Data Communication & Networks

Fall 2008 Semester

Roll Number ____________________

Name ____________________

Section ______ Signature: __________

FINAL

Thursday, 4th December 2008

Total Time: 180 Minutes
Total Marks: 100

____
Signature of Invigilator

Course Instructors: Engr. Waleed Ejaz

Q	Introduction	Physical Layer	Data Link Layer	Network Layer	Transport Layer	Application Layer	Total
Marks	20	13	16	19	16	16	100
Obtained Marks							

You are advised to READ these notes:

1. **Attempt the paper on the question paper. NO EXTRA SHEETS will be provided. Use the back of the page if more space is required. However, no extra sheet will be checked.**
2. After asked to commence the exam, please verify that you have **ten (10) different printed pages** including this title page.
3. There are **6 questions**. Attempt all of them. It is advisable to go through the paper once before starting with the first question.
4. All questions don't carry **equal marks**. Marks for subparts are indicated.
5. **Suggested time** for each question is also indicated but this is not hard and fats, its just for your convenience,
6. If part of a problem depends on a previous part that you are unable to solve, explain the method for doing the current part, and, if possible, give the answer in terms of the quantities of the previous part that you are unable to obtain.
7. Exam is closed books, closed notes. Please see that the area in your threshold is clean. You will be charged for any material which can be classified as 'helping in the paper' found near you.
8. Calculator sharing is strictly prohibited.
9. Students who attempt the paper with **lead pencils** loose the right to get them rechecked.
10. The invigilator present is not supposed to answer any questions. No one may come to your room for corrections and you are not supposed to request to call anyone. Make assumptions wherever required and clearly mark them.

Question: Introduction

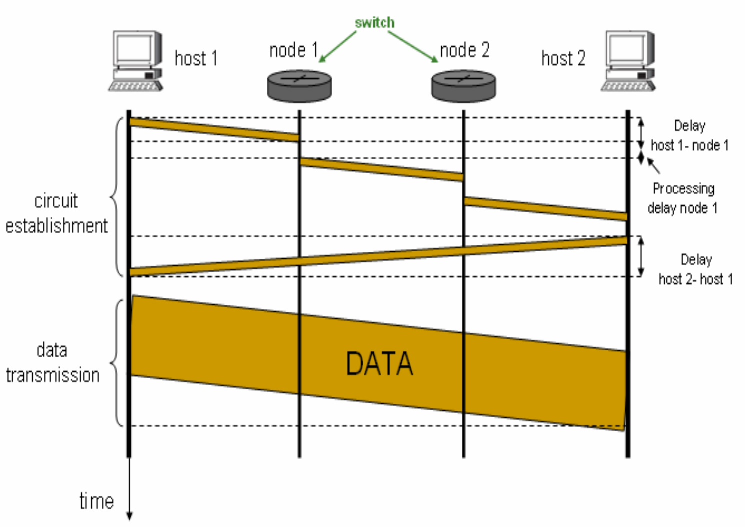
1. Sketch a diagram showing each of the layers in the TCP/IP model. Include the position of each protocol layer in the diagram which we studied in the class. [5]

Solution:

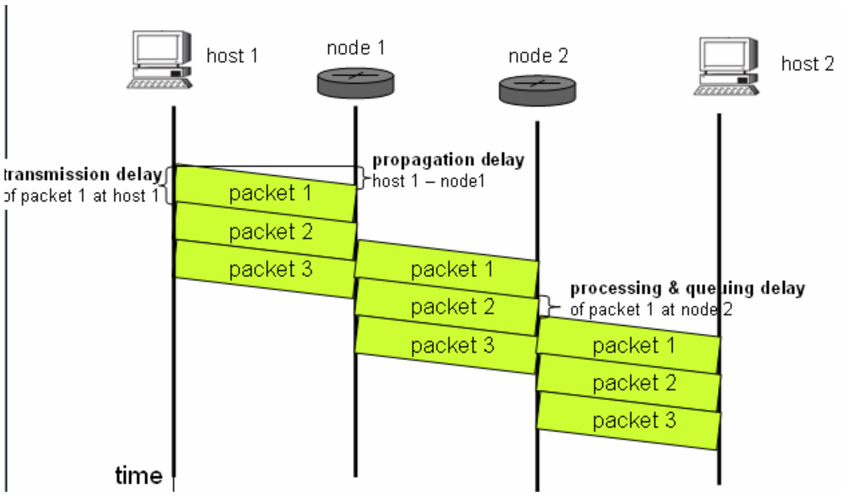
Application Layer	HTTP, DNS, FTP, SMTP, IMAP, POP3
Transport Layer	UDP, TCP
Network Layer	IP, ARP, RARP, ICMP
Data link Layer	CSMA/CD, Stop & Wait ARQ, Go Back N ARQ, Sliding Window ARQ
Physical Layer	

2. Draw timing diagram for Circuit switching and Packet switching for the following network. Assume three packets flow from source to destination and there are two intermediate nodes. [3]

