|  |  |
| --- | --- |
|  | **University of Dhaka**  **Department of Computer Science and Engineering**  **CSE 3111 – Computer Networking Laboratory Credits: 1.5 Batch: 26/3rd Year 1st Sem 2022**  Instructors: Prof. Dr. Md. Abdur Razzaque (AR), Mr. Md. Mahmudur Rahman (MRR), Mr. Md. Ashraful Islam (MAI) and Mr. Md. Fahim Arefin (FA) |

**Lab Assignment # 5**

|  |
| --- |
| **Name of the Experiment:** Implementation of TCP flow control and congestion control algorithm (TCP Tahoe).  **Objectives :**   * To gather knowledge about how TCP controls the flow of data between a sender and a receiver * To learn how TCP controls and avoids the congestion of data when a sender or receiver detects a congestion in the link in-between them. ( TCP Tahoe) |

**Theory:**

TCP is one of the protocols of the transport layer for network communication. TCP provides reliable, ordered, and error-checked delivery of a stream of bytes between applications running on hosts communicating via an IP network. TCP is connection-oriented, and a connection between client and server is established before data can be sent. The server must be listening (passive open) for connection requests from clients before a connection is established. Three-way handshake (active open), retransmission, and error-detection adds to reliability. Thus TCP can maintain various operations to establish perfect communications between a pair of hosts, e.g connection management, error detection, error recovery, congestion control, connection termination, flow control, etc. In this lab, we will have a look at the flow control mechanism and congestion control mechanisms of the TCP protocol.

TCP uses a sliding window flow control protocol. In each TCP segment, the receiver specifies in the receive window field the amount of additionally received data (in bytes) that it is willing to buffer for the connection. The sending host can send only up to that amount of data before it must wait for an acknowledgement and window update from the receiving host.

Chart, pie chart

Description automatically generated

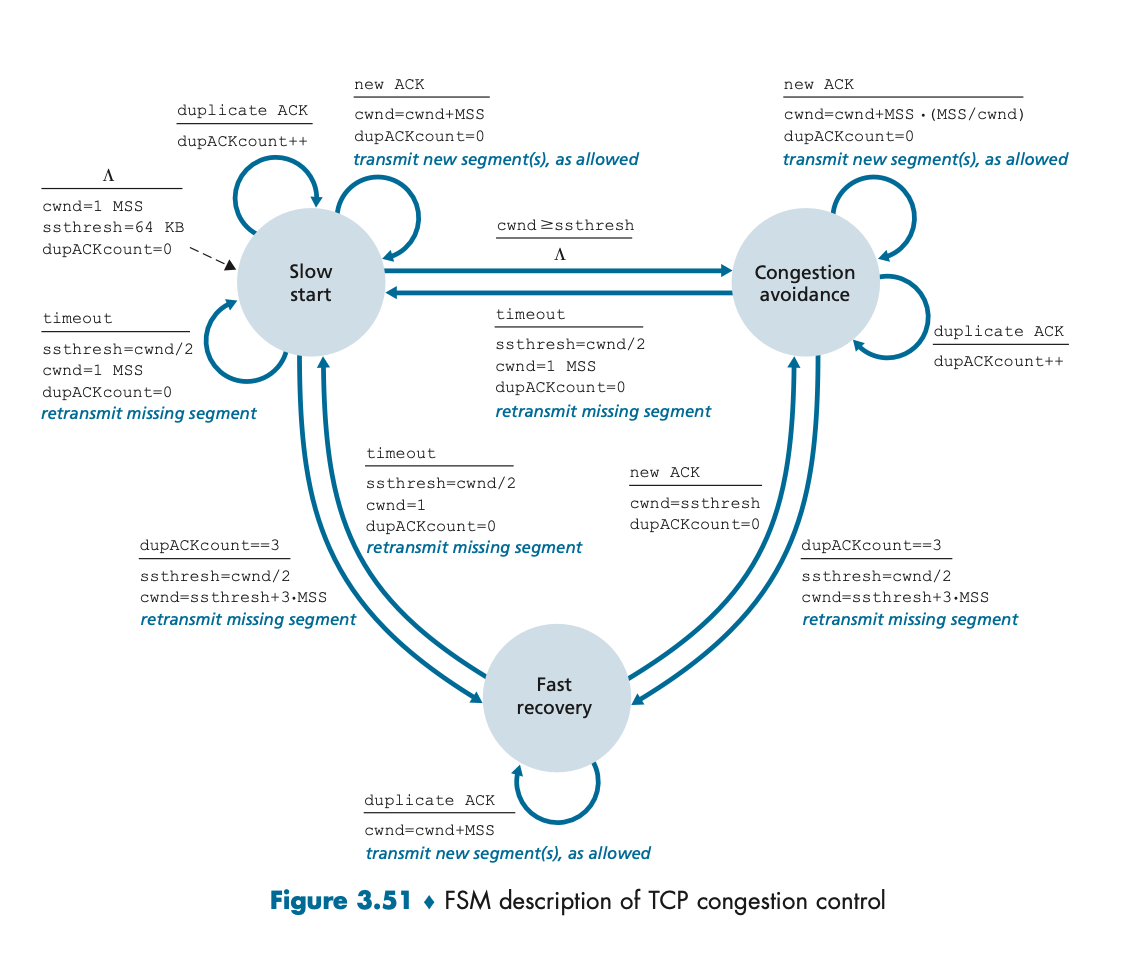
TCP Tahoe is a congestion control algorithm that is used to manage the flow of data in a network and prevent congestion. It was one of the first congestion control algorithms implemented in TCP and is still widely used today.

