Gary Chan
Fall 2001

COMP 361 — Computer Communication Networks I

Final Examination

12:30pm – 3:30pm
Saturday, December 15, 2001

OPEN BOOKS / OPEN NOTES
180 minutes

NAME : ______________________________

Student ID : ______________________________

Assigned Grade: 1)____(/40) 2)____(/20) 3)____(/20)
4)____(/40) 5)____(/30) 6)____(/30)
Total: ______ / 180

Problem 1: Short Questions *(40 points)*

(2 points each) Fill in the blanks of the following questions. In the case of multiple choice, circle your answer. There is no penalty for a wrong answer.

1. There are __________ layers in the ISO OSI model.
2. The routing mechanism in a router is at the ______________ layer
3. End-to-end error recovery is done at the ______________ layer
4. IP is ____________ layer protocol, and TCP is ____________ layer protocol.
5. A fully-connected network of N nodes requires ______________ bi-directional links.
6. A packet of size N bits is transmitted over a channel with BER equal to α. The probability that $n \leq N$ bit errors occur in the packet is given by ______________.
7. A transmitted message $T(x) = x^{10} + x^3 + 1$ is received as $T'(x) = x^9 + x^3 + x$. The error polynomial is $E(x) =$ ______________
8. With a sending window size W_S and a k-bit sending sequence field, the sending sequence is ______________
   (a) 0 to $W_S - 1$
   (b) 1 to W_S
   (c) 0 to $2^k - 1$
   (d) 1 to 2^k
   (e) None of the above
9. HDLC is a ______________ data link protocol
   (a) character count
   (b) character-oriented
   (c) bit-oriented
   (d) None of the above
10. The minimum header size excluding the data layer header for a TCP/IP packet is __________ bytes, while for a UDP/IP packet is __________ bytes.

11. In IP multicast, the sender ________ to know the IP addresses of the its recipients, and each of the recipent ________ to know the IP address of the sender.

    (a) needs, needs
    (b) does not need, needs
    (c) needs, does not need
    (d) does not need, does not need

12. Which of the following regarding link state routing is false?

    (a) Flooding is used.
    (b) Based on global topological information.
    (c) Used in RIP.
    (d) Costs to a node's immediate neighbors are exchanged.
    (e) It does not have the counting to infinity problem.

13. The acronym CIDR stands for ________________________.

14. The size of the basic header without any options for IPv4 is ____________ bytes while for IPv6 is ____________ bytes

15. Which of the following is true for IPv4 but not for IPv6?

    (a) Source and destination addresses are in the header.
    (b) There is source routing option.
    (c) There is checksum for the header.
    (d) The header size is of variable length.
    (e) Routers cannot do fragmentation.

16. RARP stands for ____________.

17. Which of the following regarding virtual circuit routing is not true?

    (a) Better QoS provisioning as compared to datagram routing
    (b) Less reassembly needs to be done at the end-host
    (c) Routing is fast once the connection is set up
    (d) There is no need of a unique address for each host in the network
    (e) None of the above

18. In IP multicast, a host indicates to its router whether it wants a multicast address or not using ______________.

    (a) ICMP
    (b) OSPF
    (c) BGP
    (d) IGMP
    (e) RIP

19. Ignoring header overhead, the total time for TCP sequence number to wrap around in a 1 Tb/s link is ______________ seconds.

20. In UDP programming, we get a socket descriptor by calling `socket(`______________, ______________, ______________`)`.

Problem 2: Network Analysis *(20 points)*

A (p, k) Shufflenet is a regular network proposed to interconnect multiple processors in a computer system together to achieve high level of connectivity with low number of links. There are kp^k processors in the network, with each processor having p outgoing links and p incoming links. The processors are arranged in k columns (hence each column has p^k processors), with the last column connected back to the first column to form a circular cylinder.