Circuit Switching:



Datagram Network (Connectionless Packet Switching):



3. This elementary problem begins to explore propagation delay and transmission delay, two central concepts in data networking.
   Consider two hosts, A and B, connected by a single link of rate R bps. Suppose that the two hosts are separated by m meters, and suppose the propagation speed along the link is s meters/sec. Host A is to send a packet of size L bits to Host B. [4]

   a. Ignoring processing and queuing delays, obtain an expression for the end-to-end delay.
      Solution:
      $D_{\text{end-to-end}}$= (m/s+L/R) seconds

   b. Suppose Host A begins to transmit the packet at time t=0. At time $t=d_{trans}$, where is the last bit of the packet?
      Solution:
      The bit is just leaving Host A.

   c. Suppose d_{prop} is greater than d_{trans}. At time $t=d_{trans}$, where is the first bit of the packet?
      Solution:
      The first bit is in the link and has not reached Host B.

   d. Suppose s=2.5×108, L=100bits,and R=28kbps. Find the distance m so that d_{trans}. Equals $d_{prop.}$
      Solution:

      Want m=L/R*S= 893 km

4. Consider a queue at a router with a output link rate (i.e. service rate) of R bits/sec. If N packets arrive at the router every LN/R seconds and each packet is L bits long, find average queuing delay for any arbitrary packet. Give a closed form solution. [3]
   Solution:

   It takes LN/R seconds to transmit the N packets. Thus the buffer is empty when a batch of N packets arrives. The first of the N packets has no queuing delay. The second packet has to wait till the first packet is served, i.e. L/R seconds. The third packet has to wait till the first two packets are served. Thus it waits for 2L/R seconds and so on. The nth packet has to wait for the first N-1 packets to be served, i.e. (N-1) L/R seconds. Thus

$$\begin{aligned} \text{Average Delay} &= \frac{1}{N}\sum_{n=1}^{N}(n-1)\frac{L}{R} \\ &= \frac{L}{R}\frac{1}{N}\sum_{n=1}^{N}(n-1) \\ &= \frac{L}{R}\frac{N-1}{2} \end{aligned}$$

5. Consider sending a file of F = M • L bits over a path of Q links. Each link transmits at R bps. The network is lightly loaded so that there are no queuing delays. When a form of packet switching is used, the M • L bits are broken up into M packets, each packet with L bits. Propagation delay is negligible. [2+2+1]

a. Suppose the network is a packet-switched virtual circuit network. Denote the VC set-up time by ts seconds. Suppose the sending layers add a total of h bits of header to each packet. How long does it take to send the file from source to destination?

b. Suppose the network is a packet-switched datagram network and a connectionless service is used. Now suppose each packet has 2h bits of header. How long does it take to send the file?

c. Finally, suppose that the network is a circuit-switched network. Further suppose that the transmission rate of the circuit between source and destination is R bps. Assuming ts set-up time and h bits of header appended to the entire file, how long does it take to send the file?

Solution:

a) The time to transmit one packet onto a link is $(L+h)/R$. The time to deliver the first of the M packets to the destination is $Q(L+h)/R$. Every $(L+h)/R$ seconds a new packet from the $M-1$ remaining packets arrives at the destination. Thus the total latency is

$$t_s + (Q+M-1)(L+h)/R.$$

b) $(Q+M-1)(L+2h)/R$

c) Because there is no store-and-forward delays at the links, the total delay is

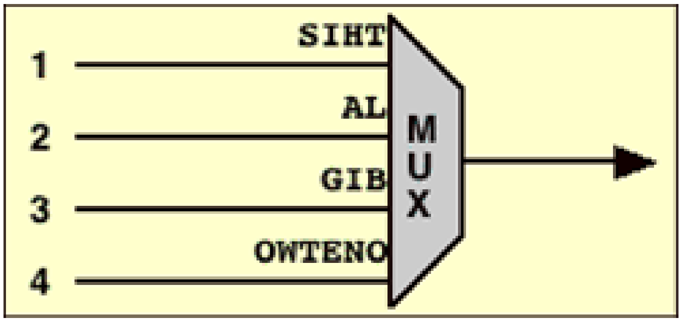
$$t_s + (h+ML)/R.$$

Question: Physical Layer (13)

