CS 39006: Networks Lab
Assignment 7: Implement a Reliable Transport Protocol (RelTP) using Datagram Socket (C/C++)

## Date: 28th March, 2017

Assignment Statement:

The objective of this assignment is to implement a reliable transmission protocol (RelTP) using datagram socket. You need to implement a variant of Go-Back-N ARQ mechanism with timeout for the implementation of reliability and flow control.

In RelTP, the sender maintains a sequence number field in the packet header, and acknowledgement (ACK) packets are used by the receiver to acknowledge the receipt of a particular packet. If a sender transmits a packet, and then does not receive an ACK within a timeout value (say, t), then it assumes the packet to be lost in the network, and retransmits the packet.

In this protocol both the sender and the receiver also maintains a window. Let the window size be w packets. This indicates that the sender can transmit a maximum of w packets without waiting for an acknowledgement. Once the sender receives the acknowledgement, it slides the window accordingly. The acknowledgement (ACK) is cumulative acknowledgement, i.e., an ACK packet with sequence number n indicates that all the packets up to sequence number n has been received correctly, and the receiver is expecting packet with sequence number $n+1$.

The protocol is explained using the following diagram. Say, the window size is 4. Packet 3 is lost, and after the timeout, the sender retransmits the packets starting from packet 3. Note that the receiver does not accept packets 4, 5 and 6 unless it receives packet 3, therefore all packets starting from 3 needs to be retransmitted – therefore, the name of the protocol is Go-Back-N.
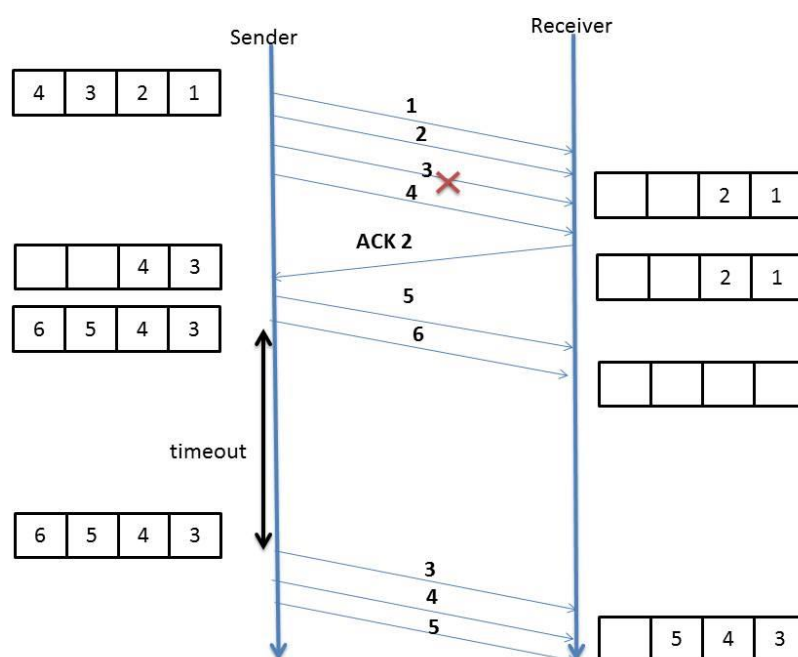
Your tasks for this assignment will be as follows.

1. Define the packet header structure. The header should contain a 16 bits sequence number field, 16 bits acknowledgement number field, and 1 bit control field. If control field = 0, then it is a DATA packet, else (control bit = 1) it is an ACK packet. You can keep additional fields in the header.

2. Implement RelTP with Go-Back-N ARQ mechanism. You should use datagram socket and build the protocol on top of that.

3. Test RelTP over a single-hop topology in Mininet where the sender and the receiver are connected through a direct link. Assume link bandwidth = 1 Mbps and delay = 1 ms. Change the loss rate between the sender and the receiver, and compute throughput and packet loss rate by transferring a file of size at least 100 MB. Compare these parameters for RelTP and the standard UDP protocol (you can implement standard UDP protocol using a datagram socket without reliability implementation). Plot graphs comparing the performance of RelTP and UDP with respect to average throughput and average packet loss rate.

4. Transfer a file of size at least 100 MB over the Mininet topology, and compare the file transfer time for RelTP and the standard UDP protocol, by varying the loss rate at the links. Plot a graph comparing the file transfer time of RelTP and UDP, with respect to the loss rate. Consider channel bandwidth = 10 Mbps, and delay = 1 ms.

Note:
1. Think of what can be a good choice of the timeout value.
2. You can assume fixed packet size for RelTP and UDP.
3. You can choose the size of the window as per your choice.
4. A circular queue can be an efficient data structure for implementation of the sliding window – explore this.

Submission Instructions:

You need to prepare a report that will contain the followings.
1. The design details of your reliable transport protocol (RelTP).
2. The design parameters (selection of window size, selection of timeout values, packet size etc.), and the rationale behind selecting those specific values.
3. Performance measures of RelTP and comparison with UDP.

You need to submit the report and relevant source files in a single compressed (tar.gz) file. Rename the compressed file as Assignment_7_Roll1_Roll2.tar.gz, where Roll1 and Roll2 are the roll numbers of the two members in the group. Submit the compressed file through Moodle by the submission deadline. The submission deadline is: **April 04, 2017 02:00 PM.**
Please note that your submission will be awarded zero marks without further consideration, if it is find to be copied. In such cases, all the submissions will be treated equally, without any discrimination to figure out who has copied from whom.