Figure 1 shows an 8-node (2,2) shufflenet, with the processors labelled from 1 to 8. As mentioned before, there are $k = 2$ columns, and the second column is "wrapped" around to the first one. Messages are forwarded to their destinations with the minimum hops according to the arrow direction shown (each link is hence unidirectional). **In this problem, you only need to consider this 8-node network.**

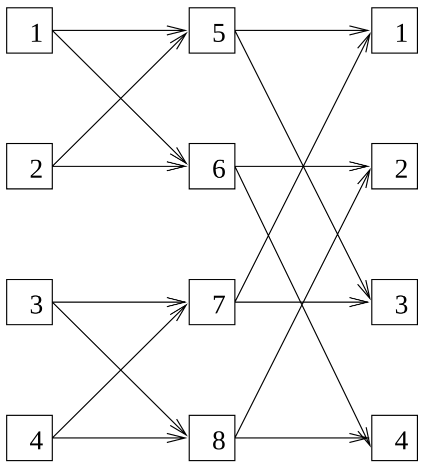


Figure 1: An 8-node (2, 2) shufflenet

a) *(5 points)* What is the diameter of the network?

b) *(5 points)* What is the total number of unidirectional links in the network?

c) *(10 points)* Suppose the destination of the messages originated from a processor is uniformly distributed among the other seven processors. What is the average number of hops of the messages in the network?

Problem 3: Data Link Sliding Window Performance *(20 points)*

In Fig. 2, node A has infinite number of frames to be sent to node C via router B. The distances between A and B, and B and C, are also shown. You are given the following:

- The data rate between A and B is 100 kb/s;
- The signal speed in both communication lines is 10^8 m/s;
- All the lines are full duplex;
- All data frames are 1000 bits long, and transmitted in the store-and-forward manner;
- Acknowledgment is done on the link-by-link (i.e., hop-by-hop) basis, i.e., it is *not* an end-to-end acknowledgement system. All the ACK frames are of negligible size (i.e., "needle" ACK);
- Between A and B, a sliding-window protocol with a sending window size of 2 is used;
- Between B and C, stop-and-wait protocol is used; and
- There are no errors.

Find the minimum transmission rate required between B and C so that the buffers at B will not be flooded. Show your steps clearly.

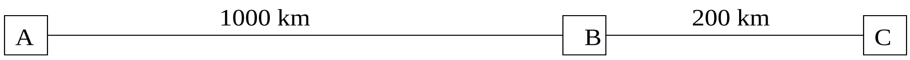


Figure 2: Message transmission from A to C.

Problem 4: Network Flooding *(40 points)*

Consider the network depicted in Fig. 3, where each pair of adjacent nodes are connected by links, some are bidirectional (without arrows) while others are unidirectional (with arrows). The label represents the propagation delay of the link. In this problem, please ignore the transmission and processing delays at a node.

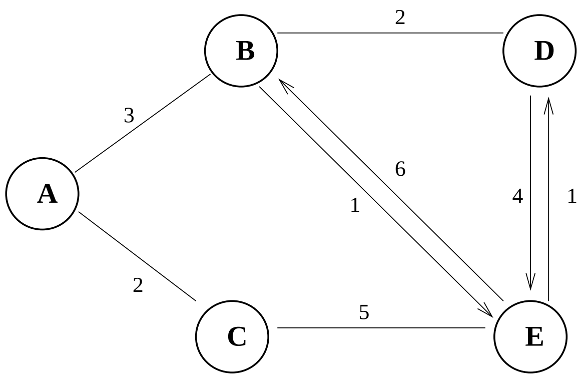


Figure 3: Network topology for Problem 3.

A has a packet to be broadcast to all the other nodes in the network. In the class, you have learnt several ways to do this:

i) Use flooding with TTL field set to the diameter (in hops) of the network;

ii) Use flooding with a unique packet sequence number (identifier);

iii) Address a packet to each individual node in the network and route the packet according to the shortest path algorithm; and

iv) Use the reversed path forwarding.

In this problem, you are to find the number of packets generated and the time the broadcast ends for the aforementioned broadcasting techniques.

a) *(10 points)* Find the number of packets generated and the time at which the broadcast ends for case i).

b) *(10 points)* Find the number of packets generated and the time at which the broadcast ends for case ii).

c) *(10 points)* Find the total number of packets generated and the time at which the broadcast is completed for case iii).

d) *(10 points)*

i) *(6 points)* Draw the sink tree corresponding to node A.

ii) *(4 points)* Find the number of packets generated and the time at which the broadcast is completed according to case iv).

