific to client (active side) and server (client side)?

Network configuration related to labs:

\* What is the route command used for on unix?

\* DHCP configuration:

+ how to configure a range of addresses for which bootp will be set up

+ what does lease time mean? How is it specified?

+ what kinds of information are configured in the global section?

Host A sends a "WRQ" to host B with source= A's TID, destination= 69.   
Host B sends a "ACK" (with block number= 0) to host A with source= B's TID, destination= A's TID.  
 What is the bandwidth-delay product for a 50-Mbps channel on a geostationary satellite? If the packets are all 1500 bytes (including overhead), how big should the window be in packets? Round-trip delay is ~540 msec. With a 50 Mbps channel, thebandwidth-delay product is 27 Mbits or 3,375,000 bytes. With packets of 1500 bytes, it takes 2250 packets to fill the pipe. Thus, window size should be at least 2250 packets.  
Determine the fastest linespeed possible sending 1500 B payload w/ 120s packet life w/o wrapping sequence numbers around.Account for Ethernet/TCP/IP overhead. 32-bit sequence numbers, so 2^32 / 120 ms = 35791394 Bps (DATA ONLY) / 1500 B/payload = 23860 payloads, payload+ overhead(TCP=20, IP=20,Ethernet=26)=1566 B. 23860/1566 =37364760 Bps = 299Mbps  
A CPU executes instructructions at 1000 MIPS.Data is copied 64 bits at a time, with a cost of 10 instr/word.If an incoming packet has to be copied four times, is a 1 Gbps line possible?To copy 8 bytes, it takes 4 copies \*10 instructions/copy = 40 instructions. Also, 40 instructions takes 40 nsec. Therefore, each byte requires 5 nsec of CPU time for copying. Therefore, the system is capable of copying 1/(5\*10^(-9)) = 200,000,000 bytes per second. This is 1600 Mbps. YES A TCP machine sends 65,535B over 1Gbps channel w/ 10ms one-way delay. Find Max Throughput and line efficiency.1window/20 sec = 50windows/sec. 65535 \* 50 = 3276750Bps \* 8 b/B = 26214000bps (T’put) 26.214/1000 = 2.62% effieciency  
Upward Multiplexing: In the above diagram multiple (N)-layer connections are multiplexed into a single (N-1)-layer connection. Here N represents the Transport Layer and (N-1) the Network Layer. Downward Multiplexing: In the above diagram a single (N)-layer connection is split among several (N-1)-layer connections. Again N represents the Transport Layer and (N-1) the Network Layer.**Uses of Upward Multiplexing: u**pward multiplexing may be employed to reduce cost. Many networks base part of their charge on virtual circuit connect time, the reason being that each virtual circuit consumes node buffer resources. If several transport users can be serviced over a single virtual link via upward multiplexing, then this charge will be less. **Uses of Downward Multiplexing:** Downward multiplexing may be employed to improve throughput. Data from a single transport user may be split and sent simultaneously through the network over several virtual circuits. Throughput can only be increased so far. If there is a single link over which all virtual circuits are multiplexed, the throughput of the transport connection cannot exceed the data rate of that link. As can be seen from the above, there is a tradeoff between cost and throughput. Upward multiplexing allows the user to choose a lower cost and the expense of lower throughput. Likewise, downward multiplexing enables greater throughput at a higher cost.   
listen() mark a socket as accepting connections accept - accept a connection on a socket - 2000 byte email sent 200km one sent over 56kbps modem other sent over 1Mbps. Which is better?  
- max packet life is one day. 32bit sequence number. what is fastest line speed?  
- 4 rules of 'design' –CPU speed moar important than network speed. –Reduce packet count to reduce overhead. –Minimize context switches –minimize copying –You can buy moar bandwidth but not lower delay –avoiding congestion is better than recovering from it –avoid timeouts