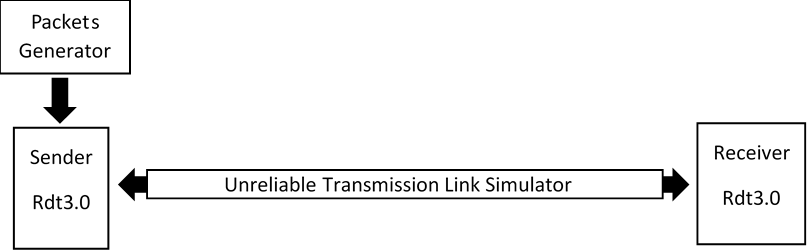
**Project2: Implementing Reliable Transfer Protocol**

**Due on Sunday March 4th @ 11:00 pm**

**Description**:

Write an application that implements the simplified rdt3.0 transport protocol discussed in class. Your implementation should include at least one sender and one receiver connected to each other. The communication channel is assumed to be **unreliable**.

You need to write a simple packets traffic simulator to avoid the necessity to test the protocol on a real network traffic, see the attached figure. The simulator should simulate all packets transmission scenarios including receiving packets corrupted or delayed by a receiver. In addition to the possibility that packets may get lost on the transmission channel.

During its running, rdt3.0 should print messages to two separate screens showing its responses to different packets delivery scenarios. Below is an output example:

|  |  |  |
| --- | --- | --- |
| ***Sender Screen*** |  | ***Receiver Screen*** |
| *.*  *.*  *.*  *send packet53*  *received ACK for packet53*  *send packet54*  *received ACK for packet54*  *send packet55*  *received ACK for packet54*  *resend packet55*  *received ACK for packet55*  *.*  *.*  *.*  *.*  *.*  *send packet101*  *time-out has been detected*  *resend the last ACKed packet*  *.*  *.*  *.*  *and so on.* |  | *.*  *.*  *.*  *packet53 received correctly*  *send ACK for packet53*  *packet54 received correctly*  *send ACK for packet54*  *packet55 received corrupted*  *resend ACK for packet54*  packet55 received correctly  send ACK for packet55  *.*  *.*  *.*  *.*  *.*  *.*  *.*  *.*  *.*  packet101 received correctly  send ACK for packet101  *.*  *.*  *.*  *.*  and so on. |

During your demo presentation, the protocol should keep **running indefinitely**, until you interrupt it by using whatever interruption method you choose.

To write the code, you can choose any programming language you are comfortable with. However, the submitted code must be runnable over Windows machines. Otherwise, you have to run the code on your own laptop.

**Deliverables**:

All deliverables below must be submitted (in a zip folder) to myCourses dropbox by the due date.

1. Compiled and runnable code
2. Source code files
3. Project report that include **at least** the following sections:
   * Cover page
   * Table of contents
   * Introduction (how rdt3.0 works)
   * Procedures (how your code addresses packets delay, lost and corruption)
   * Results and Discussion (you can use screenshots to show rdt3.0 reactions to various packets delivery scenarios)
   * Conclusion and future work!

***NOTE: The report must be written in high professional standards***. You can refer to the abundant number of online resources of professional report writing.

1. **Maximum 5** minutes presentation and demo must be scheduled with the course TA after the project deadline.

**Grading guidelines:**

1. Demo (running the code) 30%
2. Presentation (answering questions about 50%

how your code meets the project requirements)

1. Professionally written and documented code 10%
2. Project report 10%

**All questions should be posted on the myCourses forum.**

**GOOD LUCK**