CISC 4615 — Project 2

In the previous labs and project 1, we learned how to use TCP and UDP to connect different hosts. During the lectures, we discussed the details of TCP tramission, which mainly involves four major parts, path establishment, flow control, congestion control as well as connection closure.

In project 2, you will simulate an extremely simplified TCP transmission by using only the data field of the complete packet.

Implementation Details

Please note that this is not a real TCP implemention. We just want to simulate some important features in the TCP protocol.

Construct the REAL link

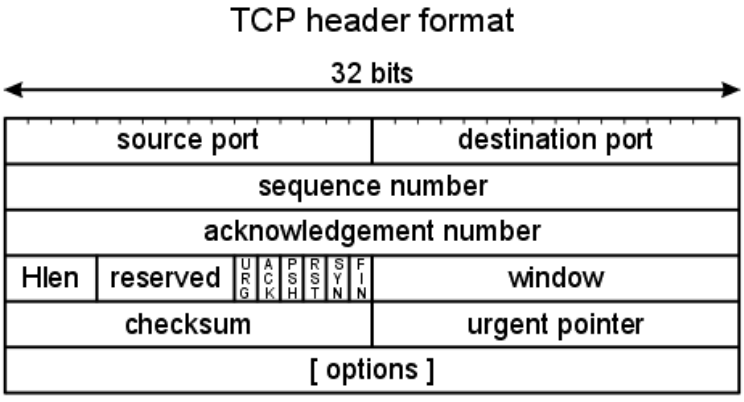
You are expected to construct a simple two-node network by using UDP.

$$A <---> B$$

The base code has done this part for you.

Prepare the packets

We learned the real TCP header in class.

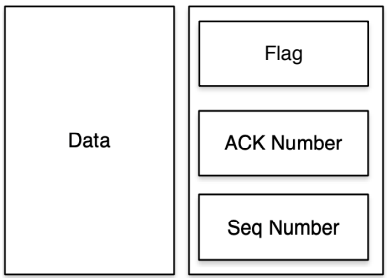


In our simplified simulation, we use three fields,

- Flag: The flag is used to identify the package type, e.g. SYN, ACK, FIN for connection establishment and closure.
- Sequence Number: The sequence number is used for in-order transmission.
- Acknowledgement Number: The acknowledgement number is used to acknowledge the previous segment and expect the next one.

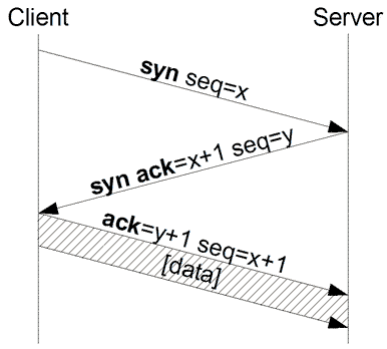
In addition to the simplified TCP header, we need to have data field.

Therefore, you are expected contruct a struct, which consists of four attributes, the tcp header (three fields) and the data (a string message acts as payload).



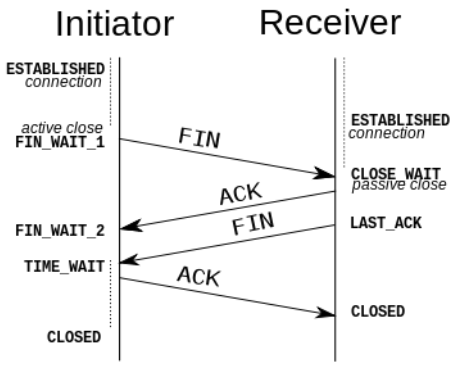
TCP connection establishment

This is the TCP three-way handshake protocol. You need to use SYN / SYN-ACK / ACK to construct the link in our simulation.



TCP connection closure

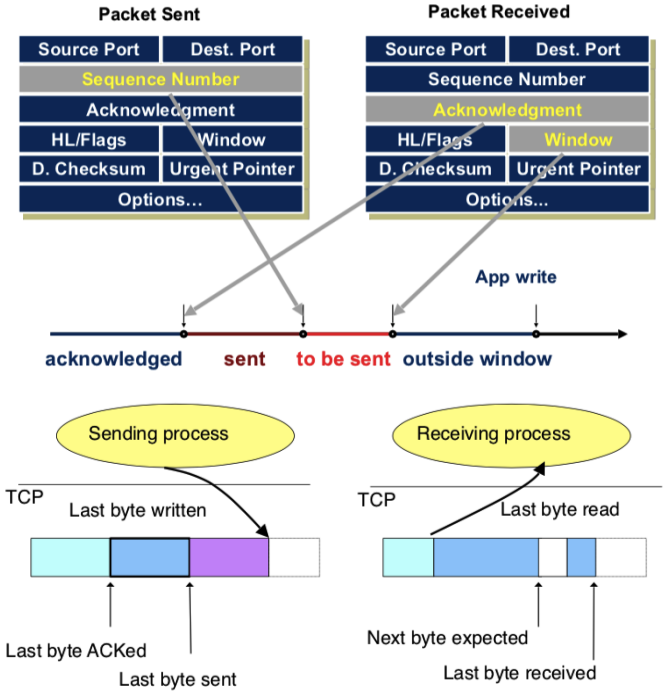
When you finished transmitting all the packets, you have to notify the receiver to close the connection.