Problem 5: TCP/IP Header Fields *(30 points)*

In Fig. 4 we show a TCP/IP network, where host A is connected to host B via two routers P and Q. Routers P and Q both have two interfaces labelled as 1 and 2 in the figure. Let $IP(x)$ and $MAC(x)$ be the IP and MAC address for interface x, respectively, i.e.,

$IP(A)$	:	IP address for host A;
$MAC(A)$	:	MAC address for host A;
$IP(B)$	:	IP address for host B;
$MAC(B)$	:	MAC address for host B;
$IP(P1)$	:	IP address for interface 1 of router P;
$MAC(P1)$	:	MAC address for interface 1 of router P;
$IP(P2)$	:	IP address for interface 2 of router P;
$MAC(P2)$	:	MAC address for interface 2 of router P;
$IP(Q1)$	:	IP address for interface 1 of router Q;
$MAC(Q1)$	:	MAC address for interface 1 of router Q;
$IP(Q2)$	:	IP address for interface 2 of router Q;
$MAC(Q2)$	:	MAC address for interface 2 of router Q.

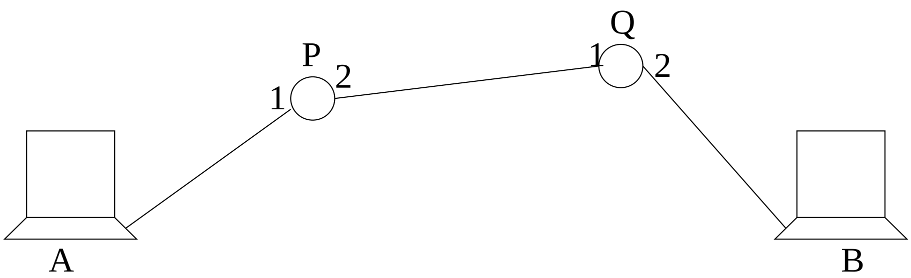


Figure 4: A TCP/IP network.

Base on this network, answer the following questions.

a) *(8 points)* Suppose that host A needs to do an ARP before it sends data to host B. We show in Fig. 5 four header fields in an ARP packet, namely, the source and destination IP addresses, and the source and destination MAC addresses. Let f be the all-1's field.

	Source	Destination
IP		
MAC		

Figure 5: Some header fields in an ARP packet.

i) *(4 points)* If host A does an ARP request, fill in the blank fields.

	Source	Destination
IP		
MAC		

ii) *(4 points)* If router P does an ARP reply, fill in the blank fields..

	Source	Destination
IP		
MAC		

b) *(15 points)* Suppose now host A is to establish a TCP/IP connection with host B. We show in Fig. 6 five fields in a TCP/IP packet. ACK, SYN, and FIN are bit fields.

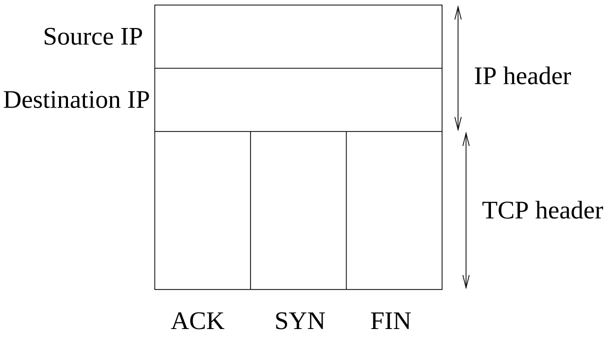
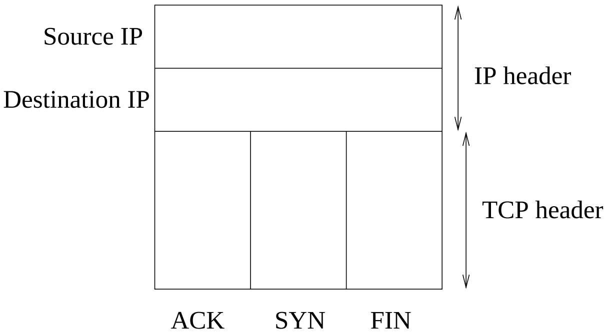
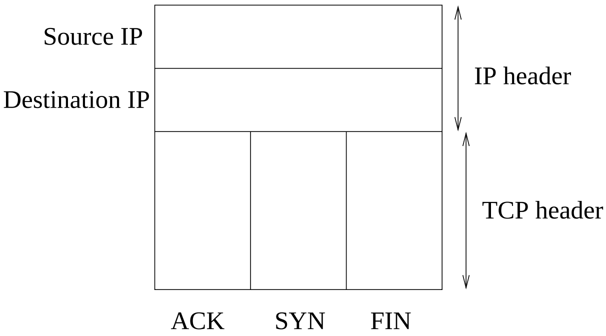


Figure 6: Some TCP/IP header fields.

