Mid-Term Exam - CSE 30264 - Spring 2014

This assignment is individual work. Students should not discuss the problems or answers with classmates, nor should jointly developed solutions be submitted.

NAME: Samantha Rack

1) **(5 pts)** Latency can be defined as (select all that apply):
   a) RTT + (1 / Bandwidth) X transfer_time
   b) Delay x Bandwidth product
   c) **(Distance / Speed_of_light) + (Size_of_message / Bandwidth) + Queue_time**
   d) **Time it takes to send a message from one end of a network to the other and back.**

   Your Answer: c & d

2) **(5 pts)** The Delay x Bandwidth product can be used to measure (select all that apply):
   a) the endian-ness of a network node
   b) the time it takes to send a message from one end of a network to the other and back.
   c) the throughput that a use would achieve on a network link
   d) **the maximum number of bits that could be in transit at any given instant**

   Your Answer: d

3) **(5 pts)** Which of the following is a valid Ethernet Media Access Controller address (select one):
   a) 02:44:3c
   b) **08:00:20:3a:ee:f9**
   c) 239.1.17.36
   d) 08:fe:77:63:87:01:11:00

   Your Answer: b

4) **(5 pts)** Which of the following is a description of shared Ethernet access control?
   a) Carrier Sense Multiple Access with Collision Avoidance
   b) **Carrier Sense Multiple Access with Collision Detection**
   c) Collision Sense Multiple Access with Carrier Detection
   d) Collision Sense Multiple Access with Carrier Avoidance

   Your Answer: b

5) **(5 pts)** You have been instructed to assign the IP address 239.2.11.71 to an Ethernet interface of a host on your network. The address will be used as the unicast address for that host. Is this a reasonable request? Why, or why not?

   This is not a reasonable request. The IP addresses in the range 224.0.0.0 to 239.255.255.255 are class D addresses, which is the multicast class. These addresses are not used for host to host communication and are instead for routers communicating with other routers, so it cannot be assigned to a host.

6) **(30 pts)** Your company is assigned the network address 214.56.78.0/24. The company has five work groups to support: Administration (11 hosts), Sales (32 hosts), Customer Support (41 hosts), Support servers (8 hosts), Web and Database Services (37 hosts). Management would like to provide for 10 percent growth in each group. Management would also like to have a "pool" of unused addresses for future use. The corporate security group wants you to set the network up such that the work groups are on different network segments. Use subnetting to break the /24 network into smaller networks as described above (6 pts). List the networks **(6pts),** the netmasks **(6pts),** the "zero" **(6pts)** and "ones" hosts **(6 pts)** on each subnet in your design.

Group	Network	# Addresses	Netmask	Zero host	Ones Host
Sales	214.56.78.0	64	255.255.255.192 (/26)	214.56.78.0	214.56.78.63
Customer Support	214.56.78.64	64	255.255.255.192 (/26)	214.56.78.64	214.56.78.127
Web & Database	214.56.78.128	64	255.255.255.192 (/26)	214.56.78.128	214.56.78.191
Support Servers	214.56.78.192	16	255.255.255.240 (/28)	214.56.78.192	214.56.78.207
Administrative	214.56.78.208	16	255.255.255.240 (/28)	214.56.78.208	214.56.78.223
Unused	214.56.78.224	32	255.255.255.224 (/27)	214.56.78.224	214.56.78.255

7) **(5 pts)** Why was the file length (size) an important piece of information to send in your file transfer program (Project 1)? Does endian-ness of this value matter? If so, why, if not, why not?

The file length was important in the ftp program because it informed the receiving node of the number of bytes to read until the transfer was complete. Without this information, the receiver would be unable to determine when it had received all of the file data and when it should be interpreting the next data as separate. If the file size was unknown, the receiver would have read the mhash for the file (transmitted by the sender immediately after all bytes of the file were sent) and immediately put it into the file instead of saving it for comparison to its own computed mhash. The endian-ness (byte order) of the file length sent was important because this value was sent as a 32 bit integer. Therefore, depending on the architecture of the nodes communicating, the value could be misinterpreted if it were not converted to network byte order by the sender and back to host byte order by the receiver.

