Router Application Specification

1. Function.

The function provided by this router application is to establish a route between server and client. It can receive packets from server (client) and forward these packets to client (server). It can also simulate packet loss and delay in the Internet. Students should make their server and client communicate with this Router application, and deal with abnormality of packet flows.

2. Router Application.

The Router application includes two files: Router.cpp and Router.h. You should create an independent VC++ project for this application.
Router application creates two UDP socks. The basic scenario of connections among router, server, and client is shown in fig. 1.
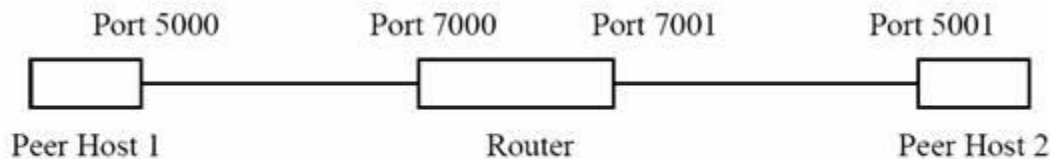

Fig. 1 Basic Scenario

It does not matter that peer host 1 (or host 2) is server or client. The key feature to establish the connections among them is to let each of them know its peer host's address and port number, because each host must know to which host and interface the packets should be sent. For example, Peer Host 1 must know Router's address and the predefined interface (port number), 7000 in this case. User will be asked to enter the name of peer host 1 and 2 at the beginning. All the port numbers shown in fig. 1 are predefined in Router header file, and can be modified by yourself:

#define ROUTER_PORT1 7000 //router port number 1
#define ROUTER_PORT2 7001 //router port number 2
#define PEER_PORT1 5000 //port number of peer host 1
#define PEER_PORT2 5001 //port number of peer host 2

Router can forward a packet from one interface to the other, and can also simulate flow abnormality as the real Internet. At the beginning, user will be asked to enter a drop rate (e.g., enter 20 for 20%) and a delay rate that control the percentage of loss and delay. If there is an incoming packet from peer host 1, Router can randomly choose from three options: forward it, drop it, or delay it.
If the Router decides to forward a packet, it will send this packet to an interface other than the interface that the packet is received. Then Router will display messages like:
    Router: Receive packet xxx from peer host 2
    Router: Send packet xxx received from peer host 2 to host 1

"xxx", start with 0, is the count of packet received from a host. These messages will be also written in a log file (log.txt).

The decision of dropping or delay will also be made randomly according to the drop rate and delay rate entered by user. If Router randomly chooses to drop a packet, router simply discards the packet, then displays and writes to the log file a message:

   Router: Packet xxx received from peer host 1 has been dropped by router!

If the packet will be delayed, Router will not forward the packet until another packet from the same source has been forwarded or the packet has been kept for more than 0.9 second (3 times of router time out cycle value). The messages from delayed packets are:

Router: Packet xxx received from peer host 2 has been delayed!
……
Router: Send delayed packet xxx received from peer host 2 to host 1

3. Debugging

To help you debug your programs, Router is designed in such a way: it will not stop for socket errors, except for errors of startup. All socket errors will be displayed for helping your debugging. The delay function will ensure that the students experience packet re-ordering (which is the purpose of the "delay" feature), but never have to deal with more than one out-of-order packet or with packets that are delayed more than three "cycle times". All information about packets will be display and written in the log file.