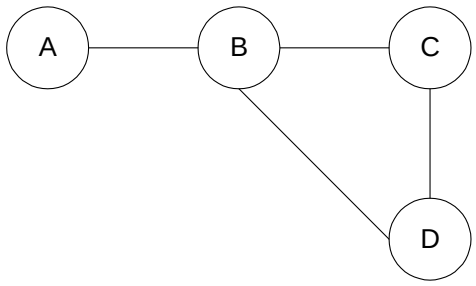
MITP 432
Communication Networks I

Homework #4

1) Consider the following small network:



   a) [15] Assuming a metric of 1 between all hosts, perform the distance-vector routing algorithm from class on this network. Show the contents of each node's routing table at each step.

   b) [15] Given the results from the previous question, show step-by-step what happens when link BC goes down. Include all steps until the distance-vector algorithm converges.

   c) [15] Assuming that the complete routing tables are known (and that all links are up), show a step-by-step computation of Dijkstra's algorithm on host A's table.

2) [10] Consider TCP congestion control. Suppose that a server is not in the fast recovery state and is sending a large file to a client using TCP. The server's internal TCP parameters are as follows:

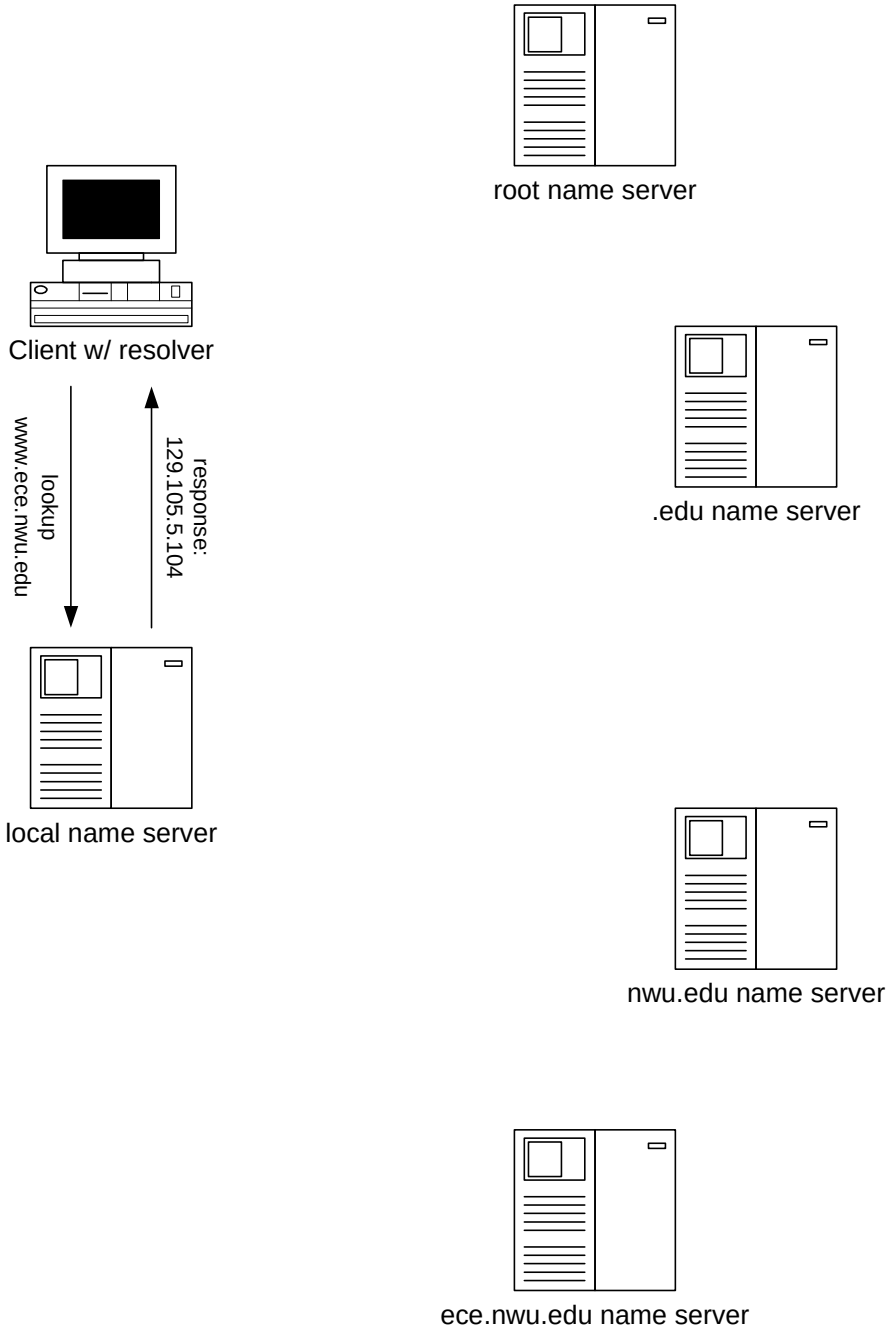
   - *MSS = 1460*
   - *cwnd = 8760*
   - *ssthresh = 5840*
   - *awnd = 14600*
   - *fs (flight size) = 5840*

   a) Suppose that the server receives a 3rd duplicate ACK. What are *cwnd* and *ssthresh* changed to? What TCP congestion state will the server be in after this happens?

   b) Suppose that the server detects a timeout. What are *cwnd* and *ssthresh* changed to? What TCP congestion state will the server be in after this happens?

.

3) [10] Below is a diagram from the notes on DNS. Show what would happen, by drawing lines representing requests and responses, if *all* of the name servers in the diagram used recursion, except for the .nwu.edu name server. The initial request and the final response are already given for you.