8) **(5 pts)** What is the motivation for ATM having extremely small frames (cells)? What are the tradeoffs that must be considered when using such small cells?

The motivation for ATM using cells is that the cells are a set size and therefore can be forwarded at the hardware level. This allows a switch to forward cells very quickly. The small size of the cell was chosen so that small payloads (such as the 1 byte sent with ssh) does not have to waste bandwidth with a large amount of padding. However, one tradeoff of the

approach of using small cells is that when large amounts of data are sent using cells, there is a high header to data ratio for each of the cells. The data will have to be broken into many cells, and the headers in each of the cells cause overhead that could be avoided with larger frames.

9) **(10 pts)** Compare and contrast the link state approach versus the distance vector approach to building routing tables. **Give an example** of a protocol for each approach.

With the distance vector approach for building routing tables, each node knows the information it has gathered about the direct links it has, and it learns from its neighbors who it can reach through them. The distance vector routing table holds vectors containing the cost to transmit packets to all other nodes. The cost begins at infinity (if the node does not know how to get to the node entry in the table), and is updated as the node collects information given to it by its neighbor. In order to fill in the routing tables, nodes distribute their current routing tables to their direct neighbor, and update their own tables based on the information they receive from their neighbors. If a node receives information about a route to another node, it will update the entry for that node only if the cost is less than the current information. If the network stays consistent, the nodes will all reach convergence in their routing tables and will then only send periodic updates to indicate that they are still there. The distance vector approach is used in the Routing Information Protocol (RIP).

In the link state approach for building routing tables, routing information known by one node is broadcasted to all nodes in the network instead of just to its direct neighbor. When a change occurs in the network or when the routing tables are being initially built, nodes broadcast their information. When no changes occur, nodes exchange "hello" messages with their neighbors to indicate that they are still active and able to receive or forward packets. The routing information from each host is broadcasted using a Link State Packet (LSP) which identifies the source node giving the information, gives the cost of the link, and contains a sequence number and Time To Live to allow for reliable flooding of the information. The last two parts of the packet mentioned in the previous sentence allow nodes to identify when they are receiving outdated information or when they should stop forwarding an LSP. Since each node has global information in the link state approach, a routing table is determined using this information with Dijkstra's shortest path algorithm. The link state approach is used with the Open Shortest Path First (OSPF) Protocol.

10) **(5 pts)** Suppose that you sign up with a local Internet Service Provider for home Internet service. You pay for a plan which touts 16 megabit/second down, and 4 megabit/second up. You notice that most of the time you do not get the advertised throughput. What circumstances might lead to degraded throughput?

A degraded throughput can be a result of many factors. If a switch in your network cannot keep up with forwarding the frames it is receiving on all of its ports, then this will degrade the throughput for the network. Throughput in the network is also reduced when there is interference in the transmission of signals in the network. This causes errors in the data being transmitted, which is detected and requires that packets be resent, using some of the bandwidth. ARP broadcast packets and LSPs will also use some of the bandwidth that is provided by the ISP so routing can be completed effectively. The three way handshake of TCP can slow down data transmission because the nodes have to wait for acknowledgements before continuing the transfer of data.

11) **(5 pts)** Several years back, the Duke campus network experienced issues related to Apple devices overwhelming the wireless network. One of the causes examined was with respect to devices being overly aggressive in their ARP (ARP timeout was too low). The Duke network at the time was similar to the University of Notre Dame network at the time: a single switched network where all wireless APs (access points) were part of a single switched network segment. Given your understanding of how ARP works, was the above cause plausible, i.e. could an overly aggressive ARP timeout cause havoc in a purely switched network?
Note: While we have not covered wireless technology in depth, the concepts of ARP are the same across wired or wireless devices. Think to the root of what ARP does and how it works for your answer.