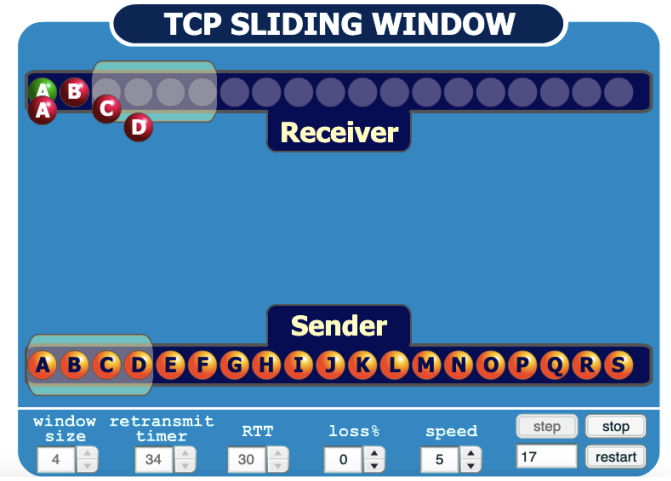
TCP Flow control with sliding window

We intensively discussed the sliding window algorithm in both Link Layer and TCP during the lectures. Your code should complete two parts.

In the original algorithm, you have to maintain a buffer for the outgoing and incoming data and necessary pointers as shown below.



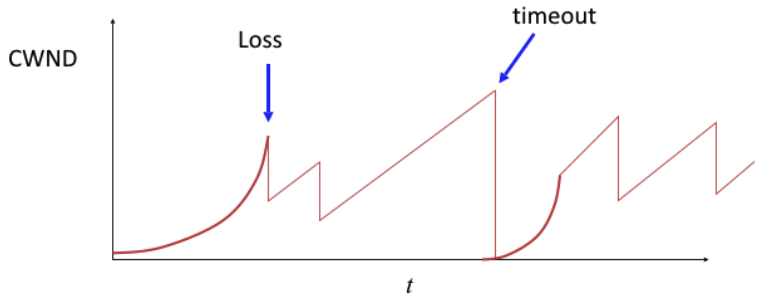
To simplify the problem, we assume that there is no package loss in our network. Therefore, you don't have to pointers and buffers. You can check out the demo from the following figure and link.



`http://www2.rad.com/networks/2004/sliding_window/`

TCP Congestion control (Bonus)

As we discussed in the lecture, TCP uses a network congestion-avoidance algorithm that includes various aspects of an additive increase/multiplicative decrease (AIMD) scheme, with other schemes such as slow start and congestion window to achieve congestion avoidance.



You are going to implement additive increase/multiplicative decrease (AIMD) with exponential slow start and congestion window. Please keep in mind that

$$TCP\_Window = min\{congestion\_window, receiver\_window\}$$

To make them work, you have to simulate package loss and the congestion window size starts from 1.

Project Workload

You are expected to implement three modules,

1. TCP Connection Establishment;
2. TCP Connection Closure;
3. Flow Control without any package loss;

When receive each package, your program should print out propriate text messages on the screen, e.g. Received a SYN message, preparing a SYN-ACK message; Received a DATA message, preparing an ACK message;

The receiver's window size is set to 4, which means that you can only send four letters in a window. For example, if you want to send "Hello, World!", you have to send it in 4 windows, "Hell", "o, W", "orld" as well as "!".

The size will not change in the transmission. In addition, we assume that the sender knows this size after the connection establishment.

To test your program, you are going to transmit the following string.

"I love Fordham University in the New York City."

Grading Rubric

You should complete the lab in a group (max 4).

(25%) TCP Connection Establishment;
(25%) TCP Connection Closure;
(30%) Flow Control;
(15%) A detailed report, including your system design, data sturctures, demos. It should be as detail as possible;
(5%) Submission format;
(10%) Congestion Control (You can just complete part of them to earn partial credits.);

Submission

By the end of April 27th, each group has to email me with a subject line **CISC4615 Project2 Submission** and a zip file that contains two folders(Bonus is optional),

- Project2: your code, your detailed report
- Bonus: your code, your report including your design of package loss and other details