 **Eastern Mediterranean University**

 **Computer Engineering Department**

**CMPE344-CMSE346 - Computer Networks– Lab. 5**

**Title: TCP (Transmission Control Protocol) Using OPNET Modeler**

**OBJECTIVES:**

This lab is designed to demonstrate the congestion control algorithms implemented by the Transmission Control Protocol (TCP). The lab provides a number of scenarios to simulate these algorithms. You will compare the performance of the algorithms through the analysis of the simulation results.

**OVERVIEW:**

The Internet’s TCP guarantees the reliable, in-order delivery of a stream of bytes. It includes a flow-control mechanism for the byte streams that allows the receiver to limit how much data the sender can transmit at a given time. In addition, TCP implements a highly tuned congestion-control mechanism. The idea of this mechanism is to throttle the rate at which TCP sends data, to keep the sender from overloading the network.

The idea of TCP congestion control is for each source to determine how much capacity is available in the network so that it knows how many packets it can safely have in transit. It maintains a state variable for each connection, called the congestion window, which is used by the source to limit how much data the source is allowed to have in transit at a given time. TCP uses a mechanism called additive increase/multiplicative decrease. With this feature, TCP decreases the congestion window when the level of congestion goes up and increases the congestion window when the level of congestion goes down. TCP interprets timeouts as a sign of congestion. Each time a timeout occurs, the source sets the congestion window to half of its previous value. This halving corresponds to the multiplicative decrease part of the mechanism.

The congestion window is not allowed to fall below the size of a single packet (the TCP maximum segment size, or MSS). Every time the source successfully sends a

congestion window worth of packets, it adds the equivalent of one packet to the congestion window; this is the additive increase part of the mechanism. TCP uses a mechanism called slow start to increase the congestion window “rapidly” from a cold start in TCP connections. It increases the congestion window exponentially rather than linearly. Finally, TCP utilizes a mechanism called fast retransmit and fast recovery. Fast retransmit is a heuristic that sometimes triggers the retransmission of a dropped packet sooner than the regular timeout mechanism.

In this lab, you will set up a network that utilizes TCP as its end-to-end transmission protocol, and you will analyze the size of the congestion window with different mechanisms.

**PRE-LAB ACTIVITIES:**

Read and check the following:

❍ TCP Connections

❍ TCP Multiplexing

❍ TCP Buffering and Sequencing

❍ User Datagram Protocol (UDP)

**PROCEDURE:**

**Create a New Project:**

1. Start OPNET IT Guru Academic Edition · Choose New from the File menu.

2. Select Project and click OK · Name the project <lab5\_TCP >, and the scenario

No\_Drop · Click OK.

3. In the Startup Wizard: Initial Topology dialog box, make sure that Create Empty Scenario is selected · Click Next · Select Choose From Maps from the Network Scale list · Click Next · Choose USA from the Map List · Click Next twice · Click OK

**Create and Configure the Network:**

Initialize the network:

1. The Object Palette dialog box should now be on the top of your project space. If it is not there, open it by clicking. Make sure that the internet\_toolbox item is selected from the pull-down menu on the object palette.
2. Add to the project workspace the following objects from the palette: Application Config, Profile Config, an ip32\_Cloud, and two subnets.
3. To add an object from a palette, click its icon in the object palette · Move your mouse to the workspace · Click to drop the object in the desired location · Right-click to finish creating objects of that type.