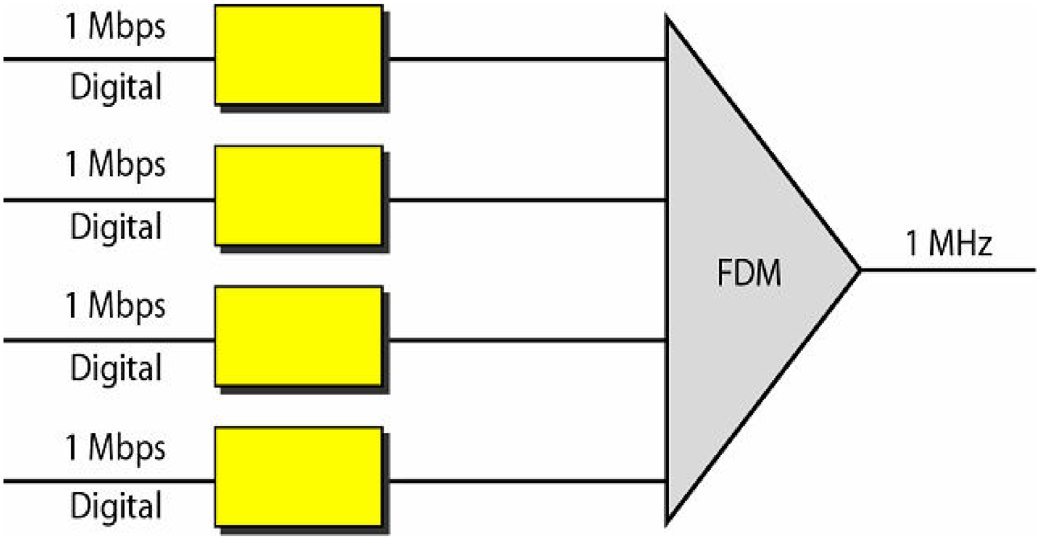
1. Shown below is a multiplexer. Assume the data is transmitted right must character first (i.e., T is sent first in source 1).

a. Draw a diagram showing the character data sent in each of the frames if the MUX is a synchronous TDM multiplexer. [1]

b. Draw a diagram showing the character data sent in each of the frames if the MUX is a statistical TDM multiplexer with a frame size of three characters. [2]



2. Four data channels (digital), each transmitting at 1 Mbps, use a satellite channel of 1 MHz. Design an appropriate configuration, using FDM. [2]
Hint: Fill the four boxes below with appropriate Digital to Analog Conversion Modulation Scheme.

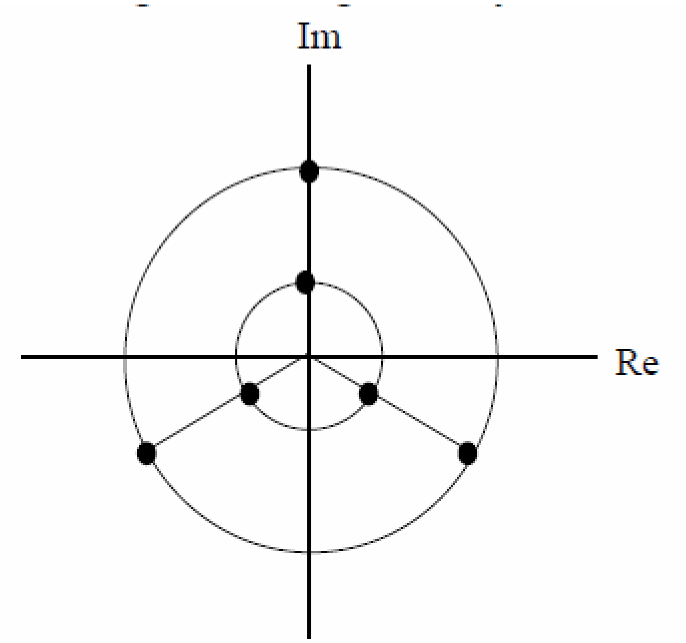


3. Suppose you are digitizing audio signal produced by a microphone. You are using a sampling rate of 20 KHz, and each sample is encoded with 16 bits. No compression is used.
   a. What is the bit rate generated by the digitization process (in Kb/s)?[1]
   b. How long will it take to fill up a disk of 100 MB with this audio stream?[1]

4. Most digital transmission systems are "self-clocking" in that they derive the bit synchronization from the signal itself. To do this, the systems use the transitions between positive and negative voltage levels. These transitions help define the boundaries of the bit intervals. [3]
   a. Plot the NRZ-L signal for the sequence n consecutive 1s followed by n consecutive 0s. Explain why this code has a synchronization problem.
   b. Repeat part (a) for NRZ-I encoding scheme. Does this scheme have a synchronization problem?

c. Repeat part (a) for the Manchester encoding scheme and explain how the synchronization problem has been addressed.