4) [10] For this problem, refer to RFCs 3513 and 4007.

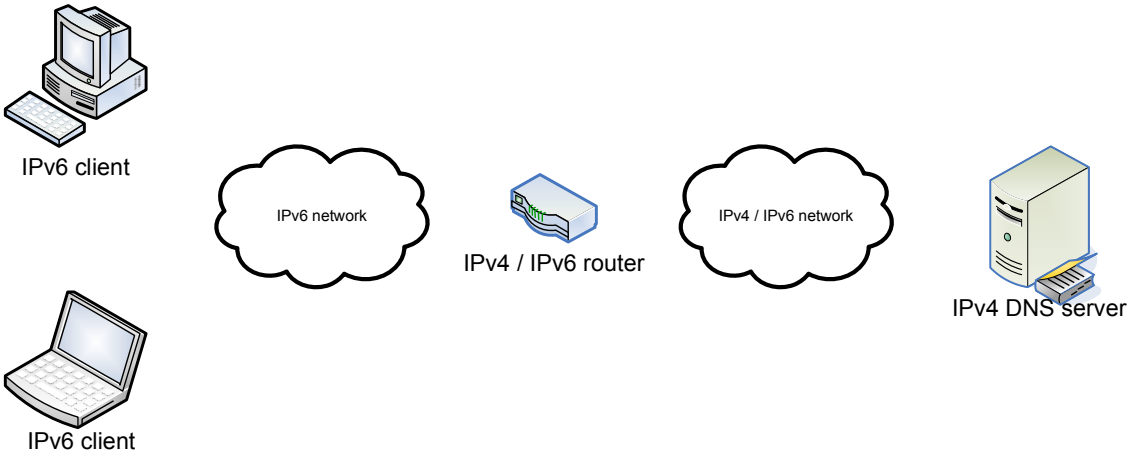
   a) What are the long and shortened notations for the IPv6 loopback address?

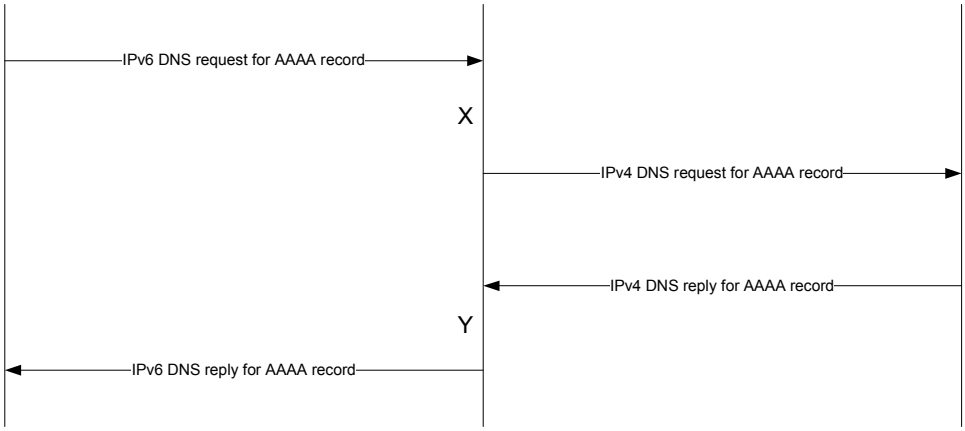
   b) What are the long and shortened notations for the IPv6 unspecified address?

   c) What is the IPv6 all-nodes link-local multicast address?

   d) How many addresses are in the IPv6 multicast address space?

5) [10] Ok, this is a tough problem but it has a real-world application so bear with me. A router that supports both IPv4 and IPv6 needs to provide client nodes access to a DNS server that supports both IPv4 (A) and IPv6 (AAAA) resource records. While the DNS server contains IPv6 DNS records, the server itself only supports IPv4. The client nodes only support IPv6. Explain how the router can proxy IPv6 DNS requests from the client nodes to the DNS server, and proxy IPv4 DNS replies from the server back to the client nodes. At a high level, there is a similarity to NAT, but be careful to make sure that there can be no ambiguity on the responses coming into the router. For what it is worth, I had to do this on a real product. Refer to the diagram below and describe what has to happen at steps X and Y, including what happens to any header and payload data.





6) [15] Consider the following TCP packet. There are at least 5 subtle “errors” in it. Barring checksum calculations and the media length, find these errors and describe each of them. You should assume that there is no payload.

```
**************************************************************************
                        Ethernet (1021838696.677283)
--------------------------------------------------------------------------
Hardware source:        00:80:3e:57:de:cf
Hardware destination:   00:50:04:32:0e:8f
Type / Length:          0x806 (IP)
--------------------------------------------------------------------------
                        IP Header
--------------------------------------------------------------------------
Version:                4
Header length:          5 (20 bytes)
TOS:                    0x00
Total length:           52
Identification:         40855
Fragmentation offset:   0
Unused bit:             0
Don't fragment bit:     0
More fragments bit:     1
Time to live:           125
Protocol:               6 (TCP)
Header checksum:        14942
Source address:         149.112.154.71
Destination address:    149.112.94.178
--------------------------------------------------------------------------
                        TCP Header
--------------------------------------------------------------------------
Source port:            22 (SSH)
Destination port:       22 (SSH)
Sequence number:        3447301758
Acknowledgement number: 2298423967
Header length:          5 (20 bytes)
Unused:                 0
Flags:                  PA
Window size:            17680
Checksum:               5971
Urgent:                 0
```

.