*Note: The ip32\_cloud node model represents an IP cloud supporting up to 32 serial line interfaces at a selectable data rate through which IP traffic can be modeled. IP packets arriving on any cloud interface are routed to the appropriate output interface based on their destination IP address. The RIP or OSPF protocol may be used to automatically and dynamically create the cloud’s routing tables and select*

*routes in an adaptive manner. This cloud requires a fixed amount of time to route each packet, as determined by the Packet Latency attribute of the node.*

1. Close the palette.
2. Rename the objects you added as shown.
3. Save your project.



Configure the applications:

1. Right-click on the Applications node · Edit Attributes · Expand the Application Definitions attribute and set rows to 1 · Expand the new row · Name the row FTP\_Application.

a. Expand the Description hierarchy · Edit the FTP row as shown (first, you will need to set the Special Value to Not Used while editing the attributes shown):



2. Click OK twice, and Save your project.

Configure the profiles:

1. Right-click on the Profiles node · Edit Attributes · Expand the Profile Configuration attribute and set rows to 1.

a. Name and set the attributes of row 0 as shown · Click OK.



Configure the West subnet:

1. Double-click on the West subnet node. You get an empty workspace, indicating that the subnet contains no objects.

2. Open the Object palette and make sure that the internet\_toolbox item is selected from the pull-down menu.

3. Add the following items to the subnet workspace: one ethernet\_server, one ethernet4\_slip8\_gtwy router, and connect them with a bidirectional 100\_BaseT link · Close the palette · Rename the objects as shown.

*Hint: The ethernet4\_slip8\_gtwy node model represents an IP-based gateway supporting four Ethernet hub interfaces and eight serial line interfaces.*



4. Right-click on the Server\_West node · Edit Attributes:

a. Edit Application: Supported Services · Set rows to 1 · Set Name to FTP\_

Application · Click OK.

b. Edit the value of the Server Address attribute and write down Server\_West.

c. Expand the TCP Parameters hierarchy · Set both Fast Retransmit and Fast Recovery to Disabled.

1. Click OK, and Save your project.

Now, you have completed the configuration of the West subnet. To go back to the top level of the project and click the Go to next higher-level button.

Configure the East subnet:

1. Double-click on the East subnet node. You get an empty workspace, indicating that the subnet contains no objects.

2. Open the Object palette and make sure that the internet\_toolbox item is selected

from the pull-down menu.

3. Add the following items to the subnet workspace: one ethernet\_wkstn, one ethernet4\_slip8\_gtwy router, and connect them with a bidirectional 100\_BaseT link · Close the palette · Rename the objects as shown.



4. Right-click on the Client\_East node · Edit Attributes:

a. Expand the Application: Supported Profi les hierarchy · Set rows to 1 · Expand the row 0 hierarchy · Set Profi le Name to FTP\_Profi le.

b. Assign Client\_ East to the Client Address attributes.

c. Edit the Application: Destination Preferences attribute as follows:

d. Set rows to 1 · Set Symbolic Name to FTP Server · Edit Actual Name · Set rows

to 1 · In the new row, assign Server\_West to the Name column.

5. Click OK three times, and then Save your project.

You have now completed the confi guration of the East subnet. To go back to the project space, click the Go to next higher-level button.

Connect the subnets to the IP Cloud:

1. Open the Object palette.

2. Using two PPP\_DS3 bidirectional links, connect the East subnet to the IP Cloud and the West subnet to the IP Cloud.

3. A pop-up dialog box will appear, asking you with what to connect the subnet to the IP Cloud with. Make sure to select the routers.

4. Close the palette.



Choose the Statistics

1. Right-click on Server\_West in the West subnet, and select Choose Individual Statistics from the pop-up menu.
2. In the Choose Results dialog box, choose the following statistic:
3. TCP Connection · Congestion Window Size (bytes) and Sent Segment Sequence Number.



*Hint: OPNET provides the following capture modes:*

*All values: This mode collects every data point from a statistic.*

*Sample: This mode collects the data according to a user-specifi ed time interval or sample count. For example, if the time interval is 10, data is sampled and recorded*

*every 10th second. If the sample count is 10, every 10th data point is recorded. All other data points are discarded.*

*Bucket: This mode collects all of the points over the time interval or sample count into a “data bucket” and generates a result from each bucket. This is the default mode.*

1. Right-click on the Congestion Window Size (bytes) statistic · Choose Change

Collection Mode · In the dialog box, check Advanced · From the drop-down menu, assign all values to Capture mode as shown · Click OK.