The cause described above is plausible given the setup of the network (as a switched network) and the short ARP timeout. When an IP address identified in a packet is not contained in the ARP table, a broadcast packet is sent out to determine which MAC address corresponds to the IP address. The host with the broadcasted IP address responds, identifying itself and giving its hardware address. If the ARP timeout is low, then the entries in the tables expire more quickly. So if an IP address is not refreshed before the timeout (as would be the case if a packet from that IP address were not seen), then it would be erased from the table, though it could be used soon after this timeout. Thus, there would be a lot of timeouts and then a necessity for a larger number of broadcasted ARP packets to identify the erased IP information. Because switches all forward broadcast packets, the network would be congested with ARP packets.

12) **(5 pts)** Would a **routed** network between the wireless APs rather than a **switched** network between the wireless APs experience the same problems as exhibited in question 11?

A routed network between the wireless APs instead of a switched network between the wireless APs would not have experienced the same issues because of the difference in the ways that routers and switches handle broadcast packets. Switches forward all broadcast packets, which, as is mentioned in the question above, caused the network to be congested with ARP broadcast packets and responses to these. Conversely, routers do not forward broadcast packets, so the congestion would not be as great as with the switches. Though there would still be a large number of ARP packets sent out with a short timeout, the routers would restrict the impact of these packets on the bandwidth since they would not be flooded to the entire campus network.

13) **(5 pts)** Your parents recently had their laptop into a big box store because the network connection was no longer working. The technician tells them that their dynamic address has expired and that it will cost $60 to reset the dynamic address lease while flushing the ARP cache. Is this a plausible service? Why or why not?

This service is not plausible. It is true that a host's dynamic address can expire if it is disconnected from the network.However, if a host's dynamic address does expire, it will use Dynamic Host Configuration Protocol (DHCP) to get a new address. The host will broadcast its hardware address (that from Ethernet) on the network, asking for an IP address. The DHCP server responds with an unallocated IP address for the host.
When a host's dynamic address expires from an ARP table (which occurs if a timeout elapses where the host was not refreshed for the given table), the address lease does not have to be refreshed through "flushing the ARP cache". If a host does not have the dynamic address in its ARP table (ie. if it expired as the technician says), then it broadcasts to determine the

hardware (Ethernet) address of the node who sent it. My parents' laptop would respond to this broadcast, since it will recognize its dynamic address in the broadcast, giving its hardware address so there can be packets exchanged with it.

14) **(10 pts)** Compare / relate the terms in the context of your coding: *socket, accept, connect, recvfrom, listen*.

The *socket* call is made to get an active socket from the operating system. The socket call requires a family (PF_INET), a type (stream–TCP or datagram–UDP), and a protocol. This is the first call made when setting up a client or server. The call returns a socket descriptor, which is much like a file descriptor, and is used throughout the program to interact via the socket.

The *accept* call is made by a TCP server who has made a *socket* call, has *bind*–ed the socket to a port, is *listen*–ing, and is waiting for a client to connect. This call blocks until a client connects to the server who is waiting to *accept*. The call to *accept* returns a new socket descriptor specific to that connection, and it is how the server will interact with the client who has just connected until the connection is closed.

The *connect* call is made by a TCP client who has made a *socket* call and is trying to connect to a specific server for communication. It is possible to use *connect* with a UDP client, but it is most common with TCP. After *connect*–ing with a server (which has made a call to *accept* and is waiting for this *connect*), the client is able to read or write to the server via the socket descriptor.

The *recvfrom* call can be made by a UDP client or server (TCP can also use it, but with this protocol, read or recv and write or send are typically used). This call is made when the node is waiting to receive data. The *recvfrom* call blocks until data is read, or there is an error.

The *listen* call is made by a TCP server to complete the three–way handshake required by the protocol. This allows the socket to begin *accept*–ing connections from clients.

15) **(5 pts)** Why are long streams of all zero or all one bits a problem on a communication link? What techniques can be used to transmit long strings of zeros or ones without encountering these problems?

Long streams of all zeros or all one bits are a problem on a communication link because if the zeros and ones are directly mapped to high or low signals, respectively, for transmission, the physical signaling on the network suffers from baseline wandering and the problem of clock recovery. In baseline wandering, distinguishing between zeros and ones becomes difficult because the receiver has seen a lot of zeros or ones and does not have a recent reference of what the other bit looks like as a signal. Devices receive or transmit bits at each clock cycle, and it is difficult to determine if the device's clock is still synchronized with the sender's clock if it is continually receiving one kind of signal (high or low). Thus, bits can be decoded incorrectly when the clocks are not in sync.

