6. In the given simulator the network layer is empty. There are two minimal routines those simulate the delivery and receipt of packets from the network layer. In this project we will implement an upper layer to our existing simulator. We will call it "super" layer (instead of network layer).

7. (i) What is the frame size of the existing data link layer? (ii) What is the packet size of the existing network layer? (iii) What modification is needed in the program so that the network layer packet size can be increased to Z (Z=A+50) bytes?

8. Now modify the simulator so that Z (that you have calculated last time) bytes of network layer packets can be transmitted by the data link layer protocols.

9. Now we will add some functionality to the 'super' layer. We want to send large files. We will implement fragmentation and re-assembly operation in 'super' layer. Update the network layer routines so that data contained in two files can be exchanged over the network using bi-directional data link protocols. The name of the data files should be specifiable as arguments to the simulator program in the following form:

   "sim protocol events timeout pct_loss pct_cksum debug_flags file1 file2"

   Typing in the above command should allow the network layer to read the two files and send their contents to the other side using fixed sized (Z bytes) packets of the bi-directional data link layer protocols. Each receiver end should save the received content by adding a .cp extension to the file names (as *file1.cp* and *file2.cp* respectively). Since, we are taking advantage of layering, it should not matter, which particular data link layer protocol (4,5,or 6) we are using.

   Note, now the 'super' layer has to perform fragmentation of re-assembly. Also, the network layer can not send full Z data bytes in the packets. It needs to send the file name, the file-size, a packet sequence number, last packet marker, etc. for co-ordination. Here is your assignment:

   a. Design a "super" layer header format, with fields necessary to implement the above file transfer. Explain the use of each field.

   b. Update your program so that when requested by data-link layer, "super layer" can fill up appropriate values to the super layer header, append data values, and pass on the packets to the data link layer.

   c. Update your program so that when a packet is delivered by a data-link layer, "super layer" can extract the appropriate field values of the super layer header, perform re-assembly and save the data in the correct file.

d. Modify the simulator so that it stops and prints out the statistics when both the file transfers are complete, or the specified event ticks expires, whichever happens first.

Report+Submission: For this part of assignment, you need to submit the modified simulator code using the rule below. In the report you need to specify the design of the protocol, data structure and example command to run the modified simulator. (Note to grader: test the accuracy of implementation with the test1.html and test2.exe files).

10. Exchange the two given files "test1.html" and "test2.exe" using all protocol-4, protocol-5 and protocol-6. Compare the performance of protocol-4, protocol-5 (bi-directional communication with n-sliding window) and protocol-6 (bi-directional communication with n-sliding window, with selective retransmission) while transferring these files.

    a. Plot and compare the *number of retransmissions* required (y-axis) as a function of *timeout* when there is no packet loss

    b. Plot and compare the *payload rate* (y-axis) as a function of *packet loss* (assuming timer internal =X ticks).

    Report: You should provide two separate charts each with two plots. Explain the relationship between the payload rate and the three variables and how each of these protocols is affected. In the report explain the (a) Experiment Setup including the commands used to obtain the data, (b) Graphs/ Plots/Tables, and (c) Explanation of Data. (Note to grader: subtract: respectively 20, 50, and 30% if any of these are missing).

How to Submit:

In this assignment you have created a set of program files *.c and one project report.
On top of each file include your name, data and project number. Add:

```
/*************************************************
Name:
Date:
Project/Question Number:
NET CS 5/5201, Spring 2008
Instructor: KHAN, KSU
*************************************************/
```

You now need to mail all of them into one package using the following procedure:

%shar *.c *.h Makefile report.doc > shar.project2
%elm –s “SUBMIT PROJECT#2 for type your name here” (EMAIL ADDRESS OF TA) < shar.project2

Check thoroughly before you submit. If you need to re-send, for any reason inform TA beforehand. Keep a copy of all the files including shar.project1 in your directory. Do not modify them afterward. If need arises, TA may want to check these files. Any modification afterward (reflected in the file date) will result in late submission penalty.

Grading:

See notes to grader in the website.

Cheating and Copy:

Projects have to be done individually. If a copy is caught, all involved submissions (original as well as the copies) will be penalized. So it is your responsibility to guard your work. Secure the read/write access of your directories. Any copy will result in ZERO grade for the assignment for both party. Only exception is when you report the theft of your work in advance.