5. An engineer designs the symmetric 6-point QAM constellation given below:



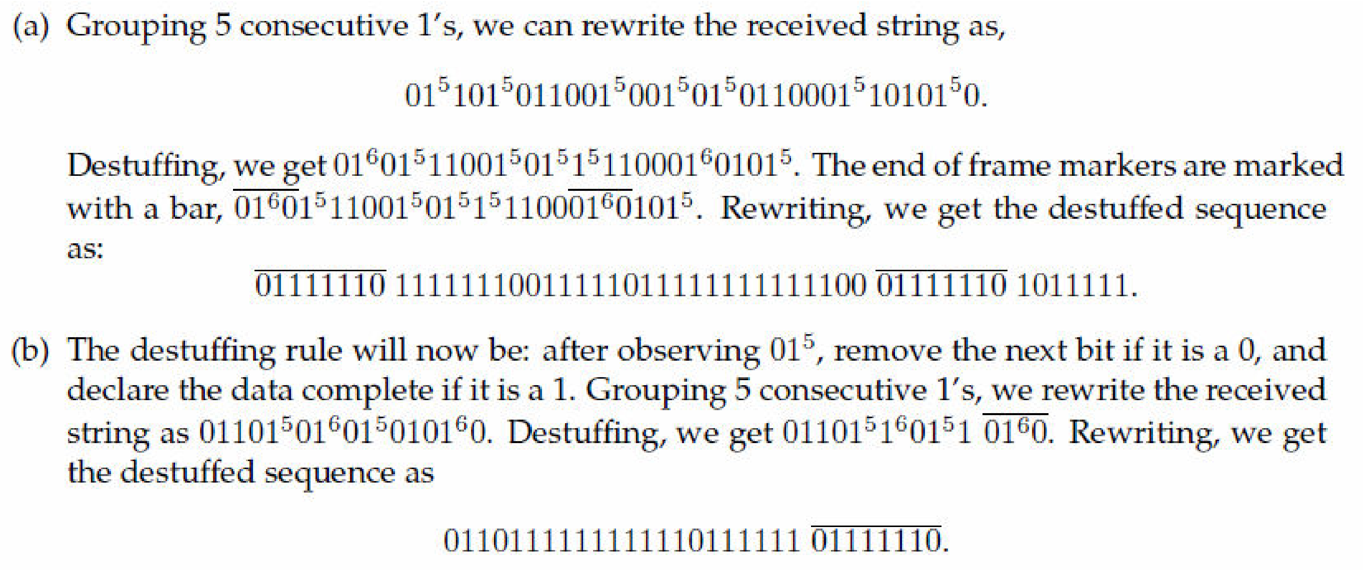
a. How many different phases are produced in this constellation? [1]
b. How many binary digits can be encoded onto each individual symbol or pulse from this constellation? Can you suggest a way to improve this number (bits/pulse) for this constellation? [2]

Question: Data link Layer (16)

1. Consider a link layer protocol in which the sender receives a sequence of bits from the higher layer that it converts into variable length frames, and then transmits to the receiver. In order to indicate the end of a frame, the sender appends the end of data flag, 01111110 (which we will write as 01^60). A problem with this approach is that the *actual data* could contain the flag 01^60. To get around this, suppose that the sender uses a technique called *bit stuffing*: it scans the data from first to last bit and replaces every occurrence of 1^5 with 1^50. The receiver operates as follows: after observing 1^5 it deletes the next bit if it is a 0 and declares the data complete if the next bit is a 1.
Example: If 1111110111111111110111110 is the original data stream then bit stuffing produces:
$11111\bar{0}1011111\bar{0}11111\bar{0}1011111\bar{0}0$
(the zeros with the bars are the stuffed bits) and the end of data sequence 01111110 is appended to the stuffed sequence. For this problem assume that the link is reliable. [4]

a. Suppose the received string is 0111111011111011001111100111110111110110001111110101111110.
Remove the stuffed bits and show where the ends of data flags are.

b. Now change the bit stuffing rule to stuff a 0 only after the appearance of 015 in the original data. Carefully describe how the receiver should destuff in this case.
Also, destuff 0110111110111111011111010111110.



(a) Grouping 5 consecutive 1′s, we can rewrite the received string as,

$$01^5101^5011001^5001^501^50110001^510101^50.$$

Destuffing, we get $01^601^511001^501^51^5110001^60101^5$. The end of frame markers are marked with a bar, $\overline{01^6}01^511001^501^51^51100\overline{01^60}101^5$. Rewriting, we get the destuffed sequence as:

$$\overline{01111110}\ 111111001111011111111111100\ \overline{01111110}\ 1011111.$$

(b) The destuffing rule will now be: after observing 01^5, remove the next bit if it is a 0, and declare the data complete if it is a 1. Grouping 5 consecutive 1′s, we rewrite the received string as $01101^501^601^50101^60$. Destuffing, we get $01101^51^601^51\ \overline{01^60}$. Rewriting, we get the destuffed sequence as

$$0110111111111110111111\ \overline{01111110}.$$

2. Suppose two nodes, A and B, are attached to opposite ends of a 900 m cable, and that they each have one frame of 1000 bits (including all headers and preambles) to send to each other. Both nodes attempt to transmit at time t=0. Suppose there are four repeaters between A and B, each inserting a 20 bit delay. Assume the transmission rate is 10 Mbps, and CSMA/CD with backoff intervals of multiples of 512 bits is used. After the first collision, A draws K=0 and B draws K=1 in the exponential backoff protocol. Ignore the jam signal. [1+5+2]

   a. What is the one-way propagation delay (including repeater delays) between A and B in seconds. Assume that the signal propagation speed is 2 * 108m/sec.
   b. At what time (in seconds) is A's packet completely delivered at B.
   c. Now suppose that only A has a packet to send and that the repeaters are replaced with bridges. Suppose that each bridge has a 20 bit processing delay in addition to a store-and-forward delay. At what time in seconds is A's packet delivered at B?

