

CS 39006: Networks Lab

Assignment 6: Sliding Window ARQ with Congestion Control

Date: 1st March, 2018

Objective:

The objective of this assignment is to implement a congestion control algorithm over the sliding window ARQ protocol.

Submission Instructions:

The code needs to be written in either C or C++ programming language. You need to prepare a report that will contain the followings.

1. **Documentation** of the code along with **compilation and running procedure, sample input and sample output**

Also include a **makefile** to compile your code. You need to submit the report, code (only C/C++ source) and Makefile in a single compressed (tar.gz) file. Rename the compressed file as **Assignment_1_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: **March 8, 2018 02:00 PM**. Please note that this is a strict deadline and no extension will be granted.

Please note that your submission will be awarded zero marks without further consideration, if it is found to be copied. In such cases, all the submissions will be treated equally, without any discrimination to figure out who has copied from whom.

The objective of this assignment is to implement the **Slow-Start** congestion control algorithm over the Sliding Window ARQ protocol over UDP that was implemented in the lab test.

Slow-Start works as follows:

- 1) Sender starts sending packets with a small window size ($w=3$).
- 2) Sender doubles its window size in the following time slot, if all the packets of a current window are transmitted and acknowledged successfully.
- 3) In case there is any packet loss, the sender reduces the current window size by half in the following time slot.

Make sure you send a large enough file for the congestion to occur.

As you are mainly working with a single system, it is very difficult to get any congestion. However, you can simulate this congestion scenario by dropping some packets at receiver side. Server should accept a argument which enable this feature with a drop-probability. The user can send a parameter **drop-probability** to the server. The server will drop packets randomly according to this probability. This will help in testing your implemented algorithm.