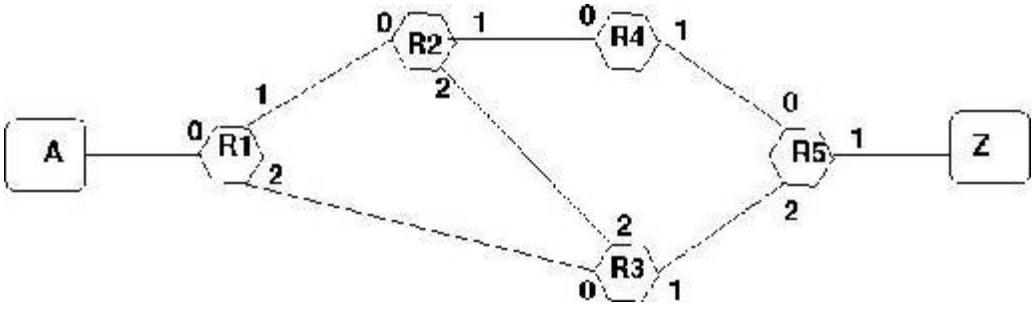
In order to avoid these problems (or at least reduce them), techniques for encoding bits that are not direct mappings of 0 and 1 bits have been developed. Some of these techniques include non–return to zero inverted (NRZI), Manchester encoding, Differential Manchester, and 4B/5B. These encodings use techniques such as ANDing with the clock, inserting bits to interrupt long strings of 0 or 1, and changing the signal on a 1 bit and staying constant on a 0 bit.

16) **(5 pts)** Name three framing techniques used on communication networks, and an example protocol for each technique.

The three framing techniques used on communication networks are byte-oriented protocols, bit-oriented protocols, and clock-based protocols. Byte-oriented protocols look at frames as collections of bytes, which can be implemented either with sentinel values indicating the beginning and end of frames or with byte counts passed in the frame header indicating how long the frame is. An example byte-oriented protocol using the sentinel approach is the Binary Synchronous Communication (BISYNC) protocol. BISYNC frames use sentinel bytes to indicate the beginning and end of the payload of the frames. Since the end of data byte could be found in the actual data, an "escape" bytes is inserted indicating that the next bit is not the end of data byte. Bit-oriented protocols look at frames as collections of bits. An example of this approach is the High-Level Data Link Control (HDLC) protocol, which uses a bit sequence to indicate the beginning and end of the frame. Because this sequence can also be found in the data of a bit-oriented frame, bit stuffing is implemented to indicate that the end of frame has not been reached. The final approach, implemented in with the Synchronous Optical Network (SONET) is clock-based framing, which defines a layout for a frame. SONET uses frames that are 2 dimensional, with 90 bytes in each of the 9 rows. The receiver looks for the starting bit pattern every 810 bytes, so bit stuffing is not necessary or used.