$$\frac{900m}{2\cdot 10^8 m/\sec}+4\cdot\frac{20bits}{10\times 10^6 bps}$$

$$=(4.5\times 10^{-6}+8\times 10^{-6})\sec$$
$$=12.5\mu\sec$$

b)

- At time $t=0$, both A and B transmit.
- At time $t=12.5\mu\sec$, A detects a collision.
- At time $t=25\mu\sec$ last bit of B's aborted transmission arrives at A.
- At time $t=37.5\mu\sec$ first bit of A's retransmission arrives at B.
- At time $t=37.5\mu\sec+\dfrac{1000bits}{10\times 10^6 bps}=137.5\mu\sec$ A's packet is completely delivered at B.

c) $12.5\mu\sec+5\cdot 100\mu\sec=512.5\mu\sec$

3. How many hamming bits are required when using the hamming code for the message “Help!”? [1]
   6

4. Consider the 2-dimensional parity check code discussed in class. There is m x m data bits. Suppose we have a parity-check bit for each row and a parity-check bit for each column. [3]

   a. What is the data rate in terms of information bits transmitted per coded bit? Is there more or less redundancy in this code compared to the one-dimensional single-bit parity check code with the same number of data bits (Prove it mathematically)?

1. Total number of information bits per codeword $= m\times m=m^2$. Total number of bits per codeword $=m^2+m+m=m^2+2m$. Therefore, data rate $=\dfrac{m^2}{m^2+2m}=\dfrac{1}{1+(2/m)}$. Data rate for the single parity code is $\dfrac{m^2}{m^2+1}=\dfrac{1}{1+(1/m^2)}>\dfrac{1}{1+(2/m)}$. So, the single parity check code has less redundancy compared to the 2-D code.

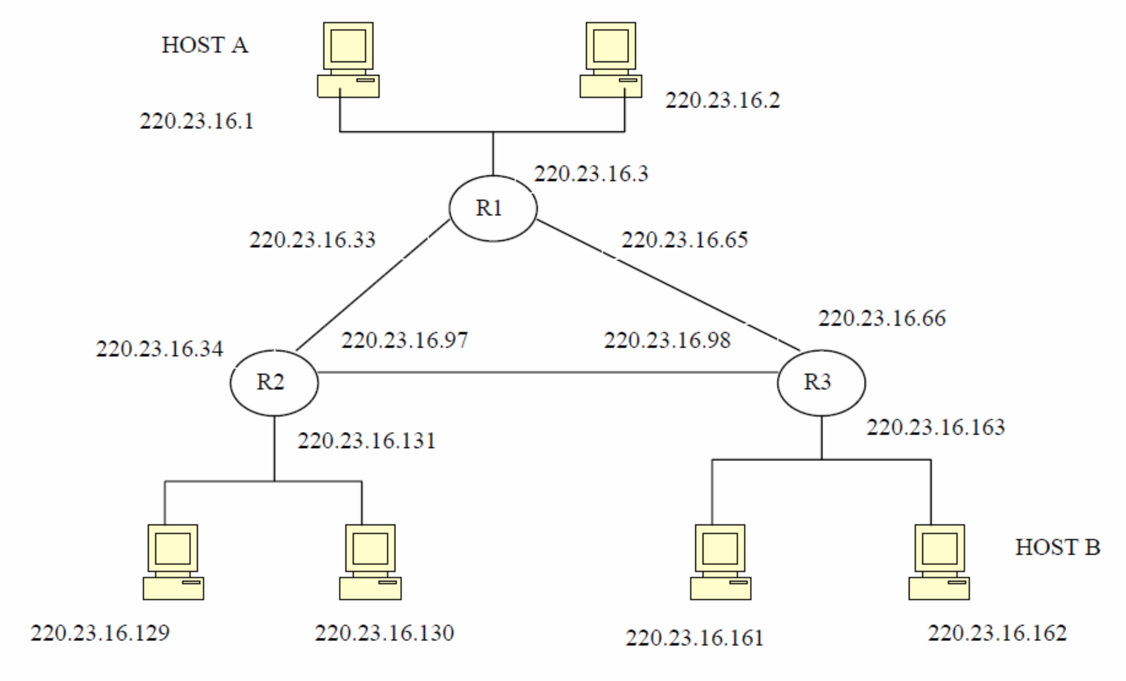
Question: Network Layer (12)

1. Assume that you are the address administrator at an ISP. You have a 128.20.224.0/20 address block. You have two customers with networks of size 1000 nodes each; two customers whose networks have 500 nodes each; and three customers whose networks have 250 nodes each. What are the addresses blocks (first and last address) you will assign to these customers? Use notation similar to