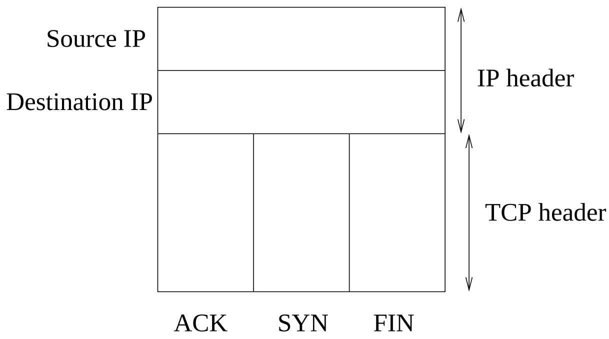
i) *(4 points)* In TCP connection establishment, show the header fields for the first packet sent from host A to host B.



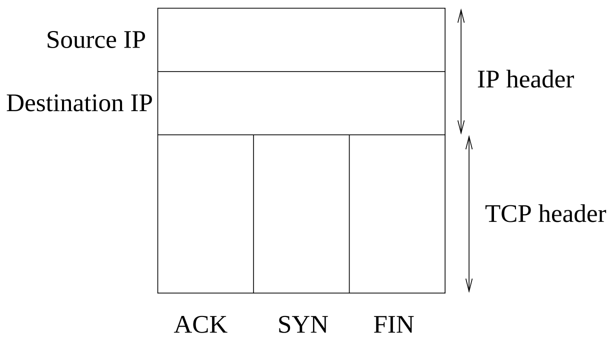
ii) *(3 points)* In TCP connection establishment, show the header fields for the first packet replied from host B to host A.



iii) *(4 points)* During data exchange, show the header fields for the packets from host A to host B in the link between P and Q.



iv) *(4 points)* Suppose A does an active close. In the connection termination, show the header fields for the first packet sent from A to B



c) *(4 points)* Suppose the dot-decimal notation for $IP(A)$ is 111.111.4.16.

i) *(2 points)* What class is the address?

ii) *(2 points)* Write down the IPv6 equivalent of the address in colon hexadecimal notation.

d) *(3 points)* Suppose now host A addresses its data to an IP address 225.240.15.17. Write down its Ethernet destination address in hexadecimal notation.

Problem 6: Applications Layer: Socket Programming

(30 points)

Recall that in TCP/IP programming, the address of a host can be in the following three formats representing different levels of abstraction:

Host name: This is the actual name of a host in which people can easily remember in the form of some alphanumeric characters followed by a dot which in turn followed by some alphanumerics, etc. An example of which is the string "cssu50.cs.ust.hk".

Dot-decimal representation: This is a lower level representation of a host name in which the IP address of the machine is represented by the dot-decimal notation. For example, the dot-decimal notation of the machine name "cssu50.cs.ust.hk" is the string "143.89.40.50".

IP address: This represents the lowest level of address representation. It is IP binary (32 bits) and is carried by the IP packet for routing purpose.

In this problem, you are to write the following three functions in C or C++ to allow you to inter-convert one format to another (Ref. Figure 7):

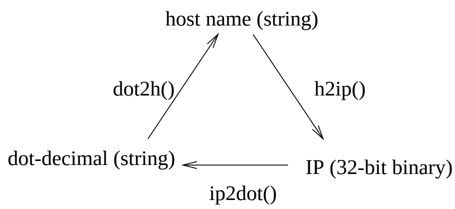


Figure 7: Representations of a host address and the corresponding inter-conversion functions.

1. `unsigned long h2ip(const char * h_name)`: This function takes a string pointer pointing to the host name of the machine and returns its IP address in binary form;
2. `char * ip2dot(const unsigned long ipaddr)`: This function takes the IP address in binary form and returns a pointer of a string with its IP address in dot-decimal notation;
3. `char * dot2h(const char * dotdec)`: This function takes a string pointer of the IP address of a host in dot-decimal representation and returns a pointer of the host name itself.

In writing the functions, you should return `NULL` or `0` whenever an error occurs. You may assume that the string size never exceeds a pre-defined value `MAX_STRG_SIZE`. You are not allowed to call any of the three function primitives within the function itself. Remember to allocate your memory.

a) *(12 points)* `unsigned long h2ip(const char * a_name)`

b) *(6 points)* `char * ip2dot(const unsigned long ipaddr)`

c) *(12 points)* `char * dot2h(const char * dotdec)`

This page is intentionally left blank for rough work or continuation of work.