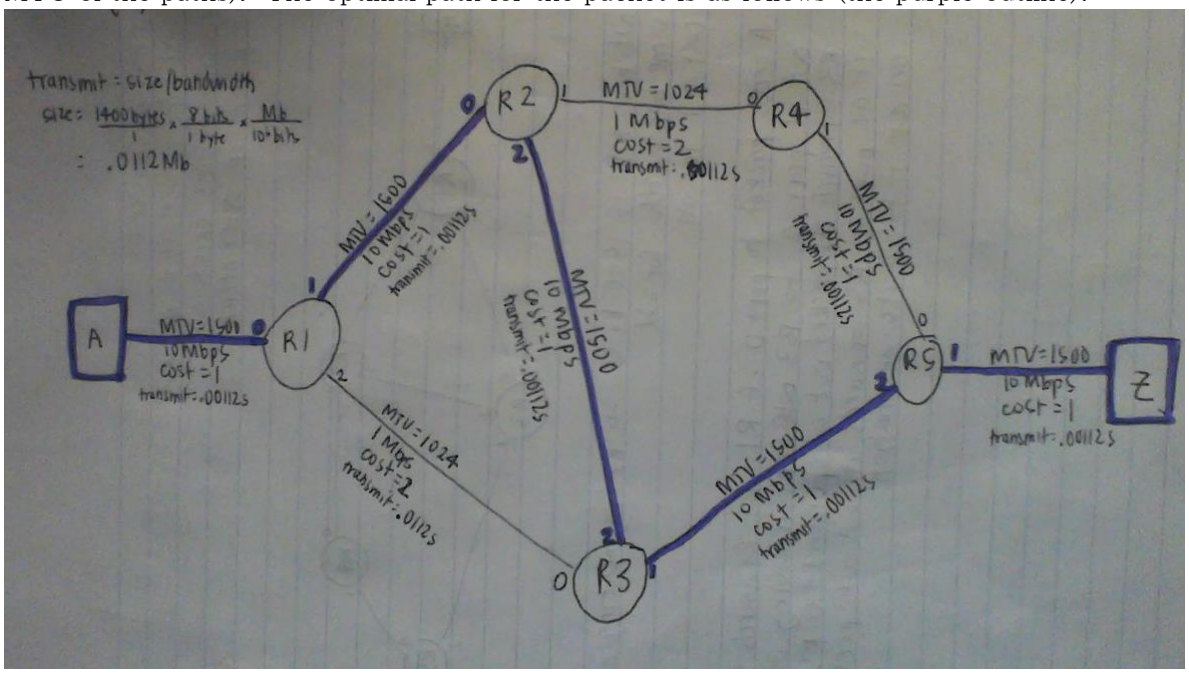
17) **(20 pts)** Using the figure as a reference, give a **hop by hop description detailing** how a 1400 byte packet would travel from Host A to Host Z given the following conditions. Explain why you chose the route that you describe. You do not need to build routing tables for the routers…assume that the tables exist, and support the route that you decide to use.

a) All links have an MTU of 1500 bytes, all links provide the same bandwidth, and the cost of every link is "1". You want the "lowest cost" route.
b) The links from R1 to R3 and R2 to R4 have an MTU of 1024 and operate at 1 megabit/second. All other links have an MTU of 1500 and operate at 10 megabit/second. The cost of the one megabit links is 2, while the cost of the 10 megabit links is 1. You want the route with the "best throughput/least delay".
c) The links from R1 to R3, R2 to R4, have an MTU of 1024 bytes. The link from R3 to R5 has an MTU of 512 bytes. All other links have an MTU of 1500 bytes. The links with an MTU of 512 provide 512 kilobit/second with a cost of 2. Other links are 1 megabit/second with a cost of 1. You want to select the route with the "lowest total overhead".



a) Since all of the links have an MTU of 1500 bytes and the packet is 1400 bytes, there will be no fragmenting along the path. Finding the "lowest cost" route consists of finding the path with the least number of hops, since each of the links has a cost of 1. So the path traveled by the packet is as follows: Host A sends the packet to port 0 of R1. R1 forwards the packet out of port 2 to R3, which receives it at port 0. R3 then forwards the packet out of port 1 to R5, which receives it at port 2. Finally, R5 forwards the packet out of port 1 for delivery to Host Z, the packet's destination.