1. Right-click on the Sent Segment Sequence Number statistic · Choose Change Collection Mode · In the dialog box, check Advanced · From the drop-down menu, assign all values to Capture mode.

3. Click OK twice, and Save your project.

4. Click the Go to next higher-level button.

**Configure the Simulation**

Here we need to configure the duration of the simulation:

1. Click on and the Configure Simulation window should appear.

2. Set the duration to 10.0 minutes · Click OK.

**Duplicate the Scenario**

In the network we just created, we assumed a perfect network with no discarded packets. Also, we disabled the fast retransmit and fast recovery techniques in TCP. To analyze the effects of discarded packets and those congestion-control techniques, we will create two additional scenarios.

1. Select Duplicate Scenario from the Scenarios menu, and give it the name Drop\_NoFast · Click OK.

a. In the new scenario, right-click on the IP Cloud · Edit Attributes · Assign 0.05% to the Packet Discard Ratio attribute · Click OK, and Save your project.

2. While you are still in the Drop\_NoFast scenario, select Duplicate Scenario from the Scenarios menu, and give it the name Drop\_Fast.

a. In the Drop\_Fast scenario, right-click on Server\_ West, which is inside the West

subnet · Edit Attributes · Expand the TCP Parameters hierarchy · Enable the Fast-Retransmit attribute · Assign Reno to the Fast Recovery attribute.

*Hint: With fast retransmit, TCP performs a retransmission of what appears to be the missing segment, without waiting for a retransmission timer to expire. After fast retransmit sends what appears to be the missing segment, congestion avoidance but not slow start is performed. This is the fast recovery algorithm. The fast retransmit and fast recovery algorithms are usually implemented together (RFC 2001).*

3. Click OK, and Save your project.

**Run the Simulation**

To run the simulation for the three scenarios simultaneously:

1. Go to the Scenarios menu · Select Manage Scenarios.

2. Change the values under the Results column to <collect> (or <recollect>) for the three scenarios. Compare with the following figure.

3. Click OK to run the three simulations. Depending on the speed of your processor, this task may take several seconds to complete.

4. After the simulation runs complete, click Close · Save your project.

**View the Results**

To view and analyze the results:

1. Switch to the Drop\_NoFast scenario (the second one) and choose View Results from the Results menu.

2. Fully expand the Object Statistics hierarchy and select the following two results:

Congestion Window Size (bytes) and Sent Segment Sequence Number.

*Hint: To switch to a scenario, choose Switch to Scenario from the Scenarios menu or just press Ctrl <scenario number>.*



3. Click Show. The resulting graphs should resemble the ones that follow.



1. To zoom in on the details in the graph, click and drag your mouse to draw a rectangle, as shown in the preceding figure.

The graph should be redrawn to resemble the following one:



Notice the Segment Sequence Number is almost flat with every drop in the congestion window.

5. Close the View Results dialog box and select Compare Results from the Result menu.

6. Fully expand the Object Statistics hierarchy as shown and select the following result: Sent Segment Sequence Number.



1. Click Show. After you zoom in, the resulting graph should resemble the one shown here.



**Important warning:**

The lab is now completed. Show your result to Lab Assistant.

(in order to grading)

**Homework:**

1. Design the same project and Create another scenario as a duplicate of the Drop\_Fast scenario. Name the new scenario Q4\_Drop\_Fast\_Buffer. In the new scenario, edit the attributes of the Client\_East node and assign 65535 to its Receiver Buffer (bytes) attribute (one of the TCP Parameters). Generate a graph that shows how the Congestion Window Size (bytes) of Server\_West gets affected by the increase in the receiver buffer. (Compare the congestion window size graph from the Drop\_Fast scenario with the corresponding graph from the Q4\_Drop\_ Fast\_Buffer scenario.)