The congestion window is one of the factors that determines the number of bytes that can be outstanding at any time. It is a means of stopping a link between the sender and the receiver from getting overloaded with too much traffic and it is calculated by estimating how much congestion there is between the sender and receiver.

Slow-start is part of the congestion control strategy used by TCP. Slow-start is used in conjunction with other algorithms to avoid sending more data than the network is capable of transmitting, that is, to avoid causing network congestion.

**The congestion window (CWND) is maintained by the sender.** Note that this is not to be confused with the TCP window size which is maintained by the receiver.

A sample pictorial behavior of TCP Tahoe is given below :



**Tasks :**

Task 1: Implement TCP Flow Control

- Configure the server to use TCP flow control by setting the receive window size. The receive window size determines how much data the receiver is willing to accept before sending an acknowledgment.

- Configure the clients to use Cumulative Acknowledgment. Cumulative Acknowledgment means that the receiver sends an acknowledgment for the highest sequence number it has received in order, and it assumes that all packets up to that sequence number have been received.

- Test the TCP flow control by sending data from the clients to the server.

Task 2: Implement TCP Congestion Control

- Configure the server to use TCP congestion control by setting the congestion window size. The congestion window size determines how much data the sender can send before receiving an acknowledgment.

- Configure the clients to use the EWMA equation to calculate the TimeOut value. The EWMA equation is used to estimate the round-trip time (RTT) and calculate the retransmission timeout (RTO) value.

- Implement the slow start algorithm on the clients. The slow start algorithm increases the congestion window size exponentially until it reaches a threshold value.

- Implement the congestion avoidance algorithm on the clients. The congestion avoidance algorithm increases the congestion window size linearly after the threshold value has been reached.

- Implement the fast retransmit algorithm on the clients. The fast retransmit algorithm retransmits a packet if it receives three duplicate acknowledgments for the same packet.

- Test the TCP congestion control by sending data from the clients to the server

Task 3: Analyze Results

- Analyze the captured network traffic to evaluate the performance of the TCP flow control and TCP congestion control algorithms.

- Compare the results under different network conditions and analyze how the algorithms affect the network

**Algorithm:**

TCP Tahoe is a congestion control algorithm that uses three mechanisms to manage the flow of data in a network: slow start, congestion avoidance, and fast retransmit. The following is a simplified version of the TCP Tahoe algorithm:

In the server side when a connection is established, server send Maximum Segment Size. Then waits for client so that client can send data.

1. Initialization:

   - Set the congestion window (cwnd) to 1 Maximum Segment Size (MSS).

   - Set the slow start threshold (ssthresh) to a large value, e.g., the size of the receive window.

   - Set the duplicate ACK counter to 0.

2. Slow Start:

   - For each ACK received, increase cwnd by 1 MSS.

   - If cwnd >= ssthresh, switch to congestion avoidance.

   - If a packet loss is detected (either by receiving 3 duplicate ACKs or a timeout), set ssthresh to cwnd/2, set cwnd to 1 MSS, and go to step 4 (fast retransmit).

3. Congestion Avoidance:

   - For each ACK received, increase cwnd by (MSS \* MSS) / cwnd.

   - If a packet loss is detected (either by receiving 3 duplicate ACKs or a timeout), set ssthresh to cwnd/2, set cwnd to 1 MSS, and go to step 4 (fast retransmit).

4. Fast Retransmit:

   - Retransmit the lost packet.

   - Set the duplicate ACK counter to 0.

   - Return to step 2 (slow start).

The TCP Tahoe algorithm repeats these steps for the duration of the connection, adjusting the congestion window and retransmitting lost packets as necessary to manage the flow of data and prevent congestion in the network.

**Deliverables:**

- Source code for the client-server application with TCP Tahoe

- Run simulations with TCP Tahoe and collect performance metrics such as throughput, packet loss rate, and round-trip time (RTT).

- A report that documents the design and implementation of the algorithms, as well as the performance comparison under different network conditions [ Do not include code in the report ]

References:

<https://www.javatpoint.com/flow-control-vs-congestion-control>

<https://www.geeksforgeeks.org/difference-between-flow-control-and-congestion-control/>