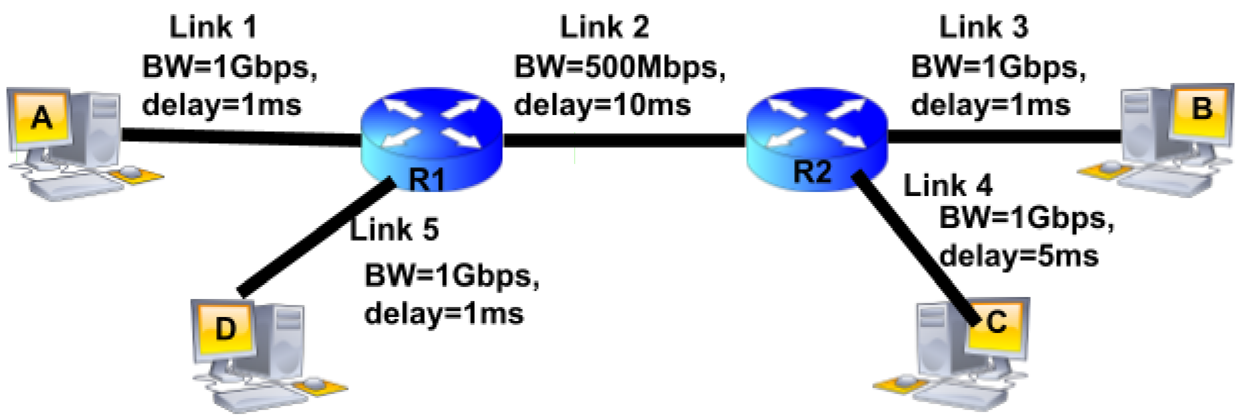
Assignment-02 **CS 433 Computer Networks (2022-23)** **100 points**
Submission Deadline: 10-Nov-2022 11:59PM.

Part I: Port over the Network Application built in Assignment-01 to Mininet: (75 points)

1. Run the code that you built as (PartI, Q1 in Assignment-1) in Mininet. Consider a simple linear topology with a single switch (s1) and two hosts (h1 and h2). ***(10 pts)***
    a. Report all the changes made to run your code on Mininet. ***(5 pts)***
    b. Performance - Did it improve or degrade? Your Observations on why? ***(5 pts)***

2. Implement a Mininet network as shown below, where you instantiate different desired (parameterized) topologies, where one of the host nodes say A acts as a server and subset of the remaining nodes act as clients. ***Refer http://mininet.org/walkthrough/***. ***(55 pts)*** ***Note: For below experiments, you should get an average over at-least 3 iterations. For Iperf, run at-least 3 iterations with duration >=120s. Hint: For throughput plot, explore wireshark, I/O graphs.***