128.20.224.0/20 to denote the address blocks you allocate. Suppose that all your remaining customers have networks of size 50 nodes each. For how many customers can you allocate address blocks with the remaining addresses you have? [10]

1000 nodes need 10 bits => 32 –10 =22 bit prefixes needed
128.20.1110 00 00. 0000 0000/22 = 128.20.224.0/22
128.20.1110 01 00. 0000 0000/22 = 128.20.228.0/22
500 nodes need 9 bits => 32 –9 =23 bit prefixes needed
128.20.1110100 0. 0000 0000/23 = 128.20.232.0/23
128.20.1110101 0. 0000 0000/23 = 128.20.234.0/23
250 nodes need 8 bits => 32 –8 =24 bit prefixes needed
128.20.11101100. 0000 0000/24 = 128.20.236.0/24
128.20.11101101. 0000 0000/24 = 128.20.237.0/24
128.20.11101110. 0000 0000/24 = 128.20.238.0/24
Four more customer networks of size 50 each can be supported (because remaining space = 256 addresses, and minimum granularity = 64 nodes)

2. Suppose that HOST A from the picture below to send an IP datagram to HOST B. Assume that A's ARP Cache is empty. [4]
    a. A starts the process by sending an ARP query. What will be the reply to its ARP query?
    b. What will be the content of the destination-IP-address field in the header of the IP-datagram sent by HOST A?
    c. Will routers R1 and R3 change any of the fields in the IP datagram's header? If yes, which field(s)?
    d. When router R3 receives the datagram, and if its ARP cache is empty, will it have to send an ARP query related to sending this datagram? If yes, what will be the reply to this query?



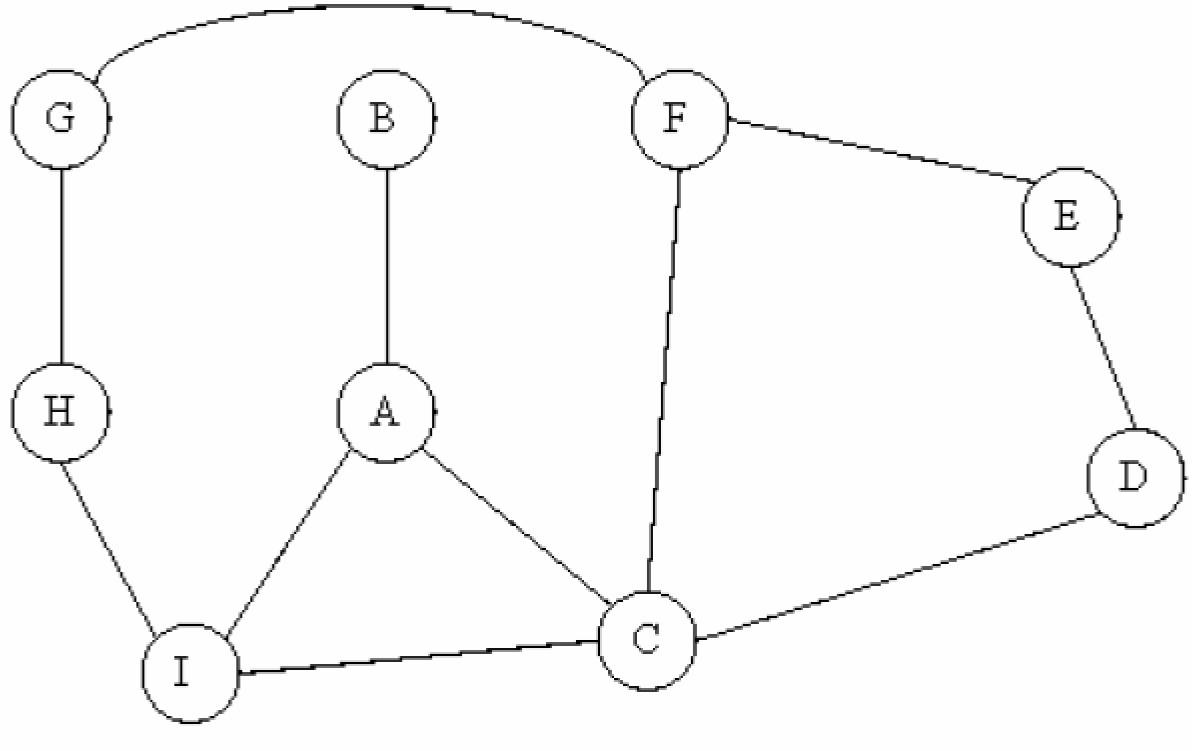
Solution:

a) The data link address of the router's interface connected to network where host A is attached.
b) The IP address of host B.
c) Routers R1 and R3 will change the Time to Live field and header checksum