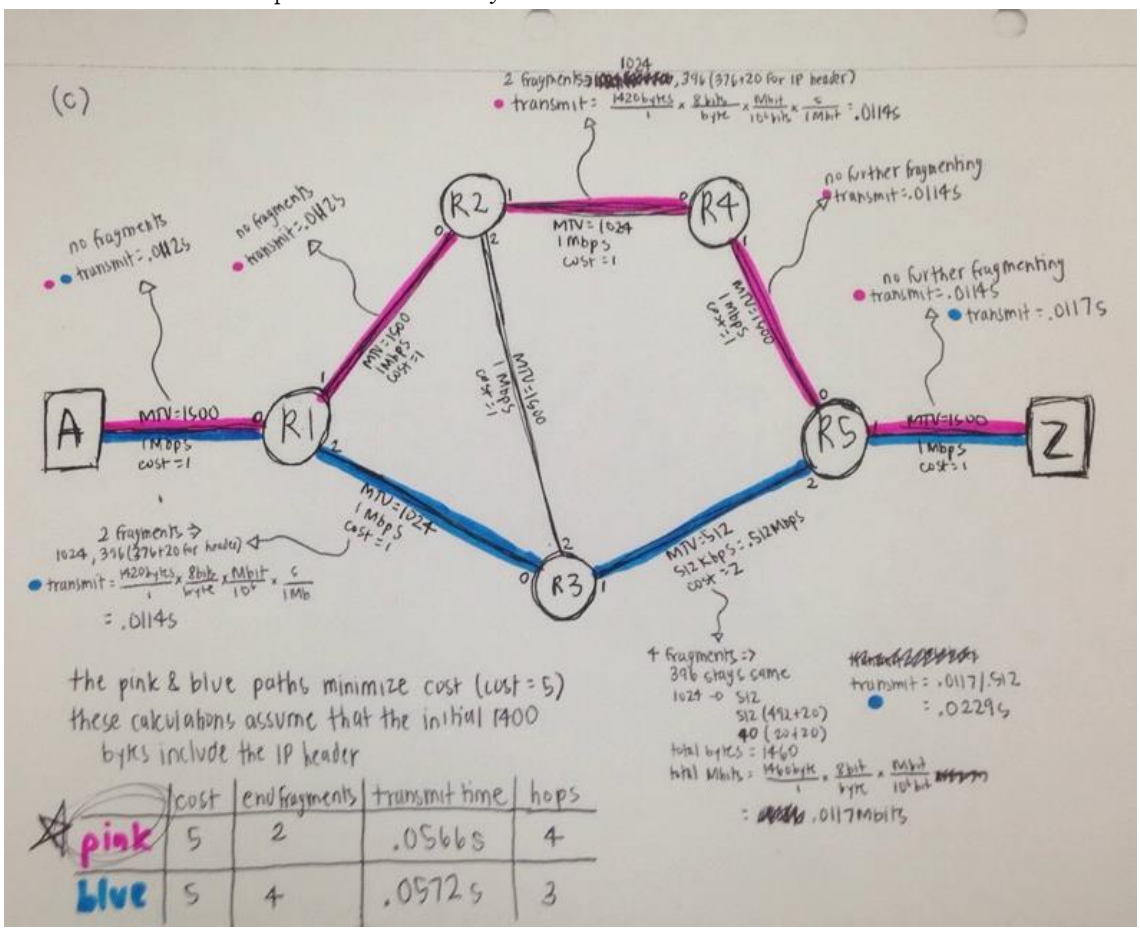
b) Since the route desired is the "best throughput/least delay", the path optimization will be based on the transmit time (assuming propagation and queue delay are constant across links), which is directly related to the bandwidth of the links (not taking into account the costs or MTU of the paths). The optimal path for the packet is as follows (the purple outline):



Host A sends the packet to port 0 of R1. The transmit time of this traveling of the packet is 0.00112s (transmit = size/bandwidth, with unit conversions: transmit = .0112Mb/10Mbps = .00112s), and no fragmenting occurs (MTU = 1500 > 1400). R1 forwards the packet out of port 1 to R2, which receives it at port 0. The time of this link is also 0.00112s, and there is no fragmenting. R2 forwards the packet out of port 2 to R3, which receives it at port 2. The transmit time is 0.00112s with no fragmenting occurring. R3 then forwards the packet out of port 1 to R5, which receives it at port 2. The transmit time is again 0.00112s with no fragmenting of the packet. Finally, R5 forwards the packet out of port 1 for delivery to Host Z. The total transmit time is 5*0.00112 = 0.0056, which is the minimum latency for the transmission of a 1400 byte packet from Host A to Host Z with these network specifications.