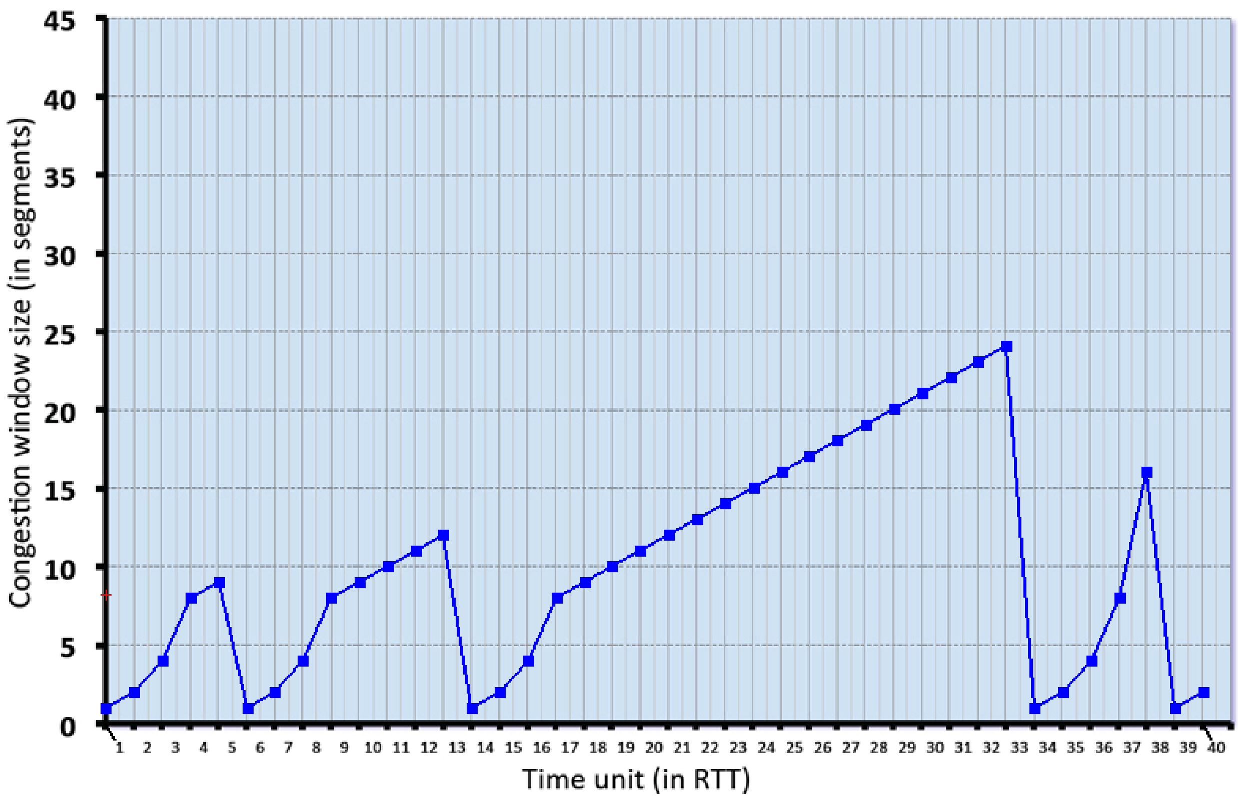


    a. **Implementation** of the custom topology using the Mininet APIs. ***(10 pts)***
    b. **Latency**: Measure RTT for following node pairs: AB, AC, AD, BC, BD. ***(5 pts)***
    c. **Throughput**: Run the iperf client on host B and iperf server on host A. Report/plot throughput over time.***(5 pts)***
    d. **Delay Impact**: Run the iperf client on host C and iperf server on host A. Report/Plot if any impact on the throughput. ***(5 pts)***
    e. **Concurrency(i)**: Run the iperf client on hosts B, C and iperf server on host A. Report/plot throughput for B,C. Run the clients simultaneously. ***(10 pts)***
    f. **Fairness(ii)**: Change the link4 delay to 1ms. Report/plot throughput for B,C, when the clients are run simultaneously. Any difference and why? ***(10 pts)***
    g. **Concurrency(iii)**: Run Iperf client on Hosts B,C, D and observe the throughput when the clients are run simultaneously. Any difference and why? ***(10 pts)***

3. **Impact of Packet Loss**: Consider either a simple linear topology or the topology shown in above figure. Configure the link loss parameter. (example: sudo mn --link tc, ***loss=1***, bw=??--topo linear,1). In each iteration, vary the link loss percentage as 1%, 2% and 5% respectively. In each iteration, report/plot the achieved throughput and report your observations. ***(10 pts)***

Part II Networking Problems: (25 points)

4. Consider the figure below, which plots the evolution of TCP's congestion window at the beginning of each time unit (where the unit of time is equal to the RTT); In the abstract model for this problem, TCP sends a "flight" of packets of size cwnd at the beginning of each time unit. The result of sending that flight of packets is that either (i) all packets are ACKed at the end of the time unit, (ii) there is a timeout for the first packet, or (iii) there is a triple duplicate ACK for the first packet. In this problem, you are asked to reconstruct the sequence of events (ACKs, losses) that resulted in the evolution of TCP's cwnd shown below.



   a. Plot the SlowstartThreshold for each RTT interval ***(5 pts)***
   b. Indicate the possible congestion control scheme (Tahoe or Reno?) ***(5 pts)***
   c. Time interval, #packets when the maximum number of packets are in flight.***(5 pts)***
   d. Indicate min,max number of packet losses possible.***(5 pts)***
   e. Indicate changes if loss at Interval 33 were to be due to 3Dup Acks ***(5 pts)***

Part III: Grace Mark (2 points)

5. Run the TCP iperf server or the server you implemented in Assignment-1 on the Kali Linux VM. And Run the iperf client or the client you implemented in Assignment-1 on your host machine (Windows/Linux/MAC). Does it work?
    a) If yes, report the results obtained and compare with the earlier results where client/server are run on the same host or in reverse mode.
    b) If not, explain why? Explore if you can get it work and detail the steps/configurations made to make it work.