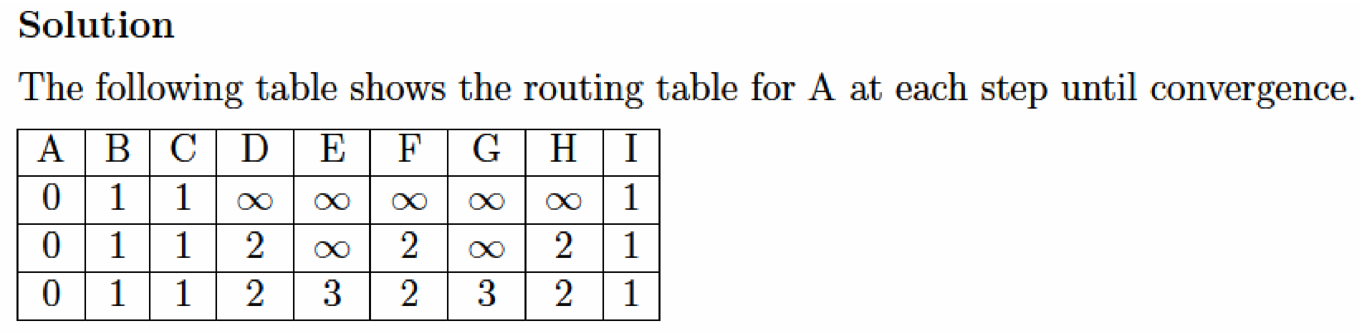
fields.

d) Yes, an ARP query is needed. The answer will be issued by host B and it will be the data link address of host B's network interface.

3. Consider the network configuration shown in Figure. Assume that each link has the same cost.



a. Run Bellman-Ford algorithm on this network to compute the routing table for the node A. Show A's distances to all other nodes at each step. [3]

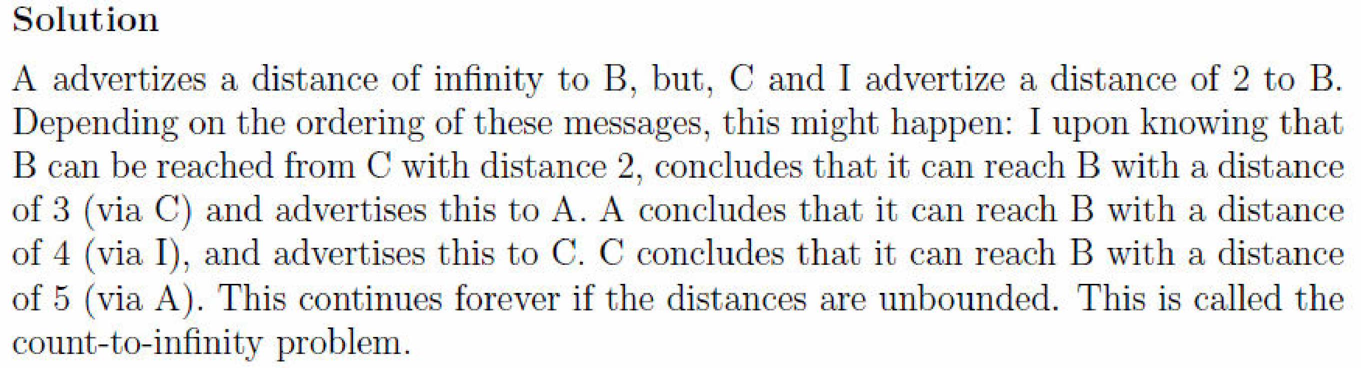


Solution

The following table shows the routing table for A at each step until convergence.

A	B	C	D	E	F	G	H	I
0	1	1	∞	∞	∞	∞	∞	1
0	1	1	2	∞	2	∞	2	1
0	1	1	2	3	2	3	2	1

b. Suppose the link A-B goes down. As a result, A advertises a distance of infinity to B. Describe in detail a scenario where C takes a long time to learn that B is unreachable. [2]



Solution

A advertizes a distance of infinity to B, but, C and I advertize a distance of 2 to B. Depending on the ordering of these messages, this might happen: I upon knowing that B can be reached from C with distance 2, concludes that it can reach B with a distance of 3 (via C) and advertises this to A. A concludes that it can reach B with a distance of 4 (via I), and advertises this to C. C concludes that it can reach B with a distance of 5 (via A). This continues forever if the distances are unbounded. This is called the count-to-infinity problem.

Question: Transport Layer (16)

1. Write down the Receiver side actions for the following events [4]

Event at Receiver	**TCP receiver Action**
Arrival of in-order segment with expected sequence number. All data up to up to expected sequence number already acknowledged. No gaps in the received data.	
Arrival of in-order segment with expected sequence number. One other in-order segment waiting for ACK transmission. No gaps in the received data.	
Arrival of out-of-order segment with higher-than expected sequence number. Gap detected.	
Arrival of segment that partially or completely fills in gap in received data	

2. Consider two TCP flows A and B:

- The RTT for flow A is 100 ms while the RTT for flow B is 200 ms.