c) (I am assuming for this problem that "lowest total overhead" finds the route with the lowest cost, smallest latency, and least number of fragments required.) The picture below shows most of the work for this problem. First, to determine candidates for the optimal path with the "lowest total overhead", the two paths with the lowest cost were determined. These two paths are traced in pink and blue, and each of them has a cost of 5 to deliver the packet from Host A to Host Z. After determining these two paths, the latency for communication across each of the links and the fragmentation needed was determined for the paths. A summary of the data calculated is shown in the table at the bottom of the diagram. The pink path (A -> R1 -> R2 -> R4 -> R5 -> Z) required fragmenting once into two fragments, so Host Z would have to put together two fragments. The pink path's total transmit time (assuming that the original 1400 bytes included the 20 bytes for the IP header and that the MTU must include the IP header as part of the packet) is 0.0566s, and the number of routers through which it hops is 4. The blue path (A ->R1 -> R3 -> R5 -> Z) required fragmenting twice, ending up with four fragments to be assembled by Z for the entire packet. The blue path's total transmit time (with the same

assumption as made for the pink path) is 0.0572s, and the number of routers through which is hops is 3. In comparing these two paths, the pink path has the least total overhead because it requires less fragmenting at the routers and less fragment assembly at Host Z, it has a smaller latency, and it has the minimal cost for the possible paths. Therefore the optimal path is as follows: Host A sends the packet to port 0 of R1, R1 forwards it out of port 1 to port 0 of R2, R2 forwards it out of port 1 to port 0 of R4, R4 forwards it out of port 1 to port 0 of R5, and R5 forwards it out of port 1 for delivery to Host Z.



	cost	end fragments	transmit time	hops
pink	5	2	.0566s	4
blue	5	4	.0572s	3

18) **(15 pts)** Suppose we have the following struct defined in our code.

```
struct ClassInfo
{
 char szName[41];
 char bTTH;
 uint16_t   nCredits;
 uint16_t   nStartTime;
};
```

Sketch the C code to send the instance of that struct (listed below) to port 9265 at address 129.74.20.40 using TCP. You may assume all appropriate header files have been included and that endian-ness is not a problem.

```
struct ClassInfo  myClass;
strncpy(myClass.szName, "Computer Networks");
myClass.bTTH = TRUE;
```

```
myClass.nCredits = 3;
myClass.nStartTime = 1230;

//////////////////////////////////////////

int sockfd = socket(PF_INET, SOCK_STREAM, 0);
if (sockfd < 0) return -1; //error checking

//set up server's structure
struct sockaddr_in serv;
bzero((char *)&serv, sizeof(serv);    //zero the structure
serv.sin_family = AF_INET;
serv.sin_port = htons(9256);
inet_pton(AF_INET, "129.74.20.40", &(serv.sin_addr));

if (connect(sockfd, &serv, sizeof(serv)) < 0) return -1;
//now we are successfully connected
int eCode;

//send the size of the string before the string so we don't have to send
//   junk data at the end of it, and the server knows how many characters
//   to receive for this string
short szName_len = htons(strlen(myClass.szName));
eCode = write(sockfd, &szName_len, sizeof(szName_len));
if (eCode < 0) return -1;
eCode = write(sockfd, myClass.szName, strlen(myClass.szName));
if (eCode < 0) return -1;

//send next component of the struct, a character
eCode = write(sockfd, &(myClass.bTTH), sizeof(char));
if (eCode < 0) return -1;

//send the nCredits short in the struct after converting to network byte
//    order
short nCredits_nbo = htons(myClass.nCredits);
eCode = write(sockfd, &(nCredits_nbo), sizeof(nbo));
if (eCode < 0) return -1;

//send the nStartTime short in a similar way
short nStartTime_nbo = htons(myClass.nStartTime);
eCode = write(sockfd, &(nStartTime_nbo), sizeof(nStartTime_nbo));

//done sending, so close connection to clean up
if (close(sockfd)) return -1;
```

Provide your answers in a file *midterm.pdf* and place this file in your dropbox. Improper submissions will not be graded. Submissions made after the deadline (10/16/14 at 12:30pm) will not be graded.

Submitting this assignment signifies that this submission is individual work, and that the student has not discussed/completed the assignment with assistance from others.

Do not forget to put your name on your submission!