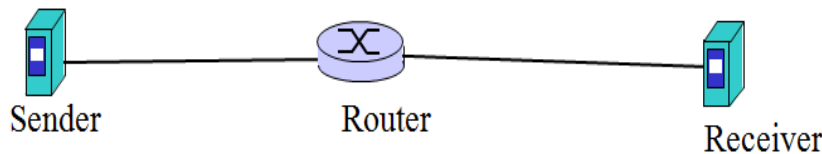


CS8220: Assignment 1: A NACK-only reliable transmission control Protocol (NTCP)

Due at 23:50pm, Feb. 8

Based on the content covered in the class, you are required to implement a simplified network store-forward simulating program. You can conduct the homework by yourself or work with another student as a group. You are required to set up your system with computers to emulate the sender, receiver and router, which are connected as shown in the following figure.



1. You are asked to design a reliable data transfer protocol that only uses negative acknowledgements (NACK). The sender operates in a selective repeat fashion which an infinite window size (you may assume an infinitely large sequence number space if necessary), and only retransmits a packet when it receives a NACK from the receiver. The channel may lose, or corrupt message and the delays are variable and unknown.
2. The sender will send **Y** packets via the router to the receiver. The receiver will reply with corresponding negative acknowledge packet as needed. The router in the middle will randomly drop a packet with a probability $(0.1 * X) \%$, where **X** is the last digit of your panther ID+ your partner's last digit of the Panther ID.

3. You have to:

- (1) Implement the protocols and programs running on the sender, router, and the receiver;
- (2) Submit your source codes to the dropbox in icollge.gsu.edu;
- (3) Demonstrate your program to the TA during the TA's office hour or make an appointment with TA;
- (4) Submit a hard-copy report of the assignment, which should include (no handwriting materials are acceptable):
 - a) The design of your protocol and program;
 - b) A plot showing that the total number of try to successfully transmit all the packets when changing **Y** from 150, 250, 450, 650, 850, 1000;
 - c) Find out what is the maximum throughput of your system
 - d) Discuss the factors that impact the throughput and how one can potentially increase the throughput performance;
 - e) The source codes that implement the whole process;