- Recall that ssthresh denotes the threshold at which the congestion window size evolution switches over from the Slow Start phase to the Congestion Avoidance phase. Both flows have the ssthresh value of 8.

At time T seconds, both flows have just had a timeout, and so their window size is set to 1. Calculate how many packets each flow is able to send in the next 1 second (including at time T+1 seconds). Assume that each packet can be transmitted in 0 time, and that there are no dropped packets for either flow during the interval [T, T+1] seconds. **[5]**

1 + 2 + 4 + 8 + 9 + 10 + 11 + 12 + 13 + 14 + 15 = 99 for flow A
1 + 2 + 4 + 8 + 9 + 10 = 34 for flow B

3. A TCP header contains number of fields that includes source port number, destination port number, sequence number, receive window, checksum, flags (SYN, FIN, RESET, PUSH, URG, ACK), etc. Identify the fields corresponding to the following TCP protocol mechanism/functions; [3]
   a. Error detection
   b. Application process Identification
   c. Reliable Data Transfer

4. Host A is sending an enormous file to Host B over a TCP connection. Over this connection there is never any packet loss and the timers never expire. Denote the transmission rate of the link connecting Host A to the internet by R bps. Suppose that the process in Host A is capable of sending data into its TCP socket at a rate S bps, where S=10*R. Further suppose that the TCP receive buffer is large enough to hold the entire file, and the send buffer can hold only one percent of file. What would prevent the process in Host A from continuously passing data to its TCP socket at rate S bps? TCP flow control? TCP congestion control? Or something else? Elaborate. [4]

Solution

In this problem, there is no danger in overflowing the receiver since the receiver's receive buffer can hold the entire file. Also, because there is no loss and acknowledgements are returned before timers expire, TCP congestion control does not throttle the sender. However, the process in host A will not continuously pass data to the socket because the send buffer will quickly fill up. Once the send buffer becomes full, the process will pass data at an average rate or R << S.

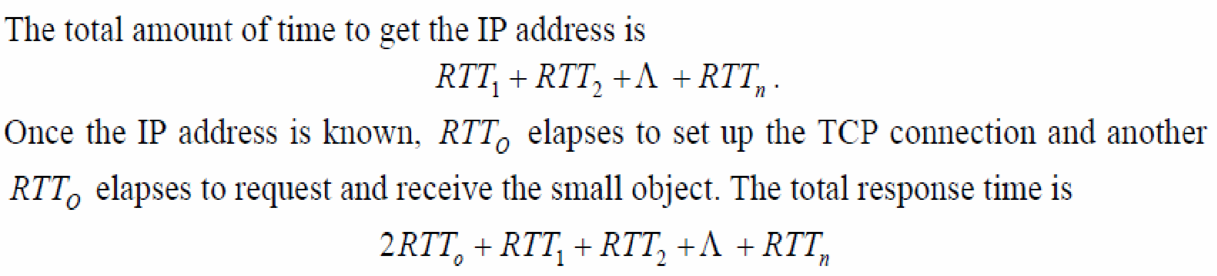
Question: Application Layer (12)

1. What are the key differences between FTP and HTTP? [2]

2. What does GET, HEAD and POST methods in HTTP? [3]

3. Consider an HTTP client that wants to retrieve a Web document at a given URL. The IP address of the HTTP server is initially unknown. The Web document at the URL has one embedded GIF image that resides at the same server as the original document. What transport and application layer protocols besides HTTP are needed in this scenario? [3]
   Solution:
   Application layer protocols: DNS and HTTP
   Transport layer protocols: UDP for DNS and TCP for HTTP

4. Suppose within your Web browser you click on a link to obtain a Web page. The IP address for the associated URL is not cached in your local host, so a DNS look-up is necessary to obtain the IP address. Suppose that n DNS servers are visited before your host receives the IP address from DNS; the successive visits incur an RTT of RTT1, … RTTn. Further suppose that the Web page is an HTML text file and four additional objects. Let RTT0 denote a RTT between the local host and the server containing the object. Assuming zero transmission time of the object, how much time elapses from when the client clicks on the link until the client receives the entire Web page? Assume a non-persistent HTTP protocol with no parallel TCP connections. [4]



The total amount of time to get the IP address is

$$RTT_1 + RTT_2 + \Lambda + RTT_n .$$

Once the IP address is known, RTT_O elapses to set up the TCP connection and another RTT_O elapses to request and receive the small object. The total response time is

$$2RTT_o + RTT_1 + RTT_2 + \Lambda + RTT_n$$

5. Here is a list of four nonproprietary Internet applications. Give application layer protocols and underlaying transport layer protocols that they use. [4]

Application	Application-Layer Protocol	Underlying Transport Protocol
E-Mail	SMTP (RFC 821)	TCP
Remote Terminal Access	Telnet (RFC 854)	TCP
Web	HTTP (RFC 2616)	TCP
File Transfer	FTP (RFC 959)	TCP