Programming Assignment 2

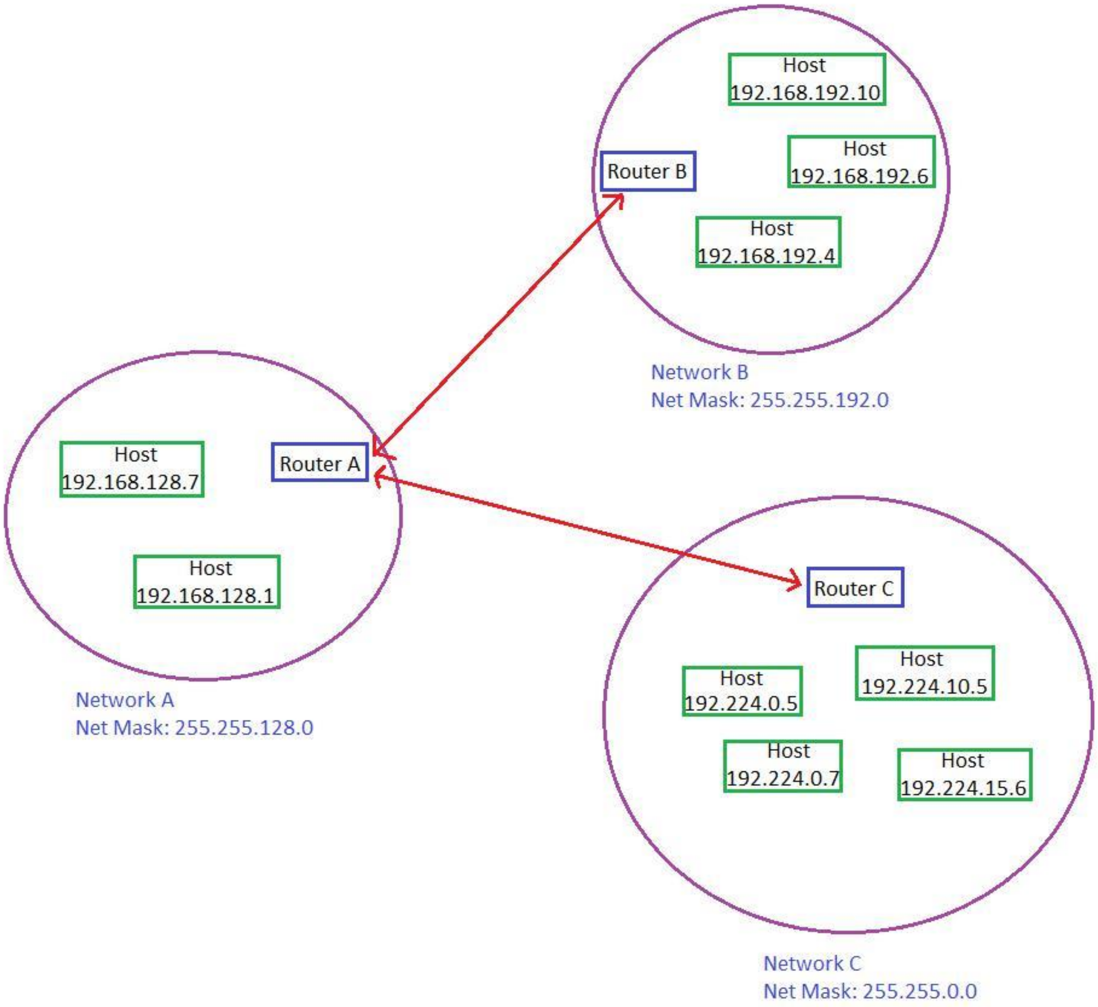
Due Feb 23rd 2016 Noon

Electronic submissions only

THE WORK MUST BE YOUR OWN!

If you are referencing any outside sources you must cite those clearly.

Note: No Extensions or Late Submissions will be accepted



In this assignment, you will **simulate subset** of tasks of simple packet router. Three networks A, B and C are defined in the figure mentioned above. Any two hosts within the same network can directly communicate. However, a host needs to contact its default router when it wants to communicate to an external host. Routers A, B and C are the designated default routers for networks A, B and C respectively. There are direct links between Router A and Router B, and Router A and Router C. But there is no direct link between Router B and Router C. In this assignment, you would simulate the routing task of **Router A**. You need to run a program that would receive incoming packets and make necessary routing decisions based on packet-destinations. You would run another Packet Generator program on the same machine that would generate and send packets to the router program. The source of each generated packet may be any host (pick any) from **network A** and packet destination would be a host (pick any) from a **network C**. Packet ID's of generated packets should be sequential numbers, and TTL values should be randomly chosen between 1 and 4 inclusive.

The **router program** will take three command-line arguments: 80 marks

<port number to listen to> <routing table file path> <statistics file path>

The router program will:

- read its routing table from the specified file
- The file will consist of multiple lines with the format:
  <subnet> <netmask> <nexthop>
  - The subnet and netmask would be in IPv4 dotted decimal. The nexthop value would 0, RouterB or RouterC. The three values will be separated by spaces. For example:
  - 192.224.0.0 255.255.0.0 RouterC.
  - A 0-value of nexthop means a direct delivery of a packet to a host within network A.
- listen to the specified UDP port
- accept simplified IP packets in the format:
  <packet ID>, <source IP>, <destination IP>, <TTL>, <payload>
  Example: 215, 192.168.192.4, 192.224.0.7, 64, testing

For each received packet, your program will:

- decrement the TTL field of the packet
  - if it reaches zero, the packet will be dropped and the "expired packets" counter will be updated
- figure out which of the entries in the routing table (if any) match the packet
  - it will do this by checking whether the destination address in the packet matches the subnet entry for a particular route, after being ANDed with the netmask of that entry in the routing table.
  - It will scan the routing table sequentially and will use the first route that matches (you can assume the table is sorted according to the longest-prefix first)
  - if there is no matching entry, the packet will be dropped and the "unroutable packets" counter will be updated
- if the value of the nexthop field for the entry is 0:
  - print the following:

Delivering direct: packet ID=<packet ID>, dest=<dest. IP address>

  - update the "delivered direct" counter

- if the value of the nexthop field is Router B or C
  - update the counter corresponding to the packets forwarded to that router

Note: whenever a routing decision about a packet is done, the packet would not be processed any further.

The **packet generator program** will take one command-line arguments:

<port number to connect to router program>

The program will:

- create packets and set <packet ID>, <source IP>, <destination IP>, <TTL>, <payload> as mentioned in the first paragraph
- sends packets to router program 20 marks

Error handling:

In addition to KeyboardInterrupt (used to terminate the program), you will need to handle any I/O-related exceptions.

If the routing table file does not exist or cannot be read, your program should print an error message and exit.

If your program is unable to listen to or connect to the specified UDP port it should print an error message and exit.

If an error occurs while receiving an incoming packet, it should print an error message and it should continue running (i.e. it should not exit in this case).